A Simple and Efficient Approach to Verifying Cryptographic Protocols

SUN YONGXING[1], WANG XINMEI[2]

[1]*sunyongxing@263.net*
[2]*xmwang@xidian.edu.cn*
National Key Lab. on ISN
Xidian University
PO Box 710071
Xi'an
China
Tel: +86 29 820-1015

Key words: Cryptographic protocol, Formal tool, Restrictive channel, Equivalent message

Abstract: It is necessary to development the formal tools for verifying cryptographic protocols because of the subtlety of cryptographic protocols flaws; In terms of the notions of restrictive channel and equivalent message, this paper presents a approach that utilizes the substitution rules of messages and deduction rules to prove whether the insecure states of cryptographic protocols are reachable or not, and the analysis of a famous protocol shows the validity of the method.

1. INTRODUCTION

A cryptographic protocol is a communication protocol that employs cryptographic primitives to achieve goals such as distribution of cryptographic keys or authentication of principals, over a open network that may contain a number of hostile intruders who may be actively trying to subvert the goals of the protocol. In general, it is assumed that an intruder has complete control of all communication channels, and thus can read all traffic, destroy or alter traffic, and generate traffic of its own. In order to avoid cryptographic protocols flaws, it is necessary to design strong cryptographic algorithms and sound protocol structures. Cryptographic

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35515-3_53

S. Qing et al. (eds.), *Information Security for Global Information Infrastructures*

algorithms are always assumed to be directly unbreakable within a protocol, and thus primary focus is on the possibility of a penetrator using messages even when she might not be able to read and/or produce them herself. But, it is not surprising that it is difficult to design cryptographic protocols that are free of flaws that is independent of the strengths and weakness of the particular cryptographic algorithm used, so it is necessary to development some rigorous means of reasoning about cryptographic protocols.

There has been a great deal of work done in developing formal models of cryptographic protocols. As in the analysis of conventional communication protocols, there have been two kinds of techniques applied to this problem. One is to use logics of knowledge and belief to model the beliefs that evolve in the course of a protocol. The best known of these is the BAN logics [BURR 1990], but the BAN logics cannot exactly specify protocols from the view of protocol instances, while on the other hand, it is the confusion among these different protocol instances that results in most of cryptographic protocols flaws, so the logics and so-called improved like-BAN logics [GONG 90] [ABAD 91] [SYVE 94] [ORSC 93], have essential limits that cannot be overcome. Another is to model the protocol as an interaction between a set of state machines and to attempt to prove a protocol secure by specifying insecure states and attempting to prove them unreachable by exhaustive search backwards from the state, and the model-checking method has proved to be a very successful approach to analysing security protocols and avoids the shortcomings of the logics methods. A number of model-checking approaches have been presented, either with general purpose [KEMM 94] [MEAD 96] [LOWE 97] [MITC 97] [LOWE 98], or special purpose [MILL 87], [KEMM 94], but due to too large state space needed to search , these method have not high efficiency. Our purpose in this paper is to improve search efficiency.

This paper is organised as follows: in section 2, the notions of restrictive channel and equivalent message are given. In terms of the two notions , in section 3, we discuss the protocol specification and method used to prove whether insecure states are reachable or not by the presented substitution and deduction rules. In section 4, we apply our method to the BAN-Yahalom protocol given in the literature [BURR 90].

2. BASIC NOTIONS

In cryptographic protocols, we believe, the channel represents not only a physical link, but also a cryptographically logical connection between principals. In general, it is assumed that an intruder can always set up a physical link with any principal taking part in a cryptographic protocol, and

thus a channel is characterised only by its cryptographic features such as secret keys and cryptographic algorithms owned by principals. From the above consideration, the following definitions are introduced.

Definition 2.1: A *restrictive channel* is a channel with some access properties restriction.

A restrictive channel is characterised by its set of readers and its set of writers(i.e., the set of principals that can receive messages via the channel and the set of principals that can send messages via the channel), and the set of readers and the set of writers of a restrictive channel C are denoted by $r(C)$ and $w(C)$, respectively. It muse be pointed out that so-called readers and writers must have the ability to comprehend the content of messages received or sent, that is, the readers and writers must hold cryptographic informations corresponding to a restrictive channel, thus to use the restrictive channel, a principal needs information about how to read and/or write from/to the channel.

In general, the access properties on a restrictive channel are implemented by particular cryptographic algorithms. The cryptographic features of a restictive channel are denoted by $\{K(C^r), A(C^r)\}$ and $\{K(C^w), A(C^w)\}$, where $K(C^r)$ and $A(C^r)$ are the set of secret keys and the set of cryptographic algorithms possessed by readers, respectively, $K(C^w)$ and $A(C^w)$ are the set of secret keys and the set of cryptographic algorithms possessed by writers, respectively. Below, we give some basic types of restrictive channels.

Definition 2.2: C is a *public channel* iff anybody in the system can read and write it (i.e., $r(C) = w(C) = \Omega$, where Ω is the set of all principals). A public channel is denoted by $Public$.

Definition 2.3: C is a *confidential channel* iff anybody can write it, but only one principal P can read it (i.e., $r(C) = \{P\}$ and $w(C) = \Omega$). Confidential channel can be established by encryption with the public key of P, and is denoted by $\alpha\ P$(where κ_p is the public key of P).

Definition 2.4: C is a *secret channel* iff it can be used only by two principals P and Q(i.e., $r(C) = w(C) = \{P, Q\}$). A secret channel is denoted by $P \leftrightarrow Q$(where κ_{pq} is the share key between P and Q).

In the above definitions, it is assumed that the trusted third party will not use the secret keys of principals.

In terms of the notion of restrictive channel, we may define the cryptographic algorithm assumption as the following: A *cryptographic algorithm assumption* is to assume that the algorithm can guarantee the access properties of corresponding restrictive channels in every run of a protocol, that is, only $r(C)$ can read messages via the channel, and only $w(C)$ can send messages via the channel.

A message in cryptographic protocols can always be expressed as tuple {protocol name, session number, message number, receiver, sender, message data}, and a receiver always deduce other components in the tuple by the message data, although sometimes the deduction is not correct. We believe, it is the incorrect deduction that is the fundamental reason of cryptographic protocols flaws, thus the exact description of deduction ability of principals is critical in cryptographic protocols analysis. We observe that there are two kinds of elements in message data, the value of one kind of element is known by receivers before the message data is received, and thus is denoted as $const(X)$, the value of another kind of element is determined by message senders and obtained after the message data is received, and thus is denoted as $\mathrm{var}(X)$. From the above consideration, the following definitions are introduced.

Definition 2.5: A *principal's distinguishability* on a received message is defined as the set of the message structure, the values of message components denoted as $const(X)$, and the values scope of message components denoted as $\mathrm{var}(X)$, in the protocol specification.

A principal's distinguishability on a received message determines the difficulty of forging this message by an attacker, and thus protocols security. For simplicity, it is assumed that a receiver can always tell protocols name, in fact, this is easily done.

Definition 2.6: if some messages are concluded to belong to the same. tuple {protocol name, session number, message number, receiver, sender} by a principal, these messages is defined as *the equivalent messages with respect to the principal.*

the equivalence messages with respect to a principal are determined by the principal's distinguishability.

3. PROTOCOL ANALYSIS

3.1 Notations

In addition to the notations in section 2, the following symbols are introduced:

$id(P)$: the identifier of the principal P. We assume that the identifier is global unique.

$\#(X)$: the message component X has been not sent in a message at any time before the current run of the protocol.

$S(X)$:the message component can not be leaked in the protocol run, and its expiration is determined by the principal owning it.

a_i: the i th run of the protocol a.
$a_i.j$: the j th message of the i th run of the protocol a.
$P < (msg) \mid C$: the principal P has received a message msg via channel C.
$P > (msg) \mid C$: the principal P has sent a message msg via channel C.

3.2 The specification of protocols

We believe, A principal's behavior in a protocol run only depends on received and sent message data, being independent of true message receivers or senders. In fact, because it is assumed that an intruder can always read all traffic, destroy or alter traffic, and generate traffic of its own in a protocol run, a principal have not the ability to tell the true receiver or sender, and thus we may think the protocol run as concurrent with some synchronization mechanism. From the above consideration, we specify the protocol by separately specifying received and sent messages by each principal in a protocol.

3.3 The substitution rules of messages

It is assumed that a message receiver can separate message components by the component separator in the message, and does not detect whether there is any other component after obtaining the last component. We define the substitution rules of messages as the following:
(1) if the type of a message component is $const(X)$, the component cannot be replaced.
(2) if the type of a message component is $\mathrm{var}(X)$, the component can be substituted by components the type of which is the following :
: $const(X)$.
: $\mathrm{var}(X)$.
: if the message component is the last, it can be substituted by the concatenation of components the type of which is $const(X)$ or $\mathrm{var}(X)$.

The messages obtained by the above substitution operations are equivalent with the original message with respect to the receiver.

For simplicity, it is assumed that message components is string with no length limit. In fact, according to different protocols, the further restriction can be emerged into the above substitution rules, but our analysis method can still be applied to.

Theorem 1: The substitution rules of messages are correct.
Proof: By the notions of $const(X)$ and $\mathrm{var}(X)$.

Theorem 2: The substitution rules of messages are complete.
Proof: By the notions of $const(X)$ and $\mathrm{var}(X)$.

3.4 The protocol analysis procedure

It is assumed that principals believe that a protocol run is successfully completed only when he succeeds in verifying all received messages. We assume that the initial knowledge of an intruder is the following :

(1) The messages sent by all principals during the formal protocol runs.

(2) Some open components such as principal identifiers, public keys.

(3) If a principal is ready to send the k th message in a protocol run, the formal $(k-1)$ messages sent by the principal have been obtained by the intruder in the protocol run.

(4) If a principal has actively interacted with an intruder, all messages sent during the interaction have known by the intruder.

From the cryptographic algorithm assumption, we can obtain the following deduction rules:

Deduction rule 1: If a restrictive channel $C=(r(C),w(C))$, and an intruder $I(I \notin w(C))$ attempts to forging a message m on C, then m can be obtained only from messages sent by $w(C)$ on C.

Deduction rule 2: If a restrictive channel $C=(r(C),w(C))$, and an intruder $I(I \notin w(C))$ attempts to obtain the value of some component X in a message m, then the value can be found only in the following situations:

(1) X is a public component.

(2) X has already been sent on a public channel .

The protocol analysis procedure is as the following:

(1) The specification of the protocol.

(2) According to the security requirements of the protocol, the purposes of the intruder and the attacked principals are determined.

(3) From the specified protocol with respect to the attacked principals, the messages to be forged on the restrictive channel C and the message components to be obtained are determined.

(4) According to the specified protocol, the following steps are in turn executed to solve the messages and the components.

$S1$ If the intruder $I \in w(C)$, and the components used to forge a message are known by I, then the message are directly constructed.

$S2$ If there are not the components the type of which is $\#(X)$ in a message, then according to the substitution rules of messages, the corresponding equivalent messages with the obtained values of the components sent by $w(C)$ in the formal i runs of the protocol is used to replace the message.

$S3$ By the substitution rules of messages, observe the equivalent messages sent by $w(C)$ in the current run of the protocol with a message, and try to obtain the value of the components, if succeed, then substitute the message.

$S4$ Think of $w(C)$ and $r(C)$ as the challenge-response entity with the response rules determined by the specified protocol, and according to the substitution rules of messages and the deduction rules, solve the required messages and components. If the new messages or components needed to solve appear , then go to $S1$, until the required messages and component are obtained or are impossibly solved.

$S5$ Output the results.

4. CASE STUDY

In this section, to illustrate the validity of the method in section 3, we apply the presented method to the BAN-Yahalom protocol given in the literature [BURR 90]. Due to space constraints, we only present the illustrative analysis of the protocols, and do not give the complete analysis.

$$1 \quad A \to B : A, N_a$$

$$2 \quad B \to S : B, N_b, \{A, N_a\}_{K_{bs}}$$

$$3 \quad S \to A : N_b, \{B, K_{ab}, N_a\}_{K_{as}}, \{A, K_{ab}, N_b\}_{K_{bs}}$$

$$4 \quad A \to B : \{A, K_{ab}, N_b\}_{K_{bs}}, \{N_b\}_{K_{ab}}$$

BAN-Yahalom protocol

The analysis of the protocol:

1 The specification of the protocol

A:

$(1) > (id(A), \#(N_a)) \mid Public$

(2)

$$< (\mathrm{var}(X) \mid Public, (id(B), \mathrm{var}(Y), \#(N_a)) \mid A \overset{K_{as}}{\leftrightarrow} S, \mathrm{var}(Z) \mid Public)$$

$(3) > (\mathrm{var}(Z) \mid Public, \mathrm{var}(X) \mid A \overset{\mathrm{var}(Y)}{\leftrightarrow} B)$

B:

$(1) < (id(A), \mathrm{var}(X)) \mid Public$

$(2) > ((id(B), \#(N_b)) \mid Public, (id(A), \mathrm{var}(X)) \mid B \overset{K_{bs}}{\leftrightarrow} S)$

$(3) < ((id(A), \mathrm{var}(Y), \#(N_b)) \mid B \overset{K_{bs}}{\leftrightarrow} S, \#(N_b) \mid A \overset{\mathrm{var}(Y)}{\leftrightarrow} B)$

S:

$(1) < ((id(B), \mathrm{var}(X)) \mid Public, (id(A), \mathrm{var}(Y)) \mid B \overset{K_{bs}}{\leftrightarrow} S$

$(2) > (\mathrm{var}(X), (id(B), S(K_{ab}), \mathrm{var}(Y)) \mid A \overset{K_{as}}{\leftrightarrow} S, (id(A), S(K_{ab}), \mathrm{var}(X)) \mid B \overset{K_{bs}}{\leftrightarrow} S)$

2 It is assumed that the purpose of the intruder I is that the principal B believes $A \overset{K_{ab}}{\leftrightarrow} B$, but I knows K_{ab}, thus the intruder needs to attack B.

3 According to the specified protocol with respect to B, the intruder needs to forge the message (3), that is $((id(A), \text{var}(Y), \#(N_b)) \mid B \overset{K_{bs}}{\leftrightarrow} S$, received by B and obtain the value of $\text{var}(Y)$.

4 $S1$ It is impossible to directly construct the message.

$S2$ There is $\#(N_b)$ in the message.

$S3$ we denote this protocol run as a_j, and according to the substitution rules and the deduction rule 1, in turn search the corresponding equivalent messages in a_j. Consequently, the lookup fails.

$S4$ According to the deduction rule 1, we try to solve the required equivalent messages and the value of $\text{var}(Y)$ in the challenge-response system consisting of B or S.

Observing the specified protocol with respect to B, we can obtain that B sends $(id(A), \text{var}(X)) \mid B \overset{K_{bs}}{\leftrightarrow} S$ when $(id(A), \text{var}(X)) \mid Public$ is received, thus by the substitution rule , we replace $\text{var}(X)$ with $(\text{var}(Y)+\#(N_b))$ (we can at will select the value of $\text{var}(Y)$, and $\#(N_b)$ can be obtained from $a_j.2$). finally, we get the required solutions.

Observing the specified protocol with respect to S, we observe that S sends $(id(A), S(K_{ab}), \text{var}(X)) \mid B \overset{K_{bs}}{\leftrightarrow} S$, but from the deduction rule 2, K_{ab} can not be obtained.

5. CONCLUSION

In terms of the notions of restrictive channel and equivalent message, we present a kind of formal method used to verify cryptographic protocols, and illustrate the validity of the method by the analysis of the BAN-Yahalom protocol. We will continue to improve our method and make the further researches in the following areas:

(1) Combine other cryptographic primitives such as hash function with the presented method.

(2) Apply the automatic deduction techniques to prove whether insecure states are reachable or not.

6. REFERENCES

[MILL 87] MILLEN J.K.; CLARK S.C.; FREEDMAN S.B.; 1987; IEEE Transactions on Software Engineering; "The Interrogator: Protocol Security Analysis"; Vol. 13 no. 2.

[KEMM 94] KEMMERER R.; MEADOWS C.; MILLEN J.; 1994; Journal of Cryptology; "Three Systems for Cryptographic Protocol Analysis"; Vol. 7 no. 2.

[MEAD 96] MEADOWS C.; 1996; Journal of Logic Programming; "The NRL Protocol Analyzer: an Overview"; Vol. 26 no. 2.

[LOWE 97] LOWE G.; ROSCOE B.; 1997; IEEE Transactions on Software Engineering; "Using CSP to Detect Errors in the TMN Protocol"; Vol. 23 no. 10.

[MITC 97] MITCHELL J.C.; MITCHELL M.; STERN U.; 1997; IEEE Symposium on Security and Privacy ; "Automated Analysis of Cryptographic Protocols Using Murφ".

[LOWE 98] LOWE G.; 1998; "Towards a Completeness Result for ModelChecking of Security Protocols"; http: //www.mcs.le.ac.uk/~glowe/security/papers/completeness.ps.gz.

[BURR 90] BURROWS M.; ABADI M.; NEEDHAM R.; 1990; ACM Transaction on Computer Systems; "A Logic of Authentication"; Vol. 8, no. 1.

[GONG 90] GONG L.; NEEDHAM R.; YAHALOM R.; 1990; In Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy ; "Reasoning about Belief in Cryptographic Protocols".

[ABAD 91] ABADI M.; TUTTLE M.; 1991; In Proceedings of the 10th ACM Symposium on Principles of Distributed Computing ; "A Semantics for a Logic of Authentication".

[SYVE 94] SYVERSON P.; ORSCHOT P.C.V.; 1994; In Proceedings of 1994 IEEE Symposium on Security and Privacy ; "On Unifying some Cryptographic Protocol Logics".

[ORSC 93] ORSCHOT P.C.V.; 1993; In Proceedings of the First ACM Conference on Computer and Communications Security; "Extending Cryptographic Logics of Belief to Key Agreement Protocols".