Title	安全なアプリケーションに向けての暗号に関する研究
Author(s)	Mamun, Mohammad Saiful Islam
Citation	
Issue Date	2014-09
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/12291
Rights	
Description	Supervisor:宮地　充子，情報科学研究科，博士

Studies on Cryptographic Solutions to Secure Applications

by

MAMUN, Mohammad Saiful Islam

submitted to
Japan Advanced Institute of Science and Technology
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Supervisor: Professor Atsuko Miyaji

School of Information Science
Japan Advanced Institute of Science and Technology

September, 2014

Abstract

Secure applications protect valuable information and defend every vulnerability. The goal of a secure application design is to create a cost effective system where information is securely protected. Cryptography is one of the effective tools that has powerful implications for information security. Since cryptographic solutions are continuously evolving, algorithms that were once considered secure are no longer secure now in practice. Therefore, poorly deployed systems are being threatened by increasing adversarial processing power, low-cost devices, weaker cryptographic algorithms, new demand of security and privacy issues, and technological advances. This has lead the US and Japan government to launch special programmes and bodies to define cryptography standards, specifications and recommendations to cope with the security and privacy requirement of the future. This theses presents our research results on the design and analysis of cryptographic solutions for Vehicle Ad hoc NETwork (VANET) and low cost Radio Frequency IDentification (RFID) systems.

Motivated by the recent attention on exploiting group signature approach in the design of VANET security scheme, we attempt to integrate all the potential properties of group signature in an individual scheme, so that it can best meet the demand and needs of the wide range of VANET services. To this end, we propose a new group signature model that is more application friendly, optimally secure with a relaxed privacy definition to satisfy practical privacy requirement of VANETs. Moreover, we investigate the feasibility of implementing batch verification of group signatures into a real life VANET environment. In addition, we improve an existing batch verification system on identity based group signature and determine where and when batch verification may be infeasible in practice.

Inspired to realize ubiquitous computing, machine perception and the rapidly growing trend in insecurity and terrorism, the RFID technology plays an indispensable role in various fields. With the use of tags and transponders (tracking & tracing), RFID technology is seeking to venture into the transport and logistics systems, pharmaceutical and clothing industry as well as monitoring and safeguarding the citizen. However, the exclusive features of RFID introduces new security and privacy concern from the end users' view point and resource restriction into the tag from the engineering perspective. Security concerns in the form of authentication of tags and reader and privacy concerns related to undercover tag/communication tracking of tagged items. Today's RFID system facilitates the real-time tracking of physical items in the supply chain. This enables the physical data flow of a tagged item with its location to be matched with the information flow in the enterprises information management systems. The weak privacy protection may jeopardize the entire supply chain exposed to industrial espionage, while vulnerable security may lead to the acts of eco-terrorism and economic sabotage. However, we first identified the major prior works in the area of RFID security such as tag authentication, tag ownership transfer, RFID-enabled supply chain path authentication etc. To this end, we adopted a new, growing and promising direction in the lightweight cryptographic research, namely Hopper-Blum (HB)-family protocol based on the Learning Parity from Noise (LPN) problem. Since the inner computations in the HB-family protocol comprises only matrix vector multiplications over GF(2) they are extremely efficient and may even be suitable for practical RFID applications. Meanwhile the security is equivalent to well-known hardness assumptions from coding theory and lattices. We ideated the demand of efficient, robust, forward secure mutual authentication protocol for RFID systems in HB-family settings. We propose two mutual authentication protocols at this end: one is between a tag and a back-end RFID reader/server. The other protocol, that may follow the former one, is among the RFID entities where an RFID reader and a back-end server are not identical. To address the ownership transfer problem in a large inventory system, we build a new, improved model consisting of several Semi Trusted Parties (STPs) and a trusted server. Our model can ease the ownership process for the consumers in the remote location, and allows simultaneous transfer ownership of multiple tags from one owner to another. Our construction uses a new variant of Homomorphic Aggregated signature, a lightweight searchable encryption, Field LPN and pseudo-inverse matrix as cryptographic primitives. Finally, we propose a path authentication protocol for an RFID-enabled supply chain. Compared to Elliptic curve Elgamal Re-encryption based construction our Homomorphic Message Authentication Code on Arithmetic circuit based solution offers a new privacy direction to the path privacy with an efficient and effective label of security and prevention of counterfeiting.

Our innovation has the potential to pave the way for more secure RFID-enabled services. All the secure and privacy-preserving protocols will enable RFID and vehicle industries to implement confidently and take advantage of emerging opportunities.

Acknowledgments

I would like to express my sincerest gratitude to two lovely ladies who played a significant role in this journey. One is my adviser Professor Atsuko Miyaji for her excellent supervision during this work. Her rich ideas and wide expertise in the field of security replete me with confidence, lead and motivate me to explore new direction in the world of cryptography. Her critical thinking, positive aptitude and constant guidance towards this research was rewarding for me. The other lady is my beloved wife for her patience, unflagging care, trust, and inspiration. She deserves special acknowledgement for her devotion, sacrifice and commitment to lifelong learning. Without you, this dissertation would have never taken shape.

I would like to convey my special thanks to Professor Hiroaki Takada at Nagoya University for serving as my sub-theme supervisor and external examiner and for providing me many valuable suggestions. I am also grateful to Associate Professor Kazumasa Omote for his support and help during and beyond my PhD program. My heartiest gratitude goes to my theses examination committee, Professor Hajime Ishihara, Professor Ryuhei Uehera for their valuable comments, their attention and effort on this theses.

I am deeply indebted to Associate Professor Yuichi Futa, Assistant Professor Jiageng Chen for their supportive suggestions and exchanging ideas. I am particularly happy to have such all the wonderful members of Miyaji Lab of JAIST for their constant support to remain a pleasant and smooth research environment. I would like to thank many of my friends and colleagues in JAIST by whom I was absolutely exhilarated.

We are very much grateful to all the anonymous reviewers of the conferences and journals for their precious comments and fruitful advices. This work has been partially supported by JAIST Foundation under Graduate Research Program (GRP), Japan Association for Mathematical Sciences Foundation, Japan Telecommunication Association Foundation (TAF) and NEC (C&C) Foundation, Japan.

Dedication

This theses is dedicated to all the Japanese people inside and outside the lab who helped me find what I needed. You are some of His finest gifts to so many people in the world who are blessed to be able to love you back. I love you forever.

Contents

Chapter 1

Introduction

1.1 Background

Our contribution can be split broadly into two major fields: vehicular network security and RFID system security. The former mainly focuses on the security model, efficiency and privacy issues in VANET. The latter examines a broad range of research on the security and privacy issues of commercial RFID application. Whilst both fields are promising and deal secure communication, they are different in terms of application role, security architecture, privacy definition and threat model etc.

RFID System Security. Recently wireless technologies are developing rapidly to construct smart communications with digital data. Networked devices are now automatically communicating among themselves without human interaction in order to carry out efficient information transaction.

A Radio Frequency IDentification (RFID) tag containing an unique identification numbers uses radio waves to transmit data at a distance. Passive RFID tag having no battery power lays dormant until it gets in contact with an RFID reader. Nowadays passive RFID tag is used at vast areas such as key-less entry, real-time location service, supply chain management, electronic passports, tracking inventory of manufacturers, tracking of patients and surgery tools in hospital, cashless point of sale, and access control, to identify things and transmit information when necessary. According to developers and vendors, RFID technology is moving to support new technologies such as Internet of Things (IoT), Machine-to-Machine (M2M) architectures where every physical object (embedded RFID) would have its own unique identity (IP address) encoded into the microchip. Prompted by the promising future goals, RFID technologies are evolving rapidly at the time, with regard to applications, performance and standards. However, due to the *wireless* nature of RFID technology, RFID system actuates various security and privacy issues concerning its owners and holders without any knowledge or consent of its users. Preventing unauthorised access to the owner data (confidentiality), tag tracing (linkability), identification of the owners (anonymity) are some classical security threats to RFID systems. This leads to design protocols in such a way that they could be adapted to the new requirements both in terms of security and privacy.

VANET Security. Vehicular Network (known as VANET) employing vehicles as mobile nodes in a Mobile Ad-hoc NETwork (MANET) is a specialization of multi-hop ad hoc network paradigm well motivated by the socio-economic value of advanced Intelligent Transportation Systems (ITS) aimed at reducing the traffic congestions, the high number of traffic road accidents, etc. The research area of VANET security is where ad hoc network security can be brought to their full potential. In order to assure public safety on the roads, safety traffic applications (s.t., collision avoidance, road obstacle warning, safety message disseminations), progress toward future autonomous vehicle, and rapid proliferation of vehicular communications via bluetooth, wifi, and cellular connectivity, VANET technologies are turning tightly incorporated into critical safety systems and are establishing suitable security architectures that can resist potential security and privacy threats. This is concerned with the design and analysis of the security aspects such as authentication and key management, threat model, security architecture, privacy issues for large-scale vehicular networks.

1.2 Motivation

Vehicle network security. Most of the prevention-based security mechanisms in VANETs [8, 9, 20, 24, 46, 56, 19, 21, 44] exploit digital signature as cryptographic primitive. Group Signature is a specialized digital signature that can be directly used to authenticate vehicular communication anonymously without generating pseudonyms [1, 3, 6, 7, 22, 23]. Note that, pseudonyms in VANETs [20, 24, 46, 56] are adopted to hide vehicles' real identity in order to ensure vehicle privacy. We found that existing group signature security models cannot support all the required secure applications in VANETs. In addition, stringent privacy of Group signature resists some real-life application to achieve. Therefore, we attempt to integrate all the potential group signature properties [8, 48, 22, 51, 53, 70]in a single scheme that can best meet the application demands of a large scale VANETs. Moreover, we relax stringent privacy definition of group signature [59] and propose an optimally secure and private and application-friendly scheme *"Secure VANET Applications with a refined Group Signature"* in Chapter 4.1. Since most safety-critical applications have stringent delay requirement [15], verifying huge signatures at a time is challenging [16]. Batch verification, where batch of signatures can be verified together, is one of the solutions to solve this problem [1, 3, 7, 8]. We observe that batch verification is not always feasible for VANET environment and choosing appropriate group signature sometimes depends on efficient batch verification system. To this end, we improve an existing batch verification system [1] and then analyze the feasibility of exploiting batch verification. Our scheme *"An efficient batch verification system for VANET"* in Chapter 4.2 describes an algorithm to determine the maximum number of signatures to batch at a time and a signature scheduling algorithm (by following single machine job scheduling algorithm [35]) if batch verification is not feasible. Finally, we find that although random oracle model is weaker security notions [74], there is no group signature scheme in the standard security model proposed for VANET. The main reason is signature size and verification cost. At this point, we choose a group signature in the standard security model [74], extend it with necessary properties (e.g. opening soundness [75], linkability [61], revocability [57]) for VANET and propose a simplified group signature scheme from standard security model *"A multi-purpose Group Signature for VANET under standard model"* in Chapter 4.3.

RFID system security. Security basis of RFID authentication protocols can be roughly divided into three fields: factorization and discrete logarithm based schemes [79, 184], elliptic curve cryptography (ECC) based scheme [84, 85, 183] and learning parity from noise (LPN) based scheme [104, 89, 90, 91, 92, 80]. In this theses, we followed the later one, LPN-based scheme (known as HB-family protocol). Note that, both factorization and ECC based scheme offers average case hardness. ECC based scheme offers smaller key size indeed. In compare to the former two, LPN based scheme has several advantages such as it offers faster computation with the same security parameter, worst case hardness, security against attacks using quantum computers etc. Motivated by the aforementioned advantages, we investigate HB-family protocols and found that there is no firmly secure and privacy preserving mutual authentication protocol under LPN problem. Therefore, we propose *"A privacy-preserving RFID authentication protocol"* in Chapter 5.1, a man-in-the-middle (MIM) attack-free mutual authentication protocol from subspace LPN problem. Later we found that mutual authentication based HB-family protocols [104, 113] cannot be used directly for insecure reader-server channel. But embedding RFID reader modules into a wireless device such as smart phone is a new research direction in the RFID inventory system [81, 126, 83, 125]. At this point, we extend our former authentication protocol and design a fully mutual authentication protocol *"An RFID authentication protocol in insecure reader-server channel"* in Chapter 5.2 where all the entities tag, reader and server can authenticate themselves among each other. RFID inventory system experiences many security and privacy oriented application. Ownership transfer is one of the significant problems among them. We found several ownership transfer models based on trusted party (TP) [120, 115, 145, 135, 136, 137]. Both with or without TP [134, 115, 143, 145] have their own drawbacks. To satisfy new application model (like issuer verification in [121]) and alleviate current shortcomings [116, 148, 139, 154, 120], we propose a semi-trusted party (STP) based RFID tag ownership transfer protocol *"A Scalable and Secure RFID Ownership Transfer Protocol"* in Chapter 5.3. Finally, we observe that ordinary tag authentication protocols cannot satisfy special security and privacy requirement of RFID-enabled path authentication [156, 149]. Depending on path verification nature there are two kind of path authentication protocols: static [150, 151] and dynamic [156, 152]. We concentrate on static path authentication protocols and propose *"An RFID Path Authentication Protocol"* in Chapter 5.4. Compared to existing Elliptic curve Elgamal Re-encryption (ECElgamal) based solution [151], our Homomorphic Message authentication Code on arithmetic circuit (HomMAC) based solution offers less memory storage (with limited scalability) and no computational requirement on the reader.

1.3 Summary of our contribution

Secure VANET Applications with a refined Group Signature. This work proposes an application-friendly group signature (GS) model for wireless ad hoc network like Wireless Sensor Networks (WSN) or Vehicle ad hoc Network (VANET). Our new GS properties can be used to carry out potential solution to some real life problem. We modify Boneh, Boyen and Shacham (BBS) short GS to meet a restricted, but arguably sufficient set of privacy properties. In particular, we aggregate several GS properties like linking, direct opening, message-dependent opening (MDO), revoking, batch-verification in a *single* scheme. Our link manager can link messages whether they are coming from the same messages or not without colluding to the *opener*. It helps relaxing strong privacy properties of GS to a lightly lesser one that fit certain application requirement. We introduce a new application to the ad hoc network security, that is, value-added service provider (VSP) with the help of MDO properties and redesign the traditional GS-friendly VANET architecture. Our revocation algorithm adapts both rekeying and verifier-local revocation (VLR) approaches to revoke illegitimate signers. Finally, we present an optional batch verification system to expedite signature verification. Note that all these properties have already been shown in the literature scatteredly. The novelty of our proposal stems from accumulating all these properties in a single GS scheme that can best fit to the application demand.

An efficient batch verification system for VANET. In this work, we improve an existing batch verification system on ID based group signature and also compare the performance achieved. We also analyze the best possible value of the number of signatures to batch at a time for a large scale VANET and present a scheduling algorithm for signature verification where batch verification cannot be implemented efficiently.

A multi-purpose Group Signature for VANET under standard model. This work adapts a new group signature (GS) scheme to the specific needs of a vehicular ad hoc network (VANET). We modify the Groth GS in order to meet a restricted, but arguably sufficient set of privacy properties. Note that Groth GS is secure in the dynamic group signature model of Bellare, Shi, and Zhang (BSZ) without relying on random oracle Model (ROM). Although some authentication schemes using GS are proposed for VANET, none of them satisfy all the desirable security and privacy properties. Either they follow GSs that rely on ROM, or unable to satisfy potential VANET application requirements. In particular, *link management* which allows any designated entities (e.g., RSUs in VANET) to link messages, whether they are coming from the same vehicle or a certain group of vehicles, without revealing their identities. Besides that *opening soundness* property prevents malicious accusations by the opener against some honest member of the group . By using this property, we propose a new secure application framework for value-added service providers (VSPs) in VANET. Meanwhile, a real-world VANET deployment must provide a mean to *revoke* system privileges from fraudulent vehicles like the traditional Public Key infrastructure (PKI). However, in order to achieve the aforementioned security properties together in VANET, we propose a new GS model where linkability, sound opening and revocability properties are assembled in a single scheme. The novelty of our proposal stems from extending the Groth GS by relaxing strong privacy properties to a scheme with a lightly lesser privacy in order to fit an existing VANET application requirements. In addition, we partially minimize the Groth GS scheme to expedite efficiency.

A privacy-preserving RFID authentication protocol. This work presents an authentication protocol of an RFID system where both the tag and reader are authenticated mutually. Optimal performance requirement, considering storage and computation constraints of low-cost tags, keeping security and privacy policies intact are some major challenges in recent research in this area. However, in order to restrain optimal security and privacy requirement in a cost effective manner, several light-weight authentication solutions have been proposed for RFID system. HB-family is one of the most promising protocol series, based on the hardness of the Learning Parity with Noise (LPN) problem. We propose a secure and private mutual authentication protocol of HB-family to meet the demand of low-cost tags. It is composed of Subspace Learning Parity from Noise problem (SLPN) and Pseudo-inverse matrix properties, both of them significantly reduce the cost in terms of computation and hardware requirements. In addition, we compare our result with other existing HB-like and ordinary RFID authentication protocols according to their construction primitives and security and privacy achievements.

An RFID authentication protocol in insecure reader-server channel. This protocol is an extension of our previous mutual authentication protocol where back-end server and RFID reader are identical. In this work, we present a secure collaborative mutual HB-like authentication protocol for an RFID system where both channels tag-reader and reader-server are considered to be insecure and thus upgrade the present HB-family protocol. More precisely, we introduce a new variant of an HB-like protocol where the complete RFID system is authenticated under LPN-based commitment scheme that can provably resist major security and privacy threats by taking advantage of properties of perfect computational hiding commitment scheme, pseudo-inverse matrix based short signature, and randomized Hill cipher. In addition, through detailed security and privacy analysis, we show that our scheme achieves required security and privacy properties under the standard model.

A Scalable and Secure RFID Ownership Transfer Protocol. In this work, we consider scenarios related to ownership transfer of RFID tags in a large inventory system. In this work, we propose a new ownership transfer model with mutual authentication protocol from Ring LPN problem that leverages the reader authentication phase to incorporate Semi-Trusted Parties (STP) seamlessly in RFID ownership transfer protocol. Employing STPs could ease the ownership transfer process for the consumers in the remote location. More precisely, we introduce a new variant of Learning Parity from Noise (LPN) based mutual authentication scheme for efficient ownership transfer protocol where ownership of multiple tags can be transferred from one owner to another by taking advantages of an efficient homomorphic aggregated signature (HomSig) and pseudo-inverse matrix properties. To the best of our knowledge, this is the first RFID ownership transfer protocol from LPN problem that is secure, private and scalable under standard model.

An RFID Path Authentication Protocol. RFID ownership transfer protocols consider how to securely transfer the ownership of the RFID tag to the other reader. In an RFID-enabled supply chain, where items are outfitted with RFID tags, path authentication based on tag enables the destination checkpoints to validate the route that a tag has already accessed. In this work, we propose a novel, efficient, privacy-preserving path authentication system for RFID-enabled supply chains. However, unlike previous schemes, we allow computational ability inside the tag that consents a new privacy direction to *path privacy* proposed by Cai *et al.* in ACNS′12. In addition, we customize a polynomial-based authentication scheme (to thwart potential tag impersonation and Denial of Service (DoS) attacks), so that it fits our new path authentication protocol.

1.4 Organization

We introduce our notations and preliminaries regarding cryptographic primitives in Chapter 2. We briefly discuss prior related works on VANET and RFID systems' security and privacy in Chapter 3.

We present our all VANET related works in Chapter 4. Firstly, Secure VANET applications with a refined group signatures in Chapter 4.1. Secondly, an efficient batch verification system in Chapter 4.2. Finally, a group signature under standard security model in Chapter 4.3. Subsequently, all the RFID related works are accumulated in Chapter 5. We describe our first RFID authentication protocol in Chapter 5.1. Extended version of our first authentication protocol (fully mutual authentication protocol) is presented in Chapter 5.2. Then we exhibit our ownership transfer protocol for RFID systems in Chapter 5.3. Finally, we provide an RFID-enabled Path authentication protocol for supply chain in Chapter 5.4. At last, Chapter 6 concludes with some future research direction.

Chapter 2

Preliminaries

This chapter presents the notations, mathematics and cryptographic background, cryptographic primitives and building tools used throughout the theses.

2.1 Notation

In this section, we define the notations used in this theses.

Table 2.1: Notations used in Chapter 5.1 & 5.2

λ	security parameter
$\mathbb{Z}_p$	set of integers modulo an integer $p \geq 1$
$\mathbf{T_{id}} \in \mathbb{Z}_2^l$	l-bit unique ID of a tag
$\mathbf{I_i} \in \mathbb{Z}_2^k$	k-bit index of the tag during session i
$\mathbf{P_i} \in \mathbb{Z}_2^{k \times k}$	$k \times k$-bit matrices as the session key for the reader during session i
$\mathbf{S'} \in \mathbb{Z}_2^{k \times l}$	$k \times l$-bit matrices as the permanent *secret* key between the server and the tag
$\mathbf{S_i} \in \mathbb{Z}_2^{k \times v}$	$k \times v$-bit matrices as the *session* key between the server/reader and tag during session i
$\mathbf{Q}, \mathbf{V} \in \mathbb{Z}_2^{k \times k}$	$k \times k$-bit randomly generated non-singular binary matrices
$s, s' \in \mathbb{Z}_2^v$	v-bit random binary vector generated by the reader and the tag respectively
$\sigma_i \in \mathbb{Z}_2^k$	k-bit lightweight signature on a challenge message s_i during session i
$\mathbf{w}(s)$	hamming weight of a vector s
τ	parameter of the Bernoulli error distribution Ber_τ where $\tau \in]0, 1/4[$
τ'	verifier acceptance threshold (Tag/Reader) where $\tau' = 1/4 + \tau/2$
$e \in \mathsf{Ber}_\tau^k$	k-bit binary vector from Bernoullli distribution Ber_τ^k such that, $\mathsf{Pr}[e = 1] = \tau$
$[\mathbf{A}]^\mathbf{T}$	transpose of a matrix $\mathbf{A}$
$\mathbf{A^{-1}}$	inverse of a matrix $\mathbf{A}$
$\mathbf{A}^+$	pseudo-inverse of a matrix $\mathbf{A}$
$\oplus, \cdot, \Vert, \bigvee$	bitwise XOR operation, inner product operation, concatenation of two vectors, logical OR operation
$(x_{\downarrow} y)$	derived vector from x by deleting all the bits $x[i]$ where $y[i] = 0$
$\lfloor x \rceil$	the nearest integer to x
$]a, b[$	$x \in \mathbb{R}$ s.t., $a < x < b$

Table 2.2: Additional notations used in the Chapter 5.3

$\mathcal{R}_{cur}, \mathcal{R}_{new}$	current and new reader corresponding to a tag
$\mathbf{uid}_n$	secret identifier of the new owner $\mathcal{U}_n$
c_i	shared secret between the reader and the tag for a session i
T	a unique identifier of a tag $\mathcal{T}$ over Field F.
$\widehat{T}$	tag index stored in the main server
F^*	multiplicatively invertible elements of a field F
$\pi_j : \{0,1\}^\lambda \to F$	a mapping to F, such that $\forall s, s' \in \{0,1\}^\lambda, \pi(s) - \pi(s') \in F/F^*$ if $c = c'$

2.2 Cryptographic primitives

2.2.1 Known Mathematical Facts

Definition 1 *A **multiplicative group** $\mathbb{G}$ is a set together with an operation '·' that combines any two elements x and y to form another element xy or $(x \cdot y)$. $(\mathbb{G}, \cdot)$ must satisfy the following requirements known as the group axioms*

- $\forall x, y \in \mathbb{G}, xy \in \mathbb{G}$.
- $\forall x, y, z \in \mathbb{G}, x(yz) = (xy)z$.
- $\exists 1 \in \mathbb{G}$, *s.t.*, $\forall x \in \mathbb{G}, x\ 1 = 1\ x = x$.
- $\forall x \in \mathbb{G}, \exists x^{-1} \in \mathbb{G}$, *s.t.*, $xx^{-1} = x^{-1}x = 1$.
- $|\mathbb{G}|$ *denotes the* order *of the group* $\mathbb{G}$ *or the number of elements in* $\mathbb{G}$.
- *A group* $\mathbb{G}$ *is called* cyclic *if* $\exists g \in \mathbb{G}$, *s.t.*, $\forall x \in \mathbb{G}$, $\exists a \in \mathbb{Z}$, $g^a = x$

Definition 2 *Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of prime order p, where possibly $\mathbb{G}_1$=$\mathbb{G}_1$. g_1 is a generator of $\mathbb{G}_1$ and g_2 is a generator of $\mathbb{G}_2$. Suppose ψ is an isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, with $\psi(g_2) = g_1$. **Bilinear groups** are a set of three algebraic groups, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, together with a non-degenerate bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = p$, such that*

- *Bilinear: for all $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}, e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$,*
- *Non-degenerate: $e(g_1, g_2) \neq 1$ (i.e., $e(g_1, g_2)$ is a generator of $\mathbb{G}_T$),*
- *Computable: There exists an efficiently computable algorithm for computing ψ, e.*

Definition 3 ***Pairing-based cryptography** is the use of a pairing between elements of two cryptographic groups ($\mathbb{G}_1$, $\mathbb{G}_2$) to a third group ($\mathbb{G}_T$) such that $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ in order to construct cryptographic systems. If the same group ($\mathbb{G}_1$, $\mathbb{G}_1$) is used for the first two groups, the pairing is called symmetric and is a mapping from two elements* *of one group to an element from a second group. For instance, in groups equipped with a bilinear mapping such as the Weil pairing or Tate pairing, generalizations of the computational DiffieHellman problem are believed to be infeasible while the simpler decisional DiffieHellman problem can be easily solved using the pairing function.*

Definition 4 *Galois Field (*GF*), named after Evariste Galois, is known as finite fields. Let* $\mathsf{GF}(p^n)$ *be a finite field where* $p \in \mathbb{P}$ *and* $n \in \mathbb{Z}^+$*. Then, the order of the field is* p^n*,* p *is called the characteristic of the Field, and the degree of polynomial of each element is at most* $n-1$*. For instance,* $\mathsf{GF}(2^3)$ *is Finite Field where* $\mathsf{GF}(2^3) = \{0, 1, 2, 2+1, 2^2, 2^2+1, 2^2+2, 2^2+2+1\} = \{0, 1, 2, 3, 4, 5, 6, 7\}$*. Hence, The order of the Field* $|\mathsf{GF}(2^3)| = 8$ *where the maximum degree of polynomial of each element is* 2 *evaluated at 2.*

The multiplicative group of a ***finite field*** $\mathsf{GF}(p^n)$*, denoted by* F_{p^n}*, is defined modulo an irreducible polynomial* $f(X)$ *of degree* n *over* F_p*. Clearly, let* $g(p)$ *and* $f(p)$ *be the polynomials in* $\mathsf{GF}(p^n)$ *and* $m(p)$ *be an irreducible polynomial of degree at least n in* $\mathsf{GF}(p^n)$*. Then,* $h(p) = (g(p) \cdot f(p) \bmod m(p))$*. The multiplicative inverse of* $f(p)$*, denoted by* F^**, is given by* $i(p)$ *such that* $(f(p) \cdot i(p)) \pmod{m(p)} = 1$*. We use binary Field* $\mathsf{GF}(2^n)$ *or* F_{2^n} *that can be implemented efficiently.*

2.2.2 Computational Hardness Assumptions

Let $\mathcal{G}$ be a probabilistic polynomial-time algorithm that takes a security parameter 1^λ as input and generates a parameter $(p, \mathbb{G}, \mathbb{G}_T, e, g)$ of *bilinear groups*, where p is a λ-bit prime. $\mathbb{G}$ and $\mathbb{G}_T$ are groups of order p, g is a generator of $\mathbb{G}$. That is, $\mathbb{G} = \langle g \rangle$ is a finite cyclic group of order p with generator g s.t., order of $\mathbb{G}$, $|\mathbb{G}| = p$ and $\lambda = log_2|\mathbb{G}|$. And e is a bilinear map: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.

Random oracle. A random oracle is a probabilistic polynomial time algorithm that for each input $x \in \{0,1\}^m$ returns a uniformly random output $y \in \{0,1\}^n$ where $m, n \in \mathbb{N}$. More clearly, random oracle starts with an empty look-up table T. When queried with an input x, it first checks whether it is available in the table $y = \mathsf{T}(x)$. If not, it chooses and returns a uniformly random value $y \in \{0,1\}^n$ and sets $\mathsf{T}(x) = y$.

The DL assumption. Let $g \leftarrow \mathbb{G}$, $a \leftarrow \mathbb{Z}_p$. The Discrete Logarithm (DL) problem in $\mathbb{G}$ is stated as follows. Given (g, g^a), output (a). The advantage of a probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$ against DL problem is defined as

$$\mathrm{Adv}_{\mathcal{A}}^{\mathsf{DL}}(\lambda) = \Pr[\mathcal{A}(g, g^a) = a].$$

We say that the DL assumption holds if $\mathrm{Adv}_{\mathcal{A}}^{\mathsf{DL}}(\lambda)$ is negligible for any algorithm $\mathcal{A}$.

The DDH assumption. Let $g \leftarrow \mathbb{G}, (a, b, c) \leftarrow \mathbb{Z}_p$. The decisional Diffie-Hellman problem (DDH) problem in $\mathbb{G}$ is stated as follows. Given (g, g^a, g^b, g^c), output 1 if $c = ab$, otherwise 0 if $c = r$. The advantage of an algorithm $\mathcal{A}$ against the DDH problem is defined as

$$\mathrm{Adv}_{\mathcal{A}}^{\mathsf{DDH}}(\lambda) = |\Pr[\mathcal{A}(g, g^a, g^b, g^c) = 1 \mid c = ab] - \Pr[\mathcal{A}(g, g^a, g^b, g^c) = 1 \mid c = r]|.$$

We say that the decision linear assumption holds if $\mathrm{Adv}_{\mathcal{A}}^{\mathsf{DDH}}(\lambda)$ is negligible for any PPT algorithm $\mathcal{A}$.

The ℓ-DHI assumption. Let $g \leftarrow \mathbb{G},\ x \leftarrow \mathbb{Z}_p$. We say that ℓ-Diffie-Hellman Inverse (DHI) holds in $\mathbb{G}$ if for every PPT algorithm $\mathcal{A}$ and for $\ell = \mathsf{poly}(\lambda)$, the advantage of algorithm $\mathcal{A}$ against ℓ-DHI problem

$$\mathrm{Adv}_{\mathcal{A}}^{\mathsf{DHI}}(\lambda) = \Pr[\mathcal{A}(g, g^x, \cdots, g^{x^\ell}) = g^{1/x}].$$

is negligible for any PPT algorithm $\mathcal{A}$.

The co-CDH assumption. Let $g_1 \leftarrow \mathbb{G}_1, g_2 \leftarrow \mathbb{G}_2, a \leftarrow \mathbb{Z}_p$. The co-computational Diffie-Hellman problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is stated as follows. Given (g_1, g_2, g_1^a), output (g_2^a). The advantage of a probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$ against co-CDH problem is defined as

$$\mathrm{Adv}_{\mathcal{A}}^{\mathsf{co-CDH}}(\lambda) = \Pr[\mathcal{A}(g_1, g_2, g_1^a) = g_2^a].$$

We say that the co-CDH assumption holds if $\mathrm{Adv}_{\mathcal{A}}^{\mathsf{co-CDH}}(\lambda)$ is negligible for any algorithm $\mathcal{A}$.

The co-DBDH assumption Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2) \leftarrow \mathcal{G}(1^\lambda)$ and $(a, b, r) \leftarrow \mathbb{Z}_p$. The co-decisional Bilinear Diffie-Hellman problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is as follows. Given $(g_1, g_2, g_1^a, g_2^b, z)$, output 1 if $z = e(g_1, g_2)^{ab}$, otherwise 0 if $z = e(g_1, g_2)^r$. The Advantage of an algorithm $\mathcal{A}$ against the co-DBDH-problem is defined as

$$\mathrm{Adv}_{\mathcal{A}}^{\mathsf{co-DBDH}}(\lambda) = |\Pr[\mathcal{A}(g_1, g_2, g_1^a, g_2^b, z) = 1 \mid z = e(g_1, g_2)^{ab}] - \Pr[\mathcal{A}(g_1, g_2, g_1^a, g_2^b, z) = 1 \mid z = e(g, g)^r]|.$$

We say that the co-DBDH assumption holds if $\mathrm{Adv}_{\mathcal{A}}^{\mathsf{co-DBDH}}(\lambda)$ is negligible for any PPT algorithm $\mathcal{A}$.

The q-SDH assumption. Let $(p, e, g, \mathbb{G}, \mathbb{G}_T) \leftarrow \mathcal{G}(1^\lambda)$, $\gamma \leftarrow \mathbb{Z}_p$ and $A_i \leftarrow g^{\gamma^i}$ for $0 \le i \le q$. The q-strong Diffie-Hellman (SDH) problem in $\mathbb{G}$ is stated as follows. Given $(g, (A_i)_{0 \le i \le q})$, output $(c, g^{1/(\gamma+c)})$ where $c \in \mathbb{Z}_p^*$. The advantage of a PPT algorithm $\mathcal{A}$ against the q-SDH problem is defined as

$$\mathrm{Adv}_{\mathcal{A}}^{q-\mathsf{SDH}}(\lambda) = \Pr[\mathcal{A}(g, (A_i)_{0 \leq i \leq q}) = (c, g^{1/(\gamma+c)})].$$

We say that the q-SDH assumption holds if $\mathrm{Adv}_{\mathcal{A}}^{q-\mathsf{SDH}}(\lambda)$ is negligible for any algorithm $\mathcal{A}$.

The DLIN assumption. Let $(u, v, h) \leftarrow \mathbb{G}, (\alpha, \beta, r) \leftarrow \mathbb{Z}_p$ and $g_1 \leftarrow u^{\alpha}, g_2 \leftarrow v^{\beta}$. The decision linear (DLIN) problem in $\mathbb{G}$ is stated as follows. Given $(u, v, h, u^{\alpha}, v^{\beta}, z)$, output 1 if $z = h^{\alpha+\beta}$, otherwise 0 if $z = h^r$. The advantage of an algorithm $\mathcal{A}$ against the DLIN problem is defined as

$$\mathrm{Adv}_{\mathcal{A}}^{\mathsf{DLIN}}(\lambda) = |\Pr[\mathcal{A}(u, v, h, u^{\alpha}, v^{\beta}, z) = 1 \mid z = h^{\alpha+\beta}] - \Pr[\mathcal{A}(u, v, h, u^{\alpha}, v^{\beta}, z) = 1 \mid z = h^r]|.$$

We say that the decision linear assumption holds if $\mathrm{Adv}_{\mathcal{A}}^{\mathsf{DLIN}}(\lambda)$ is negligible for any PPT algorithm $\mathcal{A}$.

The DBDH assumption. Let $(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathcal{G}(1^{\lambda})$ and $a, b, c, r \leftarrow \mathbb{Z}_p$. The decision bilinear Diffie-Hellman (DBDH) problem in $(\mathbb{G}, \mathbb{G}_T\ )$ is stated as follows. Given (g, g^a, g^b, g^c, z), output 1 if $z = e(g, g)^{abc}$, otherwise 0 if $z = e(g, g)^r$. The Advantage of an algorithm $\mathcal{A}$ against the DBDH-problem is defined as

$$\mathrm{Adv}_{\mathcal{A}}^{\mathsf{DBDH}}(\lambda) = |\Pr[\mathcal{A}(g, g^a, g^b, g^c, z) = 1 \mid z = e(g, g)^{abc}] - \Pr[\mathcal{A}(g, g^a, g^b, g^c, z) = 1 \mid z = e(g, g)^r]|.$$

We say that the DBDH assumption holds if $\mathrm{Adv}_{\mathcal{A}}^{\mathsf{DBDH}}(\lambda)$ is negligible for any PPT algorithm $\mathcal{A}$.

(t, Q, ϵ)**-hard protocol.** A protocol is called (t, Q, ϵ)-hard if there exist a probabilistic polynomial time (PPT) adversary $\mathcal{A}$, usually called (Q, t)-adversary that makes Q-queries in running time t to the honest prover, has an advantage at most ϵ,

$$Pr|[\mathcal{A} \ \text{ succeeds}] - 1/2| \leq \epsilon$$

LPN problem [100]. Let for a noise-parameter $\tau \in]0, 1/2[$, *Bernoulli distribution* Ber_{τ} output 1 with probability τ and 0 with probability $(1 - \tau)$. Let $\mathcal{U}_k$ denote the oracle returning independently uniform $k-$bit random strings and $\Pi_{x,\tau}$ be the LPN oracle for a hidden vector $x \in_R \mathbb{Z}_2^k$ s.t.,

$$\langle a \in_R \mathbb{Z}_2^k,\ e \in \mathsf{Ber}_{\tau}^k : \quad a.x \oplus e \rangle$$

The LPN problem is to retrieve x given access to the oracle $\Pi_{x,\tau}$. Any efficient algorithm $\mathcal{A}(q, t, \epsilon)$ can solve the LPN problem with noise parameter τ, if $\mathcal{A}$ runs in time at most t, makes maximum q queries and

$$\mathsf{Pr}[x \in_R \mathbb{Z}_2^k, \mathcal{A}^{\Pi_{x,\tau}}(1^k) = x] \geq \epsilon$$

The ***decisional*-LPN** problem is (k, t, ϵ)-hard if any distinguisher $\mathcal{D}$ running in time t can distinguish uniform binary vector (r) from noisy inner products of vector $(A.x \oplus e)$ such that:

$$\mathsf{Pr}[\mathcal{D}^{\Pi_{x,\tau}}(A, A.x \oplus e) = 1] - \text{Pr}[\mathcal{D}^{\mathcal{U}_k}(A, r) = 1] \leq \epsilon$$

where $l, k \in \mathbb{N}$, $A \in_R \mathbb{Z}_2^{k \times l}$, $x \in_R \mathbb{Z}_2^l$ and $r \in_R \mathbb{Z}_2^k$.

The ***search*-LPN** problem is (k, t, ϵ)-hard if for every $\mathcal{D}$ running in time t

$$\mathsf{Pr}[\mathcal{D}(A, A.x \oplus e) = x] \leq \epsilon$$

Let $y = A.x \oplus e$, then ***computational*-LPN** problem is to compute x and e from a given (A, y) pair. Note that, in the standard definition of the LPN problem, the error vector $e \in \mathbb{Z}_2^k$, from Bernoulli distribution Ber_τ with parameter $0 < \tau < 1/2$, comprises k bits, that yields the *expected* Hamming weight to be $k\tau$. However, in case of ***exact*-LPN** (LPN_x) in [117], the problem is defined exactly like an ordinary LPN, except that the Hamming weight of the error vector is defined exactly $\lfloor k\tau \rceil$. That means, e is chosen independently and identically from $\mathsf{Ber}_{\lfloor k\tau \rceil}$.

SLPN problem. The **Subspace LPN (SLPN)** problem is defined as a biased half-space distribution where the adversary can ask not only with secret x but also with $r'.x \oplus e'$; where $\mathbf{e}', \mathbf{r}'$ can be adaptively chosen with sufficient $rank(r')$. Let $x \in \mathbb{Z}_2^l$ and $l, n \in \mathbb{Z}$ where $n \leq l$. The Decisional SLPN problem is (t, Q, ϵ)-hard such that,

$$Adv_{\mathcal{A}}^{\mathbf{SLPN}}(\tau, l, n) = Pr[LPN_{\tau,l,n}(x, \cdot, \cdot) = 1] - Pr[U_l : LPN_{1/2}(\cdot, \cdot) = 1] \leq \epsilon$$

The **Subset LPN problem** (**SLPN***) is defined as a weaker version to SLPN problem where the adversary cannot ask for all inner products with $r' \cdot s \oplus e'$; for any $rank(r') \geq n$ but only with **subset** of s. Let $(l, n, v) \in \mathbb{Z}$ where $n \leq l$ and $w(v) \geq n$ where v can be adaptively chosen. Hence, $\text{LPN}^*_{\tau,l,n}(s, v)$ samples are of the form $([R]^T{}_{\downarrow}v \cdot s_{\downarrow}v) \oplus e$ and $\text{LPN}_{1/2}(v)$ takes v as input and output a sample of U_l. The SLPN* problem is (t, Q, ϵ)-hard such that,

$$\mathbf{Adv}_{\mathbf{A}}^{\mathbf{SLPN}^*}(\tau, l, n) = Pr[LPN^*_{\tau,l,n}(s, \cdot) = 1] - Pr[U_l : LPN_{1/2}(\cdot) = 1] \leq \epsilon$$

The Field-LPN problem. Field-LPN$_w^F$ problem in [138] states that it is hard to distinguish uniformly random samples in $F \times F$ from those sampled from $\Lambda_w^{F,c}$ for a uniformly chosen c and Hamming weight w. The (decisional) Field-LPN$_w^F$ problem is (t, q, ϵ)-hard if for every distinguisher $\mathcal{D}$ running in time t making q queries such that

$$\text{Pr }[\mathcal{D}^{\Lambda_w^{F,c}} : c \overset{\$}{\leftarrow} F = 1] - \text{Pr}[\mathcal{D}^{U(F \times F)} = 1] \leq \epsilon$$

Pseudo-inverse Matrices. In linear algebra, a **pseudo-inverse** A^+ of a matrix A is a generalization of the inverse matrix. The most widely known and popular pseudo-inverse is the **MoorePenrose** pseudo-inverse, which was independently described by E. H. Moore [96]. An algorithm for generating pseudo-random matrix on non-singular matrix $\mathbb{Z}_2$ is given in [97]. However, the matrix A is the unique matrix that satisfies the following properties:

- $AA^+A = A$
- $A^+AA^+ = A^+$
- $(A^+A)^T = A^+A$
- $(A^+)^+ = A$
- $(A^T)^+ = (A^+)^T$
- $(AA^+)^T = AA^+$ where $T : \mathbb{Z}_2^{n \times l} \to \mathbb{Z}_2^{l \times n}$
- $A^+ = (A^TA)^{-1}A^T$, such that $col(A)$ is linearly independent
- $A^+ = A^T(AA^T)^{-1}$, s.t. $row(A)$ is linearly independent.

Subset Sum problem. *Subset Sum problem* (SSP) is to take decision whether summation of subset of a given set of integers $L := \{a_1, \cdots a_n\}$ s.t., $a_i \in \mathbb{Z}_p, 1 \leq i \leq n$ is $t \in \mathbb{Z}_p$. Let $t = x_1a_1 + \cdots + x_na_n$ for a binary vector $X = \langle x_1, \cdots, x_n\rangle$ s.t., $x_i \in \{0, 1\}$. Then given L and t, it is hard (NP-complete) to find out X.

Polynomial Reconstruction Problem. Security of the authentication scheme described in [160] is based on the hardness of the well-known *Noisy Polynomial Interpolation Problem*(NPI) [161]. Authors consider *query and recovery* attack where the adversary queries the tag in order to recover the polynomial assigned to the tag. Because of the difficulty of *query and recovery* attack can be realized by the difficulty of the NPI problem. We refer to [160] for necessary definitions. Note that we slightly modify the existing protocol to reduce communication and computational overhead of the protocol. Moreover, our modified version is more secure, but requires more parameters to share between the reader and tag.

In order to respond to the challenge r, the tag evaluates a univariate polynomial $r' = \mathfrak{f}_{\mathcal{T}_i}(r + y_0)$. Since y_0 is a shared secret between the tag and reader, $y_0 + r$ can be considered as random to the adversary. In addition, using *secure hash* causes h(`reader_data`) to be considered as random even if the adversary knows `reader_data`. This r' is forwarded along with extra $b - 1$ random elements. In every consecutive m queries ($m < Q$) by the adversary, the tag employs all of its polynomial $\mathfrak{f}_i, 1 \leq i \leq m$ one after another, but in random manner.

2.2.3 Ciphers and Encoding

Hill cipher. It was the first proposed in [118] for the matrix based cryptosystem, where the ciphertext is obtained from the plaintext by means of a linear transformation. The a plaintext vector $X \in \mathbb{Z}^k$ is encrypted to get ciphertext Y as:

$$Y = XK \pmod m \in \mathbb{Z}_m^k$$

where the key $K \in \mathbb{Z}_m^{k\times k}$ is an invertible matrix, $\mathbb{Z}_m$ is a ring of integers modulo m. Decryption is done as follows:

$$X = YK^{-1}(\text{mod } m).$$

0/1-ENCoding and VLR. In [49], authors present an encoding scheme, namely 0/1-encoding, that helps converting the *greater than* predicate to the *set intersection* predicate. This property allows the GM to embed the *key expiration* date into the signer's certificate and the signer to sign a message with a *signature expiration* date. Since the signer should not expose its key expiration date d (for privacy purpose), it sets an expiration date t (such that $d > t$) for each signature. Later verifier can check if the current date $\bar{t}$ is no later than the signature expiration date t. It ensures ($d > t \geq \bar{t}$) that the signature is generated by a non-expired signer. Clearly, verifier will pass the signature if there exists a common element between the signer's (key) expiration date and signature expiration date.

It converts a date format (in binary) to a value in $\mathbb{Z}_p$ in the following way.

- Let $t \leftarrow t_{[l]} \dots t_{[1]}$ be an l-bit date encoded in binary string.
- 0-Enc: $T_t^0 = \{t_{[l]} \dots t_{[i+1]}\mathbf{1} \| t_{[i]} = 0, 1 \leq i \leq l\}$,
  1-Enc: $T_t^1 = \{t_{[l]} \dots t_{[i]} \parallel t_{[i]} = 1, 1 \leq i \leq l\}$.
- If $x > y$, there is a common element in T_x^1 and T_y^0.
- To ensure that the sets start with 1, redefine the sets as the decimal number set as follows
  $\overline{T_t^0} = \{1 \cdot 10^{l-i+1} + t_{[l]} \cdot 10^{l-i} + \dots + t_{[i+1]} \cdot 10^1 + 1 \| t_{[i]} = 0, 1 \leq i \leq l\}$,
  $\overline{T_t^1} = \{1 \cdot 10^{l-i+1} + t_{[l]} \cdot 10^{l-i} + \dots + t_{[i+1]} \cdot 10^1 + t_{[i]} \| t_{[i]} = 1, 1 \leq i \leq l\}$,
- Padding with dummy elements so that the number of elements in the sets are same. For 0-Enc:

$$t_{[i]} = \begin{cases} z & \text{if } z \in \overline{T_t^0} \text{ and } \lfloor \mathsf{log}_{10} z \rfloor - 1 = i \\ 2 \cdot 10^i & \text{otherwise,} \end{cases}$$

For 1-Enc:

$$t_{[i]} = \begin{cases} z & \text{if } z \in \overline{T_t^1} \text{ and } \lfloor \mathsf{log}_{10} z \rfloor - 1 = i \\ 3 \cdot 10^i & \text{otherwise.} \end{cases}$$

- Assume two dates $x =$ "10100010111" ('1303' for March,2013) and $y =$ "1010001010" ('1301' for January,2013) in a format $'\mathsf{YYMM}'$. Now

  $$T_x^1 = \{1, 101, 1010001, 101000101, 1010001011, 10100010111\},$$
  $$T_y^0 = \{11, 1011, 10101, 101001, 10100011, 1010001011\}.$$

  and

  $$\overline{T_x^1} = \{11, 1101, 11010001, 1101000101, 11010001011, 110100010111\},$$
  $$\overline{T_y^0} = \{111, 11011, 110101, 1101001, 110100011, 11010001011\}$$

- After padding
  0-$\mathsf{Enc}(y) \rightarrow \{20, 111, 2000, 11011, 110101, 1101001, 20000000, 110100011, 2000000000, \mathbf{11010001011}, 200000000000\}$,
  1-$\mathsf{Enc}(x) \rightarrow \{11, 300, 1101, 30000, 300000, 3000000, 11010001, 300000000, 1101000101, \mathbf{11010001011}, 110100010111\}$.

- Since $x > y$, 1-$\mathsf{Enc}(x)$ and 0-$\mathsf{Enc}(y)$ have a common element **11010001011**. For detailed proof, please find the theorem in [49].

2.2.4 Protocol Building Blocks

Group Signature. A Group signature scheme is a method for allowing a member of a group to anonymously sign a message on behalf of the group. A group signature may have several properties as follows

- Integrity: No outsider can spoof
- Completeness and Soundness: Valid signatures by members should be verified correctly, and invalid signatures should fail in verification.
- Anonymity: Given a message and its signature, the identity of the signer cannot be determined without secret key.
- Linkability: Given two messages and their signatures, an authority can tell if the signatures were from the same signer or not
- Traceability: Given any valid signature, an authority should be able to trace the member issued the signature by breaking member's anonymity.
- Revocability: No revoked member can make a valid group signature. It can be achieved through Verifier local Revocation (VLR) and Re-keying the group.
- Batch Verification: Batching signatures from different group members in order to accelerate verification.

However, a group signature scheme $\mathsf{GS} = (\mathsf{GKg}, \mathsf{GSig}, \mathsf{GVf}, \mathsf{Open})$ consists of basic four polynomial-time algorithms:

- $\mathsf{GKg}(1^k, 1^n)$: Group key generation algorithm takes input $(1^k, 1^n)$ where $k \in \mathbb{N}$ is the security parameter and $n \in \mathbb{N}$ is the number of members of the group and returns a tuple $(gpk, gmsk, gsk)$, where gpk is the group public key, $gmsk$ is the group managers secret key, and $gsk[i]$ is a secret signing key for each member $i \in [n]$.
- $\mathsf{GSig}(gsk[i], m)$: Group signing algorithm takes as input a secret signing key $gsk[i]$ and a message m to return a signature of m under $gsk[i](i \in [n])$.
- $\mathsf{GVf}(gpk, m, \sigma)$: Group signature verication algorithm is a deterministic algorithm that takes as input the group public key gpk, a message m, and a signature σ for m and returns either 1 or 0.
- $\mathsf{Open}(gmsk, m, \sigma)$: This algorithm takes as input the group manager secret key $gmsk$, a message m, and a signature σ of m and returns either an identity i or $\perp$ if fails.

Batch verification. Batch verification was first introduced for RSA. Later a number of batch verification schemes have been proposed [25][26][27]. M. Bellare *et al.* gave the first idea about fast batch verification on digital signatures [29].

- **Small exponent test.** Choose $\delta_1 \dots \delta_n \in \{0,1\}^l$. Then compute $a = \sum_{j=1}^{n} a_j \delta_j \ mod\ q$ and $y = \prod_{j=1}^{n} y_j^{\delta_j}$. where $a_j, y_j \in \mathbb{Z}_q$. After this, check whether it holds: $g^a = y$. if yes then accept, else reject.

Later A. L. Ferrara *et al.* proposed *three* techniques to develop batch verification for bilinear equation [30]. Here are their techniques in brief:

- **Technique 1.** Sigma-protocols, usually called $\sum$-protocols, have three move structures: commitment, challenge and response. This is one of the implementable protocols of *Proof of knowledge.* These three steps degrade the verification mechanism more. Ferrara *et al.* suggests to reduce it as (commitment,response) policy to achieve much more verifiable equations. For pairing, they propose two sub-steps:
  - *Check membership.* Only elements that an adversary could attack need to be checked. Public parameters need not be checked, or it can be checked once.
  - *Small Exponent Test.* Perform the test to combine all the equations into one.
- **Technique 2.** Move the exponent into the pairing, for example, Replace $e(g_i, h_i)^{\delta^i}$ with $e(g_i^{\delta^i}, h_i)$. It speeds up the exponentiation process.
- **Technique 3.** If two pairings with common elements appear. It will reduce n pairing to 1. For example, replace $\prod_{i=1}^{n} e(g_i^{\delta^i}, h)$ with $e(\prod_{j=1}^{n} g_i^{\delta^i}, h)$

Homomprphic Aggregated Signature. A *Homomprphic Aggregated Signature* is defined by the following algorithms:

- $\mathsf{Kgen}(1^\lambda, m)$ On input security parameter λ and $m \geq 1$, it outputs (pk, sk) where pk is the public verification key and sk is the secret signing key. Here m is the dimension of the vector space.

- $\mathsf{Sign}(sk, \mathbf{uid}_c, \mathbf{uid}_n, \mathrm{T}, \widehat{T})$ On input secret key sk, current and new owner IDs $\mathbf{uid}_c, \mathbf{uid}_n$, a set of tag ID and index $\{\mathrm{T}, \widehat{T}\}$, it outputs a signature Σ.

- $\mathsf{CombSign}(pk, \mathbf{uid}_c, \mathbf{uid}_n, \widehat{T}_i, \Sigma_i)$ Given the public key pk, Owner IDs, a set of tag index $\widehat{T}^{(i)}$ and their signature Σ_i, it outputs a new aggregated signature Σ.

- $\mathsf{VerSign}(pk, \mathbf{uid}_c, \mathbf{uid}_n, \mathrm{T}^{(i)}, \Sigma)$ Based on the public key pk, a set of tag ID and index $\{T_i, \widehat{T}_i\}$ and a signature Σ, it can verify the signature and outputs 0 (reject) or 1 (accept).

Stateful Signature. In a stateful signature scheme, the signer updates some state after every signature is produced. A stateful signature scheme consists of three efficient algorithms:

- $\mathsf{KGen}(1^k)$: On input 1^k, compute (pk, sk)$\leftarrow$ KGen(1^k). Let $[X, X^+] := \mathsf{PseudoInvGen}(S)$, where S is a random parameter and X^+ be the initial state of a stateful signature. Then $sk := X^+$ and $pk := X^+X$.

- $\mathsf{Sign}(m, sk)$: To sign a message m using the current state, it outputs a signature σ_m and updates the current state by $\sigma_m := mX^+$.

- $\mathsf{Vrfy}(\sigma_m, pk, m)$: Verify algorithm outputs 1, if and only if $\mathsf{Vrfy}_{pk}(m, \sigma_m) = 1$ such that $\sigma_m \overset{?}{=} \sigma_m \cdot\ pk$.

Commitment scheme. It is a two-phase protocol between a *sender* and a *receiver* where the sender holds a message m. In the first phase, the sender picks a random key ck and then encodes m using ck and sends the encoding message c (a commitment to m) to the receiver. In the second phase, the sender sends the key ck to the receiver and it can open the commitment and find out the content of the message m. More formally, A triple of algorithms ($\mathsf{KGen}, \mathsf{Com}, \mathsf{Ver}$) is called a commitment scheme if it satisfies the following:

- On input 1^l, the key generation algorithm KGen output a commitment key ck.

- The commitment algorithm Com takes as input a message m from a message space $\mathcal{M}$ and a commitment key ck, and output a commitment-opening pair (c, d).

- The verification algorithm Ver takes a key ck, a message m, a commitment c and an opening d and output 1 or 0.

Searchable Encryption. A *Searchable Encryption* can be defined by the following algorithm:

- $\mathsf{Kgen}(m, n)$ On input the size of the matrix, it outputs a pair of keys (P_c, S_c) where P_c is public key and S_c is secret key.
- $\mathsf{Enc}(P_c, Q)$ Given a challenge matrix $Q^{n \times n}$ and public key P_c, it generates ciphertext $E := \mathsf{Enc}(P_c, Q)$.
- $\mathsf{TDoor}(S_c, Q)$ This algorithm takes secret key S_c, challenge matrix Q and outputs a trapdoor $T := \{C, D\}$ correspond to $\{S_c, Q\}$.
- $\mathsf{Test}(P_c, E, T)$ On input P_c, trapdoor T and ciphertext E, it proves whether T and E are generated from the same Q and outputs 0 (reject) or 1 (accept).

Labelled Program. The notion of labeled data or program was first introduced by Gennaro *et al.* [159]. Let an entity (e.g., checkpoint) want to authenticate some data $\tau := \{\tau_0, \tau_1, \cdots \tau_r\}$ (e.g., tag/reader's data) with respect to their corresponding labels $\mathcal{I} := \{\iota_0, \iota_1, \cdots \iota_r\}$ (e.g., tag/reader's unique identifier) where $\iota_i \in \{0, 1\}^*$. A labeled program can be defined by $\mathcal{P} := (f, \mathcal{I})$ where $f : \{0, 1\}^r \to \{0, 1\}$ is a circuit on data τ. Output of a labeled program can be computed over data τ provided by different entities (e.g., Readers) at different times.

Arithmetic Circuit. An arithmetic circuit f over the variables or data $\tau = \tau_0, \tau_1, \cdots, \tau_r$ is a labelled directed acyclic graph G with its leaves labelled as $\mathcal{I} := \{\iota_0, \iota_1, \cdots \iota_r\}$ and internal nodes labelled as gate $\mathcal{O} := \{+, \times\}$ operations. The circuit has a designated output ρ.

In this work, we consider an arithmetic circuit f over a field $\mathbb{Z}_p$ such that $f : \mathbb{Z}_p^r \to \mathbb{Z}_p$ for a prime p. The circuit f has bounded fan-in, that is, each of its internal nodes has at most two children. The size of a circuit, $\mathsf{size}(f)$ is the number of gates/vertices in underlying graph. The depth of the circuit, $\mathsf{depth}(f)$ is the length of the longest directed path in the circuit. Note that an arithmetic circuit can compute a polynomial in the natural way and every polynomial defines a unique function. An input gate of an arithmatic circuit can compute a polynomial it is tagged by the labels. A sum gate '+' computes the sum of *two* polynomials obtained from the incoming wire in the graph. Similarly, a product gate '×' computes the product of *two* polynomials.

However, the *degree of a circuit* is delineated by the maximal degree of the gates in the circuit while the *degree of a gate* is defined by the total degree of the polynomial it computes. Note that all the polynomials belong to the class VP, the algebraic analog of class P. That is, all polynomials of polynomially bounded degree can be realized by an arithmetic circuit family with polynomially bounded size [162].

Pseudo Random Function. Pseudo-random function (PRF) is a family of functions F such that $F : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$ such that each function $F_k \in F$ can be identified by a unique index $k \in \{0,1\}^\lambda$. Given the key $k \in \mathcal{K}$, the function $F_k(\cdot)$ can be efficiently evaluated at all point $x \in \mathcal{X}$. F_k can be computed by a deterministic polynomial time algorithm: on input $(k, x) \in \mathcal{K} \times \mathcal{X}$ the algorithm outputs $F_k(x) \in \mathcal{Y}$.

In this theses, we use circuit-PRF $F : \mathcal{K} \times \{0,1\}^n \rightarrow \mathcal{Y}$. For every polynomial circuit $\mathcal{C}$ there is a key k_c that enables the evaluation of $F_{k_c}(x)$ at all points $x \in \{0,1\}^n$ s.t., $C(x) = 1$. Consider a security experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{prf}-b}$ where adversary $\mathcal{A}$ interacts with a challenger. Let challenger choose k and initialize an oracle that on input $x \in \mathcal{X}$ outputs $F_k(x) \in \mathcal{Y}$ if $b = 1$ and $r \in_R \{0,1\}^m$ otherwise. $\mathcal{A}$ returns a bit b' and wins the security experiment if $b = b'$. We define adversary's advantage $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{prf}}(\lambda)$= $\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{prf}-1} = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{prf}-0} = 1]| \leq \epsilon$.

Homomorphic Message Authentication. In a homomorphic message authentication scheme, an entity can authenticate data τ with its secret key sk. Later evaluators can homomorphically execute an arbitrary program $\mathcal{P}$ over τ and subsequently generate an authentication tag σ without knowing sk. Note that σ certifies $\mathcal{P}(\tau)$. Finally a verifier that knows sk can assert whether σ is indeed the output of the $\mathcal{P}(\tau)$ without knowing τ. A Homomorphic Message Authentication scheme consists of the following *four* algorithms:

- $\mathsf{KGen}(1^\lambda)$: On input of the security parameter λ, it generates a key pair $(\mathsf{sk}, \mathsf{ek})$ where **sk** is the secret key and ek is the public evaluation key.

- $\mathsf{Authentication}(\mathsf{sk}, \iota, \tau)$: Given the secret key sk, a label ι and a message data τ, it outputs a succinct tag σ.

- $\mathsf{Evaluate}(\mathsf{ek}, f, \sigma)$: On input of the evaluation key ek, a circuit $f : \mathbb{Z}_p^r \rightarrow \mathbb{Z}_p$ and a set of authenticating tags $(\sigma_0, \cdots, \sigma_r)$, this algorithm outputs a new tag σ.

- $\mathsf{Verify}(\mathsf{sk}, \tau, \mathcal{P}, \sigma)$: On input of the secret key sk, a program $\mathcal{P} := (f, \mathcal{I})$ where $\mathcal{I} := \{\iota_0, \iota_1, \cdots \iota_r\}$, a message data τ (computed on f), and an authentication tag σ, the verification algorithm outputs 0 (reject) or 1 (accept).

Chapter 3

Previous Work

3.1 VANET security

Security mechanisms proposed for VANET in the literatures can be divided into two techniques: prevention techniques and detection technique. **Detection techniques** include:

- *Signature-based* detection where attacks are detected by comparing network traffic to known signatures of attacks (stored in the attack signature database).
- *Anomaly-based* detection is a statistical approach that can detect any deviation (anomalies) from the normal communication system behaviour.
- *Context Verification* collects information from any reliable sources (e.g. telemetric monitoring) in order to create an independent view of its current status and then evaluate the situation based on predefined rule-sets depending on the application.

However, considering the critical and safety nature of VANET application, *prevention techniques* are more popular. Most of the existing security solutions [8, 9, 20, 24, 46, 56, 19, 21, 44] aim to prevent security breaches rather than detecting them. **Prevention technique** includes the followings:

- *Digital signature* that exploit cryptography, either with certification (e.g., [6]) where cryptographic digital signatures are applied to messages or hashes over messages, or without certificate (e.g., [181]) where signatures are combined with digital certificates provided by a trusted Certificate Authority (CA).
- *Proprietary system* exploits proprietary (non-public) protocols or hardwares in order to control unauthorized access to the network.
- *Temper proof hardware* ensures secure input to the communication system and allow only authorized entity to access it.

In this section, we provide some preliminaries regarding wireless communication, vehicle addressing mechanisms, vehicle characteristics from Sevecom Project in [182]. In addition, we gather major VANET security and privacy solutions and architectures proposed in the literature. Meanwhile, we discuss some projects and standardization studies that have been conducted on the security of VANET.

VANET wireless communication: Wireless Communication involves vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and infrastructure-to-vehicle (I2V) communication. It does not include in-car wireless (e.g. Bluetooth used with mobile or PDA) transaction.

- Sender/Receiver
  - Vehicle-to-Vehicle (V2V): Car originates communication to other car.
  - Vehicle-to-Infrastructure (V2I): Car originates communication with infrastructure.
  - Infrastructure-to-Vehicle (I2I): Infrastructure originates communication with other Infrastructure.
- Communication characteristics
  - Single-hop: We assume a single-hop range of at least 150m in normal road conditions. In case of curve or turn applications, the range may be shorter
  - Multi-hop: Multi-hopping is assumed to be realized by a position-based routing protocol.
- Message characteristics
  - One-way: Messages are sent without response.
  - Two-way: Messages are sent with response.
  - Periodic: The periodic sending of messaging may be triggered by some external events, like setting the indicators or activating the blue light in an emergency vehicle.
  - Relevancy: Messages are transported passively, using a content-based and situation-based relevancy calculation.
- Addressing mechanism
  - Unicast: Receiver is a unique network entity (e.g. a vehicle, RSU etc.).
  - Broadcast: Receivers are all network entities that receive a packet.
    - Single-hop: Every receiver in wireless transmission range.
    - Multi-hop: Time-To-Live (TTL)-limited flooding.
  - Geocast: All network entities receiving a packet must check their own position to decide whether they are intended to process the packet.
    - Single-hop: Only the entities in the defined region are receivers. No relaying.
    - Multi-hop: If the receivers are already in the target region, flood the packet within the region. If not (outside the target region), forward the packet to the target region based on routing protocol, then flood.

Vehicle characteristics: Security application strongly involves Vehicle Characteristics, e.g. in-car sensors or software systems. This is the case, for instance, if vehicle software is updated or integral parts of the vehicle (like brakes or engine) are influenced. This has security implications because these parts are critical for safe operation of the vehicle.

- Electronic license plate: The electronic license plate allows reading of vehicle license plates via wireless interface. It must be available to authorized communication partners.
- Electronic driver's license: A driver has to issue his/her license to the car (for instance, a car would not start without driver's license).

Security Requirements: VANET presents some inherited challenges from that of MANETs and sensor networks such as mobility, security versus privacy, availability, low error tolerance, key distribution problem etc. We gather all the security requirements for vehicular network listed in the literature:

- Authentication: This is the obvious requirement for any scheme where the vehicles response only to the legitimate entities and messages. In VANET, authentication is required for the OBUs, RSUs during vehicle-to-vehicle and vehicle-to-infrastructure communication.
- Data consistency: VANET requires not only the sender to be legal but also the legitimacy of the data the sender sent. It is also known as *plausibility*.
- Data integrity: Protocol message could be altered by malicious adversary during transaction. Therefore, message integrity should be ensured at the destination entities.
- Availability: Many kinds of Denial of Service (DoS) attacks such as desynchronization, jamming attacks may deteriorate network performance. Therefore, availability of the message should be provided by any means.
- Non-repudiation: Vehicles causing accidents should be reliably identified to prove liability. That is, a sender (signer) should not be able to decline its responsibility after the transmission of a message (for instance after having an accident).
- Privacy: This is the most important research area in VANET where the vehicles and their locations can be monitored or tracked intentionally. Therefore, privacy of the drivers or vehicles against unauthorized observers should be preserved.
- Traceability and Revocation: This is opposite to the privacy requirement. Sometimes it becomes essential for the authority to trace vehicles' identity and revoke membership/license of the driver for the sake of unspeculated or abnormal behavior of the vehicles.

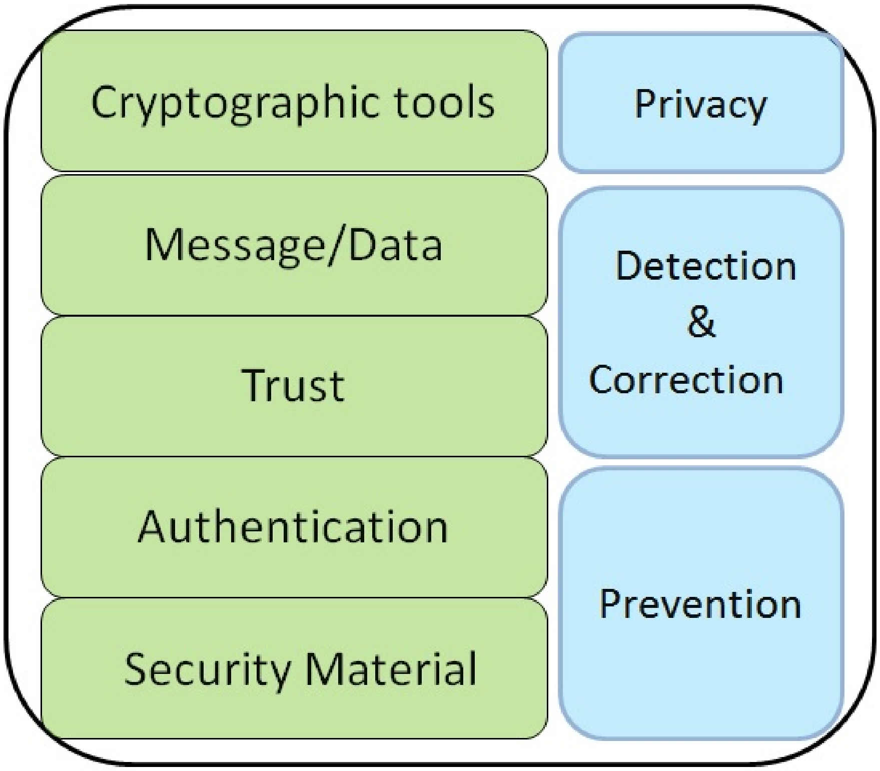

Figure 3.1: VANET general architecture.

- Real-time constraints: Sometimes security could be a barrier to efficiency. Since nodes in the VANET are mobile and fast locomotive, they require strict time constraints to process data and feedback. This imposes computation and communication efficient schemes. For example, Sevecom project in [182] defines time constraints as follows:
  - Highly time-critical (∼0.5 seconds)
  - Time critical (∼ 1 second)
  - Time is relevant (∼ 5 seconds)
  - Time is not a critical issue (> 10 seconds)

General security architecture: In [168], authors propose a general security architecture of VANET (See Fig. 3.1). Five-level security architecture consists of: security stuffs, authentication, trust, message/data and cryptographic tools for security aspects. At the bottom level (Security Material) of the architecture, security hardware, such as On-Board Unit (OBU), Global Positioning System (GPS) receivers, radars, Event Data Recorder (EDR), antennas remains. Hardware Security are built around a Trusted Platform Module (TPM) specifications that include a software infrastructure to protect and store data. *Authentication level* is responsible for all kind of authentications such as *driver's authentication* to avoid unauthorized users' access to the system, *message authentication* to ensure the messages to be received from the legitimate entities and not to be modified in transit, property authentication to guarantee legitimate transmission equipment, location authentication to ensures the sender's position. *Trust level* is where a trust/reputation system is installed. It uses local information such as speed, acceleration etc. as well as information from other nodes to create a suggestion concerning a node with which it intends to communicate. At *message/data level*, the message itself should be secure by some primitives such as digital signature with the existence of the vehicular PKI. If a malicious node remained undetected at the higher levels (authentication and trust), the message being transmitted would be verified here to ensure the security and integrity of the network transaction. Cryptographic level is responsible for drivers' privacy where privacy solutions such as public key infrastructure (PKI), group signatures, or anonymous identity protocol are applied to ensure privacy. However, authors divide this architecture into three sections: the prevention section (security material and authentication level), the detection and correction section (Trust level and the message/data level), and the privacy section (Cryptographic level).

Security Standardization: We discuss two standard families (ISO and IEEE) and one project (SeveCom) concerning VANET security and privacy.

- International Standard Organization (ISO) family such as **ISO/TR 12859:2009** and **ISO 21217:2014** [163, 164] provides general guidelines to the developers of intelligent transport systems (ITS) standard and systems on data security and privacy aspects. Specially ISO 21217:2014 specifies the minimum set of requirements for a physical instantiation of an ITS station. Communication Access for Land Mobiles (CALM) standard (in [163]) defines the common architectural framework of CALM-compliant ITS stations that could be located at vehicles, RSUs. They also define how lawful interception should be performed. According to CALM, there should be a security module in the vehicle that includes a firewall and intrusion detection system called Hardware Security Module (HSM), an authentication, authorization and profile management system, and an identity, certificates and cryptographic algorithm system.

- IEEE family defines three standards such as **IEEE 1609.0-2013**, **IEEE 1609.2-2013** and **IEEE 1609.11-2010** [165, 166, 167]. IEEE 1609.0 defines the general architecture for VANET called Wireless Access in Vehicular Environments (WAVE) architecture and services that can be used in conjunction with the family of IEEE 1609 standards. IEEE 1609.2 determines the secure message formats and processing for using inside WAVE-enabled devices. This standard also defines the method to secure WAVE management messages and application messages and some functions necessary to support the core security functions s.t., confidentiality, authentication, authorization and integrity. For cryptographic implementations, this standard considers public key certificates (PKC). For the privacy purpose, it refers to the use of pseudonyms as identifier within the certificates by which WAVE devices can conceal its real-world identity. For revocation management, they propose to publish Certificate Revocation Lists (CRLs) by the certificate manager. IEEE 1609.11 standardizes the Over-the-Air electronic payment data exchange protocol for ITS that includes the application service layer and profile for payment and identity authentication. It also determines a basic level of technical interoperability between an electronic payment equipment (such as On Board unit (OBU)) and roadside equipment (RSE) using Dedicated Short Range Communication (DSRC) based application.

- SeVeCom (Secure Vehicular Communication) is an EU-funded project that focuses on providing a full definition and implementation of baseline security requirements for vehicular communications [182]. SeVeCom project addresses security of the future vehicle communication networks, including both inter-vehicular and vehicle-infrastructure communication.

VANET Security Models: We briefly describe some schemes related to VANET security.

- In [175], Lee *et al.* presented a secure incentive framework for commercial VANET called Signature-Seeking Drive (SSD) by leveraging a Public Key Infrastructure (PKI) to provide secure incentives for cooperative nodes without relying on tamper-proof hardware.
- In [176], Sun *et al.* proposed an ID-based cryptosystem to provide message integrity, confidentiality and authentication for VANET. It avoids using certificates for authentication. In addition, it can detect spoofing attacks and correct malicious data for data consistency.
- In [177], Zhang *et al.* proposed RAISE scheme, an RSU-aided message authentication scheme that to improve authentication scheme in a large scale VANET specially for metropolitan-area network.
- In [178], Isaac *et al.* proposed an on-road payment model for VANET by using an authenticated encryption scheme where payment can be accomplished by the smart card (e.g., credit card).
- In [179], Wagan *et al.* presented an efficient, secure and hardware-friendly infrastructure to exchange safety messages using standard asymmetric PKI as well as symmetric cryptography. Authors emphasize on developing trusted relationship among the neighboring nodes to form clusters of trusted vehicles.
- In [180], Biswas *et al.* presented an ID-based anonymous authentication scheme and a cross-layer verification approach for a WAVE-enabled vehicular safety messages. They propose an Elliptic Curve Digital Signature Algorithm (ECDSA) along with an identity-based signature that verifies the received messages based on their MAC traffic class and intensity. In addition, they suggest a cross-layer prioritized message verification system for the periodic road safety messages.

VANET Privacy Models: We briefly describe some privacy-enabled VANET security schemes .

- In [13], Dotzer *et al.* discusses the privacy issues in VANET from a manufacturers point of view and provide solutions to overcome the issues.

- In [169], Lu *et al.* presented a conditional privacy preserving protocol for secure vehicular communications. This scheme can efficiently handle the linear revocation list and allows the authorities to trace vehicular identifiers with some pre-defined condition.

- In [170], Li *et al.* presented an efficient key management scheme to support privacy. They use non-interactive identity-based public key cryptography, blind signature and one-way hash chain as building blocks in order to achieve mutual authentication, reduced computational costs, dynamic session key among the nodes.

- In [171], Burmester *et al.* described a hybrid scheme that utilizes both symmetric and public key algorithm for authentication and encryption. It established a balance between the privacy and accountability in vehicular nodes. They considered both pairwise and group communication among the VANET entities. They allow the vehicles to change pseudonyms frequently in order to achieve strong privacy.

- In [172], Yan *et al.* considered vehicular geographical positions as a valuable information in VANET and hence proposed a novel approach to enhance vehicle position security. In addition, they introduced on-board radar to detect neighboring vehicles with their announced coordinates.

- In [173], Wei *et al.* proposed a received signal strength indicator-based User Centric Anonymization (UCA) model for location privacy of the vehicles without any centralized trusted party. They introduced an adversary namely called Global Passive Adversary (GPA) that can locate and track a vehicle within its region by intercepting the broadcast message. Their model improved the location privacy by considering four parameters such as pseudo position, velocity, direction and Random silent period simultaneously.

- In [174], Prado *et al.* proposed a privacy preserving geo-casting protocol, a variant of direction-based geocasting protocol. They suggest to encrypt the vehicular messages first and then geocast the messages across VANET.

VANET security and Group Signature: Security and privacy in VANETs are discussed in the literature suggest the use of a public key infrastructure (PKI) and digital signatures to secure VANETs [8, 9, 19, 21, 44]. Besides that, several proposals suggest pseudonym mechanisms (e.g., [20, 24, 46, 56]) in order to guarantee vehicle privacy. The pseudonymity approach focuses on how often a node should change a pseudonym and with whom it should communicate. Moreover, some schemes adopt group signature based solution (e.g., [1, 3, 6, 7, 22, 23]), traceable ring signature based solution (e.g., [4, 5]) for VANET security and privacy. Group signatures can be directly used to anonymously authenticate vehicular communications without additionally generating a pseudonym.

Unlike traditional digital signature schemes, GS allows a member to create an *anonymous* (and *unlinkable*) signature that conceals the identity of the vehicle and hence preserves privacy [23]. Following the foundation of GS [45], a number of different security requirements have been proposed as primitives. Consequently, BBS-model in [59], proposes the shortest GS scheme with three security notions anonymity, traceability and exculpability.

Linkability feature is discussed in several GS schemes such as short GS based scheme in [64], direct anonymous attestation scheme in [61], ring signature schemes in [63, 67]. All of them do not support either traceability or revocability. In [70], authors propose a special type of GS with short-term linkability for VANET where the signer will keep remain three group signature elements unchanged (without randomizing) for a short term. Although it gears up verification process, but signatures generated this way are linkable by all the group members. Whereas, in general, linkable GS has *linking key* to link signatures and members who have linking key can only link the signatures.

Traceability is a fundamental properties of BBS GS [59]. In [53], authors introduce a new direction to traceability. In order to subside the power of the *opener*, they bring in a new authority called *admitter* which generates *tokens* corresponding to messages without which tracing manager (*opener*) cannot proceed. Once the *token* is generated, no interaction between the *opener* and the *admitter* is required for further operation. Although message-dependent traceability is more application-friendly, sometimes authorities like TSD in VANET requires to revoke a member's anonymity *directly* without depending on any other authority (e.g., emerging national security threats).

Revocability properties for a GS was first explored in [58] and later followed by [57, 68, 69]. All the revocable GS schemes that have been proposed so far are reluctant to backward unlinkability, verification cost (VLR) etc. The GS scheme in [22] combines hybrid revocation mechanism with [59] that works with the list of revoked members (RL) and a threshold value. If the size of RL is less than the threshold value, the scheme follows VLR scheme for revocation. Otherwise the scheme uses *rekeying* process to update the public/private group keys of all non-revoked members. In [51], authors introduce a special VLR supported GS scheme with time-bound keys. Although they minimize the revocation check to a greater extent, but the verifier still needs to perform revocation check against all the members in RL. Note that VLR scheme with RL is not practical for a large scale VANET where a verifier needs to check whether a signer belong to the RL each time it receives a signature.

In [25], authors propose a GS with batch verification with drawbacks like impersonation attack, tractability etc. [48]. In order to reduce the the burden of signature verification in a large scale VANET environments and make group signatures as a practical solution to the intelligent traffic system, a number of batch verification mechanisms have been proposed in the literature, either on group signature (e.g., [1, 3, 7, 8]) or on ring signature (e.g., [4, 5]). A short GS based on [59] and an Identity Based Group Signature (IBGS) based on [1] with fast batch verification are proposed in [8] for a large scale VANET. Nonetheless, the verification speed of [8] can be accelerated more by simplifying verification equation. Besides that, sometimes signature verification system get even worse for using batch verification depending on the volume of communication message in a fixed interval.

3.2 RFID Authentication Protocols

RFID tags are found with either read-write chips or read-only chips. With read-write chips, the reader can modify the internal data to the tag except the serial number (e.g., identifier), while read-only microchips' data are stored into them during the manufacturing process. Moreover, based on the source of power supply, RFID tags are classified into three categories:

- *Active tags* that consist of a transmitter and internal power source like a battery.
- *Semi-active tags* use their own power source to run internal circuitry, but utilize the reader's power for communication.
- *Passive tags* have no internal power repository. Alternatively, they utilize the readers as power source.

Furthermore, the operating frequency of RFID tags can be categorized into three ranges, that are, Low-frequency (124 kHz-135 kHz), High-frequency (13.56 MHz), and Ultra-High-frequency (2.45 GHz). In this theses, we concentrate on the passive RFID tags with read-write microchip that can operate in any allowed frequency ranges. In order to protect the privacy of consumers and ensure the security of resource-constrained low-cost RFID tags, we propose an enhancement to patch an existing authentication protocols.

In general, RFID authentication protocols experience three major type of attacks: *Passive attacks* allow adversaries to acquire data from the tag only, not from the communication channel between the reader and tag. *Active attack* allows the adversary not only to inquire data from the tag, but also to control the reader-tag channel. *Man-in-the-middle* (MIM)is the most powerful attack where the adversary can not only access the communication channel (like Active attack), but also can modify the protocol transactions between the reader and tag.

Limited processing and storage capability of traditional RFID tags limit the effective use of cryptographic techniques such as RSA, ECC [79, 85]. For example, EPC Class-1 Gen-2 permits only 2500 gates for security operations. This will resist standard cryptographic techniques s.t., RSA to implement [79]. Similarly, stronger security primitives like SHA-1, MD5 require 16000-20000 gates for implementation. In addition, popular Elliptic curve cryptography (ECC) as implemented in ERAP [84] require approximately 15000 gates [85].

HB-family protocols based on LPN assumption require a few thousand gates for implementation making them an attractive option for securing low cost EPC tags [113]. In 2001, Hopper and Blum (HB) proposed a lightweight authentication protocol based on the Learning Parity with Noise (LPN) problem. This protocol along with its subsequent protocols are commonly known as the HB-family protocol. HB-family consists of a series of protocols such as HB+, HB++, HB-MP, HB-MP+, Trusted-HB, HB#, GHB#, F-HB, Tree-HB+, Tree-LSHB+ etc. We discuss some of these protocols with their limitations. Although mutual authentication protocol adds an additional protection for an RFID system in the protocol construction to safeguard the query is, in fact, coming from a legitimate entity, and therefore, ensures that the tag information is available to only valid reader and server. Most authentication protocols (specially HB-family protocol) proposed so far either presume reader and server as an identical entity, or assume the communication channel between a server and a reader is secure [89, 90, 91, 92, 95, 108, 93, 105, 111, 109, 106].

LPN assumption used in HB-like protocols, inquires to distinguish *noisy* linear equation from uniformly random. Since its first introduction in 2001, numerous applications i.e., lightweight crypto system, symmetric encryption etc. have introduced LPN problem as the assumption underlying provably secure cryptosystems [86]. Its popularity is due to robust security against quantum algorithms. Unlike most number theoretic problems used in applied cryptography, LPN based constructions are inclined to be extremely efficient in view of computation time and memory requirement which lead LPN based cryptosystem to be a good candidate for resource-constraint devices like RFID tags, smart phone device etc. There has been a lot of research on HB protocol that outputs a number of protocols s.t., HB^{+}, HB^{++}, $\text{HB}^{\#}$, HB-MP, HB-MP^{+}, HB^{*}, F-HB etc. [89, 90, 91, 92, 95, 108, 93]. Unfortunately, most of them later shown to be insecure, or susceptible to particular attacks [94, 95]. In addition, no scheme consider each entity in the RFID system individually against security and privacy threats.

Nowadays, RFID tags are accompanied by robust, globally accessible mobile/wireless readers that contain current and/or historical information on that tagged object's physical properties, origin, ownership, and sensory context. Embedding RFID reader modules into a wireless device such as a smart phone is a new research direction in the RFID environment [126]. Some emerging applications e.g., detecting fraudulent production [83], *green taxi* service where an RFID tag enables the passengers to retrieve the detail travel information using mobile device, transferring ownership transfer in an *inventory management* system [125], using *mobile agents* in RFID management [128] etc. Readers are basically used to circulate around the tagged objects freely and read any tag nearby to build an inventory [127]. Yang *et al.* (in [81]) proposed the first authentication protocol presuming the communication channel between the reader and back-end server to be insecure like wireless channel. Later some other protocols (e.g., [82, 125]) adopted similar assumption. As a result, the whole communication of RFID system is considered to be insecure and an adversary is allowed to impersonate as a legitimate reader.

In this section, we will discuss some of these protocols including their contribution and drawbacks. Let

- a, b: random k-bit binary vectors,
- x, x', y, y': k-bit secret key vectors,
- ν: noise bit(=1 with probability $\eta \in [0, 1/2]$),
- $rot(p, u)$: bitwise left rotate operator such that operand p is rotated u position,
- f: a permutation function,
- ρ: index of the current round,
- $x_{\uparrow}m$: m Left Significant Bits (LSB) of x.
- $f(\cdot)$: a non-linear function.

Table 3.1: One-round HB protocol

Tag (secret x)		Reader (secret x)
	$\longleftarrow$	Generate challenge $a \in_r \{0,1\}^k$
Compute $z = a \cdot x \oplus \nu$	$\longrightarrow$	
		Check $a \cdot x \approx z$

Hopper and Blum (HB) protocol [86]: An overview of a round of HB protocol is given in Table 3.2. Unlike classical symmetric key cryptography solutions such as [91], the HB protocol relies on the computational hardness of the LPN problem. This protocol is secure only against passive attacks, not against active attacks. A simple active attack on HB protocol is: allowing an adversary to transmit a fixed challenge string a to the tag several times in order to retrieve the secret x.

Table 3.2: One-round HB+ protocol

Tag (secret x, y)		Reader (secret x, y)
Generate blinding vector $b \in_r \{0,1\}^k$	$\longrightarrow$	
	$\longleftarrow$	Generate challenger $a \in_r \{0,1\}^k$
Compute $z = a \cdot x \oplus b \cdot y \oplus \nu$	$\longrightarrow$	
		Check $a \cdot x \oplus b \cdot y \approx z$

HB$^+$ protocol [90]: Juels and Weis (2005) presented a modified version of the **HB** protocol and proved the modified protocol (HB$^+$) to be secure against active attacks. A round of HB$^+$ is given in Table 3.2. They introduced an additional shared secret key y, recommended the tag to initiate the protocol. However, this protocol is vulunarable to the well-known MIM attacks [93].

Table 3.3: One round HB^{++} Protocol

Tag (Secret x, y, x', y')		Reader (Secret x, y, x', y')
Generate blinding vector $b \in_r \{0,1\}^k$	$\longrightarrow$	
	$\longleftarrow$	Generate challenger $a \in_r \{0,1\}^k$
Compute $\begin{cases} z = a \cdot x \oplus b \cdot y \oplus \nu \\ z' = rot(f(a), \rho) \cdot x' \\ \qquad \oplus\, rot(f(b), \rho) \cdot y' \oplus \nu' \end{cases}$	$\overrightarrow{(z, z')}$	
		Check $\begin{cases} a \cdot x \oplus b \cdot y \approx z \\ rot(f(a), \rho) \cdot x' \\ \qquad \oplus\, rot(f(b), \rho) \cdot y' \approx z' \end{cases}$

HB^{++} Protocol [91]: To resist the MIM attack on HB^{+}, Bringer *et al.* (2006) proposed the updated version HB^{++} (in Table 3.2) that was assumed to be secure against MIM attacks. The protocol transaction message costs remain same as that of HB+, but the tag computation and storage costs have been increased with bit-wise rotations, small-block permutation f, an additional universal hash function that is required to derive new secrets at the beginning of each authentication.

Table 3.4: One round HB-MP Protocol

Reader (Secret x, y)		Tag (Secret x, y)
Generate challenge vector $a \in_r \{0,1\}^k$	$\longrightarrow$	
		$x = rot(x, y_i)$
		$z = a \cdot x_{\uparrow}m \oplus \nu$
	$\longleftarrow$	Choose b such that $z = b \cdot x_{\uparrow}m$
$x = rot(x, y_i)$		
Check if $a \cdot x_{\uparrow}m = b \cdot x_{\uparrow}m$		

HB-MP protocol [92]: Munilla and Peinado (2007) has introduced a new variant of HB protocol, namely HB-MP protocol (Table 3.2). Although HB-MP protocol reduces the communication cost (in compare to HB^{+}), this protocol was not MIM attack free [108].

Table 3.5: One round HB-MP+ Protocol

Reader (Secret $x, f(\cdot)$)		Tag (Secret $x, f(\cdot)$)
Generate challenge $a \in_r \{0,1\}^k$	$\longrightarrow$	
		$x_s = f(a, x)$
		$z = a \cdot x_s \oplus \nu$
	$\longleftarrow$	Choose b such that $z = b \cdot x_s$
$x_s = f(a, x)$		
Check if $a \cdot x_s = b \cdot x_s$		

HB-MP^{+} protocol [108]: Leng *et al.* (2008) proposed HB-MP^{+} protocol (Table 3.2) to alleviate MIM attack on HB-MP. This protocol introduces one way hash function ($f(\cdot)$) to avoid repeat update of the round key. Nonetheless, **HB-MP^{+}** protocol experience synchronization problem and the hash function is defined in an abstract way [17]. Moreover, the hash function demands at least 5000 logic gates that is not practical for passive RFID tags.

Table 3.6: HB$^{\#}$ Protocol

Reader (Secret X, Y)		Tag (Secret X, Y)
Choose $b \in \mathbb{F}_2^{k_Y}$	$\longrightarrow$	
	$\longleftarrow$	Choose $a \in \mathbb{F}_2^{k_X}$
$z = (a.X) \oplus (b.Y) \oplus \nu$	$\longrightarrow$	
		If $\mathbf{w}(z \oplus (a.X) \oplus (b.Y)) \leq \tau$ Accept

HB$^{\#}$ protocol [93]: Gilber*et al.* (2008) proposed the HB$^{\#}$ protocol (Table 3.2) that is a nature matrix extension of the HB$^{\#}$ protocol where Tag/Reader share two binary matrices $X^{k_x \times m}$, $Y^{k_x \times m}$ instead of two vectors (x, y). HB$^{\#}$ is an improvement to HB^{+} in terms of security and practicality. Likewise HB^{+} protocol, the HB$^{\#}$ has low computational complexity,low transmission costs. Additionally, it provides more practical error rates. However, since the tag needs to store two secret matrices instead of vectors, it requires more memory bits for the secret keys.

Table 3.7: The GHB$^{\#}$ protocol

Reader (Secret X, Y)		Tag (Secret X, Y)
Choose $b \in \mathbb{F}_2^{k_Y}$	$\longrightarrow$	
	$\longleftarrow$	Choose $a \in \mathbb{F}_2^{k_X}$
$z = \phi(a \cdot X) \oplus \phi(b \cdot Y) \oplus \nu$	$\longrightarrow$	
		If $\mathbf{w}(z \oplus \phi(a \cdot X) \oplus \phi(b \cdot Y)) \leq \tau$ Accept

GHB$^{\#}$ protocol [124]: Rizomiliotis *et al.* (2012) presents a non-linear variant of the HB# protocol. Likewise HB#, both the tag and reader share two secret binary matrices X and Y. The single round of the protocol appears in Table 3.2 experience the advantage of the properties of the Gold power functions $\phi(\cdot)$. This protocol is supposed to be secure against all the major attacks including the MIM attack where the attacker can modify all messages exchanged between an honest tag and the reader.

Table 3.8: The F-HB protocol

Reader $[I_i, s_{old}, s_{cur}, \mathrm{ID}]$		Tag $[I_i, s_i]$
$c \in_R \{0,1\}^{m_1}$	$\xrightarrow{c}$	
		$a \in_R \{0,1\}^{m_2}, b \leftarrow g(s_i, a)$
		$v_1 \leftarrow \mathsf{Ber}_\eta^{l_1}, v_2 \leftarrow \mathsf{Ber}_\eta^{l_2}$
		$(z_1, z_2, z_3) \leftarrow (T_{s_i} \cdot (c, b)) \oplus (v_1, v_1, v_2)$
		$t \leftarrow b \oplus (T_{s_i} \cdot (c, z_1, I_i))$
		$(I, I_{i+1}) \leftarrow (I_i, z_2)$
	$\xleftarrow{I, a, z_1, t}$	
If using I as hash-table index: $I = I_i$		
$\quad$ If $t = g(s_{cur}, a) \oplus (T_{s_{cur}} \cdot (c, z_1, I_i))$		
$\quad\quad$ $(z'_1, z'_2, z'_3) \leftarrow T_{s_{cur}} \cdot (c, g(s_{cur}, a))$		
$\quad\quad$ $v'_1 \leftarrow z'_1 \oplus z_1$		
$\quad\quad$ If $\mathbf{w}\ (v'_1) \leq Th$		
$\quad\quad\quad$ $I_{i+1} \leftarrow v'_1 \oplus z'_2$		
$\quad\quad\quad$ $(s_{old}, s_{cur}) \leftarrow (s_{old}, s_{cur} \oplus v'_1)$		
$\quad\quad\quad$ accept the tag		
$\quad\quad$ Else reject the tag		
$\quad$ Else reject the tag		
Else if brute-force search an $[I', s_{old}, s_{cur}, \mathrm{ID}] : \exists s \in (s_{old}, s_{cur})$, $t = g(s, a) \oplus (T_s \cdot (c, z_1, I))$		
$\quad$ $(z'_1, z'_2, z'_3) \leftarrow T_s \cdot (c, g(s, a))$		
$\quad$ $v'_1 \leftarrow z'_1 \oplus z_1$		
$\quad$ If $\mathbf{w}\ (v'_1) \leq Th$		
$\quad\quad$ $I' \leftarrow v'_1 \oplus z'_2$		
$\quad\quad$ $(s_{old}, s_{cur}) \leftarrow (s, s \oplus v'_1)$		
$\quad\quad$ accept the tag		
$\quad$ Else reject the tag		
Else reject the tag		
	$\xrightarrow{z'_3}$	
		If $z'_3 \oplus z_3 = v_3$
		$\quad$ $s_{i+1} \leftarrow s_i \oplus v_1$
		Else reject the reader

F-HB protocol [104]: Cao *et al.* (2011) proposes a forward private mutual authentication scheme based on the unpredictable privacy notion (See Table 3.2). Their building block is composed of two LPN problem and a pseudo random number generator (PRNG). They provide security proof of the proposed authentication scheme under the standard model. However, we carefully observe that the Toeplitz matrix multiplication (EX-OR operation) for the multiple bit LPN problem and MAC generation in the main protocol of are not consistent to matrix size, although the authors did not clarify the specific matrix size in operation; and the threshold value for LPN problem is not specified concretely. Moreover, in the last protocol transcripts, where a tag's secret key is updated, *if-checking*, is not consistent and is not based on the LPN problem; but an EX-OR vector computation. Unlike [104], our protocol follows the SLPN based problem for tag authentication, where the secret key is not a vector but a binary matrix. In addition, we introduce pseudo-inverse matrix for updating the secret key of the tag and apply to the SLPN problem for both the tag and the reader authentication. As a consequence, our proposed protocol is more robust against quantum adversaries while been efficient like the previous HB-protocol family.

Table 3.9: The SLPN protocol

Tag $\mathcal{P}_{\tau,n}(\mathrm{s} \in \mathbb{Z}_2^{2\ell})$		Reader $\mathcal{V}_{\tau',n}(\mathrm{s} \in \mathbb{Z}_2^{2\ell})$
	$\xleftarrow{\mathrm{v}}$	
		$\mathrm{v} \leftarrow \{\mathrm{x} \in \mathbb{Z}_2^{2\ell} : \mathbf{w}(\mathrm{x}) = \ell\}$
If $\mathbf{w}(\mathrm{v}) \neq \ell$ abort		
$\mathrm{R} \leftarrow \mathbb{Z}_2^{\ell \times n}$; $\mathrm{e} \leftarrow \mathsf{Ber}_\tau^n$		
	$\xrightarrow{\mathrm{R}, \mathrm{z}}$	
$\mathrm{z} := \mathrm{R}^\mathsf{T} \cdot \mathrm{s}_{\downarrow \mathrm{v}} \oplus \mathrm{e} \in \mathbb{Z}_2^n$		If $\mathsf{rank}(\mathrm{R}) \neq n$ reject
		If $\mathbf{w}(\mathrm{z} \oplus \mathrm{R}^\mathsf{T} \cdot \mathrm{s}_{\downarrow \mathrm{v}}) > n \cdot \tau'$reject, else accept

SLPN problem protocol[80]: Kiltz *et al.* (2011) builds a new efficient two-round authentication protocol that is secure against active adversaries (See Table 3.2). Its security can be reduced to the subset learning parity with noise (SLPN) problem. In addition, they construct two efficient MACs, and thus two-round authentication protocols that is secure against MIM attacks from the LPN assumption.

Table 3.10: Insecure Reader-Server channel protocol

Server k_1, k_2, C		Reader $r, S = h_k(r)$		Tag k_1, k_2, C
			$\xrightarrow{S}$	
				$ID = h(k_1 \oplus S \oplus C)$
	$\xleftarrow{ID, S, r}$		$\xleftarrow{ID}$	
Verify$S \overset{?}{=} h_k(r)$ (abort if not) then Retrieve $\langle k_1, k_2, C \rangle$ from $\langle T_1, T_2, CN \rangle \in D$ Verify $ID \overset{?}{=} h(k_1 \oplus h_k(r) \oplus C)$ (abort if not) then $ID' = h(k_2)$				
	$\xrightarrow{ID', E_{h_k(S)}(DATA)}$		$\xrightarrow{ID'}$	
				Verify $ID' \overset{?}{=} h(k_2)$
$k_1 \leftarrow k_1 \oplus ID'$ $k_2 \leftarrow k_2 \oplus ID$				(abort if not) then
		$D_{h_k(S)}(DATA)$		$k_1 \leftarrow k_1 \oplus ID'$ $k_2 \leftarrow k_2 \oplus ID$

Insecure Reader-Server communication protocol[81]: Yang *et al.* (in Table 3.2) first introduced a robust mutual authentication protocol between a tag and a back-end server. They assume the reader and the back-end server communication over an insecure channel like wireless and their communications are subject to eavesdropping or modification. Moreover, they assume the readers are not to be a trusted third party. They provide the formal proof of correctness of the proposed protocol on GNY logic.

3.3 RFID ownership transfer

In [154], authors present two models for RFID ownership transfer protocols to protect the security and privacy of the current and new owner. One is ordinary two-party model where a key is shared between the current reader and the tag. The other one is three-party model where an additional key is shared between the trusted third party (TTP) and the tag. Online TTP helps the new owner to update key securely. A number of ownership transfer protocols ([120, 115, 145, 135, 136, 137]) have been proposed based on TTP. However, protocols with trusted party create a bottleneck in the inventory system and overloads the trusted server. For instance, each owner needs to contact directly to a manufacturer (resp. the trusted server). The situation would become worse in case of simultaneous ownership transfer of multiple tags.

In [120], authors propose an ownership transfer protocol to resolve privacy issues of ownership transfer by using pseudonyms and a tree based time-limited key structure. In this scheme, current owner delegates the ownership of a tag temporarily (for a period of time) to a new owner. Nonetheless, each tag requires a counter in the non-volatile memory to count the number of authentications occurred. A trusted center (TC) that stores tag secret keys helps the readers to authenticate the tag. Therefore, this scheme suffers from the similar issues as in TTP-based scheme.

Protocols without trusted party, usually termed as *ownership sharing* protocol in [134, 115, 143, 145] allow sharing a secret key among current and previous owners. This threatens owner's privacy, since former owners as well as current owners can track the same tag legitimately.

In [116], authors introduced some mandatory security properties for ownership transfer protocol by using ordinary hash function: owner initiation, tag assurance, ownership proof, and undeniable transfer. Issuer verification during the ownership transfer has been addressed in [121]. Ownership transfer protocol in [148] has formally defined the security and privacy issues that prevents the attacker from injecting fake tags in the supply chains by verifying the tag prior to the ownership transfer.

A scalable RFID authentication protocol with ownership transfer support has been proposed in [122] that allows the current owners to delegate ownership without using non-volatile memory into the tag to store a counter. However, desynchronization and denial-of-service attack may occur in this scheme without using the TTP [148]. Moreover, storage cost on the server is questionable in this protocol when the maximum size of the hash chains increased. In [139], authors defines two roles regrading ownership: the tag owner and the tag holder. While, both of them can pass the ownership verification, only the tag owner can transfer the ownership. Note that the tag holder may not be the owner in decentralized systems. Security of ownership transfer protocols are based mainly on the authentication of the tag. Almost all the symmetric-key based ownership transfer protocols such as [154, 120]assume that the tag is temper-resistant.

3.4 RFID-enabled path authentication

A number of tag authentication schemes, that have been in the Section 3.2, cannot be used directly for path authentication, either because they incur high computational overhead or they lack simultaneous online access to all the parties in the supply chain. Alternatively, security requirements of the RFID ownership transfer protocols based on RFID authentication in Section 3.3, where ownership of a tag can be transferred securely and privately, are very close to that of path authentication protocols [156, 149]. However, path authentication protocols demand additional privacy requirements: forward and backward privacy. More clearly, the former executions should not be traced by the new owner (forward privacy) and the succeeding transaction should not be traced by the former owner (backward privacy).

There are two kinds of path authentication systems in the literature: *static path*, where a valid *path* is predetermined and is shared with the destination checkpoint (e.g., [151]); and, *dynamic path*, where the path is generated dynamically and every node in the path can track the validation of the path (e.g., [152]).

After the first proposal by Blass *et al.* [150], the construction was improved by Cai *et al.* [151]. Both of them are based on EIGamal encryption to encrypt the identity of the RFID tag and symmetric key primitives to check the path. The path of the RFID tag is predetermined (static) and the issuer of the tag distributes secret keys to the RFID reader. Wang *et al.* proposed a dynamic path authentication protocol (in [153]) based on Hierarchical Identity-Based Encryption (HIBE) and Boneh-Lynn-Shacham (BLS) digital signature scheme [155] where tag's identity is encrypted by the HIBE and the path is generated by hashing the past identities of the reader with digital signature. Some other solutions for the dynamic path authentication includes the schemes in [152, 156]. These schemes are based on Ordered Multi-signature and Pseudo Random Function respectively. Nonetheless, no prior *static path* based authentication schemes consider mutual authentication between the tag and intermediate readers, either because they assume that the communication channel between the reader and tag during path authentication is secure, or because they presume tag authentication implicitly. For instance, in [151], authors assume that the reader will update the tag's state only after successful authentication. However, this scheme does not include any tag authentication explicitly. Some dynamic path authentication schemes (e.g., [156]) incorporate mutual authentication into their proposal.

Chapter 4

Vehicle Network Security

4.1 Secure VANET Applications with a refined Group Signature.

4.1.1 Introduction

Although complete untraceability (strong privacy) among the members is an important properties for applications like WSN where nodes are bounded to places or human body in order to measure data and position, or, VANET where vehicles with On Board Unit (OBU) are considered as preliminary nodes, sometimes stringent privacy policy prevents some reasonable case of application. For example, pseudonym mechanisms (e.g., [46, 20]) and GS scheme (e.g., [50, 66]) are two popular approaches to guarantee privacy in VANET, but sometimes application demands diverse privacy requirement. Members might benefit from established trust relations among them in order to communicate private data in an unobservable manner [55, 56].

For better understanding, from now on we would consider our proposed solution to VANETs only. However, this solution can be applied to any ad hoc network systems where different labels of privacy, jurisdiction access, and revocability are necessary on dense communication. VANET offers two types of wireless communication, namely, V2V$-$ communication among the vehicles, V2I$-$ communication between vehicles and a VANET infrastructure like Road Side Unit (RSU). In this work, we address some real life application scenarios as follows:

Scenario 1. Let a car C be registered to some Value Added Service Provider (VSP) for some special events or services (fuel filling station, garage service, auto mechanic center etc.). Generally service stations need to ensure the right client and services it had agreement to. For instance, C has subscribed to 'gasoline from filling station F' through VSP. VSP issues a *token* regarding C's subscription. When C appears physically to the service station F, it would request for the service providing the *token* it received from the VSP. Note that, a service center can expose C's identity if and only if the *token* admits the service as C is claiming for and is generated from the VSP.

Scenario 2. In *Scenario* 1, we have seen that service provider can revoke signer's anonymity depending on *token.* But in case of culprit members, such as a vehicle involved in an accident, sometimes it becomes essential for the Traffic Security Division (TSD) to forcefully revoke the signer's identity.

Scenario 3. Let an accident occur and vehicles in the vicinity of the accident start broadcasting warning messages through V2V communication. Car C that moves towards the accident area, would receive more warning messages even from the same sender including periodical broadcast messages from other vehicles. C must conceive the validity of these messages in order to decide the next route. Note that, VANET allows maximum message processing time to be 300 *ms* [50]. Using batch verification is one of the solutions to verify a batch of signatures quickly. However, batch verification is not always efficient if the number of messages to batch is not decided intelligently [50], or if the number of bogus messages in a single batch is more than 15% [54].

Scenario 4. In addition to *Scenario 3*, a signature verifier may need additional processing time when it considers local revocation check. Group signature approach with VLR (e.g., [62]) incurs expensive verification phase specially for a long-sized *revocation list*. Moreover, revocation list grows linearly with time when new revoked members are added into the list unless member keys with public parameters are reinitialized (called re-keying). Nonetheless, re-keying process is not feasible, and hence, is often pre-scheduled to get rid of the burden of communication overhead.

Scenario 5. Let a licit (may be hijacked) vehicle keep sending doubtful messages for a number of times. In general case, the messages together with signatures would be forwarded to the TSD (tracer in GS) to revoke. But it is not always wise to request TSD for every single suspicious message. It would convey serious burden to the TSD.

Main challenges in the security proposals of VANET are to connect security, privacy, efficiency and management capability. *Scenario 1-5* are some real life problems that can be solved using GS approach. Prior works in this field try to solve some of these problems scatteredly in different schemes. In this work, we tried to solve all the aforementioned problems in a *single* scheme efficiently. To the best of our knowledge, this is a complete GS scheme from short BBS GS where almost all the GS properties (available in the literatures) are accumulated.

4.1.2 Our Contribution

We introduce a short GS scheme based on [59] with additional properties for a large scale VANET: (1) selective linkability, (2) direct traceability, (3) message-dependent traceability, (4) hybrid revocability with constant computation. Our proposed solution is more application-friendly than the related works. Clearly, we focus on solving some real-life problems described in *Scenario 1-5* efficiently.

- We propose two new authorities, namely Admitter and Linker before revoking a signer's anonymity. Linker can partially break anonymity by linking the signatures from the same signer (without exploring member identification) while Admitter assists Opener to break full anonymity (by exposing member identification). It introduces a fine-grained control on the anonymity of the members.

- We suggest two different algorithms for traceability, namely Direct tracing and Attested tracing. Direct tracing algorithm can trace any signer directly with its own key. On the other hand, Attested tracing algorithm rely on the *token* issued by Admitter to trace a signer [53].

- We introduce a hybrid revocation algorithm with limited VLR and rekeying process. To avoid the inefficient checking of RL during signature verification, our proposal uses 0/1 *encoding*-enabled signing and verfication and the expired-date bound signing key [51]. This encoding system enables *set intersection* predicate in [49]. With this property, if there is a common element between two sets of encoded expired dates (signer's key and signature), verifier will pass the signature.

- To solve *Scenario 1*, in V2I communication, our proposal uses the modified scheme of [53]. For value added service, let a vehicle C request VSP (Message attestation authority) to generate a *token* T_c regarding the service (e.g., fueling) to subscribe. When C will go into the subscribed service station e.g., fuel filling station (Attested tracing authority), first it verify the signature on service, later, it will check whether the token T_c was generated by VSP on the same service. Note that it can only expose C's identity if and only if T_c admits the service C is claiming for.

- To solve *Scenario 2*, in I2I communication, RSU will request TSD (Direct Tracing authority) with culprit member's generated message and signature who can forcibly revoke signer's identity.

- To solve *Scenario 3*, in V2V communication, verifier should first check whether batch verification is feasible for the current situation following algorithm in [50]. If yes, it uses efficient batch verification process to verify a bunch of signatures together. In addition, it can adapt categorized verification (in [70]) by providing linking key (Managing linkability algorithm) to the vehicle where the signatures from the *known* vehicles are batched together in order to resist bogus messages in the batch. Note that the verifier recognizes a vehicle to be known if the incoming signature is linkable to the former signature it received.

- To solve *Scenario 4*, we propose the revocation system to comply with both VLR and rekeying process. To optimize the cost of VLR checking we propose a revocability-enabled credentials with natural expiration date that is generally used for authentication in mobile roaming [52]. It helps the verifier to ascertain that the message is not generated by an expired signer key at a *fixed* cost. We use the modified VLR scheme from [51]. Note that our limited version of VLR is more efficient, but do not consider the members that are forcedly revoked prematurely. Nonetheless, our rekeying system from [47] will take care of that. This hybrid approach will lead to a substantial reduction (constant) on revocation check (for each message) specially in a situation where prematurely revoked credentials are very few in number.

- To solve *Scenario 5*, we propose a novel solution with short-term linkability where vehicle will forward messages with signatures to some designated entity like RSU. Let an RSU have the linking key and a counter q. It increases the counter value by 1 after it receives any suspicious message from the identical vehicle (by linking signatures). According to some preset value of the counter, RSU would finally request the TSD to revoke the member from the group.

To the best of our knowledge, there is no GS scheme proposed in the literature that satisfy all the aforementioned properties together. We accumulate the cited properties in a single scheme and this challenging effort helps to induce relaxation from a strong privacy to a scheme with a lesser but adaptive privacy hierarchy, and hence make the GS scheme applicable to certain application environment by being simplistic, yet efficient way.

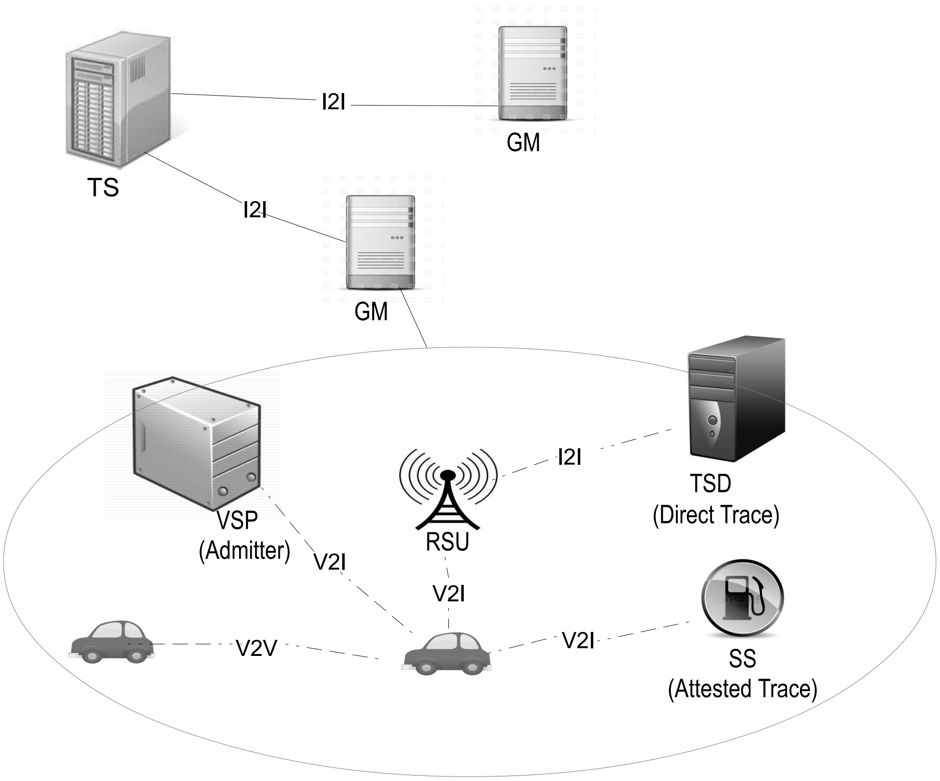

Figure 4.1: VANET Security Model.

4.1.3 Network model and Scheme Description

We refer to a symbolic hierarchical network model for VANET described in Fig.4.1. It consists of a Trusted System (TS), a Group Manager (GM), Traffic Security Division (TSD), Value-added Service Provider (VSP), Service Station (SS), and Members (Vehicle, RSU). Vehicular groups could be formed by region, social spots/services, vehicle category etc. Each vehicle in the network is equipped with an On Board Unit (OBU) consisting of an Event Data Recorder (EDR) that records all the received messages and a Tamper Proof Device (TPD) that implements cryptographic tools. Three types of communication exist in the network: Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I) and Infrastructure to Infrastructure (I2I).

- TS creates and manages the groups in the network. It generates the public parameters for cryptographic operation.

- GM manages the registration of the members such as vehicles, RSUs by providing group secret keys with expiration date. It is securely connected to other pertaining authorities like VSP, TSD, SS. It periodically announces the new group public key for revocation (rekeying). We assume the GM to be honest and secure. However, it cannot reveal any member's identification.

- Admitter works for the Attested Trace authority. It generates token for the vehicles according to their subscription.

- Attested trace authorities are service stations (SS) approved by GM. It provides services to the subscribed vehicles upon receiving the token generated by Admitter (VSP).

- Direct Trace authority is securely connected with RSUs. It can trace and open the member's identity upon request (by the designated RSUs).
- Members includes RSUs and vehicles with embedded OBUs. They collect certificates from GM during registration. Vehicles can communicate with other vehicles through V2V communication. Moreover, they can communicates with RSU through V2I communication to report any malicious message (vehicles are not allowed to communicate directly to TSD).

4.1.4 Security Requirement

For security model, we extend the definition of [59, 53, 64, 70]. We define correctness, anonymity, admitter-anonymity, direct traceability, attested-traceability by the following games.

$\mathsf{AddU}(i)$: Add User oracle adds an honest user to the set **HU** with $gsk[i]$ by using registration protocol.

$\mathsf{CorU}(i)$: Corrupt User oracle adds a corrupt user to the set **CU** with $gsk[i]$ of user i.

$\mathsf{StoU}(i, M)$: Send-to-User oracle sets public/secret key pair to a user i and add i to **HU** set. It allows the adversary to engage in registration protocol with message M. The response of the protocol is returned to adversary.

$\mathsf{StoI}(i, M)$: Send to Issuer oracle, a corrupted user i sends M to the honest Issuer.
$\mathsf{RR}(i)$: Read Registration oracle retrieves the corresponding registration table entry $reg[i]$ in input i.

$\mathsf{WR}(i, \rho)$: Write Registration oracle writes or modifies $reg[i]$ with ρ.

$\mathsf{RS}(i)$: Reveal-secret oracle discloses the secret key $gsk[i]$ and an honest user is turned into a corrupted user.

$\mathsf{GSig}(i, d, M)$: Signing oracle returns a signature σ on the message M and a date d where i is under the set **HU**.

$\mathsf{Ch_b}(M, i_0, i_1)$: Challenge oracle returns i_b's signature σ_{i_b} for a random bit b and records (σ_{i_b}, M) in the message-signature pair set **GSet**.

$\mathsf{Dtrace}(M, \sigma)$: The Direct open oracle returns the identity i of σ on M.

$\mathsf{Open}(M, \sigma, t_M)$: The Attested open oracle returns identity i of σ based on t_M.

$\mathsf{Link}(M_0, \sigma_0, M_1, \sigma_1)$: This oracle returns 1 if two signatures are generated by the same user in the set **GSet**.

$\mathsf{TG}(M)$: The token generation oracle returns a token t_M to the message M.

Correctness: If a signature σ is generated by an honest member with a non-expired key will be verified correctly. On input a message M, a signature σ, a current date d, a token t_M on M: GVerify should verify the signature correctly, Open and DTrace should correctly identify the signer, Link should link the signatures from a signer. We say that the group signature scheme is correct $\Pr[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{correct}}(\lambda) = 1] = 0$.

```
$\mathsf{Exp}_{\mathcal{A}}^{\mathsf{correct}}(\lambda)$:
$(gpk, aok, dok, ak, lk, (gsk_i)_{i\in[1,n]}) \leftarrow \mathsf{SetUp}(1^\lambda)$;
$\mathbf{CU} \leftarrow \emptyset; \mathbf{HU} \leftarrow \emptyset$;
$(i, M) \leftarrow \mathcal{A}(gpk : \mathsf{AddU}, \mathsf{RR})$;
If $(i \notin \mathbf{HU})$ and $(gsk_i = \varepsilon)$ then return 0;
$\sigma \leftarrow \mathsf{GSig}(gpk, gsk_i, d, M)$;
If $\mathsf{GVerify}(gpk, M, d', \sigma) = 0$ then return 1;
  $j \leftarrow \mathsf{Open}(gpk, aok, reg_1[i], M, t_M, \sigma)$;
If $i \neq j$ then return 1;
  $j \leftarrow \mathsf{DTrace}(gpk, dok, reg_2[i], M, \sigma)$;
If $i \neq j$ then return 1;
$(i, M_0, j, M_1) \leftarrow \mathcal{A}(gpk : \mathsf{AddU}, \mathsf{RR})$;
If $i = j$, then
  $\sigma_0 \leftarrow \mathsf{GSig}(gpk, gsk_i, d, M_0)$;
  $\sigma_1 \leftarrow \mathsf{GSig}(gpk, gsk_i, d, M_1)$;
  $b \leftarrow \mathsf{Link}(gpk, lk, M_0, \sigma_0, M_1, \sigma_1)$;
  If $b = 0$ then return 1 Else return 0;
```

Signature Anonymity: Our group signature scheme has anonymity if for all probabilistic polynomial time adversaries $\mathcal{A}$, a bit $b \in \{0, 1\}$, the advantage of $\mathcal{A}$ in the following game $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{anon-b}}$ with a challenger $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{anon}}(\lambda) = |\Pr[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{anon}-1}(\lambda) = 1] - \Pr[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{anon}-0}(\lambda) = 1]|$ is negligible. We say that our GS is *CCA-anonymous* if $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{anon}}$ is negligible in the security parameter λ.

```
$\mathsf{Exp}_{\mathcal{A}}^{anon-b}(\lambda)$:
$(gpk, aok, dok, ak, lk, (gsk_i)_{i\in[1,n]}) \leftarrow \mathsf{SetUp}(1^\lambda)$;
$\mathbf{CU} \leftarrow \emptyset$; $\mathbf{HU} \leftarrow \emptyset$; $\mathbf{GSet} \leftarrow \emptyset$;
$(i_0, i_1, t_k, M) \leftarrow \mathcal{A}(gpk, (gsk_i)_{i\in[1,n]} :$ STolU, WR, RS, CorU$)$;
  where $\{t_k = d_{i_0k} = d_{i_1k}\}_{k\in[1,l]}, \{t_j\}_{j\in[1,l]} \leftarrow$0-ENC$(t)$, $\{d_{ij}\}_{j\in[1,l]} \leftarrow$1-ENC$(d_i)$;
$\sigma_{ib} \leftarrow \mathsf{Ch}_b(M, i_0, i_1)$;
$\theta \leftarrow \mathcal{A}(gpk, (gsk_i)_{i\in[1,n]}, t_k, \sigma_{ib} :$ SToU, WR, RS, CorU$)$,
where $\mathcal{A}$ is not allowed to query $\mathsf{Link}(\sigma_{ib}, M, \cdot, \cdot)$ oracle (for either $b = 0$ or $b = 1$);
Return $\theta$.
```

Attested Opener Anonymity: Attested Opener has anonymity if it is unable to identify the signer without cooperation with Admitter, even if some group members are corrupted. Formally, for all PPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following game with a challenger, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{anon\_open}}(\lambda) = |\Pr[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{anon\_open}-1}(\lambda) = 1] - \Pr[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{anon\_open}-0}(\lambda) = 1]|$ is negligible.

```
 $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{anon\_open-b}}(\lambda)$:
  $(gpk, aok, dok, ak, lk, (gsk_i)_{i\in[1,n]}) \leftarrow \mathsf{SetUp}(1^\lambda)$;
  $\mathbf{CU} \leftarrow \emptyset$; $\mathbf{HU} \leftarrow \emptyset$; $\mathbf{GSet} \leftarrow \emptyset$;
  $(i_0, i_1, t_M, M) \leftarrow \mathcal{A}(gpk, aok :\ \mathsf{SToU}, \mathsf{TG}, \mathsf{WR}, \mathsf{RS}, \mathsf{CorU})$;
  $\sigma_{ib} \leftarrow \mathsf{Ch}_b(M, i_0, i_1)$;
  $\theta \leftarrow \mathcal{A}(gpk, aok, \sigma_{ib} :\ \mathsf{SToU}, \mathsf{TG}, \mathsf{WR}, \mathsf{RS}, \mathsf{CorU})$,
  where $\mathcal{A}$ is not allowed to query $\mathsf{Link}(\sigma_{ib}, M, \cdot, \cdot)$ oracle (for either $b = 0$ or $b = 1$);
  Return $\theta$.
```

Admitter Anonymity: Admitter has anonymity if it is unable to identify the signer without cooperation with Attested Opener, even if some group members are corrupted. Formally, for all PPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following game with a challenger, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{anon\_admit}}(\lambda) = |\Pr[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{anon\_admit-1}}(\lambda) = 1] - \Pr[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{anon\_admit-0}}(\lambda) = 1]|$ is negligible.

```
 $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{anon\_admit-b}}(\lambda)$:
  $(gpk, aok, dok, ak, lk, (gsk_i)_{i\in[1,n]}) \leftarrow \mathsf{SetUp}(1^\lambda)$;
  $\mathbf{CU} \leftarrow \emptyset$; $\mathbf{HU} \leftarrow \emptyset$; $\mathbf{GSet} \leftarrow \emptyset$;
  $(i_0, i_1, t_M, M) \leftarrow \mathcal{A}(gpk, ak :\ \mathsf{SToU}, \mathsf{Open}, \mathsf{WR}, \mathsf{RS}, \mathsf{CorU})$;
  $\sigma_{ib} \leftarrow \mathsf{Ch}_b(M, i_0, i_1)$;
  $\theta \leftarrow \mathcal{A}(gpk, ak, \sigma_{ib} :\ \mathsf{SToU}, \mathsf{Open}, \mathsf{WR}, \mathsf{RS}, \mathsf{CorU})$,
  where $\mathcal{A}$ is not allowed to query $\mathsf{Link}(\sigma_{ib}, M, \cdot, \cdot)$ oracle (for either $b = 0$ or $b = 1$);
  Return $\theta$.
```

Attested Traceability: Our group signature scheme has attested traceability. That is, even if the admitter and opener collude, they cannot produce any forged or untraceable signature. Formally, if for all probabilistic polynomial time adversaries $\mathcal{A}$, the success probability of $\mathcal{A}$ in the following game $\Pr[\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Open}}(\lambda) = 1]$ with a challenger is negligible in the security parameter λ.

```
 $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{Open}}(\lambda)$:
  $(gpk, aok, dok, ak, lk, (gsk_i)_{i\in[1,n]}) \leftarrow \mathsf{SetUp}(1^\lambda)$;
  $\mathbf{CU} \leftarrow \emptyset$; $\mathbf{HU} \leftarrow \emptyset$;
  $(M, t_M, \sigma) \leftarrow \mathcal{A}(gpk, aok, ak :\ \mathsf{AddU}, \mathsf{SToI}, \mathsf{RR}, \mathsf{CorU}, \mathsf{RS}, \mathsf{TG})$;
  If $\mathsf{GVerify}(gpk, M, d', \sigma) = 0$ then return 0;
  $i \leftarrow \mathsf{Open}(gpk, aok, reg_1[i], M, \mathsf{TAtd}(gpk, ak, M), \sigma)$;
  If $i = 0$ then return 1 Else return 0.
```

Direct Traceability: Our group signature scheme has direct traceability if for all probabilistic polynomial time adversaries $\mathcal{A}$, the success probability of $\mathcal{A}$ in the following game $\Pr[\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DTrace}}(\lambda) = 1]$ with a challenger is negligible in the security parameter λ.

```
 $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{Dtrace}}(\lambda)$:
  $(gpk, aok, dok, ak, lk, (gsk_i)_{i\in[1,n]}) \leftarrow \mathsf{SetUp}(1^\lambda)$;
  $\mathbf{CU} \leftarrow \emptyset$; $\mathbf{HU} \leftarrow \emptyset$;
  $(M, \sigma) \leftarrow \mathcal{A}(gpk, dok :\ \mathsf{AddU}, \mathsf{SToI}, \mathsf{RR}, \mathsf{CorU}, \mathsf{RS})$;
  If $\mathsf{GVerify}(gpk, M, d', \sigma) = 0$ then return 0;
  $i \leftarrow \mathsf{DTrace}(gpk, dok, reg_2[i], M, \sigma)$;
  If $i = 0$ then return 1 Else return 0.
```

Linkability: Our GS scheme has linkability. Any colluding members should not be able to generate two message-signature pairs even with the help of the Linker or Direct opener.

Formally, the success probability of $\mathcal{A}$ in the following game $\Pr[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{Link}}(\lambda) = 1]$ with a challenger is negligible in the security parameter λ.

```
$\mathsf{Exp}_{\mathcal{A}}^{\mathsf{SignLink}}(\lambda)$:
$(gpk, aok, dok, ak, lk, (gsk_i)_{i\in[1,n]}) \leftarrow \mathsf{SetUp}(1^\lambda)$;
$(M_0, \sigma_0, M_1, \sigma_1) \leftarrow \mathcal{A}(gpk, dok, lk : \mathsf{SToU}, \mathsf{RR}, \mathsf{CorU}, \mathsf{GSig}, \mathsf{RS})$;
If $\mathsf{GVerify}(gpk, M_b, d', \sigma_b) = 0$ (either $b = 0$ or $b = 1$ ) then return 0;
$i_0 \leftarrow \mathsf{DTrace}(gpk, dok, reg_2[i], M_0, \sigma_0)$;
$i_1 \leftarrow \mathsf{DTrace}(gpk, dok, reg_2[i], M_1, \sigma_1)$;
If $i_0 \neq i_1$ and $1 = \mathsf{Link}(gpk, lk, M_0, \sigma_0, M_1, \sigma_1)$ then return 1;
Elseif $i_0 = i_1$ and $0 = \mathsf{Link}(gpk, lk, M_0, \sigma_0, M_1, \sigma_1)$ then return 1;
Else return 0.
```

4.1.5 Our Proposal

Our scheme employs the GS scheme in [53] which expands the BBS GS scheme in [59] by replacing the linear encryption with multiple encryption of ordinary PKE and Identity based encryption (IBE). Additionally, we extend the GS with several potential functionality for VANET, such as, revocation following works in ([51, 47]), batch verification with ([54, 70, 50]) and direct traceability from [70], linkability with [64].

Let g is a generator of $\mathbb{G}$. The possession of SDH tuple is (A, x) where $A \in \mathbb{G}, x \in \mathbb{Z}_p, w = g^\gamma$ such that $A^{\gamma+x}$. This can be verified by $e(A, wg^x) = e(g, g)$. The short GS in ([59, 70]) is based on the Signature Proof of Knowledge (SPK):$\{(A, x) : A^{\gamma+x} = g\}(M)$ on message M. Since our secret keys are associated with an additional expiration date d, we modify the underlying signature. The possession of a tuple (A, x) such that $A^{\gamma d+x} = g$ can be verified by $e(A, w^d g^x) = e(g, g)$. Hence, SPK:$\{(A, x) : A^{\gamma d+x} = g\}(M)$.

System Setup: Consider a probabilistic polynomial time algorithm $\mathcal{G}(1^\lambda)$ with a security parameter 1^λ that generates a parameter of bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e, g)$. The proposed scheme uses two hash functions $H_1 : \{0,1\}^* \to \mathbb{G}$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_p$ that are modeled as random oracles in the security analysis.

Issuing Credentials $\mathsf{GKgen}(1^\lambda, 1^l, 1^n)$: On input security parameter 1^λ, the maximum length of the date format l and the maximum number of vehicles n, this algorithm selects random integers $\xi_1, \xi_2, \xi_3, \zeta, \gamma, \ell \leftarrow \mathbb{Z}_p$ and random elements $u, v, h \leftarrow \mathbb{G}\backslash\{1\}$. Then it sets $g_1 \leftarrow u^{\xi_1} h^{\xi_3},\ g_2 \leftarrow v^{\xi_2} h^{\xi_3},\ y \leftarrow g^\zeta,\ w \leftarrow g^\gamma$, and $f \leftarrow u^\ell$. The algorithm computes $\{d_{ij}\}_{j\in[1,l]} \leftarrow \text{1-}\mathsf{ENC}(d_i)$, where d_i is the expiration date of a signer i. The algorithm then selects $x_{ij} \leftarrow \mathbb{Z}_p$ and sets $A_{ij} \leftarrow g^{1/(\gamma d_{ij}+x_{ij})}$ such that $\gamma d_{ij} + x_{ij} \neq 0$ for each vehicle i $(i \in [1, n])$. Finally, the algorithm outputs:

- Group public key
  $gpk \leftarrow (p, \mathbb{G}, \mathbb{G}_\mathbb{T}, f, e, g, u, v, h, g_1, g_2, y, w, H_1, H_2)$
- Signing key $gsk_i \leftarrow (A_{ij}, x_{ij}, d_i)_{i\in[1,n],\ j\in[1,l]}$
- Linking key $lk \leftarrow h^\ell$
- Registration table
  $reg_1[i]_{1\le i\le n} \leftarrow \{A_{ij}, e(A_{ij}, g)\}_{i\in[1,n],\ j\in[1,l]}$
  $reg_2[i]_{1\le i\le n} \leftarrow \{x_{ij}, e(g, g)^{x_{ij}}\}_{i\in[1,n],\ j\in[1,l]}$

- Admitter key $ak \leftarrow \zeta$
- Direct tracing key $dok \leftarrow (\ell, reg_2[i]_{1 \leq i \leq n})$
- Attested tracing key $aok \leftarrow (\xi_1, \xi_2, \xi_3,\ reg_1[i]_{1 \leq i \leq n})$

Signature Generation $\mathsf{GSign}(gpk, t, i, gsk_i, M)$: On input the group public key gpk, user i, the signing key $gsk_i \leftarrow (A_{ij}, x_{ij}, d_i)_{i \in [1,n],\ j \in [1,l]}$, the signature expiration date t, and a message M, this algorithm generates a group signature σ as follows.

- If $t \geq d_i$, output $\perp$.
- Compute $\{d_{ij}\}_{j \in [1,l]} \leftarrow 1\text{-}\mathsf{ENC}(d_i)$ and $\{t_j\}_{j \in [1,l]} \leftarrow 0\text{-}\mathsf{ENC}(t)$. Find an index $k \in [1, l]$ such that $d_{ik} = t_k$.
- Choose random $\alpha, \beta, \rho, \eta \leftarrow \mathbb{Z}_p$ and compute

  $$\begin{aligned}
  &(T_1, T_2, T_3, T_4) \leftarrow (u^\alpha, v^\beta, h^{\alpha+\beta}, g_1^\alpha g_2^\beta A_{ik} g^\eta)\\
  &(T_5, T_6, T_7) \leftarrow (g^\rho, e(y, H_1(M))^\rho e(g, g)^{-\eta}, g^{1/x_{ik}} f^\alpha)
  \end{aligned}$$

- Choose blinding values randomly $r_\alpha, r_\beta, r_\rho, r_\eta, r_x, r_{\alpha x}, r_{\beta x}, r_{\rho x}, r_{\eta x} \leftarrow \mathbb{Z}_p$ and compute

  $$\begin{aligned}
  &R_1 \leftarrow u^{r_\alpha},\\
  &R_2 \leftarrow v^{r_\beta},\\
  &R_3 \leftarrow h^{r_\alpha + r_\beta},\\
  &R_4 \leftarrow e(T_4, g)^{r_x} e(g_1, w)^{-r_\alpha d_{ik}} e(g_1, g)^{-r_{\alpha x}} e(g_2, w)^{-r_\beta d_{ik}}\\
  &\qquad .e(g_2, g)^{-r_{\beta x}} e(g, w)^{-r_\eta d_{ik}} e(g, g)^{-r_{\eta x}},\\
  &R_5 \leftarrow g^{r_\rho},\\
  &R_6 \leftarrow e(y, H_1(M))^{r_\rho} e(g, g)^{-r_\eta},\\
  &R_7 \leftarrow T_1^{r_x} u^{-r_{\alpha x}},\\
  &R_8 \leftarrow T_2^{r_x} v^{-r_{\beta x}},\\
  &R_9 \leftarrow T_5^{r_x} g^{-r_{\rho x}},\\
  &R_{10} \leftarrow T_6^{r_x} e(y, H_1(M))^{-r_{\rho x}} e(g, g)^{r_{\eta x}},\\
  &R_{11} \leftarrow e(T_7, g)^{r_x} e(f, g)^{-r_{\alpha x}}
  \end{aligned}$$

- Compute $c \leftarrow H_2(t, M, T_1, \ldots, T_7, R_1, \ldots, R_{11})$, and then compute

  $$\begin{aligned}
  &s_\alpha \leftarrow r_\alpha + c\alpha,\\
  &s_\beta \leftarrow r_\beta + c\beta,\\
  &s_\rho \leftarrow r_\rho + c\rho,\\
  &s_\eta \leftarrow r_\eta + c\eta,\\
  &s_x \leftarrow r_x + cx_{ik},\\
  &s_{\alpha x} \leftarrow r_{\alpha x} + c\alpha x_{ik},\\
  &s_{\beta x} \leftarrow r_{\beta x} + c\beta x_{ik},\\
  &s_{\rho x} \leftarrow r_{\rho x} + c\rho x_{ik},\\
  &s_{\eta x} \leftarrow r_{\eta x} + c\eta x_{ik},
  \end{aligned}$$

- Output group signature on message M:
  $\sigma \leftarrow (t, k, T_1, \ldots, T_7, c, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}, s_{\eta x})$.

Intuition: The above $R_1, \ldots, R_{11}, c, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}$, and $s_{\eta x}$ prove knowledge $\alpha, \beta, \rho, \eta$, and x_{ij} satisfying the equations

$$\begin{aligned}
&T_1 = u^\alpha,\\
&T_2 = v^\beta,\\
&T_3 = h^{\alpha+\beta}\\
&T_5 = g^\rho,\\
&T_6 = e(y, H_1(M))^\rho e(g,g)^{-\eta}.\\
&e(g,g) = e(T_4 g_1^{-\alpha} g_2^{-\beta} g^{-\eta}, w^d g^x),\\
&e(g,g) = e(T_7 f^{-\alpha}, g^x)
\end{aligned}$$

Clearly, by using four intermediate variables $\delta_1, \delta_2, \delta_3$, and δ_4 (such that $\delta_1 = \alpha x, \delta_2 = \beta x, \delta_3 = \rho x$, and $\delta_4 = \eta x$), the underlying protocol proves knowledge of $\alpha, \beta, \rho, \eta, x, \delta_1, \delta_2, \delta_3$, and δ_4 satisfying the equations

$$\begin{aligned}
&T_1 = u^\alpha,\\
&T_2 = v^\beta,\\
&T_3 = h^{\alpha+\beta}\\
&T_5 = g^\rho,\\
&T_6 = e(y, H_1(M))^\rho e(g,g)^{-\eta},\\
&1 = T_1^x u^{-\delta_1},\\
&1 = T_2^x v^{-\delta_2},\\
&1 = T_5^x g^{-\delta_3},\\
&1 = T_6^x e(y, H_1(M))^{-\delta_3} e(g,g)^{\delta_4},\\
&e(g,g)/e(T_4, w^d) = e(T_4, g)^x e(g_1, w^d)^{-\alpha} e(g_1, g)^{-\delta_1}\\
&\qquad\qquad .e(g_2, w^d)^{-\beta} e(g_2, g)^{-\delta_2} e(g, w^d)^{-\eta} e(g,g)^{-\delta_4},\\
&e(g,g) = e(T_7, g)^x e(f, g)^{-\delta_1}.
\end{aligned}$$

Signature verification $\mathsf{GVerify}(gpk, \bar{t}, M, \sigma)$: On input group public key gpk, current date $\bar{t}$, and the signature σ on message M, this algorithm verifies the validity of the signature and ensures that σ is not generated by a revoked user. It verifies the signature in the following steps.

- If $\bar{t} > t$, output $\perp$.
- $\{t_j\}_{j \in [1,l]} \leftarrow \mathsf{0\text{-}ENC}(t)$.
- Recompute $R'_1, R'_2, R'_3, R'_4, R'_5, R'_6, R'_7, R'_8, R'_9, R'_{10}$ and R'_{11} as follows

$$\begin{aligned}
&R'_1 \leftarrow u^{s_\alpha} T_1^{-c},\\
&R'_2 \leftarrow v^{s_\beta} T_2^{-c},\\
&R'_3 \leftarrow h^{s_\alpha + s_\beta} T_3^{-c},\\
&R'_4 \leftarrow e(T_4, g)^{s_x} e(g_1, w)^{-s_\alpha t_k} e(g_1, g)^{-s_{\alpha x}} e(g_2, w)^{-s_\beta t_k}\\
&\qquad .e(g_2, g)^{-s_{\beta x}} e(g, w)^{-s_\eta t_k} e(g,g)^{-s_{\eta x}},\\
&\qquad .(e(g,g)/e(T_4, w^{t_k}))^{-c},
\end{aligned}$$

$R'_5 \leftarrow g^{s_\rho} T_5^{-c}$,
$R'_6 \leftarrow e(y, H_1(M))^{s_\rho} e(g,g)^{-s_\eta} T_7^{-c}$,
$R'_7 \leftarrow T_1^{s_x} u^{-s_{\alpha x}}$,
$R'_8 \leftarrow T_2^{s_x} v^{-s_{\beta x}}$,
$R'_9 \leftarrow T_5^{s_x} g^{-s_{\rho x}}$,
$R'_{10} \leftarrow T_6^{s_x} e(y, H_1(M))^{-s_{\rho x}} e(g,g)^{s_{\eta x}}$.
$R'_{11} \leftarrow e(T_7, g)^{s_x} e(f,g)^{-s_{\alpha x}} e(g,g)^{-c}$

- Verify whether the equation
  $c \stackrel{?}{=} H_2(t, M, T_1, \ldots, T_7, R'_1, \ldots, R'_{11})$
  holds. If the equation holds, the algorithm outputs 1, otherwise outputs $\perp$.

Batch verification $\mathsf{BVerify}(gpk, \bar{t}, (M_1, \ldots, M_\eta), (\sigma_1, \ldots, \sigma_\eta))$: Computing R'_4, R'_{11} are the most expensive part of the verification algorithm. However, we need to increase the signature size by six elements $(R_4, R_7, \ldots, R_{11})$ to accelerate the verification procedure. Let $\sigma_j \leftarrow (t_j, k_j, T_{j,1}, \ldots, T_{j,7},\ R_{j,4}, R_{j,7}, \ldots, R_{j,11}, c_j, s_{j,\alpha}, s_{j,\beta}, s_{j,\rho}, s_{j,\eta}, s_{j,x}, s_{j,\alpha x}, s_{j,\beta x}, s_{j,\rho x}, s_{j,\eta x})$ be the new j^{th} signature on the message M_j for $j \in [1, \eta]$. Now we define a batch verifier where the main goal is to minimize the number of pairing calculation. For each $j \in [1, \eta]$, compute only $(R'_{j,1}, R'_{j,2}, R'_{j,3}, R'_{j,5}, R'_{j,6})$ following the above mentioned way. For each $j \in [1, \eta]$, check that $c_j \stackrel{?}{=} H_2(t_j, k_j, T_{j,1}, \ldots, T_{j,7}, R_{j,1}, \ldots, R_{j,11})$. Then check the following pairing based equation:

$$\prod_{j=1}^{\eta} R_{j,4}^{\delta_j} \stackrel{?}{=} e(\prod_{j=1}^{\eta} (T_{j,4}^{s_{j,x}} \cdot g_1^{-s_{j,\alpha x}} \cdot g_2^{-s_{j,\beta x}} \cdot g^{-s_{j,\eta x} - c_j})^{\delta_j}, g)$$
$$\cdot e(\prod_{j=1}^{\eta} (g_1^{-s_{j,\alpha t_k}} \cdot g_2^{-s_{j,\beta t_k}} \cdot g^{-s_{j,\eta t_k}} \cdot T_4^{-c_j t_k})^{\delta_j}, w)$$

$$\prod_{j=1}^{\eta} R_{j,11}^{\delta_j} \stackrel{?}{=} e(\prod_{j=1}^{\eta} (T_{j,7}^{s_{j,x}} \cdot f^{-s_{j,\alpha x}} \cdot g^{-c_j})^{\delta_j}, g)$$

and

$$1_{\mathbb{G}} \stackrel{?}{=} (R_{j,7} R_{j,8} R_{j,9} R_{j,10})^{-\delta_j} (T_{j,1} T_{j,2} T_{j,5} T_{j,6})^{-\delta_j s_{j,x}}\ u^{-s_{j,\alpha x}} v^{-s_{j,\beta x}} g^{-s_{j,\rho x}} e(y_j, H_1(M_j))^{-s_{j,\rho x}}$$
$$e(g,g)^{s_{j,\eta x}}$$

where $(\delta_1, \ldots, \delta_\eta) \in \mathbb{Z}_p$ is a random vector of l_b bit. Accept if and only if all checks pass successfully.

Message attestation $\mathsf{TAtd}(gpk, ak, M)$: Given attestation $ak = \zeta$, and M, the algorithm generates a token t_M on M such that $t_M \leftarrow H_1(M)^\zeta$ and outputs t_M. This token can be used together with $\mathsf{Open}(gpk, ok, M, \sigma, t_M)$ algorithm to extract signer's identity.

Attested tracing$\mathsf{Open}(gpk, aok, M, \sigma, t_M)$: Given gpk, aok, M, σ, and a token t_M on message M, this algorithm first verifies the signature using the algorithm $\mathsf{GVerify}$. If the signature is invalid, the algorithm outputs $\perp$. Otherwise, it searches i in the registration table $reg_1[i]$ to find $e(A_{ij}, g) \leftarrow reg[i]$ that satisfies the following equation $e(\frac{T_4}{T_1^{\zeta_1} T_2^{\zeta_2} T_3^{\zeta_3}}, g). \frac{T_6}{e(T_5, t_M)} \stackrel{?}{=} e(A_{ij}, g)$. The algorithm outputs i if it exists, otherwise outputs $\perp$.

Direct tracing $\mathsf{DTrace}(gpk, dok, M, \sigma)$: By accessing the registration table $reg_2[i]$, this algorithm can revoke the signer's identity i of a valid signature σ on message M. Note that unlike **Attested tracing** ($\mathsf{Open}(gpk, ok, M, \sigma, t_M)$), this algorithm use no *token* in order to trace the identity of the signer. It extracts the part of the member group secret key $e(T_7/T_1^{\ell}, g) \overset{?}{=} e(g,g)^{x_{ij}}$ and match the record in the $reg_2[i]$.

Managing Linkability $\mathsf{SignLink}((\sigma, M), (\sigma', M'), lk)$: Given two message (M, M') and their corresponding signatures (σ, σ'), and linking key $lk \leftarrow h^{\ell}$, this algorithm tries to find links among signatures whether they are generated from the same signer i. It first verifies the signatures' validity by using the algorithm $\mathsf{GVerify}$. Then it checks $e(T_7/T_7', h) \overset{?}{=} e(T_1/T_1', lk)$. It returns 1 if successful, otherwise outputs $\perp$. We assume that x_i is picked uniformly at random so that $x_i \neq x_j$ for any i, j.

Revocation $\mathsf{Revoke}(gpk, gsk_i, A'', w'')$: Revocation would be accomplished in two ways:

- Verifier-Local Revocation (VLR): Adopting 0/1 *encoding system* enables the group manager (Issuer) to embed the key expiration date in each signing key. It ensures that the signature will pass the verification algorithm $\mathsf{GVerify}(gpk, \bar{t}, M, \sigma)$ only if the *key expiration* date is larger than the *signature expiration* date. Although proposed scheme is not completely satisfying the requirement of traditional VLR scheme where verifier holds a list of special information called Revocation List (RL) for each revoked signer. But it partially helps the verifier to revoke the expired signers (vehicles) locally.

- Re-keying the signature scheme: In Re-key based revocation solution, the issuer updates its public key *gpk*, and hence, the execution of signing and verification algorithms are affected subsequently. At each update of the key, a former signer would become no longer a legitimate signer unless it updates its credentials it holds.

  The Re-key revocation process is done in a fixed time interval. The advantage of this mechanism is that each signer knows when the rekey process will take place. The drawback is that no legitimate signer will be revoked within this interval. Note that, this interval could be flexible, that is, rekeying will happen when the group shrinks with some members leaving. But the later choice is opposite to the former one and also inefficient. The length of the interval is then dependent on applications.

  During $\mathsf{GKgen}(1^{\lambda}, 1^{l}, 1^{n})$ algorithm execution, Issuer generates the credential $gsk_i \leftarrow (A, x, d)$ for each signer. To update group public key gpk and credential gsk_i for each currently legitimate signer i, the issuer first choose its private key by deriving a new value $\gamma'' \in \mathbb{Z}_p$. For each currently legitimate signer, the issuer updates the credential element A with

$$A'' \leftarrow g^{1/(\gamma'' d_j + x_j)_{j \in [1,l]}}$$

The issuer makes A'' available to corresponding signer (new credential $gsk_i \leftarrow (A'', x, d)$) and publishes $w'' \leftarrow g^{\gamma''}$ to replace w in its public key gpk. The signer may optionally check whether the new gsk_i is associated to the gpk by

$$e(A'', w^d g^x) \overset{?}{=} e(g, g)$$

Theorem 1 *Our group signature scheme is correct.*

Theorem 2 *If the decisional Diffie-Hellman assumption holds in $\mathbb{G}$, our construction with time-bound keys has anonymity in the random oracle model.*

Theorem 3 *If the discrete logarithm assumption holds, our construction has linkability in the random oracle model.*

Theorem 4 *If the decision bilinear Diffie-Hellman assumption holds, our construction has attested opener anonymity in the random oracle model.*

Theorem 5 *If the decision linear assumption holds, our construction has admitter anonymity in the random oracle model.*

Theorem 6 *If the q-strong Diffie-Hellman assumption holds, our construction has traceability in the random oracle model.*

4.1.6 Security and Performance comparison

We compare our GS scheme based on BBS GS [59] with the other related VANET GS proposals such as Hwang et al.[64], Qin et al. [1], Mamun et al.[50], Zhang et al.[65], Malina et al.[70], Zhang et al.[8]. Table I. shows a comparative study on the aforementioned schemes.

Table 4.1: Comparison with related VANET schemes

	Ours	Hwang et al.[64]	Qin et al. [1]	Mamun et al.[50]	Zhang et al.[65]	Malina et al.[70]	Zhang et al.[8]
Signature length	381 B	161 B	845 B	542 B	241 B	281 B	302 B
Anonymity	CCA	CPA	CCA	CCA	CPA	CPA	CPA
Linkability	Yes	Yes	No	No	No	Yes	No
Attested Traceability	Yes	No	No	No	No	No	No
Revocability	Hybrid	Rekeying	No	No	VLR	VLR	Rekeying
Batch verification	Yes	No	No	Yes	No	Yes	Yes
Signature Verification (pairing + exponent)	4+14n	1n+4n	11n + 19n	3+16n	2n+17n (+RList)	2+11n (+RList)	2+ 14n

We provide construction for more stringent security notions (CCA anonymity). In compare to the GS scheme [53], we introduce only one additional element (in $\mathbb{G}_1$) in the basic signature to satisfy two additional properties (linkability, direct opening). Moreover, unlike other proposals, we refer hybrid revocation (limited VLR + Rekeying) system. Our VLR solution works only with signer's expiration date (constant verification cost). It rules out expensive revocation check (checking revoked member list) for each signature verification. It is worth mentioning that the verification cost in [70, 65] (as authors claimed) does not reflect the literal cost. It actually depends on the size of revoked member list (RList).

For signature length, we consider the MNT curve with $\mathbb{G}_1 = 161$ bits, $\mathbb{G}_T = 483$ bits and $\mathbb{Z}_p = 160$ bits. In general, bilinear pairing T_p is the most expensive operation ($10 \times$ exponentiation operation T_e) while one point multiplication T_m is the least. Our proposal achieves the maximum functionality of the GS with optimum cost (signature length and verification). Our efficient batch verification cost includes $(4T_p + 14nT_e)$ for n signatures. Note that, we need to increase the signature size by 6 elements ($3\mathbb{G}_1$, $3\mathbb{G}_T$) for batch verification.

We implement our scheme on an Intel Core i3 model CPU @2.43 GHz using the PBC library [42] running on top of Gnu GMP [41] on Ubuntu 12.10. They use a supersingular curve (order is a Solinas prime). The processing time for one bi-linear operation T_p, a single exponentiation T_e are respectively 3.1 *ms* and .3 *ms.* The verification of a single signature takes approximately 21 *ms* (considering some pre-computation like $e(g, g), e(g, w), e(f, g)$ etc.) that is very close to Mamun *et al.* scheme (19 *ms*) in [50]. However, for batch verification, it will be much more efficient on average.

4.1.7 Conclusion

In this work, we have presented a CCA-secure short group signature solution considering hybrid revocability, linkability and message-depend opening for an application-friendly VANET environment. We focus on relaxed privacy that can be efficiently used for a hierarchical VANET architecture.

4.2 An efficient batch verification system for VANET

4.2.1 Introduction

Intelligent Vehicle Technologies comprise electromechanical and electromagnetic components operating in conjunction with computer controlled devices and radio transceivers. These *intelligent* vehicles with other units self-organize a variant of MANET called VANET. It commonly applies to car safety systems and commercial application for communication. Security objectives and solutions in VANET are different from common PC-based environment. For example, embedded low-cost computing platform in vehicles are unlikely to form complex cryptographic primitives and protocols like traditional PC with respect to processors and memory; bandwidth for external communication is limited; attackers might include the owner or third parties that have physical access to the vehicles e.g., motor mechanics, valets etc. Therefore, VANET becomes an emerging research area, both in industry and in academia.

To differentiate between trusted and untrusted vehicles, messages should be usually bound to certificates belonging to vehicles. A certificate consists of vehicle's public key, identifier and additional data such as supported algorithm, lifespan etc., which are signed by the Certificate Authority (CA). It invalidates a certificate, called *revocation*, to resist an untrusted vehicle from the VANET.

In a large scale VANET, e.g., densely populated downtown area, pseudonym-based schemes deal with the challenges of generating, distributing, verifying, storing and revoking a large number of certificates while group signature-based schemes addresses problems such as managing huge number of vehicles including compromised vehicles. Interestingly, both schemes have a common concern as to how to verify the large volume of messages received in real time. In order to gear up the process of verification, a batch verification must be an alternative solution since it cuts down the verification delay, particularly when verifying large amount of signatures received in a time window rather than verify the signatures sequentially one after another.

To provide identity authentication and message integrity, one of the promising solutions is providing a digital signature before the message is sent. Usually two Public Key Infrastructures (PKI) architecture are commonly used for VANET. One is RSA-based PKI and other is Elliptic Curve Cryptosystem (ECC) based PKI. It is generally agreed that ECC-based anonymity is better than that of RSA as ECC has a smaller key size and lower computation time[14]. In order to ensure message integrity, authenticity with anonymity, one of the attractive solutions is to use identity based group signature (IBGS) for message passing between vehicles. But most of the IDGSs have ID-based key pairs for group members only, while other entities such as group manager, opening authority are not ID-based. Nonetheless, ID-based group managers and Opening authorities are very useful for a real life VANET environment where a vehicle is usually a member of many groups at a time. Therefore, we select a fully identity based dynamic group signature scheme described in [2].

Meanwhile, conventional signature verification mechanisms are not sufficient enough to satisfy the stringent time requirement in VANET. For example, in a VANET environment where hundreds of vehicles send messages within a period of 100-300 mili-second (ms) travel time, a receiver vehicle needs to verify hundreds of message signatures per ms which is obviously a tough requirement for any current digital signaturing system

[15]. Usually vehicles in VANET communicate by means of Dedicated Short Range Communication (DSRC) standard that employs the IEEE 802.11p standard [15] for wireless communication. Although only one signature is received by DSRC transmission at a time, a large number of signatures are buffered at the receiving station. According to [16], consumption period of a DSRC transmission is shorter than that of verifying a signature. For this reason, verifying a huge number of message signatures one by one is impractical and may cause bottlenecks at Temper Proof Device (TPD) in the vehicle. As a consequence, many messages coming from other vehicles may be discarded due to time constraints or improper scheduling. Nevertheless, some signatures might arrive from emergency vehicles e.g., SOS messages from ambulance, messages from vehicles sensing life-threatening after airbags are deployed etc. that incur urgent responses (short due time) from the receivers.

Besides that, it is generally believed that certificates are twice as large as signatures. Therefore, a big challenge arising from the PKI-based schemes in VANET is the heavy burden of certificate generation, storage, delivery, verification, and revocation.

In this work, we consider the fully ID-based group signature scheme described in [2] and the *selfish verification* system described in [1] that was designed for a large scale VANET, where the authors choose the group signature scheme described in [2] and propose batch verification mechanism to speed up signature processing. We carefully observe that the batch verification system proposed in [1] is not consistent according to the standard form of batch verification and the extended part of the signature they proposed for batch verification include some redundant parts. Therefore, we first propose to reduce the extended part of the signature[1] and then improve the *batch verification* system described in [1], and hence the total amount of computations for each individual signature and batch verification. In addition, we devise an algorithm to determine the maximum number of signatures to batch at a time depending on the signature scheme used for a certain VANET environment. This algorithm could be used off-line with the available traffic data of any specific area. Moreover, we analyze the impact of *the number of signatures* in batch verification from the view point of DSRC standard. Furthermore, we devise a *signature scheduling algorithm* to prioritize certain type of messages where batch verification is not pragmatic.

4.2.2 Preliminaries

VANET architecture

Recently, some security hardwares have been introduced to VANET which make it feasible to implement robust cryptographic tools, for instance, Tamper Proof Devices (TPD) that provide the ability of processing, signing and verifying messages, protect hardware from tampering by a set of sensors, Event Data Recorder (EDR) that records all received data s.t., position data, speed data, acceleration data, time etc. [11].

A VANET involves two types of communication devices: On-Board Unit (OBU) and Road-side Unit (RSU) while OBUs are installed in the vehicle and RSUs (stationary devices) are mounted on roadside. Manufacturers along with telecommunication industries encourage each car to equip with an OBU. OBUs together with RSUs help to broadcast messages to other vehicles or transport system terminals in their range. Likewise RSU, the

[1] Note that we do not modify the original signature described in [2], but the extended part for batch verification accounted in [1].

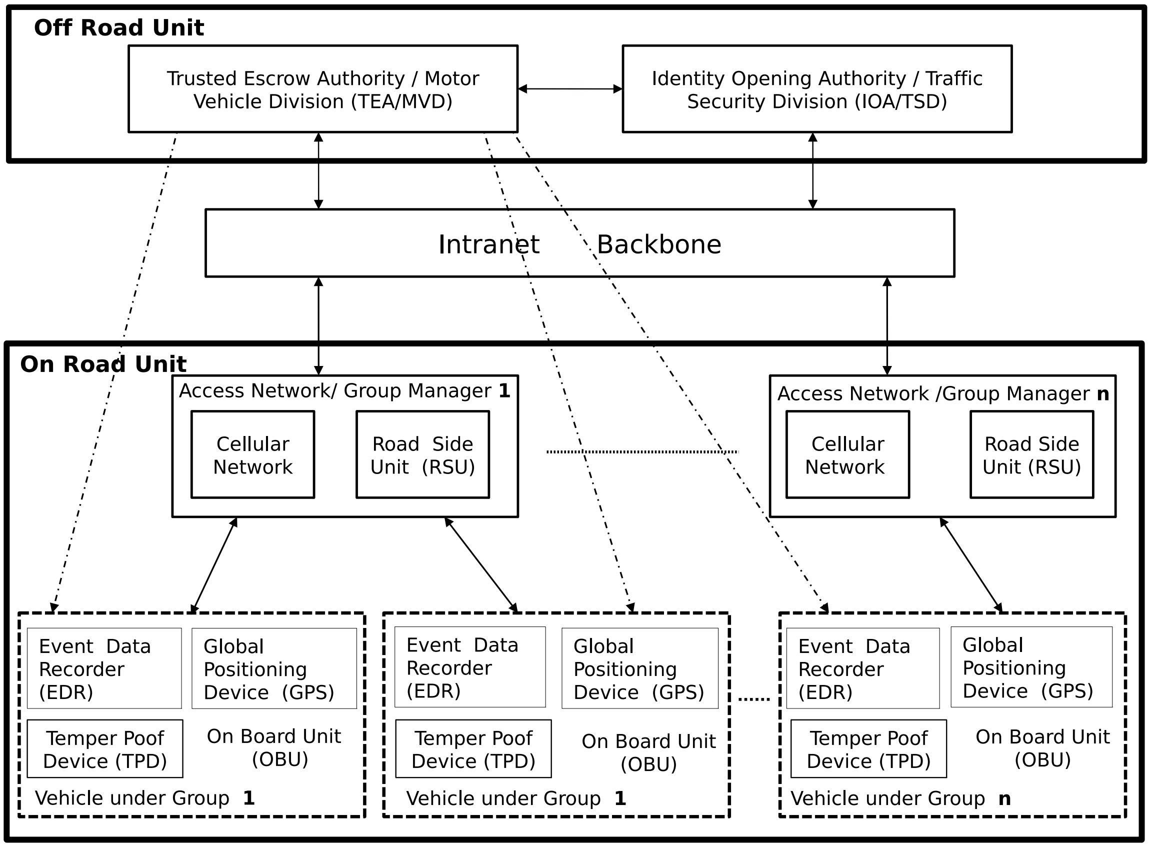
Off Road Unit
Trusted Escrow Authority / Motor Vehicle Division (TEA/MVD)
Identity Opening Authority / Traffic Security Division (IOA/TSD)
Intranet Backbone
On Road Unit
Access Network/ Group Manager 1
Cellular Network
Road Side Unit (RSU)
Access Network /Group Manager n
Cellular Network
Road Side Unit (RSU)
Event Data Recorder (EDR)
Global Positioning Device (GPS)
Temper Poof Device (TPD)
On Board Unit (OBU)
Vehicle under Group 1
Event Data Recorder (EDR)
Global Positioning Device (GPS)
Temper Poof Device (TPD)
On Board Unit (OBU)
Vehicle under Group 1
Event Data Recorder (EDR)
Global Positioning Device (GPS)
Temper Poof Device (TPD)
On Board Unit (OBU)
Vehicle under Group n

Figure 4.2: VANET Network Model

computing device in OBU is used to verify signatures or other computations in response to the messages received from other communicating vehicles [1, 6]. Nonetheless, traditional VANET possesses two kinds of communication policy, namely, Vehicle to Vehicle (V2V) and Vehicle to RSU (V2R) [26].

Figure 4.2. briefly describes a typical VANET architecture that consists of an On-Road Unit (OBU, RSU) and an Off-Road Unit (TEA,TSD). There is a trusted authority, called Trusted Escrow Authority (TEA), registers the operating entities like vehicles, RSUs etc. Vehicles are attached, accessed and managed by Access Network (RSU) and subsequently RSUs are attached to TEA and Traffic Security Division (TSD). Note that TSD can disclose a vehicle's identification and hence can break anonymity if necessary.

Groups in VANET can be formed in many ways. For example, by **region**: New York city, Tokyo city etc., by **social spot**: shopping mall, official zone, military zone, educational institutions etc., by **category**: personal cars, ambulance, police cars, fire trucks etc. Forming the groups can help in applying policy to manage the vehicles intelligently.

Why fully ID-based group signature?

It is a general curiosity: why identity based group signature would be the best choice to VANET environments? Firstly, we explain about the essence of group signature. The major problem associated with traditional digital signature schemes is to ensure privacy, the vehicles would have to store a very large number of public/private key pairs, and keys must be changed often. Secure distribution of keys, key management, and storage are very difficult in this type of scheme. In contrast, a group signature scheme provides user anonymity of the members by signing the messages on behalf of the group. In addition, Group Signatures can be verified with respect to a single group public key, but they do not reveal the identity of the signer. Furthermore, it is not possible to decide whether two signatures have been issued by the same group member, which effectively prevents a user from being tracked. Compared to the traditional certificate based scheme, ID based group signatures achieves authenticity, data integrity, anonymity, traceability and accountability at the same time.

Secondly, we focus on requirement of ID based group signature. In a public Key Infrastructure (PKI), sender must have recipient's certificate that includes complexity of certificate management and Certificate Revocation Lists (CRLs). Alternatively, Identity based schemes use identities such as Electronic license plate of a car as public key. In a VANET infrastructure, it is very helpful. The sending vehicle only needs to know the recipient car's identity attribute which can be traced seamlessly via wireless interface of the vehicular network. Therefore, public key of the entity can be traced immediately in compare to the burden of downloading certificate databases from the certification authority (CA) in a traditional Group signature scheme.

Finally, we concentrate on the necessity of fully ID based group signature. The group signature scheme used in the work is *fully* ID based where not only group members are identity-based, but also group manager and opening authority follow the same manner. As signer's identity is intractable to the verifier, there is no impact to the verifier whether the member is identity based or not [2]. In a VANET environment, where vehicles need to be connected to several group managers or open authorities at the same time, it is constructive to exploit a fully identity-based group signature scheme. However, the identities of the entities can be uniquely revealed by the trusted open authority called TSD under certain circumstances or policy.

Vehicle (x_V)		**GM** (C, x_R)
		Run Proof of Knowledge for x_V[34] Compute $D = A_5/H_V(ID_V)^{1/(t+x_R)}$ $t \in \mathbb{Z}_p^*$ Compute $W = e(H_V(ID_V), B)$ Record (ID_V, D, t, W) for future use.
	$\xleftarrow{(D,t,C)}$	
Check validity of C [33] and accept if $e(A_5, B) = e(D, B)^t e(D, S) e(H_V(ID_V), B)$ holds for $S = {CK_T}^{H_R(C\|\|ID_R)}$		

Figure 4.3: Group Joining Protocol

4.2.3 Identity based Group Signature

A fully ID based dynamic group signature scheme [2], where there are group managers (GMs), group members, and the Open Authorities (OAs), provides full traceability, full anonymity and non-frameability as needed for VANET application. We summarize [2] as follows:

Setup(1^l)**:** Let a security parameter (1^l) , p is a prime and finite cyclic groups $\mathbb{G}_1 = \langle A \rangle$ and $\mathbb{G}_2 = \langle B \rangle$, where there exist a computable isomorphism $\Psi = \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and a non-degenerate bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$.

For initial setup **TEA** will do the followings:

1. Set $\Re = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, A, B, e)$
2. Choose $A_1, A_2, A_3, A_4, A_5 \in \mathbb{G}_1$
3. Define cryptographic function
   - $H_V : \{0,1\}^* \rightarrow \mathbb{G}_1$ for Vehicles
   - $H_O : \{0,1\}^* \rightarrow \mathbb{G}_1$ for TSD
   - $H_R : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ for GM
   - $H : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ for vehicles to compute or, to verify challenges.
4. Generate Private Key $(x_T) \in \mathbb{Z}_p^*$ and Public key $(K_T) = B^{x_T} \in \mathbb{G}_2$
5. Finally, produce the system's Public Parameter:
   $param = (\Re, A_1, A_2, A_3, A_4, A_5, H_V, H_O, H_R, H, K_T)$

Key Generation: Since the signature is fully ID based, it needs to generate private keys not only for the member vehicles but also for the GMs and OAs (TSDs). Therefore, TEA generates private keys for all the entities in the system including vehicles, GMs, and TSDs with their identities. Let the identity of the vehicles, GMs and OAs (TSD) are ID_V, ID_R, and ID_O respectively.

- With the identity ID_R, TEA generates private key for the GM $x_R = r + H_R(C||ID_R)x_T \mod p$ where $r \in \mathbb{Z}_p^*$ and $C = B^r$. Then TEA issues (C, x_R) to each GM, where C is used by the GM to register vehicles.
- With the identity ID_O, TEA generates private key x_O for OA $x_O = H_O(ID_O)^{x_T}$ and issues x_O to TSD for exposing the identity of any vehicles on demand.
- TEA issues private key x_V for the vehicles: $x_V = H_V(ID_V)^{x_T}$ where ID_V is the identity of a vehicle.

Group Join Protocol: Vehicles need to complete registration with their corresponding GM. First GM runs a proof of knowledge on a vehicular private key x_V for its Identification without any information leakage. Additionally GM sends C as it got from TEA and manages a registration table or database for all the member vehicles.

Before joining the group, a vehicle must have its ID ID_V and secret key x_V generated from TEA. To get a membership certificate a vehicle needs to perform a protocol with the certificate issuer called GM as shown in Fig. 4.2.2 At the end of the protocol, a vehicle becomes a member of the group and obtains a membership certificate (D, t, C) as a *Group Signing Key.* GM computes W which is stored in the database for future use.

Signing and Authentication: A registered vehicle under a group having a secret key x_V and *Group Signing Key*(D, t, C) can anonymously generate a signature Υ on a message M. At the same time, it allows TSD, or Open authority to open the signature if needed. The signature is based on a *proof of knowledge*(SPK)

$$\begin{aligned} \{SPK(ID_V, x_V, (D, t), d) : x_V &= H_V(ID_V)^{x_T} \\ \wedge \quad D^{t+x_R} H_V(ID_V) &= A_5 \quad \wedge \quad V_2 = B^d \\ \wedge \quad v_1 = (e(H_V(ID_V), B) e(H_O(ID_O), K_T)^d\} \;\; (M) \end{aligned}$$

which means a group signature of message M by a signer vehicle V who knows the secret values $(ID_V, x_V, (D, t), d)$ satisfying several relations $R(r_1, \wedge \cdots \wedge r_4)$. Detailed descriptions are as follows:

- Choose $s_1 \in \mathbb{Z}_p^*$ and set $(\Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_5, s_2) = (A^{s_1}, x_V A_1^{s_1}, H_V(ID_V) A_2^{s_1}, D A_3^{s_1}, \Gamma_3^t A_4^{s_1}, ts_1 \bmod p)$
- Choose $d \in \mathbb{Z}_p^*$ and set $(v_1, V_2) =$ $\Big(e(H_V(ID_V), B) e(H_O(ID_O), K_T)^d,\ B^d\Big)$
- Select randomly $(r_1, r_2, r_3, r_4) \in \mathbb{Z}_p^*$ and $(R_1, R_2, R_3) \in \mathbb{G}_1$ and compute:
  - $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_5, \beta_7) =$ $\Big(A^{r_1}, R_1 A_1^{r_1}, R_2 A_2^{r_1}, R_3 A_3^{r_1}, \Gamma_3^{r^3} A_4^{r_1}, B^{r_4}\Big)$
  - $(\beta_4, \beta_6, \beta_8) = \Big([e(A_1, B)^{-1} e(A_2, K_T)]^{r_1}, e(A_3, B)^{r_2}$ $[e(A_3, S) e(A_2 A_4, B)]^{r_1}, e(H_O(ID_O), K_T)^{r_4}$ $e(A_2, B)^{-r_1}\Big)$
- Compute the challenge: $f = H\Big((\Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_5) || C || v_1 || V_2 || M ||$ $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8)\Big)$
- Compute:
  - $(z_0, z_1, z_2, z_3) = (r_1 - f s_1 \bmod p,$ $r_3 - ft \bmod p, r_2 - f s_2 \bmod p, r_4 - fd \bmod p)$
  - $(Z_1, Z_2, Z_3) = (R_1 x_V^{-f}, R_2 H_V(ID_V)^{-f}, R_3 D^{-f})$
- Signature for batch verification: $\Upsilon = \Big(\Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_5) || (z_0, z_1, z_2, z_3, Z_1, Z_2, Z_3)$ $|| f || C || v_1 || V_2\Big)$

In [1], authors propose the following signature for batch verification. They extend the original signature in [2] with some other parameters to verify in batch at the receiving vehicle.

$$\Upsilon' = \Upsilon \mid\mid (\Gamma_4, \Gamma_6, \Gamma_8) \mid\mid (\beta_0, \beta_1, \beta_2, \beta_3, \beta_5, \beta_7) \mid\mid \\ (\beta_4, \beta_6, \beta_8) \mid\mid S$$

Individual Signature Verification: After getting a signature Υ, a vehicle verifies the signature as follows:

- Compute $(\Gamma_4,\Gamma_6,\Gamma_8) =$ $(e(\Gamma_1, B)^{-1}e(\Gamma_2, K_T),\ e(A_5, B)^{-1}$ $e(\Gamma_3, S)e(\Gamma_2\Gamma_5, B),\ V_1e(\Gamma_2, B)^{-1})$
- Compute $S = C{K_T}^{H_R(C||ID_R)}$
- Compute $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_5, \beta_7) =$ $\Big(A^{z_0}\Gamma_0^f, Z_1A_1^{z_0}\Gamma_1^f, Z_2A_2^{z_0}\Gamma_2^f, Z_3A_3^{z_0}\Gamma_3^f,$ $\Gamma_3^{z_4}A_4^{z_0}\Gamma_5^f, B^{z_6}V_2^f\Big)$
- Compute$(\beta_4, \beta_6, \beta_8) =$ $\Big([e(A_1, B)^{-1}e(A_2, K_T)]^{z_0}\Gamma_4^f,\ \ e(A_3, B)^{z_5}$ $[e(A_3, S)e(A_2A_4, B)]^{z_0}\Gamma_6^f, e(H_O(ID_O), K_T)^{z_6}$ $e(A_2, B)^{-z_0}\Gamma_8^f\Big)$
- Compute $f' = H\Big((\Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_5)||C||v_1||V_2$ $||M||(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8)\Big)$

After computing f', it compares to the challenge f received in the signature. If they are equal, signature is valid and the message will be accepted, otherwise rejected.

4.2.4 The Proposal

We start with the IBGS scheme in [2] as described in section 3. Any extended parts of the modified signature Υ' will not affect any security properties of the group signature, because it is clearly seen that the extended parts $(\Gamma_4, \Gamma_6, \Gamma_8) \mid\mid (\beta_0, \beta_1, \beta_2, \beta_3, \beta_5, \beta_7) \mid\mid (\beta_4, \beta_6, \beta_8) \mid\mid S$ can be easily reconstructed from Υ [1]. However, we carefully observe that the extended signature Υ' described in [1] includes some redundant parts.[2] For instance $(\Gamma_4, \Gamma_6, \Gamma_8) \mid\mid (\beta_0, \beta_1, \beta_2, \beta_3, \beta_5, \beta_7) \mid\mid S$ of Υ'. Since the target of batch verification is to verify the elementary parameters of several signatures together to accelerate the total verification time and $(\beta_4, \beta_6, \beta_8)$ is the most complex part of the signature Υ, verifying it covers verification of almost all the elementary parameters of Υ. In addition, we need to consider the communication overhead of the protocol. Although batch verification enhances the performance of verifying signatures in the receiver vehicle, the extended size of signatures (for batch verification) increases the communication overhead

[2] Size of the signature in [1] is larger than [2].

dramatically. However, in a VANET environment where thousands of secure messages be transferred among vehicles at any time instance, large signature size may degrade the overall network performance. As a consequence, we propose to trim the redundant part of Υ'. It reduces the communication overhead without any compromise to the security of the basic signature scheme[3]. Then we minimize the pairing calculation in $(\beta_4, \beta_6, \beta_8)$ for more efficiency.

Modified individual Signature Verification: A signer who possesses the *group signing key* can anonymously generate a signature on a message M. Our modified trimmed signature will be:

$\Upsilon' = \Upsilon || (\beta_4, \beta_6, \beta_8)$, **Or**
$\Upsilon' = (\Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_5) || (z_0, z_1, z_2, z_3, Z_1, Z_2, Z_3) ||$
$(\beta_4, \beta_6, \beta_8) || f || C || v_1 || V_2$

As mentioned before, $(\beta_4, \beta_6, \beta_8)$ is the most expensive part of the verification. In section **3**, we gave the verification mechanism of [1] which is modified here according to the above trimmed signature Υ' to reduce the cost of individual verification. To verify signature Υ' of a message M, perform the following:

- Compute $S = C{K_T}^{H_R(C||ID_R)}$
- Compute $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_5, \beta_7) =$
  $(A^{z_0}\Gamma_0^f, Z_1 A_1^{z_0}\Gamma_1^f, Z_2 A_2^{z_0}\Gamma_2^f, Z_3 A_3^{z_0}\Gamma_3^f, \Gamma_3^{z_4} A_4^{z_0}\Gamma_5^f, B^{z_6} V_2^f)$
- Check $f = H\Big((\Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_5) || C || v_1 || V_2 || M ||$
  $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8)\Big)$
- Verify $(\beta_4, \beta_6, \beta_8) =$
  $\Big([e(A_1, B)^{-1} e(A_2, K_T)]^{z_0} [e(\Gamma_1, B)^{-1} e(\Gamma_2, K_T)]^f,$
  $e(A_3, B)^{z_2} [e(A_3, S) e(A_2 A_4, B)]^{z_0} [e(A_5, B)^{-1}$
  $e(\Gamma_3, S) e(\Gamma_2 \Gamma_5, B)]^f, e(H_O(ID_O), K_T)^{z_6}$
  $e(A_2, B)^{-z_0} [v_1 e(\Gamma_2, B)^{-1}]^f\Big)$

Notice that, in contrast to the verification mechanism in [1], $(\Gamma_4, \Gamma_6, \Gamma_8)$, S, and $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_5, \beta_7)$ are not included to the signature, and the definition of $(\beta_4, \beta_6, \beta_8)$ is redefined.

Modified Batch Verification: The scheme in [1] exploits the techniques of [30] keeping in mind that a multi-base exponentiation (pairing) takes a similar time as a single-base exponentiation. Though authors mention the use of *Small Exponent Test* in the work, the equation for batch verification do not include any parameter for small exponent test. We followed the standard procedures for batch verification by suing the techniques described

[3] As we have trimmed the extended part of the signature, it has no affect on the security requirement of the signature.

in Section 2.3. Let a VANET device receive n message-signature pair $(M_i, {\Upsilon'}_i)$ where ${\Upsilon_i}' = (\Gamma_{0,i}, \Gamma_{1,i}, \Gamma_{2,i}, \Gamma_{3,i}, \Gamma_{5,i})||(\beta_{4,i}, \beta_{6,i}, \beta_{8,i})||f_i||C_i||v_{1,i}||V_{2,i}||$
$(z_{0,i}, z_{1,i}, z_{2,i}, z_{3,i}, Z_{1,i}, Z_{2,i}, Z_{3,i})$.

- Compute $S = C{K_T}^{H_R(C||ID_R)}$ once, because all vehicles share the parameters C, ID_R, K_T [*Technique 1*, in section 2.]

- For all $i = 1, ..., n$ compute the non-pairing equations:$(\beta_{0,i}, \beta_{1,i}, \beta_{2,i}, \beta_{3,i}, \beta_{5,i}, \beta_{7,i}) =$
$(A^{z_0}\Gamma_{0,i}^{f_i}, Z_{1,i}A_1^{z_{0,i}}\Gamma_{1,i}^{f_i}, Z_{2,i}A_2^{z_{0,i}}\Gamma_{2,i}^{f_i}, Z_{3,i}A_3^{z_{0,i}}$
$\Gamma_{3,i}^{f_i}, \Gamma_{3,i}^{z_{4,i}}A_4^{z_{0,i}}\Gamma_{5,i}^{f_i}, B^{z_{6,i}}V_2^{f_i})$

- For each $i = 1, ..., n$ check the following:
$f_i = H\Big((\Gamma_{0,i}, \Gamma_{1,i}, \Gamma_{2,i}, \Gamma_{3,i}, \Gamma_{5,i})||C_i||v_{1,i}||V_{2,i}$
$||M_i||(\beta_{0,i}, \beta_{1,i}, \beta_{2,i}, \beta_{3,i}, \beta_{5,i}, \beta_{7,i})\Big)$

- Before starting a batch verification, we can simplify the equation as follows:

$$\begin{aligned}\beta_4 &= [e(A_1, B)^{-1}e(A_2, K_T)]^{z_0}[e(\Gamma_1, B)^{-1}e(\Gamma_2, K_T)]^f \\ &= e({A_1}^{-z_0}, B)e({A_2}^{z_0}, K_T)e({\Gamma_1}^{-f}, B)e({\Gamma_2}^f, K_T) \\ &= e({A_1}^{-z_0}\Gamma_1^{-f}, B) \;\; e({A_2}^{z_0}{\Gamma_2}^f, K_T)\end{aligned}$$

$$\begin{aligned}\beta_6 = e(A_3, B)^{z_5}&[e(A_3, S)e(A_2A_4, B)]^{z_0}[e(A_5, B)^{-1} \\ &e(\Gamma_3, S)e(\Gamma_2\Gamma_5, B)]^f \\ = e&({A_3}^{z_5}, B)e({A_3}^{z_0}, S)e({A_2}^{z_0}{A_4}^{z_0}, B) \\ &e({A_5}^{-f}, B)e({\Gamma_3}^f, S)e({\Gamma_2}^f{\Gamma_5}^f, B) \\ = e&({A_3}^{z_5}{A_2}^{z_0}{A_4}^{z_0} \;\; {A_5}^{-f}{\Gamma_2}^f{\Gamma_5}^f, B) \\ &e({A_3}^{z_0}{\Gamma_3}^f, S)\end{aligned}$$

$$\begin{aligned}\beta_8 = e(H_O(ID_O), K_T)^{z_6} \;\; &e(A_2, B)^{-z_0} \\ &[v_1 \;\; e(\Gamma_2, B)^{-1}]^f \\ = e&([H_O(ID_O)]^{z_6}, K_T) \;\; e({A_2}^{-z_0}, B)({v_1}^f) \\ &e({\Gamma_2}^{-f}, B) \\ = (&{v_1}^f) \;\; e({A_2}^{-z_0}{\Gamma_2}^{-f}, B) \;\; e([H_O(ID_O)]^{z_6}, K_T)\end{aligned}$$

- Let $\xi_b = {A_1}^{-z_0}\Gamma_1^{-f}$,
$\xi_k = {A_2}^{z_0}{\Gamma_2}^f$,
$\zeta_b = {A_3}^{z_5}{A_2}^{z_0}{A_4}^{z_0}{A_5}^{-f}{\Gamma_2}^f{\Gamma_5}^f$,
$\zeta_s = {A_3}^{z_0}{\Gamma_3}^f$,
$\chi_b = {A_2}^{-z_0}{\Gamma_2}^{-f}$,
$\chi_k = [H_O(ID_O)]^{z_6}$

- Hence $(\beta_4\beta_6\beta_8)$
$= (e(\xi_b, B)e(\xi_k, K_T)e(\zeta_b, B)e(\zeta_s, S)e(\chi_b, B)$
$e(\chi_k, K_T)({v_1}^f))$
$= (e(\xi_b\zeta_b\chi_b, B) \;\; e(\xi_k\chi_k, K_T) \;\; e(\zeta_s, S)({v_1}^f))$

- Applying *Technique 1,3* in section 2., choose the random vector $(\delta_1, ..., \delta_l)$ where $\delta_i \in \mathbb{Z}_p$; and check the following pairing equations $\forall i = 1, ..., n$:

$$\prod_{i=1}^{n}(\beta_{4,i}\beta_{6,i}\beta_{8,i})^{\delta_i} =$$
$$e(\textstyle\prod_{i=1}^{n}\delta_i\xi_{b,i}\zeta_{b,i}\chi_{b,i}, B)\ \ e(\prod_{i=1}^{n}\delta_i\xi_{k,i}\chi_{k,i}, K_T)$$
$$e(\textstyle\prod_{i=1}^{n}\delta_i\zeta_{s,i}, S)\ \ (\prod_{i=1}^{n}\delta_i {v_{1,i}}^{f_i})$$

Again for simplicity let:
$\mathbf{M_i} = (\beta_{4,i}\beta_{6,i}\beta_{8,i})^{\delta_i}$, $\mathbf{B_i} = \delta_i\xi_{b,i}\zeta_{b,i}\chi_{b,i}$,
$\mathbf{Q_i} = \delta_i\zeta_{s,i}$, $\mathbf{K_i} = \delta_i\xi_{k,i}\chi_{k,i}$ and $\nu_{\mathbf{i}} = \delta_i {v_{1,i}}^{f_i}$

Hence, $\prod_{i=1}^{n}\mathbf{M_i} =$
$$e(\textstyle\prod_{i=1}^{n}\mathbf{B_i}, B)\ \ e(\prod_{i=1}^{n}\mathbf{K_i}, K_T)$$
$$e(\textstyle\prod_{i=1}^{n}\mathbf{Q_i}, S)\prod_{i=1}^{n}\nu_{\mathbf{i}}$$

If the above equation is satisfied then verification is successful; otherwise not.

Algorithm 1 Batch size

```
INPUT: Due time $d_i$, Priority queues $\mathbf{M}, \mathbf{B}, \mathbf{K}, \mathbf{Q}, \mathbf{V}$
OUTPUT: (Batch size $n$ , Total Completion time $t_n$ $(ms)$, Max. Lateness $L_n$
• FOR ($n = 2$ to $\infty$)
    1. Initial TS := $t_1$
    2. Pop $i^{th}$ value from $\mathbf{M}, \mathbf{B}, \mathbf{K}, \mathbf{Q}, \mathbf{V}$ and
    3. Calculate $\eta = e(\prod_{i=1}^{n}\mathbf{B_i}, B)\ e(\prod_{i=1}^{n}\mathbf{K_i}, K_T)$
       $e(\prod_{i=1}^{n}\mathbf{Q_i}, S)\prod_{i=t}^{n}\nu_{\mathbf{i}}$
    4. Calculate $\mu = \prod_{i=1}^{n}\mathbf{M_i}$
    5. Check $\mu = \eta$. IF (successful) then:
        – Current TS:= $t_2$
        – Max. Completion time $C_{max_n} = t_2$
        – IF ($C_{max_n} > d_b$) return;
        – Calculate $L_n := C_{max_n} - d_1$
        – Completion time $t_n = t_2 - t_1$
        – Record $(n,\ t_n, L_n)$.
    6. Else if $\mu \neq \eta$, report batch error and exit.
    7. Continue until Queues: $\mathbf{M}$, $\mathbf{B}$, $\mathbf{K}$, $\mathbf{Q}$, $\mathbf{V}$ become empty.
```

Batch size: Is batch verification of any group signature always provide efficient solution to the application? Unfortunately, not always! The first point to note is: batch verification starts on the availability of certain number of signatures called *batch size*(n). For instance, at an uptown area or rural area where there is no rush of vehicles, or at a fast express highway with sporadic traffic; using batch verification might yield the efficiency even worsened. Because, the receiving vehicles need to wait for the minimum number of signatures n to commence verification.

Secondly, in a VANET environment, signatures arrive sequentially with respect to time like a stream. OBU usually processes the signature on a First Come First Serve (FCFS) basis. But we cannot allow FCFS always, because sometimes prioritized emergency messages e.g., SOS services like fire service, ambulance vehicle message signatures etc. might arrive with a tight *deadline* to finish. So it is more practical to process the signatures according to their priority. That is why, VANET application usually consider *due time*[4] (d_i) of each signature by which it can be scheduled to get a faster response. Alternatively, we can set fixed *due time* to different groups of vehicles like public service buses, personal cars, ambulance etc. For implementation purpose, we propose using *priority queue*[5] to store incoming signatures. Therefore, the lower the *due time* of a signature the higher the *priority.*

However, it is obvious that if we increase the *batch size* (n), it will give us more optimized computation time for n signatures, but similarly increases *completion time* (C_i) of individual signatures. So, there is a trade-off between the *batch size* n and completion time C_i of signatures. That is why choosing the appropriate value for n is not easier for a VANET environment. For example, the value of n during rush hour should be greater than off-peak traffic period at downtown to achieve better performance.

On the other hand, we can utilize idle time (waiting time) with partial computations of incoming signatures as it arrives while batch verification continues, where the value of n is large enough. We observe that the right side (B, K_T, S) of each pairing is constant for n signatures. It contributes to partial computation of left part of pairing (B_i, K_i, Q_i) for consecutive n signatures. That is why, we split the i^{th} signature's pairing verification into 2 parts:

- **Part 1:** Compute $\mathbf{M_i}$, $\mathbf{B_i}$, $\mathbf{K_i}$, $\mathbf{Q_i}$, and $\mathbf{V_i}$.
- **Part 2:** Verify $\prod_{i=1}^{n} \mathbf{M_i} = e(\prod_{i=1}^{n} \mathbf{B_i}, B) e(\prod_{i=1}^{n} \mathbf{K_i}, K_T) e(\prod_{i=1}^{n} \mathbf{Q_i}, S) \prod_{i=1}^{n} \nu_i$

Algorithm 1. determines the optimum value for batch size n by parallelizing partial computations and by maintaining due time (d_i) of each individual signature. Usually a Motor Vehicle Division (MVD) of a country has the statistics of the vehicle movement in a specific zone/highway categorized by periods of time (called peak/ off-peak hours). These traffic statistical data can be utilized *off-line* to fix the best optimum value of n before setting up the batch size into a vehicle.

Consider five *priority queues* $\mathbf{M}, \mathbf{B}, \mathbf{K}, \mathbf{Q}, \mathbf{V}$. As signatures arrive sequentially, $\mathbf{M_i}, \mathbf{B_i}, \mathbf{K_i}, \mathbf{Q_i}$, and $\mathbf{V_i}$ will be calculated and stored in the respective queues. We assume that **Part 2**, where mainly pairings are calculated, take time enough to calculate several future **Part 1** computations in parallel. Total completion time of a batch is calculated from the real-time Time Stamp (TS) differences.

Algorithm 1. works offline with real time traffic data on any specific area where it tries to find out total completion time t_n, and maximum lateness L_n for any batch size

[4] Time Stamp to guarantee the finishing time of a signature verification

[5] This is a regular queue or stack, but additionally, elements are associated with their *priority*

n. It depends on the due time d_i of the arriving signatures into the priority queue. The algorithm *aborts/returns* if maximum completion time of the batch of size n exceeds any signature's due time d_i. However, the output data of Algorithm 1. helps a Motor Traffic Division fixing the potential values of n to fit to the specific region/time. It is obvious that the value of n will reflect a distinct VANET environment and will keep the batch verification system to remain consistent, flexible and efficient.

Signature (i)	1	2	3	4
Processing time p_i	2	2	2	2
Release time r_i	1	2	3	8
Due time d_i	5	7	6	11
Completion time C_i	3	7	5	10
Lateness $L_i := (C_i - d_i)$	-2	0	-1	-1
Unit penalty U_i	0	0	1	0

Max. Completion time C_{max}	10
Max Lateness L_{max}	1

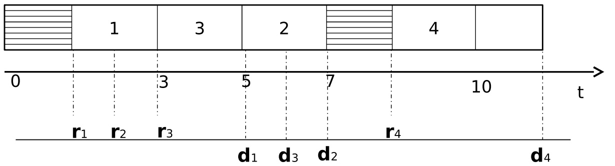

Figure 4.4: A scheduling problem $1|r_i;p_i = p$ with release time r_i, due time d_i, where $p_i = p = 2$, $w_i = w = 1$

Signature without Batch: As mentioned earlier batch verification is not always an efficient solution to the VANET environment. For instance, unlike downtown area, in an uptown area batch verification may cause worse performance. However, in case of tackling sporadic traffic congestion in downtown area, we propose a real-time online *signature scheduling* approach where batch verification is not practical; but still require better performance for prioritized message signatures.

Let each signature include *due time* d_i by which it should be scheduled; *weight* w_i for each signature to provide higher priority to certain vehicles e.g., SOS message from emergency vehicles etc. Alternatively, we can make different groups of vehicles with different w_i values such that public service buses, personal cars etc. By default, w_i has a fixed value. A practical example of scheduling signatures is given in **Figure 3**.

We consider *Single Machine Scheduling Problem* with *release time* r_i and *identical processing time* $p_i = p$ for scheduling [35]; where r_i, d_i and sequence-dependent setup time s_i of signatures are taken into consideration. The objective is to find a feasible set of signatures which are to be completed before or at their deadlines. Let there are n signatures $(i = 1, ..., n)$ to be scheduled, and each signature has its own r_i, d_i, p_i and completion time C_i. A total schedule is expressed as a set X of signatures such that the total weight $\sum_{i \in X} w_i$ is maximal. Let $r_i + p_i \leq d_i$ where r_i is not multiples of p. Signatures are indexed in a non-decreasing order of d_i. Such kind of scheduling problem (described

Algorithm 2 Signature scheduling $(1|r_i; p_j = p| \sum w_i U_i)$

INPUT: r_i, d_i, p_i

OUTPUT: Feasible schedule of signatures with maximal total weight $W_k(s, e)$**.**

```
1. Enumerate the signatures s.t., $d_1 \leq d_2 \leq ... \leq d_n$;
2. $\forall s, e \in T$ with $s \leq e : W_0(s, e) := 0$;
3. FOR ($k = 1$ TO $n$)
   $\forall s, e \in T$ with $s \leq e$
       $W_k(s,e) := \begin{cases} W_{k-1}(s,e) \quad \text{if } r_k \notin [s,e) \\ max\{W_{k-1}(s,e), W'_k(s,e)\} \\ \ otherwise \end{cases}$
   where
       $W'_k(s,e) := \begin{cases} max\{w_k + W_{k-1}(s,s') + W_{k-1}(s',e)\} \\ \text{such that} \ \ s' \in T_i \text{ and} \\ \ max\{r_k, s+p\} \leq s' \leq min\{d_k, e\} - p; \end{cases}$
4. Calculate $W_n(s, e) := (\ \min t - p,\ \max t)$ for $t \in T$
```

in **Algorithm 2.**) is called: $1|r_i; p_j = p| \sum w_i U_i$ where U stands for Unit penalty per late job in the standard scheduling terminology.

$$U_i = \begin{cases} 0 & \text{if } C_i \leq d_i \\ 1 & otherwise \end{cases}$$

A schedule for subset X is feasible if and only if:

1. All signature in the set X start after or at r_i and are completed before or at their d_i
2. They do not overlap in time.

An optimal schedule will exist if each computation of the signature starts at a time belonging to the set. $\{T := r_i + lp | i = 1, ..., n; l = 0, ..., n-1\}$. Let S be an optimal schedule with $i_1, i_2, ..., i_n$ order. It can be transferred to a feasible schedule; for example i_v can be shifted to the left until (r_v, d_v) coincide. For any integer $k \leq n$ and $s, e \in T$ with $s \leq e$. Let $U_k(s, e)$ be the set of signatures where $i \leq k$ with $s \leq r_i < e$, and $W^*_k(s, e)$ is the maximal total weight of a subset of $U_k(s, e)$ with S is idle before $s + p$ and after e and the start time of all $i \in T$. When (r_i, d_i) of signatures are ordered such that $(r_i < r_{i+1}, d_i \leq d_{i+1})$, the problem is solvable in $O(n^2)$ with a dynamic programming algorithm [37].

Jurisdictional access: In addition to the *privacy requirement*, sometimes designated public authority e.g. TSD[6] wants to have access to identity of the vehicles. Consider the verifying entity receives a message which has a valid signature but the message is doubtable to be forged. It might happen if the signer's secret key is compromised. However, if the message is found to be fraudulent then the certificate of the compromised signer is revoked upon request and the revocation list is updated. In order to trace the actual signer of a given signature Υ, TSD does the followings:

- The verifying vehicle submits the message M with its corresponding signature Υ to TSD which computes: $v_1/e(x_O, V_2) = e(H_V(ID_V), B) = \upsilon$
- TSD checks $(\upsilon = W)$ with the entry W previously stored in registration table. If no entry is found, output $\perp$; else issues proof of knowledge ω s.t. $e(x_O, V_2) = v_1/\upsilon$ for justification by the group members.
    - Select $({s_0}', {r_0}', {r_1}') \in \mathbb{Z}_p^*$ and compute:
      $(\Gamma'_0, \Gamma'_1, \Gamma'_2) = (x_O A^{{s_0}'}, e(A, V_2)^{{s_0}'},$
      $e(A, B)^{{s_0}'})$
    - Compute $(\beta'_0, \beta'_1, \beta'_2) =$
      $(H_O(ID_O)^{{r_1}'} A^{{r_0}'}, e(A, V_2)^{{r_0}'}, e(A, B)^{{r_0}'})$
    - Compute $f' =$
      $H((\Gamma'_0, \Gamma'_1, \Gamma'_2)||(\beta'_0, \beta'_1, \beta'_2)||v_1||V_2, \upsilon)$
    - Compute $({z_0}', {z_1}') =$
      $({r_0}' - f'{s_0}'), H_O(ID_O)^{{r_1}'} {x_O}^{f'})$
    - Outputs the proof: $\omega = (\Gamma'_0||f'||({z_0}', {z_1}'))$

Justification: Any group member specially the member vehicle can check the validity of the proof ω whether vehicle ID exposed by TSD is the real signer of corresponding signature Υ for message M by the following:

- Compute:
  $(\upsilon, \upsilon') = (e(H_V(ID_V), B), v_1/\upsilon)$
  $(\Gamma'_1, \Gamma'_2) = (e(\Gamma'_0, V_2)/M', e(\Gamma'_0, B) e(H_O(ID_O), K_T)$
  $(\beta'_0, \beta'_1, \beta'_2) = ({z_1}') {\Gamma'_0}^{f'} A^{{z_0}'}, e(A, V_2)^{{z_0}'} {\Gamma'_1}^{f'},$
  $e(A, B)^{{z_0}'} {\Gamma'_2}^{f'})$
- Compare:
  $f' \overset{?}{=} H((\Gamma'_0, \Gamma'_1, \Gamma'_2)||(\beta'_0, \beta'_1, \beta'_2)||v_1||V_2, \upsilon)$

If the above equation holds, the verifier will be assured about the message signer's ID, hence justified.

[6] It works with Opening algorithm of the Group Signature.

4.2.5 Security Analysis

Three basic requirements for VANET security are reliability, privacy and auditability. We adopt the identity-based group signature scheme described in [2]. Before commencing security analysis, we first give two important theorems on which the security of the system is established. For formal definitions and detailed security proof we refer the interested readers to [2].

Theorem 7 *The IBGS scheme in* [2] *is secure from the random oracle model if and only if the* DDH*-assumption in* $\mathbb{G}_1$*, the* coDBDH*-assumption in* $(\mathbb{G}_1, \mathbb{G}_2)$*, the* k-SDH*-assumption and the* coCDH*-assumption hold.*

Theorem 8 *The IBGS scheme described in* [2] *is traceable in the random oracle model if and only if the* k-$\mathsf{CCA2}$*-assumption holds.*

Our IBGS scheme described in **Section 3** is *anonymous* under DDH and coDBDH assumption. It implies that any PPT-attacker that can corrupt and control a polynomial number of messages and signatures held by the corresponding vehicles, can not reveal the *identity* of a vehicle without the help of GM. Since the $\mathsf{Group\ Join\ Protocol}$ in Section 3.3. is secure, only a registered vehicle can get a membership certificate (D, t, C) w.r.t. its ID. Since $\mathsf{Signing}$ algorithm is secure, without holding a valid secret credential no adversary can produce a valid signature for any message. An attacker cannot cheat other vehicles even by forging a new valid message or modifying a valid message. This ensures the scheme's liability. In this way the scheme provides data integrity and authentication.

Let a vehicle sign the same *message content* several times. Due to *unlinkability* properties, no attacker can link among the signatures whether they are generated from the same vehicle or not.

Non-frameability properties under coCDH-assumption ensures that even if all the vehicles are compromised, none could produce a valid signature such that the TSD would attribute it to a different member vehicle. This strong security notion guarantees that if a message is accepted as valid, it must have been generated by a registered vehicle and not have been tempered with since it has been sent.

The tracing manager TSD of the scheme helps to fulfill the requirement of auditability while preserving honest vehicles' privacy. It can trace a forged vehicle identity by breaking any vehicle's anonymity and issues a proof of knowledge for justification. Later any member entity can justify the ownership of a valid signature by *justification* algorithm. Revocation can also be supported by announcing the counterfeit-list that contains the certificates of all rogue vehicles and later by updating the GM's key and legitimate vehicles' credentials.

4.2.6 Performance Analysis

We compare our scheme with the group signature and batch verification mechanism in [1] as both of them have the same goals and followed the same group signature scheme. Our scheme provides the same security and privacy features with a faster verification mechanism. This is due to the reasons: (1) New signature length (overhead of the signature for verification) for the batch verification is less than the previous one; (2) We reduce the complexity of an individual signature verification and batch verification as well; (3) We follow standard methods for batch verification; (4) We devise an off-line algorithm to select the number of signatures (n) to be batched together; (5) We propose an on-line signature scheduling algorithm for scheduling signatures where batch verification cannot be used.

Signature length: The length of the vehicular generated message can be expressed as the summation of DSRC message and the signature:

DSRC message:
$L_{\mathsf{M}} = L_{\mathsf{type}} + L_{\mathsf{payload}} + L_{\mathsf{TS}} + L_{\mathsf{TTL}} + L_{\mathsf{GID}}$
$= 2 + 100 + 4 + 1 + 2 = 109$ Bytes
where type:= Message (M) type, TS:= Time Stamp, TTL:= Time to Live, GID:= Group ID.

Length of the signature (L_{sig}) is determined by considering a physical security level 2^{80}; setting p to be a 160 bit long prime and the element in $\mathbb{G}_1$ is 161 bit long. Therefore, all the elements in $\mathbb{G}_1$ and $\mathbb{G}_2$ are 161 bits, elements in $\mathbb{G}_3$ is 483 bits, elements in $\mathbb{Z}_p$ are 160 bits. We determine the signature length of the original signature without batch be 361 Bytes and considering batch verification 845 Bytes in [1]. However, our trimmed signature size is 542 Bytes that helps reducing communication overhead without compromising any security requirements.

Communication efficiency: Consider a high traffic metropolitan area with an 8 lane two-way road where the width of each lane is 3 m. According to DSRC a vehicle sends messages within a time interval from 100 to 300 ms and the data rate in DSRC is $\Omega = 6$ Mbps [38]. Let inter-vehicular space be 3 m wide; and vehicles are in movement and transmit DSRC message every 300 ms (3.33 message/second) over 300 m communication range. Therefore, in our signature scheme, a vehicle can hear from a maximum of V_{lis} vehicles [*Upper bound*]:

$V_{lis} = \Omega/(3.33\times(L_M + L_{sig})\times 8)$
$= (6 \times 1024 \times 1024)/(3.33 \times 651 \times 8) = 362$
where $(L_M + L_{sig}) = 651$ Bytes.

If the time interval is 100 ms (10 message/second); a maximum number of vehicles will be $V_{lis} = \Omega/(10 \times 651 \times 8) = 120$. Note that, for $\Omega = 6$, a vehicle can send/receive messages from maximum 120 and 364 vehicles in every 100 ms and 300 ms respectively. However, it will be 82 and 247 respectively in [1] .

Computational efficiency: We consider the signature size and the verification delay of the proposed scheme and the scheme in [1]. The overhead of n signature verification can be expressed as $O(N)$ multi-base pairing computations and multi-base exponentiation. How-

Table 4.2: Batch verification and Signature size

Method	Batch equation	n Sig. verification	Sig. size (Bytes)
V. Wei et.al. [2]	13	11n Pairing + $19n$ Exp. + $19n$ Mult.	361
B. Qin et. al. [1]	13	11 Pairing + $19n$ Exp. + $19n$ Mult.	845
ours	1	3 Pairing + $16n$ Exp. + $12n$ Mult.	542

ever, a vehicle with batch verification needs $O(1)$ multi-base pairing and exponentiation computation. Bi-linear map operation or pairing is the most time-consuming operation in the batch verification algorithm. A typical pairing takes approximately 10 times longer than one exponentiation in $\mathbb{G}_1$. Therefore, we consider the cryptographic computation delay due to pairing, exponentiation, and multiplication operations on elliptic curve since these are the most time consuming operations. It must be noted that there is a probability of invalid signatures as 2^{l_b} for using *small exponent test* where l_b is the security level [29].

For a Miyaji, Nakabayashi, and Takano (MNT) curve of embedding degree $k = 6$ and with order of 160 bit prime the measured processing time[7] for one bi-linear operation τ_p and one point exponentiation are 4.5 *ms* and .5 *ms* respectively [40]. Batching the signatures causes huge cost reduction where vehicular density is too high. Table I shows the summery of the number of computational elements and the signature size for the scheme under consideration.

We test for the verification time on an Intel Core i3 model CPU clocked at 2.43 GHz from the PBC library [42] running on top of Gnu GMP [41] on Ubuntu 12.10. This test makes use of the *a* supersingular curve where the group order is a Solinas prime and the processing time for one bi-linear operation and one point exponentiation are 3.1 *ms* and .3 *ms* respectively.

Table II presents the verification delay in *ms* vs. the number of the received messages. It can be seen that our scheme provides the lowest verification delay among the group signature schemes under comparison. The maximum number of signatures that can be verified simultaneously are 9 and 30 in 100 *ms* and 300 *ms* respectively.

Note that maximum allowed end-to-end message processing delay are 100 *ms* to 300 *ms* in DSRC standard [43]. It can be seen from the **Table II** that although a vehicle may receive 120 (100 *ms* delay) or 364 (300 *ms* delay) signatures at a time to process simultaneously, in practice, it can not process more than 9 (100 *ms* delay) or 30 (300 *ms* delay) because of the maximum *allowed latency*.

Choosing the number of signatures to batch is a challenging problem for any real life VANET environment. It depends on the group signature mechanism used, location of the VANET, standard of wireless communication used, computational efficiency of OBU etc. VANET implementation needs to consider these entities.

[7] Implementation run on an Intel Pentium IV 3.0-GHz machine [39]

Table 4.3: Batch Verification Performance

	Sig.	Ver. time (ms)	Max. Allowed
V. Wei et.al. [2]	1	47.4	2 out of 217 (100 ms)
	10	474	6 out of 660 (300 ms)
	100	4740	
B. Qin et. al. [1]	1	47.4	4 out of 82 (100 ms)
	10	167.1	20 out of 247 (300 ms)
	100	1364.1	
ours	1	18.9	9 out of 120 (100 ms)
	10	105.3	30 out of 364 (300 ms)
	100	969.3	

4.2.7 Conclusion

In a VANET environment, usually large numbers of signatures need to be verified simultaneously. Designing a batch verification mechanism partially addresses this problem, but batch verifications need to be optimized as much as possible. In this work, we have presented an efficient batch verification system from a IBGS group signature scheme for VANET environments. We have further presented a way to determine the upper bound of the number of signatures for batch verification to reduce message loss ratio in a VANET environment. Our scheme not only provides the desired level of security requirements, but also is efficient in storage and computation. We believe it can be implemented in any ad hoc network with limited resource constraints, especially in MANET environments. Signature scheduling can be applied to any MANET system where batch verification cannot be used efficiently. In future, we would like to evaluate the result on a large scale VANET testbed with varying different group signature batch verifications.

4.3 A multi-purpose Group Signature for VANET under standard model

4.3.1 Introduction

Unlike traditional digital signature schemes, GS allows a vehicle to create an *anonymous* (and *unlinkable*) signature that conceals the identity of the vehicle and hence preserves privacy [23][50]. Following the foundation of GS [45], a number of different security requirements have been proposed as primitives. Consequently, BSZ-model in [71], proposes the dynamic GS scheme where members may join or leave the group dynamically. BSZ-model includes three security notions anonymity, traceability and non-frameability that implies all the previously proposed notions of security. Moreover, it separates the role of Group Manager (GM) into: *issuer* and *opener* that meet the requirement of a typical VANET environment where Motor Vehicle Division (MVD) is responsible for issuing license to vehicles (may act as *issuer*) while Traffic Security Division (TSD) is accountable for fraud prevention (may act as *opener*). Furthermore, non-frameability property (by $\mathsf{Judge}(\cdot)$ algorithm) protects the member against being falsely accused of making a signature, even if both the issuer and the opener are corrupt. We utilize this property to sketch a new application framework with value-added service providers (VSPs) in VANET. Note that, a VSP is a third party service provider that could operate as an ordinary group member with additional access to the $\mathsf{Judge}(\cdot)$ algorithm in order to verify the signature as well as the owner of the resp. signature.

We exploit the GS proposed by Groth [74] for several reasons: (1) this scheme is secure in BSZ-model, (2) it offers a constant number of group elements for *group public key* and generated *signatures* (this property is a prerequisite to support scalability), (3) it satisfies strong security requirements, that is, security proof does not rely on weak random oracle model (security proofs in the random oracle model are not sound with respect to that in the standard model). All these features may best fit to a vehicular network model.

Furthermore, security must be considered as an aspect of reliability; and the reliability of the network may lessen due to poor security policy and/or vulnerable cryptographic constructions. Authors in [75] address a security threat (*opening soundness* in [74]) to the reliability of ownership of a signature and provide a solution regarding this. Let a vehicle be registered to a VSP for a certain service. It is mandatory for a VSP to ascertain that it is providing service to the right vehicle to which it has agreement to. But lack of *opening soundness* may allow a malicious vehicle to claim for service as if it is an honest vehicle. This potential threat can be resolved by accumulating *opening soundness* to the *signature* so that by using $\mathsf{Judge}(\cdot)$ VSP can verify the identitiy of the vehicle correctly.

We propose the *opener* to issue a *token* (a proof of ownership of the signature) θ on a *ticket* (message m containing service name and its signature Σ) to the vehicle for a certain service. In order to obtain services from VSPs, a vehicle must submit a valid *ticket* (m, Σ_i) together with *token* (θ) generated on it and its *identity* i. VSPs in response verify the signature Σ_i on m and the identity i of the owner of the signature by examining the proof sealed in the *token* θ.

Although a vehicular network demands group signature schemes that exhibit strong privacy properties, but sometimes stringent privacy policy prevents some reasonable case of application. In order to guarantee vehicle privacy, group signatures can be directly used to anonymously authenticate vehicular communication. We observe that standard

GSs like Groth's GS, is unsuitable for diverse privacy requirement needed for VANET. Therefore, we refer to relax strong privacy properties of Groth GS by introducing *Link manager* (LM) where a designated entity (e.g., RSU) could link the signers anonymously without revealing their identifiers. For instance, let an RSU intend to keep the record of the average number of emergency vehicles pass through a certain junction during business hours without revealing the identity of the vehicles. That is, RSUs need to track the vehicle while preserving the privacy intact. Therefore, we propose a LM to be installed in each RSU that offers linkability while preserving anonymity. When a message together with its signature has been received by the LM, it can link the message with any of the previously received messages from the same vehicle. This feature significantly introduces a privacy hierarchy in VANET from the low level *vehicles* to the upper level *opener*. More clearly, vehicles are fully anonymous in the network, RSUs can only link among vehicles but cannot circumvent anonymity, a VSP is offered to break privacy of the subscribed vehicles only, and an opener can crack full anonymity.

However, revocation is another feature of a GS where members' (vehicles) signing capability are revoked, e.g., if they are declared by the *opener* as *illegal*, or if their secret keys get expired/compromised over time. Therefore, a revocation system is added to the GS that improves key-update efficiency on the Key Issuer side (s.t., constant computation) while restraining efficiency for the individual vehicles (s.t., constant signature size, no secure channel needed to update keys). This also satisfies *backward unlinkability*, that is, signatures produced by a revoked vehicle cannot be linked to its prior signatures.

Note that all the aforementioned GS properties are not completely novel. Firstly, *linkability* feature is discussed in several traceable GS schemes such as [61, 63, 67] and very recently [64]. But all of them either do not support opening algorithm and hence do not allow anonymity revocation, or the security proof belongs to ROM. Secondly, *revocability* properties for a GS was first explored in [58] and later followed by [57, 68, 69]. All the revocable GS schemes have been proposed so far were either reluctant to backward unlinkability, constant signature size/ verification cost/ public key size, or rely on ROM. Recently, two scalable revocation approaches: [76, 77] have been proposed from standard security model. Since the revocation techniques are inspired by broadcast encryption tree, the cardinality of the group becomes fixed and more harshly their signature size is 6 times larger than that of our scheme which could cause performance bottleneck in a large scale VANET application. Thirdly, we followed the *opening soundness* property described in [75] which protect the signature from getting hijacked by other member vehicles.

Main contributions: We introduce a GS scheme, based on pairing-based construction of Groth with additional properties: (1) *linkability* (Link Manager in RSUs), (2) *opening soundness* (token provided by the opener) (3) *revocability* (run by Issuer and group members periodically) We accumulate the aforementioned properties in a single scheme. In addition, for accelerating efficiency we use a simplified version of Groth GS that is CPA-secure, and later suggest applying batch verification technique for standard GS [60] for signature verification.

Network model: We refer to the hierarchical network model described in [23]. In this model, vehicles are remained at the bottom of the hierarchy (see Fig. 4.5). Vehicular groups could be formed: by region (ex. east region), social spots/services (ex. shopping mall, hospital area), category (ex. public service, emergency, personal vehicles) etc. Each vehicle in the network must be equipped with an On Board Unit (OBU) consisting of Event Data Recorder (EDR) that records all the received messages, Tamper Proof Device (TPD) that implements cryptographic tools and ensures authenticated access control. Each GM consists of an *issuer* for the purpose of registration and an *opener* (TSD) to explore the identification of vehicles. Subsequently, all the RSUs would act as LMs.

4.3.2 Extended GS Properties with prior works

Link Manager: Let an RSU intend to collect traffic data (e.g., frequency of emergency vehicles passing through a specific road, which type of vehicles tend to violate traffic rules such as driving over the speed limit etc.) from the road for future traffic analysis without revealing identities of the vehicles. We propose to set Link manager (LM) up into the designated RSU and create vehicular groups according to *category* (such as emergency vehicles).

Besides that, we render traceability with the help of on-demand delegated linkability as follows:

- Firstly, if any suspicious vehicle discovers a *doubtful* message arriving from a group member, it would forward the message with corresponding signature to the LM (preset in the RSUs) instead of *opener* (TSD) for revocation.

- Secondly, RSU is delegated the linking capability by the *opener* that introduces a fine-grained control on the anonymity of vehicles. By using the linking key, RSU can check if two or more doubtful messages have been arrived from the same vehicle.

- Finally, if RSU determines a specific vehicle as *malicious member*, the message together with its signature would be forwarded to the *opener* to reveal the vehicle identity. Usually an *opener* responds only to the privileged verifiers (e.g. RSUs in VANET).

Note that, Traffic Security Division (TSD) should have *policies* on how RSUs would confirm fraudulent vehicles. An example of this would be, if a certain vehicle produces several deceitful messages within a short period of time, or if a vehicle keeps sending multiple messages indicating same events on the road e.g., Sybil attack.

It is worth pointing out that *full anonymity* can not be achieved here since RSUs can link certain vehicles or a group of vehicles, and hence, *absolute privacy* can not be guaranteed. We termed this as *relaxed privacy.* Providing linking capability to a group signature is not novel. For example, direct anonymous attestation scheme (in [61]), ring signature scheme (in [63]) hold linkability algorithm. Unfortunately, these group signature schemes do not include any traceability algorithm. However, a recently proposed GS scheme (in [64]) has both linkabilty and traceability, but the security of the scheme is considered in the random oracle model. Moreover it cannot be guaranteed whether the scheme has *opening soundness* or not.

Note that LM can provide long-term linkability (until the group public key and linking key are refreshed). Sometimes we require short-term linkability for efficient verification with privacy. Short-lived pseudonym is one of the solutions to provide short-term linkability while protecting privacy in VANET. Here we discuss a solution to achieve short-term linkability with pseudonym mechanism. Consider several Group Managers (GMs) under a fully Trusted Party (TP) where Setup phase of each GS would be performed by TP.

Each group member under a GM should use pseudonym signed by the TP instead of original identifier of a vehicle (ID_{V_i}) during Registration with *Issuer* ($\mathsf{User}\ i$ Registration at Section III). Each time TP signs a new pseudonym generated by ID_{V_i}, it ensures that the member is not already revoked.

More clearly, during Setup phase, TP chooses a signature scheme with key pair ($\mathsf{Sign}_T, \mathsf{Ver}_T$) and public Key scheme (PK) with key pair ($\mathsf{sk}_T, \mathsf{pk}_T$). Similarly, GM chooses PK key pair ($\mathsf{sk}_G, \mathsf{pk}_G$). In Registration phase, first each member V_i chooses PK key pair ($\mathsf{sk}_{V_i}, \mathsf{pk}_{V_i}$) for secure communication with TP and seeks a certified pseudonym for its real identifier ID_{V_i}. In response, TP provides the pseudonym Π_i padded with expiration date (Timestamp) and its signature (encrypted by pk_{V_i}) to the member vehicle. After first successful registration to the TP, a member vehicle may update its pseudonym Π_i any time *online*. However, Issuer in Registration phase of GS uses Π_i (instead of x_i in the current scheme). Vehicles entering a new GM$_i$ area should provide a valid pseudonym (not expired) to receive the group secret key (gsk_i) for future communication within a GM$_i$'s area.

In this scenario, Revocation would be accomplished by the cooperation of global TP and GMs. For instance, during revocation Traffic Security Division (with Open algorithm) of GM$_i$ can extract the member pseudonym that would be forwarded to the TP in order to extract the original ID of a vehicle (ID_{V_i}). TP then updates its global revocation list accordingly and ensures that any malicious member in the revocation list cannot update its pseudonym in the next registration phase. Later TP broadcasts updated revoked member list to all the active GMs so that they can check the temporary revoked members until the lifetime of the pseudonym expire. Hence, using pseudonym facilitates flexible linkability with expiration date (while LM provides long-term linkablity inside a group) independently of the GMs.

Furthermore, sometimes short-term linkability can be achieved by fixing some parameter during Authentication phase (with Gsign algorithm). For instance, if ρ is unchanged in our scheme in the consecutive n signatures, it will generate same $(a, \varkappa)$ (part of Σ) for n signatures. Hence, short-term linkability is accomplished.

Opening soundness: Groth's group signatures are susceptible to be hijacked by a malicious member by forging the *proof of ownership* generated by the *opener* [75]. We present a secure application framework by utilizing this property. For instance, let a vehicle have an agreement with a third party service provider. It would generate a message (citing the VSP's name and requested service information) with its signature (we termed it as a *ticket*) and submit them to the Traffic Security Division (conveying *opener* algorithm) for attestation. Opener would issue a proof of ownership (we termed it as a *token*) of the signature in order to bind a credential to its legitimate owner (see Fig. 4.5). Subsequently, later when the vehicle requests for a service to the VSP, it would attach a *ticket* and its corresponding *token* issued by the *opener*. VSPs (conveying *judge* algorithm) could justify the message with the credential of the vehicle.

Revocation: Like standard PKIs, GS does not have any efficient revocation system in practice. Many existing solutions do not scale well due to either high overhead or tight operational requirements, such that, computational complexity belongs to $O(n)$

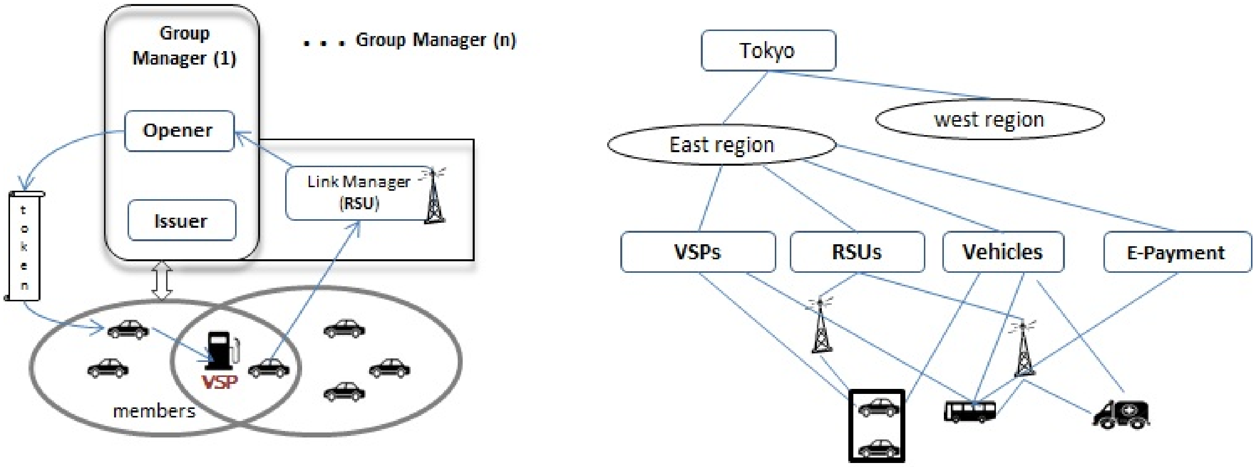

Figure 4.5: An example demonstrating: vehicles' group formation (right) and communication among the GS members (left) in a traditional VANET Network.

or $O(r)$, where n and r are group size and number of revoked members respectively. Revocation solution was first introduced in [58], where the signature size was linear to the number of revoked members. Authors in [72] proposed a forward secure revocation system with constant signature size. But, one of the features of this scheme was to use fixed time periods to revoke a member, which is in fact, impossible to implement in VANET environment. Schemes in [68] [69] have $O(1)$- cost for signing and verification time but $O(n)$-size (linear) group public keys.

Recently, two revocations approaches have been proposed, mainly based on the Naor-Naor-Lotspiech (NNL) Broadcast Encryption framework that yields a scalable revocable group signatures to obtain private keys of constant size in the standard model [76] [77]. Unfortunately, signature size of both the schemes are too large for practical deployment. They are approximately 3 and 6 times larger, respectively, than that of our scheme[8]. Moreover, since NNL is a tree-based technique, unlike ordinary dynamic GS schemes the maximal cardinality of the group would be fixed. Therefore, even though the revocation schemes are truly scalable, they cannot be used for VANET application where larger signature size causes increased communication overhead and hence degrades overall performance and the number of group member vehicles should be flexible, not fixed.

We exploit the idea of [57] in our GS, where they offer a CRL-like revocation with constant length signature as well as constant computation for revocation, that means, the complexity is $O(1)$ with respect to n and r.

If a group member vehicle leaves the group or is declared as an *illegal*, Issuer updates the *RList* accordingly. We propose not to update group public key (*gpk*) in every case when a new member leaves or is forcibly revoked from the group for the sake of efficiency. Instead, information regarding new/revoked group members can be accumulated between two successive revocation events.

[8] Group signature size of [76] and [77] are comprised of 144 and 92 group elements respectively while our signature size consists of 28 group elements.

4.3.3 The Proposal

Groth GS applies *certified signature* method based on the **DLIN** and the $q-\mathbf{U}$ assumption (see [74] for details) using Non-interactive Witness-indistinguishable (NIWI) proofs[73]. Note that we present a relaxed (CPA-secure) notion of Groth GS e.g., allow no adversarial access to the *open* algorithm and add/modify some generic algorithm e.g., adding: $\mathsf{SignLink}()$, $\mathsf{Revoke}()$ modifying: $\mathsf{Keygen}()$, $\mathsf{Registration}()$, $\mathsf{Open}()$ algorithms.

System Setup: Consider a probabilistic polynomial time algorithm $\mathcal{G}$ that generates $gk := (p, \mathbb{G}, \mathbb{G}_{\mathbb{T}}, e, g) \leftarrow \mathcal{G}(1^k)$ such that: p is a k-bit prime, $(\mathbb{G}, \mathbb{G}_{\mathbb{T}})$ are cyclic group of order p. Let g generate $\mathbb{G}$ and e be a non-degenerate and efficiently computable bilinear map s.t., $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{\mathbb{T}}$ so that $e(g, g)$ generates $\mathbb{G}_{\mathbb{T}}$, and $e(g^a, g^b) = e(g, g)^{ab}$ for any $a, b \leftarrow \mathbb{Z}_p$.

Key Generation $\mathsf{GKg}(1^k)$: Group Manager (Traffic Escrow Authority) generates secret keys: ik for Issuer (Registration managers), ok for Opener (Traffic Security Division), lk for Linker (Designated RSUs) and public system parameters gpk. Let $(u, h, z, K, L) \leftarrow \mathbb{G}$, $(l, r, s) \leftarrow \mathbb{Z}_p$, $f = u^l$, $T := e(f, z)$, $xk := (\phi, \eta) \leftarrow gk, F := g^{\phi}, H := g^{\eta}$, , $R := g^r, S = g^s$, Hash $\leftarrow \mathcal{H}(1^k)$; $\mathsf{Parse}(crs) := (F, H, therest)$. $pk := (F, H, K, L)$, $lk := l$, $ik := z$, $ok := xk$, and $gpk := (gk, \text{Hash}, u, f, h, T, crs)$

Registration (User $i : gpk$, Issuer: gpk, ik): Group members with their identity i (e.g., vehicles, RSUs) need to complete registration with Issuer. Let total number of non-revoked vehicles be n in an instance. A vehicle i and Issuer run a 5-move key generation protocol (described in [28]) in order to generate a key pair $\{(v_i, x_i),\ v_i\ \}$, where $v_i \leftarrow g^{x_i}$ Issuer then signs v_i to produce certificate
$\mathsf{certSign}_i := (a_i, b_i) \leftarrow (f^{-r_i},\ (v_i h)^{r_i} z)$, where $r_i \leftarrow \mathbb{Z}_p$.
Vehicle i accepts the certificate $\mathsf{certSign}_i$ if $e(a_i, hv_i)\ e(f, b_i) = T$. Finally, the Issuer maintains a database to store $reg[i] \leftarrow v_i$ for the $\mathsf{open}()$ and the $\mathsf{judge}()$ algorithm, and $rev[i] \leftarrow r_i$ for the $\mathsf{revocation}()$ algorithm and the vehicle i stores group signing key $gsk[i] \leftarrow (x_i,\ \mathsf{certSign}_i)$

Authentication $\mathsf{GSign}(gpk, gsk[i], m)$: In order to sign a message m a registered vehicle i first generates a certified signature σ using her private key x_i. Then it produces a NIWI proof[9] π that consist of a commitment to σ. The detailed instantiation is as follows. Let a vehicle i select $\rho \leftarrow \mathbb{Z}_n$ and compute $a := a_i f^{-\rho}$, $b := b_i(hv_i)^{\rho}$, $\varkappa = u^{-\rho}$ and $\sigma := g^{1/x_i + \mathcal{H}(m)}$. $\pi \leftarrow \mathrm{P}_{\mathrm{NIWI}}\ (crs, (gpk, a, \mathcal{H}(m)), (b, v_i, \sigma))$ The resulting signature on a message m is: $\Sigma := (a, \varkappa, \pi, \sigma)$.

Message verification $\mathsf{GVerify}(gpk, m, \Sigma)$: To verify a signature Σ on message m, receiving vehicle or RSU checks NIWI proof π:

```
IF $\mathrm{V}_{\mathrm{NIWI}} \leftarrow (crs, (gpk, a, \mathcal{H}(m)), \pi) = true$;
    return 1
ELSE return 0
```

[9] To demonstrate that ciphertext contains a valid certified signature

Traffic Security Division $\mathsf{Open}(gpk, ok, m, \Sigma)$: By accessing the registration table $reg[]$[10] generated by the Issuer, by using opening key ok it can revoke the signer's identity i of a valid signature Σ on message m. This algorithm can be used for two purposes: Firstly, it helps to exhibit the signer of a doubtable message/signature sender and later revoke the member vehicle from the group. Secondly, it promotes accountability of certain applications by providing proof of ownership of a certain signature. Consider a member vehicle i that requires a credential regarding a service which is mentioned in the message m. The vehicle could first generate a signature Σ on m and then request the Opener() to provide a proof of ownership or token on m. After that, it could submit m to the VSP along with Σ and the token to justify. The detailed are as follows:
First, it verifies the signature by using $\mathsf{GVerify}\ (gpk, m, \Sigma)$. If successful, then it extracts v of the corresponding vehicle i and searches the registration table to find $v \overset{?}{=} v[i] \leftarrow reg[i]$.

$(b, v, \sigma) \leftarrow \mathsf{Extract}\ _{ok}(crs, (gpk, a, \mathcal{H}(m)), \pi)$.

In order to generate proof of ownership, it randomly selects $(c, d) \leftarrow \mathbb{Z}_p$ and computes: $(y_1, y_2, y_3) := (F^c, H^d, v_i g^{c+d})$ and a Non Interactive Zero Knowledge (NIZK) proof $\theta \leftarrow (\theta_1, \theta_2)$ of corresponding vehicle i where $\theta_1 := y_1^{1/\phi}$, $\theta_2 := y_2^{1/\eta}$ and $(\phi, \eta) \leftarrow ok$. Finally, it issues (i, y, σ, θ) which is termed as a proof of ownership of a signer i on a certain message m, or a *token*.

Validating Ownership $\mathsf{Judge}(gpk, i, v_i, m, \Sigma, \theta)$: This algorithm verifies whether the opening is correct or not. It returns 1 if the opening is correct. VSPs in VANET could use this algorithm to verify the beneficiary of a certain service.

```
IF $\Big(\textbf{GVerify}(gpk, m, \Sigma) = 1 \bigwedge (i \neq 0) \bigwedge e(\sigma, v_i g^{\mathcal{H}(m)}) = e(g, g) \bigwedge e(F, \theta_1) = e(y_1, g) \bigwedge e(H, \theta_2) = e(y_2, g) \bigwedge \sigma\theta$
return 1
ELSE return 0
```

Managing Linkability $\mathsf{SignLink}((\Sigma_1, m_1), (\Sigma_2, m_2), lk)$: By using lk, the LM (e.g., designated RSUs in VANET) tries to find a link among existing list of signatures with a new signature, or between two signatures whether they are generated from the same signer i. It returns 1 if sucessful Let $a_1, \varkappa_1 \leftarrow \Sigma_1$ and $a_2, \varkappa_2 \leftarrow \Sigma_2$.

```
    IF $\mathsf{GVerify}\ (gpk, m_1, \Sigma_1) \bigwedge \mathsf{GVerify}\ (gpk, m_2, \Sigma_2)$
        IF $e(a_1, h)\ e(\varkappa_1, h^{lk})^{-1} = e(a_2, h)\ e(\varkappa_2, h^{lk})^{-1}$ Or,
            $e(a_1/a_2, h) = e(\varkappa_1/\varkappa_2, h^{lk})$
            return 1
    ELSE return 0
```

Intuition: $\rho_i \neq \rho_j$ and $gsk[i] \neq gsk[j]$ for any (i, j)

$$e(a_1/a_2, h) = e(\varkappa_1/\varkappa_2, h^{lk})$$
$$\Rightarrow e(a_i f^{-\rho_1}/a_i f^{-\rho_2}, h) = e(u^{-\rho_1}/u^{-\rho_2}, h^l)$$
$$\Rightarrow e(u, h)^{l(\rho_2 - \rho_1)} = e(u, h)^{l(\rho_2 - \rho_1)}$$

Since $a_i \leftarrow \mathsf{certSign}(a_i, b_i)$ is randomized by ρ to generate a in $\mathsf{GSign}()$, there would be no security compromise.

[10] The opener has read access to the registration table $reg[]$

Revocation $\mathsf{Revoke}(gpk, RList)$: Revocation would be accomplished in two steps: Firstly, GM issues a new group public key gpk including all new parameters, termed as $\mathcal{R}$, and publish it for all the non-revoked members. Usually, the Issuer publishes a signed and time-stamped $\mathcal{R}$ in a publicly accessible bulletin board or server. Unlike ordinary GS schemes, in our scheme vehicles do not need to contact the *issuer* privately (following interactive *join/issue* protocol) to update their certificates. Secondly, after getting the public parameters $\mathcal{R}$ for revocation, all the non-revoked member vehicles can update their certificates (a_i, b_i) with the newer one consequently. However, it is quite likely that no revoked members can update their certificates from the revocation information available in public. Moreover, all other non-revoked member vehicles need $O(1)$ operation to update, irrespective of the size of the revocation list or the group members.

This algorithm allows Issuer and all non-revoked member vehicles to update their keys according to the revoked users list $RList$ provided by the GM. Let $t := \{\prod_{i=1}^{n} r_i, s.t.\ r_i \leftarrow rev[i]\}$ be known to all the last known non-revoked n group member vehicles. Note that, t considers of all the current non-revoked members including the *new* member vehicles that join between two consecutive revocation events.

Let m member vehicles be adjudged as *illegal* vehicles between two successive revocation events, and $r_{ki} \leftarrow rev[i]$ be selected for the revoked members (m). Then, $RList := k_1, k_2 \cdots k_m$ where $m < n$; and $r_k = \prod_{i=1}^{m} r_{ki}$.

Issuer: update $rev[i]$ according to the new list of non-revoked member vehicles (n)
$\tau \leftarrow \mathbb{Z}_n; \delta := \tau^l;\ u' := u \cdot \tau;\quad f' := f \cdot \delta;\quad h' = h \cdot \delta$
$T' := e(f', z)$; and $\gamma := \delta^{\frac{t}{r_k}} \bmod n$
new $gpk := (gk, \mathrm{Hash}, u', f', h', T', crs)$
publish $\mathcal{R} \leftarrow (t, gpk, \gamma, r_k)$ for the non-revoked members.
Member vehicle $(i \neq k_i)$: update non-revoked member's certificate $certSign_i(a_i, b_i)$:
$gsk[i] := (x_i, {a_i}', {b_i}') \leftarrow (x_i, a_i, b_i)$
set $s_i = \frac{r_i \cdot r_k}{t}$
set $a_i' = a_i \cdot \gamma^{-s_i}$ and $b_i' = b_i \cdot \gamma^{s_i}$

4.3.4 Security Requirement

Some of the notations and security definitions we use from [74] [75] and also omit the description of security proof due to space constraint. Interested readers are referred to [74] [75] for further discussion.

Definition 5 Let $(p, \mathbb{G}, \mathbb{G}_{\mathbb{T}}, e, g)$ be a bilinear group. The Decision linear ***(DLIN)*** assumption states that it is hard to distinguish in (f, g, h, f^r, g^s, h^t) and $(f, g, h, f^r, g^s, h^{r+s})$ for random group elements $a, b \leftarrow \mathbb{Z}_p$ *[59]*.

Lemma 1 Modified Groth GS satisfies the revocability under the DL-assumption and provides backward unlinkability.

Proof: Issuer publishes $\mathcal{R} \leftarrow (t, gpk, \gamma, r_k)$ that includes group public key *gpk* and other necessary parameters in public. Note that, all the updated *gpk* parameters (u, f, h, T) are randomized by δ, and $\gamma := \delta^{(t/r_k)}$ is published as part of $\mathcal{R}$. γ is calculated only from the non-revoked members r_i (from $rev[i]$ pre-stored to Issuer). In order to sign a message, a non-revoked member need to create a valid *certSign* by following

$$(a_i', b_i') \leftarrow (a_i * \delta^{-r_i}, b_i * \delta_i^{r}) \text{ s.t., } \gamma^{s_i} = \delta^{(t/r_k)*(r_i*r_k)/t}$$

However, it is impossible for a revoked member to produce new *certSign*. Because it is hard to explore δ from γ under DL-assumption. Therefore, it is hard for a PPT adversary $\mathcal{A}$ to produce a colluding non-revoked member.

Let the adversary $\mathcal{A}$ be able to link signatures generated before and after a revocation phase. Thus, in order to break backward unlinkability, $\mathcal{A}$ needs to distinguish two signatures Σ_a (generated after revocation), Σ_b (generated before revocation). It appears that Groth GS scheme provides anonymity under DLIN assumption[11]. Moreover, during each signature generation, the parameters $(a, b, \varkappa)$ are randomized by ρ, and σ is independent of the updated parameters during revocation, since it is generated from the secret x_i. Furthermore, linkability from π is also infeasible, since it is a proof from NIWI that assures indistinguishability from the secrets/witnesses it possess, based on a variant of DDH assumption. □

Lemma 2 Modified Groth GS is linkable under DL assumption.

Proof: We use CPA-anonymous version of the Groth GS. That is, signature is untraceable under DLIN assumption. Similarly, we assume that any PPT adversary $\mathcal{A}$ does not have access to $open(\cdot)$ oracle and thus does not have access to open key *ok*. Unlike anonymity-game, in the linkability-game $\mathcal{A}$ has access to the *linking* key *lk* in order to find a link among signatures from the same signer, or a group of signers while not being aware of the real signers of the signatures. However, the adversary $\mathcal{A}$ can compute a linking index: $e(a_i f^{-\rho_i}, h)$ associated with each signer i where (a_i, ρ_i) pair is associated with a signer i. Let LM create a database that is indexed by $e(a_i f^{-\rho_i}, h)$. We assume this index is singular and uniformly distributed from adversarial point of view. Clearly, this index is unique and independent of the signer's signing key $gsk[i] \leftarrow x_i$. Therefore, it is hard for a PPT adversary $\mathcal{A}$ to guess the identity i of the signer from a given signature Σ.

4.3.5 Efficiency

We minimize and exploit a simpler variant of Groth GS [74]. Therefore, we provide construction for relaxed security notions (CPA anonymity) that removes the non-essential features of the main GS. Meanwhile, we extend the existing Groth GS to satisfy some essential security notions with minor performance overhead. However, ordinary CCA-anonymous Groth GS consist of 50 group elements in $\mathbb{G}$ while the lighter version, where CPA-anonymity is sufficient and the adversary is not allowed to access *opening* oracle[12], the size of signature can be reduced to 28 group elements. Still it supports dynamic

[11] A natural extension of DDH assumption

[12] In VANET, Traffic Security Division (Opener) is commonly assumed to be tamper-proof

Table 4.4: VANET Security properties

	Ours	J. Hwang[64]	MSI Mamun[50]	L. Zhang[65]	W. Lingbo[66]
Security Proof	Standard	ROM	ROM	ROM	ROM
Anonymity	CPA	CPA	CCA	CPA	CPA
Linkability	Yes	Yes	No	No	No
Revocability	Yes	Yes	No	Yes	Yes
Non-frameability	Yes	Yes	Yes	Yes	No
Opening Soundness	Yes	No	No	No	No
Batch verification	Yes	No	Yes	No	Yes

member enrollment, constant number of group elements in *keys* and *group signatures*, opening soundness, feasible revocation, linkability to achieve relaxed privacy through LM. In [50], the authors show how efficiency degrades in relation to pairing computation in VANET environment and propose some solutions to speed up the signature verification process. In [60], the authors address this challenge for Groth signature and propose a batch verification system to reduce almost 90% of the pairing calculation.

However, introducing batch verification for single signature has reduced expensive pairing equation per signature from 68 to 11. While the batched version requires only $4n + 7$ pairings for n signatures. In addition, introducing off-line signature scheduling algorithm to find an optimum value of the batch size n, and paralleling partial pairing calculation using *thread*, as described in [50], can further optimize the final operation time for signature verification.

However, if we allow LM to be used in each vehicle for short-term linkability, it significantly improves signature verification. As the message with signature arrives to the vehicle, it will first search the local database whether the sending vehicle is already known to it (by using LM key it can easily link the incoming signature with any previous record from the same vehicle). If the sending vehicle is enlisted already in the receiving vehicle's local database (e.g., second (or higher) message from the same sending vehicle), expensive verification part (e.g., 11 pairing calculation) can be omitted. For instance, if a receiving vehicle requires 11 pairing calculation for the first signature it has received from a vehicle i, it presumably need no pairing calculation from the second or any subsequent signatures coming from the vehicle i until no suspicious/deceitful message is claimed by the receiving vehicle. Nevertheless, over time the local database of the receiving vehicle can become enlarged that would cause performance bottleneck in database searching.

Finally, in Table 1. we compare our scheme with some other recent GS schemes proposed for VANET in terms of security properties, security proof method, Linkability, Non-frameability, Revocability, Opening Soundness and Performance etc.

4.3.6 Conclusion

In this work, we focus on hierarchical privacy-preserving among all entities of VANET by using Groth GS. We have presented a reliable and standard CPA-secure GS solution to a vehicular network application considering revocability, linkability and opening soundness. We consider the lighter version of Groth GS to enhance efficiency while preserving optimal security with several essential properties. Further, we suggest LM that provides restricted privacy appropriate for a real time VANET environment. Moreover, this can protect against DoS and Sybil attacks as well. In addition, using batch verification can significantly improve the performance of signature verification that makes the solution applicable for real life vehicular communication.

Chapter 5

RFID system Security

5.1 A privacy-preserving RFID authentication protocol

5.1.1 Introduction

Tag authentication is an indispensable approach to prevent an RFID tag from impersonation. In particular, tag authentication is more significant since tags are much vulnerable to counterfeit than readers. However, mutual authentication protocols add an additional protection for the RFID system in the protocol construction to safeguard the query is, in fact, coming from a legitimate reader.

Unlikability or Untraceability, sometimes referred to interchangeable with same meaning, conveys the property that an adversary cannot distinguish whether two events occurring in an RFID system are related to the same tag or not. In addition, anonymity is another indispensable security property that assure the inability to identify a tag within an RFID system. This definition can be framed in terms of unlinkability by saying that a tag is anonymous in any transactions between the reader provided that adversary cannot link the tag to a transaction. In order to provide aforementioned security properties, ample research has been done in this area targeting enhanced privacy, security and performance issues. Since asymmetric key ciphers are too expensive for a compact hardware such as low-cost RFID tag, majority of the authentication protocols use *symmetric key* as secret. For example, RSA require more than 30,000 gates, which is too expensive for low-cost tag where maximum 2,000 gates out of 10,000 gates are available for the purpose of security [79].

The LPN problem is a light-weight provably-secure cryptographic scheme which was first introduced in 2001 by Hopper & Blum [86]. LPN based authentication is not only *theoretically secure* in terms of provable security, but also provides better *efficiency* than classical symmetric ciphers that are not related to hard problems. There has been a large body of research on HB protocol that outputs protocols such as HB$^+$, HB^{++}, HB$^{\#}$, HB-MP, HB-MP$^+$, HB* etc.[89, 90, 91, 92, 93, 95, 108]. Unfortunately all of them later shown to be insecure or susceptible to particular attacks [94, 95]. In [80], authors propose an authentication protocol based on the Subspace LPN (SLPN) problem with tight security reduction which is as efficient as the previous HB-family, but has twice the key length; in addition, their proof works in quantum setting, which leads the protocol to be secure against quantum adversaries.

5.1.2 Preliminaries

In this section, we discuss some inevitable assumptions followed by useful definitions for primitives and security notions.

Assumption: We assume the RFID system described in this work consist of a single legitimate reader and a set of tags (EPC global Class 1 generation 2). The reader is connected to the back-end server that stores all the relevant data including the tag database. Initially, the reader generates and set T_{id} and *public parameters* depending on security parameter λ. Each tag has its unique identification T_{id} and session key S_i. T_{id} is used as the shared secret key between the tag and the reader. The authentication protocol is an interactive protocol executed between *tags/prover* and a *reader/verifier* where both are *probabilistic polynomial time* (PPT) algorithms. All communications between the server and the reader are assumed to be secure and over an authentic channel. For simplicity, we consider the reader and server as identical. Throughout the work, we use the term reader and server interchangeably. A tag is not a tamper-resistant device; so its session key S_i is refreshed after each session is completed successfully. For updating the key, the tag authenticates the reader first. An adversary cannot compromise the reader/server and cannot corrupt the tag until it compromises both T_{id} and S_i at the same time. However, if both of the secret keys are exposed at a time, the adversary can trace the tag for a certain period i until the next authentication cycle starts. We assume tag binary identification T_{id} is unique within an RFID system. To avoid an exhaustive database search at the reader, hash-index (I) is used. Database at the server associates the tag index with other tag-related data e.g., T_{id}, S_i, P_i etc.

Definitions for security notions:

Definition 6 *A protocol is secure against* **passive attacks***, if there exists no PPT adversary $\mathcal{A}$ that can forge the verifying entity with non-negligible probability by observing any number of interactions between the tag and reader.*

Definition 7 *A (t, Q, ϵ)-hard protocol is called secure against* **active attacks** *where the adversary $\mathcal{A}$ runs in two stages: First, it observes and interrupts all the interactions between the target tag T and legitimate reader with concurrent executions according to the defined security. Then, it is allowed only one time to convince the reader. Note that, this time $\mathcal{A}$ is not allowed to continue his attacks in time instance t; but can utilize several discrete or successive time period.*

Definition 8 *In the* **Man-In-the-Middle** (MIM) *attack, adversary $\mathcal{A}$ is allowed to maintain connections with both the tag and the reader, making the tag believe that they are talking directly to the reader over a secure connection, when in fact, the entire communication is controlled by $\mathcal{A}$. Then, $\mathcal{A}$ interacts with the reader to authenticate. The goal of the attacker $\mathcal{A}$ is to authenticate successfully in Q rounds. $\mathcal{A}$ is successful if and only if it gets accept response from all Q rounds.*

Definition 9 *The* **Forward security** *property means that even if the adversary obtains the current secret key, it cannot derive the keys used for past time periods.*

Definition 10 *The* **Backward security** *is opposite to the forward security. If the adversary can explore the secret of the tag at time i, it cannot be traced in future using the same secret. In other words, exposure of a tag's secret should not reveal any secret information regarding the future of the tag. But if an adversary is allowed to obtain full access to the tag's secret, and thus can trace the target tag at least during the current session of authentication immediately following the attack, it does not make any sense to perfect security in practice. Therefore, it is impossible to provide backward security for an RFID-like device practically.*

Definition 11 **Tracking a tag** *refers the attacker could guess the tag identity or link multiple authentication sessions of the same tag. In our protocol, the adversary cannot recover S_i or any other information identifying that particular tag.*

Definition 12 *In* **De-synchronization attack***, the adversary aims to disrupt the key update, leaving the tag and the reader in a desynchronized state and renders future authentication impossible.*

Definition 13 **Denial of Service** (DoS) *is an attempt to make a tag unavailable to its intended users. DoS resistance capability of the protocol is infinite as tag updates the key after reader authentication is successful.*

Definition 14 **Tag cloning** *entails that the data on a valid tag is scanned and copied by a malicious RFID reader, and later the copied data will be embedded onto a fake tag.*

Definition 15 *In the* **replay attack***, an adversary reuses the communication scripts from the former sessions to perform a successful authentication between each tag and their reader.*

Definition 16 *An* RFID *system, is said to unconditionally provide privacy notion X, if and only if for all adversaries $\mathcal{A}$ of type X, it holds that $Adv_{\mathcal{A}}^{X}(\lambda) = 0$. In case of computational privacy, it is $Adv_{\mathcal{A}}^{X}(\lambda) \leq \epsilon$ for all PPT adversaries $\mathcal{A}$*[102].

Definition 17 *An* RFID *system is said to be (Q, t, ϵ)* ***strong private****, if there exist no (Q, t) adversary $\mathcal{A}$ who can break its strong privacy with advantage $Adv_{\mathcal{A}}^{b}(k) \geq \epsilon$.*

5.1.3 Construction

We adopt the idea of key-insulation to slightly twist our 3-round mutual authentication protocol described in Fig. 1. The protocol allows significantly less computations to a tag. On the other hand, the most expensive computations of the protocol are handled by the reader. We use only random generation, bitwise XOR and matrix multiplication as tag operation. The protocol uses $(\lambda, \tau, \tau', n, l)$ as public parameters, where (τ, τ') are constant while (l, n) depends on the security parameter λ. For initialization, the server generates the initial index I_0, the session key S_0 and its corresponding P_0 and other public parameters; and set the necessary data into a tag non-volatile memory. Note that, we use *matrix* as a secret, not a *vector*. Therefore, for each tag, there is a tuple

Reader $(I_i, T_{id} \in \mathbb{Z}_2^{2l}, \mathbf{S_i} \in \mathbb{Z}_2^{l \times n}, \mathbf{P_i} \in \mathbb{Z}_2^{l \times l})$	**Tag**$(I_i, T_{id} \in \mathbb{Z}_2^{2l}, \mathbf{S_i} \in \mathbb{Z}_2^{l \times n})$
$s \in_R \mathbb{Z}_2^{2l}$; where $\mathbf{w}(s) = l$	
$\xrightarrow{\mathbf{s}}$	
	if $\mathbf{w}(s) \neq l$ **return**; $\mathbf{e} \in_R \mathbf{Ber}_{\tau}^{n}$; $\mathbf{r} := [\mathbf{S_i}]^{\mathbf{T}}.\ (T_{id\downarrow} s) \oplus \mathbf{e}$ $s' \in_R \mathbb{Z}_2^{2l}$; where $\mathbf{w}(s') = l$ $I_{i+1} = r$
	$\xleftarrow{(\mathbf{I_i}, \mathbf{s'}, \mathbf{r})}$
Lookup T_{id} by using hash-table index: **Direct match**: $I = I_i$; if (not found) then **Brute-force search**: find an entry $[I_i, T_{id}, \mathbf{S_{i-1}}, \mathbf{S_i}, \mathbf{P_{i-1}}, \mathbf{P_i}]$ s.t., $\exists$ ($\mathbf{S_i}$ or, $\mathbf{S_{i-1}}$), for which the following satisfies: **If** $\mathbf{w}([\mathbf{S_i}]^{\mathbf{T}}.(T_{id\downarrow} s) \oplus \mathbf{r}) > n.\tau'$ **return**; **Else** $I_{i+1} = r$ if $\mathbf{w}(s') \neq l$ **return**; Generate non-singular $[\mathbf{Q}] \in_R \mathbb{Z}_2^{l \times l}$ $[\mathbf{S_{i+1}}] = [\mathbf{Q}].[\mathbf{S_i}] \in \mathbb{Z}_2^{l \times n}$ where $rank(\mathbf{S_{i+1}}) = n$ Compute $\mathbf{P_{i+1}} = [\mathbf{S_{i+1}}][\mathbf{S_{i+1}}]^{+} \in \mathbb{Z}_2^{l \times l}$ where $[\mathbf{S_{i+1}}]^{+} = ([\mathbf{S_{i+1}}]^{\mathbf{T}}[\mathbf{S_{i+1}}])^{\mathbf{-1}}[\mathbf{S_{i+1}}]^{\mathbf{T}} \in \mathbb{Z}_2^{n \times l}$ $\mathbf{P_i}' = [\mathbf{P_i}][\mathbf{Q}] \in \mathbb{Z}_2^{l \times l}$; $\mathbf{e}' \in_R \mathbf{Ber}_{\tau}^{n}$; $\mathbf{r}' := [\mathbf{S_i}]^{\mathbf{T}}.\ (T_{id\downarrow} s') \oplus \mathbf{e}'$	
$\xrightarrow{(\mathbf{P_i}', \mathbf{r}')}$	

	if $\mathbf{w}([\mathbf{S_i}]^{\mathbf{T}}.\ (T_{id\downarrow} s') \oplus \mathbf{r}') > n.\tau'$
	return; else **accept**
	$\mathbf{S_{i+1}} = (\mathbf{P_i}'.\mathbf{S_i}) \in \mathbb{Z}_2^{l \times n}$
	if $rank([\mathbf{S_{i+1}}]) \neq n$ **return**;

Figure 5.1: RFID Authentication Protocol

$[I_i, T_{id}, S_{i-1}, S_i, P_{i-1}, P_i]$ to be stored in the back-end database of the server at any time instance i.

For *tag* authentication, let a tag have S_i and I_i, which have been derived from the previous $(i-1)$ successful authentication sessions.

- Reader: Generate a random binary challenge string s, and sends it to a tag.

- Tag: Check the *hamming weight* of the string s and generate an n-bit noise vector

$\mathbf{e}$, a random $2l$-bit challenge string s' for a reader with hamming weight l. Next an n-bit LPN problem is computed as $\mathbf{r} := [\mathbf{S_i}]^{\mathbf{T}} \cdot (T_{id\downarrow}s) \oplus \mathbf{e}$. To eliminate brute-force searching at the server end, maintain an index I_i and send it to the reader. Finally, update index I_{i+1} to r and send $(I_i, s', \mathbf{r})$ to the server.

- Reader: First search database to find a tuple $[I_i, T_{id}, S_{i-1}, S_i, P_{i-1}, P_i]$ with index I sent by the server. But searching might fail sometimes e.g., due to synchronization attack etc. If it fails, then apply brute-force method targeting to explore $\mathbf{S_i}$ or $\mathbf{S_{i-1}}$ such that it satisfies LPN problem: $\mathbf{w}([\mathbf{S_i}]^{\mathbf{T}} \cdot (T_{id\downarrow}s) \oplus \mathbf{r}) \leq n \cdot \tau']$, or $[\mathbf{w}([\mathbf{S_{i-1}}]^{\mathbf{T}} \cdot (T_{id\downarrow}s) \oplus \mathbf{r}) \leq n \cdot \tau']$. If the brute-force method passes, it accepts the tag, update the index to I_{i+1} and enter *reader authentication* phase.

For *reader authentication*, it has secret S_i, P_i and other public parameters which has been derived from previous $(i-1)$ successful authentication sessions.

- Reader: First test whether *hamming weight* of s' is exactly l. Then generate a non singular binary matrix Q to update session key S_{i+1} as $[Q \cdot S_i]$ and compute pseudo inverse-matrix S^{+}_{i+1}, and P_{i+1} as $[S_{i+1} \cdot S^{+}_{i+1}]$. To send the new session key S_{i+1} to the tag and blinding the matrix Q, P'_i is computed by $[P_i \cdot Q]$ which is actually equivalent to a binary matrix $[S_i S_i^{+} Q]$. Assume the adversary cannot reveal S_i from P'_i in polynomial time. Next, for reader authentication, generate an n-bit noise vector e' and compute multiple bit LPN problem as $\mathbf{r}' := [\mathbf{S_i}]^{\mathbf{T}} \cdot (T_{id\downarrow}s') \oplus \mathbf{e}'$. Finally answer the tag with string $(\mathbf{P_i}', \mathbf{r}')$.
- Tag: Check the *hamming weight* of $([\mathbf{S_i}]^{\mathbf{T}} \cdot (T_{id\downarrow}s') \oplus \mathbf{r}') \leq n \cdot \tau'$ where $(n \cdot \tau')$ is the pre-defined accepted threshold value for the LPN problem. If this check passes, accept the reader and update session key $\mathbf{S_{i+1}}$ by $[(\mathbf{P_i}' \cdot \mathbf{S_i}) = (\mathbf{S_i S_i^{+} Q} \cdot \mathbf{S_i}) = (\mathbf{S_i Q})]$ where $[\mathbf{S_i S_i^{+} S_i} = \mathbf{S_i}]$[1]. However, if the check fails, tag's session key remains unchanged.

Note that, in the protocol, session key generated by the reader is used by the tag. To be precise, session key S_{i+1} is generated from the former key S_i and random matrix $[Q]$. Sending S_{i+1} as plain text is not secure since $[S_{i+1}]$ will act as the next session key between the tag and the reader. Therefore, random matrices $[S_{i+1}]$ is sent with encryption to the tag. We first use $[Q]$ for randomizing S_i and then pseudo-random matrix computation for blinding the matrix $[S_i]$. However, a tag's session key is updated each time period i by computing S_{i+1} from simple decryption using pseudo-inverse matrix properties. More precisely, tag's session key is not updated until a successful reader authentication.

Hash-table lookup: An appropriate lookup hash-function can offer efficient database searching. In our protocol, *index* is updated in both the tag and the reader, as the transaction becomes successful. This demands an efficient hash-table that provide $O(1)$ query, insertion and deletion operations at high loads[2]. We suggest segmented hash table architecture described in [131], that provides high collision resistance and comparatively low search cost in *worst case* performance. A traditional hash table maps the key e.g., *index* into a single hash bucket, whereas N-segmented hash table maps into N potential buckets. Therefore, a table with capacity m has equally sized logical segments containing m/N buckets. Here the hash function is defined as $\mathcal{H} : I \rightarrow \{0, 1, \ldots, m/N - 1\}$ where

[1] From the properties of pseudo-inverse matrix.

[2] To provide scalability.

I is the index space of size n. Let *Linear chaining* be used as *searching* technique, then average and worse search time will be $\Theta(1+\alpha)$ and $\Theta(logn/loglogn)$ respectively, where $\alpha = n/m$. To ensure $O(1)$ searches, they utilize N-independent bloom filter[3] to achieve low false positive rates.

5.1.4 Security Analysis

5.1.5 SLPN problem

We use a proof method similar to that described in [80] as Theorem 1. follows. Even though the protocol in our model and that in [80] are different, a similar proof can be used as both are based on the $SLPN^*$ problem. The hardness of $SLPN^*$ can be defined using an indistinguishability game. More formally, the security of the proof is based on the computational indistinguishability of the two oracles $SLPN^*$ and uniform distribution U_{2l}. From the protocol description, it can be found that noise is a vector rather than a single bit; and the secret is not a vector but a pseudo-random matrix.

Theorem 9 *For any constant $\gamma > 0$, let $d = l/(2+\gamma)$. If the $SLPN^*(s,\cdot)$ problem is (t, nQ, ϵ)-hard, then the authentication protocol from Figure 1, is (t', Q, ϵ')-secure against active adversaries, where the constants $(c_\gamma, c_\tau > 0)$ depend only on γ and τ respectively.*

$$t' = t - poly(Q,l) \quad \epsilon' = \epsilon + Q.2^{-c_\gamma.l} + 2^{-c_\tau.n} = \epsilon + 2^{-\theta(n)}$$

The protocol has completeness error $2^{-c_\tau.n}$ where $c_\tau > 0$.

Theorem 10 *Let an oracle be $\mathcal{O}$ which is either an $SLPN^*(s,\cdot)$ oracle or $U_{2l}(\cdot)$. Let $\mathcal{B}$ be a simulator that uses (t, Q, ϵ)-adversary $\mathcal{A}$ such that:*

$$Pr|\mathcal{B}^{SLPN^*(s,\cdot)} = 1| \geq \epsilon - Q.\alpha'_{l,d} \quad \textit{and} \quad Pr|\mathcal{B}^{U_{2l}(\cdot)} = 1| \leq \alpha''_{\tau',n}$$

where $\alpha'_{l,n} \leftarrow Pr|(w(l) < w(d))| \leq 2^{-c_\gamma \cdot l}$ and
$\alpha''_{\tau',n} \leftarrow Pr|(w(r) \leq n \cdot \tau' : r \in_R \mathbb{Z}_2^n)| \leq 2^{-c_\gamma \cdot n}$
Therefore, $\mathcal{B}$ can distinguish between two oracles $SLPN^*(s,\cdot)$ and $U_{2l}(\cdot)$ with advantage $\epsilon - Q \cdot \alpha'_{l,d} - \alpha''_{\tau',n}$. Now we can upper bound the gap between two probability that $\mathcal{B}$ outputs:

$$|Pr[\mathcal{B}^{SLPN^*(s,\cdot)} = 1] - Pr[\mathcal{B}^{U_{2l}(\cdot)} = 1]| \leq Q.\alpha'_{l,d}$$

This implies the probability of success of the simulator $\mathcal{B}$, and hence the adversary $\mathcal{A}$, in the indistinguishability game.

Interested readers are referred to [80], for further clarification and proof of the theorem.

[3] An on-chip predictive filter that supports space-efficient membership queries.

5.1.6 Man-in-the Middle Attack

The most sophisticated and realistic attack in an RFID system is the Man-in-the Middle (MIM) attack. Our protocol is MIM-secure against an active attack from the SLPN assumption. Note that, first the reader authenticates the tag, and then vice versa. In case of tag authentication, it runs a two-round MIM-secure authentication protocol where the *reader* chooses a random variable as challenge, and *tag* returns the response according to the challenge. The authentication tag $\gamma = (S, r : S^T f_k(s) \oplus e)$, where $f_k(s)$ is the secret key derivation function which uniquely encodes challenge s according to k by selecting l bits from the key[4] k. The main technical difficulty to build a secure MIM-free authentication from LPN is to make sure the secret key k does not leak from verification queries. In [80], they use randomize- mapping function $f_k(s) = (k \downarrow s : \mathbb{Z}_2^{2l} \to \mathbb{Z}_2^{l})$ for some random s and prove that if LPN is hard, then the construction is MIM-secure. We have twisted a little the original idea. In our construction, we remain both S and k secret, that enhances security. We use an EX-OR operation for hiding s' using T_{id} as key. Note that, the XOR cipher is vulnerable to frequency analysis; therefore, even if the adversary compromises T_{id}, it cannot generate S_i for any subsequent sessions using only T_{id}. In the third phase of the protocol, we introduce a pseudo-random matrix as blinding factor to transfer the new session key S_{i+1}, which is secure from the pseudo-random matrix property assumption.

5.1.7 Pseudo-random matrix

We followed the security analysis in [97], where it is claimed that, having known the messages $XX^+Q \in \mathbb{Z}_2^{l\times l}$, it is impossible to recover the secrets $X \in \mathbb{Z}_2^{l\times n}$, or $Q \in \mathbb{Z}_2^{l\times l}$. Given $XX^+Q \in \mathbb{Z}_2^{l\times l}$, suppose that $rank(X) = r$, and

$$X^+X = \begin{pmatrix} I^{r\times r} & 0 \\ 0 & 0 \end{pmatrix} \Rightarrow X^+XQ = \begin{pmatrix} Q^{r\times r} & 0 \\ 0 & 0 \end{pmatrix}$$

where $I^{r\times r}$ is an *Identity matrix* and $Q^{r\times r}$ is the left upper sub-matrix of Q. Then the probability that an adversary determines the correct Q is $2^{-(l-r)n}$. To ensure security, we need to ensure that $l \gg r$, which can be obtained with $l > n$. In our authentication protocol, we let $n \leq l/2$ to ensure a large value of l. **Forward Security:** For each operation, the tag uses session key S_i and the reader also uses its corresponding P_i for verification of authentication tags. At the end of each valid session, (S_i, P_i) is updated with the random matrix and the previous key is deleted permanently in the tag. We say that, even if S_i is exposed by the attacker during the authentication session i, the tag's privacy is fully guaranteed for $(i-1)$ periods.

Backward Security: Typical RFID tags and their reader communicate only for a short period of time because of the power constraint of a tag. Thus, either we restrict the

[4] We use T_{id} as the secret key k

adversary in such a way that it can obtain neither T_{id} nor S_i at any time instance i, or there should exist some non-empty gap between the time of a reveal query and the attack, while the tag is not accessible by the adversary. This entails the adversary miss the protocol transcripts needed to update the compromised secret key and hence our protocol claims **reduced** backward security.

Tracking a tag: Protocol can resist tracking the tag due to the following reason: it refreshes the random vector (s, s', e, e'), updates the keys (P_i, S_i) while assumptions like the SLPN problem, the pseudo-random matrix makes the protocol indistinguishable from the adversarial perspective.

De-synchronization attack: We introduce indexing of the tag to get rid of the attack. When the reader and the tag maintain synchronization, searching hash table becomes very fast with *direct match* technique. However, synchronization attack may take place in the third protocol transcript from the reader to the tag; while the tag may not receive (p', r') to update its shared key. In the later case, brute-force search will be used for successful authentication. Although it yields worse performance, but after successful authentication synchronization would be recovered. **Tag cloning:** We use two different keys T_{id} and S_i for the tag. Therefore, even if the tag is cloned by a malicious reader, we assume either of the keys is not compromised. For instance, an EPC generation 2 allows a password-enabled *secure state* configuration that prevents anyone from reading or writing onto a tag memory bank. Let T_{id} be stored in a password-protected memory bank. Moreover, the tag is not allowed to update the key S_i until it authenticates the reader. This verification thwarts the cloning attack as well. **Replay Attack:** Assuming that the random challenges sent by the reader and the tag are the same in two different sessions, an adversary can launch a *replay attack* by snooping the random numbers; but in our protocol, the reader queries the tag each time with a new random challenge s, and then the tag queries the reader with random s', I_i. So, it is very unlikely to find a match between a pair of (I_i, t, r) from two different sessions of the same tag.

5.1.8 Privacy

We define oracles according to the following:

- **CTag**$(ID) \rightarrow T_{id}$: On input of a tag identifier, this oracle registers the new tag to the reader/server and return a reference T_{id} to resist duplicate IDs.

- **Launch()**$\rightarrow \pi, s$: This oracle launches a new protocol by returning a session identifier π and first transcript s by the reader to ensure reader-initiated protocol.

- **DTag** $(T_i, T_j)_b \rightarrow vtag$: On input of a tag reference (T_i, T_j), this oracle generates a virtual tag reference $vtag$ and stores the triple $(vtag, T_i, T_j)$ in a table D, provided that none of the (T_i, T_j) are already referenced in the table. Depending on the value of the random bit b by the challenger, $vtag$ either refers to T_i or T_j.

- **Free**$(vtag)_b$: On input of $vtag, b$, it erases the volatile memory of the tag T_i $(b = 0)$ or $T_j(b = 1)$ and removes the entry $(vtag, T_i, T_j)$ from D.

- **SendTag** $(vtag, s)_b \rightarrow t'$: On input of $vtag$, this oracle sends s to either $T_i(b = 0)$ or $T_j(b = 1)$. It returns the reply t' of the tag or $\perp$.

- **UKey**$(S_i) \rightarrow S_{i+1}$: A tag key update oracle performed on the tag side which takes S_i as input and outputs an updated key S_{i+1}.

- **SReader**$(\pi, s') \rightarrow s''$: On input of (π, s'), this oracle sends s' to the reader in session π and returns the reply s'' of the reader or $\perp$.

- **Result** (π): This oracle returns either 1 or 0 on successful authentication of a tag. But If the session π is not finished, or there exists no session π it returns $\perp$.

- **Corrupt** (T_i): On input of T_i, this oracle returns the non-volatile internal state of T_i. Note that, corruption is done w.r.t. tag, not the $vtag$. Therefore, the adversary is forced to corrupt tags T_i that are currently not drawn.

First, we analyze our protocol using the *privacy model* in [102]. where challenger runs the $\mathbf{Exp}^b_{\mathcal{A}}(S)$ experiments with the above oracles.

- $b \in_R \{0, 1\}$

- SetupReader (1^λ)

- $b' \leftarrow \mathcal{A}^{CTag,Launch,DTag,Free,STag,SReader,Result}()$

- return $(b' == b)$

We assume that $\mathcal{A}$ queries the challenger with $\mathbf{Exp}^b_{\mathcal{A}}(S)$ experiments a number of times and hence guess bit b' and wins the privacy game if and only if $(b' == b)$ The advantage of the adversary to win is defined as

$$\mathbf{Adv}^b_{\mathcal{A}}(k) = |Pr[\mathbf{Exp}^0_{\mathcal{A}}(k)] + Pr[\mathbf{Exp}^1_{\mathcal{A}}(k)] - 1|$$

The reader sends out a random vector s and the tag computes the protocol transcript from the challenge s, combined with shared key k_i and $(e, [R])$. The reader decrypt the tag's reply and verify whether it gets right e under the shred key k in the database. In the second phase, it encrypts the random matrix $[Q]$ with the session key P_i and computes the protocol transcript from the challenge vector s' sent from the tag under the shared secret key k_i. Tag can decrypt the matrix $[Q]$ with session key S_i and verify e' under the shared secret key K_i and MAC value s''.

Theorem 11 *If the encoding in the proposed protocol is indistinguishable then the protocol is strong private for narrow adversaries.*

Proof: We analyze our protocol using the *privacy model* in [102]. Given an adversary $\mathcal{A}$ that wins the privacy game with non-negligible advantage, we consider another adversary $\mathcal{B}$ that can break the *indistinguishability* game with non-negligible advantage. The adversary $\mathcal{B}$ runs the adversary $\mathcal{A}$ to answer queries with the following exceptions:

- S, T_{id} are two different keys of the indistinguishability game.

- *SendTag* $(vtag, s)_b$: By retrieving the tag T_i and T_j references from the table D using virtual tag *vtag*; it generates two references $m_0 = \mathbf{w}([\mathbf{S_i}]^{\mathbf{T}}.(T_{i\downarrow}s) \oplus \mathbf{r}) > n.\tau'$ and $m_1 = \mathbf{w}([\mathbf{S_j}]^{\mathbf{T}}.(T_{j\downarrow}s) \oplus \mathbf{r}) > n.\tau'$. The references m_0, m_1 are sent to the indistinguishability oracle of SLPN problem, which returns whether the *hamming weight* satisfies $w \leq n.\tau'$ under one of the references .

- $\mathcal{B}$ cannot query for *Result*() oracle.

At the end of the game, $\mathcal{B}$ outputs according to $\mathcal{A}$'s guess. Hence, $\mathcal{B}$ is perfectly simulated for $\mathcal{A}$. If $\mathcal{A}$ breaks privacy, then $\mathcal{B}$ wins the indistinguishability game; but indistinguishability with only one call to the oracle is equivalent to indistinguishibility with multiple calls to the oracle that proves the *narrow privacy* of the protocol. □

In [103], the authors have categorized RFID authentication protocols into *four* types according to their constructions and distinguished *eight* privacy levels by their natures on accessing *Corrupt*() oracle in the strategies of the adversary and whether *Result*() oracle is used or not.

- *Nil*: No privacy protection at all.

- *Weak*: Adversary has access to all oracles except *Corrupt* (T_i).

- *Forward* : Adversary has access to *Corrupt*(T_i) but other oracles are not allowed as *Corrupt*(T_i) oracles are accessed.

- *Destructive* : No restriction on accessing other oracles after *Corrupt* (T_i), but T_i is not allowed to use again.

- *Strong* : It is the strongest defined privacy level with no restrictions.

Each of these levels has its *narrow* counterpart to restrict the access of *Result*() oracle. Our protocol belongs to *Type 2a* for construction where the shared key S_i has been updated just after the reader is authenticated. We now redefine our protocol privacy according to the model described in [103].

Without reader authentication, any adversary can keep querying a tag with any compatible reader until it is desynchronized with a legitimate reader. Therefore, the tag's secret can only be desynchronized by one update. As the reader has both the keys S_i and S_{i-1}, in case of tag failure to update its shared key S_i, the reader can still try to authenticate the victim using the previous key S_{i-1} in the next protocol conversation. Thus, it provides *weak* privacy to the protocol construction. Let an adversary $\mathcal{A}$ try to send authentication transcripts to the tag by blocking a valid reader authentication message, or by intercepting of the tag in an online attack. This causes the tag to be in a DoS attack or in a deadlock condition, as it cannot update the key without reader authentication.

We can reduce the protocol to *narrow-forward* privacy level by two ways. Firstly, by *Reduced Backward security*, where we restrict the adversary in such a way that there should exist some non-empty gap between the time of a reveal query and the attack, while tag is not accessible by the adversary; which means the adversary misses the protocol transcripts needed to update the compromised secret key [132]. Secondly, note that **Corrupt**$(\cdot)$ oracle operates w.r.t. *a tag* not with a *virtual tag vtag*, which means adversary is forced to corrupt tags T_i that are currently not drawn. Therefore, after single **Corrupt**$(\cdot)$ oracle, henceforth adversary is allowed to use **DrawTag**$(\cdot,\cdot)$ oracle. Of course, here adversary is not allowed to access **Result**$(\cdot)$ oracle.

Theorem 12 *Considering aforementioned assumptions (Reduced Backward security or disallowing Result*$(\cdot)$ *oracle), our protocol is semi-forward narrow privacy preserved.* □

Table 5.1: Tag Resources and Security Comparison with HB family and Others

Scheme	**Storage**	**Computation**	**Authentication**	**Security achieved**	**Hardware**
		(major)			(gates)
[80]	S	1 LPN	tag	0,4	$\approx$ 1600
HB$^+$ [90]	2 S	2 LPN	tag	7	$\approx$ 1600
HB-MP[92]	2 S	1 LPN	tag	0, 5,6,7	$\approx$ 1600
HB-MP$^+$ [108]	2 S	1 LPN,1 HASH	tag	1,5,6,7	$\approx$ 3500
F-HB [104]	1I , 1 S	1 PRNG,2 LPN	mutual	$1, 2, 4^*, 5, 6, 7$	$\approx$ 3500
ours	**1 *I*, 1 S**	**1 LPN,1 PIM**	**mutual**	1,2,3,4*,4†,5,6,7,8	$\approx$ **1600**
[107]	1 S	1 PRF,1 HASH	tag	2,4,6,8	$\approx$ 6000
[105]	1I ,1 S	1 PRF	mutual	2,4*,6,8	$\approx$ 6000
[133]	1 S	1 PRNG,1 UH	tag	2,4	$\approx$ 3500
[106]	1 S	1 SC	mutual	2,4*,8	$\approx$ 2000
[109]	1 S ,2 TS	1 HASH	tag	4*	$\approx$ 1500
[110]	1 S , 1 TS, 1 RN	2 HASH	mutual	$4^*, 8$	$\approx$ 1500
[111]	1 RN,1 C, 1 TS,1 S	3 HASH	mutual	$2, 4^*, 6, 8$	$\approx$ 1500

where SC:= Stream Cipher; S:= Secret key; C:= Counter; I:= Index;
PRNG:= Psudo Random Number Generator; UH:= Universal Hash; PIM:= Pseudo Inverse Matrix; LPN:= Learning parity from noise TS:= Time Stamp; RN:= random number;
Security attributes: Active attack (0), Man-in-the-Middle attack(1), Forward Security (2), Reduced Backward Security (3), IND-Privacy (4), UNP-Privacy (4*), Strong-private (4†) Tag tracking (5), De-synchronization (6), Tag Cloning (7), Replay attack (8).

5.1.9 Comparison and Performance analysis

In order to support dynamic scalability, the proposed protocol requires to search and store the *lookup hash table* for each transaction, based on the index value in on-line, to retrieve the corresponding data in the hash-table. However, the data can be pre-computed in the hash-table either in off-line or dynamically in online.

In case of the tag, protocol operations include *two* random binary vector generation, *one* SLPN problem, *one* EX-OR operation, and *three* binary linear matrix multiplications. For computation, we only consider the SLPN problem and assume the rest of the operations (e.g., calculation hamming weight) to be trivial in terms of computational complexity. The protocol is roughly as efficient as the HB$^+$ protocol with just twice the key length. Since it is a reduction of the LPN to the SLPN problem, the protocol is secure against quantum adversaries, assuming LPN is secure against such adversaries. There is a natural trade-off between the communication cost and key size. For any constant c $(1 \leq c \leq n)$, the communication cost can be reduced by a factor of c by increasing the key size with the same factor.

Major computations of the proposed authentication scheme on the tag include linear binary matrix multiplication and the LPN problem. And, in case of storage, only a secret key and an index for the key. As bitwise XOR, matrix multiplication, the hamming weight $w(\cdot)$ and $(a_{\downarrow}b)$ are all binary operation, they can easily be implemented using bit-by-bit serialization to save hardware gates. In the e-STREAM project, the PRNG operation needs only 1,294 gates to achieve 80-bit security level using Grain-v1[87]. A PRNG requires a linear feedback shift register (LFSR) structure to compute, so LPN problem can share the same LFSR. s' can be deduced from the state variable of PRNG. The cost of a LPN problem and of storing the index and secret key may not be greater than that of a PRNG, and should be less than that of a CRC as well. However, the LPN problem can be implemented using an LFSR (for Transpose matrix), a 1-bit multiplier plus 1-bit accumulator (for binary multiplication), XOR gates (for $\oplus$ operation), 1-bit counter(for hamming weight) and a 1-bit comparator (for $a_{\downarrow}b$ operation). Thus, to achieve a λ-bit security level, the overall hardware cost of the proposed protocol for the above mentioned functions on a tag is no more than 1600 gates, including the cost of non-volatile memory to store the secret key, the index value and protocol intermediate values; and the protocol is suitable for Class-1 Generation-2 EPC tags, where PRNG and CRC are used as hardware.

In Table 2, we show a comparative study on some general attributes e.g., storage consumption, major computations, authentication party, achieved security, approximate hardware cost etc., between our protocol and several HB-like and non-HB protocols. It appears that, although the tag's hardware cost of the proposed protocol is optimal, it achieves most common security requirements. Additionally, it achieves $O(1)$ time complexity during the synchronized state that resists brute-force searching in each authentication session. Alternatively, hardware cost of the reader is expensive for the purpose of complex computing[5], that results in reduced computing in tag and hence hardware cost. Besides that, the hash-indexed searching technique at the reader, where all the data related to certain tags are stored efficiently as *index*, reduces an exhaustive database search at the reader end. As a consequence, in an RFID system with *remote authentication*[6], reader can use this *index* in *batch mode* operation to aggregate responses from several tags together, that reduces the communication cost between the reader and the server, where each tag contains unique index within the reader's *field of view* at a specific time instance.

[5] Searching the database and generating a pseudo-random matrix are the most complex part of the protocol

[6] Tag readers are portable and server access is costly

5.1.10 Conclusion

This work presents a novel hardware-friendly RFID authentication protocol based on the SLPN problem that can meet the hardware constraints of the EPC Class-1 generation-2 tags. In comparison to other protocols as described in Table 2, it requires less hardware and has achieved major security attributes. The protocol is also compliant to *semi forward for narrow adversaries* privacy settings. Moreover, scalability of the protocol can be realized best in synchronized and desynchronized modes that ensures infinite DoS resistance. Security and privacy can be protected as long as we allow an adversary not to cope with both tag ID and the secret key simultaneously. In addition, the security and privacy proof follows the standard model that uses indistinguishability as basic privacy notion. Our future research will focus on how to reduce the communication cost between the reader and server, assuming the wireless link between them is insecure, to figure a realistic privacy-preserving RFID environment.

5.2 An RFID authentication protocol where reader-server channel is insecure

5.2.1 Introduction

Nowadays, RFID is not just a futuristic vision but rather a technology that is being deployed successfully in applications ranging from aviation systems, smart homes, and public health to supply chain management. For example, airlines are preparing a switch to RFID solutions for improving passenger baggage processes at their main hubs, RFID based ecosystem is evolving as an example of ubiquitous information technology, low-level RFID data is transformed into meaningful, high-level information, and thus, is used in the Internet of Things (IoT) [78]. Therefore, the rise of RFID-based applications in the last decade has brought about major attention on RFID data security and privacy settings .

We followed a very simple, efficient and perfectly binding string commitment scheme with an *exact* version of the LPN-problem, whose security is based on the hardness of the LPN problem [117]. Unlike other HB-protocols, our protocol follows the exact LPN based commitment scheme for authentication, the secret keys are *binary matrix* and pair of secret keys shared between entities are different. In order to update session key and to verify protocol transcripts, we introduce pseudo-inverse matrix properties and randomized Hill cipher techniques. This makes the proposed protocol more robust against quantum adversaries while being efficient like the previous HB-protocol family. In addition, our privacy notion captures the privacy of both tag-reader and reader-server transactions. Hence, compared to existing tag-reader based mutual authentication protocols, it is more practical, rigorous, powerful and concise.

Our contribution. In this chapter, we propose a new variant of RFID authentication system from an exact LPN problem, that can provably withstand all known attacks. In addition, unlike other traditional authentication protocols for RFID systems, all communications between a server and a reader are assumed to be insecure and over inauthentic channel. Therefore, reader and server are not identical but two individual entities. More precisely, we use an identical scheme to authenticate all the entities (Tag, Reader, Server) together in an RFID system. The main objective of our scheme is to improve the security scope of a recently proposed variant of HB-protocol in [104] and [113] by adding some non-linear components without increasing its complexity significantly. Unlike authentication scheme described in [104][113], we adopt several new ideas for construction such as:

- Only a server is considered to be *fully trusted* and *keys* are shared among the entities accordingly.
- Use the commitment scheme from exact LPN, in compare to the decisional-LPN problem in [104] and subspace-LPN problem in [113] in order to remove completeness/correctness error.
- More properties of pseudo-inverse matrix, such as signature-like light authentication in the reader-tag transaction.
- A variant of Hill cipher in the reader-server communication.

To the best of our knowledge, we propose the first HB-like authentication protocol for RFID system that is *fully secure*[7], private and scalable. Moreover, the protocol supports forward privacy under zero-knowledge (ZK) indistinguishable notion and also provides all security proof under standard model. Consequently, the protocol could be realized through several RFID security applications in the real life environment like *authorization recovery, ownership transfer, controlled delegation* etc. [115, 116].

Assumption: RFID system in this work consists of a single legitimate server, a set of readers and a set of tags (EPC global Class 1 generation 2). Readers are connected to the back-end server that stores all the data related to the tags and their corresponding readers in the database. Each tag has its unique identification T_{id}, a permanent key $\mathbf{S}'$ and a session key $\mathbf{S}''$. However, T_{id} is used as the shared secret among all the 3 parties while $\mathbf{S} \leftarrow \mathbf{S}' \| \mathbf{S}''$ is shared only between the tag and the server.

We refer to the computational hiding property of the commitment scheme described in [117] that is polynomially equivalent to the security of the well known LPN problem. Note that the hardness of *exact* LPN lies under the hardness of the traditional LPN problem. This assumption distinguishes *noisy* linear equations from uniformly random.

Our protocol borrows some basic ideas from Hill cipher in [118] that is computationally hard under matrix multiplication with random permutations. We use pseudo-inverse matrix in order to transfer session key from the server to the tag and to offer The most widely known and popular pseudo-inverse is the Moore-Penrose pseudo-inverse, which was described by E. H. Moore [96].

Since an RFID tag is not tamper-resistant, its session key $\mathbf{S}''_i$ is refreshed after each i^{th} session completes successfully. To update the key, each tag authenticates not only its licit reader but also the legitimate server. In addition, we assume the tag identifier T_{id} be unique and secure within an RFID system. However, an adversary cannot corrupt the reader and the tag until it compromises their secrets P and (T_{id}, $\mathbf{S}$) respectively at a time.

Nevertheless, if all the secret keys are exposed at a time, the adversary can trace the tag for a period i until the next authentication cycle starts. To avoid exhaustive database search at the server hash-index I_i is used. Database at the server associates the tag index with other tag related data e.g., T_{id}, $\mathbf{S}_i$, P_i etc.

5.2.2 Construction

Our construction is based on the LPN-based commitment scheme [117], but customized to work with the authentication protocol. We adopt three different cryptographic tools: LPN based commitment scheme, pseudo inverse matrix properties and a secure variant of Hill cipher in order to achieve 3-round mutual authentication protocol described in Fig. 1. We use the term *fully-secure*, because the protocol attains mutual authentication not only in Tag-Reader pair, but also in Reader-Server. The protocol is partitioned/organized into a hierarchy of computation units. Therefore, it sets aside significantly less computations to the tag. On the other hand, the most expensive computations of the protocol are handled by the server. We use only random vector generation, bitwise XOR and matrix multiplication as tag operation. The protocol uses (τ, k, l, v, τ') as public parameters,

[7] Where both the channels: tag/reader and reader/server are assumed to be insecure.

where $l \in \mathbb{N}$ is the length of tag identifier, $v \in \mathbb{N}$ ($v \leq l$) is the length of commitment message, $k \in \mathbb{N}$ ($k = (l + v)$) is the length of secret key. Note that (τ, τ') are constant while (l, k, v) depend on the security parameter. In the setup phase, Server generates the initial index I_0, the permanent key $\mathbf{S}'$, the session key $\mathbf{S}''_0$ and its corresponding $\mathrm{P}_0 \leftarrow \mathbf{S}''_0\mathbf{S}''^{+}_0$[8] and other public parameters; and set them into a tag non-volatile memory and into the reader. Note that, we use different secret keys for entities. For instance, T_{id} is shared among three entities of the protocol. In contrast, each tag has 2 secrets ($\mathbf{S}'$,$\mathbf{S}''$) and each reader has 1 secret (P) respectively to share with the server. However, for any time instance i a tuple $[\mathrm{I}_i, \mathrm{T}_{id}, \mathbf{S}'_{i-1}, \mathbf{S}', \mathbf{S}''_i, \mathrm{P}_{i-1}, \mathrm{P}_i, r_i]$ needs to be stored in the back-end database of the server while a reader needs to memorize $[\mathrm{P}_{i-1}, \mathrm{P}_i, \mathrm{P}_i^{-1}]$.

Table 5.2: Tag-Reader Communication (Step-1)

Reader $(\mathbf{P_i}, \mathbf{T_{id}})$	**Tag** $(\mathbf{I_i}, \mathbf{T_{id}}, \mathbf{S_i}', \mathbf{S_i}'')$
$s \in_R \mathbb{Z}_2^v$; s.t. $\mathbf{w}(s) = v/2$	
$\xrightarrow{s}$	
	If $\mathbf{w}(s) \neq v/2$ **return**; $e \in_R \mathbf{Ber}_{k\tau}^k$ $\mathbf{S_i} = \mathbf{S_i}' \parallel \mathbf{S_i}'' \in \mathbb{Z}_2^{k \times (l+v)}$ $r := \mathbf{S_i} \cdot (\mathbf{T_{id}} \parallel s) \oplus e$ $\sigma_\mathbf{i} = \mathbf{S_i}'' \cdot s$ $s' \in_R \mathbb{Z}_2^v$ s.t., $\mathbf{w}(s') = v/2$ $\xleftarrow{(\mathbf{I_i}, r, \sigma_i, s')}$

For *tag* authentication, a tag holds $\mathbf{S}''_i$ and I_i that have been derived from the previous $(i - 1)$ successful sessions.

- Reader: Generate a random binary v-bit challenge string s, and sends it to a tag.
- Tag: Check the *hamming weight* of the string s and generate a k-bit noise vector e from Bernoulli distribution $\mathsf{Ber}_{k\tau}$, a random v-bit challenge string s' with hamming weight $v/2$. Next a k-bit commitment string r on the message s is generated as $r := \mathbf{S}_i \cdot (\mathrm{T}_{id} \parallel s) \oplus e$. Note that $\mathbf{S}_i$ consists of 2 keys: the permanent key $\mathbf{S}'_i$ and the session key $\mathbf{S}''_i$.

  In addition, σ_i is generated by using the key $\mathbf{S}''_i$ to demonstrate the authenticity of the message s and to give an impression to the reader that it was created by its known tag. In order to extinguish brute-force searching at the server end, an index I_i is maintained and updated each time ($\mathrm{I}_{i+1} \leftarrow r$) by the tag. Finally, the tag forwards $(\mathrm{I}_i, r, \sigma_i, s')$ to the reader.
- Reader: The reader conveys the messages it received from the tag. But before forwarding, it apparently verifies the tag with σ_i, whether it is generated from the

[8] $\mathbf{S}''^{+}_0$ is the pseudo-inverse of the matrix $\mathbf{S}''_0$ by following the algorithm in [88]

challenge s. Note that $\mathrm{P}_i\sigma_i = \mathbf{S}_i''\mathbf{S}_i''^{+}\mathbf{S}_i''s = \mathbf{S}_i''s = \sigma_i$. Subsequently, it also checks the hamming weight of s'.

- Server: First search the database with I_i in order to find out a tuple $[\mathrm{I}_i, \mathrm{T}_{id}, \mathbf{S}_i'', r_{i-1}, \mathbf{S}_{i-1}'']$. Note that $(r_{i-1}, \mathbf{S}_{i-1}'')$ would be stored to resist synchronization attack. However, searching with index I_i might fail sometimes e.g., due to synchronization attack etc. In that case, server could apply brute-force searching method[9] targeting to explore the previous transaction parameters: $(\mathbf{S}_{i-1}'', r_{i-1})$. Then, given a commitment r on a message s sent by the reader, it accepts the commitment if and only if: $\mathbf{w}(\mathbf{S}_i \cdot (\mathrm{T}_{id} \parallel s) \oplus r) \overset{?}{=} \lfloor k\tau' \rceil$ and $\mathbf{w}(s') \overset{?}{=} v/2$ where s' is the new challenge (commitment) message for the server. Consequently, it accepts the tag, update the index to I_{i+1} and enter *server/reader* authentication phase.

[9] Server can search $[\mathrm{I}_i \overset{?}{=} r_{i-1}]$ the database with previous index stored for $(i-1)^{th}$ session.

Table 5.3: Reader-Server Communication (Step-2)

Server $(\mathbf{I_i}, \mathbf{T_{id}}, \mathbf{S_i}', \mathbf{S_i}'', \mathbf{P_i})$	**Reader** $(\mathbf{P_i}, \mathbf{T_{id}})$
	If $(\mathbf{P_i} \cdot \sigma_i \neq \sigma_i \bigvee \mathbf{w}(s') \neq v/2)$
	return;
$\xleftarrow{(\mathbf{I_i}, r, s, s')}$	
Lookup $\mathbf{T_{id}}$ by using I_i:	
Direct match: **If** $(\mathbf{I} \neq \mathbf{I_i})$ then	
Brute-force search:	
$\exists$ $(\mathbf{T_{id}}, \mathbf{S_i}''$ or $\mathbf{S''_{i-1}})$ that satisfies:	
$\mathbf{S_i} = \mathbf{S_i}' \parallel \mathbf{S_i}''$	
If $\mathbf{w}\big((\mathbf{S_i} \cdot (\mathbf{T_{id}} \parallel s) \oplus r) \neq k \cdot \tau'$	
$\bigvee \mathbf{w}(s') \neq v/2\big)$	
return;	
$\mathbf{I_{i+1}} = r$	
Generate non-singular $\mathbf{Q} \in_R \mathbb{Z}_2^{k \times k}$	
$\mathbf{S''_{i+1}} = \mathbf{Q} \cdot \mathbf{S_i}'' \in \mathbb{Z}_2^{k \times v}$	
where $rank(\mathbf{S''_{i+1}}) = v$	
$\mathbf{S''_{i+1}}^{+} := (\mathbf{S''_{i+1}}^{\mathbf{T}} \mathbf{S''_{i+1}})^{-\mathbf{1}} \mathbf{S''_{i+1}}^{\mathbf{T}} \in \mathbb{Z}_2^{v \times k}$	
$\mathbf{P_{i+1}} := [\mathbf{S''_{i+1}}] \cdot [\mathbf{S''_{i+1}}]^{+} \in \mathbb{Z}_2^{k \times k}$	
$\mathbf{P_i}' := \mathbf{P_i} \cdot \mathbf{Q} \in \mathbb{Z}_2^{k \times k}$	
$e' \in_R \mathbf{Ber}^{k}_{k\tau}$;	
$r' := \mathbf{S_i} \cdot (\mathbf{T_{id}} \parallel s') \oplus e'$	
$\mathbf{P''_i} := \mathbf{Q^{-1}} \cdot \mathbf{P_{i+1}} \in \mathbb{Z}_2^{k \times k}$	
$s'' := \mathbf{P''_i} \cdot (\mathbf{T_{id}} \parallel s') \oplus e'$	
$\xrightarrow{(\mathbf{P_i}', \mathbf{P''_i}, \mathbf{r}', s'')}$	

For *server authentication*, it has secret: $(\mathrm{T}_{id}, \mathbf{S}'_i, \mathbf{S}''_i)$ and $(\mathrm{T}_{id}, \mathrm{P}_i)$ respectively shared with the tag and the reader, where except $(\mathrm{T}_{id}, \mathbf{S}'_i)$, the rest of the parameters would have been derived directly from the previous $(i-1)^{th}$ successful authentication session.

- Server: First generate a non singular binary matrix Q to update session key $\mathbf{S}''_{i+1}$ as $[Q{\cdot}\mathbf{S}''_i]$ for the next $i+1$ session and compute pseudo inverse-matrix $\mathbf{S}''^{+}_{i+1}$, and P_{i+1} as $\mathbf{S}''_{i+1}{\cdot}\mathbf{S}''^{+}_{i+1}$. In order to send the new session key $\mathbf{S}''_{i+1}$ to the tag and blinding the matrix Q, P'_i is computed by $\mathrm{P}_i \cdot Q$ which is actually equivalent to $\mathbf{S}_i\mathbf{S}^{+}_i Q$.

  Subsequently, a k-bit commitment r' on s' will be generated with a view to authenticate server to the tag: $r' := \mathbf{S}_i \cdot (\mathrm{T}_{id} \parallel s') \oplus e'$, where e' is a k-bit randomly generated noise vector.

After this, P_i'' is generated in order to update P_i at the reader, where Q^{-1} is used to randomize P_{i+1}. Finally, compute s'' from P_i'': $s'' := \mathrm{P}_i'' \cdot (\mathrm{T}_{id} \parallel s') \oplus e'$ for authenticating server to the reader. Subsequently, the communication string $(\mathrm{P}_i', \mathrm{P}_i'', r', s'')$ is forwarded to the reader.

- Reader: First check the *hamming weight*: $\mathbf{w}(\mathrm{P}_i'' \cdot (\mathrm{T}_{id} \parallel s') \oplus s'') \overset{?}{=} \lfloor k\tau' \rceil$. It ensures P_i'', consequently, (Q,P_i') to be generated by the server; and hence, server is authenticated. If any of the parameters is replicated during transmission the above equation will not hold. Then the reader updates P_i by using Hill deciphering technique: $\mathrm{P_i'P_i^{-1}P_i'' = P_i\ Q\ P_i^{-1}Q^{-1}P_{i+1} = P_{i+1}}$. Note that $\mathrm{P_i^{-1}}$ can be precomputed and stored in the reader for efficiency.

For *reader authentication*, it has *shared secret* T_{id} to the tag. It is quite certain that the reader would forward the protocol message (P_i', r') to the tag if it could verify the hamming weight equation $\mathbf{w}(\cdot) \overset{?}{=} \lfloor k\tau' \rceil$ successfully.

Table 5.4: Reader-Tag Communication (Step-3)

Reader	**Tag**
$(\mathbf{P_i}, \mathbf{T_{id}})$	$(\mathbf{I_i}, \mathbf{T_{id}}, \mathbf{S_i}', \mathbf{S_i}'')$
If $\mathbf{w}\big(\mathbf{P_i}'' \cdot (\mathbf{T_{id}} \parallel s') \oplus s''\big) \neq k \cdot \tau'$	
return;	
$\mathbf{P_{i+1}} := \mathbf{P_i'} \cdot \mathbf{P_i^{-1}} \cdot \mathbf{P_i''} \in \mathbb{Z}_2^{k \times k}$	
$\xrightarrow{(\mathbf{P_i}', r')}$	
	If $\mathbf{w}(\mathbf{S_i} \cdot (\mathbf{T_{id}} \parallel s') \oplus r') \neq k \cdot \tau'$
	return;
	$\mathbf{S_{i+1}''} = (\mathbf{P_i}' \cdot \mathbf{S_i''}) \in \mathbb{Z}_2^{k \times v}$
	if $rank(\mathbf{S_{i+1}''}) \neq v$
	return;
	$\mathbf{I_{i+1}} = r$

- Tag: Verify the commitment r' on the message s' by checking the *hamming weight* of $(\mathbf{S_i} \cdot (\mathbf{T_{id}} \parallel s') \oplus r')$ is exactly $\lfloor k\tau' \rceil$. If the check passes, accept the reader as well as the server and update the session key to $\mathbf{S_{i+1}''}$ [i.e., $\mathbf{S_{i+1}''} = \mathbf{P_i}' \cdot \mathbf{S_i''} = \mathbf{S_i''S_i''^{+}S_i''Q} = \mathbf{QS_i''}$][10], the session index to $\mathrm{I}_{i+1} = r$. However, if the check fails, the tag's session key remains unchanged.

Note that, in the protocol, session keys are generated and updated at i^{th} instance by the server and later followed by the reader and the tag. To be precise, session key is updated in each transaction of the protocol: inside the tag $\mathbf{S}_{i+1}''$ by randomizing the former key $\mathbf{S}_i''$ with Q, and inside the reader P_{i+1} by secure Hill cipher.

5.2.3 Security Analysis

Commitment Scheme: A commitment scheme should satisfy *three* security properties: correctness, prefect hiding and binding. Our constructing satisfies the following security properties:

[10] From the properties of pseudo-inverse matrix ($AA^{+}A = A$).

- *Correctness*: $\mathsf{Ver}(ck, m, c, d)$ should result to 1 if the inputs are computed by an honest party, such that,

$$\mathsf{Pr}[\mathsf{Ver}(ck, m, c, d) = 1;\ ck \leftarrow \mathsf{KGen}(1^l), m \in \mathcal{M}, (c, d) \leftarrow \mathsf{Com}(m, ck)] = 1$$

- *Computation hiding*: Receiving a commitment c to a message m should give no information to the receiver about m. A commitment c computationally hides the committed message with overwhelming probability over the choice of ck, s.t.,

$$\mathsf{Pr}[ck \leftarrow \mathsf{KGen}(1^l); \forall\ m, m' \in \mathcal{M} \wedge (c, d) \leftarrow \mathsf{Com}(m, ck), (c', d') \leftarrow \mathsf{Com}(m', ck) : c = c'] = 1/2$$

- *Perfect binding*: It means that the *sender* cannot cheat in the second phase and sending a different commitment key ck' causes the commitment to open to a different message m'. That is, with overwhelming probability over the choice of the commitment key $ck \leftarrow \mathsf{KGen}(1^l)$, no commitment c can be opened in two different ways, s.t.,

$$\mathsf{Pr}[(\mathsf{Ver}(ck, m, c, d) = 1) \wedge (\mathsf{Ver}(ck, m', c, d') = 1) : m \neq m'] \leq \epsilon$$

In order to ensure the commitment scheme is *hard* enough, the length of the parameter l should be chosen carefully. Although the length of the challenging messages ($|s| = |s'| = v$) can be chosen arbitrarily, for efficiency reasons it is better to choose the same size as l. In our protocol, we consider $k = v + l$ s.t., $v = l$, where k would be large enough to make the commitment scheme accomplished computationally hiding and perfectly binding with high probability over the choice of secret matrix $\mathbf{S}$. Note that *binding* property is ascertained by large distance of the code generated by the random matrix $\mathbf{S}''$, while the hiding property directly from the LPN assumption that outputs pseudo random string r or r'.

Theorem 13 *Let decisional exact* LPN_x *be hard under* $\tau \in]0, 1/4[, (k, l, v) \in \mathbb{Z}$, *and* $k = \mathcal{O}(l + v)$. *And for any* $\mathbf{S} \in_R \mathbb{Z}_2^{k \times (l+v)}$ *such that,* $\mathbf{w}(\boldsymbol{S} \cdot x) > 2\lfloor k\tau \rceil$, *where* $x \in_R \mathbb{Z}_2^{l+v}$. *Then the commitment scheme used in the protocol is perfectly binding and computationally hiding.*

Proof: Assume $[(\mathbf{T}_i, s_i)$ for $i = 1, 2]$ be two different openings for a commitment r. Then, $e_i = r \oplus \mathbf{S} \cdot (\mathbf{T}_i \parallel s_i)$, and norm of e_i for $i = 1, 2$ is at most $\lfloor k\tau \rceil$. Therefore, $e_1 \oplus e_2 = \mathbf{S} \cdot (\mathbf{T}_1 \parallel s_1 \oplus \mathbf{T}_2 \parallel s_2)$ and $\mathbf{w}(e_1 \oplus e_2) \leq \mathbf{w}(e_1) + \mathbf{w}(e_2) \leq 2\lfloor k\tau \rceil$ which contradicts our initial assumption $\mathbf{w}(S \cdot x) > 2\lfloor k\tau \rceil$, thus, satisfies *perfect binding* property. On the other hand, it would appear that we have

$$r = \mathbf{S}' \cdot \mathbf{T} \oplus e \oplus \mathbf{S}'' \cdot s$$

Since $\mathbf{S}' \cdot \mathbf{T} \oplus e$ is pseudorandom from the exact LPN_x assumption, r is also pseudorandom. Thus, distribution of r is computationally indistinguishable and hence, satisfies *computational hiding* property. □

Theorem 14 *The commitment scheme from LPN is computationally indistinguishable.*

Proof: If a commitment c computationally hides the committed message with overwhelming probability, the distributions of the commitments are computationally indistinguishable. From Theorem 1. we conclude that *decisional exact* LPN_x is perfectly computationally hiding. Let a prover and verifier share a common input y and the prover has a private secret input x. Therefore, for a binary relation $\mathcal{R}$ such that $(x, y) \in \mathcal{R}$. Then For every potentially malicious (Q, t)-adversary $\mathcal{A}$, there exists a PPT simulator V^*, that takes y as an input, but its output is indistinguishable from an honest prover's conversations. In [117], authors describe an efficient simulator for indistinguishability game, where for each challenge c outputs an accepting protocol transcript the distribution of which is computationally indistinguishable from real protocol transactions with an honest prover for challenge c. For more detail clarification, we refer to the respected literature. However, due to the fact that bernoulli random noise might exceed the acceptable threshold, false rejection and false acceptance probability will be:

$$\mathrm{P}_{FA} = \sum_{i=0}^{\tau k} \binom{k}{i}\ 2^k \text{ and } \mathrm{P}_{FR} = \sum_{i=\tau k+1}^{k} \binom{k}{i} \tau^i (1-\tau)^{(k-i)}$$

Pseudo-inverse matrix: We followed the security analysis in [97], where it is claimed that, having known the messages $\mathbf{XX^+Q} \in \mathbb{Z}_2^{k\times k}$, it is impossible to recover the secrets $\mathbf{X} \in \mathbb{Z}_2^{k\times v}$, or $\mathbf{Q} \in \mathbb{Z}_2^{k\times k}$. However, to ascertain security, we need to ensure that $k \gg v$, that can be obtained with $k = \Theta(v + l)$. So, we let $|v| = |l|$ to ensure a large value of k.

A pseudo-invertible matrix X has its unique inverse. If Y, Z be two pseudo-inverse matrices of X, then we have XYX = X and XZX = X. It appears that $(\mathrm{XYX})^{\mathrm{T}} = \mathrm{X}^{\mathrm{T}}\mathrm{Y}^{\mathrm{T}}\mathrm{X}^{\mathrm{T}}$ $= \mathrm{X}^{\mathrm{T}} = \mathrm{X}^{\mathrm{T}}\mathrm{Z}^{\mathrm{T}}\mathrm{X}^{\mathrm{T}} = (\mathrm{XZX})^{\mathrm{T}}$. Similarly, YXY = Y and ZXZ = Z. Since $\mathrm{XY} = (\mathrm{XY})^{\mathrm{T}} =$ $\mathrm{Y}^{\mathrm{T}}\mathrm{X}^{\mathrm{T}} = \mathrm{Y}^{\mathrm{T}}(\mathrm{X}^{\mathrm{T}}\mathrm{Z}^{\mathrm{T}}\mathrm{X}^{\mathrm{T}}) = (\mathrm{XY})^{\mathrm{T}}(\mathrm{XZ})^{\mathrm{T}} =$XYXZ = XZ, YX = ZX. Thus, Y = YXY = ZXY = ZXZ = Z. Hence, pseudo-inverse matrix exists uniquely.

Secure Hill Cipher: The security of the ordinary Hill cipher relies on the rank of Key matrix $rank(K)$. However, Hill cipher succumbs to the most popular *Chosen Plaintext Attack* (CPA) that is in effect a linear transformation on the message space.

Theorem 15 *Hill cipher used in the protocol can resist CPA attack.*

Proof: We use the matrix P_i as the secret symmetric key for the Hill cipher and show that P_i is the only matrix that can decrypt the cipher P''_i correctly. We use non-singular matrix $Q \in \mathbb{Z}_2^{k\times k}$ as the *permutation matrix* in the scheme while P_{i+1} is the *message* to transfer from the server to the reader. We could consider a special case: $Q^{-1} = Q^T$ when $QQ^T = Q^TQ = I$ where I is the identity matrix. For contradiction, suppose there is a non-singular matrix G, such that G=P_i. In that case, for every valid (P''_i,P_{i+1}, Q) there exist $\mathrm{G}^{-1}\mathrm{P}''_i\mathrm{P}'_i$ =P_{i+1}. This clearly concludes that whatever Q is, we have G=P_i. This should also hold for Q such that $QG = Q\mathrm{P}_i$ =P'_i, but that is not possible. So the only matrix that can decrypt successfully is P_i^{-1} that contradicts our assumption on G. Since CPA attack enquires k-pairs of plaintext-ciphertext pairs,using a linear transformation by a fixed matrix leads to linear dependency that results weak security. In our scheme, both Q and P_i is refreshed in each session. It is like *one time one key matrix* for each block ciphering where the key has been derived from the preceding key matrix i.e., $\mathrm{P}_{i+1} \leftarrow \mathrm{P}_i$. More concisely, we use two different matrices: one is to randomize P_{i+1} by permutation matrix Q, another is to convey Q. However, commitment s'' is generated on the message s' by the commitment key P''_i from LPN. Therefore, reader can verify the commitment and hence the permutation matrix Q.

Let rewrite the ciphertext $\mathrm{P}''_i \in \mathbb{Z}_2^{k\times k}$ as: $\mathrm{P}''_i = \mathrm{P}_i \mathrm{P}'^{-1}_i \mathrm{P}_{i+1}$ such that, Y = HZX (mod 2) for simplicity. Since Q is refreshed at each transaction, the equation can be written as follows:

$$\begin{gathered} Y_0 = HZ_0X_0 \ (mod\ 2) \\ Y_1 = X_0Z_1X_1 \ (mod\ 2) \\ Y_2 = X_1Z_2X_2 \ (mod\ 2) \\ \vdots \\ Y_k = Y_{k-1}Z_kX_k \ (mod\ 2) \end{gathered}$$

It can be clearly seen from the above equations: although the attacker knows k-pairs of (Y X), k equations cannot be used to solve a $k \times k$ non-singular matrix P_i at any time instance i that resist CPA attack.

Let a valid ciphertext-plaintext pair (P''_i,P_{i+1}) with a permuting matrix Q yield a set of key matrices G_q. Then the number of solution matrices for G_q is $2^{k(k-rank(\mathrm{P}_{i+1}))}$. Although the knowledge of all valid pair (P''_i,P_{i+1}) is sufficient to determine P_{i+1}, but it demands exponential time/memory considering the size of the set G_q. Therefore, the probability that a key matrix $G \in G_q$ decrypts correctly a randomly and uniformly chosen pair (P''_i,P_{i+1}) is negligible ($1/2^{k(k+1)}$). In the optimal case, this probability is $1/2^{k^2}$ where a non-trivial permutation matrix is used. □

Secure Exact LPN:

Proposition 1 *The exact version of search* LPN_x *is hard if and only if standard search* LPN_τ *is hard* [117].

Proof: Let an adversary $\mathcal{A}$ find out the secret x with advantage ϵ for LPN_x where the error vector e', sampled in LPN_τ, has weight $\lfloor k\tau \rceil$ with probability at least $1/\sqrt{k}$ such that

$$\mathsf{Pr}[\mathcal{A}(A, A.x \oplus e') = x] \leq \epsilon/\sqrt{k}$$

where $e' \in \mathsf{Ber}^k_{\lfloor k\tau \rceil}$. It is not hard to see that error distribution on the above case is exactly same as that of *exact search* LPN_x.

Proposition 2 *The hardness of decisional* LPN_x *is polynomially related to that of search* LPN_τ[117].

Proof: The standard search and decision LPN_τ are equivalent. However, reduction from the search to decision incurs the number of samples k in the decision-LPN_τ to be larger than that in the search-LPN_τ [89, 129]. However, the hardness of the LPN_x problem holds assuming the hardness of the standard LPN_τ problem, where the reduction is based on the Goldreich-Levin theorem described in [112]. Note that if security of the scheme is considered on the standard LPN assumption in a provable manner, there is no efficient attacks against LPN_x than against LPN_τ. However, if the loss in the reduction is taken into account, it might result in large parameters. The security of the commitment scheme is directly based on the standard LPN_τ. Actually it replaces the LPN_τ assumption with an assumption where the upper bound on the weight of the error vector is fixed, i.e., $\lfloor k\tau \rceil$, thus removes the completeness error. In [117], authors show a protocol for proving knowledge of committed values whose security relies directly on the standard decisional LPN_τ assumption. However, the protocol has a soundness or knowledge error 4/5, and thus requires running the protocol roughly twice in order to achieve the same knowledge error. Interested readers are referred to [117], for further clarification and proof of the theorem.

Lightweight signature scheme:

Theorem 16 *If Sum of Subset (SSP) is NP-complete then Binary Matrix Factorization (BMF) problem is also NP-complete.*

Proof: SSP is a decision problem that is proved as NP-complete [114]. We now convert BMF problem to SSP problem by following PPT algorithm:

- Let replace " + " operation of SSP to " $\oplus$ ".
- Consider BMF problem that given $B = XA$ where $B \in \mathbb{Z}_2^k$, $X \in \mathbb{Z}_2^n$ and $A \in \mathbb{Z}_2^{n\times k}$, we have to find A
- Let $X = \langle x_1, \cdots, x_n\rangle$ s.t., $x_i \in \mathbb{Z}_2$ then we have $XA = x_1a_1 \oplus \cdots \oplus x_na_n$ where a_i is the i^{th} row of the matrix A.

For any large n, we will find a list of integer $L = a_j$ s.t., $1 \leq j \leq n$ and $a_j \in \{0, 1, \cdots, 2^k - 1\}$. Given the list L and B finding out X is surely an SSP.

Theorem 17 *If BMF is a hard problem, then construction of the lightweight stateful signature is existentially unforgeable under a one-time chosen message attack.*

Proof: Let $\mathcal{A}$ be a PPT adversary such that $(pk,\ sk) \leftarrow \mathsf{KGen}(1^k)$; $(m, \sigma) \leftarrow \mathcal{A}^{\mathsf{Sign}_s{}^{k(\cdot)}}(pk)$ where $\mathcal{A}$ is allowed only a single query to its signing oracle. Given the public key $pk = X^+X$, $\mathcal{A}$ needs to find an X^+ such that $f(X^+) = X^+X$ where X^+ is the unique pseudo-inverse of a matrix X.

Let m' be the message that $\mathcal{A}$ queries to its signing oracle then the signature scheme is assumed to be forged on the event: $\mathrm{Vrfy}(m, \sigma) = 1$ and $m \neq m'$. In a certain experiment $\mathrm{Exp}_{\mathcal{A},\Pi}(1^k)$ of the signature scheme Π, success probability can be defined:

$$\mathsf{Succ}_{\mathcal{A},\Pi}(k) := \mathsf{Pr}[(m, \sigma) \leftarrow \mathrm{Exp}_{\mathcal{A},\Pi}(1^k) : \mathsf{Vrfy}(m, \sigma) = 1 \text{ and } m \neq m']$$

Since BMF is hard, $\mathcal{A}$ runs the experiment $\mathrm{Exp}_{\mathcal{A},\Pi}(1^k)$ by choosing a random $X^+ \in \mathbb{Z}_2^{m\times n}$ where m is the rank of matrix X^+. Then setting $pb := X^+X$, $\mathcal{A}$ will try to output forgery. Since pb is updated at every state by using a random matrix M. Given $XX^+M \in \mathbb{Z}_2^{n\times n}$, suppose that $rank(X^+) = r$, and $I^{r\times r}$ is an *Identity matrix* s.t.,

$$X^+X = \begin{pmatrix} I^{r\times r} & 0 \\ 0 & 0 \end{pmatrix} \Rightarrow X^+XM = \begin{pmatrix} Q^{r\times r} & 0 \\ 0 & 0 \end{pmatrix}$$

Then the probability of $\mathcal{A}$ to determine the correct M is $2^{-(n-r)m}$ [113] where $n \gg r$ such that $n \gg m$. Hence the probability that $\mathcal{A}$ outputs forgery is at least $\mathrm{Succ}_{\mathcal{A},\Pi}(k)/2^{-(n-r)m}$. Since sk is updated at each protocol transaction $\mathrm{Succ}_{\mathcal{A},\Pi}(k)/2^{-(n-r)m} \leq neg(k)$. We conclude that $\mathrm{Succ}_{\mathcal{A},\Pi}(k)$ is negligible. It is worth mentioning that the signature scheme is might be insecure if the sk is used to sign more than one message. Adversary $\mathcal{A}$ who obtains several signature w.r.t the same pk might learn the entire sk by using Gaussian elimination method.

Man-in-the Middle Attack: The most sophisticated and realistic attack in an RFID system is the Man-in-the Middle (MIM) attack. Our protocol is MIM-secure against an active attack from several assumptions i.e, the exact LPN, secure Hill cipher and pseudo-inverse matrix properties. In case of tag-reader, the authentication tags $(\gamma_1, \gamma_2) \leftarrow [(\mathrm{I_i}, r, \sigma_i, s'), (\mathrm{P}', r')]$ is MIM-free: $\gamma_1 = (s, \sigma : f_{k_1}(s))$, $\gamma_2 = (s', r' : \mathbf{S} \cdot \bar{f}_{k_2}(s') \oplus e')$ where $(f_{k_1}, \bar{f}_{k_2})$ are secret key derivation functions which uniquely encode challenges resp. s and s' according to the keys $(k_1,\ k_2)$ where we use resp. $\mathbf{S}''$ and $(\mathbf{S}, \mathbf{T}_{id})$ as the secret keys $(k_1,\ k_2)$. The main technical difficulty to build a secure MIM-free authentication from LPN is to make sure the secret key k_i does not leak from verification queries. Since we randomize $\mathbf{S}''$, and hence $\mathbf{S}$ at every protocol session i and protocol transcripts are computationally indistinguishable from the exact $\mathsf{LPN}x$ assumption, the tag-reader communication is MIM-secure. On the other hand, reader-server authentication tag $(\gamma_3, \gamma_4) \leftarrow [(\mathrm{I}_i, r, s'), (\mathrm{P}', \mathrm{P}'', r', s'')]$ is MIM-free from exact LPN_x (γ_3 likewise γ_2) and secure Hill cipher assumption. Let $\gamma_4 = (\mathrm{P}', \mathrm{P}'' : \hat{f}_{k_3}(\mathrm{P}', \mathrm{P}''))$ be an authentication tag for Hill cipher, where $\hat{f}_{k_3}(\cdot)$ is the secret key derivation function with the secret key $k_3 \leftarrow \mathrm{P}^{-1}$. Since the variation of Hill Cipher used in this protocol can resist Chosen plaintext attack and we update $\mathbf{P}$ at each protocol session i, the reader-server communication is MIM-secure. In addition, we use a pseudo-random matrix as blinding factor that is secure under pseudo-inverse matrix properties. Therefore, even if the adversary compromises T_{id}, it cannot generate $\mathbf{S}''$ and hence $\mathbf{S}$ for any subsequent sessions using only T_{id}.

Tag tracking: Let a tag's responses to a certain reader be linkable to each other, or distinguisahble from other tags. As a result, location of the tag can be tracked by a malicious adversary, called unauthorized *tag tracking.* This yields a serious threat to the privacy that can be avoided if the protocol responses appear to an attacker is random and uniformly distributed. Our authentication scheme is *tracking resistant* under the security of exact-LPN, the pseudo-randomness of LPN. We consider tag tracking in the context of the server-tag communication. Note that under the decisional LPN assumption $(s, \mathbf{S}.(\mathbf{T} \parallel s) \oplus e)$ samples are pseudorandom. Even if an adversary has access to the protocol message (s, r) or (s', r'), session key $\mathbf{S_i}$ cannot be deduced efficiently. Because the success probability of guessing the correct secret key is negligible ($\approx 2^{k\times(l+v)}$) even without considering the permanent secret $\mathbf{T}$. In order to resist *side channel* and *replay* attack, we use random nonce (s, s') in tag-server communication. In addition, since $\mathbf{S}''$ is refreshed at each protocol session, even if an attacker replays the previous session message in an attempt to eavesdrop necessary number of outputs of the tag to break security, authentication will fail as each $\mathbf{S}''$ is used only once. Note that the tag-reader communication is not important to consider for tag tracking, since the shared secret for the LPN problem (tag-server communication) is not identical to that of tag-reader. However, in case of tag-reader communication, we use a stateful signature scheme where signing key is refreshed at each protocol session. This will also help the protocol transaction to be unlinkable.

Desynchronization: *Desynchronization* is a kind of Denial of Service (DoS) attack where the tag and reader/back-end server cannot recognize each other (due to adversarial interruption or impediment), and so finally the tag gets disabled. If tag authentication involve random initialization from the tag, it can cause replay attack and hence yields desynchronisation in the tag-server communication. Therefore, our protocol has been initialized by the reader. An adversary can produce a valid protocol message only if it successfully discovers the shared secrets among the entities. Since the tag is assumed to be uncorrupted and the LPN and BMF problems are NP-hard, a PPT adversary cannot learn the secrets. The only way to desynchronize the tag is to block the last message of the protocol and to update the shared session secret key ($\mathbf{S''_i}$) (contrary to the server). More precisely, let the tag $\mathcal{T}$ and server $\mathcal{S}$ share a secret key $\mathbf{S}$. $\mathcal{T}$ requires to prove its status to $\mathcal{S}$. Either $\mathcal{T}$ or $\mathcal{S}$ knows the authentication is successful. Assume that $\mathcal{S}$ knows the authentication result, while $\mathcal{T}$ is unaware and desynchronization occurs. In order to address this problem, we propose to preserve the session key ($\mathbf{S''_{i-1}}$) used in the previous successful authentication session in the server. If the server fails to match the current index $\mathbf{I}_i$, and consequently the current session key ($\mathbf{S''_i}$), it will try with ($\mathbf{S''_{i-1}}$).

However, the adversary cannot execute the same attack twice consecutively. First, the tag could not be totally desynchronized, since it has another permanent key ($\mathbf{S}'$). Secondly, the server immediately re-synchronizes the key in the next consecutive session by *brute-force* searching (seek and match the previous session key $\mathbf{S''_{i-1}}$).

5.2.4 Privacy analysis

Security of a protocol ensures that if a valid tag is impersonated or interfered by an adversary, the reader rejects the transaction session. However, there are several privacy models proposed for RFID authentication protocols. They are roughly divided into: indistinguishability based (IND-privacy), unpredictability based (UNP-privacy), Zero-Knowledge based (ZK-privacy) simulation based (SIM-privacy), universal composability based etc. Relationship among the privacy definitions have been discovered recently [130]. According to [130], ZK-privacy is equivalent to IND-privacy. And the gap between IND-privacy and SIM-privacy comes from whether the protocol message is public verifiable. Otherwise they are also equivalent.

Since LPN-based authentication use shared secret key (not public verifiable) and protocol adapts mutual authentication, we analyzed our protocol according to the privacy framework based on ZK-privacy described in [123] that rely on a zero-knowledge formulation. They assume that the protocol is always initiated by the reader, transaction message does not disclose any tag secret, and consists of $\pi \leftarrow 2\lambda+1$ $s.t.$ $\lambda \geq 1$ rounds. Our mutual authentication protocol follows ($\pi = 3$ $s.t.$ $\lambda = 1$) and adopts the identical assumption. Due to space constraint, we refer to the definitions of generic oracles from [123].

Let $\hat{\mathcal{A}}$ be a PPT CMIM (Concurrent Man in the Middle) adversary equivalent to $\mathcal{A}$ (respectively, simulator Sim) that takes on input the system public parameters Pub_T, the reader $\mathcal{R}$ and the set of tags $\widehat{\mathcal{T}}$; and interacts with $\widehat{\mathcal{T}}, \mathcal{R}$ via the oracles. $\hat{\mathcal{A}}$ outputs an arbitrary tags $C \subseteq \widehat{\mathcal{T}}$ called *clean tags.*

Let $\widehat{\mathcal{A}}$ be composed of a pair of adversaries $(\widehat{\mathcal{A}}_1, \widehat{\mathcal{A}}_2)$ and their corresponding simulators $(\mathsf{Sim}_1, \mathsf{Sim}_2)$ for $\mathbf{Exp}^{\mathsf{ZK}}_{\mathcal{A}}(\widehat{T})$ experiments.

Experiment $\mathbf{Exp}^{\mathsf{ZK}}(\widehat{\mathcal{T}})$

- Initialize RFID system, the reader $\mathcal{R}$, the tag set $\widehat{\mathcal{T}}$ (s.t., $|\widehat{\mathcal{T}}| = l$) by $\mathbf{SetupTag}(\cdot)$
- let $\mathcal{O} \leftarrow \mathrm{Launch}, \mathrm{Dtag}, \mathrm{STag}, \mathrm{SReader}, \mathrm{Ukey}, \mathrm{Corrupt}$
- Real: $(\mathcal{T}, st) \leftarrow \hat{\mathcal{A}}_1^{\mathbf{DTag}}(\mathcal{R}, \widehat{\mathcal{T}}, \mathsf{Pub}_T)$
  Simulation: $(\mathcal{T}, st) \leftarrow \mathsf{Sim}_1^{\mathbf{DTag}}(\mathcal{R}, \widehat{\mathcal{T}}, \mathsf{Pub}_T)$
  where $\mathcal{T} = \{T_{i_1}, T_{i_2}, \cdots, T_{i_\delta}\} \in \mathcal{T}$ s.t., $0 \leq \delta \leq l$
- $c \in_R C \leftarrow \{1, 2, \cdots, l-\delta)\}$ and $C = \widehat{\mathcal{T}} - \mathcal{T}$
  Real: $T_c = T_{i_c}$
  Simulation: c is unknown to Sim_2
- Real: $view \leftarrow \hat{\mathcal{A}}_2^{\mathcal{O}}(\mathcal{R}, \widehat{\mathcal{T}}, T_c, st)$
  Simulation: $sview \leftarrow \mathsf{Sim}_2^{\mathcal{O}}(\mathcal{R}, \widehat{\mathcal{T}}, st)$
- Real: output $(c,\ view_{\hat{A}})$
  Simulation: output $(c,\ sview_{Sim})$

We assume that $\hat{\mathcal{A}}$ queries the challenger with $\mathbf{Exp}^{\mathsf{ZK}}(\widehat{T})$ in the *real* and *simulation* mode. Note that if $\delta = 0$, no challenge tag is selected and the number of clean tags $|C| = l - \delta$.

ZK-privacy implies that adversary $\hat{\mathcal{A}}$ cannot distinguish any challenge tag T_c from any set C of tags. That's why, $\hat{\mathcal{A}}_1$ is used to output an arbitrary set C and to limit $\hat{\mathcal{A}}_2$ to blind access to a challenge tag from C. Therefore, the advantage of the adversary with security parameter κ to win the privacy game is negligible that defined as

$$\mathbf{Adv}^{\mathsf{ZK}}_{\mathcal{A}}(\kappa, \widehat{T}) = |\mathsf{Pr}[\mathbf{Exp}^{\mathsf{ZK}}_{\hat{\mathcal{A}}}(c, l, view(.) = 1)] - \mathsf{Pr}[\mathbf{Exp}^{\mathsf{ZK}}_{Sim}(c, l, sview(.) = 1)]| \leq \epsilon$$

ZK-privacy. RFID Authentication protocol described in Fig. 1 satisfies the ZK-privacy model if for any PPT adversary $\hat{\mathcal{A}}$ (resp. PPT simulator Sim), $\mathbf{Adv}^{\mathsf{ZK}}_{\mathcal{A}}(\kappa, \widehat{T})$ is negligible.

Theorem 18 *Our proposed authentication protocol is forward (resp., backward)-*ZK *private.*

Proof: ZK-privacy allows to give the secrets to the adversary $\mathcal{A}$ at the end of the experiment. Let a pair (k^f, s^f) be a final key (k) and internal state (st) of a challenged tag T_c from the initial (k^0, st^0). Then the protocol is forward (resp., backward)-ZK private if any PPT distinguisher $\mathcal{D}$ cannot distinguish $(k^f, s^f, c, T_c, view_{\mathcal{A}}(\kappa, l))$ from $(k^f, s^f, c, T_c, sview_{Sim}(\kappa, l))$ after the oracle $\mathbf{Ukey}(\cdot)$ is run by $\hat{\mathcal{A}}_2$. Note that T_c should not be in the oracle table D (related to $\mathbf{DTag}(\cdot)$) before the experiment $\mathbf{Exp}^{\mathsf{ZK}}(\widehat{\mathcal{T}})$ ends. However, *forward* (resp., *backward*)-ZK privacy cannot be achieved if $\mathcal{A}$ has corrupted the challenging tag T_c before the experiment finishes. □

Table 5.5: Tag Resources and Security Comparison with HB family

Scheme	**Storage**	**Computation** (major)	**Authentication**	**Security achieved**	**Hardware** (gates)
HB-MP[92]	2 S	1 LPN	tag	0,5,6	≈ 1600
HB-MP$^+$ [108]	2 S	1 LPN,1 HASH	tag	1,5,6	≈ 3500
GHB# [124]	2 S	1 LPN	tag	1,5,6	≈ 1600
F-HB [104]	1 I , 1 S	1 PRNG,2 LPN	mutual	$1, 2, 4^*, 5, 6$	≈ 3500
[113]	1 I, 1 S	1 SLPN,1 P	mutual	1,2,3,4†,5,6,7	≈ 1600
ours	1 I, 2 S	1 LPNx,1 P, 1 H	Full mutual	1,2,3,4,5,6,7	≈ 2000

where S:= Secret key; I:= Index; H:= Hill cipher; PRNG:= Psudo Random Number Generator; P:= Pseudo Inverse Matrix; LPN:= Learning parity from noise SLPN:= Subset LPN; $\mathsf{LPN}x$:= exact LPN
Security attributes: Active attack(0), MIM attack(1), Forward Security (2), Reduced Backward Security (3), ZK-privacy (4), IND-privacy (4†), UNP-privacy (4*) Tag tracking (5), De-synchronization (6), Replay attack (7).

5.2.5 Comparison and Performance

Computation Requirement: We focus on tag, which is the computationally weakest. Most of the expensive computations will be performed at the server site. The *exact* version of the LPN problem used in the protocol is of independent interest as this assumption removes the completeness error [117]. Setting $v = l$ in the public parameters, it results $k = \theta(v + l) = \theta(v)$ and commitment scheme requires $2^{\theta(v/\ log\ v)}$ time. Thus, commitment proof is quasi-linear in the length of the committed messages.

Major protocol operations regarding the tag include *one* LPN problem generation and checking and *two* binary linear matrix multiplications. As bitwise XOR, matrix multiplication, and calculating the hamming weight $\mathbf{w}(\cdot)$ are all binary operations, they can easily be implemented using bit-by-bit serialization to save hardware gates.

In order to compute a Hill ciphertext with randomized permutation need $2k$ vector products over $\mathbb{Z}_2$ If the vectors are stored in *words*, the vector product can be simply reduced to a *logical* AND (&) and *parity check* operations. Therefore, $\sum_{i=1}^{k} a_i b_i (mod\ 2)$ is equivalent to $a\&b$ that needs only $12k$ operations [119]. In decryption case (in the reader), we need $3k$ vector products over $\mathbb{Z}_2$ and an inverse operation that can be pre-computed to enhance efficiency. That's why, we need k^3 (multiplication) $+(k^3 - k^2)$ (addition) over

$\mathbb{Z}_2$.

Storage Requirement: All the parties in the protocol need to store the public parameters. However, a tag needs to store only 2 secret keys and an index for the session $(k \cdot l + k \cdot v + k)$ bits, a reader requires to store a tag identifier and 1 secret key $(k^2 + l)$ bits while the server needs to maintain a database for all the tags (for session i and $i-1$) with index, tag identifier and 3 secret keys $(2k \cdot l + 2k \cdot v + 2k^2 + 2k + l)$ bits for each tag. Consequently, storage requirement for the tag and the reader can be expressed by $\mathcal{O}(1)$ while that is $\mathcal{O}(n)$ for the server such that n is the number of tags in an RFID system.

Communication complexity: The protocol requires $(k^2 + 2v + 4k)$ bits in the tag-reader communication and $(2k^2 + 2k + 3v)$ bits in the reader-server communication. There is a natural trade-off between the communication cost and key size. For any constant c $(1 \leq c \leq k)$, the communication cost can be reduced by a factor of c by increasing the key size with the same factor.

In **Table 5.2.4**, we show a comparative study on some general attributes e.g., storage consumption, major computations, authentication party, achieved security, approximate hardware cost etc., between our protocol and several HB-like and non-HB protocols. It appears that although the tag's hardware cost of the proposed protocol is optimal, it achieves most common security requirements and uniquely *full mutual authentication* properties from exact LPN assumption.

5.2.6 Conclusion

This chapter presents a novel hardware-friendly RFID authentication protocol, based on a commitment scheme from the exact LPN problem, that can meet the hardware constraints of the EPC Class-1 generation-2 tags. In comparison with other protocols as described in Table 2, it requires less hardware and has achieved major security attributes. The protocol is also compliant to ZK-*private* privacy settings. Moreover, this is the first protocol that allows mutual authentication for the whole system i.e., tag, reader and server from the LPN problem. Furthermore, security and privacy proofs are given in the standard model that uses indistinguishability as basic privacy notion. Note that the proposed protocol can be easily utilized for other popular security protocols of RFID application s,t., ownership transfer, supply chain management etc.

5.3 A Scalable and Secure RFID Ownership Transfer Protocol

5.3.1 Introduction

Modern inventory systems often rely upon RFID tags to allow automatic identification of tagged objects where readers can read even thousands of unique RFID tags in a single snatch. A secure RFID inventory system often needs to transfer ownership of RFID tags. Once tagged objects passes through distributed supply chain from a manufacturer to a consumer, *ownership* of the objects could be transferred among consumers several times.

Therefore, our solution utilizes Semi-Trusted Parties (STPs) where consumers (resp. owner) do not need to contact the manufacturer (trusted main server) each time it needs to transfer ownership. In real life, we may think that main server is located at the manufacturer's *Head office* and all other STPs are located at remote sites such as *Regional offices*. STPs on behalf of a trusted server can anonymously monitor and verify ownership transfer process without revealing any secrets. Later STPs could forward the ownership data to other STPs, or to the main server. By the term *semi-trusted*, we mean that an STP is an online designated server that follows prescribed protocol correctly and communicates to the readers so that the communicating readers yields on a mutually satisfactory agreement.

HB-family protocols based on LPN assumption are booming as one of the attractive candidates for secure low cost EPC tags [90, 91, 104, 113, 99, 89, 101]. due to its security against quantum adversaries, efficient computational time and memory requirement etc. In this work, we have designed a novel ownership transfer protocol by modifying a recently proposed RFID authentication protocol based on Ring-LPN problem [138] where the secret key and other parameters are taken over the field $\mathbb{F}_2$. It allows us to seamlessly use the same parameters (as used in authentication protocol) in the aggregated signature scheme.

Consider an *inventory management* of a large *supply chain system* where *vendors* contribute goods or services to the next link in the chain. Usually each vendor (owner) holds several RFID tags. In order to manage ownership transfer among the vendors in a large supply chain, we employ, after effective customization, the HomSig scheme described in [140] where signatures generated by the vendors[11] can be aggregated by an STP. The scheme is secure under standard model. Similarly, several aggregated signatures of a vendor (owner) could be combined by any legitimate intermediate STP and later signatures will be verified by the trusted main server.

Main contribution. In this work, we first modify the scheme in [138] in order to achieve a MIM-attack free mutual authentication protocol[12]. The protocol employs a STP to avoid the communication overload on the trusted main server. It supports ownership of multiple tags (to update ownership record of tags in the trusted server) of an owner to be transferred simultaneously. Unlike other authentication protocols for ownership transfer system, communications between the server and readers are assumed to be

[11] each signature is generated from several tags of a vendor or resp. owners.

[12] Authentication protocol in [138] is susceptible to MIM-attack [142] and do not support mutual authentication.

insecure and over inauthentic channel. Therefore, readers and servers are not identical. For construction, we adopt several new ideas such as:

- Only main server (not STPs) is assumed to be secure and keys are shared among the entities accordingly.
- Applying exact (not decisional) version of the LPN problem in order to repel completeness error.
- Using Field-LPN problem described in [138] to cope with the parameters needed in homomorphic signature scheme.
- Employing a lightweight searchable encryption and signature scheme based on the properties of pseudo-inverse matrix between the readers (resp. owners) so that an STP can verify anonymously which owner is transferring ownership to whom.
- A lightweight homomorphic aggregated signature (HomSig) to forward ownership data to the main server.

Assumption and System architecture: An inventory system described in this work consists of a single legitimate trusted server called *main server*, a set of intermediate servers called STPs, a set of readers and their corresponding owners, and a set of tags (EPC class). Note that STPs are assumed to be *semi-trusted*[13] and are constituted by the Main Server. Each owner has a unique ID. A reader could be shared among owners, or owned by a RFID tag owner. Readers would be connected to the back-end intermediate STPs during ownership transfer. We introduce the inclusion of STPs for 2 reasons:

- In order to ease physical communication between the owners and a remote trusted server.
- To act as a witness between the current and new owner on behalf of the trusted server.

Main server stores all the data related to the tags in the database. Each tag has a unique identifier T used as a permanent key, an index $\widehat{T}$ and a session key c. We assume index-owner tuple $[\widehat{T}, \mathcal{U}_{cur}]$ in the server database is unique for efficient searching. Since an RFID tag is not tamper-resistant, its session key is refreshed after each i^{th} session completes successfully. For updating key, each tag authenticates its legitimate reader.

We assume a hierarchical architecture where tags are placed in the lowest level in the hierarchy and trusted main server is set at the highest level. Readers and STPs are located somewhere in between. Only the main server is assumed to be trusted while other STPs are considered to be *semi-trusted.* Imagine a situation in an inventory management system where manufacturer preserves the main server and delegates its task to STPs placed in different locations for consumer's convenience.

In case of updating the ownership data on the trusted main server, the current reader should not be considered as *honest* (too strong assumption). Because the malicious current

[13] A form of *honest-but-curious* attacker model. However, multiple STPs are not allowed to collude.

owner could claim that he/she is still the current owner without performing the last step (Step-3) of the protocol. In this protocol, we consider the new reader (resp. owner) to be honest and hence is responsible to transfer ownership records to the STP. Meanwhile, new reader has to update the keys of the tags individually to finalize ownership transfer.

5.3.2 Construction

We exploit Field version of the Ring-LPN problem described in [138]. We set aside significantly less computations to the tag than any other entities (e.g., readers, STPs). We divide the ownership transfer protocol in 3 phases: Step-1 describes the communication between a tag and its current and new readers. It includes a mutual authentication protocol between a reader and a tag. Step-2 delineates the protocol transactions between the current and new readers through a designated STP server. Finally, Step-3 outlines the homomorphic signature scheme applied to the readers, STPs, and the main server.

Tag registration: When a tag is registered in the inventory system main server retains the tag associated data such as a unique identifier T, an initial index $\widehat{T}$[14] and current owner $\mathcal{U}_c$ data in the database. Similarly, the main server will set the necessary data into the tag's non-volatile memory such as public parameters for LPN problem $(F, n, \pi_1, \pi_2, \tau)$, a permanent key T, an initial session key c_0 and an initial index $\mathrm{I}_0 \leftarrow \widehat{T}^0$.

User registration: Each potential user $\mathcal{U}_i$ of the system needs to register with the Main Server. Main Server generates a key pair (sk, pk) for homomorphic signature scheme and provide the secret key $k_i \leftarrow sk$. User $\mathcal{U}_i$ will provide k_i to the reader $\mathcal{R}_j$ at the time it initializes ownership transfer process. Besides this, each user $\mathcal{U}_i$ retains the initial tags' data such as $(T, \widehat{T}, c)$ for all the tags it owns.

Reader registration: Readers in the system need to register themselves with Semi-trusted Party (STP). STP generates a key pair (sk, pk) for every reader $\mathcal{R}_j$ (for searchable encryption scheme). Moreover, any two readers in ownership transferring require a shared secret key ρ in order to transfer tags data. Let X^+ be pseudo-inverse of a matrix X, $S_i := X^+ \in \mathbb{Z}_2^{m \times n}$ and $P_i := XX^+ \in \mathbb{Z}_2^{m \times n}$. STP generates key pair (S_c, P_c) for $\mathcal{R}_{cur}$ and (S_n, P_n) for $\mathcal{R}_{new}$ respectively.

Encouraged by the proposal described in [138], we define 2 suitable mappings π_1, π_2 such that $\pi_{(i)} : \{0,1\}^\lambda \to F$. Let $s \in \{0,1\}^\lambda$ for the security parameter $\lambda = 80$ be defined as: $(s_1, \cdots, s_{10})$ or $(s_1, \cdots, s_{16})$ where s_i is a number between (1 to 256) or (1 to 32) respectively. Defining the coefficient of the polynomial $v = \pi_i(s) \in F$ as zero except all positions of i such as $i = 10 \cdot (j-1) + s_j,\ j = 1, \cdots 10$ (for π_1) and $i = 16 \cdot (j-1) + s_j,\ j = 1, \cdots 16$ (for π_2). Therefore, both $\pi_1(s)$ and $\pi_2(s)$ are sparse and injective since they will have exactly 10 and 16 non-zero coefficients respectively.

[14] First index I_i of a tag $\mathcal{T}$ after a successful ownership transfer.

Step-1.1: Although we follow the Field version of the Ring-LPN problem [138], we restrict the Field-version of the Ring-LPN problem (to finite field of characteristic 2) according to the following. Let an irreducible polynomial $f(X)$ be taken over the field $\mathbb{F}_2$ where the degree of f is n, we consider an extended field[15] on $\mathbb{F}_2$ defined as: $F = \mathbb{F}_2[X]/(f) = \mathbb{F}_{2^n} = \mathbb{F}_q$. Therefore, any element $a \in \mathbb{F}_2[X]/(f)$ has a multiplicative inverse in F^{*}[16].

For *tag authentication*, a shared secret key pair (T, c_i) and an index $(\mathrm{I}_i \leftarrow \widehat{T})$ have been derived either from initial tag registration process or from the previous $(i-1)$ successful sessions.

[15] E.g., $f(X) = X^{532} + X + 1$ of degree $n = 532$.

[16] F^* is the set of elements in F that have multiplicative inverse.

Step-1.1: Tag-Current reader communication (Authentication)

$\mathcal{R}_{cur}$		**Tag** $\mathcal{T}$
$\{\mathrm{T}, \mathrm{I}_i, c_i\} \in F$		$\{\mathrm{T}, \mathrm{I}_i, c_i\} \in F$
$s \overset{\$}{\leftarrow} \{0,1\}^\lambda$		
	$\xrightarrow{s}$	
		$r \overset{\$}{\leftarrow} F^*$, $e \overset{\$}{\leftarrow} \mathbf{Ber}_w^F$ $z := r \cdot (c_i \cdot \pi_1(s) + \mathrm{T}) + e$
	$\xleftarrow{r, z, \mathrm{I}}$	
Lookup T by using I: Direct match: If $(\mathrm{I} \neq \mathrm{I}_i)$ Brute-force search: $\exists$ $(\mathrm{T}, c_{i-1}$ or $c_i)$ that satisfies: IF $(r \notin F^*)$ return; $\hat{e} := z - r \cdot (c_i \cdot \pi_1(s) + \mathrm{T})$ IF $\mathbf{wt}(\hat{e}) \neq w$ return;		
IF (AUTH) $e' \overset{\$}{\leftarrow} \mathbf{Ber}_w^F$ $z' := r \cdot (c_i \cdot \pi_2(s) + \mathrm{T}) + e'$ $c_{i+1} = c_i + \hat{e}$ $\mathrm{I}_{i+1} = z + \hat{e}$		
	$\xrightarrow{z'}$	
IF (OT) Go to Step-2		AUTH: $\hat{e} := z' - r \cdot (c_i \cdot \pi_2(s) + \mathrm{T})$ IF $\mathbf{wt}(\hat{e}) \neq w$ return; $c_{i+1} = c_i + e$ $\mathrm{I}_{i+1} = z + e$

- Reader: Generate a random binary λ-bit challenge string s, and send it to the tag.

- Tag: Generate a random noise vector e from $\mathbf{Ber}_w^F$ and a random element r from F^*. Next a multiple bit Field-LPN problem z is computed from $r \cdot (c_i \cdot \pi_1(s) + \mathrm{T}) + e$. Finally, the tag forwards (I_i, r, z) to the reader.

- Reader: It first searches local database to match a tuple $\{\mathrm{I}_i, \mathrm{T}, c_i\}$. If failed, then apply brute-force search with $\{\mathrm{T}, c_i$ or $c_{i-1}\}$. Note that, the reader would store the secret key c_{i-1} from the previous session in order to resist *De-synchronization* attack. Then it checks whether r is chosen from F^* and then calculates $\hat{e}$ and check whether $\mathbf{wt}(\hat{e})$ is exactly $\lfloor k\tau \rceil$. If the check passes, it accepts the tag and go to the reader authentication phase.

Reader authentication can be decomposed in two phases: (1) Authentication (AUTH), (2) Ownership transfer (OT)

- Reader: During AUTH phase, e' is generated from $\mathbf{Ber}_w^F$ and z' is calculated accordingly. Note that unlike [138], we use the same r as it received from the tag to resist synchronization attack and use a different mapping π_2 with the same challenge s in a view to reduce communication overhead. Update session secret key $c_{i+1} \leftarrow c_i + \hat{e}$ and index $\mathrm{I}_{i+1} \leftarrow z$.

- Tag: Verify the Field-LPN problem by checking the $\mathbf{wt}(\hat{e})$ whether it is exactly w or not. If the check passes, accept the reader and update the session key $c_{i+1} \leftarrow c_i + e$ and index $\mathrm{I}_{i+1} \leftarrow z$.

Step-1.2: New reader-tag communication (Ownership Transfer)

$\mathcal{R}_{new}$ $\{\mathrm{T}, c_i\} \in F$	**Tag** $\mathcal{T}$ $\{\mathrm{T}, \mathrm{I}_i, c_i\} \in F$
$\{\mathrm{T}, \widehat{T}, c\} \leftarrow \mathcal{D}_\rho(\Gamma_j)$	
$s' \overset{\$}{\leftarrow} \{0,1\}^\lambda$	
$e' \overset{\$}{\leftarrow} \mathbf{Ber}_w^F$	
$z' := r \cdot (c_i \cdot \pi_2(s') + \mathrm{T}) + e'$	
$c_{i+1} = c_i + e'$	
$\mathrm{I}_{i+1} = z' + e'$	
$\widehat{T}_n := \mathrm{I}_{\mathrm{i+1}}$	
Go to Step-3 $(\mathbf{tid}, \widehat{T}, \widehat{T}_n)$	
$\xrightarrow{s', z'}$	
	OT:
	$\hat{e} := z' - r \cdot (c_i \cdot \pi_2(s') + \mathrm{T})$
	IF $\mathbf{wt}(\hat{e}) \neq w$ return;
	$c_{i+1} = c_i + \hat{e}$
	$\mathrm{I}_{i+1} = z' + \hat{e}$

Step-1.2 During OT phase, $\mathcal{R}_{cur}$ records the ownership index (Step 1.1): $\widehat{T}_n \leftarrow \mathrm{I_i}$. Later $\widehat{T}_n$ would be forwarded to the $\mathcal{R}_{new}$ and consequently to the main server. If the verification in Step-2 is passed successfully, $\mathcal{R}_{new}$ commences reader authentication.

- Reader: It first decrypts Γ_j to retrieve tag data $\{\mathrm{T}, \widehat{T}_n, c\}$. $\mathcal{R}_{new}$ generates s', e' and hence calculates $z' \leftarrow r \cdot (c_i \cdot \pi_2(s) + \mathrm{T}) + e'$. Next it updates the session key $c_{i+1} \leftarrow c_i + e'$ and forwards (s', z') to the tag. Note that we assume $\mathcal{R}_{cur}$ is honest enough not to intercept the protocol transcript (s', z'), or $\mathcal{R}_{new}$ would forward (s', z') through some secret channel. Once this protocol transaction is executed successfully, both the parties update c_i in order to achieve *forward-secure* privacy.

- Tag: Since the protocol transcripts (s', z') in OT phase is different from that of AUTH phase (z'), a tag adopts OT phase for new reader authentication. Note that unlike AUTH phase, it calculates $\hat{e}$ from $\pi_2(s')$ and index I_i is not updated in OT phase. However, if the $\mathbf{wt}(\cdot)$ check passes, it updates the session key $c_{i+1} \leftarrow c_i + \hat{e}$ from $\hat{e}$ (not from e in AUTH phase).

Step-2: Let an owner $\mathcal{U}_c$ using reader $\mathcal{R}_{cur}$ intend to transfer ownership of m tags $\mathrm{T}^{\{1,\cdots,m\}}$ with previous ownership index $\widehat{T}^{\{1,\cdots,m\}}$ and new ownership index $\widehat{T}_n^{\{1,\cdots,m\}}$ to a new owner

$\mathcal{U}_n$ using reader $\mathcal{R}_{new}$ in the presence of an STP server $\mathcal{S}_i$. All the operating parties such as $\mathcal{R}_{cur}$, $\mathcal{R}_{new}$ and $\mathcal{S}_i$ share common secrets (P_n, P_c). In addition, $\mathcal{R}_{cur}$ and $\mathcal{R}_{new}$ have their own secrets resp. S_c and S_n. However, they also share a common secret key ρ for transferring tag related data after a successful verification by STP.

We use pseudo inverse matrix properties for key generation. Let $S_c \leftarrow X^+ \in \mathbb{Z}_2^{m\times n}$ be a pseudo-inverse of a matrix $X \in \mathbb{Z}_2^{n\times m}$ and $P_c \leftarrow XX^+ \in \mathbb{Z}_2^{n\times n}$. In the same way, we define $S_n \leftarrow Y^+ \in \mathbb{Z}_2^{m\times n}$ and $P_n \leftarrow YY^+ \in \mathbb{Z}_2^{n\times n}$.

- $\mathcal{R}_{cur}$ randomly generates 2 non-singular $n \times n$ matrices Q, V and send challenge Q to $\mathcal{R}_{new}$ as a challenge matrix.

- $\mathcal{R}_{new}$ will calculate ciphertext $E = P_c Q \in \mathbb{Z}_2^{n\times n}$ and by selecting the first column vector $q \in \mathbb{Z}_2^{n\times 1}$ of Q, it generates a signature $\alpha = q \cdot S_n \in \mathbb{Z}_2^{m\times 1}$. It then forwards E, α to the $\mathcal{S}_i$ for justification.

- Meanwhile, $\mathcal{R}_{cur}$ generates trapdoor $(C, D) := (VX^+, VX^+Q)$ on (S_c, Q) and sends (C, D) to $\mathcal{S}_i$ to justify.

- The STP server $\mathcal{S}_i$ checks $CE \overset{?}{=} D$. Note that $CE = VX^+XX^+Q = VX^+Q = D$ i.e., $X^+XX^+ = X^+$ from pseudo-inverse matrix properties. However, if the verification passes, $\mathcal{S}_i$ will forward signature α to $\mathcal{R}_{cur}$ for notification.

- $\mathcal{R}_{cur}$ ensures that $\mathcal{S}_i$ has justified the agreement by checking $\alpha P_n = qY^+YY^+ = qY^+ = \alpha$. Then it generates signature $\beta \leftarrow \alpha \cdot S_c = \alpha X^+$ by taking α as a challenge, encrypt tag data by the secret key ρ to output Γ. Then $\mathcal{R}_{cur}$ sends (Γ, β) to S_i.

- S_i checks whether Γ has been arrived from $\mathcal{R}_{cur}$ by checking $\beta P_c = \alpha X^+XX^+ = \alpha X^+ = \beta$. Then, STP generates a unique transaction ID $\mathbf{tid} \in \mathbb{Z}_p^*$ for each successful agreement between $\mathcal{R}_{cur}$ and $\mathcal{R}_{new}$ and securely shares it with Main Server. Finally, it forwards (Γ, $\mathbf{tid}$) to $\mathcal{R}_{new}$.

- $\mathcal{R}_{new}$ decrypts Γ to retrieve tag data and enters Step-3 to update ownership information at the main server. Meanwhile, it runs Step-1.2 to complete tag/reader authentication.

- Now S_i needs to update the shared key among $\mathcal{R}_{new}$ and $\mathcal{R}_{cur}$. It generates two random $n \times n$ matrix M, N and follow the session key update procedure described in [99].

- Finally, $\mathcal{R}_{new}$ and $\mathcal{R}_{cur}$ updates their key pair as $\{S_{(c/n)+1}, P_{(c/n)+1}\}$.

Step-3: $\mathcal{R}_{new}$ is responsible for updating ownership data on the trusted main server through a legitimate STP. We customize a HomSig scheme in [140] so that it fits our ownership transfer protocol. The application specifies global parameters $m \in \mathbb{N}$ such that $m \geq 1$. Not that, each owner in the system is registered with the Main Server with a shared secret k. Let new owner $\mathcal{U}_{new}$ want to register its ownership of ℓ tags ($\mathrm{T}^{\{1,\cdots,\ell\}}$,

Step-2: Reader-STP Communication

$\mathcal{R}_{\mathbf{cur}}(\mathcal{U}_c)$	IntServer $(\mathcal{S}_i)$	$\mathcal{R}_{\mathbf{new}}(\mathcal{U}_n)$
(P_c, P_n, S_c, ρ)	$\{P_n, P_c,\} \in \mathbb{Z}_2^{n\times n}$ $\{S_c, S_n\} \in \mathbb{Z}_2^{m\times n}$	(P_n, P_c, S_n, ρ)

Transfer Ownership:

$\mathrm{T}^{\{1,\cdots,m\}}$ with $\{\widehat{T}^{\{1,\cdots,m\}}, \widehat{T}_n^{\{1,\cdots,m\}}\}$

Non-singular $Q, V \xleftarrow{\$} \mathbb{Z}_2^{n\times n}$

$\xrightarrow{\quad Q \quad}$

$C := VS_c = VX^+$ $\qquad E := P_cQ = XX^+Q \in \mathbb{Z}_2^{n\times n}$

$D := CQ = VX^+Q$ $\qquad q \leftarrow Q^{1\times n}$

$\alpha := q \cdot S_n = qY^+ \in \mathbb{Z}_2^{m\times 1}$

$\xrightarrow{\quad C, D \quad}$ $\qquad \xleftarrow{\quad E, \alpha \quad}$

IF $(CE \neq D)$ return;

$\xleftarrow{\quad \alpha \quad}$

IF$(\alpha \cdot P_n \neq \alpha)$ return;

$\beta := \alpha S_c = \alpha X^+$

$\Gamma \leftarrow \mathcal{E}_\rho(\mathrm{T}^{\{1,\cdots,m\}}, \widehat{T}_n^{\{1,\cdots,m\}}, c^{\{1,\cdots,m\}})$

$\xrightarrow{\quad \Gamma, \beta \quad}$

IF$(\beta \cdot P_c \neq \beta)$
  return;

$\xrightarrow{\quad \Gamma, \mathbf{tid} \quad}$

Run Step-1.2

Generate non-singular $M, N \in_R \mathbb{Z}_2^{n\times n}$

$S_{c+1} = M \cdot S_c \in \mathbb{Z}_2^{n\times m}$

$S_{n+1} = N \cdot S_n \in \mathbb{Z}_2^{n\times m}$

where $rank(S_{c+1}) = (S_{n+1}) = m$

$S_{c+1}{}^+ := ({S_{c+1}}^T S_{c+1})^{-1} {S_{c+1}}^T \in \mathbb{Z}_2^{m\times n}$

$S_{n+1}{}^+ := ({S_{n+1}}^T S_{n+1})^{-1} {S_{n+1}}^T \in \mathbb{Z}_2^{m\times n}$

$P_{c+1} := [S_{c+1}] \cdot [S_{c+1}]^+ \in \mathbb{Z}_2^{n\times n}$

$P_{n+1} := [S_{n+1}] \cdot [S_{n+1}]^+ \in \mathbb{Z}_2^{n\times n}$

$P_c{}' := P_c \cdot M \in \mathbb{Z}_2^{n\times n}$

$P_n' := P_n \cdot M \in \mathbb{Z}_2^{n\times n}$

$\xleftarrow{\quad P_c{}' \quad}$ $\qquad \xrightarrow{\quad P_n' \quad}$

$S_{c+1} = (P_c{}'S_c) \in \mathbb{Z}_2^{m\times n}$ $\qquad S_{n+1} = (P_n{}'S_n) \in \mathbb{Z}_2^{m\times n}$

If *rank* $(S_{c+1}) \neq n$ $\qquad$ If *rank* $(S_{n+1}) \neq n$

return; $\qquad$ return;

$P_{c+1} = [S_{c+1}] \cdot [S_{c+1}]^+$ $\qquad P_{n+1} = [S_{n+1}] \cdot [S_{n+1}]^+$

resp. old index $\widehat{T}^{\{1,\cdots,\ell\}}$ and new index $\widehat{T_n}^{\{1,\cdots,\ell\}}$) from $\mathcal{U}_{cur}$ to the Main Server. $\mathcal{U}_{new}$ possesses a transaction ID **tid** provided by the STP. Each signer ($\mathcal{R}_{new}$) has its own key pair (pk, sk) for the HomSig scheme. Note that all the operations in the scheme are defined over F.

Key generation: Main Server generates a pair of keys for each reader $(pk, sk) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, m)$ in the system.

$\mathsf{KGen}(1^\lambda, m)$: Let $\mathbb{G}, \mathbb{G}_T$ be bilinear groups of prime order p such that $p < q$ and $e : \mathbb{G}\times\mathbb{G} \to \mathbb{G}_T$ is a bilinear map with $g \in \mathbb{G}$ as a generator. Let $k \xleftarrow{\$} \mathbb{Z}_p$ and $h, g_1, \cdots, g_m$, $h_1, \cdots, h_m \xleftarrow{\$} \mathbb{G}$. Set $K := g^k$ and output public key $pk := (p, g, K, h, g_1, \cdots, g_m, h_1, \cdots, h_m)$ and the secret key $sk := k$.

Ownership Transfer: On input a set of ℓ old index $\widehat{T}^{\{1,\cdots,\ell\}} \in F_q^m$ and new index $\widehat{T_n}^{\{1,\cdots,\ell\}} \in F_q^m$, $\mathcal{R}_{new}$ generates signature $\sigma_i \xleftarrow{\$} \mathsf{Sign}(\cdot)$ on behalf of $\mathcal{U}_{new}$.

$\mathsf{Sign}(sk, \mathbf{tid}, \widehat{T}^{(i)}, \widehat{T}_n^{(i)})$: $\mathcal{R}_{new}$ picks a random $t \xleftarrow{\$} \mathbb{Z}_p$ and compute the following:

$$\sigma_i := \left(h^t \prod_{i=1}^m g_i^{\widehat{T}^{(i)}} \prod_{i=1}^m h_i^{\widehat{T_n}^{(i)}}\right)^{\frac{1}{k_n+\mathbf{tid}}}$$
$$\Sigma_i := (\sigma_i, t)$$

where $k_n \in sk$ is the secret key of $\mathcal{R}_{new}$ and $\Sigma_i \in \mathbb{G} \times \mathbb{Z}_q$. Finally, it sends the tuple $(\mathbf{tid}, \Sigma_i, \mathcal{U}_{new}, \widehat{T}^{(i)}, \widehat{T}_n^{(i)})$ to the intermediate Server $\mathcal{S}_i$.

Intermediate Server: When an intermediate server $\mathcal{S}_i$ receives ℓ signatures from readers, it generates the combined signature $\Sigma \leftarrow \mathsf{CombSign}(\cdot)$.

$\mathsf{CombSign}(pk, \mathbf{tid}, \{(\widehat{T}^{(i)}, \widehat{T}_n^{(i)}, \Sigma_i, \eta_i)\}_{i=1}^{\ell}, \mathcal{U}_{new})$: First, it verifies every Σ_i as a valid signature on $(\widehat{T}^{(i)}, \widehat{T}_n^{(i)})$ with respect to **tid** by $\mathsf{VerSign}(\cdot)$ algorithm. Then, for each $i \in \{1, \cdots, \ell\}$, it randomly generates a coefficient $\eta_i \xleftarrow{\$} F$ and computes:

$$\widehat{T} = \sum_{i=1}^{\ell} \eta_i \cdot \widehat{T}^{(i)}, \quad \widehat{T_n} = \sum_{i=1}^{\ell} \eta_i \cdot \widehat{T_n}^{(i)}, \quad \sigma = \prod_{i=1}^{\ell} (\sigma_i)^{\eta_i}, \quad t = \sum_{i=1}^{\ell} \eta_i \cdot t_i \bmod p$$
$$\Sigma := (\sigma, t)$$

Finally, it forwards $(\mathbf{tid}, \widehat{T}, \widehat{T}_n, \Sigma, \mathcal{U}_{new})$ to the Main Server.

Main Server: When the server obtains the combined signature Σ on ℓ tags with the respect to transaction ID **tid**, it checks the validity of Σ by $\mathsf{VerSign}(\cdot)$ algorithm.

$\mathsf{VerSign}(pk, \mathbf{tid}, \widehat{T}, \widehat{T}_n, \Sigma, \mathcal{U}_{new})$: Let $\Sigma = (\sigma, t) \in \mathbb{G} \times \mathbb{Z}_p$. It returns 1 if Σ is a valid signature on $(\widehat{T}, \widehat{T}_n)$ with respect to the transaction ID **tid**, otherwise returns 0 by the following equation:

$$e(\sigma, K \cdot g^{\mathbf{tid}}) \stackrel{?}{=} e\left(h^t \prod_{i=1}^m g_i^{\widehat{T}^{(i)}} \prod_{i=1}^m h_i^{\widehat{T_n}^{(i)}}, g\right)$$

After successful verification, Server updates its database with new indexes $\widehat{T}_n^{(i)}$ to the $\widehat{T}^{(i)}$ for the new owner $\mathcal{U}_{new}$.

5.3.3 Security Analysis

Definition 18 *In the Man-In-the-Middle (MIM) attack, adversary $\mathcal{A}$ is allowed to eavesdrop both the connections tag-reader and reader-server, making the tag and the reader believe that they are talking directly to the reader and the server respectively over a secure connection, when in fact, the entire communication is controlled by $\mathcal{A}$. Then, $\mathcal{A}$ interacts with the server to authenticate. The goal of the attacker $\mathcal{A}$ is to authenticate successfully in Q rounds. $\mathcal{A}$ is successful if and only if it receives accept response from all Q rounds.*

Theorem 19 *If mapping function π_i is suitable for field F and the Field-LPN^F_w problem is (t, Q, ϵ)-hard then the authentication protocol is (t, Q, ε)-secure against **active adversaries**, where*

$$t' = t - Q.exp(F) \qquad \epsilon' = \epsilon + Q.2^{-\lambda} + s(\tau, \tfrac{1}{2})^{-n}$$

and $exp(F)$ is the time to perform $O(1)$ exponentiations in F.

Proof: We refer to the Ring-LPN based authentication work in [138] for detail proof.

Proposition 3 *The hardness of decisional exact-LPN is polynomially related to that of search LPN and the protocol has no completeness error $\epsilon_c(w, n) \approx 0$.*

Theorem 20 *If Sum of Subset (SSP) problem is NP-complete then Binary Matrix Factorization (BMF) problem is also NP-complete.*

Theorem 21 *If BMF is* hard, *then construction of the lightweight stateful signature in Step-2 is existentially unforgeable under a one-time chosen message attack (OT-CMA).*

Proposition 4 *Authentication protocol between the reader and tag is free from Man-In-the-Middle (MIM) attack.*

Proof: Authentication protocol from Field-LPN is not proved secure against MIM attack. By using a universal hash function described in [141], it can be converted to a MIM-secure scheme. However, our protocol is not vulnerable to MIM attack for the following reason. In order to recover the entire secret (T,c), an adversary $\mathcal{A}$ needs to repeat (MIM attack described in [138]) $\mathcal{O}(n)$ times[17] successful attacks and then to apply Gaussian elimination method. Since our protocol enjoys the advantage of session key c_i, it updates one of the secret keys c_i in each transaction (during authentication or ownership transfer). Consequently, it resists the adversary $\mathcal{A}$ to obtain $2n$ linearly-independent equations from the same secret key pair (T,c) and hence MIM attack. A very recent proposal in [142] claims an attack against Ring-LPN in [138]. Authors first describe a matrix varient of the Ring-LPN protocol and state their changes to reduce communication and computation complexity. They propose to query the Ring-LPN oracle repeatedly Q times with the same secret c in order to obtain a sequence of $(z_1, z_1c + e_1), (z_2, z_2c + e_2), \cdots, (z_Q, z_Qc + e_Q)$. Although the attack is not practical as they claimed but nevertheless can not break our protocol's security since we would update the secret key c in each session. However, space constraints inhibit us to present a full-blown proof here. It will appear in the full version of the work.

[17] To obtain $2n$ linearly-independent equations

We consider a potential MIM-attack scenario regarding ownership transfer protocol, while the old owner $\mathcal{R}_{cur}$, listening to the insecure channel between the new owner and the tag on Step-1.2 to compute the new session key c_{i+1}. Therefore, we assume $\mathcal{R}_{cur}$ to be *honest* minimum for a single protocol transaction just after the ownership transfer occurs. After that, both the new reader and tag would update their secret to retain forward privacy.

5.3.4 Privacy

One of the major privacy issues in Ownership transfer protocol is to satisfy previous owner and new owner privacy settings in terms of ownership transfer. In step 1.2. of the protocol the new reader is given the tuple $\{T, I_i, c_i\}$ which includes the tag's secret. But $\mathcal{R}_{new}$ immediately authenticates the tag and updates the session secret c_i. Subsequently, the authentication protocol is forward (resp. backward) privacy secure due to updating session key. That is why the new reader is unable to interpret the tag's previous communication and current reader cannot trace the tag after the ownership transfer even if the secret is transferred from the current reader to the new reader.

In order to define privacy, we analyzed our protocol according to the privacy framework based on *zero-knowledge* (ZK) formulation [123] where it is assumed that no secret will be revealed from the protocol transactions. This model rely on the unpredictability of the entity's (e.g., the tag) output in the protocol execution $\pi \leftarrow 2\lambda + 1$ *s.t.* $\lambda \geq 1$ (our case: $\pi = 3$ *s.t.* $\lambda = 1$).

Let $\hat{\mathcal{A}}$ be a PPT CMIM (Concurrent Man in the Middle) adversary equivalent to $\mathcal{A}$ (respectively, simulator Sim) that takes on input the system public parameters $\mathbf{Pub}_T$, the reader $\mathcal{R}$ and the set of tags $\widehat{\mathcal{T}}$; and interacts with $\widehat{\mathcal{T}}, \mathcal{R}$ via the oracles mentioned above. Let $\hat{\mathcal{A}}$ be composed of a pair of adversaries $(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2)$ and their corresponding simulators (Sim_1, Sim_2) for $\mathbf{Exp}_{\mathcal{A}}^{ZK}(\widehat{T})$ experiments with the above oracles.

Experiment $\mathbf{Exp}^{ZK}(\widehat{\mathcal{T}})$

- Initialize RFID system, the reader $\mathcal{R}$, the tag set $\widehat{\mathcal{T}}$ (s.t., $|\widehat{\mathcal{T}}| = l$) by $\mathbf{SetupTag}(\cdot)$
- let $\mathcal{O} \leftarrow \text{Launch}, \text{Dtag}, \text{STag}, \text{SReader}, \text{Ukey}, \text{Corrupt}$
- Real: $(\mathcal{T}, st) \leftarrow \hat{\mathcal{A}}_1^{\mathbf{DTag}}(\mathcal{R}, \widehat{\mathcal{T}}, \mathbf{Pub}_T)$
  Simulation: $(\mathcal{T}, st) \leftarrow Sim_1^{\mathbf{DTag}}(\mathcal{R}, \widehat{\mathcal{T}}, \mathbf{Pub}_T)$
  where $\mathcal{T} = \{T_{i_1}, T_{i_2}, \cdots, T_{i_\delta}\} \in \mathcal{T}$ s.t., $0 \leq \delta \leq l$
- $c \in_R C \leftarrow \{1, 2, \cdots, l - \delta)\}$ and $C = \widehat{\mathcal{T}} - \mathcal{T}$
  Real: $T_c = T_{i_c}$
  Simulation: c is unknown to Sim_2
- Real: $view \leftarrow \hat{\mathcal{A}}_2^{\mathcal{O}}(\mathcal{R}, \widehat{\mathcal{T}}, T_c, st)$
  Simulation: $sview \leftarrow Sim_2^{\mathcal{O}}(\mathcal{R}, \widehat{\mathcal{T}}, st)$
- Real: output (c, $view_{\hat{A}}$)
  Simulation: output (c, $sview_{Sim}$)

We assume that $\hat{\mathcal{A}}$ queries the challenger with $\mathbf{Exp}^{ZK}(\widehat{T})$ in the *read world* and *simulation* mode. Note that if $\delta = 0$, no challenge tag is selected and the number of clean tags $|C| = l - \delta$. ZK-privacy implies that adversary $\hat{\mathcal{A}}$ cannot distinguish any challenge tag T_c from any set C of tags. That's why, $\hat{\mathcal{A}}_1$ is used to output an arbitrary set C and to limit $\hat{\mathcal{A}}_2$ to blind access to a challenge tag from C. Therefore, the advantage of the adversary with security parameter κ to win the privacy game can be defined as

$$\mathbf{Adv}^{ZK}_{\mathcal{A},Sim,\mathcal{D}}(\kappa,\widehat{T}) = |\Pr[\mathbf{Exp}^{ZK}_{\hat{\mathcal{A}}}(c,l,view(.)=1)]-\Pr[\mathbf{Exp}^{ZK}_{Sim}(c,l,sview(.)=1)]| \le \epsilon$$

Definition 19 *RFID authentication protocol described in Step-1 satisfies the ZK-privacy in [123] security model if for any adversary $\hat{\mathcal{A}}$, there exist a simulator Sim such that for any distinguisher $\mathcal{D}$, $\mathbf{Adv}^{ZK}_{\mathcal{A},Sim,\mathcal{D}}(\kappa,\widehat{T})$ is negligible.*

Theorem 22 *From the Field-LPN problem, the protocol described in Step-1 satisfies ZK-privacy.*

Theorem 23 *An RFID authentication protocol described in* Step. 1. *is forward (resp., backward)-ZK private.*

5.3.5 Performance evaluation

We concentrate on the computationally weakest of the entities, the tag. Ring-LPN has an outstanding lower communication overhead targeting lightweight ultra constrained tags equipped with tiny CPUs e.g., EPC class tags (the price range of a few cents) [138]. Subsequently, we slightly modify the field version of the protocol to a mutually authentication protocol with less computation and communication complexity and to make the protocol MIM-free and to resist a very recent attack proposed in [142] against Ring-LPN.
Computation Requirement: Following exact-LPN version in [117] yields the completeness error $\varepsilon_c = 0$ (whereas $\varepsilon_c \approx 2^{-55}$ in [138]). Field-LPN as we followed can do sparse multiplication for π_i that takes $21k$ clock cycles while other multiplication requires $150k$. Time to build e from Ber^F_w need $3k$ clock cycle [138]. If we ignore EX-OR operation cost, we need approximately $345k$ clock cycle for mutual authentication and require 20 *ms* to respond at 2 MHz clock rate. This response time is sufficient in many application scenarios since a delay of 1 sec is often considered acceptable [138].

For anonymous verification by an STP requires 1 $(n \times n)$ matrix multiplication while the $\mathcal{U}_{cur}$ requires (2 matrix + 2 vector) multiplication and $\mathcal{U}_{cur}$ needs only (1 matrix + 2 vector) multiplication.

HomSig is comprised of only 1 group element in $\mathbb{G}$ and 1 element in $\mathbb{Z}_p$. In order to provide a typical security level of 2^{80}, we can set p a 170 bit prime number and then the element in $\mathbb{G}_1$ is 171 bits long. Then the aggregated signature size from the reader to other readers/server would be 42 bytes in total. Signing costs include a multi-exponentiation in $\mathbb{G}$ and verification requires to compute only two pairings, one exponentiation in $\mathbb{G}$.
Communication complexity: During reader-tag communication, the protocol requires 4 elements from field F and 1 λ-bit string for authentication while 2 λ-bit string for ownership transfer. However, reader-reader communication involves total $4n^2 + 2n$-bit for communication.

Table 5.6: Tag Resources and Security Comparison with HB family

Scheme	$\mathbf{P}_1$	$\mathbf{P}_2$	$\mathbf{P}_3$	$\mathbf{P}_4$	**Others**
Afifi et al.*'07* [115]	k_1, k_2	5 Encryption 3 PRNG	5	No	
Kuseng et al.*'10* [145]	I_n, I, s, c	2 PUF 1 LFSR 4 PRNG	2	No	$\star$
Cai et al.*'11* [143]	k, s	2 Hash 1 MAC	2	No	$\dagger$
Yang et al.*'11* [135]	k_1, k_2, k_3	3 Encryption	3	Yes	$\diamond$
Song et al.*'11* [134]	I, k, c	4 Hash 2 Encryption	4	No	$\star$
Kapoor et al.*'12* [137]	s, k_1, k_2	2 keyed Hash 2 PRNG	4	No	$\diamond$
Doss et al.*'13* [144]	I, s, r, n	3 mod-squaring 1 CRC 3 PRNG	7	Yes	$\star$
Our scheme	I, k, s	2 Field-LPN	3	Yes	$\dagger \diamond \ddagger\star$

P_1: Tag secret type,
P_2: Cryptographic techniques used on tag,
P_3: Number of Protocol transaction related to the tag,
P_4: Mutual Authentication,
*Includes EPC class compliance,
$^\diamond$TTP supported,
†Aggregated Signature,
‡Semi-trusted Server.
PRNG:= Pseudo Random Number Generator

Storage Requirement: All the parties in the protocol need to store the public parameters. However, a tag needs to store 3 secrets from F A reader requires to store the same for authentication. However, for ownership transfer it needs to store 3 keys for pseudo inverse matrix operation $(2n \cdot n + 1m \cdot n)$ bits, user identifiers it works for (1 element from F for each user), tag ownership index for a set of m tags (m elements from F) and 1 shared secret key for suitable encryption. Nevertheless, storage requirement for the tag can be expressed by $\mathcal{O}(1)$ while that is $\mathcal{O}(m)$ for the readers/server such that m is the number of tags in an RFID system.

5.3.6 Conclusion

This work presents a novel scalable RFID ownership transfer protocol leveraging the reader authentication phase based on a lightweight Field-LPN problem that can meet the hardware constraints of the EPC Class tags. Moreover, using an efficient homomorphic aggregated signature facilitates transferring ownership of a set of tags together without direct-attachment to a trusted main server that makes the protocol to be compliant with an inventory system context. Furthermore, our protocol enables ownership transfer with readers verification that preclude operating partners in an inventory management system from injecting fake products.

5.4 An RFID-enabled Path Authentication Protocol

5.4.1 Introduction

A Supply Chain Management (SCM) controls and manages all of materials and information in the logistics process from acquisition of raw materials to product delivery to the end user. This yields convenience and efficiency, which leads to productivity gains. With the growing nature of SCM, it is crucial to construct protocols that enable the end user to verify the security and privacy not only of the tags but also the path that the tag passes through. Therefore, path authentication in RFID-enabled SCM is important, for it helps defend product genuineness by ensuring product derivation. Hence, the integrity of a supply chain. More clearly, when a tag reaches to the end of its supply chain, it would be desirable that, if the authentication results of several intermediate readers could be accompanied by a cryptographic proof guaranteeing their correctness from the authentication tag, *s.t.*, no intermediate reader was omitted (or selected wrongly) by the tag, either deliberately or not.

In this work, we stress on static path based path authentication by Cai *et al.* [151][18], where the authors define a new combined privacy notion and provide an efficient solution for path authentication without sharing any secrets among supply chain parties.

Main Contribution. Our contribution includes the following:

- We instantiate a new variant of path authentication scheme with arithmetic circuit based HomMAC. Note that building blocks of previous static path based authentication systems were mainly from expensive elliptic curve ElGamal re-encryption (ECElGamal) and security of the schemes were primarily either from Pseudo Random Function (PRF) or Homomorphic MAC (HMAC). Security of our scheme also stems from PRF.

- We propose state update operations to be held inside the tag. It offers more security and reasonable privacy since the intermediate readers obtain no knowledge about current state of the tag. However, it introduces a lightweight computation (polynomial operation) in the tag. Note that likewise other existing schemes, it is also manageable (even easier) to update state information into the readers (and hence no computation inside the tag).

- We consider a relaxed privacy assumption[19] that allows adversary to query the reader during Move oracle. Since the reader in our scheme coveys no information about the tag's current state during tag movement, disallowing adversary to query only the tag is sufficient enough for the path privacy experiment to fail. This assumption is more practical and formal. Thus, we redefine the generic privacy oracles of Cai *et al.* [151] in section 4.1.

- Unlike the scheme in [151], we propose two strategies (with or without path information) for checkpoint verification that conform to a more stringent protection of path privacy in the supply chain.

[18] an extended and more practical privacy variant of [150].

[19] Adversary in [151] is not allowed to query either the reader or tag during Move operation run by the game challenger.

- Compare to [151], our scheme requires less storage but poses conditional scalability. However, it could be transformed into a fully scalable variant[20].

- We propose a polynomial based mutual authentication scheme from [160] that can optionally be integrated to our path authentication solution. We modify the protocol in order to conform secret and public parameters of our path authentication solution. In addition, we convert the existing *tag authentication* protocol to a *mutual authentication* protocol, significantly reduce communication, storage and computation overhead into the tag effectively.

- We purport how to accommodate a batch of tags that must follow the same path to the destination.

5.4.2 Supply Chain Management

In a SCM network every product that reaches an end user represents the cumulative effort of multiple parties like *manufacturer, distributor, wholesaler.* These parties are referred to collectively as the supply chain. Parties in a supply chain are *linked* together through information flows that allow various supply chain partners to coordinate the day-to-day flow of products up and down through the supply chain path. It can be represented as a Directed Acyclic Graph (DAG) $G = (V, E)$, where V is a set of nodes and E is a set of edges. Each edge $e \in E$, $e = (v_i, v_{i+1})$ s.t., $(v_i, v_{i+1}) \in V$ represents a *step* in the supply chain path. An RFID tag attached onto every product in the supply chain contains a unique identification about the product. A valid supply chain is a path (or a set of paths) in the DAG. Supply chain authentication is about verifying that an item (or rather, a cryptographic token, supposed to be attached to the item) is forwarded along a valid supply chain.

A valid finite path $P = (v_0, \cdots, v_r)$ is a pre-defined path set by the coordinator e.g., the manufacturer that an RFID-enabled product requires to follow, where v_0 is the entrance of a product to supply chain and v_r is its final destination to arrive. An RFID-enabled SCM consists of an issuer (e.g., manufacturer) $\mathcal{M}$, a set of check points (e.g., retailers) $\mathcal{D}$, a set of ordinary readers (e.g., distributors, wholesalers etc.) $\mathcal{R}$, and a set of tags $\mathcal{T}$. $\mathcal{M}$ initializes the whole system by providing identifiers to the $\mathcal{R}, \mathcal{T}$ and storing necessary information into the tag and reader. Each reader in the path provides the contents to run status update operation inside the tag, while it moves through a supply chain. Once the tag arrives at any of the checkpoints $\mathcal{D}$, it can check the validity of the *tag* as well as the *path* it followed from the $\mathcal{M}$ to $\mathcal{D}$. More precisely, the system has the following functions:

- **Initialize**(λ): Given the security parameter λ, an SCM system defines a supply chain network G including an issuer $\mathcal{M}$, a set of d checkpoints $\mathcal{D}$, a set of n tags $\mathcal{T}$, a set of r ordinary readers $\mathcal{R}$, and a set of v valid paths $\mathcal{P}_v$.

- **Reader Authentication** ($\mathcal{R}_j$): This function transforms the identity information $ID_{\mathcal{R}_j}$ of the reader $\mathcal{R}_j$ to the tag. We assume that the tag along the path to be honest (without mutual authentication), that means, it accepts data from the reader only after successful authentication and updates its internal state st thereby.

[20] postponed to the full version of the work.

- **Tag Evaluation**($\mathcal{T}_i$): A function that incorporates the new reader's information $ID_{\mathcal{R}_j}$ into the tag $\mathcal{T}_i$ in order to update the internal state $st_{\mathcal{T}_i}$ of the tag $\mathcal{T}_i$.
- **Verification** ($st_{\mathcal{T}_i}$): This function verifies whether a certain $\mathcal{T}_i$ has followed a valid path P_v and returns *True*. Otherwise it returns *False*.

System parameter	$(\lambda, \mathcal{F}, h, K, s, p, f, \mathfrak{f}, b)$ p is a prime of λ bits Polynomial $\mathfrak{f} := \{\mathfrak{f}_1(\alpha, \beta), \mathfrak{f}_2(\alpha, \beta), \cdots, \mathfrak{f}_m(\alpha, \beta)\}$ Hash $h : \{0,1\}^\lambda \rightarrow \mathbb{Z}_p$ Pseudo Random Function $\mathcal{F}_K : \{0,1\}^\lambda \rightarrow \mathbb{Z}_p$ Arithmetic circuit f: $\mathbb{Z}_p^r \rightarrow \mathbb{Z}_p$, where $\lvert f \rvert = r$
Initialize Tag $\mathcal{T}_i$ Without `Auth` (y_0, y_1, f) With `Auth` $(y_0, y_1, Q, b, f, \mathcal{T}_i, \mathfrak{f})$	$y(z) = y_0 + y_1 z$ **s.t,** $\sigma = (y_0, y_1)$ $y_0 = h(\texttt{tag\_data})$ $\mathcal{T}_i = \mathcal{F}_K(\iota_0)$ s.t., $\iota_0 =$ `Tag ID` $y_1 = (\mathcal{T}_i - y_0)/s \texttt{ mod } p$ $Q \leftarrow \mathsf{max}(\ \lvert f \rvert,\ (b-1)m^2 + m)$
Initialize Readers ($\mathcal{R}_1, \cdots, \mathcal{R}_r$) Without `Auth`$(y_0^j, y_1^j, K, s, p, \mathfrak{f}, h\mathcal{F})$ With `Auth`$(y_0^j, y_1^j, \mathcal{T}_i, y_0^{\mathcal{T}_i}, K, s, p, \mathfrak{f}\ , h, \mathcal{F})$	$y^j(z) = y_0^j + y_1^j z$ s.t, $\sigma_j = (y_0^j, y_1^j)$ $y_0^j = h(\texttt{reader\_data})$ $y_1^j = (\mathcal{F}_K(\iota_j) - y_0^j)/s \texttt{ mod } p$ s.t., $\iota_j =$ `Reader ID` $y_1^{\mathcal{T}_i} = y_0$ of $\mathcal{T}_i$ (`Tag ID` ι_i)
Initialize Checkpoint $\mathcal{D}_k$ Without `Path`-info $(s, \tau, \Lambda, \sigma)$ With `Path`-info $(s, p, f, K, \tau, \mathcal{F}, \{\iota_0, \cdots, \iota_r\}, \sigma)$	$\tau = y_0$ where $\sigma_{i,r} = (y_0, \cdots, y_d)$ ($\sigma_{i,r}$ is evaluated by $\mathcal{T}_i$ with $\mathcal{R}_r$) $\Lambda = f(\eta_0, \cdots, \eta_r)$ where $\eta_i = \mathcal{F}_K(\iota_i)$ and $\iota_i \leftarrow\ P = \{\iota_0, \cdots, \iota_r\}$

Figure 5.2: Path Authentication Initialization

5.4.3 Protocol Construction

We propose a privacy preserving path authentication protocol. We assume the supply chain path of a certain product is pre-determined (static) by the manufacturer. Each tag $\mathcal{T}_i$ conveys its identity information (a 1-degree polynomial), a path code f (gate sequence of the arithmetic circuit). We employ a homomorphic message authentication code (HomMAC) with labelled program and a one-way PRF scheme as building blocks of the protocol.

Path Authentication Protocol:

Consider a real-life scenario where a tag-enabled product traverses an automated supply chain, the tag is scanned at multiple locations: the manufacturer, logistics carrier, distribution centers, wholesalers and retailers etc. Assume a supply chain path authentication system consists of a manufacturer $\mathcal{M}$, a set of n tags $\mathcal{T}$, a set of d checkpoints $\mathcal{D}$, and a set of r intermediate readers $\mathcal{R}$. Readers in the supply chain are *semi-honest*,

```
**Reader** $(s, y_0, \mathcal{T}_i, K \in \mathbb{Z}_p)$                          **Tag** $(b, y_0, y_1, Q, \mathcal{T}_i \in \mathbb{Z}_p)$
$\mathcal{T}_i \leftarrow \mathcal{F}_K(\texttt{tag\_id})$                         $\mathfrak{f} := \{\mathfrak{f}_{1,\mathcal{T}_i}(\beta), \mathfrak{f}_{2,\mathcal{T}_i}(\beta), \cdots, \mathfrak{f}_{m,\mathcal{T}_i}(\beta)\}$
$\mathfrak{f} := \{\mathfrak{f}_1(\alpha, \beta), \mathfrak{f}_2(\alpha, \beta), \cdots, \mathfrak{f}_m(\alpha, \beta)\}$
-----------------------------------------------------------------------------------------------
$r \leftarrow h(\texttt{cur\_date})$
                    $\xrightarrow{r}$
                                        IF $(Q = 0)$
                                          For$(j := 1$ to $b)$
                                              Generate $z_j \in_r \mathbb{Z}_p$
                                        Else
                                          Choose randomly $\mathfrak{f}_k(\mathcal{T}_i, \beta) = \mathfrak{f}_{k,\mathcal{T}_i}(\beta) \in \mathfrak{f}$
                                          $\beta = (r + y_0) \texttt{ mod } p$ where $r, y_0 \in \mathbb{Z}_p$
                                          Compute $r' := \mathfrak{f}_i(\mathcal{T}_i, \beta)$
                                          For$(j := 2$ to $b)$
                                            Generate $z_j \in_r \mathbb{Z}_p$
                                          Randomly add $r'$ in the list $z_2, \cdots, z_b$
                                          $Q := Q - 1$
                    $\xleftarrow{\{z_1, \cdots, z_b\}}$
For$(k := 1$ to $m)$
For$(j := 1$ to $b)$
  Solve to match $z_j \overset{?}{=} \mathfrak{f}_k(\mathcal{T}_i, (r + y_0) \texttt{ mod } p)$
IF$(\texttt{no\_match})$
   Abort
Else
  $y_1^{(r)} = \mathcal{F}_K(\texttt{reader\_id}) - r)/s \texttt{ mod } p$
  $y_1^{(t)} = (\mathcal{T}_i - y_0)/s \texttt{ mod } p$
  $z^{(t)} := \mathfrak{f}_k(\mathcal{T}_i, (r + y_0 + y_1^{(t)}) \texttt{ mod } p)$
For$(j := 2$ to $b)$
  Generate $z_j \in_r \mathbb{Z}_p$
Randomly add $z^{(t)}$ in the list $\{z_2, \cdots, z_b\}$
                    $\xrightarrow{\{z_1^{(t)}, \cdots, z_b^{(t)}\}, y_1^{(r)}}$
                                            For$(j := 1$ to $b)$
                                              IF $z_j^{(t)} \neq \mathfrak{f}_k(\mathcal{T}_i, (\beta + y_1) \texttt{ mod } p)$
                                                Abort;
                                              Else
                                                Run Tag Evaluation algorithm by $y_1^{(r)}$
```

Figure 5.3: Integrating Tag Authentication

independent and have no knowledge about the path P. More clearly, a reader $\mathcal{R}_i$ in a valid path P_v follows protocol transaction correctly on tags. For building construction, we adapt the practical HomMAC described in [158] but customized to work with our path authentication scheme. Security of the scheme relies on the security of one-way function (PRF).

We divide our path authentication protocol in three steps: Initial setup, Tag evaluation, Verification. Initially, $\mathcal{M}$ sets up the whole system and stores the necessary protocol data into the tag, checkpoints and intermediate readers. Tags then get into the supply chain system and proceed towards the intended path. However, a tag would update its status as it comes across a new reader during its journey towards the destination checkpoint. Finally, the tag's evaluated data would be justified by the checkpoint in order to validate a certain path.

Initial Setup: $\mathcal{M}$ first chooses a PRF $\mathcal{F}_K : \{0,1\}^* \to \mathbb{Z}_p$ where K is the seed of $\mathcal{F}$ and p, a λ-bit prime number. Then $\mathcal{M}$ runs $\mathsf{KGen}(1^\lambda)$ and outputs $(\mathsf{sk}, \mathsf{ek}) = (\{K, s\}, p)$ where $s \in \mathbb{Z}_p$. $\mathcal{M}$ stores sk to the readers and checkpoint and ek to the tag.

We consider all the entities (e.g., $\mathcal{T}$, $\mathcal{R}$) possess unique ID or label $\iota_i \in \{0,1\}^\lambda$. The supply chain path from the manufacturer to the checkpoint is defined by $(\iota_0, \cdots \iota_r)$ where ι_0 is the tag's ID and $(\iota_1, \cdots \iota_r)$ are the IDs of intermediate readers $(1, \cdots, r)$ and an arithmetic circuit $f : \mathbb{Z}_p^r \to \mathbb{Z}_p$.

Modern efficient inventory control policy includes exact knowledge of the flow of products: the amount of inventory at each location, predicted arrival date of an item etc. We address the issues in our solution. Let `reader_data`, `tag_data` be a certain reader and tag's meta data respectively. For instance, `reader_data` may include information about the expected arrival date, location etc., while tag data includes manufacture date, description of the product etc.

$\mathcal{M}$ defines a secure hash function $h : \{0,1\}^* \to \mathbb{Z}_p$ that converts any meta data to $\mathbb{Z}_p$. $\mathcal{M}$ provides h to $\mathcal{R}$ and store $h($`tag_data`$)$ in $\mathcal{T}$. However, $h($`reader_data`$)$ will be used as a *nonce* (during mutual authentication). In addition, both $h($`reader_data`$)$ and $h($`tag_data`$)$ will be used as a constant part (y_0) of the polynomial $y(z)$.

Every entity in the system (tags, readers) will be represented by a 1-degree polynomial $y(z) = y_0 + y_1 z, \quad y \in \mathbb{Z}_p[z]$ where $y_0 = h($`tag_data`$)$ or $h($`reader_data`$)$, $y_1 = (\mathcal{F}_K(\iota) - y_0)/s \bmod p$ and outputs coefficients of the polynomial $y(z)$, that is, $\sigma = (y_0, y_1)$.

For each $\mathcal{T}_i$, $\mathcal{M}$ generates $y^{i0}(z) = \sum_j y_j^{i0} z^j$ where $\sigma_{i0} = (y_0^{i0}, y_1^{i0})$ and sets initial state $st_0 := \sigma_{i0}$ of the path. $\mathcal{M}$ sets secrets an evaluation key **ek** and a path code f where $\mathsf{size}(f) = |P|$. $\mathcal{M}$ computes $\tau \leftarrow f(y_0^{i0}, \cdots, y_0^{ir})$ and shares τ with the checkpoint $\mathcal{D}$.

Tag Evaluation: As the product moves through the path P, $\mathcal{T}_i$ updates the path state st_j. When a tag $\mathcal{T}_i$ reaches $\mathcal{R}_j$, $\mathcal{R}_j$ runs $\mathsf{Authentication}(\mathsf{sk}, \iota, \tau)$ algorithm to compute $\sigma_{ij} = (y_0^{ij}, y_1^{ij})$ and forwards σ_{ij} to the tag $\mathcal{T}_i$ to update current state st_j.

Upon receiving σ_{ij} from $\mathcal{R}_j$, $\mathcal{T}_i$ runs the $\mathsf{Evaluate}(\mathsf{ek}, f, \sigma)$ algorithm and evaluates the existing circuit f on $\{\sigma_{i(j-1)}, \sigma_{ij}\}$ according to the current secret gate:

- If current gate is '+': $\mathcal{T}_i$ evaluates the new polynomial $y(z) = y^{j-1}(z) + y^j(z)$. Let d^j be the maximum degree of a polynomial $y^{j-1}(z)$, then coefficient of $y(z)$ will be $\sigma_{ij} \leftarrow (y_0^j, \cdots, y_d^j)$ where $d = \mathtt{max}(d^j, d^{j-1})$. Since $y^j(z)$ is always 1-degree polynomial, it is obvious that $d^{j-1} \geq d^j$. Note that the degree of $y(z)$ remain fixed after evaluating addition gate.

- If the current gate is '×': $\mathcal{T}_i$ evaluates new polynomial $y(z) = y^{j-1}(z) \times y^j(z)$ and determines the coefficients of $y(z)$ as $\sigma_{ij} \leftarrow (y_0^j, \cdots, y_d^j)$ where $d = d^j + d^{j-1}$. Note that the degree of $y(z)$ increases by 1 after evaluating multiplication gate.

Finally, $\mathcal{T}_i$ stores $\sigma_{ij} := (y_0^j, \cdots, y_d^j)$ as the current state st_j.

Verification at the Checkpoint: $\mathcal{T}_i$ arrives at the destination checkpoint $\mathcal{D}$ with $st_r = \sigma_{i,r} = (y_0, \cdots, y_d)$. Now $\mathcal{D}$ verifies whether $\mathcal{T}_i$ has followed a valid path P_v by using $\mathsf{Verify}(\mathsf{sk}, \tau, \mathcal{P}, \sigma)$ algorithm. We consider two variants of verification process. First where $\mathcal{D}$ knows the path traversed $(\iota_1, \cdots, \iota_r)$ by a tag $\mathcal{T}_i$. Alternatively, where $\mathcal{D}$ has no knowledge of the path P_v (due to strict privacy).

Case-1: When $\mathcal{D}$ knows the valid path P_v of a tag $\mathcal{T}_i$:

- Check $y_0 \overset{?}{=} \tau$. If it outputs 1 (success), go to the next step.
- For every $\iota_i \in \mathcal{I}$, compute $\eta_i = \mathcal{F}_K(\iota_i)$
- Evaluate the circuit $f = \{o_1, o_2, \cdots, o_r\}$ on $\eta_0, \cdots, \eta_r$ s.t., $\Lambda = f(\eta_0, \cdots, \eta_r)$
- Evaluate the equation on $\sigma_{i,r}$ and check whether the following holds:

$$\Lambda \overset{?}{=} \textstyle\sum_{\ell=0}^{d} \quad y_\ell s^\ell$$

Output 1 (accept) if true, else output 0 (reject).

Case-2: If $\mathcal{D}$ has no knowledge of the path P_v of a tag $\mathcal{T}_i$, $\mathcal{M}$ does not need to share $\mathcal{P} \leftarrow (f, \mathcal{I})$, PRF $\mathcal{F}$, and K, instead it shares Λ with $\mathcal{D}$. Then the verification algorithm will look like $\mathsf{Verify}(\mathsf{sk}, \tau, \Lambda, \sigma)$ where $\mathsf{sk} = \{s\}$. and proceeds as follows:

- Check $y_0 \overset{?}{=} \tau$. If it outputs 1 (success), go to the next step.
- Evaluate the equation on $\sigma_{i,r}$ and check whether the following holds:

$$\Lambda \overset{?}{=} \textstyle\sum_{\ell=0}^{d} \quad y_\ell s^\ell$$

Output 1 (accept) if true, else output 0 (reject).

An Example: We illustrate our homomorphic path authentication system with a small and simple example. Suppose the manufacturer $\mathcal{M}$ initializes a tag T and a path with 3 intermediate readers $R = (R_1, R_2, R_3)$ with system parameters $p = 23$, secret $x = 4$. For simplicity, let $h(\texttt{tag\_data})$ be 1, $h(\texttt{reader\_data})$ of (R_1, R_2, R_3) be $(2, 3, 4)$, unique identifier labels of (T,R) and corresponding PRF output be $(\iota_0, \iota_1, \iota_2, \iota_3)$ and $(5, 10, 19, 12)$ respectively. Now we can construct 1-degree polynomials with coefficient $\sigma \leftarrow (y_0, y_1)$ for (T,R) according to the following:

- Tag T: $y^0(z) = 1 + ((5-1)/4 \texttt{ mod } 23)\ z = 1 + z$ s.t., $\sigma_0 = (1, 1)$
- Reader R_1: $y^1(z) = 2 + ((10-2)/4 \texttt{ mod } 23)\ z = 2 + 2z$ s.t., $\sigma_1 = (2, 2)$

- Reader R_2: $y^2(z) = 3 + ((19-3)/4 \texttt{ mod } 23)\ z = 3 + 4z$ s.t., $\sigma_2 = (3, 4)$
- Reader R_3: $y^3(z) = 4 + ((12-4)/4 \texttt{ mod } 23)\ z = 4 + 2z$ s.t., $\sigma_3 = (4, 2)$

Let T possess a secret path code $f :=$ '$\times$++' or '100'. $\mathcal{M}$ computes $\tau(= 1\times2+3+4 = 9)$ by using the circuit f and shares τ with checkpoint D. As T moves through the valid path, it executes evaluation algorithm on f. Evaluation proceeds *gate-by-gate* as follows.

- On arrival R_1, for gate '$\times$': $y^{01}(z) = y^0(z) \times y^1(z) = 2 + 4z + 2z^2$
- On arrival R_2, for gate '+': $y^{012}(z) = y^{01}(z) + y^2(z) = 5 + 8z + 2z^2$
- On arrival R_3, for gate '+': $y^{0123}(z) = y^{012}(z) + y^3(z) = 9 + 10z + 2z^2$

As T arrives at checkpoint D, it first checks $y_0 \overset{?}{=} \tau(= 9)$. Then it computes $\rho = f(5, 10, 19, 12) = 5 \times 10 + 19 + 12 = 81$ (by using $\mathcal{F}$ and identifiers $(\iota_0, \iota_1, \iota_2, \iota_3)$) and checks whether the following equation holds:

$$\rho \overset{?}{=} \textstyle\sum_{k=0}^{2} y_k x^k = 9 + 10.4 + 2.4^2 = 81 \text{ (for } 9 + 10z + 2z^2\text{)}$$

Integrating Mutual Authentication:

Path authentication protocol cannot resist desynchronization, tag impersonation, or replay attack without mutual authentication. For instance, it is sufficient for an adversary to capture a protocol message from an honest reader and later replay it to the tag with counterfeit message to update current path state st. To address the above-mentioned attacks, we propose to extend our path protocol with a mutual authentication protocol in Fig. 2. We adopt polynomial-based authentication protocol described in [160] with major modifications (e.g., mutual authentication).

Unlike [160], we use two tag parameters $(\mathcal{T}_i, y_0)$ as secret, 1-degree bivariate set of polynomials $\mathfrak{f}$, no hash function in the tag, only b random numbers between the tag and reader. Reader initiates the protocol with $h(\texttt{reader\_data})$, the tag follows the protocol transcripts in [160]. Upon receiving the feedback, the reader authenticates the tag and forwards $y_1^{(r)}$ (to update path status) and $z_i^{(t)}$ (to authenticate the reader) to the tag. Note that the tag would update current status st_j only if it can authenticate the reader successfully.

Batch Initialization: In [152], authors introduce path verification of a *batch of tags* that share the same path. However, we can accommodate the same construct in our protocol. Let a supply chain enrol a batch of n tags where each tag $\mathcal{T}_i$ is represented by a 1-degree polynomial $y^{(i)}(z) = y_0 + y_1^{(1)}z$. Since all the tags convey same meta data, they share same y_0. After initializing the batch of tags, $\mathcal{M}$ evaluates the circuit on polynomials $y^{(i)}(z)$, $1 \le i \le n$ by using $\mathsf{Evaluate}(\mathsf{ek}, f_b, \sigma_b)$ algorithm, where $|f_b| = n - 1$

and $\sigma_b = \{\sigma_1, \cdots, \sigma_n\}$. Then it initializes the batch of $\mathcal{T}_i$ with the evaluated polynomial and releases it into the system. Meanwhile $\mathcal{M}$ shares necessary information (f_b, σ_b) with the checkpoints $\mathcal{D}$ for verification.

5.4.4 Security Analysis

Security of Path authentication. Security of our path authentication scheme relies on the homomorphic message authentication code (HMAC) in [158] and subsequently the security of the HMAC relies only on a Pseudo Random Function (PRF). Let there exist $L := (\iota_1^*, \tau_1), \cdots, (\iota_r^*, \tau_r)$. A labeled program $\mathcal{P}^* = (f^*, \iota_1^*, ..., \iota_r^*)$ is *well defined* on L, if there exist $(\iota_i^*, *) \notin L$, there is $f^*(\tau_j \leftarrow (\iota_j, \tau_j) \in L \cup \hat{\tau}_j \leftarrow (\iota_j, *) \notin L)$ that provides same output irrespective of choosing $\hat{\tau}_j$.

Consider an experiment $\mathsf{Exp}_{\mathcal{A},\mathsf{HMAC}}(\lambda)$ with a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

- Setup. The challenger generates $(\mathsf{sk}, \mathsf{ek}) \leftarrow \mathsf{KGen}(1^\lambda)$ and gives ek to $\mathcal{A}$. It also initializes a list $L = \{\emptyset\}$.

- $\mathsf{Authentication\ queries}$. $\mathcal{A}$ can adaptively ask for label-message pair (ι, τ) s.t., $\tau \leftarrow h(\mathtt{date})$ of its choice. If $\mathcal{C}$ receives any query (ι, τ) that is available in the list (s.t., $(\iota, *) \in L$), it simply ignores the query and feedback with the (ι, τ) as it received before. Else, it runs $\sigma \leftarrow \mathsf{Authentication}(\mathsf{sk}, \iota, \tau)$ algorithm, forwards σ to $\mathcal{A}$, and updates $L = L \cup (\iota, \tau)$.

- $\mathsf{Forgery}$. Alike verification query $(\tau^*, \mathcal{P}^*, \sigma^*)$, adversary $\mathcal{A}$ is allowed to output a forgery $(\tau^*, \mathcal{P}^* = (f^*, \iota_1^*, ..., \iota_r^*), \sigma^*)$.

- $\mathsf{Verification\ queries}$. Given a query $(\tau, \mathcal{P}, \sigma)$ by $\mathcal{A}$, $\mathcal{C}$ replies with either 1 (accept) or 0 (reject) by using algorithm $\mathsf{Verify}(\mathsf{sk}, \tau, \mathcal{P}, \sigma)$.

The experiment $\mathsf{Exp}_{\mathcal{A},\mathsf{HMAC}}(\lambda)$ outputs 1 if and only if $\mathsf{Verify}(\mathsf{sk}, \tau, \mathcal{P}, \sigma) = 1$ and one of the following conditions holds:

- *Type 1 Forgery:* $\mathcal{P}^*$ is not well-defined on L.

- *Type 2 Forgery:* $\mathcal{P}^*$ is well defined on L and τ^* is not the correct output of $\mathcal{P}^*$ s.t., $\tau^* \notin f^*(\tau_j \leftarrow (\iota_j, \tau_j) \in L)$.

Two major improvements of HMAC by [158] that constitute our protocol are: allowing adversary to query verification oracle and adapting the definition of forgery. Since tag/reader IDs are unique, HMAC scheme does not allow re-using a label (ι) to authenticate input data $h(\mathtt{date})$ in order to track authenticated inputs uniquely. The notion of well defined programs is to define an adversary generated tuple (ι_j, τ_j) as *forgery.* That is why, even if the adversary trivially modify the circuit f by adding dummy gates and inputs, it does not violate security requirements. Verifier ensures that either $\mathcal{P}$ is run on valid inputs $(\iota_j, \tau_j) \in L$, otherwise, $(\iota_j, \tau_j) \notin L$ do not affect computation process in anyway.

Notice that in **Case-2**, *Checkpoint verification* of our protocol, we disallow the manufacturer $\mathcal{M}$ to share $\mathcal{P}$ with checkpoints (to obtain more privacy), instead $\mathcal{M}$ shares $\Lambda \leftarrow f(\eta_0, \cdots, \eta_r)$. However, it constitute an infringement to the security of the protocol. Because it will allow the adversary to modify $\mathcal{P}$ in such a way (e.g., adding dummy gates and inputs so that output remains same) that the modified circuit will remain equivalent to the previous one semantically. More clearly, scheme in [158] defines *well defined program* and hence *forgeries* without considering any tuples $(\iota_j, *) \notin L$ for verification. Therefore, for **Case-2**, we consider slightly weaker assumption, that is, we will not allow the adversary $\mathcal{A}$ to modify the circuit anyway, that will best match with the *Type 2 Forgery* definition of Gennaro and Wichs in [159].

Correctness: An authentication tag σ can correctly authenticate a message τ under a set of label identifiers ι if

$$\Pr\Big[\mathsf{Verify}(\mathbf{sk}, \tau, \mathcal{P}, \sigma) = \mathsf{accept} \;\; | \; (\mathbf{ek}, \mathbf{sk}) \leftarrow \mathsf{KGen}(1^\lambda),\, \sigma \leftarrow \mathsf{Authentication}(\mathbf{sk}, \iota, \tau)\Big] = 1$$

where $\mathcal{P}$ is the identity program on a label $\iota \in \mathcal{I}$ with circuit f.

Our scheme consider a special 1-degree polynomial for a certain tag $\mathcal{T}_i$ s.t., $y^0(z) = y_0^0 + y_1^0 z$ where $y^0(0) = \tau$ and $y^0(x) = \eta_0 = \mathcal{F}_K(\iota_0)$. To preserve homomorphic property this is also followed by the intermediate readers $\mathcal{R}_j$ for evaluating the circuit $f : \mathbb{Z}_p^r \rightarrow \mathbb{Z}_p$ over $y^1, \cdots, y^r$. If a set of r triples $\{\tau_i, \mathcal{P}_i, \sigma_i\}$ such that $\mathsf{Verify}(\mathbf{sk}, \{\tau_i, \mathcal{P}_i, \sigma_i\}) = \mathsf{accept}$ then

$$\Pr\Big[\mathsf{Verify}(\mathbf{sk}, \tau^*, \mathcal{P}^*, \sigma^*) = \mathsf{accept} \;\; | \; \tau^* = f(\tau_1, \cdots, \tau_r), \mathcal{P}^* = f(\mathcal{P}_1, \cdots, \mathcal{P}_r), \sigma^* = \mathsf{Evaluate}(\mathbf{ek}, f, (\sigma_1, \cdots, \sigma_r)\Big] = 1$$

This definition briefly explains the correctness of the evaluation over the data.

Succinctness: The size of authentication tag $\texttt{size}(\sigma)$ is bounded by a fixed $\texttt{poly}(\lambda)$, where λ is a security parameter, irrespective of the input size of the arithmetic circuit f.

Theorem 24 *Let $\mathcal{A}$ be a (Q, t, ϵ)-PPT adversary that can query a tag Q times $(m < Q$. Then the probability that $\mathcal{A}$ can successfully recover any polynomial of a tag in time t is*

$$\Pr[Adv_{\mathcal{A}}^{NPI(Q/m, mb-m+1, 1)}] \leq \epsilon.$$

Maximum number of queries allowed for a tag Q_{max} is

$$Q_{max} \approx ((b-1)m^2 + m).$$

Proof: We assume the maximum degree of α and β in a polynomial $\mathfrak{f}_i$ is 1 $(k = 1)$. We refer to the polynomial based authentication work in [160] for detail proof.

Theorem 25 *The homomorphic authentication based protocol described in this work is secure if and only if the Pseudo Random Function (PRF) $\mathcal{F}$ is secure.*

Proposition 5 Let a PPT adversary $\mathcal{A}$ has compromised n tags targeting to recover $\mathfrak{f} := \mathfrak{f}_1, \cdots, \mathfrak{f}_m$. Hence the probability that $\mathcal{A}$ can compute $\mathfrak{f}_i$ is:

$$\Pr[Adv_{\mathcal{A}}] \leq m^{-k-2}.$$

where k is the maximum degree of $\mathfrak{f}_i$ and $n \geq (k+1)^2/k$

Proof: If $\mathcal{A}$ knows $\mathcal{T}_i$ of n tags, then using Sudan *et al.* algorithm in [157] could recover $\mathfrak{f}_1, \cdots, \mathfrak{f}_m$. However, if the tags are assumed to be *tamper-proof*, then no algorithm can recover $\mathfrak{f}$ efficiently.

Let n tags be compromised and the adversary $\mathcal{A}$ obtains the univariate polynomials $(\mathfrak{f}_{1,\mathcal{T}_i}(\beta), \cdots, \mathfrak{f}_{m,\mathcal{T}_i}(\beta))$. Target of $\mathcal{A}$ is to recover the bivariate polynomial $(\mathfrak{f}_1(\alpha, \beta), \cdots, \mathfrak{f}_m(\alpha, \beta))$. We can express polynomial assignment in matrix form. Let $X \in \mathbb{Z}_p^{n \times k+1}$ be a matrix with secret $\mathcal{T}_i$ of n tags and $S_1, \cdots, S_m$ where $S_i \in \mathbb{Z}_p^{(k+1)\times(k+1)}$ is the matrix representation of bivariate polynomial $\mathfrak{f}_i(\alpha, \beta)$ stored in the server. Let $Y_i = XS_i \in \mathbb{Z}_p^{n\times(k+1)}$, then univariate polynomials assigned to tag i are $Y_1[i], \cdots, Y_m[i]$. Let $\mathcal{A}$ know Y_i and intend to recover S_i. A necessary condition to solve the problem is $n \geq (k+1)^2/k$ [160]. Since a tag selects $\mathfrak{f}_i$ from randomly $\mathfrak{f}_1, \cdots, \mathfrak{f}_m$, probability of $\mathcal{A}$ to recover $\mathfrak{f}_i$ is

$$\Pr[Adv_{\mathcal{A}}] = 1/m^{n-1} \leq m^{-k-2} \texttt{ i.e.}, n \geq (k+1)^2/k$$

Note that, unlike authentication scheme in [160], we consider 1-degree polynomial for $\mathfrak{f}_i$ in our scheme. Therefore, probability of $\mathcal{A}$ to recover $\mathfrak{f}_i$ is $\Pr[Adv_{\mathcal{A}}] \leq 1/m^3$ in our case. However, in order to keep the system remain secure, we could increase m and/or k (to lower $Adv_{\mathcal{A}}$). We carefully observe that increasing m is more effective than to increase k (when $k < m$). Therefore, we propose to increment the value of system parameter m so that $Adv_{\mathcal{A}}$ remain same.

5.4.5 Privacy

In order to define privacy, we analyzed our protocol according to the *path-privacy* framework in [151] where the privacy of *tag identity* (tag unlinkability) and *path information*(step unlinkability) are formulated together in a single game. Our privacy notion is quite similar to the one proposed in [151], except for some minor modifications. First, we explicitly allow the adversary to query readers during Move operation. Second, unlike path authentication scheme in [151], our state update operation takes place inside the tag with some secrets, such as, coefficient of the tag's polynomial σ_0 and circuit information f.

Let $\mathcal{A}$ be a PPT adversary against RFID path authentication that takes as input the system public parameters, a set of readers $\mathcal{R}$, a set of tags $\mathcal{T}$, and a set of checkpoints $\mathcal{D}$. $\mathcal{A}$ has access to the following oracles $\texttt{Read\_frm\_R}(\mathcal{R}_i)$, $\texttt{Eval\_to\_T}(\mathcal{R}_i, \mathcal{T}_j)$, $\texttt{Path\_Verify}(st_{\mathcal{T}_j})$, $\texttt{Move}(\mathcal{T}_j, k, \mathcal{K}, b)$, where $1 \leq k < |\mathcal{P}|$ for a certain path $\mathcal{P}$, $\mathcal{K} \in \{\mathcal{P}, G\}, b \in \{0, 1\}$.

Let $\mathbf{Exp}_{\mathcal{A}}^{\texttt{Path-Privacy}}[\lambda]$ be a path-privacy experiment that initializes the system $(\mathcal{M}, \mathcal{R}, \mathcal{D}, \mathcal{T})$ through $\mathsf{Setup}(\lambda)$. Adversary $\mathcal{A}$ consists of two algorithms, namely $\mathcal{A}_1$ and $\mathcal{A}_2$. We redefine generic oracles according to the following:

- $\texttt{Read\_frm\_R}(\mathcal{R}_i)$: This oracle returns identity information of a reader $\mathcal{R}_i$ to a tag $\mathcal{T}_j$. We assume that the readers along the path are honest, that is, they will send protocol transcript only if the tag is authenticated.

- $\texttt{Eval\_to\_T}(\mathcal{T}_j,)$: On input tag-reader references $\mathcal{T}_j, \mathcal{R}_i$, this oracle evaluates the internal state $st_{\mathcal{T}_j}$ of a tag $\mathcal{T}_j$. We assume the tags to be honest, i.e., they follow protocol transcripts.

- $\texttt{Path\_Verify}(st_{\mathcal{T}_j})$: On input state information st, this oracle verifies whether $\mathcal{T}_j$ has followed the valid path $\mathcal{P}_v$ and outputs 1 (successful). Otherwise it returns ϕ (fail).

- $\texttt{Move}(\mathcal{T}_j, k, \mathcal{K}, b)$: If $\mathcal{K} = G$, $\mathcal{T}_j$ evaluates the current state st, k times as it moves arbitrarily in the directed acyclic graph G irrespective of the value of b. However, if ($\mathcal{K} = \mathcal{P}$ and $b = 1$), it evaluates the current st along the valid path $\mathcal{P}_v$ in the supply chain k steps that outputs a new state $st_{\mathcal{T}_j}$. However, if $b = 0$, move the tag k steps arbitrarily to any path $\mathcal{P}'$ such that $\mathcal{P}' \cap \mathcal{P} = \varnothing$. The tag $\mathcal{T}_j$'s state is evaluated in each step of the path. Consequently, it returns the state transcript $st_{\mathcal{T}_j}$.

On input of public parameters as mentioned in Fig.1, a probabilistic polynomial time (PPT) algorithm $\mathcal{A}$, denoted by $\mathcal{A}^{\texttt{Read\_frm\_R, Eval\_to\_T, Path\_Verify, Move}}(\lambda)$, runs a supply chain system via the above-mentioned oracles.

In the learning phase, a PPT adversary $\mathcal{A}_1$ queries the four oracles at certain times and outputs two tags $\mathcal{T}_0, \mathcal{T}_1$, a path $\mathcal{P}$ that has at least k readers to reach a checkpoint $\mathcal{D}$ for both tags, and the tag's internal state information st. In the challenge phase, after tossing a coin, $\mathbf{Exp}_{\mathcal{A}}^{\texttt{Path-Privacy}}$ chooses either $\mathcal{T}_0$ or $\mathcal{T}_1$ and moves through k readers remaining along the path and updates the internal state st. Let $\mathcal{T}_0$ reach its last state st_0 by following valid path. Alternatively, $\mathcal{T}_1$ reaches its last state st_1 without following the path. Although $\mathcal{A}_1$ has access to the readers, it has no access to the tag during the Move operations. In the challenge phase, the experiment $\mathbf{Exp}_{\mathcal{A}}^{\texttt{Path-Privacy}}$ provides $\mathcal{A}_2$ with last state of $\mathcal{T}_i$, that is, st_i and previous state information st. Then, $\mathcal{A}_2$ guesses $\mathcal{T}_i$. The experiment outputs 1, and hence, the adversary wins the game if $\mathcal{A}_2$ can guess $\mathcal{T}_i$ correctly with a probability more than 1/2.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\texttt{Path-Privacy}}[\lambda]$

- Run $\mathsf{Setup}(\lambda)$ to set $\mathcal{M}, \mathcal{R}, \mathcal{T}, \mathcal{D}$.

- $\{\mathcal{T}_0, \mathcal{T}_1, \mathcal{P}, k, st\} \leftarrow \mathcal{A}_1^{\texttt{Read\_frm\_R, Eval\_to\_T, Path\_Verify, Move}}$
  where $|\mathcal{P}| \geq k \geq 1$ and st is current state information.

- $b \leftarrow \{0, 1\}$.

- $st_{\mathcal{T}_b} \leftarrow \texttt{Move}(\mathcal{T}_b, k, \mathcal{P}, b)$. $st_{\mathcal{T}_b}$ represents the state of $\mathcal{T}_b$.

- $b' \leftarrow \mathcal{A}_2^{\texttt{Read\_frm\_R, Eval\_to\_T, Path\_Verify, Move}}(st_{\mathcal{T}_b}, st)$.

- Output 1 if $b' = b$, or 0 otherwise.

Advantage of $\mathcal{A}$, denoted by $Adv_{\mathcal{A}}^{\texttt{Path-Privacy}}(\lambda)$, in the path privacy experiment is $|\Pr[\mathbf{Exp}_{\mathcal{A}}^{\texttt{Path-Privacy}}[\lambda] = 1] - \frac{1}{2}|$.

Theorem 26 *If PRF is secure and pseudorandom, then our path authentication protocol is private under the semantic security of Homomorphic MAC scheme.*

Table 5.7: Comparison among path authentication protocols

	Ours	ACNS'12[151]	NDSS'11 [150]	SEC'12[152]	RFIDSec'09[156]
Path generation	static	static	static	dynamic	dynamic
Building blocks	HomMAC	ECElGamal + PRF	ECElGamal	OMS	PRF
Privacy	PUlink[‡] [†]	PUlink[†]	TUlink + SUlink[†]	TUlink + PUlink	NG
Path evaluation	Tag	Reader	Reader	Reader	Tag
Mutual authentication	Yes	No	No	No	Yes
Tag storage	257[*] bits	480 bits	960 bits	720 bits	NG
Reader storage	$O(1)$	$O(1)$	$O(1)$	$O(n)$	O(N)
Checkpoint storage	$O(N)$	$O(N)$	$O(N + vP)$	-	-
Tag computation	PolyA or PolyM	–	–	3H	3H
Reader computation	1H	2ECM, 2ECA, 1PRF	10ECM, 3ECA	1DEC, 3P, 4EX, 6M	1PRF, 1OWF, 1H

HomMAC: Homomorphic MAC on Arithmetic Circuit, ECElGamal: Elliptic Curve ElGamal re-encryption, OMS: Ordered Multi-Signature scheme, PRF: Pseudo Random Function, NG: Not Given, TUlink: Tag Unlinkability, SUlink: Step Unlinkability, PUlink: Path unilinkability, N: Number of total tags, n:Size of batch of tags, vP: Number of valid paths, ECM: Elliptic Curve multiplication, ECA: Elliptic Curve addition, H: Hash function, OWF: Keyed One-Way Function, P: Pairing, EX: Exponentiation, M: Multiplication, PolyA: 1-degree Polynomial addition, PolyM: 1-degree Polynomial multiplication

[‡] Path unilinkability where adversary has access to the reader during `Move` operation.
[†] Privacy proof included.
[*] Considering $32d + 1$ s.t., max degree of a polynomial $d = 8$ with prime p (32-bit).

5.4.6 Performance evaluation

Our scheme is secure and highly efficient, especially if we ignore system initialization process and the computations that can be pre-processed offline. Since this scheme offers a more practical and rigorous security assumption (e.g., adversary having access to the Readers during the Move operation, a polynomial based tag authentication scheme with the same parameter used in path authentication), we consider the tag to perform some lightweight computation at each step. All the major computations are performed by the manufacturer and Checkpoint verifier.

The cost of Tag evaluation depends on the size and gate types of the circuit f. Nevertheless, the evaluated polynomial inside the tag grows with a degree d, finally yields an overhead of $O(d)$ for addition gate and $O(d \log d)$ (using FFT)for multiplication gate. Yet the succinctness of the evaluated polynomial can be assured while $d < |f|$.

Table 5.8: Comparison among path authentication protocols

	Ours	ACISP′09
Authentication	Mutual	Tag
Shared secret	2	1
Tag storage	$2m$	$m \cdot (k+1)$
Random numbers generated by Tag	$2b-1$	$b-1$
Tag Computation	$2\mathfrak{f}(.)$	$1\mathsf{H}, \mathfrak{f}(.)$
Communication cost	$b+3$	$2b+1$

In each step of the path, a tag evaluates either an addition or a multiplication with a 1-degree polynomial (received from the Reader). *Addition* operation can be done simply by adding two vectors of coefficient. A polynomial of degree d has $d+1$ coefficients. So simple addition (with a 1-degree polynomial) requires only $d \geq 1$ addition, while *Multiplication* operation use the convolution operator $'*'$ that requires $2(d+1)$ multiplication and $d-1$ addition operation, resulting in, $3d$ operations in total. However, for very large d the number of operations can be reduced to $O(d \,\mathsf{log}\, d)$ using FFT. Initially, the manufacturer stores 2 secret items in $\mathbb{Z}_p$ (2 coefficient of a 1-degree polynomial) into the tag. Subsequently the tag evaluates the existing polynomial recursively as it moves. Note that the addition gates will not increase the value of d while each multiplication gate increases the value of d by 1. If we consider a 32-bit long prime, then initially a tag requires 64-bits, that grows upto $32d+1$ bits (tag requires to store $d+1$ items for a polynomial of degree d). For simplicity, we allow a maximum of 8 multiplication gates arbitrarily in f, which yields a 257-bit tag storage.

On the other hand, the maximum cost of verification in the checkpoint includes the cost of computing $\Lambda = f(\eta_0, \cdots, \eta_r)$, clearly $O(|f|)$ and $\sum_{l=0}^{d} y_l s^l$, that is $O(d)$. Note that in Case-2 (without $\mathsf{Path} - \mathsf{info}$) of the verification algorithm does not require the calculation of Λ since it is pre-shared between the manufacturer and Checkpoint.

We propose a polynomial-based protocol described in [160] with some modifications for authenticating the tag (optional) at each step of the supply chain. At each step, a tag needs to generate $b-1$ random numbers in $\mathbb{Z}_p$, evaluate a 1-degree polynomial over $\mathbb{Z}_p$. Moreover, the tag is required to store m 1-degree univariate polynomials randomly, that is, the tag needs to store $2m$ items in $\mathbb{Z}_p$. In addition, it takes only 1 modular addition and 1 modular multiplication (Horner's rule) over $\mathbb{Z}_p$ to evaluate the polynomial. It is fairly certain that the value of m ($m = 16$ in [160]) must be larger in our scheme to reach same security settings as that of [160]. We carefully observe that incrementing the value of m comparatively demands more space and computational cost in the reader, instead of the tag. However, if $N(= 2^\lambda$, s.t., $2^\lambda + 1 \leq p)$ be the maximum number of tags in a supply chain, then a tag requires λ-bits ROM, corresponding to λ gates in hardware for each $\mathbb{Z}_p$ element. In addition, modular multiplier takes several hundred more hardware gates. Meanwhile, each reader needs to store m 1-degree bivariate polynomials, that is, it needs $4m$ items in $\mathbb{Z}_p$ to store. In the worst case, it needs to solve mb 1-degree polynomial over $\mathbb{Z}_p$.

5.4.7 Conclusion

In this work, we studied the existing RFID-enabled path authentication schemes for a supply chain management. We present a new direction for using an arithmetic circuit based Homomorphic MAC. In addition, we introduce a refined privacy notion, an appropriate but optional mutual authentication scheme, a potential batch initialization of the tags.

Chapter 6

Conclusion and Future works

Most of the prevention-based security mechanisms in VANETs exploit digital signature as cryptographic primitive. Group Signature is a specialized digital signature that can be directly used to authenticate vehicular communication anonymously without generating pseudonyms. Since existing group signature based security models cannot support all the required secure applications in VANETs and stringent privacy properties of group signature resists several real-life application, we attempt to integrate all the potential group signature properties in a single scheme that meet the application demands of a large scale VANETs that relax stringent privacy definition of group signature to achieve optimally private and application-friendly scheme. Moreover, since most safety-critical applications have stringent delay requirement, verifying huge amount of signatures within a time interval subject to a constraint. We found that batch of signatures verification (a potential solution to accelerate signature verification) is not always feasible for VANET environment. Introducing batch verification increases the size of signature and verification time of each individual signature. This research facilitates choosing appropriate group signature scheme for VANET environment. Meanwhile, we improve an existing batch verification system and analyze the feasibility of exploiting efficient batch verification by comparing our scheme with the former one. Finally, we discover that there is no group signature scheme in the standard security model proposed for VANET because of large signature size and verification cost. At this end, we propose a simplified and application-friendly group signature scheme from standard security model.

In RFID system security, we mainly focus on LPN-based HB-family protocol and Homomorphic Message Authentication Code (HomMAC) on Arithmetic Circuit as security basis. In compare to the other security assumption, LPN based scheme has several advantages such as it offers faster computation with the same security parameter, worst case hardness, security against quantum computers based attacks etc. On the other hand, arithmetic circuit based HomMAC is succinct, composable, extremely efficient and simple to implement. Motivated by the aforementioned advantages, we propose a man-in-the-middle (MIM) attack-free mutual authentication protocol from subspace LPN problem. Since mutual authentication based HB-family protocols cannot be used directly for insecure reader-server channel, we extend our former authentication protocol and design a fully mutual authentication protocol where all the entities (tag, reader and server) can authenticate themselves among each other. Ownership transfer is one of the significant problems in RFID inventory system security and privacy. In order to satisfy new application model and alleviate current shortcomings of both with or without trusted party based ownership transfer application, we propose a semi-trusted party (STP) based RFID tag ownership transfer protocol. Nonetheless, RFID-enabled path authentication is another research area in RFID based inventory control system. In this theses, we concentrate on static path authentication protocols and propose a refined and practical privacy notion for path authentication. Compared to existing Elliptic curve Elgamal Re-encryption (ECElgamal) based solution, our Homomorphic Message authentication Code on arithmetic circuit (HomMAC) based solution offers less memory storage and no computational requirement on the reader.

Future application perspective: We are excited to see what the future will bring and look forward to delivering the latest developments from worldwide leaders in automotive and RFID technology. RFID Technology has evolved and shifted from basic identification solutions to more advanced, boosted, complex solutions including tele-operation, telepresence, software agents, advanced software fusion that are now allowing a business enterprise to trace, track, monitor, locate, control, and utilize their assets.

The Internet of Things (IoT) that refer to ubiquitous connection of uniquely identifiable objects and their virtual representations in an Internet-like structure lead to increased levels of intelligent and autonomous decision making. On the other hand, Near Field Communications (NFC) is a very short-range wireless technology that is based on and is similar to RFID technology is becoming an attractive technology option for some human interaction transactions in the IoT world. Advanced RFID technologies including IOT and NFC can work for a business by providing more aggregated data regarding tagged assets. This refers to the asset communicating, not just being read. Modern RFID is about not only opening up identification options and limits, but about making the tag work for its user.

RFID applications are now successfully established in seismic sensing, real-time parking, self-check ins at libraries, building security, airport baggage tracking, environmental pollution equipments, smart home controls, toll/road payment collection, vehicles equipments monitoring measures, e-processing. Technology experts predicted RFID would be ubiquitous (Smart RFID means things that think) by 2030. The future for RFID is thus making objects not only communicate with human, but to be smart and think for themselves and to build them more complex, applying advancing technologies such as virtual reality and remote access. According to technologist, there are five potential areas that future RFID could develop and innovate into: identification, payment, vehicles, buildings, and animals.

However, RFID data security is a critical issue that must be addressed correctly from both a technical and business point of view in order to ensure widespread ubiquity of RFID technology. Moreover, it must meet the public demand for data security so that general people perceive RFID technology as safe and secure to alleviate legitimate concerns about data security and personal privacy. The key security threats includes front-end communication such as IP communication between RFID readers and the network and back-end communication between tags and readers. These issues must be addressed by future protocols and additional research and development.

Data security threats could evolve in different forms such as clone tags, unauthorized readers, side-channel attacks etc. Future deployments would require new security and privacy enhancement and hence new protocol deployment. Note that new security and privacy measures must equilibrate effectiveness with cost and complexity implications. Since data security is evolving day by day, future authentication protocols will enable RFID technologies to aim security to a new level.

Vehicular network is moving towards vehicular Cloud vision that could provide intelligent transport by sensing the environment and combining with content and user preferences to optimize urban surveillance and vehicular traffic management. The increased compute power of mobile nodes (devices, sensors) with cloud enables vehicle-to-vehicle (V2V) communication. That could make driving safer and traffic flows more efficient. Other Internet connectivity could personalize travel advisories or navigation. Moreover, vehicular networking is going to be one of the most advanced and concrete developments of the *opportunistic networking* paradigm, a natural evaluation from MANET that aim at enabling communication between mobile nodes in highly challenged conditions, which raise new networking and security issues. Therefore, car researchers predict that future vehicular network could prevent up to 80% of car accidents. While future cars could act as witnesses to accidents and later be investigated by law enforcement agency.

In this theses, we explore various studies related to VANET security and privacy focusing on group signatures as cryptographic primitives. We analyse some application treats and challenges that appeal for designing a new model for VANET security. We find that no schemes in the literature have a comprehensive security protocol or framework that cover all security aspects of VANET. Thus, it is necessary to develop a suitable security framework that mitigates all the security and privacy issues. Furthermore, future research in VANET security must deal with the new and emerging technologies such as vehicular cloud, opportunistic network etc.

Future cryptographic perspective: Recently, there has been a significant amount of research on quantum computers (computers that exploit quantum mechanical phenomena for solving problems). It has been believed that existing factorization or discrete logarithm based cryptosystems would be broken if large-scale quantum computers had been constructed efficiently. As a result, Lattice-based cryptographic constructions is drawing attentions to the cryptographic researcher nowadays. They hold a great promise for post-quantum cryptography, as they come up with very strong security proofs based on worst-case hardness as well as relatively efficient implementations. Lattices will be a future source of fascinating problems in computer science and mathematics. Compare to conventional number theoretic cryptography, lattice based cryptosystem offers several interesting properties and intriguing advantages. For instance, lattice-based schemes could be efficient to the greatest extent since the basic operations involve very simple computation (only adding small integers) in compare to the traditional requirement of modular exponentiating large integers, the security of lattice-based cryptography is based on worst-case assumption (as hard as possible to break) in compare to average-case assumption in the standard cryptographic schemes. In recent years, Learning with Errors (LWE) problem has turned out to be a potential flexile basis for cryptographic constructions for its theoretical reasons that the hardness of the problem follows from the worst-case hardness of lattice assumption. Similarly, Learning Parity from Noise (LPN) is another extensive studied problem in learning/coding theory and also believed to be hard. Although LWE is a generalization of LPN, it lacks the simplicity of LPN. LWE requires many multiplications modulo some prime as opposed to inner products of bit-vectors as for LPN. Therefore, LWE based schemes are less suited for applying in the resource constrained devices like RFID tags etc. Therefore, any progress concerning LPN would be important also in the coding theory or LWE and vice versa. Over the past few years we have been conducting research on the cryptographic solutions to secure application using the LPN problem as primitive. Our contributions include building new cryptographic protocols having rich features and strong security properties. Some of these techniques and perspective have led to significant progress in not only in lattice-based cryptography but also in conventional number-theoretic cryptography. Our focus will be mainly on the practical aspects of lattice-based cryptography and less on the methods used to establish their security. From the view point of security, lattice-based cryptographic constructions can be divided into two types. The first includes practical proposals, which are typically very efficient, but often do not provide a rigorous proof of security. The second type admits strong provable security guarantees based on the worst-case hardness of lattice problems, but only a few of them are sufficiently efficient to be used in practice. For instance, one of the open problems regarding LWE problems is that unlike LPN they tend not to be efficient enough for practical application. Even the simplest primitives, e.g., one-way functions have the key size at least quadratic in the primary security parameter that needs to be a quite high for the security against the best known attack. We plan to consider both types in my future research work, with more emphasis on the former type. As my previous research background was to construct secure and private cryptographic protocols for resource constrained devices like RFID, VANET etc. using lattice-based assumption, we plan to investigate more in the near future: How can additional mathematical structure like pseudo-inverse matrices be exploited efficiently to design practical schemes for resource constrained devices (e.g., smart phone, Sensor nodes etc.)?, How existing special algebraic structure lattice (Ring-LWE) can be adopted avoiding the lack of efficiency? How to integrate other important cryptographic notions with natural lattice based realizations? Out of Seven Current (National institute of Standard and Technology) NIST cryptographic research projects we have experience working with four of them, namely, Pairing-based cryptography, Post-Quantum Cryptography, Privacy enhancing cryptography, Group Signatures. Specially, Lattice-based assumption under Post-quantum cryptography is yet to be standardized by NIST. Undoubtedly there is room for further improvement with new ideas, or concepts in this field.

On the other hand, although group signature is widely used in the literature for VANET security and privacy, it is not efficient (computationally expensive) enough to comply with real-time response requirement of large scale vehicular network. Moreover, if the security of a group signature is proved in the standard model (rather in the random oracle), system performance degrades. Therefore, security experts are looking for new cryptographic primitives to ensure security and performance are balanced and optimized. For example, group signature used in our protocol is secure under random oracle model. Recently, a group signature scheme with the some of its functionality has been proposed where security is in the standard model but the signature size is not $O(1)$ group elements per signature but logarithmic to the number of group members (not suitable for a large scale VANET). Thus, open problems still remain in the technical perspective: achieving a truly structure-preserving IBE or efficient partially structure-preserving IBE. Besides that, group signature with message dependent opening model conveys an inherent limitation. That is, once a message is used to generate token, it can be used by any group members. Since tokens are computationally independent of the signers, if the signers of a group collide among themselves, attested tracer can trace signer's identity depending on a token, even if the token was not generated for the corresponding signer.

Bibliography

[1] Bo Qin, Qianhong Wu, Josep Domingo-Ferrer, and Lei Zhang. Preserving Security and Privacy in Large-Scale VANETs. ICICS 2011, LNCS 7043, pp. 121-135, 2011.

[2] Wei, V.K., Yuen, T.H., Zhang, F. Group Signature Where Group Manager, Members and Open Authority Are Identity-based. In ACISP, LNCS, vol. 3574, pp. 468-480. Springer, Heidelberg, 2005.

[3] B. K. Chaurasia, S. Verma, and S. M. Bhasker. Message broadcast in VANETs using Group Signature. Fourth International Conference on Wireless Communication and Sensor Networks, Indian Institute of Information Technology, Allahabad, pp. 131-136, 2008.

[4] B. K. Chaurasia and S. Verma. Conditional Privacy through Ring Signature in Vehicular Ad-hoc Networks. Trans. on Computational Science XIII, LNCS 6750, pp. 147-156, Springer-Verlag Berlin Heidelberg, 2011.

[5] Gamage, Chandana, Ben Gras, Bruno Crispo, and Andrew S. Tanenbaum. An identity-based ring signature scheme with enhanced privacy. In Securecomm and Workshops, 2006, pp. 1-5. IEEE, 2006.

[6] M. S. I. Mamun, A. Miyaji. An Optimized Signature Verification System for Vehicle Ad hoc NETwork, The 8th IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM2012), IEEE, 2012.

[7] Wu, Q., Domingo-Ferrer, J., Gonzalez-Nicolas, U. Balanced Trustworthiness, Safety and Privacy in Vehicle-to-vehicle Communications. IEEE Transaction on Vehehicluar Technology 59(2), 559-573, 2010.

[8] Zhang, L., Wu, Q., Solanas, A., Domingo-Ferrer, J. A Scalable Robust Authentication Protocol for Secure Vehicular Communications. IEEE Transactions on Vehicular Technology 59(4), 1606-1617, 2010.

[9] Zarki, M.E., Mehrotra, S., Tsudik, G., Venkatasubramanian, N. Security Issues in a Future Vehicular Network. In European Wireless, vol 2, 2002.

[10] IEEE Intelligent Transportation System Society.
web: http://www.ewh.ieee.org/tc/its/

[11] Sumra, I.A., Hasbullah, H., Ahmad, I., bin Ab Manan, J.-L. New card based scheme to ensure security and trust in vehicular communications. Electronics, Communications and Photonics Conference (SIECPC), 2011.

[12] Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems - 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ASTM E2213-03, September 2003.

[13] F. Dotzer, Privacy issues in vehicular ad hoc networks. In Proc. PET, vol. 3856, Lecture Notes in Computer Science, pp. 197-209, 2005.

[14] N. Koblitz. Elliptic curve cryptosystems. Math. Comput., vol. 48, no. 177, pp. 203-209, 1987.

[15] 5GHz Band Dedicated Short Range Communications (DSRC), ASTM E2213-03, web: http://www.iteris.com/itsarch/html/standard/dsrc5ghz.htm

[16] Zhang, C., Lin, X., Lu, R., Ho, P.-H., Shen, X. An efficient message authentication scheme for vehicular communications. IEEE Transactions on Vehicular Technology, 57(6), 3357-3368, 2008.

[17] Lin, Chia-Min, *et al.* HB Family RFID Mutual Authentication Protocol. The seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing IIH-MSP, 2011.

[18] Raya, M., Papadimitratos, P., Aad, I., Jungels, D., Hubaux, J.-P. Eviction of Misbehaving and Faulty Nodes in Vehicular Networks. IEEE Journal Selected Areas in Communication 25(8), 1557-1568, 2007.

[19] Raya, M., Hubaux, J.-P. The Security of Vehicular Ad-Hoc Networks. In ACM Workshop on Security of Ad hoc and Sensor Networks-SASN, pp. 11-21, ACM Press, 2005.

[20] Papadimitratos, P., Buttyan, L., Hubaux, J.-P., Kargl, F., Kung, A., Raya, M. Architecture for Secure and Private Vehicular Communications. In International Conference on ITS Telecommunications, pp. 1-6, 2007.

[21] Parno, B., Perrig, A. Challenges in Securing Vehicular Networks. In the Workshop on Hot Topics in Networks (HOTNETS), 2005.

[22] Lin, X., Sun, X., Ho, P.-H., Shen, X. GSIS: A Secure and Privacy Preserving Protocol for Vehicular Communications. IEEE Transaction on Vehicular Technology 56(6), 3442-3456, 2007.

[23] Guo, J., Baugh, J.P., Wang, S. A Group Signature Based Secure and Privacy preserving Vehicular Communication Framework. In Mobile Networking for Vehicular Environments 2007, pp. 103108, 2007.

[24] Fonseca, E., Festag, A., Baldessari, R., Aguiar, R.L. Support of Anonymity in VANETs - Putting Pseudonymity into Practice. In IEEE Wireless Communications and Networking Conference-WCNC, pp. 3400-3405, IEEE Press, 2007.

[25] Chenxi Zhang, Rongxing Lu, Xiaodong Lin, Pin-Han Ho, Xuemin Shen. An Efficient Identity-Based Batch Verification Scheme for Vehicular Sensor Networks, INFOCOM, The 27th IEEE Conference on Computer Communications, Page(s): 246 - 250, 2008.

[26] Qianhong Wu, Domingo-Ferrer, J., Gonzalez-Nicolas, U. Balanced Trustworthiness, Safety, and Privacy in Vehicle-to-Vehicle Communications. IEEE Transactions on Vehicular Technology, Volume. 59, Issue. 2, 2010.

[27] H. Zhu, X. Lin, R. Lu, P.-H. Ho and X. Shen. AEMA: An Aggregated Emergency Message Authentication Scheme for Enhancing the Security of Vehicular Ad Hoc Networks, IEEE International Conference on Communications - ICC, 2008.

[28] J. Groth. Fully anonymous group signatures without random oracles, (2012) (manuscript), http://www.cs.ucl.ac.uk/staff/J.Groth/CertiSignFull.pdf

[29] Mihir Bellare et.al. Fast Batch Verification for Modular Exponentiation and Digital Signatures. Eurocrypt Proceedings, Lecture Notes in Computer Science Vol. 1403, K. Nyberg ed., Springer-Verlag, 1998.

[30] Ferrara, A.L., Green, M., Hohenberger, S., Pedersen, M.O. On the Practicality of Short Signature Batch Verification,
web: http://eprint.iacr.org/2008/015.pdf

[31] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. In Proc. EUROCRYPT, pages 268-286, Springer-Verlag, Lecture Notes in Computer Science No. 3027, 2004.

[32] D. Boneh and M. Franklin. Identity-based encryption from the weil paring. In Proc. CRYPTO, pages 213229. Springer-Verlag, LNCS 2139, 2001.

[33] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. In Proc. EUROCRYPT 2004, pages 268286, Springer-Verlag, Lecture Notes in Computer Science No. 3027, 2004.

[34] Qin, B., Wu, Q., Susilo, W., Mu, Y. Publicly Verifiable Privacy-Preserving Group Decryption. In Proc. Inscrypt 2008, LNCS, vol. 5487, pp. 7283. Springer, Heidelberg, 2008.

[35] P. Baptiste. Polynomial time algorithms for minimizing the weighted number of late jobs on a single machine with equal processing times. Journal of Scheduling, 2: 245-252, 1999.

[36] Amotz Bar-Noy et el. Throughput maximization of real-time scheduling with batching. Proceedings of the 13th annual ACM-SIAM symposium on Discrete algorithms, 2002.

[37] Hiroshi Kise, Toshihide Ibaraki and Hisashi Mine, A Solvable Case of the One-Machine Scheduling Problem with Ready and Due Times, Operations Research 26(1), 121-126, 1978.

[38] L. Zhang, Q. Wu, A. Solanas, and J. Domingo-Ferrer. A Scalable Robust Authentication Protocol For Secure Vehicular Communications. IEEE Transactions on Vehicular Technology, 2009.

[39] A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. IEICE Trans. Fundam., vol. E84-A, no. 5, 2001.

[40] M. Scott, Efficient implementation of cryptographic pairings.
http://www.pairing-conference.org/2007/invited/Scottslide.pdf

[41] The GNU Multiple Precision Arithmetic Library
web: http://gmplib.org/

[42] The Pairing based Cryptography(PBC)Library,
web: http://crypto.stanford.edu/pbc/

[43] Legislative resolution on the proposal for the provision of public electronic communication services and amending Directive 2002/58/EC(COM(2005)0438 C6-0293/2005 2005/0182(COD)), 2005.

[44] M. Raya and J. P. Hubaux. Securing vehicular ad hoc networks. Journal of Computer Security, Vol. 15, No. 1, pp. 39-68, 2007.

[45] D. Chaum and E. V. Heyst. Group signatures. In EUROCRYPT, volume 547 of Lecture Notes in Computer Science, pages 257-265, 1991.

[46] Gerlach, M., Festag, A., Leinmuller, T., Goldacker, G., Harsch, C. Security architecture for vehicular communication. In: The 5th International Workshop on Intelligent Transportation, 2007.

[47] Chen, Liqun, and Jiangtao Li. Revocation of direct anonymous attestation. In Trusted Systems, pp. 128-147. Springer Berlin Heidelberg, 2011.

[48] Chim, T.W., Yiu, S.M., Hui, L.C.K., Li, V.O.K. SPECS: Secure and privacy enhancing communications schemes for VANETs. Ad Hoc Networks 9(2), 189-203, 2011.

[49] Lin, Hsiao-Ying, and Wen-Guey Tzeng. An efficient solution to the millionaires problem based on homomorphic encryption. In Applied Cryptography and Network Security, pp. 456-466. Springer Berlin Heidelberg, 2005.

[50] MSI Mamun, Atsuko Miyaji. An efficient batch verification system for large scale VANET. Intl. J. of Security and Communication Networks (SCN), Wiley Publication DOI: 10.1002/sec.980, 2014.

[51] Chu, Cheng-Kang, Joseph K. Liu, Xinyi Huang, and Jianying Zhou. Verifier-local revocation group signatures with time-bound keys. In Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, pp. 26-27. ACM, 2012.

[52] Guomin Yang, Qiong Huang, Duncan S. Wong, and Xiaotie Deng. Universal authentication protocols for anonymous wireless communications.IEEE Transactions on Wireless Communications ,9(1):168174, 2010.

[53] Ohara, K., Sakai, Y., Emura, K., Hanaoka, G. (2013, May). A group signature scheme with unbounded message-dependent opening. In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security (pp. 517-522). ACM.

[54] Ferrara, A. L., Green, M., Hohenberger, S., Pedersen, M. (2009). Practical short signature batch verification. In Topics in CryptologyCT-RSA 2009 (pp. 309-324). Springer Berlin Heidelberg.

[55] Heen, O., Guette, G., Genet, T. On the unobservability of a trust relation in mobile ad hoc networks. In Information Security Theory and Practice. Smart Devices, Pervasive Systems, and Ubiquitous Networks (pp. 1-11). Springer Berlin Heidelberg, 2009.

[56] Buttyan, L., Holczer, T., Weimerskirch, A., Whyte, W. SLOW: A practical pseudonym changing scheme for location privacy in vanets. In Vehicular Networking Conference (VNC),(pp. 1-8). IEEE 2009.

[57] G. Ateniese, G. Song, and G. Tsudik. Quasi-efficient revocation of group signatures, In Financial Crypto 2002, Lecture Notes in Computer Science (LNCS), 2002.

[58] E. Bresson and J. Stern. Efficient Revocation in Group Signatures, In Proceedings of Public Key Cryptography (PKC'2001), Springer-Verlag, 2001.

[59] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In CRYPTO, LNCS(3152), pages 41-55,2004.

[60] O. Blazy, G. Fuchsbauer, M. Izabachene, A. Jambert, H. Sibert, and D. Vergnaud. Batch Groth-Sahai. In Proc. ACNS 2010, volume 6123 of LNCS, pages 218-235. Springer-Verlag,2010.

[61] E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In CCS 04, pages 132145, New York, NY, USA, 2004. ACM Press.

[62] Wei, Lingbo, and Jianwei Liu. Shorter verifier-local revocation group signature with backward unlinkability. In Pairing-Based Cryptography-Pairing 2010, pp. 136-146. Springer Berlin Heidelberg, 2010.

[63] J. Liu, W.Susilo, D. Wong. Ring signature with designated linkability. IWSEC2006, LNCS4266, pp.104-119, 2006.

[64] J. Hwang, S. Lee, B. Chung, H. Cho, D. Nyang. Short Group Signatures with Controllable Linkability. In IEEE LightSec2011, Pages: 44-52, 2011.

[65] L. Zhang, Q. Wu, B. Qin, J. Ferrer. Practical Privacy for Value-Added Applications in Vehicular Ad Hoc Networks. In IDCS2012, LNCS(7646), pp 43-56, 2012.

[66] W. Lingbo.On a Group Signature Scheme Supporting Batch Verification for Vehicular Networks. In IEEE Multimedia Information network and Security (MINES2011), pp 436-440, 2011.

[67] Chow, S. S., Susilo, W., Yuen, T. H. Escrowed linkability of ring signatures and its applications. In Progress in Cryptology-VIETCRYPT 2006 (pp. 175-192). Springer Berlin Heidelberg,2006.

[68] Libert, B., Vergnaud, D. Group signatures with verifier-local revocation and backward unlinkability in the standard model. In Cryptology and Network Security (pp. 498-517). Springer Berlin Heidelberg, 2009.

[69] Nakanishi, T., Fujii, H., Yuta, H., Funabiki, N. Revocable group signature schemes with constant costs for signing and verifying. IEICE transactions on fundamentals of electronics, communications and computer sciences, 93(1), 50-62, 2010.

[70] Malina, Lukas, *et al.* Short-Term linkable group signatures with categorized batch verification. Foundations and Practice of Security. Springer Berlin Heidelberg, 2013.

[71] M. Bellare, H. Shi, C. Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 136-153. Springer, Heidelberg 2005.

[72] D. Song. Practical Forward-Secure Group Signature Schemes, In Proceedings of ACM Symposium on Computer and Communication Security. November 2001.

[73] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In EUROCRYPT, volume 4965 of Lecture Notes in Computer Science, pages 415-432, 2008.

[74] J. Groth. Fully Anonymous Group Signatures Without Random Oracles. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 164180. Springer, Heidelberg (2007)

[75] Y. Sakai, J. C.N. Schuldt, K. Emura, H. Hanaoka, K. Ohta. On the security of Dynamic Group Signaturs: Preventing Signature Hijacking, LNCS 7293, pp. 715-732, PKC 2012.

[76] B. Libert, T. Peters, M. Yung. Group Signatures with Almost-for-free Revocation. CRYPTO2012, LNCS7417, pp. 571-589, 2012.

[77] B. Libert, T. Peters, M. Yung. Scalable Group Signature with Revocation. Eurocrypt2012, LNCS7237, pp 609-627, 2012.

[78] E. Welbourne, L. Battle, G. Cole . Building the Internet of Things Using RFID: The RFID Ecosystem Experience. Internet Computing, IEEE , vol.13, no.3, pp.48-55, 2009.

[79] Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. In Victor Shoup, editor, CRYPTO 2005, volume 3621 of LNCS, pages 293-308. Springer, August 2005.

[80] K. Pietrzak, E. Kiltz, D. Cash, A. Jain, D. Venturi, Authentication from Hard learning problem, Eurocrypt 2011, LNCS 6632, pp 7-26,2011.

[81] Yang, Jeongkyu, *et al.* Mutual Authentication Protocol. Workshop on RFID and Lightweight Crypto, 2005.

[82] Lo, Nai-Wei, and Kuo-Hui Yeh. An efficient mutual authentication scheme for EPC-global class-1 generation-2 RFID system. Emerging Directions in Embedded and Ubiquitous Computing. Springer Berlin Heidelberg, 2007.

[83] Lo, N. W., Yeh, K. H., Yeun, C. Y. New mutual agreement protocol to secure mobile RFID-enabled devices. Information security technical report, 13(3), 151-157, 2008.

[84] Ahamed, Sheikh Iqbal, Farzana Rahman, and Endadul Hoque. ERAP: ECC based RFID authentication protocol. IEEE Future Trends of Distributed Computing Systems, FTDCS'08, 2008.

[85] Batina, Lejla, *et al.* An elliptic curve processor suitable for RFID tags. International Association for Cryptologic Research ePrint Archive, 2006.

[86] N. J. Hopper and M. Blum. Secure human identification protocols. Advances in Cryptology - ASIACRYPT 2001, Lecture Notes in Computer Science, Vol. 2248, Springer, pp. 52-66, 2001.

[87] C. Cid, M. Robshaw, The eSTREAM Portfolio 2009 Annual Update. July 2009. Available from http://www.ecrypt.eu.org/stream/.

[88] Golub, Gene, W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. Journal of the Society for Industrial & Applied Mathematics, Series B: Numerical Analysis 2.2: pp. 205-224, 1965.

[89] Jonathan Katz, Ji Sun Shin, and Adam Smith, Parallel and concurrent security of the HB and HB+ protocols, Journal of Cryptology, 23(3):402-421, July 2010

[90] Henri Gilbert, Matt Robshaw, and Herve Sibert. An active attack against HB+ - a provably secure lightweight authentication protocol. Cryptology ePrint Archive, Report 2005/237, 2005.

[91] Julien Bringer, H. Chabanne, and Emmanuelle Dottax. HB++: a lightweight authentication protocol secure against some attacks. In SecPerU, pp. 28-33, 2006.

[92] Jorge Munilla and Alberto Peinado. HB-MP: A further step in the HB-family of lightweight authentication protocols. Computer Networks, 51(9):2262-2267, 2007.

[93] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. Good variants of HB+ are hard to find. In Gene Tsudik, editor, FC 2008, volume 5143 of LNCS, pp. 156-170. Springer, 2008.

[94] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. HB++: Increasing the security and efficiency of HB+. In Nigel P. Smart, editor, EUROCRYPT 2008, volume 4965 of LNCS, pp. 361-378. Springer, 2008.

[95] Khaled Ouafi, Raphael Overbeck, and Serge Vaudenay. On the security of HB# against a man-in-the-middle attack. In Josef Pieprzyk, editor, ASIACRYPT 2008, volume 5350 of LNCS, pages 108-124. Springer, 2008.

[96] Moore, E. H. On the reciprocal of the general algebraic matrix. Bulletin of the American Mathematical Society 26 (9), 394-395, 1920.

[97] Thuc, D.N., Hue, T.B.P., Van, H.D. An Efficient Pseudo Inverse Matrix-Based Solution for Secure Auditing. IEEE-RIVF, pp. 712 2010.

[98] Y. Dodis, J. Katz, S. Xu, M. Yung. Key-Insulated Public Key Cryptosystems. In Eurocrypt02, volume 2332 of LNCS, pages 6582. Springer, 2002.

[99] MSI Mamun, A. Miyaji. A fully-secure RFID authentication protocol from exact LPN assumption, IEEE TrustCom'13, page 102-109, DOI: 10.1109/TrustCom.2013.17

[100] Pietrzak, Krzysztof. Cryptography from learning parity with noise. In SOFSEM 2012: Theory and Practice of Computer Science, pp. 99-114. Springer Berlin Heidelberg, 2012.

[101] MSI Mamun, A. Miyaji. A privacy-preserving efficient RFID authentication protocol from SLPN assumption. International Journal of Computational Science and Engineering (IJCSE), Special Issue on Converged Networks, Technologies and Applications, Inderscience Publishers, Vol. 9, 2014.

[102] Jens Hermans, Andreas Pashalidis, Frederik Vercauteren, Bart Preneel. A New RFID Privacy Model, ESORICS 2011.

[103] Ching Yu Ng, Willy Susilo, Yi Mu, and Reihaneh Safavi-Naini. New Privacy Results on Synchronized RFID Authentication Protocols against Tag Tracing. In Michael Backes and Peng Ning, editors, ESORICS, volume 5789 of LNCS, pages 321336. Springer, 2009.

[104] Cao, X , ONeill, M. (2011). F-HB: An Efficient Forward Private Protocol. Workshop on Lightweight Security and Privacy: Devices, Protocols and Applications(Lightsec), 2011.

[105] T. V. Le, M. Burmester, and B. de Medeiros. Universally Composable and Forward-secure RFID Authentication and Authenticated Key Exchange. ACM Symposium on InformAtion, Computer and Communications Security (ASIACCS), 2007.

[106] O. Billet, J. Etrog and H. Gilbert. Lightweight Privacy Preserving Authentication for RFID Using a Stream Cipher. International Workshop on Fast Software Encryption (FSE), 2010.

[107] G. Avoine and P. Oechslin, A Scalable and Provably Secure Hash- Based RFID Protocol, IEEE International Workshop on Pervasive Computing and Communication Security, March 2005.

[108] Xuefei Leng, Keith Mayes, Konstantinos Markantonakis. HB-MP+ Protocol: An Improvement on the HB-MP Protocol. IEEE International Conference on RFID. pp. 118-124, 2008.

[109] G. Tsudik, Ya-trap: Yet another trivial RFID authentication protocol, in PerCom Workshops, pp. 640-643, 2006.

[110] C. Chatmon, T. van Le, and M. Burmester, Secure anonymous rfid authentication protocols, Computer & Information Sciences, Florida AM University, Tech. Rep., 2006.

[111] L. He, S. Jin, T. Zhang, and N. Li, An enhanced 2-pass optimistic anonymous rfid authentication protocol with forward security, in WiCOM, pp. 1-4, 2009.

[112] B. Applebaum, Y. Ishai, E. Kushilevitz. Cryptography with Constant Input Locality. Journal of Cryptology 22(4), 429469, 2009.

[113] MSI Mamun, A. Miyaji, M. Rahman. A Secure and Private RFID Authentication Protocol under SLPN Problem. NSS2012, LNCS 7645, pp. 476-489, 2012.

[114] Murty, Katta G., and Santosh N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. Mathematical programming 39.2: 117-129, 1987.

[115] S. Fouladgar and H. Afifi. An efficient delegation and transfer of ownership protocol for RFID tags. In First International EURASIP Workshop on RFID Technology, Vienna, Austria, 2007.

[116] C. Yu Ng, W. Susilo, Y. Mu, and R. Safavi-Naini. Practical RFID Ownership Transfer Scheme. Journal of Computer Security - Special Issue on RFID System Security, 2010.

[117] A. Jain, S. Krenn, K. Pietrzak, A. Tentes. Commitments and Efficient Zero-Knowledge Proofs from Learning Parity with Noise. ASIACRYPT 2012, LNCS,Volume 7658, pp 663-680, 2012.

[118] S. Hill, Cryptography in an Algebraic Alphabet, The American Mathematical Monthly Vol.36, pp. 306-312, 1929.

[119] S. Saeednia, How to Make the Hill Cipher Secure, Cryptologia, Vol.24, No.4, pp. 353-360, 2000.

[120] Molnar, D., Soppera, A., Wagner, D. A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags. In SAC 2005, LNCS, vol. 3897, pp. 276290, Springer, Heidelberg, 2006.

[121] Elkhiyaoui, K., Blass, E.-O., Molva, R. ROTIV: RFID ownership transfer with issuer verification. In RFIDSec 2011. LNCS, vol. 7055, pp. 163-182, Springer, Heidelberg, 2012.

[122] Fern andez-Mir, A., Trujillo-Rasua, R., Castell a-Roca, J., Domingo-Ferrer, J. A scalable RFID authentication protocol supporting ownership transfer and controlled delegation. In RFIDSec 2011. LNCS, vol. 7055, pp. 147-162. Springer, Heidelberg, 2012.

[123] R. H. Deng, Y. Li, M. Yung, and Y. Zhao. A new framework for RFID Privacy. in Proceedings of the 15th European Symposium on Research in Computer Security (ESORICS 2010), vol. 6345 of LNCS, pp. 118, Springer, 2010.

[124] Rizomiliotis, Panagiotis, and Stefanos Gritzalis. GHB#: a provably secure HB-like lightweight authentication protocol. Applied Cryptography and Network Security. Springer,DEAKIN 2012.

[125] Chen, C. L., Lai, Y. L., Chen, C. C., Deng, Y. Y., Hwang, Y. C. RFID ownership transfer authorization systems conforming EPCglobal class-1 generation-2 standards. International Journal of Network Security, 13(1), 41-48, 2011.

[126] N. Park, H. Kim, K. Chung, and S. Sohn. Design of an Extended Architecture for Secure Low-Cost 900MHz UHF Mobile RFID Systems. Proceedings of the International Symposium on Consumer Electronics (ISCE06), pp. 1-6, 2006.

[127] K. Rhee, J. Kwak, S. Kim, and D. Won. Challenge-Response Based RFID Authentication Protocol for Distributed Database Environment. Proceedings of the International Conference on Security in Pervasive Computing (SPC'05), pp. 70-84, 2005.

[128] Yeo, S. S., Kim, S. C., Kim, S. K., Park, G., Kim, S. S., Yang, K. S., Cho, S. E. Protecting your privacy with a mobile agent device in RFID environment. Wireless personal communications, 51(1), 165-178, 2009.

[129] D. Micciancio and P. Mol. Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions. CRYPTO 2011, volume 6841 of LNCS, pages 465-484. Springer, 2011.

[130] Moriyama, D., Matsuo, S. I., Ohkubo, M. Relations among notions of privacy for RFID authentication protocols. In Computer Security ESORICS (pp. 661-678). Springer LNCS, 2012.

[131] S. Kumar, P. Crowley, Segmented hash: an efficient hash table implementation for high performance networking subsystems, ANCS'05 Proceedings of the 2005 ACM symposium on Architecture for networking and communications systems, Pages 91-103.

[132] Song, B. and Mitchell, C.J.: RFID Authentication Protocol for Low-cost Tags. The ACM Conference on wireless Network Security, WiSec, ACM Press (2008)

[133] C. Berbain, O. Billet, J. Etrog and H. Gilbert, An Efficient Forward Private RFID Protocol, ACM Conference on Computer and Communications Security (CCS), November 2009.

[134] B. Song and C. J. Mitchell. Scalable RFID security protocols supporting tag ownership transfer. Comput. Commun. , vol. 34, no. 4, pp. 556566, 2011.

[135] M. H. Yang. Across-authority lightweight ownership transfer protocol. Electronic Commerce Res. Applicat., vol. 10, no. 4, pp. 375383, 2011.

[136] K. Osaka, T. Takagi, K. Yamazaki, and O. Takahashi. An efficient and secure RFID security method with ownership transfer. in Proc. ICCIS, 2006.

[137] G. Kapoor and S. Piramuthu, Single RFID tag ownership transfer protocols. IEEE Trans. Systems, Man, Cybern. C, Appl. Rev., vol. 42, no. 2, pp. 164173, Mar. 2012.

[138] H. Stefan, E. Kiltz, V. Lyubashevsky, C. Paar, and K. Pietrzak. Lapin: an efficient authentication protocol based on Ring-LPN. FSE, pp. 346-365. Springer 2012.

[139] Van Deursen, T., Mauw, S., Radomirovic, S., Vullers, P. Secure ownership and ownership transfer in RFID systems. In ESORICS 2009. LNCS, vol. 5789, pp. 637-654. Springer, Heidelberg, 2009.

[140] D. M. Freeman. Improved security for linearly homomorphic signatures: A generic framework. PKC 2012.

[141] D. Yevgeniy, E. Kiltz, K. Pietrzak, and D. Wichs. Message authentication, revisited. In Advances in EUROCRYPT 2012, pp. 355-374. Springer, 2012.

[142] Bernstein, Daniel J., and Tanja Lange. Never trust a bunny. In Radio Frequency Identification. Security and Privacy Issues, pp. 137-148. Springer, 2013.

[143] Cai, Shaoying, *et al.* Protecting and restraining the third party in RFID-enabled 3PL supply chains. Information Systems Security. LNCS, 246-260, 2011.

[144] Doss, Robin, Wanlei Zhou, and Shui Yu. Secure RFID Tag Ownership Transfer Scheme based on Quadratic Residues. 1-1, 2013.

[145] L. Kuseng, Z. Yu, Y. Wei, and Y. Guan. Lightweight mutual authentication and ownership transfer for RFID systems. in Proc. IEEE Infocom, pp. 15, 2010.

[146] RFID Security and Privacy Lounge, www.avoine.net/rfid/

[147] M. Ohkubo, K. Suzuki, and S. Kinoshita, Cryptographic approach to privacy-friendly tags, in RFID Privacy Workshop, 2003.

[148] Li, Nan, Yi Mu, Willy Susilo, and Vijay Varadharajan. Secure RFID Ownership Transfer Protocols. In Information Security Practice and Experience, pp. 189-203. Springer Berlin Heidelberg, 2013.

[149] Y. Li and X. Ding, "Protecting RFID communications in supply chains, in ASIACCS 2007, pp. 234-241, 2007.

[150] E.O. Blass, K. Elkhiyaoui, and R. Molva. Tracker: Security and privacy for RFID-based supply chains. in NDSS, pp. 455472, 2011.

[151] Cai, Shaoying, Robert Deng, Yingjiu, Li, Yunlei, Zhao. A new framework for privacy of RFID path authentication. Applied Cryptography and Network Security. Springer Berlin Heidelberg, 2012.

[152] Cai, Shaoying, Yingjiu Li, Yunlei Zhao. Distributed Path Authentication for Dynamic RFID-Enabled Supply Chains. Information Security and Privacy Research. Springer Berlin Heidelberg, pp501-512, 2012.

[153] Wang, Hongbing, *et al.* Two-level path authentication in EPCglobal Network, IEEE International Conference on. IEEE, 2012.

[154] Saito, J., Imamoto, K., Sakurai, K. Reassignment scheme of an RFID tags key for owner transfer. In EUC-WS 2005. LNCS, vol. 3823, pp. 1303-1312. Springer, Heidelberg, 2005.

[155] D. Boneh, B. Lynn, and H. Shacham, Short signatures from the weil pairing," Journal of Clyptology, vol. 17, no. 4, pp. 297-319, 2004.

[156] Ouafi, Khaled, and Serge Vaudenay. Pathchecker: an RFID Application for Tracing Products in Suply-Chains. Workshop on RFID SecurityRFIDSec. Vol. 9. 2009.

[157] Guruswami, V., Sudan, M. Improved decoding of Reed-Solomon and algebraicgeometry codes. IEEE Transactions on Information Theory 45(6), 17571767, 1999.

[158] Catalano, Dario, and Dario Fiore. Practical Homomorphic MACs for Arithmetic Circuits. Advances in CryptologyEUROCRYPT. Springer Berlin Heidelberg, 336-352, 2013.

[159] Gennaro, R., Wichs, D. Fully homomorphic message authenticators. Cryptology ePrint Archive, Report 2012/290 (2012), http://eprint.iacr.org

[160] Wu, Jiang, and Douglas R. Stinson. A highly scalable RFID authentication protocol. ACISP 2009. Springer Berlin Heidelberg, 2009.

[161] Naor, Moni, and Benny Pinkas. Oblivious polynomial evaluation. SIAM Journal on Computing 35.5: 1254-1281, 2006.

[162] Shpilka, A., Yehudayo, A. Arithmetic circuits: A survey of recent results and open questions. Foundations and Trends in Theoretical computer science 5(3-4), 207-388, 2010.

[163] ISO/TR 12859:2009, Intelligent transport systems– System architecture– Privacy aspects in ITS standards and systems.

[164] ISO 21217:2014, Intelligent transport systems– Communications access for land mobiles (CALM) Architecture.

[165] IEEE 1609.0-2013, IEEE Guide for Wireless Access in Vehicular Environments (WAVE) Architecture.

[166] IEEE 1609.2-2013, IEEE Standard for Wireless Access in Vehicular Environments, Security Services for Applications and Management Messages.

[167] IEEE 1609.11-2010, Standard for Wireless Access in Vehicular Environments (WAVE), Over-the-Air Electronic Payment Data Exchange Protocol for Intelligent Transportation Systems (ITS).

[168] Engoulou, Richard Gilles, Martine Bellache, Samuel Pierre, and Alejandro Quintero. VANET security surveys. Computer Communications 44, PP 1-13, 2014.

[169] Lu, R., *et al.* ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications. INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, 2008.

[170] Li, C.-T., *et al.* A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks. Computer Communications 31(12): 2803-2814, 2008.

[171] Burmester, M., *et al.* Strengthening privacy protection in VANETs. Networking and Communications, 2008. WIMOB'08. IEEE International Conference on Wireless and Mobile Computing, IEEE, 2008.

[172] Yan, G., *et al.* Providing VANET security through active position detection.,Computer Communications 31(12): 2883-2897, 2008.

[173] Wei, Y.-C., *et al.* Rssi-based user centric anonymization for location privacy in vehicular networks. Security in Emerging Wireless Communication and Networking Systems, Springer: 39-51, 2010.

[174] Prado, A., Ruj, S., & Nayak, A. Enhanced privacy and reliability for secure geocasting in VANET.In Communications (ICC),IEEE International Conference on (pp. 1599-1603). IEEE, 2013.

[175] Lee, S.-B., *et al.* Secure incentives for commercial ad dissemination in vehicular networks. Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing, ACM, 2007.

[176] Sun, J., *et al.* An id-based framework achieving privacy and non-repudiation in vehicular ad hoc networks. Military Communications Conference, 2007. MILCOM 2007. IEEE, 2007.

[177] Zhang, C., *et al.* RAISE: an efficient RSU-aided message authentication scheme in vehicular communication networks. Communications, 2008. ICC'08. IEEE International Conference on, IEEE, 2008.

[178] Isaac, J. T., *et al.* A secure vehicle-to-roadside communication payment protocol in vehicular ad hoc networks. Computer Communications 31(10): 2478-2484, 2008.

[179] Wagan, A. A., *et al.* VANET security framework for trusted grouping using TPM hardware. Communication Software and Networks, 2010. ICCSN'10. Second International Conference on, IEEE, 2010.

[180] Biswas, S., & Misic, J. A Cross-layer Approach to Privacy-preserving Authentication in WAVE-enabled VANETs. Vehicular Technology, IEEE Transactions on 62(5): 2182-2192, 2013.

[181] Leinmuller, T., Schoch, E., Maihofer, C. Security requirements and solution concepts in vehicular ad hoc networks. In IEEE 4th Annual Conference on Wireless on Demand Network Systems and Services, pp. 8491, 2007.

[182] SeVeCom, Secure Vehicular Communications: Security Architecture and Mechanisms for V2V/V2I, Delivarable 2.1 (2007-2008), http://www.sevecom.org

[183] Batina, Lejla, Stefaan Seys, Dave Singele, and Ingrid Verbauwhede. Hierarchical ECC-based RFID authentication protocol. In RFID. Security and Privacy, pp. 183-201. Springer Berlin Heidelberg, 2012.

[184] Arbit, Alex, Yoel Livne, Yossef Oren, and Avishai Wool. Implementing public-key cryptography on passive RFID tags is practical. International Journal of Information Security (2014): 1-15.

Publications

[1] M.S.I. Mamun, Atsuko Miyaji. **An Optimized Signature Verification System forVehicle Ad hoc NETwork**. The 8th IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2012), pp 1-8, Shanghai, China.

[2] J. Chen, M.S.I. Mamun, Atsuko Miyaji. **An Efficient Batch Verification System for Large Scale VANET**. International Journal of Security and Communication Networks (SCN), Wiley Publication, 2014. DOI: 10.1002/sec.980. (extended version of WiCOM 2012)

[3] M.S.I. Mamun, Atsuko Miyaji, M. Rahman. **Secure and Private RFID Authentication Protocol under SLPN Problem**. The 6th International Conference on Network and System Security (NSS 2012), LNCS 7645 (2012), Springer-Verlag, pp 476-489, Fujian, China.

[4] M.S.I. Mamun, Atsuko Miyaji. **A Privacy-preserving Efficient RFID Authentication Protocol from SLPN Assumption**. International Journal of Computational Science and Engineering (IJCSE), Special Issue on Converged Networks, Technologies and Applications, Volume 9, Inderscience Publication, 2014. (extended version of NSS 2012)

[5] M.S.I. Mamun, Atsuko Miyaji. **A Fully-secure RFID Authentication Protocol from Exact LPN**. The 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2013), page 102-109, Melbourne, Australia.

[6] M.S.I. Mamun, Atsuko Miyaji. **An LPN-based RFID Authentication Protocol where Reader-Server Channel is Insecure**, Journal of Cryptographic Engineering, Springer. (Under Submission: extended version of TrustCom 2013)

[7] M.S.I. Mamun, Atsuko Miyaji. **A Scalable Secure RFID Ownership Transfer Protocol for a Large Supply Chain**. The 28th IEEE International Conference on Advanced Information Networking and Applications (AINA 2014), page 343-350, Victoria, Canada.

[8] M.S.I. Mamun, Atsuko Miyaji. **RFID Path Authentication, Revisited**. The 28th IEEE International Conference on Advanced Information Networking and Applications (AINA 2014), page 245-252, Victoria, Canada.

[9] M.S.I. Mamun, Atsuko Miyaji. **Secure VANET Applications with a refined Group Signature**, The 12th Annual Conference on Privacy, Security and Trust (PST 2014), Toronto, Canada.

[10] M.S.I. Mamun, Atsuko Miyaji, Hiroaki Takada. **A multi-purpose Group Signature for Vehicular Network Security**. International Workshop on Trustworthy Computing, in conjunction with (NBiS 2014), Salerno, Italy.