Knowledge Based Approach for Mechanically Verifying Security Protocols

Xiaoqi Ma, Xiaochun Cheng and Rachel McCrindle
Department of Computer Science
The University of Reading
Whiteknights, Reading RG6 6AY, UK
xiaoqi.ma@reading.ac.uk

Abstract

A new knowledge-based security protocol verification approach is proposed in this paper. A number of predicates, functions, assumptions and rules are used to infer the knowledge of participating principals. These items are implemented with Isabelle, which enables mechanical proving. This approach can prove protocols concerning interleaving protocol sessions and can prove the correctness of a medium-sized security protocol in a couple of seconds. The mechanical proofs of a number of important secure properties and then of the correctness of the Needham-Schroeder-Lowe protocol are given as examples to show the effectiveness of this method.

1 Introduction

To evaluate and verify security protocols in a systematic way, significant research work has been conducted in the area of designing formal methods for analysis of cryptographic protocols, and many good models have been proposed. Generally speaking, all these methods can be broadly divided into two categories, namely state based methods and rule based methods.

State based methods model security protocols using finite state machines. They search the state space exhaustively to see whether all the reachable states are safe [Paulson, 1997]. If some reachable state in a security protocol is proved to be unsafe, a flaw may be reported; otherwise, the protocol will be reported correct and safe. State based methods are usually complete and can find most flaws in protocols.

Rule based methods formally express what principals can infer from messages received [Paulson, 1997]. In these approaches, the protocols, the necessary assumptions and the goals of the protocols are formulated in formal logic. Some specific properties of the protocols can be proved by using the axioms and rules of the logic. Since rule based methods do not have to search large state space, they can normally converge quickly.

To gain effectiveness from state based methods and efficiency from rule based methods, we propose in this paper a new security protocol verification method, which is based on a knowledge-based framework. The method analyses the knowledge of participating principals and infers what they can know and can never know. It takes protocols concerning multiple interleaving sessions into consideration and can find flaws which are often overlooked by many rule based methods. Unlike state based method, our method avoids searching large state space. By implementing the method in a mechanical reasoning platform, Isabelle [Nipkow *et al*., 2003], it can be used to mechanically verify cryptographic protocols and can converge very quickly. The Needham-Schroeder public key authentication protocol [Needham and Schroeder, 1978] with Lowe's fix [Lowe, 1995] is analysed in this paper as an example to show the effectiveness and efficiency of our method.

2 The Knowledge Based Method

Our method focuses on the knowledge of all participators in the protocol. We describe their initial knowledge and infer what they can and cannot know with the progress of the protocol. In other words, it concerns the knowledge analysis of all participators. This method can be implemented in Isabelle to enable mechanical verification.

All *principals* taking part in network communications can be divided into three categories: the *server*, the *friends*, and the *spy*.

Random numbers chosen by principals serve as *nonces* to identify protocol runs uniquely and to avoid replay attack. Every principal has some *keys*. All the nonces and keys are represented by natural numbers.

A message is a piece of information sent from one principal to another. A message consists of principals' names, nonces, keys, encrypted messages, signed messages, hashed messages, or a combination of these. It can be recursively defined in Isabelle.

We define a number of *functions* and *predicates*. Key functions map principals to their certain keys. Function *Nonce_of* maps principals to their nonces. To determine whether a message is a part of another one, we introduce the function *msg_part*.

The predicate *Know* describes a principal's knowledge about a certain message. Similarly, the predicate *Auth* describes one principal's authentication state about another principal on a certain message, i.e., whether the message is sent by the expected principal and is unmodified.

To describe cryptographic protocols, two *action functions* need to be introduced. One is *Send*, representing that one principal sends a message to another principal. Correspondingly, the other function is *Rcv*, meaning that a principal receives a certain message from others.

Our method is based on a number of assumptions, which are widely accepted by most researchers in this field.

We introduce a group of inference rules into the method to infer new knowledge from the old. All these rules can be divided into four categories: *encryption/decryption* rules, *message combination/separation* rules, *sending/receiving* rules and *authentication* rules.

3 Verifying Needham-Schroeder-Lowe Protocol Mechanically

To verify the Needham-Schroeder-Lowe protocol, we first model it in our framework, then prove a number of important properties, and finally prove the security guarantees.

Normally, the protocol can be formalised into three steps for all honest principals. The first step is an application of the *Send* action function for sending *A*'s nonce and name to *B*. The second step states that if the first step has been successfully carried out, principal *B* will correspondingly send a compound message consisting of *A*'s nonce, its own nonce and name to principal *A*. Similarly, the third step describes that if the second step has been successfully carried out, principal *A* will correspondingly send *B*'s nonce back to principal *B*.

These steps are sufficient for honest principals, but not for the spy who does not necessarily obey the rules. Therefore, we introduce four extra formulae. The first one is *Fake*, stating that the spy may send messages it knows to others. Another two formulae describe how honest principals respond to faked messages. The honest principals take the forged messages as protocol steps, and then respond them according to the protocol, as if they are legal messages.

To avoid interleaving attack, Lowe introduced the receiving principal's name into step 2 of the protocol [Lowe, 1995]. Accordingly, we introduce the *decline* rule: if the name in the message is not the name of the sender, the receiving principal should decline the message and terminate the communicating session.

With above modelling, we first prove some properties before we prove the final guarantees for the two participating principals.

One of these properties is that if a principal knows a message M, and M_1 is a part of M, then it should know the message M_1 as well. Due to the inductive definition of the data type *message*, we need to prove this lemma by induction. Several subgoals have been produced after we apply the induction command. The subgoals concerning principal, nonce, key and encrypted, signed and hashed massages can be proved by using the implication introduction, conjunction elimination and conjunction introduction rules of higher order logic. For subgoals concerning compound messages, we design a rule to decompose and resolve them automatically.

The correctness of the Needham-Schroeder-Lowe protocol relies greatly on the secrecy of the nonces used, and therefore the key point for proving the Needham-Schroeder-Lowe protocol is to prove the secrecy of nonces. We have two lemmas, one stating that the spy will never see *B*'s nonce, and the other describing that the spy will never see the content of the second protocol message, which is a compound message consisting of *A* and *B*'s nonces and *B*'s name.

We have to point out that the spy may intercept and know *A*'s nonce. But it does not affect the secrecy of the second protocol message sent from *B* to *A*.

We now can prove the guarantees. The guarantee for *B* after step 3 is that *B* authenticates that *A* has sent *B*'s nonce (encrypted by *B*'s public key) to *B* and this message has not been modified by the spy. The first authentication rule is applied and three subgoals are produced. With above lemmas, these subgoals can be easily solved by Isabelle. The guarantee for *A* can be similarly proved.

4 Conclusions

In this paper, we have presented a new framework to prove the correctness of security protocols. We model the protocols and infer them by analysing principals' knowledge.

To improve the efficiency of security protocol verification, we implement our framework using Isabelle. We have implemented all the data structures, functions, predicates, assumptions and inference rules in Isabelle. With this implementation, we are able to prove the correctness of protocols mechanically. All the proving details are generated by Isabelle, thereby saving users a significant amount of time. In addition, our framework also takes the cases concerning multiple interleaving sessions into consideration, making the method more powerful.

To show the effectiveness of our framework, we have used the Needham-Schroeder-Lowe public key authentication protocol as an example. The example shows how to use the framework and its implementation. Our implementation can prove the correctness of this protocol in a couple of seconds. Additionally, most of the codes can be reused for similar applications.

References

[Lowe, 1995] Gavin Lowe. An attack on the Needham-Schroeder public-key authentication protocol. *Information Processing Letters*, 56(3):131–133, 1995.

[Needham and Schroeder, 1978] Roger Needham and Michael Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999, 1978.

[Nipkow et al., 2003] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic.* Springer Verlag, Heidelberg, 2003.

[Paulson, 1997] Lawrence C. Paulson. Proving properties of security protocols by induction. In *Proceedings of the 10th Computer Security Foundations Workshop*, pages 70–83, Rockport, Massachusetts, June 1997.