Private Multiparty Set Intersection Protocol in Rational Model

Keita Emura[†], Atsuko Miyaji[††], Mohammad Shahriar Rahman[†††]
[†] National Institute of Information and Communications Technology (NICT),
Tokyo, Japan. Email: k-emura@nict.go.jp
[††]Japan Advanced Institute of Science and Technology (JAIST),
Ishikawa, Japan. Email: miyaji@jaist.ac.jp
[†††] Department of Computer Science and Engineering, University of Asia Pacific (UAP),
Dhaka, Bangladesh. Email: shahriar.rahman@uap-bd.edu

***Abstract*—Privacy-preserving set intersection protocol is desirable for many practical applications. Malicious and semi-honest adversarial models in cryptographic settings have been considered mostly to design such protocols for privacy-preserving set intersection. In a semi-honest or malicious model an adversary is assumed to follow or arbitrarily deviate from the protocol, respectively. Protocols in semi-honest model can utilize cheaper cryptographic primitives, but that comes with a cost of weaker security. On the other hand, strong security is guaranteed by the malicious model whereby expensive cyptographic primitives are required. However, achieving a desired level of privacy with efficient computation is what we need for practical implementations. In this paper, we address the multiparty private set intersection problem using simple cryptographic primitives, in which each of the N parties learns no elements other than the intersection of their N private datasets. The private set intersection is constructed in game-theoretic model, where instead of being semi-honest or malicious the parties are viewed as rational and are assumed (only) to act in their own self-interest. We consider both single player deviation and coalitions, and show that our protocol satisfies computational strict Nash equilibrium.**

KeyWords: Set intersection, Game theory, Computational Strict Nash equilibrium, Commutative Encryption.

I. Introduction

A huge amount of sensitive data is produced by day-to-day business operational applications. A major utility of databases today is available from external sources such as market research organizations, independent surveys and quality testing labs, scientific or economic research. Data mining is used to efficiently determine valuable sensitive knowledge patterns from large databases. However, this is often not possible due to the confidentiality issues which leads to concerns over privacy infringement while performing the data mining operations. To address the privacy problem, several privacy-preserving data mining protocols using cryptographic techniques have been suggested. In data mining area, private set intersection protocols allow parties to interact on their respective input sets. These protocols address several realistic privacy issues. For example, to determine which customers appear on a do-not-receive-advertisements list, a store must perform a set intersection operation between its private customer list and the producer's list.

Privacy-preserving set intersection protocols use different models based on the adversarial behavior assumptions. Two main categories of adversaries have been considered in cryptographic literature:

Semi-honest adversaries: Following Goldreich's definition [13], protocols secure in the presence of semi-honest adversaries (or honest-but-curious) assume that parties faithfully follow all protocol specifications and do not misrepresent any information related to their inputs, e.g., set size and content. However, during or after protocol execution, any party might (passively) attempt to infer additional information about the other party's input. This model is formalized by requiring that each party does not learn more information that it would in an ideal implementation relying on a trusted third party (TTP).

Malicious adversaries: Security in the presence of malicious parties allows arbitrary deviations from the protocol. Under this model, parties may refuse to participate in the protocol, modify their private input sets, or abort the protocol prematurely, and these cases can not be prevented by such security model.

The assumption of semi-honest behavior may be unrealistic in some settings. In such cases, parties may prefer to use a protocol that is secure against malicious behavior. It is clear that the protocols secure in the malicious model offer more security. However, these are not efficient enough to be used in practice. Most of these constructions use general zero-knowledge proofs for fully malicious multiparty computation (MPC) protocols. These zero-knowledge compilers lead to rather inefficient constructions [34]. In typical cryptographic MPC protocols, parties are allowed to abort when they can find some malicious behavior from other parties. This means that the parties have to start the protocol from the scratch which is undesirable for operations on huge data sets.

Protocols for some cryptographic tasks (e.g., secret sharing, multiparty computation) have begun to be re-evaluated in a game-theoretic light (see [9], [24] for an overview) since the work of Halpern and Teague [17]. In this setting, parties are neither honest nor corrupt but are instead viewed as rational and are assumed (only) to act in their own self-interest. This feature is particularly interesting for data mining operations where huge collection of data is used, since parties will not

deviate (i.e., abort) as there is no incentive to do so. In many real-world settings, parties are willing to actively deviate/cheat, but only if they are not caught.

A. Related Work

Cryptographic techniques have been used to design many different distributed privacy-preserving data mining algorithms. In general, there are two types of assumptions on data distribution: vertical and horizontal partitioning. In the case of horizontally partitioned data, different sites collect the same set of information about different entities. For example, different credit card companies may collect credit card transactions of different individuals. Secure distributed protocols have been developed for horizontally partitioned data for mining decision trees [28], k-means clustering [27], k-nn classifiers [22]. In the case of vertically partitioned data, it is assumed that different sites collect information about the same set of entities but they collect different feature sets. For example, both a university and a hospital may collect information about a student. Again, secure protocols for the vertically partitioned case have been developed for mining association rules [40], and k-means clusters [20], [39]. In [38] the authors propose a user-centric private k-means algorithm combined with a decentralized cryptographic protocol and a gossip-based protocol in multiparty setting. All of those previous protocols claimed to be secure only in the semi-honest model. In [11], [23], authors present two-party secure protocols in the malicious model for data mining. They follow the generic malicious model definitions from the cryptographic literature, and also focus on the security issues in the malicious model, and provide the malicious versions of the subprotocols commonly used in previous privacy-preserving data mining algorithms. Assuming that at least one party behaves in semi-honest model, they use threshold homomorphic encryption for malicious adversaries presented by Cramer et al. [7]. Since homomorphic encryption is considered too expensive [30] and zero-knowledge proof is often one of the most expensive parts of cryptographic protocols, the protocols proposed in malicious adversarial model are not very practical for operations on large data items. Set operations using commutative encryption have been proposed in [2], where the adversaries have been considered as semi-honest parties. Commutative property of encryption schemes has also been considered in [5] for collusion resistant anonymous data collection in a malicious model. Game theory and data mining, in general, have been combined in [21], [36] for constructing various data mining algorithms. Rational adversaries have also been considered in privacy-preserving set operations [41], [3]. These protocols consider Nash equilibrium to analyze the rational behavior of the participating entities. As in all of cryptography, computational relaxations are meaningful and should be considered; doing so allows us to get around the limitations of the information-theoretic setting. So, analyzing set operations from the viewpoint of computational Nash equilibrium is interesting, since it gives a more realistic results. Fairness and privacy have been considered in [4], [16], [18].The use of game-theoretic concepts for capturing the cryptographic properties of privacy and fairness in two-party protocols is shown by [4]. Based on [4], the properties of two-message oblivious transfer protocols is game-theoretic concept is characterized in [18]. It presents a single two-player game, where it captures the cryptographic properties of correctness and privacy in the presence of malicious adversaries. [16] shows that it is possible to perform rational fair computation for arbitrary functions and utilities as long as the parties have a strict incentive to compute the function in the ideal world where fairness in guaranteed. This approach shows that the impossibility results claimed in [4] are not due to the inherent limitations of rational parties, rather due to the choice of specific functions and utilities. However, how to handle multiparty cases for fairness and privacy is not clear from the these protocols. Again, there have been several works on game theory based MPC/secret sharing schemes [1], [17], [26], [32], [12], [37], [19]. But [17], [37] require the continual involvement of the dealer even after the initial shares have been distributed or assume that sufficiently many parties behave honestly during the computation phase. Some schemes [1], [26], [32] rely on multiple invocations of protocols. Other work [19] relies on physical assumptions such as secure envelopes and ballot boxes. [12] proposed efficient protocols for rational secret sharing. But secret sharing schemes cannot be directly used for our purpose since they require much heavier computation, the existence of TTP, and their set up is different.

B. Our Contribution

In this work, we build multiparty private set-intersection protocol in game-theoretic setting using cryptographic primitives[1]. It is assumed that parties are neither honest nor corrupt but are instead rational and are assumed to act only in their own self-interest. Our construction avoids the use of expensive tools like homomorphic encryption, zero knowledge proof, and oblivious transfer. We have used commutative encryption and unique signatures as the underlying cryptographic primitives which are simple and efficient. The parties run the protocol in a sequence of r rounds and learn the complete result at the end of the r-th round. We also show that our protocol satisfies computational version of strict Nash equilibrium. In short, our protocol achieves the following:

- Any party may cheat with incorrect input. But cheating does not help winning the game.
- At any round earlier than r, aborting the protocol does not give any higher pay off to the aborting party than following the protocol.
- The protocol is resilient to coalitions of upto $t-1$ parties under the assumption that exactly t parties are active during the computation phase.

Organization of the paper: The remainder of the paper is organized as follows: Section II presents the background,

[1]Previously, two-party private set-intersection protocol in game-theoretic setting has been proposed [35].

definitions, and model. Section III includes protocol construction. In Section IV, we analyze the protocol formally. We give some concluding remarks in Section V.

II. Background and Preliminary

A. Cryptographic Considerations in Game Theory

Achieving a secure protocol is the objective in the cryptographic setting. Eliminating the trusted party is one of the main tasks while maintaining the privacy. On the other hand, in game theory, some particular equilibrium is defined to achieve stability. The existence of the trusted party/mediator is a parameter setting resulting in a more desirable, but harder to implement equilibrium concept for rational behaviors. Thus, privacy is a goal in the cryptographic setting while in the game theory setting it is a means to an end.

Games are treated in a modified way with a differently defined equilibrium notions in a cryptographic setting with. Katz, in [24], gives some examples of how this can be done for the specific case of parties running a protocol in the cryptographic setting. A security parameter n is introduced which is provided to all parties at the beginning of the game. The action of a player P_j now corresponds to running an interactive Turing Machine (TM) T_j. The T_j takes the current state and messages received from the other party as the input, and outputs message of player P_j along with updated state. The message m_j is sent to the other party. In a computational sense, it is required that T_j runs in PPT meaning that the function is computed in time polynomial in n. T_j is thus allowed to run for an unbounded number of rounds and, it can be added that the expected number of rounds is also polynomial for which T_j runs. The security parameter n is given as input to the utility functions. Utility functions map transcripts of a protocol execution to the reals that can be computed in time polynomial in n. Let Δ be a computational game in which the actions of each player correspond to the PPT TMs. Also, the utilities of each player are computed in time polynomial in n. Thus, mixed strategies are no longer needed to be considered, since a polynomial time mixed strategy corresponds to a pure strategy (since pure strategies correspond to randomized TMs) [24]. The parties are not assumed to be curious in negligible changes in their utilities, and this is an important difference between the cryptographic setting and the setting that has been considered here.

B. Definitions and Model

In this section, we will state the definitions of computational strict Nash equilibrium, Commutative encryption, and Unique signatures. A protocol is in Nash equilibrium if no deviations are advantageous. In other words, there is no incentive to deviate in the case of a Nash equilibrium. We assume that a party exhibits its malicious behavior by aborting early or sending non-participate message. However, a malicious party does not manipulate its own datasets to provide wrong data. Preventing malicious parties from sharing false data is difficult since the data are private and non-verifiable information. To prevent such malicious behavior, there can be auditing mechanism where a TTP can verify the integrity of data. Further investigation is needed to thwart this kind of misbehavior without a TTP. In this regard, mechanism design could be a potential tool to motivate parties to share their data. We denote the security parameter by l. A function ϵ is negligible if for all $c > 0$ there is a $l_c > 0$ such that $\epsilon(l) < 1/l^c$ for all $l > l_c$; let $negl$ denote a generic negligible function. We say ϵ is noticeable if there exist c, l_c such that $\epsilon(l) > 1/l^c$ for all $l > l_c$.

We consider the strategies in our work as the PPT interactive Turing machines. Given a vector of strategies $\vec{\sigma}$ for t^* parties in the computation phase, let $u_j(\vec{\sigma})$ denote the expected utility of P_j, where the expected utility is a function of the security parameter n. This expectation is taken over the randomness of the players' strategies. Following the standard game-theoretic notation, $(\sigma'_j, \vec{\sigma}_{-j})$ denotes the strategy vector $\vec{\sigma}$ with P_j's strategy changed to σ'_j. Given that j parties are active during the computation phase, let the outcome o of the computation phase be a vector of length j with $o_j = 1$ iff the output of P_j is equal to the exact intersection (i.e., P_j learns the correct output). Let $\nu_j(o)$ be the utility of player P_j for the outcome o. Following [17], [12], we make the following assumptions about the utility functions of the players:

- If $o_j > o'_j$, then $\nu(o_j) > \nu(o'_j)$.
- If $o_j = o'_j$ and $\sum_j o_j < \sum_j o'_j$, then $\nu(o_j) > \nu(o'_j)$.

In other words, player P_j first prefers outcomes in which he learns the output; otherwise, P_j prefers strategies in which the fewest number of other players learn the result. From the point of view of P_j, we consider the following three cases of utilities for the outcome o where $U^* > U > U'$:

- If only P_j learns the output, then $\nu_j(o) = U^*$.
- If P_j learns the output and at least one other player does also, then $\nu_j(o) = U$.
- If P_j does not learn the output, then $\nu_j(o) = U'$.

So, we have the expected utility of a party who outputs a random guess for the output[2] (assuming other parties abort without any output, or with the wrong output) as follows:

$$U_{rand} = \frac{1}{\mid \mathcal{D} \mid} \cdot U^* + (1 - \frac{1}{\mid \mathcal{D} \mid}) \cdot U'.$$

Also, we assume that $U > U_{rand}$; else players have almost no incentive to run the computation phase at all. As in [12], we make no distinction between outputting the wrong secret and outputting a special 'don't know' symbol- both are considered as a failure to output the correct output.

We begin by giving definitions for the case of single-player deviations, and then consider the case of coalitions.

Definition 1: Π induces a computational Nash equilibrium if for any set $J = \{j_1, \ldots, j_{t^*}\}$ of $t^* \geq t$ parties who are

[2]We do not consider U''- the utility when neither party learns the output, since 'not learning the output' is not the target of a rational adversary in practice.

active during the computation phase, any $j \in J$, and any PPT strategy σ_1' we have $u_j(\sigma_j', \vec{\sigma}_j) \leq u_j(\vec{\sigma}_j) + negl(l)$.

Definition 2: Fix a set $J \subseteq n$ of t or more parties active during computation phase, an index $j \in J$, and a strategy γ_j. Define $view_{-j}^{\Pi}$ as follows:

All parties play their prescribed strategies. Let mes denote the messages sent by P_j, but not including any messages sent by P_j after it writes to its output tape. Then $view_{-j}^{\Pi}$ includes the information given by the trusted party to all parties in $J \setminus j$, the random coins of all parties in $J \setminus j$, and the (partial) transcript mes.

Given algorithm A, define the random variable $view_{-j}^{A,\gamma_j}$ as follows:

Strategy γ_j yields equivalent play with respect to Π, denoted $\gamma_j \approx \Pi$, if there exists a PPT algorithm A such that for all PPT distinguishers D

$$| \Pr[D(1^l, view_{-j}^{A,\gamma_j}) = 1] - \Pr[D(1^l, view_{-j}^{\Pi}) = 1] | \leq negl(l)$$

With this in place, we can define a computational strict Nash equilibrium.

Definition 3: Π induces a computational strict Nash equilibrium if

- Π induces a computational Nash equilibrium;
- For any set $J \subseteq n$ with $|J| \geq t$, any $j \in J$, and any PPT strategy σ_j' for which $\sigma_j' \not\approx \Pi$, there is a $c > 0$ such that $u_j(\vec{\sigma}) \geq u_j(\sigma_j', \vec{\sigma}_{-j}) + 1/l^c$ for infinitely many values of l.

We view a coalition C as a set of parties who may coordinate their strategies in an arbitrary way. Since the coalition acts in unison, we treat the utility of the coalition as a whole and, in particular, view the coalition as having only a single output value (rather than viewing each member of the coalition as potentially outputting a different value). Let $\nu_C(\cdot)$ denote the utility of the coalition C. As before, we assume the following utilities:

- If o is an outcome in which C learns the secret and no player outside C does, then $\nu_C(o) = U^*$.
- If o is an outcome in which all parties active during the computation phase (including C) learn the secret, then $\nu_C(o) = U$.
- If o is an outcome in which C does not learn the secret, then $\nu_C(o) = U'$.

If $\vec{\sigma} = (\sigma_C, \vec{\sigma}_C)$ then $U_C(\vec{\sigma})$ denotes the expected utility of C when parties in C follow σ_C and the remaining parties follow σ_{-C} .

Definition 4: Π induces an r-resilient computational Nash equilibrium if for any set $J \subseteq n$ of t-or-more players active during the computation phase, any $C \subset J$ with $|C| \leq r$, and any PPT strategy σ_C' we have

$$U_C = (\sigma_C', \vec{\sigma}_C) \leq U_C(\vec{\sigma}) + negl(l)$$

We define the notion of two coalition strategies σ_C, σ_C' yielding equivalent play in a manner analogous to Definition 3, except that now the transcript included in $view_{-C}^{\Pi}$ does not include messages sent by the parties in C once any party in C writes its output.

Definition 5: Π induces an r-resilient computational strict Nash equilibrium if

- Π induces an r-resilient computational Nash equilibrium;
- For any set $C \subset J \subseteq n$ with $|J| \geq t$ and $|C| \leq r$, and any PPT strategy σ_C' for which $\sigma_C' \not\approx \Pi$, there is a $c > 0$ such that $U_C(\vec{\sigma}) \geq U_C(\sigma_C', \vec{\sigma}_{-C}) + 1/l^c$ for infinitely many values of l.

Commutative Encryption: Here, we define commutative encryption by using [2] as a reference. Briefly, a commutative encryption is a pair of encryption functions f and g such that $f(g(v)) = g(f(v))$. By using this property, each party can apply their function regardless of its computation order, and can get the same result. Moreover, we can ensure that a party cannot compute the encryption of a value without the help of others.

Definition 6 (Commutative Encryption [2]): A commutative encryption F is a computable (in polynomial time) function $f : \mathsf{KeyF} \times \mathsf{DomF} \to \mathsf{DomF}$, defined on finite computable domains, that satisfies all properties listed below. We denote $f_e(x) \equiv f(e, x)$.

1) Commutativity: For all $e, e' \in \mathsf{KeyF}$ we have $f_e \circ f_{e'} = f_{e'} \circ f_e$
2) Each $f_e : \mathsf{DomF} \to \mathsf{DomF}$ is a bijection.
3) The inverse f_e^{-1} is also computable in polynomial time given e.
4) The distribution of $\langle x, f_e(x), y, f_e(y) \rangle$ is indistinguishable from the distribution of $\langle x, f_e(x), y, z \rangle$, where $x, y, z \in_r \mathsf{DomF}$ and $e \in_r \mathsf{KeyF}$.

Due to the property 1, the result of encryption with two different keys is the same regardless of the order of encryption. Due to the property 2, no collision occure, i.e., two different values will never have the same encrypted value. Anyone who knows e can compute the inverse of f efficently due to the property 3. Due to the property 4, $f_e(y)$ and a random z are indistinguishable even $(x, f_e(x), y)$ are given. Note that the adversary does not control the choice of x since it is required that property 4 holds only if x is a random value from DomF. Indistinguishability is formaly defined as follows.

Definition 7 (Indistinguishability [2]): Let $\Omega_l \in \{0,1\}^l$ be a finite domain of l-bit numbers. Let $D_1 = D_1(\Omega_l)$ and $D_2 = D_2(\Omega_l)$ be distributions over l. Let $A_l(x)$ be an algorithm that, given $x \in \Omega_l$, returns either true or false. We define distribution D_1 of random variable $x \in \Omega_l$ to be computationally indistinguishable from distribution D_2 if for any family of PPT algorithms $A_l(x)$, any polynomial $p(l)$, and all sufficiently large l

$$Pr[A_l(x)|x \in D_1] - Pr[A_l(x)|x \in D_2] < \frac{1}{p(l)}$$

where x is distributed according to D_1 or D_2, and $Pr[A_l(x)]$ is the probability that $A_l(x)$ returns true.

Next we give an example of commutative encryption, which is given in [2]. Let p be a safe prime, where $q := (p-1)/2$ is also prime, and $\mathbb{G}$ be a group with prime order q. Then, $\mathsf{DomF} := \mathbb{G}$ and $\mathsf{KeyF} := \mathbb{Z}_q$. f is defined as $f_e(x) = x^e \bmod p$. Since the group order q is knowen, f_e^{-1} can be computed such that $f_e^{-1} = f_{e^{-1}}$. Now, we assume that the DDH (desicion Diffie-Hellman) assumption holds, where $\langle g, g^a, g_1^a, g_1^a \rangle$ and $\langle g, g^a, g_1^a, g_1^b \rangle$ are indistinguishable. Here g is a generator of $\mathbb{G}$ and $a, b \in_r \mathbb{Z}_q$. Then, $\langle x, f_e(x), y, f_e(y) \rangle = \langle x, x^e, y, y^e \rangle$ and $\langle x, x^e, y, z \rangle$ are indistinguishable under the DDH assumption.

Unique Signatures: Next, we define unique signature by using [31] as a reference. Briefly, the Sign algorithm is deterministic in unique signatures. In [31], a unique signature is constructed under the Gap DH assumption, where the computational DH problem, where given $(g, g^a, g^b) \in \mathbb{G}$ for a generator g and random values $a, b \in \mathbb{Z}_q$ and compute g^{ab}, is hard and the DDH problem is easy (i.e., bilinear groups).

Definition 8 (Unique Signatures [31]): A function family $\Sigma_{(\cdot)}(\cdot) : \{0,1\}^l \mapsto \{0,1\}^{s(l)}$ is a unique signature scheme (US) if there exists probabilistic algorithm, efficient deterministic algorithm, and probabilistic algorithm such that $\mathsf{G}(1^l)$ generates the key pair PK, SK, $\mathsf{Sign}(SK, x)$ computes the value $\Sigma = \Sigma_{PK}(x)$ and $\mathsf{Verify}(PK, x, \Sigma)$ verifies that $\Sigma = \Sigma_{PK}(x)$. More formally,

1) (Uniqueness of $\Sigma_{PK}(m)$) There do not exist values $(PK, m, \Sigma_1, \Sigma_2)$ such that $\Sigma_1 \neq \Sigma_2$ and $\mathsf{Verify}(PK, m, \Sigma_1) = \mathsf{Verify}(PK, m, \Sigma_2) = 1$.
2) (Security) For all families of PPT machines $A_l^{(\cdot)}$, there exists a negligible function $negl(l)$ such that

$$\begin{aligned}&\Pr[(PK, SK) \leftarrow \mathsf{G}(1^l);\\ &\quad (Q, x, \Sigma) \leftarrow A_l^{\mathsf{Sign}(SK,\cdot)}(1^l);\\ &\quad \mathsf{Verify}(PK, x, \Sigma) = 1 \wedge (x, \Sigma) \notin Q] \leq negl(l),\end{aligned}$$

where the contents of the query tape are denoted by Q.

III. Rational Set-Intersection Protocol

In this section, we give our rational set-intersection protocol. In a typical protocol, parties are viewed as either honest or semi-honest/malicious. To model rationality, we consider players' utilities. Let $\mathcal{D}$ be the domain of output which is polynomial in size. The function returns a vector I that represents the set-intersection where I_m is set to one if item m is in the set intersection. In other words, for all the data items of the parties (i.e., $D_{1,\ldots,n}$), we will compute $D_1 \cap D_2 \cap \ldots \cap D_n$, and we get I as the output of the function.

A. An Overview of the Protocol

Our protocol is composed of two stages, where the first stage can be viewed as a key generation stage and the second stage that computes the intersection takes place in a sequence of $r = r(n)$ iterations. More specifically, in the key generation stage the parties generate their encryption keys. They also choose $i^* \in r$ according to some random distribution α in which step they can learn the complete intersection result. In every round $i \in \{1, \ldots, r\}$ the parties exchange the encrypted data for the current round, which enables P_j perform the Intersection Computation. Clearly, when all the parties are honest, the parties produce the same output result which is uniformly distributed. Briefly speaking, the stages have the following form:

Key Generation Stage:

- Each party j randomly chooses a secret key for itself, i.e. $e_j \in \mathsf{KeyF}$ for commutative encryption. Also, it chooses PK_j, SK_j for its unique signature.
- A value $i^* \in \{1, \ldots, r\}$ is chosen according to some random distribution $0 < \alpha < 1$ where α depends on the players' utilities (discussed later). This represents the iteration, in which parties will learn the complete result.

Intersection Computation Stage:

Each party encrypts its items with its key, signs the encrypted result and passes these along to the other parties. On receiving a set of (encrypted) items and corresponding signatures, a party verifies signatures, encrypts each item and permutes the order before sending it to the next party. This is repeated until every item has been encrypted by every party. Since encryption is commutative, the resulting values from different sets will be equal if and only if the original values were the same (i.e., the item was present in the sets). Thus, we need only take the values that are present in all of the encrypted itemsets. This can be done by any party. More concretely, in each iteration i, for $i = 1, \ldots, r$, the parties do the following: First, P_j sends c_j and $\mathsf{Sign}(SK_j, c_j)$ to P_k (where P_k is any other party in J) and then after verifying the signature P_k sends c_k and $\mathsf{Sign}(SK_k, c_k)$ to P_j, where c_k and c_j are the ciphertexts computed by party P_k and P_j respectively. After receiving the ciphertexts, P_j and P_k compute the set-intersection using commutative property of the encryption scheme. If a party aborts in some iteration i, then the other parties output the value computed in the previous iteration. If some party fails to follow the protocol, the other parties abort. In fact, it is rational for P_j to follow the protocol as long as the expected gain of deviating is positive only if P_j aborts exactly in iteration i^*; and is outweighed by the expected loss if P_j aborts before iteration i^*. The intersection computation phase proceeds in a series of iterations, where each iteration consists of one message sent by each party.

B. Protocol Construction

As described above, our protocol Π consists of two stages. Let p be an arbitrary polynomial, and set $r = p \cdot \mid Y \mid$. We implement the first stage of Π using a key generation algorithm. This functionality returns required keys to all the parties. In the second stage of Π, the parties exchange their ciphertexts in a sequence of r iterations. The protocol returns I at the end of the operations on all the data items as follows:

Key Generation Stage:

- Each party j randomly chooses a secret key $e_j \in \mathsf{KeyF}$ for commutative encryption. Also, it generates keys for signatures PK_j, SK_j.

- A value $i^* \in \{1, \ldots, r\}$ is chosen according to some random distribution $0 < \alpha < 1$ where α depends on the players' utilities. This represents the iteration, in which parties will learn the complete result.

Set Intersection Computation Stage:
for all j do

1) P_j encrypts its input dataset $Z_j = f_{e_j}(D_j)$, signs Z_j as $\Sigma_j = \mathsf{Sign}(SK_j, Z_j)$ and sends Z_j, Σ_j along to the other parties P_k.
2) P_k encrypts its input dataset $Z_k = f_{e_k}(D_k)$, signs Z_k as $\Sigma_k = \mathsf{Sign}(SK_k, Z_k)$ and sends Z_k, Σ_k along to the other parties P_j.
3) For P_j, if it has not received any messages from P_k or fails to verify Σ_k, then output the result of iteration $i-1$ and halt. Otherwise, compute $Z'_j = f_{e_j}(f_{e_k}(D_k))$ and sends the pairs $\langle Z_k, Z'_j \rangle$ to the other parties P_k.
4) For P_k, if it has not received any message from P_j or fails to verify Σ_j, then output the result of iteration $i-1$ and halt.. Otherwise, compute $Z'_k = f_{e_k}(f_{e_j}(D_j))$. Also, from pairs $\langle f_{e_k}(D_k), f_{e_j}(f_{e_k}(D_k)) \rangle$ obtained in previous step for each $d_k \in D_k$, it creates pair $\langle d_k, f_{e_j}(f_{e_k}(d_k)) \rangle$ replacing $f_{e_k}(d_k)$ with corresponding d_k.
5) For P_k, for $d_k \in D_k$ for which $(f_{e_j}(f_{e_k}(d_k)) \in Z'_k$, these values form the intersection result $I = D_k \cap D_j$. P_k publishes this result.

Remark: In [31], a message m to be signed is encoded by an error correcting code C such that $C(m)$, and the code C is publicly available. So, we can assume that a ciphertext $Z_j = f_{e_j}(D_j)$ is first encoded, and then this encoded-ciphertext is signed by using the Sign algorithm. That is, we can use different groups for the underlying commutative encryption (i.e., a DDH-hard group) and the underlying unique signature (i.e., a Gap-DH group), respectively.

IV. Protocol Analysis

Single Player Deviation: Here we will give some intuition as to why the computation phase of Π is a computational strict Nash equilibrium for an appropriate choice of α. Let us assume that P_j follows the protocol, and any single party P_k deviates from the protocol. When P_k aborts in some iteration $i < i^*$, the best strategy P_k can adopt is to output $Z_k^{i^*}$ hoping that $i = i^*$. Thus, following this strategy, the expected utility that P_k obtains can be calculated as follows:

- P_k aborts exactly in iteration $i = i^*$. In this case, the utility that P_k gets is at most U^*.
- When $i < i^*$, P_k has 'no information' about correct I and so the best it can do is guess. In this case, the expected utility of P_k is at most U_{rand}.

Considering the above, P_k's expected utility of following this strategy is at most:

$$\alpha \times U^* + (1 - \alpha) \times U_{rand}$$

Now, it is possible to set the value of α such that the expected utility of this strategy is strictly less than U, since $U_{rand} < U$ by assumption. In such a case, P_k has no incentive to deviate. Since there is always a unique valid message a party can send and anything else is treated as an abort, it follows that the protocol Π induces a computational Nash equilibrium.

Theorem 1: The protocol Π induces a computational strict Nash equilibrium given that

$$0 < \alpha < 1, \quad U > \alpha \times U^* + (1 - \alpha) \times U_{rand}$$

and the properties of commutative encryption and unique signatures.

Proof: We first show that Π is a valid set-intersection protocol. Computational secrecy follows from the proof, below, that the intersection computation is a computational strict Nash equilibrium. Because if secrecy did not hold then computing the output locally and not participating in the intersection computation phase at all would be a profitable deviation. We next focus on correctness. Assuming all the parties run the protocol honestly, the output is computed correctly if the properties of commutative encryption are not achieved, which has negligible probability. We next show that Π induces a computational Nash equilibrium. Assume P_j follows the strategy σ_j prescribed by the protocol, and let σ'_k denote any PPT strategy followed by P_k. In a given execution of the computation phase, let i denote the iteration in which P_k aborts (where an incorrect message is viewed as an abort); if P_k never aborts then set $i = 1$. Let $early$ be the event that $i < i^*$; let $exact$ be the event that $i = i^*$; and let $late$ be the event that $i > i^*$. Let $correct$ be the event that P_k outputs the correct output. We will consider the probabilities of these events in two experiments: the experiment defined by running the actual intersection computation scheme, and a second experiment where P_k is given Z_k, t_k chosen uniformly at random from the appropriate ranges. We denote the probabilities in the first experiment by $\Pr_{real}[\cdot]$, and the probabilities in the second experiment by $\Pr_{ideal}[\cdot]$. We have the following equation using the fact (as discussed above) that whenever late occurs P_j outputs the correct result. Since when all the parties follow the protocol P_k gets utility U, we need to show that there exists a negligible function ϵ such that

$$\begin{aligned} & u_k(\sigma'_k, \sigma_j) \leq U + \epsilon(n) : \\ & u_k(\sigma'_k, \sigma_j) \leq U^* \times \Pr_{real}[exact] \\ & + U^* \times \Pr_{real}[correct \wedge early] \\ & + U' \times \Pr_{real}[\overline{correct} \wedge early] \\ & + U \times \Pr_{real}[late] \end{aligned}$$

Now we have the following claim that follows from the indistinguishability property of commutative encryption and unique signatures:

Claim 1: There exists a negligible function ϵ such that

$$\begin{aligned} |\Pr_{real}[exact] &- \Pr_{ideal}[exact]| \leq \epsilon(l) \\ |\Pr_{real}[late] &- \Pr_{ideal}[late]| \leq \epsilon(l) \\ |\Pr_{real}[correct \wedge early] &- \Pr_{ideal}[correct \wedge early]| \leq \epsilon(l) \\ |\Pr_{real}[\overline{correct} \wedge early] &- \Pr_{ideal}[\overline{correct} \wedge early]| \leq \epsilon(l) \end{aligned}$$

Now, we have

$$\begin{aligned} U_{ideal} = U^* \cdot \Pr_{ideal}[exact] + U^* \cdot \Pr_{ideal}[correct \wedge early] \\ + U' \cdot \Pr_{ideal}[\overline{correct} \wedge early] + U \cdot \Pr_{ideal}[late] \end{aligned}$$

From Claim 1 we get that

$$|u_k(\sigma'_k, \sigma_j) - U_{ideal}| \leq \epsilon(l)$$

for some negligible ϵ. We bound U_{ideal} as follows: Let $abort = exact \vee early$, so that $abort$ is the event that P_k aborts before iteration $i^* + 1$. We have

$$\Pr_{ideal}[exact \mid abort] = \alpha$$

and

$$\Pr_{ideal}[correct \mid early] = 1/\mathcal{D}.$$

It is easy to find that

$$U_{ideal} = U + (\alpha \cdot U^* + (1-\alpha) \cdot U_{rand} - U) \cdot \Pr_{ideal}[abort] \leq U$$

given that

$$\alpha \cdot U^* + (1-\alpha) \cdot U_{rand} - U < 0.$$

This shows that Π induces a computational Nash equilibrium.

That Π induces a computational strict Nash equilibrium follows easily from the above analysis along with the fact that there is always a unique valid message each party can send. Specifically, say P_k plays a strategy σ'_k with $\sigma'_k \not\approx \Pi$. This implies that

$$\Pr_{ideal}[abort] \geq 1/poly(l)$$

for infinitely many values of l. Claim 1 then shows that

$$\Pr_{ideal}[abort] \geq = 1/poly(l)$$

for infinitely many values of l, and so $U - U_{ideal} \geq 1/poly(l)$ infinitely often as well. Since $|u_k(\sigma'_k, \sigma_j) - U_{ideal}|$ is negligible, we conclude that

$$U - u_k(\sigma'_k, \sigma_j) \geq 1/poly(l)$$

for infinitely many values of l, as required.

Coalitions:

Theorem 2: If $\alpha > 0, U > \alpha \times U^* + (1-\alpha) \times U_{rand}$, and the properties of commutative encryption and unique signatures hold, then Π induces a $(t-1)$-resilient computational strict Nash equilibrium given that exactly t parties are active during the computation phase.

The proof is similar to that of Theorem 1. We give a sketch of the proof here. Let us assume some set of t parties J running the computation phase, and consider some coalition $C \subset J$ of size at most $t-1$. Let P' be any player in J but not in C. As usual, the best strategy for C is to not abort until it can definitively identify iteration r^*, which occurs only after it receives the iteration-(r^*+1) message from P'. But P' only sends its iteration-(r^*+1) message after it has received (valid) iteration-r^* messages from all the parties in C. By this point, no matter what the parties in C do, P' has the correct result written on its output tape.

Till now we have assumed that exactly t parties are active during the computation phase. However, as a general case, during the protocol execution when the t parties run Π and all other parties remain silent - that protocol is not a $(t-1)$-resilient computational Nash equilibrium if $t^* > t$ parties are active. Briefly explaining, let the active parties be $I = \{1, \ldots t+1\}$ and let $C = \{3, \ldots t+1\}$ be a coalition of $t-1$ parties. When P_1 and P_2 send their values in each iteration r, the parties in C can compute $t+1$ values. Due to the computation steps, when $r = r^*$, the C parties are guaranteed to lie on $(t-1)$ values, but when $r < r^*$, are unlikely to lie on $(t-1)$ values. This helps the C parties to determine r^* as soon as that iteration is reached. At this point, preventing P_1 and P_2 from outputting the result, they can abort and output the result. To overcome this, we let the parties run Π for t^*, where t^* denotes the number of active players. It follows as an easy corollary of Theorem 2 that this induces a $(t-1)$-resilient computational strict Nash equilibrium regardless of how many parties are active during the computation phase.

V. Conclusion

In this paper, we have proposed a private multiparty set-intersection protocol from the view point of rational cryptography. We have used tools like commutative encryption and unique signatures as the underlying cryptographic primitives which are simple and efficient. Also, we show that our protocol satisfies computational strict Nash equilibrium in the case of single party deviations or coalitions.

References

[1] Abraham, I., Dolev, D., Gonen, R., and Halpern, J.: Distributed Computing Meets Game Theory: Robust Mechanisms for Rational Secret Sharing and Multi-party Computation. In 25th ACM Symposium Annual on Principles of Distributed Computing, pp. 53-62, 2006.

[2] Agrawal, R., Evfimievski, A., and Srikant, R.: Information Sharing Across Private Databases. In the Proceedings of the ACM SIGMOD Conference, pp. 86-97, 2003.

[3] Agrawal, R. and Terzi, E.: On Honesty in Sovereign Information Sharing. In the 10th International Conference on Extending Database Technology- EDBT'06, pp. 240-256 2006.

[4] Asharov, G., Canetti, R., and Hazay, C.: Towards a Game Theoretic View of Secure Computation. In Advances in Cryptology- EUROCRYPT'11, pp. 426-445, 2011.

[5] Ashrafi, M.Z. and Ng, S-K.: Collusion-resistant anonymous data collection method. In the Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 69-78, 2009.

[6] Bellare, M. and Micali, S.: Non-interactive Oblivious Transfer and Applications. In Advances in Cryptology- CRYPTO'89. pp. 547-557, 1989.

[7] Cramer, R., Damgard, I., and Nielsen, J.B.: Multi-party Computation from Threshold Homomorphic Encryption. In Advances in Cryptology-EUROCRYPT'01, pp. 280-299, 2001.

[8] Dodis, Y.: Efficient Construction of (distributed) Verifiable Random Functions. In 6th International Workshop on Theory and Practice in Public Key Cryptography- PKC'03, pp. 1-17, 2003.

[9] Dodis, Y. and Rabin, T.: Cryptography and Game Theory. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, Algorithmic Game Theory, pp. 181-207, Cambridge University Press, 2007.

[10] Dodis, Y. and Yampolskiy, A.: A Verifiable Random Function with Short Proofs and Keys. In 8th International Workshop on Theory and Practice in Public Key Cryptography- PKC'05, pp. 416-431, 2005.

[11] Emura, K., Miyaji, A., and Rahman, M.S.: Efficient Privacy-Preserving Data Mining in Malicious Model. In The 6th International Conference on Advanced Data Mining and Applications, ADMA'10. pp. 370-382, 2010.

[12] Fuchsbauer, G., Katz, J., and Naccache, D.: Efficient Rational Secret Sharing in Standard Communication Networks. In Theory of Cryptography- TCC'10, pp. 419-436, 2010.

[13] Goldreich, O.: Foundations of cryptography: Basic applications. Cambridge Univ. Press, Cambridge, 2004.

[14] Goldwasser, S, and Ostrovsky, R.: Invariant signatures and non-interactive zero-knowledge proofs are equivalent. In Advances in Cryptology CRYPTO'92, pp. 228-244, 1992.

[15] Gordon, S.D., Hazay, C., Katz, J., Lindell, Y.: Complete Fairness in Secure Two-party Computation. In 40th Annual ACM Symposium on Theory of Computing- STOC'08, pp. 413-422, 2008.

[16] Groce, A. and Katz, J.: Fair Computation with Rational Players. In Advances in Cryptology- EUROCRYPT'12, pp. 81-98, 2012.

[17] Halpern, J. and Teague, V.: Rational Secret Sharing and Multi-party Computation: Extended abstract. In 36th Annual ACM Symposium on Theory of Computing- STOC'04, pp. 623-632, 2004.

[18] Higo, H., Tanaka, K., Yamada, A., and Yasunaga, K.: A Game-Theoretic Perspective on Oblivious Transfer. In 17th Australasian Conference on Information Security and Privacy- ACISP'12, pp. 29-42, 2012.

[19] Izmalkov, S., Micali, S., and Lepinski, M.: Rational Secure Computation and Ideal Mechanism Design. In 46th Annual Symposium on Foundations of Computer Science- FOCS'05, pp. 585-595, 2005.

[20] Jagannathan, G. and Wright, R.N.: Privacy-preserving Distributed k-means Clustering over Arbitrarily Partitioned Data. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining-KDD'05, pp. 593-599, 2005.

[21] Jiang, W., Clifton, C. and Kantarcioglu, M.: Transforming Semi-Honest Protocols to Ensure Accountability. In Data and Knowledge Engineering (DKE), 65(1), pp. 57-74, 2008.

[22] Kantarcioglu, M. and Clifton, C.: Privately Computing a Distributed k-nn Classifier. In 7th European Conference on Principles and Practice of Knowledge Discovery in Databases- PKDD'04, pp. 279-290, 2004.

[23] Kantarcioglu, M., and Kardes, O.: Privacy-preserving Data Mining in the Malicious model. In International Journal of Information and Computer Security, Vol. 2, No. 4, pp. 353-375, 2008.

[24] Katz, J.: Bridging Game Theory and Cryptography: Recent Results and Future Directions. In Theory of Cryptography- TCC'08. pp. 251-272, 2008.

[25] Katz, J.: On Achieving the Best of Both Worlds in Secure Multi-party Computation. In 39th Annual ACM Symposium on Theory of Computing- STOC'07, pp. 11-20, 2007.

[26] Kol, G. and Naor, M.: Cryptography and Game Theory: Designing Protocols for Exchanging Information. In Theory of Cryptography-TCC'08, pp. 320-339, 2008.

[27] Lin, X., Clifton, C. and Zhu, M.: Privacy-preserving Clustering with Distributed EM Mixture Modeling. In Knowledge and Information Systems, July, Vol. 8, No. 1, pp. 68-81, 2005.

[28] Lindell, Y. and Pinkas, B.: Privacy-preserving Data Mining. In Advances in Cryptology- CRYPTO'00, pp. 36-54, 2000.

[29] Lysyanskaya, A.: Unique Signatures and Verifiable Random Functions from the DH-DDH Separation. In Advances in Cryptology-CRYPTO'02, pp. 597-612, 2002.

[30] Liu, J., Lu, Y.H., and Koh, C.K.: Performance Analysis of Arithmetic Operations in Homomorphic Encryption. In ECE Technical Reports, Purdue University, 2010.

[31] Lysyanskaya, A.: Unique Signatures and Verifiable Random Functions from the DH-DDH Separation. In Advances in Cryptology-CRYPTO'02, pp. 597-612, 2002.

[32] Lysyanskaya, A., Triandopoulos, N.: Rationality and Adversarial behavior in Multi-party computation. In Advances in Cryptology-CRYPTO'06, pp. 180-197, 2006.

[33] Micali, S., Rabin, M. O., and Vadhan, S. P.: Verifiable Random Functions. In 40th Annual Symposium on Foundations of Computer Science-FOCS'99, pp. 120-130, 1999.

[34] Miyaji, A., and Rahman, M.S.: Privacy-preserving Data Mining in Presence of Covert Adversaries. In the 6th International Conference on Advanced Data Mining and Applications, ADMA'10. pp. 429-440, 2010.

[35] Miyaji, A., and Rahman, M.S.: Privacy-Preserving Set Operations in the Presence of Rational Parties. In the Fifth International Symposium on Mining and Web, MAW'12, pp. 869-874, 2012.

[36] Nix, R. and Kantarcioglu, M.: Incentive Compatible Distributed Data Mining. In IEEE International Conference on Privacy, Security, Risk and Trust, pp. 735-742, 2010.

[37] Ong, S. J., Parkes, D., Rosen, A., and Vadhan, S.: Fairness with an Honest Minority and a Rational Majority. In Theory of Cryptography-TCC'09, pp. 36-53, 2009.

[38] Sakuma, J. and Kobayashi, S.: Large-scale k-means clustering with user-centric privacy-preservation. In Knowl. Inf. Syst., Vol 25, No. 2, pp. 253-279, 2010.

[39] Su, C., Bao, F., Zhou, J., Takagi, T., Sakurai, K.: Security and Correctness Analysis on Privacy-Preserving k-Means Clustering Schemes. In IEICE Trans. Fundamentals, Vol.E92-A, No.4, pp. 1246-1250, 2009.

[40] Vaidya, J. and Clifton, C.: Privacy Preserving Association Rule Mining in Vertically Partitioned Data. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining- KDD'02, pp. 639-644, 2002.

[41] Zhang, N. and Zhao, W.: Distributed Privacy-preserving Information Sharing. In the 31st International Conference on Very large data bases-VLDB'05, pp. 889-900, 2005.