Algebraic formalization of security protocol

Naoki Kinjo

Language Design Labratory
School of Information Science,
Japan Advanced Institute of Science and Technology

February 15, 2001

Keywords: Security protocol, NSPK Protocol, Specification, CafeOBJ.

The purposes of our research are to describe a specification of the security protocol according some formalism in order to verify, and to consider the described specification.

The aims of the security protocol are a secret communication using cryptographies, and are to apply the electronic commerce, the electronic auctions, the vote on network and so on. Therefore, some protocols are proposed. In case of that protocols are practicalities, the safety has to be verified in advance. In discussing about the safety of security protocol, it is necessary to separate from the applied cryptography. Even if the cryptography is perfect, a maliciousness user can know secret data using the bug, when a protocol has a bug. The correctness of security protocol often has been depended on man's instinct. However, it is expected that the correctness of security protocol is verified with formal method, because simple problems with some well-known protocols were found by using it.

CafeOBJ is a multi-paradigm algebraic specification language, which is a successor of OBJ. CafeOBJ is based on the combination of several logics consisting of order sorted algebra, rewriting logic and hidden algebra. According to its semantics, CafeOBJ can fit in several specification and programming paradigms such as equational specifications, rewriting logic specification, behavioral specification. In recent year, CafeOBJ has proposed how to describe a specification of abstract state machine, which methodology is to depend some hidden signature and some equation. The specification described by this methodology is called behavioural specification.

There are some formalism about security protocol. After this, two formalism are introduced, G.Denker, J.Meseguer and C.Talcott formalism and Masami Hagiya, Yozo Toda and Yoshiki Fukuba formalism.

In order that Denker described a specification of security protocol, she used Configuration and described by Maude. Configuration was defined in order to formalize a

concurrent system model. A network protocol can be automatically expressed as concurrent system. There are some Object and some Message on Configuration. When Object receives zero or more Message, Configuration changes. Change of Configuration expresses rewrite rules.

Hagiya defined the state of network system. The state of network was expressed principal set and message set. A principal has a state for communicating. A message format depends in protocol. The state of network and principal change at one step of a protocol.

In our research, we describe security protocol by CafeOBJ according this two formalism. Denker's formalism expects rewriting logic specification. Hagiya's formalism expects behavioural specification.

Needham-Schroeder Public-Key Protocol (It is henceforth written as NSPK Protocol) is taken up as an exercise of a security protocol which we describe the specification. NSPK Protocol is adopted the public-key cryptosystem. The public-key cryptosystem has prepared two different keys, which called public-key and secret-key to encrypt and decrypt message. The message encrypted by a certain public-key can be decrypted only by corresponding secret-key as well as the message encrypted by a certain secret-key is decrypted only by corresponding public-key. However NSPK Protocol consists of seven steps, it omits about public-key distribution and discusses at three steps. In step1, Alice seeks to establish a connection with Bob by selecting a nonce *Na*, and sending it along with its identity toBob, encrypted using Bob's public-key (message1). When Bob receives this message, it decrypts the message to obtain the nonce *Na*. It returns the nonce *Na*, along with a new nonce *Nb*, to Alice, encrypted with Alice's public-key (message2). When Alice receives this message he should be assured that he is talking to Bob, since only Bob should be able to decrypt message1 to obtain *Na*. Alice returns the nonce *Nb* to Bob, encrypted with Bob's public-key. When Bob receives this message he should be assured that he is talking to Alice, since only Alice should be able to decrypt message2 to obtain *Nb*. Thus, NSPK Protocol proposes the authentication between persons by exchanging an encrypted message obtain a nonce. The nonce can ẗ be guessed in a random number. If the encrypted message includes a nonce, only the person, who generates itself or has the corresponding key, can see. However, Gavin Lowe disproved authentication by NSPK Protocol. Basis of Lowe's opinion is when Alice seeks Sam, and sends message1 to Sam, thus Sam receives Alice's message1, and connects with Bob using the nonce and Alice's identify to obtain Alice's message1, the result, Sam impersonates another Alice and communicates with Bob; we call a person like Sam Spy, and call a person like Alice and Bob Friend. Spy can also behave like Friend. This behaviour of Spy calls Lowe's Attack. Lowe also reported the revised edition of NSPK Protocol with Lowe's Attack. The revision details sender's identity adds to message2 in NSPK Protocol, which prevents Lowe's Attack.

We described the specification of these two protocols according to two formalism mentioned previously. One of two formalism applied rewriting logic specification in CafeOBJ, and another applied behavioural specification.

The rewriting logic specifications can be executable automatically to give a random Configuration. When the specification of NSPK Protocol gave the Configuration, which in Friend and Friend and Spy, it checked out Lowe's Attack, and the revision didn't. The

protocol wasn't complete, if Lowe's Attack applied the specification of NSPK Protocol revision. This means authentication isn't performed.

The behavioural specifications can be executable one step in protocol steps to give a random network state. The messages that can generate the principal on a network are checked for every step, and the possible following steps are checked. As a result, the possibility of Lowe's Attack was checked in the behavioural specification of NSPK Protocol, and a meaningless authentication was checked in the revision, which is two principals masquerade as the others.

The rewriting logic specification is easy to understand intuitively, is legible, and is executable automatically; this specification is a good simulator. However, this specification can't verify the safety of protocol. If this specification detects no protocol bug by the simulation, it is not necessarily the safety protocol. By the rewriting logic specification, it is necessary to enumerate all rules about the change of Configuration. However, human can write some rules only a part to know; it is not necessarily all rules, and the proof that it is all is difficult.

On the other hand, the behavioural specifications expect verification of security protocol, however it is difficult to intuitive understanding and not automatically to simulate comparing with the rewriting logic specification. We think the behaviour for the protocol being safe exists. If the behaviour for protocol being safe doesn 't change in any protocol steps, it is safe. And we expect the verification of safe in a security protocol is possible at induction. For that purpose, it is necessary to formalize the behaviour to express the safety of protocol.

In our research, we described the specification of NSPK Protocol in the rewriting logic specification and the behavior specification using CafeOBJ. The specification can execute on example, and check out Lowe's Attack in NSPK Protocol. But we don't describe the verification for a security protocol; therefore, safety needs to be formalize, more details.