密码协议的一种安全模型*

刘怡文[1+]， 李伟琴[1]， 冯登国[2]

[1](北京航空航天大学 计算机科学与工程系,北京 100083)
[2](中国科学院 软件研究所 信息安全国家重点实验室,北京 100080)

A Security Model for Cryptographic Protocols

LIU Yi-Wen[1+], LI Wei-Qin[1], FENG Deng-Guo[2]

[1](Department of Computer Science and Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)
[2](State Key Laboratory of Information Security, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)
+ Corresponding author: Phn: 86-10-66842714, E-mail: liuyw@buaa.edu.cn
http://www.buaa.edu.cn

Received 2002-05-14; Accepted 2002-09-04

Liu YW, Li WQ, Feng DG. A security model for cryptographic protocols. *Journal of Software*, 2003,14(6):1148~1156.
http://www.jos.org.cn/1000-9825/14/1148.htm

Abstract: In this paper, a cryptographic protocol together with its cryptographic algorithms is regarded as one system, and a security model for the system is built. Based on assume-guarantee compositional reasoning techniques, a new assume-guarantee based reasoning rule and algorithm are proposed, and the soundness of the rule is proved. In realizing model checking to the cryptographic protocol system, several difficulties are solved chiefly such as decomposition of the system, generation of assumed functions, and specifying security properties in forms of both logic formulas and processes. Using this security model and assume-guarantee based reasoning techniques, the kerberos cryptographic protocol system is verified.

Key words: protocol verification; security model; model checking; compositional reasoning

摘 要: 将密码协议与密码算法视为一个系统,建立了密码协议系统的一种安全模型.基于假设/保证的组合推理技术提出了新的假设/保证推理规则和假设/保证推理算法,证明了该规则的完备性,实现了密码协议系统的模型检查,并重点解决了系统分解问题、假设函数的设定问题、进程+逻辑的系统特性描述问题等难题.以 kerberos 密码协议系统为例,利用该安全模型和假设/保证推理技术对密码协议系统进行了安全验证.

关键词: 协议验证;安全模型;模型检查;组合推理

* Supported by the National Grand Fundamental Research 973 Program of China under Grant No.G1999035802 (国家重点基础研究发展规划(973))

LIU Yi-Wen was born in 1966. She is a Ph.D. candidate at the BUAA (Beijing University of Aeronautics and Astronautics). Her research interests are cryptographic protocols and protocol analysis. **LI Wei-Qin** was born in 1937. She is a professor and doctoral supervisor at the BUAA. Her research areas are computer network and security management. **FENG Deng-Guo** was born in 1965. He is a professor and doctoral supervisor at the Institute of Software, the Chinese Academy of Sciences. His current research areas are information and network security.

中图法分类号: TP309　　文献标识码: A

Cryptographic protocols are a sort of communication protocols based on cryptographic techniques, which use cryptographic algorithms and protocols to achieve objects of entity authentication and key distribution. Since the 80's of the 20th century, formal analyses of cryptographic protocols have been the international focus of research. In 20 years development, many formal methods such as BAN logic (Burrows, Abadi, Needham), model checking, and theorem proving have been put forward. Among all of these methods, the formal model[1] for security protocols built by Dolev and Yao is significant, based on which most of the formal analysis tools were built, such as Roscoe's FDR (Failures Divergences Refinement), Dill's Murφ, Meadows and Syverson's NRL (Naval Research Laboratory Analyzer), Millen's Interrogator, Paulson's inductive method, Isabeller, Thayer, Herzog and Guttman's Strand Space, Owre, Rushby, and Shankar's PVS (Prototype Verification System), etc.

In the Dolev-Yao model, cryptographic algorithms were regarded as a black box. So cryptographic algorithms are assumed perfect in all the formal methods based on Dolev-Yao model, which only verify protocol logic. These formal methods may have some disadvantages, because they can not find flaws caused by the interaction of the cryptographic algorithms and protocols, such as some attacks on the Otway-Rees protocol related to the cryptographic algorithm[2], known or chosen-plaintext attacks, analysis to encryption modes, etc. Obviously, it is unilateral to neglect the security strength of cryptographic algorithms. Considering the insecure factors of the cryptographic algorithms used in cryptographic protocols, it is needed to regard a cryptographic protocol and cryptographic algorithms used in the protocol as one system, called the cryptographic protocol system, and build a new security model for the cryptographic protocol system.

Model checking of cryptographic protocols is an effective formal method, but suffers from state explosion problems. This led to a series of researches on state reduction techniques. Compositional reasoning[3] is a sort of state reduction techniques, which mainly includes partitioned transition relations, lazy parallel composition, interface processes, assume-guarantee reasoning[4], etc. In the assume-guarantee reasoning, a system is divided into several components and each component is verified respectively. Since the behavior of each component depends on the behavior of the rest of the system, i.e., its environment, this method first assumes that the environment satisfies some properties, which are called assumptions, when these assumptions are satisfied, it can verify that the component satisfies other properties, which are called guarantees. By combining the set of assume/guarantee properties in an appropriate way, it can verify the properties of the whole system without constructing the global state transition graph. Using assume-guarantee based compositional reasoning in the analysis of the cryptographic protocol system, several difficulties must be overcome, foremost among which are the decomposition of the system, generation of assumed functions, and specifying security properties in forms of both logic formulas and processes, etc.

In this paper, a cryptographic protocol together with its cryptographic algorithms is regarded as one system, and a new security model for the system is built. Based on assume-guarantee compositional reasoning techniques, a new assume-guarantee based reasoning rule and algorithm are proposed, and the soundness of the rule is proved. This compositional reasoning method can be used to solve the model checking of the cryptographic protocol system and to avoid state explosion problems effectively. In realizing model checking to the cryptographic protocol system, several difficulties are solved chiefly such as decomposition of the system, generation of assumed functions, and specifying security properties in forms of both logic formulas and processes. As an example, using this security model and assume-guarantee based reasoning techniques, the kerberos cryptographic protocol system is verified, which illustrates the usefulness of this method.

1 The Security Model for Cryptographic Protocol Systems

A cryptographic protocol system is mainly composed of two components, the cryptographic protocol logic and the cryptographic algorithm process. Cryptographic algorithms mainly include stream ciphers, block ciphers, public-key cipher algorithms, and hash functions. Different types of cryptographic algorithms may satisfy different security requirements. In a cryptographic protocol system, the protocol and cryptographic algorithms can be regarded as a series of parallel components, in which the protocol provides initial inputs for cryptographic algorithms and waits for results being returned from the cryptographic algorithms. During one execution of the protocol, the same or different cryptographic algorithms can be invoked many times, and the security strength of these cryptographic algorithms is decided by the weakest one. For simplifying description, here we only discuss the system with a cryptographic protocol and a cryptographic algorithm, the two components.

1.1 The structural model for the protocol component

Using Kripke structure, the finite state model of cryptographic protocols is defined as follows:

Definition 1.1 (**The Structural Model for Cryptographic Protocols**). The structural model for cryptographic protocols is a tuple $M=\langle S,S_0,R,A,\mathscr{L},F\rangle$, where S is the finite state set of a protocol; $S_0\subseteq S$ is the initial state set of the protocol; $R\subseteq S\times S$ is the set of transition relations of the protocol; A is the finite set of proposition formulae, it includes all the security properties for the protocol to satisfy, which form the security requirement database[5]; $\mathscr{L}$ is a function which maps each state to the set of proposition formulae true in that state, $\mathscr{L}$: $S\rightarrow 2^A$; F is a set of final states or acceptable states which satisfy fairness constrains. A fair path $\pi=s_0\ldots s_is_j\ldots s_n$ is one run of the protocol, with $R(s_i,s_j)$, $s_0\in S_0$, $s_i,s_j\in S$, $s_n\in F$, $i,j\in N$, N is a set of natural number.

Definition 1.2 (**The Attack Model of the Intruder**). The attack model simulates the abilities of the intruder, which can select at will each action below in every state transition process of the protocol run:

① Intercept each message and prevent it from transmitting;

② Decompose messages into components, and remember them;

③ Generate nonces according to necessary;

④ Compose known messages into new messages, and send them.

In the attack model, a set of intruder rules is built explicitly in the form of temporal logic, which includes the message intercepting and remembering rules, the message decomposing rule, the message composing rule, the nonce generation rule, and known-key attack, impersonation attack, tampering attack, replay attack, interleaving attack, reflection attack, chosen-text attack, reset attack, forward search attack, conspiracy attack, etc.

Definition 1.3 (**Protocol's Satisfaction of Security Requirements**). Protocol's satisfaction of security requirements is defined as follows: for every $\forall\varphi\in A$, $M\models\varphi$, where

$M\models\varphi$, if and only if $\forall s_0\in S_0$, $s_0\models\varphi$ if $(\varphi\in\mathscr{L}(s)\wedge s\in S\wedge\exists\ \pi=s_0\ldots s\ldots s_n)$.

$M\models\neg\varphi$, if and only if $\exists s_0\in S_0$, $s_0\models\neg\varphi$ if $(\varphi\notin\mathscr{L}(s)\wedge s\in S\wedge\exists\ \pi=s_0\ldots s\ldots s_n)$.

1.2 The structural model for the cryptographic algorithm component

Cryptographic algorithms can be divided in detail into stream ciphers, block ciphers, public-key cipher algorithms, and one-way hash functions, which can be discussed as follows:

- Stream ciphers: A stream cipher algorithm itself is a finite state automaton, which has the internal states representing the current states of the cryptographic algorithm, the next state functions (the state transition functions), and the output functions.

- Block ciphers: Since block ciphers cannot remember states generally, it is difficult to build a finite state structure for them. Most block ciphers are based on the sequential repetitions of a round function, and the output of i-round depends on that of the previous round, so we can represent the output of every round as the internal states of a block cipher algorithm, and represent the output of the final round as the output. This way we can check the security properties of every round and find whether the security consistence is violated among each round.

• One-Way hash functions: A message M of arbitrary finite length is divided into fixed length blocks M_i (i=1,2,…,n), and its hash value is computed through several rounds of iterated processing. Representing the hash values of each round as internal states, and the hash value of the final round as output, we can construct the finite state structure for one-way hash functions.

• Public-Key cipher algorithms: Except TAO Ren-ji's finite state automaton public-key ciphers, public-key cipher algorithms are regarded difficult to be represented as the form of state structure. Here, we concentrate on the descriptions and analyses of the processes of encryption and signature using public-key algorithms.

Generally, all of the cryptographic algorithms can be represented as processes, which include a series of input variables, control variables, output variables, and sentences of assignments, conditions, and loops. If each different value of input variables and control variables denotes a state, and each executable sentence invoking state transition denotes a state transition function, the Kripke structure of cryptographic algorithms is defined as follows:

Definition 1.4 (**The Structural Model for Cryptographic Algorithms**). The structural model for cryptographic algorithms is a tuple $M'=\langle S',S'_0,R',A',\mathscr{L}',F'\rangle$, where S' is the finite state set of a cryptographic algorithm. Suppose V is a set of variables appearing in the cryptographic algorithm process, which can be divided into a set of input variables V^i, a set of control variables V^k, and a set of output variables V^o. Each different value of $xi\in V^i+V^k$ denotes a state $s'=x_0x_1x_2\ldots$, $s'\in S'$; $S'_0\subseteq S'$ is the initial state set of the cryptographic algorithm, which corresponds to the set of initial values of variables; $R'\subseteq S'\times S'$ is the set of state transition functions of the cryptographic algorithm. Suppose s_i' is a current state, and s'_j is a next state, then $R'(s'_i,s'_j)$ is an executable sentence or a cryptographic algorithm modular invoking the state transition; A' is the finite set of security properties for the cryptographic algorithm to satisfy, including a series of cryptographic attacks which the cryptographic algorithm should resist (see Definition 1.5); $\mathscr{L}'$ is a function which maps each state to the set of cryptographic attacks which the cryptographic algorithm can resist, $\mathscr{L}'$: $S'\rightarrow 2^{A'}$; F' is a set of final states or acceptable states. A fair path $\pi'=s'_0\ldots s'_is'_j\ldots s'_n$ is one run of the cryptographic algorithm process, with $R'(s'_i,s'_j)$, $s'_0\in S'_0$, $s'_i,s'_j\in S'$, $s'_n\in F'$, $i,j\in N$.

Definition 1.5 (**The Cryptographic Attack Model**). Suppose that the cryptographic algorithm is open, whose security strength depends on the security of keys. In this cryptographic attack model, there is a cryptanalyst who can do a series of cryptographic attacks to the cryptographic algorithm, such as a ciphertext-only attack, a known-plaintext attack, a chosen-plaintext attack, an adaptive chosen-plaintext attack, a chosen-ciphertext attack, an adaptive chosen-ciphertext attack, a chosen-key attack, and a verifiable text attack, etc. According to different types of cryptographic algorithms, these cryptographic attacks can be divided in details as follows[6~8]:

① Attacks on stream ciphers: Statistical tests (including the frequency test, the serial test, the poker test, the runs test, the auto-correlation test and the mutual-correlation test), linear complexity test, linear consistency test, chaos test, code follow test, correlation attack, divide-and-conquer attack, embedding correlation attack, primitive polynomial test, subsequence attack, inserting attack, etc.

② Attacks on block ciphers: Statistical tests (including the balance test, the code follow test, plaintext-ciphertext independence), completeness cryptanalysis, the avalanche effect, correlation test, diffusion cryptanalysis, non-linear cryptanalysis, differential cryptanalysis, linear cryptanalysis, partitioning cryptanalysis, differential-linear cryptanalysis, interpolation attacks, the output bit independence criterion, reversibility of key schedule, related-key attack, boomerang attack, rectangle attack, meet-in-the-middle attack, etc. Attacks on encryption modes are: tampering attack, impersonation attack, exchange ciphertext block attack, block replay attack, etc.

③ Attacks on public-key cipher algorithms: Attacks of big integer factorization, attacks of computing discrete logarithm over finite fields, attacks of computing discrete logarithm in elliptic curves, attacks on knapsack problems, attacks to modulus length, common modulus attack, small encryption/decryption exponent attacks, cycling attack, primality tests, lattice-based attack, statistical tests, etc.

④ Attacks on one-way hash functions: Attacks based on properties of underling ciphers (such as the complementation property, weak keys, fix-point attacks, and key collisions), collision attack, pseudo-collision attack, compression function attack, chaining attack, birthday attack, etc.

Besides, there are some attacks on password-based authentication, such as verifiable password attack, replay of fixed passwords, password-guessing attack, dictionary attack, etc. All these attacks (stated above) are stored in the cryptographic attack database in forms of processes.

Definition 1.6 (**Cryptographic Algorithm's Satisfaction of Security Requirements**). Algorithm's satisfaction of security requirements is defined as follows: $\forall \varphi' \in A'$, $M' \models \varphi'$, where $M' \models \varphi'$, if and only if, $\forall s'_0 \in S'_0$, $s'_0 \models \varphi'$ if $(\varphi' \in \mathscr{L}'(s') \wedge s' \in S' \wedge \exists\, \pi' = s'_0 \ldots s' \ldots s'_n)$.

1.3 The security model for cryptographic protocol systems

Considering each run of a cryptographic protocol system, the interconnection between the protocol and one cryptographic algorithm used in the cryptographic protocol system is sequential, in which there is a relation of call. By composing the sequential components of the protocol and the cryptographic algorithm, the structural model for cryptographic protocol systems is defined as follows:

Definition 1.7 (The Security Model for Cryptographic Protocol Systems). The security model M'' for cryptographic protocol systems is a sequential composition of M and M', which is represented as $M /\!/ M'$, $M'' = \langle S'', S''_0, R'', A'', \mathscr{L}'', F'' \rangle$, where $S'' = S \times S'$; $S''_0 = S_0 \times S'_0$; $R''((s_i, s'_t)(s_j, s'_k)) = R(s_i, s_j) \wedge R'(s'_t, s'_k)$, $s_i, s_j \in S$, $s'_t, s'_k \in S'$; $A'' = A \cup A'$; $\mathscr{L}''(s, s') = \mathscr{L}(s) \cup \mathscr{L}'(s')$; $F'' = F \cup F'$.

2 Assume-Guarantee Based Compositional Reasoning

2.1 Preliminaries

O. Grumberg and D.E. Long have defined the preorder $\leqslant$ and its properties[9]. Now we introduce the part which we will use as follows:

Definition 2.1. For structure M and M'', $s \in S$ and $s'' \in S''$, $(M'', S'') \leqslant (M, S)$, if there is a homomorphism from (M'', S'') to (M, S). $M'' \leqslant M$, if there is a homomorphism from M'' to M.

Theorem 2.2.

① $\leqslant$ is a preorder.

② For all structure M and M', $M /\!/ M' \leqslant M$.

③ For all structure M , M' and M'', if $M \leqslant M'$ then $M'' /\!/ M \leqslant M'' /\!/ M'$.

④ For all structure M, $M \leqslant M /\!/ M$.

Corollary 2.3. Suppose $M \leqslant M'$, then for every $\varphi \in A'$, $M' \models \varphi$ implies $M \models \varphi$.

If formula φ can be represented as a structure, the following corollary can be obtained:

Corollary 2.4. For every $\varphi \in A$, $M \models \varphi$ if and only if $M \leqslant \varphi$.

2.2 A new assume-guarantee based reasoning rule

Theorem 2.5. Suppose that M'' is the structure of a cryptographic protocol system be composed of a cryptographic protocol and a cryptographic algorithm, M is the structure of the cryptographic protocol, and M' is the structure of the cryptographic algorithm. Then $M'' \leqslant M$, $M'' \leqslant M'$.

This result can be inferred obviously from Definition 1.1, 1.4, 1.7 and 2.1.

Definition 2.6 (**Assumption Functions**). An assumption function Af on a Kripke structure M is defined as follows: Af: $A \rightarrow 2^{S1} \cup \{\perp\}$, where $S_1 \subseteq S$.

$Af(\varphi)$ is the set of all states in S_1 in which the formula φ holds, where $\varphi \in A$.

Under an assumption function Af, satisfaction of a formula φ in a state $s \in S$ is represented as $s \models_{Af} \varphi$, which is equal to $s \in Af(\varphi)$.

If $Af(\varphi)=\perp$, then $s \models_{Af} \neg\varphi$, where $s \in S$, $\varphi \in A$.

Under an assumption function Af, protocol's satisfaction of security requirements is defined as follows:

$\forall \varphi \in A$, $M \models_{Af} \varphi$, which is similar to Definition 1.3.

Theorem 2.7 (**The Assume-Guarantee Based Reasoning Rule**). Suppose that M'' is the structure of a cryptographic protocol system, $M''=M /\!/ M'$, where M is the structure of a cryptographic protocol, and M' is the structure of a cryptographic algorithm. If for every $\forall \varphi' \in A'$, $M' \models \varphi'$, and for every $\forall \varphi \in A$, $M \models_{Af} \varphi$, then for every $\forall \varphi \in A$, $M'' \models \varphi$.

The soundness of this rule can be demonstrated as follows:

Proof. From $\forall \varphi' \in A'$, $M' \models \varphi'$, we have $M' \models A'$.

From Corollary 2.4, we have $M' \preccurlyeq A'$.

From Theorem 2.2, we have $M /\!/ M' \preccurlyeq M /\!/ A'$.

We know that for every $\forall \varphi \in A$, $M \models_{Af} \varphi$, from which we have $M /\!/ A' \models \varphi$.

From Corollary 2.3, we have that for every $\forall \varphi \in A$, $M /\!/ M' \models \varphi$.

Therefore, for every $\forall \varphi \in A$, $M'' \models \varphi$.

Using this assume-guarantee based reasoning rule, the result of $M'' \models A''$ can be inferred in succession, that is, under the conditions of this rule, it can be inferred for the cryptographic protocol system to satisfy all the security requirements A'' defined in the security model (see Definition 1.7). This can be demonstrated as follows:

Proof. From $M' \models A'$ and $M'' \preccurlyeq M'$ (Theorem 2.5),

We have $M'' \models A'$ (Corollary 2.3).

From Theorem 2.7, we have $M'' \models A$.

We know that $A''=A \cup A'$ (Definition 1.7).

Therefore, $M'' \models A''$.

2.3 The assume-guarantee based model checking algorithm

According to the call relation between the protocol and cryptographic algorithms, a cryptographic protocol system P can be decomposed into several components by means of the hierarchy tree as shown in Fig.1.

In Fig.1, P_1 is a protocol process, and P_2, P_3,…,P_n are different cryptographic algorithm processes called by the protocol process at different call points. These call points s_{e1},s_{e2},…, $s_{en} \in S$ denote ending points in the protocol, and form a set $S_{end}=\{s_{e1},s_{e2},\ldots,s_{en}\}$. To simplify specification, here only the protocol with one cryptographic algorithm process is discussed (of course, this assume-guarantee based model checking algorithm is suitable for protocols with several cryptographic algorithm processes), in which there is a call relation r: $S_{end} \rightarrow S'_0$, where S'_0 is the set of initial states in the cryptographic algorithm process.

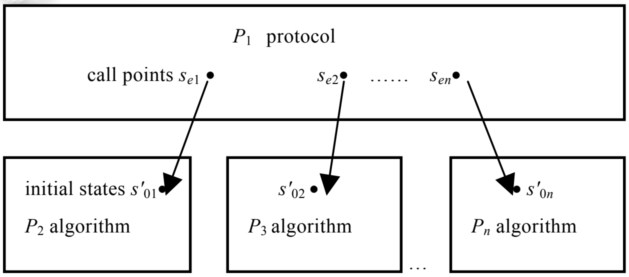

Fig.1 A partition graph for a cryptographic protocol system

According to the partition of the cryptographic protocol system, model check the whole system from bottom components upwards. The processing steps of the assume-guarantee based model checking algorithm is as follows:

Step 1. Model check each underling cryptographic algorithm process under no assumption, using ModelCheck(P_2),…,ModelCheck(P_n). If the cryptographic algorithm process being verified can resist the cryptographic attacks (see Definition 1.5), and satisfy security properties $\varphi'\in A'$, then returns the analysis result *Ret* to its upper caller. The result *Ret* is the set of security properties which the cryptographic algorithm can satisfy, $Ret=\{\varphi'_i \mid (i=1,2,\ldots), \varphi'_i\in A'\}$, which can be used to determine the security strength of the cryptographic algorithm.

Step 2. Represent the analysis result as an assumption function Af' on the cryptographic algorithm, and build the set of initial states S'_{init} which satisfy properties in *Ret*. This process is as follows: Fetch every $\varphi'_i\in Ret$, and find backwards along the searching path (from Definition 1.6) the initial state set S'_i which satisfies properties φ'_i, that is, $Af'(\varphi'_i)=S'_{i.}$ Suppose $S'_{init}=\cup S'_i$, $(i=1,2,\ldots)$, then Af': $Ret\rightarrow 2^{S'_{init}}$.

Step 3. According to Definition 2.8, draw the assumption function *Af* on the cryptographic protocol, which is the assumption function of S_{end} on the protocol.

Definition 2.8. Suppose that *M* is the structure of a cryptographic protocol, *M′* is the structure of a cryptographic algorithm, and a call relation r: $S_{end}\rightarrow S'_{init}$, where $S_{end}\subseteq S$, $S'_{init}\subseteq S_0'$.
Let Af': $Ret\rightarrow 2^{S'_{init}}$ be the assumption function on the cryptographic algorithm, where $Ret\subseteq A'$, and let
Af : $Ret\rightarrow 2^{S_{end}}$ be the assumption function on the cryptographic protocol, $Af(\varphi')$ is defined recursively as follows:

① For every $\forall\varphi'\in Ret$, $Af(\varphi')=\{s\in S_{end} \mid \exists s'_0\in S'_{init}, r(s,s'_0)\wedge\varphi'\in\mathscr{L}'(s'_0)\}$

② If $Af'(\varphi')=\perp$, then $Af(\varphi')=\perp$, where $\varphi'\in A'\setminus Ret$

③ $Af(\varphi'_1\wedge\varphi'_2)=Af(\varphi'_1)\cap Af(\varphi'_2)$, where $\varphi'_1,\varphi'_2\in A'$

④ $Af(\varphi'_1\vee\varphi'_2)=Af(\varphi'_1)\cup Af(\varphi'_2)$

⑤ $Af(\neg\varphi')=S_{end}\backslash Af(\varphi')$, where $\varphi'\in A'$.

Step 4. Under the assumption function *Af*, model check the cryptographic protocol component[10], focusing on the checking of security correspondence between the cryptographic algorithm and the protocol to see whether they have the same security strength. For example, if the cryptographic algorithm cannot resist a chosen-plaintext attack, then it is valueless and blind to verify whether the protocol can resist the chosen-plaintext attack; if the cryptographic algorithm can resist a chosen-plaintext attack, then it is needed to verify whether the protocol can resist the same attack under this assumption.

In conclusion, this new security model and the assume-guarantee based model checking algorithm may have some advantages, such as reducing the blind searching of pure protocol analysis, avoiding state explosion problems, verifying the cryptographic protocol system completely, and ensuring the security of the whole system.

3 An Example

Using the new security model for cryptographic protocol systems and the assume-guarantee based model checking algorithm stated above, the Kerberos v4 and v5 protocol system, as an example, are verified. The Kerberos protocol serves as authentication and key distribution based on the block cipher DES. The Kerberos v5[6] protocol steps are specified as follows:

① $C\rightarrow AS$: C, TGS, L_1, N_1

② $AS\rightarrow C$: $C, T_{c,tgs}, \{TGS, K_{c,tgs}, T_{start}, T_{expire}, N_1\}K_c$

③ $C\rightarrow TGS$: $S, L_2, N_2, T_{c,tgs}, A_{c,tgs}$

④ $TGS\rightarrow C$: $C, T_{c,s}, \{S, K_{c,s}, T'_{start}, T'_{expire}, N_2\}K_{c,tgs}$

⑤ $C\rightarrow S$: $T_{c,s}, A_{c,s}$

⑥ $S\rightarrow C$: $\{T'\}K_{c,s}$

This protocol has four honest principals, they are a client C, an authentication server AS, a ticket granting server TGS, and a server S. In the first step of the protocol, C sends a message to AS to ask for a ticket granting ticket TGT, and L_1 is the valid period of TGT, N_I is a nonce generated by C. In the second step, AS sends TGT to C, in which $T_{c,tgs}=\{C,TGS,K_{c,tgs},T_{start},T_{expire}\}K_{tgs}$, $K_{c,tgs}$ is a session key shared between C and TGS distributed by AS. T_{start} and T_{expire} are the starting and ending time of TGT respectively. K_{tgs} stands for encryption with TGS's symmetric key, and K_c stands for encryption with C's main key. In the third step, C sends a message to TGS to ask for a service granting ticket of server S, and L_2 is the valid period of the ticket, N_2 is a nonce generated by C. $A_{c,tgs}=\{C,T\}K_{c,tgs}$ is a verification code of $T_{c,tgs}$ produced through encrypting C and a time-stamp T with the key $K_{c,tgs}$. In the fourth step, TGS sends the service ticket $T_{c,s}$ to C, in which $T_{c,s}=\{C,S,K_{c,s},T'_{start},T'_{expire},N_2\}K_s$, $K_{c,s}$ is a session key shared between C and S distributed by TGS, T'_{start} and T'_{expire} are the starting and ending time of the ticket respectively, K_s stands for encryption with S's symmetric key. In the fifth step, C sends the ticket $T_{c,s}$ and its verification code $A_{c,s}=\{C,T'\}K_{c,s}$ to S. In the sixth step, S encrypts the time-stamp T' with the key $K_{c,s}$ and sends it to C to achieve mutual authentication.

Decompose the Kerberos v4 protocol system into the cryptographic protocol component and the cryptographic algorithm component. Using the assume-guarantee based model checking algorithm to verify the Kerberos v4 protocol system, first we verify the cryptographic algorithm component which includes the DES algorithm, a propagation CBC mode (PCBC mode), and a password verification algorithm. Model checking DES algorithm, fetching every round's data and calling the cryptographic attack database, we get a result that DES algorithm satisfies complete diffusion after 4 rounds, satisfies the strict avalanche criterion after 8 rounds, and resists the balance test, completeness cryptanalysis, correlation test, non-linear cryptanalysis, the output bit independence criterion, related-key attack, differential cryptanalysis and linear cryptanalysis. Verifying the PCBC mode, we find that there exists a tampering attack to ciphertext blocks, that is, the attacker can modify messages through exchange two adjoining ciphertext blocks in the middle of the message. PCBC provides the integrity checking as a complement of this mode. Since it only checks the integrity of the last block, it cannot resist this attack caused by exchange ciphertext blocks. In message ②, the computation of client's main key K_c depends on client's password pwd, that is, $K_c=h(pwd)$, where h is a one-way hash function. Through intercepting this message, the attacker can do a series of password attacks such as verifiable password attack, password-guessing attack, and dictionary attack. If the attacker succeeds in password attacks, there exists impersonation attack, in which the attacker can impersonate the client C. After verify the cryptographic algorithm component, we get the following result: resistance of the balance test, completeness cryptanalysis, the avalanche effect, correlation test, diffusion cryptanalysis, non-linear cryptanalysis, the output bit independence criterion, related-key attack, differential cryptanalysis and linear cryptanalysis, no resistance of tampering attack and password attack. Under this assumption, model check the protocol component. In the verification of the protocol component, the tampering attack needs not to be analyzed. Since client C can ask for a special service from server S re-using the same ticket $T_{c,s}$, in which C and S's session key $K_{c,s}$ is included. This violates the freshness requirement of session keys, and exists a replay attack in the valid period of the ticket. Using an old session key, in a new session the intruder can impersonate client C (or server S) to send messages of the old session to server S (or client C), which forms an impersonation attack. Meanwhile, using the weakness of clock synchronization, the intruder can replay the verification code during a valid period of the time-stamp. A path of replay attack is as follows:

⑤' $I(C) \rightarrow S$: $T_{c,s}$, $A_{c,s}$

⑥' $S \rightarrow C$: $\{T'\}K_{c,s}$

note: In Kerberos v4, $T_{c,s}=\{C,S,K_{c,s},T',L'\}K_s$, where C stands for the identification of a user, T' is a time-stamp, and L' is the valid period of the ticket. $I(C)$ stands for the intruder I impersonating client C to replay message ⑤.

Now the whole protocol system of Kerberos v4 is verified, and the vulnerabilities are as follows: Exists replay attack, impersonation attack, tampering attack, and password attack.

Verify the Kerberos v5 protocol system. In Kerberos v5, the cryptographic algorithm has been separated to different software modular, the standard CBC mode is used for encryption, and the plaintext is embedded into the checksum, which can resist tampering attack, but still exists password attack. In the protocol logic, there exist the same replay attack and impersonation attack, and the path of the replay attack is as above. If the optional subkey field is used in message ⑤, in which *C* and *S* negotiate a new fresh subkey in each session, then the impersonation attack can be resisted. However, the intruder can still replay the ticket and verification code due to the weakness of clock synchronization. A path of replay attack is as follows:

③' $I(C) \rightarrow TGS$: $S, L_2, N_3, T_{c,tgs}, A_{c,tgs}$

④' $TGS \rightarrow C$: $C, T'_{c,s}, \{S,K'_{c,s},T'_{start},T'_{expire},N_3\}K_{c,tgs}$

note: ticket $T'_{c,s}=\{C,S,K'_{c,s},T'_{start},T'_{expire},N_3\}K_s$, where N_3 is a nonce generated by the intruder *I* impersonating client *C*. This attack violates the requirement that honest principals' records of protocol runs should match[11], and increases the burden of the ticket granting server *TGS*. Lots of such replay attacks may lead to *TGS*'s Denial-of-Service.

It is thus clear that it should depend on two aspects to adapt the Kerberos v5 protocol system, one is to modify the cryptographic algorithm, using secure public-key ciphers replacing the password verification; the other is to modify the protocol, using nonces + time-stamps replacing pure time-stamps.

4 Conclusions

This paper builds the new security model for the cryptographic protocol system composed of a cryptographic protocol and cryptographic algorithms, proposes the new assume-guarantee based reasoning rule and model checking algorithm, and realizes model checking to the cryptographic protocol system. This method is practicable and has some advantages, which provides an efficient tool for formal analyses of cryptographic protocol systems. However, the real problem is complicated. There are complicated hierarch call relations between a protocol and its cryptographic algorithms, and both the parallel relation in concurrent protocol runs and the sequential relation between the protocol and its each cryptographic algorithm process may cause problems more complex. One can see that, the security model for cryptographic protocol systems, stated in this paper, is only a simple model for small systems (such as finite state systems), which still needs improving continuously. Meanwhile, in order to verify the security requirements of cryptographic protocol systems sufficiently, it is needed to collect continuously new cryptographic attacks and protocol attacks and to augment the cryptographic attack database and the security requirement database defined in the security model to verify whether cryptographic protocol systems are secure under the current attacks. All of these are included in the further research.

References:

[1] Dolev D, Yao A. On the security of public key protocols. IEEE Transactions on Information Theory, 1983,29(2):198~208.

[2] Mao W, Boyd C. Towards the formal analysis of security protocols. In: Gray J, ed. Proceedings of the Computer Security Foundations Workshop Ⅵ. Los Alamitos: IEEE Computer Society Press, 1993. 147~158.

[3] Berezin S, Campos S, Clarke EM. Compositional reasoning in model checking. In: Roever W, Langmaack H, Pnueli A, eds. Proceedings of the Workshop COMPOS'97. Berlin Heidelberg: Springer-Verlag, 1997. 81~102.

[4] Amla WN, Emerson EA, Namjoshi K, Trefler R. Assume-Guarantee based compositional reasoning for synchronous timing diagrams. In: Margaria T, Yi W, eds. Tools and Algorithms for the Construction and Analysis of Systems, 7th International Conference, TACAS 2001. Berlin Heidelberg: Springer-Verlag, 2001. 465~479.

[5] Liu YW, Li WQ. Hierarchy requirements and verification for cryptographic protocols. Journal of Beijing University of Aeronautics and Astronautics, 2002,28(5):589~592 (in Chinese with English abstract).

[6] Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. 2nd ed, New York: John Wiley & Sons, Inc., 1996.

[7] Menezes A, Oorschot PC, Vanstone S. Handbook of Applied Cryptography. New York: CRC Press, 1997.

[8] Feng DG, Wu WL. The Design and Analysis for Block Ciphers. Beijing: Tsinghua University Press, 2000 (in Chinese).

[9] Grumberg O, Long D. Model checking and modular verification. ACM Transactions on Programming Languages and Systems, 1994,16(3):843~871.

[10] Liu YW, Li WQ. The model reasoning verifier for cryptographic protocols. In: Wu ZH, Zhuang YT, He QM, *et al*., eds. Proceedings of the 6th International Conference for Young Computer Scientist. Beijing: International Academic Publishers, 2001. 290~295.

[11] Lowe G. A hierarchy of authentication specifications. In: Guttman J, ed. Proceedings of the 10th IEEE Computer Security Foundations Workshop. CA: IEEE Computer Society Press, 1997. 31~43.

附中文参考文献:

[5] 刘怡文,李伟琴.密码协议的分层安全需求及验证.北京航空航天大学学报,2002,28(5):589~592.

[8] 冯登国,吴文玲.分组密码的设计与分析.北京:清华大学出版社,2000.