Special English Edition of Journal of KIISC, Vol. 18 No.6 (B), December 2008

Security Analysis of Cryptographic Protocols Based on Trusted Freshness

(Invited paper)

Kefei Chen, [1] *Ling Dong,* [2] *and Xuejia Lai* [3]
Shanghai Jiao Tong University, 800 Dongchuan RD, Shanghai, China

Abstract

A novel idea of protocol security analysis is presented based on trusted freshness. The idea has been implemented not only by hand but also by a belief multisets formalism for automation. The key of the security analysis based on trusted freshness is a freshness principle: for each participant of a cryptographic protocol, the security of the protocol depends only on the sent or received one-way transformation of a message, which includes a trusted freshness. The manual security analysis method and the belief multisets formalism are all established on the basis of the freshness principle. Security analysis based on trusted freshness can efficiently distinguish whether a message is fresh or not, and the analysis results suggest the correctness of a protocol convincingly or the way to construct attacks intuitively from the absence of security properties. Furthermore, the security analysis based on trusted freshness is independent of the idealization of a protocol, the concrete formalization of attackers' possible behaviors, and the formalization of concurrent runs of protocols.

Key words : cryptographic protocol, formal analysis, trusted freshness

I. Introduction

Cryptographic protocols, also called security protocols, use cryptography in communication protocols to provide confidentiality, authenticity, integrity or nonrepudiation in an insecure network. But, unfortunately, due to subtleties of cryptographic protocols, many protocols have shown to be flawed even a long time after they were published [1-3]. A variety of useful rigorous ways have been developed for analyzing and reasoning about cryptographic protocols and they have been proved useful while designing and analyzing a cryptographic protocol [4-15]. However, after two decades of research, some important issues remain without satisfactory treatment: (1) some formalisms could not efficiently distinguish the freshness of messages, hence they could not defense some replay attacks. For example, Burrows et al. presented the famous BAN logic [5], which states: the formula X has not been sent in a message at any time before the current run of the protocol, then X is fresh. This is subtle, hence there exists an interleaving attack which is also a replay attack on Needham-Schroeder public key protocol (N-S protocol for short) even if the N-S protocol can be proved secure by BAN logic [1-2, 5-6]; (2) Some formalisms are not independent of the formalizations of concurrent protocol runs and also attackers' possible behaviors [7-9].

In this paper, we try to present a mechanism that can improve above issues and to provide an efficient rigorous formalism that is simple and precise for automation of security analysis. A **Freshness Principle** is presented for analyzing the security of a protocol: *for each participant of a cryptographic protocol, the security of the protocol depends only on the sent or received one-way transformation of a message, which includes a trusted freshness.* Based on the freshness principle, we give 4 precision definitions of security goals which guarantee that the security of a cryptographic protocol is adequate, we also give a manual security analysis method, and a belief multisets formalism for automation. Our security analysis based on trusted freshness gives a detailed view of the conditions under which a protocol achieves its security goals. Sufficiency and necessity of the precision definitions of security goals is formally argued and rigorously proved. To analyze a cryptographic protocol, all one needs to do is to simply prove the security of a protocol via the manual security analysis method or the belief multisets formalism, then one obtains the adequacy

Manuscript received September 30,, 2008
Manuscript revised November 5, 2008.
1) Kefei Chen, Shanghai Jiao Tong University, kfchen@sjtu.edu.cn
2) Ling Dong, Shanghai Jiao Tong University, ,ldong@sjtu.edu.cn
3) Xuejia, Shanghai Jiao Tong University, ,laix@sjtu.edu.cn

of the security of a cryptographic protocol in a realistic adversary-controlled network. The efficiency, rigorousness, automation possibility of the manual security analysis method, and the belief multisets formalism are illustrated via the analysis of the Needham-Schroeder public key protocol.

The approach presented here has several advantages. First, the analysis results based on freshness principle can either establish the correctness of protocols when they are in fact correct (as defined here), or find flaws in those protocols that are not correct and suggest the improvements to be made; second, analyzers can not only discover attacks but also construct various attacks directly from the absence of security properties as we have illustrated; third, each participant's beliefs are independent of the idealization of a protocol, the concrete formalization of attackers' possible behaviors; finally, the definitions and the formalism identify simple and precise characteristics of protocols, which can be easily developed not only by hand but also by automation.

II. Preliminaries

1. Basic Notions and Initial Assumptions

In order to explain what is meant by the security of cryptographic protocols, we first review some basic notions and make some assumptions on the primitives [12].

Principals, probabilistic polynomial time machines, they are interconnected by point-to-point links over which messages can be exchanged. *Trusted Third Party* (TTP), a principal that provides a centralized authentication service in an open system. *Protocol*, a communication procedure run between or among co-operative principals. *Message-driven protocols*, protocols are initially triggered at a party by an external "call" and later by the arrival of messages. *Freshness identifier*, a unique freshness component generated for a particular protocol run, it can be a nonce, a timestamp, a session key or a shared part of a session key. *Challenge-Response protocol*, in a challenge-response mechanism, one participant can verify the lively correspondence of the intended opposite partner by inputting a freshness identifier (challenge) to a composition of a protocol message and the composition involves a cryptographic operation (response) performed by the intended opposite partner. *Session*, a copy of a protocol run at a party, several copies of any protocol may be simultaneously run by each party.

Unilateral Entity Authentication: the identity of one protocol participant is authenticated. *Mutual Entity Authentication*: the identities of both protocol participants are authenticated to each other. *Unilateral authenticated key transport*: the identity of one protocol participant is authenticated, and the opposite unauthenticated party believes that the session key generated by the authenticated participant or a TTP can provide a secure channel over an insecure network. *Mutual authenticated key transport*: the identities of both protocol participants are authenticated to each other, and both protocol participants believe that the new session key generated by one of the participants or TTP can provide a secure channel over an insecure network. *Mutual authenticated key exchange (or key agreement)*: the identities of both protocol participants are authenticated to each other, and both protocol participants believe that the new session key which is the output of a function of all protocol participants' random input can provide a secure channel over an insecure network.

Suppose we have a PPT attacker I that has full control of the communication links as described in Dolev-Yao threat model [4]. Besides this, we suppose that the Dolev-Yao attacker I in this paper can also launch the adaptive Chosen Ciphertext Attacks (CCA2) without limitations.

Suppose we have cryptographic primitives with security against Indistinguishable Adaptive Chosen Ciphertext Attack (IND-CCA2). That is, in IND-CCA2 security strenghth, the failures in cryptographic protocols are not in any way related to the strength or weakness of the particular cryptographic primitive used, but related to the protocol logic flaws, which permits the attacker to break the security goals of cryptographic protocols without necessarily breaking the particular cryptographic primitive used. And we also suppose that a legitimate party is either totally corrupted or totally secure.

Suppose that each participant has his own private key and all other parties' public keys (respectively, the shared long term key between co-operative principals or trusted third party) in public key case (respectively, in shared key case), which are deployed safely before the cryptographic protocol run via authenticated channel or even traditional communication means. Furthermore, private keys and shared keys are commonly assumed to be too long to guess in a computationally feasible way.

In general, an authentication protocol is considered flawed if a principal concludes a normal run of the protocol with its intended communication partners while the intended partner would have a different conclusion. We mainly discuss Challenge-Response authentication protocols in this paper.

2. Definition of Security

The security definition under computational model provides a high confidence of the security of a cryptosystem [10-13]. A conversation is a sequence of timely ordered messages that a participant sent out (respectively, received), and as consequent responses,

received (respectively, sent). Let $\tau_1 < \tau_2 < \cdots \tau_n$ be a time sequence recorded by the participant when it converses. The conversation can be denoted by the following sequence: $conv = (\tau_1, m_1, m'_1), (\tau_2, m_2, m'_2), \cdots (\tau_n, m_n, m'_n)$.

This sequence encodes that at time τ_1, the participant was asked m_1 and responded with m'_1; and then, at some later time $\tau_2 > \tau_1$, the participant was asked m_2, and responded with m'_2; and so on, until, finally, at time τ_n it was asked m_n, and responded with m'_n. If m_1="", then the participant is the initiator, otherwise, we call it the responder. If m_n = "no message output", the participant ends the conversation. At the end of a protocol run, each participant makes a decision about the authentication of the intended partner: accept, reject, or undetermined [12]. Suppose there exists a cryptographic protocol run between principal A and B. Let $conv = (\tau_1, "", m_1), (\tau_2, m'_1, m_2), (\tau_3, m'_2, m_3), \cdots, (\tau_{2i}, m'_{i-1}, m_i)$ be a conversation of A. We say that B has a conversation $conv'$ which matches $conv$ if there exists time sequence $\tau_1 < \tau_2 < \cdots \tau_n$ and $conv' = (\tau_1, m_1, m'_1), (\tau_2, m_2, m'_2), (\tau_3, m_3, m'_3), \cdots, (\tau_{2i}, m_i, m'_i)$ where m_t= "no message output". These two conversations are called matching conversations [11,10].

Given a protocol π between principal A and B, if a principal like A (or B) with a conversation $conv$ believes that B (or A) always has a conversation $conv'$ which matches $conv$ whenever they are allowed to complete a protocol run, then this authentication protocol is secure from the point of view of A (or B). Here the attacker wins if principal A or B has reached "accept" decision while A or B does not have a matching conversation in B or A.

Semantic security is widely accepted in the cryptographic area, and we follow the probabilistic indistinguishability definitional approach [13] presented by Goldwasser and Micali to define confidentiality security. In this paper, the attacker has broken the scheme means that: without breaking any cryptographic algorithm and knowing the corresponding key, the attacker can still learn something about the established new session key under the run of a cryptographic protocol. Here we define ``learn" as distinguishing the value of a key generated by the cryptographic protocol from an independent randomly chosen key.

Based on the security definition of authenticity, we present the Unilateral entity Authentication Secure definition (UA-Secure) and Mutual entity Authentication Secure definition (MA-Secure); based on the security definition of authenticity and confidentiality, we present the Unilateral authenticated Key Secure (UK-Secure) and Mutual authenticated Key Secure (MK-Secure) [15].

Definition 1: Unilateral entity Authentication Secure (**UA-Secure**): an authentication protocol π is called UA-Secure from the point of view of A if the attacker cannot win with a non-negligible probability for any attacker I in Dolev-Yao threat model. Here the attacker wins if principal A has reached "accept" decision while A does not have a matching conversation in B.

Definition 2: Mutual entity Authentication Secure (**MA-Secure**): an authentication protocol π is called MA-Secure if the attacker cannot win with a non-negligible probability for any attacker I in Dolev-Yao threat model. Here the attacker wins if any principal A or B has reached "accept" decision while A or B does not have a matching conversation in B or A.

Let k be the value of the corresponding new session key. We toss a coin b, $b \xleftarrow{R} \{0,1\}$. If b = 0, we provide the attacker I with the value k. Otherwise we provide the attacker I with a value r randomly chosen from the probability distribution of keys generated by protocol π. At the end of its run, the attacker I outputs a bit b' (as its guess for b).

Definition 3: Unilateral authenticated Key Secure (**UK-Secure**): an authentication protocol π is called UK-Secure from the point of view of A if the following properties hold for any attacker I in Dolev-Yao threat model:

(1). If uncorrupted party A believes that A has completed a session with the intended opposite party B, then A trusts that the uncorrupted party B must have responded to the same session, and they both output the same key k; and

(2). The probability that the attacker I guesses correctly the bit b (i.e., outputs b' = b) is no more than 1/2 plus a negligible fraction in the security parameter.

Mutual authenticated Key Secure includes mutual authenticated key transport secure (for Mutual authenticated key transport) and mutual authenticated key agreement secure (for Mutual authenticated key agreement). Let k be the value of the corresponding session key. We toss a coin b, $b \xleftarrow{R} \{0,1\}$. If b = 0, we provide the attacker I with the value k. Otherwise we provide the attacker I with a value r randomly chosen from the probability distribution of keys generated by protocol π. At the end of its run, the attacker I outputs a bit b' (as its guess for b).

Definition 4: Mutual authenticated Key Secure (**MK-Secure**) (SK-Secure [15]): an authentication protocol π is called MK-Secure if the following properties hold for any attacker I in Dolev-Yao threat model:

(1). Protocol π satisfies the property that if two uncorrupted parties complete matching sessions then they both output the same key; and

(2). The probability that the attacker I guesses correctly the bit b (i.e., outputs b' = b) is no more than 1/2 plus a negligible fraction in the security parameter.

III. Freshness Principle

A novel idea of protocol security analysis based on trusted freshness will be presented in this section [17]. The presentations include the freshness principle, which is the key of the security analysis based on trusted freshness, and a manual analysis method based on the freshness principle.

1. Notions

We give some notions in our security analysis idea based on trusted freshness.

Definition 5 (Term): A term $\hat{m}$ is a fresh message that may be exchanged in a particular protocol run. A term set $\hat{M}$ is the collection of all terms in a protocol run. Terms can be recursively defined as:

(1). If $\hat{m}$ is a trusted freshness identifier, then $\hat{m}$ is a term;

(2). If $\hat{m}$ is a term, o is a principal identity or a freshness identifier, then $\{\hat{m}, o\}$ or $\{o, \hat{m}\}$ is a term.

(3). If $\hat{m}$ is a term, k is a cryptographic key, then $\{\hat{m}\}_k$ is a term. If $\hat{m}$ is a term, o is a principal identity or a freshness identifier, then $\{o\}_{\hat{m}}$ is a term.

Definition 6 (Signed Term): A signed term is a binary group $(\delta, \hat{m})$, where δ is a sign, $\hat{m} \in \hat{M}$. A signed term is stated as $+\hat{m}$ or $-\hat{m}$. $+\hat{m}$ and $-\hat{m}$ states a sent out or received fresh message respectively.

Freshness: from the point of view of a participant in a protocol run, a freshness identifier or a message is confirmed to be new for a particular run of a protocol. If a freshness identifier or a message is generated by the principal itself for this run or conveyed with this generated new component by a one-way transformation, then the freshness is confirmed.

Trusted freshness: also called trusted freshness identifiers. From the point of view of a participant in a protocol run, the freshness of the freshness identifier has already been authenticated by himself, including trusted nonce, trusted timestamp, trusted session key or trusted shared parts of a session key. Note that the trusted freshness is different for each participant in different protocol run.

Fresh message: from the point of view of a participant in a protocol run, the sent or received one-way transformation that includes a trusted freshness.

Liveness of principal: from the point of view of a participant in a protocol run, the intended opposite participant is in lively correspondence with him in this session.

Confidentiality of a freshness identifier: from the point of view of a participant in a protocol run, the freshness identifier is transmitted in the form of an encryption that cannot be decrypted by the attacker. We note that the signature of a freshness identifier is not confidential.

Freshness of a freshness identifier: from the point of view of a participant in a protocol run, the freshness identifier is new generated for this particular protocol run, not an old one or a compromised one. A principal believes the freshness of the freshness identifier generated by the principal itself.

Association of a freshness identifier: from the point of view of a participant in a protocol run, the freshness identifier is bound to some legitimate participants of this particular protocol run.

2. Freshness Principle

Freshness Principle: *for each participant of a cryptographic protocol, the security of the protocol depends only on the sent or received one-way transformation of a message, which includes a trusted freshness.*

The security goals are the security characteristic at the end of the protocol run. The set of security goals constructs the security properties of a cryptographic protocol to achieve. In our security analysis of a cryptographic protocol based on trusted freshness, the security properties are described by beliefs, which are the beliefs about the security of a cryptographic protocol owned by each participant in a particular protocol run. The beliefs are about the liveness of intended participants, the confidentiality, freshness and association of freshness identifiers.

In practice, a one-way transformation *[M]k* can be realized by a pair (*M*, prf*k*(*M*)) where prf*k* denotes a keyed pseudorandom function (e.g., a message authentication code in cipher-block-chaining mode of operation, CBC-MAC, or a keyed cryptographic hash function, HMAC) for the case of symmetric technique realization, or a digital signature algorithm for the case of asymmetric technique realization. These are practically efficient realizations [12].

Let *A*, *B* be the participants of an authentication protocol.

Lemma 1: The liveness of a principal *B* (or *A*) can be achieved by a participant *A* (or *B*) via a sent or received one-way transformation that includes a trusted freshness identifier of *A* (or *B*), where the one-way transformation can only be accomplished by the principal *B* (or *A*).

Lemma 2: The confidentiality of a freshness identifier can be achieved by a participant *A* (or *B*) if the identifier is transmitted in the form of an encryption that cannot be decrypted by the attacker.

Lemma 3: The freshness of a new freshness identifier can be achieved by a participant *A* (or *B*) via a sent or received one-way transformation that includes a new freshness identifier and another trusted one of *A* (or *B*), where the one-way transformation can only be accomplished by *B* (or *A*).

Lemma 4: The association of a freshness identifier can be achieved by a participant A (or B) via a sent or received one-way transformation that includes a trusted freshness identifier of A (or B), where the one-way transformation can only be accomplished by the principal B (or A), or the identity of the principal is explicitly stated in the one-way transformation.

Note that: only one-way transformation that includes a trusted freshness identifier is considered as an efficient message of a conversation in our security analysis of cryptographic protocols based on trusted freshness. That is to say, we only concern with the fresh messages but omit the message parts that do not contribute to our protocol security properties to be proved.

3. Guarantee of Security Adequacy

Suppose there exists a protocol δ between A and B, the security goal of δ is to authenticate the liveness of a principal entity or establish a new session key to build a secure channel in an insecure network. The new session key k_{ab} can either be generated by any of the authenticated participants or a trusted third party S, or be the output of a function of all protocol participants' random input N_a, N_b.

Table 1: The Guarantee of the Security Adequacy of a Cryptographic Protocol

Security Goals	Security Properties Achieved by A					Security Properties Achieved by B				
	B	S	N_a	N_b	k_{ab}	A	S	N_a	N_b	k_{ab}
UA-secure (Authenticate B)	1[I]									
UA-secure (Authenticate A)						1				
MA-secure	1					1				
UK-secure (Authenticate B)	1				1[II]1[III]AB[IV]					
UK-secure (Authenticate A)						1				11AB
MK-secure (Key Transport)	1				11AB	1				11AB
MK-secure (Key Agreement)	1		11AB	11AB		1		11AB	11AB	

Note:

[I] "1" or "" means that the liveness of the principal is authenticated or unknown respectively.

[II] "?" or "" means that the confidentiality of the freshness identifier is unknown; "1" or "0" means that the confidentiality of the freshness identifier is confidential, or open respectively.

[III] "?" or "" means that the freshness of the freshness identifier is unknown; "1" means that the freshness identifier is fresh; "0" means that the freshness identifier is a compromised or an old one.

[IV] "#" or "" means that the freshness identifier is not associated with any principals; "11A" (or "11B") means that the freshness identifier is associated with principal A (or B); "11AB" or "11BA " means that the freshness identifier is associated with principal A and B.

Table 1 lists the security properties that are adequate to guarantee the security goals of a cryptographic protocol, and the listed security goals are not only necessary but also substantial. For lack of space, we only prove the adequacy of the listed MK-secure security properties of a protocol.

Theorem 1: An authentication protocol π is called MK-Secure if and only if each participant like A (or B) believes the liveness of the intended opposite principal B (or A), and believes the confidentiality, the freshness and also the association of the new session key k with the principal A and B [16].

Proof: substantial proof. We show that Definition 4(1) is satisfied by protocol π. Since principal like A believes the liveness of the intended opposite principal B and only one-way transformation that includes a trusted freshness identifier is considered as an efficient message of a conversation in our approach, according to Lemma 1, A must have generated a challenge N_a for this particular protocol run, and received a one-way transformation that includes a trusted freshness identifier N_a, where the one-way transformation can only be accomplished by the principal B. Recall that we have an ideal cryptographic algorithm with security against IND-CCA2, if A sees the conversation $conv_A=(\tau_1,\cdots,N_a),(\tau_2,\{N_a\}_{k_{ab}},\cdots)$ in shared key case or $conv_A=(\tau_1,\cdots,N_a),(\tau_2,\{N_a\}_{k_b^{-1}},\cdots)$ in public key case, then A sees that the uniformly random string $\{N_a\}_{k_{ab}}$ or $\{N_a\}_{k_b^{-1}}$ is computed using N_a invented by itself; it can therefore conclude that the probability for this bit string not having been computed by its intended partner (in other words, having been computed by the attacker) is at the level of 2^{-k}. Consequently, A can conclude that its intended partner has a conversation which is prefixed by $conv_B=(\tau_1,N_a,\{N_a\}_{k_{ab}})$ or $conv_B=(\tau_1,N_a,\{N_a\}_{k_b^{-1}})$. This essentially shows that there exists a conversation $conv_B$ matching $conv_A$ and the conversation $conv_B$ has been computed by the intended partner in an overwhelming probability(in k_{ab} or k_b^{-1}). According to the security definition of authentication, A believes that B is in lively correspondence with A in this session. Similarly, B believes that A is in lively correspondence with B in this session.

Since principal A believes the freshness of the new session key k, according to Lemma 3, k must be a new generated session key for this run. Since principal A believes the association of the new session key k with the principal A and B, according to Lemma 4, k must be a

session key for a particular protocol run between A and B. Up to now, k must be a new generated session key for a particular protocol run between A and B, hence k is the same for both A and B, and it is different from other generated key in any other sessions. That is to say, if two uncorrupted parties complete matching sessions then they both output the same key.

We show that Definition 4(2) is also satisfied by protocol π. Recall that the Dolev-Yao attacker I is a PPT machine who has full control of the communication links and is allowed to perform a kind of cryptanalysis training course. We can specify l be an upper bound of the number of sessions invoked by I in any interactions. Let k be the value of the corresponding session key selected randomly in this protocol π. Let A play the game with the attacker I in Definition 4. Let *Bad* be the events that the information of k may be leaked during the cryptanalysis training courses. Let I_{wins} denote the event that I makes a correct guess of the challenge bit b. It is clear that in absence of the event *Bad*, due to the uniform randomness of the selected session key k, the challenge bit b is independent from the challenge ciphertext b'. Thus we have:

$$\text{Prob}[I_{wins} \mid \overline{\text{Bad}}] = \frac{1}{2}$$

Since

$$\text{Prob}[I_{wins} \mid \overline{\text{Bad}}] = \frac{\text{Prob}[I_{wins} \cap \overline{\text{Bad}}]}{\text{Prob}[\overline{\text{Bad}}]},$$

then we have

$$\text{Prob}[I_{wins} \cap \overline{\text{Bad}}] = \frac{1}{2}\text{Prob}[\overline{\text{Bad}}] = \frac{1}{2}(1 - \text{Prob}[\text{Bad}])$$

While

$$\text{Prob}[I_{wins}] = \text{Prob}[I_{wins} \cap \text{Bad}] + \text{Prob}[I_{wins} \cap \overline{\text{Bad}}]$$

therefore

$$\text{Prob}[I_{wins}] \le \text{Prob}[\text{Bad}] + \text{Prob}[I_{wins} \cap \overline{\text{Bad}}]$$

$$= \text{Prob}[\text{Bad}] + \frac{1}{2}(1 - \text{Prob}[\text{Bad}]) = \frac{1}{2}(1 + \text{Prob}[\text{Bad}])$$

Since we have an ideal cryptosystem, then even the attacker I has invoked l times cryptanalysis training course, the probability that the information of k may be leaked to I by the underlying cryptosystem (say *Bad*1) is negligible in the security parameter, that is: $\text{Prob}[Bad1] \le \ell * Adv$ where Adv is a negligible fraction. Since principal A believes the liveness of the intended partner B, then the probability that the information of k might be leaked by B to I (say *Bad*2) is 0. Since principal A believes the association of the new session key k with the principal A and B, then the probability that the attacker I could persuade B to believe a key between I and B (or A) to be the key k between A and B (say *Bad*3) is 0. Then we:

$$\text{Prob}[\text{Bad}] = \text{Prob}[\text{Bad1}] + \text{Prob}[\text{Bad2}] + \text{Prob}[\text{Bad3}]$$
$$= Adv * \ell$$

therefore

$$\text{Prob}[I_{wins}] \le \frac{1}{2}(1 + \text{Prob}[\text{Bad}]) = \frac{1}{2}(1 + Adv * \ell) = \frac{1}{2} + \frac{Adv * \ell}{2}$$

Since attacker I is a PPT attacker, then $(\ell * Adv)/2$ is negligible. Therefore, the probability that I guesses correctly the bit b is no more than 1/2 plus a negligible fraction in the security parameter.

Necessary proof. Suppose a MK-secure protocol π doesn't hold the listed security properties.

If a participant like A (or B) does not believe the liveness of the intended opposite principal B (or A), then, according to Lemma 1, A (or B) has either sent a compromised or an old challenge to the intended opposite partner B (or A) (That is to say, the attacker can replay a recorded stale message to A (or B) by impersonating B (or A)), or does not require a response to A (or B)'s challenge (That is to say, the attacker can launch an attack directly). The typical examples are Otway-Rees protocol [18], Woo-Lam-Abadi protocol [3]. Hence the protocol π cannot be MK-secure.

If a participant like A (or B) believes that the confidentiality of the new session key k is open, then, according to Lemma 2, the attacker wins with the probability of 1 when playing the game in Definition 4. Hence the protocol π cannot be MK-secure.

If a participant like A (or B) does not believe the freshness of the session key k, then, according to Lemma 3, the attacker can replay a recorded message including a compromised key k' to response to A (or B). A typical example is the Needham-Schroeder shared key protocol [19]. Hence the protocol π cannot be MK-secure.

If a participant like A (or B) does not believe the association of the session key k with the co-operative participants, then, according to Lemma 4, the attacker can cheat a legitimate participant by confusing a key between the attacker and another to be the key between two legitimate participants. A typical example is the Needham-Schroeder public key protocol [2]. Hence the protocol π cannot be MK-secure.

Therefore, we can conclude that the listed MK-secure security properties is not only substantial but also necessary for the protocol π to be MK-secure. □

4. Manual Analysis Based on Trusted Freshness

Based on the freshness principle and the accurately presented security goals, we present an analysis method based on trusted freshness, which can be accomplished easily even by hand. The manual freshness analysis method is refined as follows for the same authentication protocol δ in section III.3:

A. Specify the security goals to be reached based on table 1.

B. Specify the premise before the start of the protocol.

Recall that each participant has his own private key and

all other parties' public keys (respectively, the shared long term key between co-operative principals or trusted third party) in public key case (respectively, in shared key case).

(1) Public key case: A knows: K_a^{-1}, K_a, K_b; B knows: K_b^{-1}, K_b, K_a; the attacker I knows: K_a, K_b.

(2) Shared key case without trusted third party: A knows: K_{ab}; B knows: K_{ab}.

(3) Shared key case with trusted third party: A knows: K_{as}; B knows: K_{bs}.

C. From the point of view of each legitimate participant, establish the security properties of a cryptographic protocol based on the freshness principle and Lemma 1,2,3,4 while sending or receiving a one-way transformation that includes a trusted freshness.

D. Compare with the security goals established in step 1. The analysis results can either establish the correctness of the protocol convincingly when they are in fact correct, or identify the absence of the security properties and the structure to construct attacks based on the absence.

5. Application of the Manual Freshness Analysis

Example 1: Needham-Schroeder public key protocol (say, N-S protocol) [1] is a well known cryptographic protocol, whose intended goal is to establish secure communication (MK-secure) between two principals Alice (say, A) and Bob (say, B) via the shared parts N_a and N_b. Fig. 1 is the analysis procedure of the N-S protocol based on the freshness analysis method.

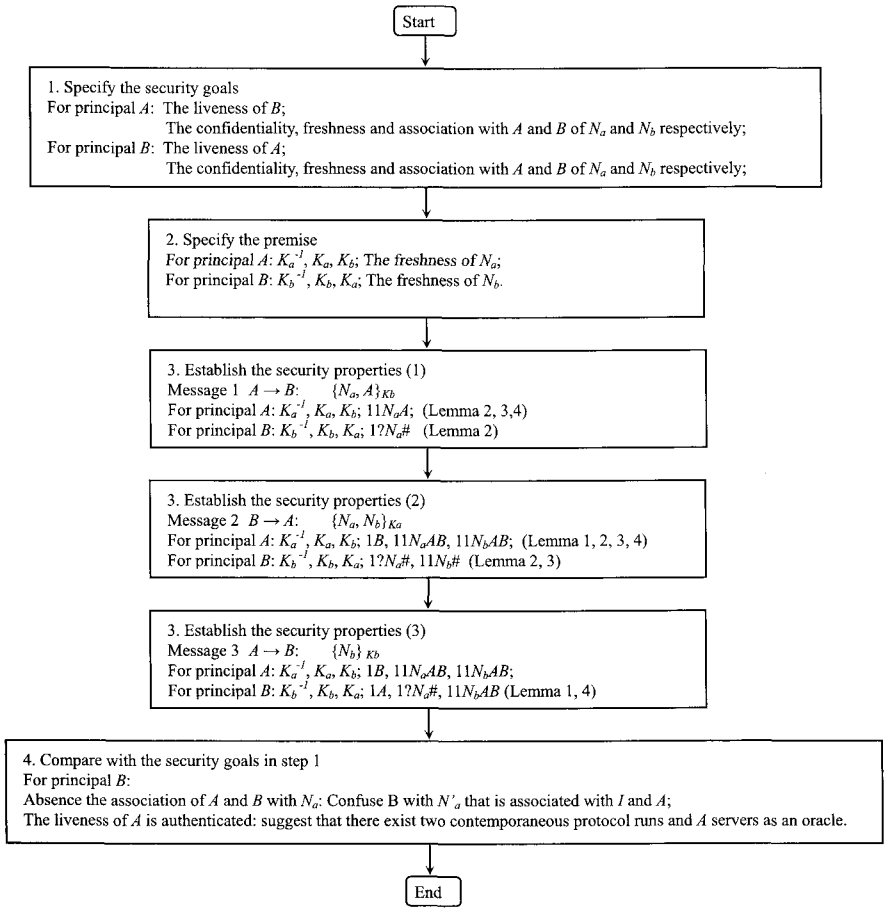

Fig. 1 Security analysis of the N-S protocol based on trusted freshness

IV. Belief Multisets Formalism

We have presented a manual analysis method based on trusted freshness in subsection III.4 and we will introduce a belief multisets formalism [16] for automation in this section to show the efficiency and the rigorous of the security analysis idea based on the freshness principle. Each legitimate participant has a belief multiset independently. $b_{\rho,t}$ means a belief multiset possessed by a principal ρ at time t in the form of $\lfloor <\cdots\cdots N\cdots>,\cdots<\cdots\rho> \rfloor$.

1. Belief Logic Language

ρ	Arbitrary principals, range over participants of the protocol.
p_i or p_j	A principal indexed by subscript in a protocol run.
S	Trusted third party.
t	Arbitrary time, a moment not a period of time.
$t_1, t_2, \ldots t_s$	Time before the start of a run, ... at the termination of a run respectively.
N or N'	Arbitrary freshness identifiers, including nonce, timestamp, session key or shared part of a session key.
N_{pi}	A freshness identifier generated by subscript principal p_i.
k	A cryptographic key; k^{-1}, the inverse of k. In shared-key case, k and k^{-1} are equal.
k_{ij}	The long term key shared by principal p_i and p_j in shared-key case.
k_i and k_i^{-1}	The private key and public key subscripted by principal identity respectively in public key case, e.g. the key pairs of principal p_i.
$\Rightarrow\{\ldots N,N'\ldots\}_k$	A fragment $\Rightarrow\{\ldots N,N'\ldots\}_k$ owned by p_i who asserts the binding of N' with N.
$\prec\{N,p_j\}$ or $\prec\{N,p_i,p_j\}$	An expectation. Only the participant p_j can obtain the freshness identifier N from a one-way transformation. N is associated with a session of p_i in $\prec\{N,p_i,p_j\}$.
key (ρ, k)	Principal ρ has the knowledge of k.
$<\ldots\rho>$	A belief about the liveness of a principal ρ, default $<\ldots\rho>$. Statement $<1\rho>$ asserts that the intended partner ρ is in lively correspondence in this session; $<\ldots\rho>$, the contrary.
$<\ldots_1\ldots_2 N\ldots_3>$	A belief about the freshness identifier N, default $<\ldots\ldots N\ldots>$. "$\ldots_1$", confidentiality property; "$\ldots_2$", fresh property; "$\ldots_3$", association property. Statement $<1\ldots N\ldots>$ asserts that N is confidential in this run; $<0\ldots N\ldots>$ asserts the contrary. Statement $<\ldots 1N\ldots>$ asserts that N is fresh in this run or N is a long-term key; $<\ldots\ldots N\ldots>$ asserts the contrary. Statement $<\ldots\ldots Np_i>$ (or $<\ldots\ldots N\,p_ip_j>$) asserts that N is associated with a participant p_i (or with both p_i and p_j) of a run. If N is a key in a belief like $<\ldots\ldots N\ldots_3>$, then only the principals indicated in "$\ldots_3$" know N. $<01Np_i>$,$<11N\ldots>$,$<11N\,p_ip_j>$ and so on could be a conjunction of above statements.
$B_{\rho,t}\varphi$	The principal ρ believes that the statement Γ holds at time t. $B_{pi,t2}(<11k_{ij}p_ip_j>)$ means: at time t_2, principal p_i believes that the statement $<11k_{ij}p_ip_j>$ holds. That is, p_i believes that k_{ij} is confidential, k_{ij} is fresh, and only principal p_i and p_j know this key k_{ij}.

2. Rules and Axioms

Suppose φ and ψ are formula, formula in belief multisets formalism obey following inference rules:

R1: From $\vdash\varphi$ and $\vdash(\varphi\rightarrow\psi)$, infer $\vdash\psi$

R2: From $\vdash\varphi$ infer $B_{p,t}\varphi$

$\vdash\varphi$ means that formula φ is valid at all time. For example, φ can be a theorem that is derivable from axioms alone. R1 is the modus ponens and states that if φ can be deduced and $\varphi\rightarrow\psi$ can be deduced, then ψ can also be deduced. R2 is the generalization rule, which states that if φ is a theorem then principal ρ believes that statement φ is true at time t.

If A or B is a belief multiset respectively, and o is an element in a belief multiset which can be a principal or a freshness identifier, then the union, intersection, and sum of A and B, denoted by $m_{A\cup B}(o)$, $m_{A\cap B}(o)$, and $m_{A+B}(o)$ respectively, are defined as follows:

R3 : $m_{A\cup B}(o)=m_A(o)\bigcup m_B(o)$

R4 : $m_{A\cap B}(o)=m_A(o)\bigcap m_B(o)$

R5 : $m_{A+B}(o)=m_A(o)+m_B(o)$

Axioms are statements of truth that apply to a logic theory. In the case of the logic theory being developed here, axioms are general statements of classical propositional calculus, and statements of the hypotheses underlying shared-key and public-key cryptographic communication protocols. The axioms given below reflect the underlying assumptions of the logic, and most of them are intuitively clear, so we just give a brief explanation here.

A0 (Generation Rule)

$+\{\ldots N_{pi}\ldots\}\rightarrow B_{p_i,t}(<\cdots 1N_{p_i}\cdots>)$

The generation axiom A0 states: since the principal p_i has generated this freshness identifier N_{pi}, then p_i believes the freshness of the freshness identifier N_{pi}, that is $B_{p_i,t}(<\cdots 1N_{p_i}\cdots>)$.

Manuscript received September 30,, 2008
Manuscript revised November 5, 2008.
1) Kefei Chen, Shanghai Jiao Tong University, kfchen@sjtu.edu.cn
2) Ling Dong, Shanghai Jiao Tong University, ,ldong@sjtu.edu.cn
3) Xuejia, Shanghai Jiao Tong University, ,laix@sjtu.edu.cn

A1 (Fragment Rule)

(a) $-\{\ldots N, N'\ldots\}_{k_{ij}} \wedge B_{p_i,t}(<11k_{ij}p_ip_j>) \wedge B_{p_i,t}(<\ldots 1N\ldots>)$
$\rightarrow B_{p_i,t}(\Rightarrow \{\ldots N, N'\ldots\}_{k_{ij}})$

(b) $-\{\ldots N, N', p_j\ldots\}_{k_{ij}} \wedge B_{p_i,t}(<11k_{ij}p_iS>) \wedge B_{p_i,t}(<\ldots 1N\ldots>)$
$\rightarrow B_{p_i,t}(\Rightarrow \{\ldots N, N'\ldots\}_{k_{ij}})$

(c) $-\{\ldots N, N'\ldots\}_k \wedge B_{p_i,t}(<01kp>) \wedge B_{p_i,t}(<11k_j^{-1}p_i>) \wedge B_{p_i,t}(<\ldots 1N>)$
$\rightarrow B_{p_i,t}(\Rightarrow \{\ldots N, N'\ldots\}_k)$

(d) $-\{\ldots N, N'\ldots\}_{k'} \wedge B_{p_i,t}(<01kp>) \wedge B_{p_i,t}(<11k_j^{-1}p_j>) \wedge B_{p_i,t}(<\ldots 1Np_i\ldots>)$
$\rightarrow B_{p_i,t}(\Rightarrow \{\ldots N, N'\ldots\}_{k'})$

(e) $+\{\ldots N, N'\ldots\}_k \wedge B_{p_i,t}(<11k_{ij}p_ip_j>) \wedge B_{p_i,t}(<\ldots 1N\ldots>)$
$\rightarrow B_{p_i,t}(\Rightarrow \{\ldots N, N'\ldots\}_k)$

(f) $+\{\ldots N, N', p_j\ldots\}_{k_{ij}} \wedge B_{p_i,t}(<11k_{ij}p_iS>) \wedge B_{p_i,t}(<\ldots 1N\ldots>)$.)
$\rightarrow B_{p_i,t}(\Rightarrow \{\ldots N, N'\ldots\}_{k_{ij}})$

(g) $+\{\ldots N, N'\ldots\}_k \wedge B_{p_i,t}(<01kp>) \wedge B_{p_i,t}(<11k_j^{-1}p_j>) \wedge B_{p_i,t}(<\ldots 1Np_i\ldots>)$
$\rightarrow B_{p_i,t}(\Rightarrow \{\ldots N, N'\ldots\}_k)$

(h) $+\{\ldots N, N'\ldots\}_{k'} \wedge B_{p_i,t}(<01kp>) \wedge B_{p_i,t}(<11k_i^{-1}p_i>)$
$\wedge B_{p_i,t}(<\ldots 1N\ldots>) \rightarrow B_{p_i,t}(\Rightarrow \{\ldots N, N'\ldots\}_{k^{-1}})$

The fragment axioms A1(a) to A1(h) relate to the fragment hold by the principal p_i who asserts that the new freshness identifier N' is bound to the trusted freshness identifier N via a one-way transformation including N' and N.

The fragment axiom A1(a) states: the principal p_i received a term $-\{\ldots N, N'\ldots\}_{k_{ij}}$ including trusted freshness N, since p_i believes that k_{ij} is the shared long term key between p_i and p_j, hence p_i asserts that it must be p_j who has sent this one-way transformation. Therefore, p_i believes that N' is generated for the same protocol run , between p_i and p_j, as N. That is, p_i asserts that N' and N are bound to the same protocol run.

The meaning of the fragment axioms A1(b) to A1(h) are intuitively clear and similar to fragment axiom A1(a), so we omit the detail explanation for lack of space.

A2 (Expectation Rule)

(a) $+\{\ldots N\ldots\}_{k_{ij}} \wedge B_{p_i,t}(<11k_{ij}p_ip_j>) \wedge B_{p_i,t}(<11N\ldots>)$
$\rightarrow B_{p_i,t}(\prec\{N, p_i, p_j\})$

(b) $+\{\ldots N\ldots\}_{k_j} \wedge B_{p_i,t}(<11k_j^{-1}p_j>) \wedge B_{p_i,t}(<11N\ldots>) \rightarrow B_{p_i,t}(\prec\{N, p_j\})$

(c) $+\{\ldots p_i, N\ldots\}_{k_j} \wedge B_{p_i,t}(<11k_j^{-1}p_i>) \wedge B_{p_i,t}(<11N\ldots>) \rightarrow B_{p_i,t}(\prec\{N, p_i, p_j\})$

The expectation axioms A2(a) -A2(c) refer to the expectation held by p_i who expects that only the intended opposite partner p_j with the relevant key can obtain the freshness identifier N.

The expectation axiom A2(a) and A2(c) states: the principal p_i has sent a term $+\{\ldots N\ldots\}_{k_{ij}}$ or $+\{\ldots p_i, N\ldots\}_{k_j}$ including the trusted freshness N, since p_i believes that only the intended opposite partner p_j with the corresponding decryption key can obtain the freshness identifier N from the encryption, where N is related to a session of p_i. The explanation of the expectation axiom A2(b) is omitted.

A3 (Confidentiality Rule)

(a) $-\{\ldots m\ldots\}_k \wedge B_{p_i,t}(\neg key(I,k^{-1})) \wedge B_{p_i,t}(<11k^{-1}\ldots>) \rightarrow B_{p_i,t}(<1\ldots m\ldots>)$

(b) $+\{\ldots m\ldots\}_k \wedge B_{p_i,t}(\neg key(I,k^{-1})) \wedge B_{p_i,t}(<11k^{-1}\ldots>) \rightarrow B_{p_i,t}(<1\ldots m\ldots>)$

(c) $-\{\ldots m\ldots\}_k \wedge B_{p_i,t}(key(I,k^{-1})) \rightarrow B_{p_i,t}(<0\ldots m\ldots>)$

(d) $+\{\ldots m\ldots\}_k \wedge B_{p_i,t}(key(I,k^{-1})) \rightarrow B_{p_i,t}(<0\ldots m\ldots>)$

The confidential axioms A3(a) to A3(d) state: if the principal p_i believes that the key k^{-1} is not a promised one (promised means without confidentiality) or an old one(old means without freshness) or known by the attacker, then the message m is confidential, otherwise it is open.

A4 (Liveness Rule)

(a) $-\{\ldots N\ldots\}_{k_{ij}} \wedge B_{p_i,t}(<11k_{ij}p_ip_j>) \wedge B_{p_i,t}(<\ldots 1N\ldots>) \rightarrow B_{p_i,t}(<1p_j>)$

(b) $-\{\ldots N\ldots\}_{k^{-1}} \wedge B_{p_i,t}(<01kp>) \wedge B_{p_i,t}(<11k_j^{-1}p_j>)$
$\wedge B_{p_i,t}(<\ldots 1N\ldots>) \rightarrow B_{p_i,t}(<1p_j>)$

The liveness axioms A4(a) and A4(b) refer to the liveness beliefs held by the principal p_i. That is, p_i believes that the intended opposite partner p_j is in lively correspondence with p_i in this session.

The liveness axiom A4(a) states: the principal p_i received a term $-\{\ldots N\ldots\}_{k_{ij}}$ including the trusted freshness identifier N, since p_i believes that k_{ij} is the shared long term key between p_i and p_j, hence p_i asserts that it must be p_j who has encrypted the message including the trusted freshness identifier N using the key k_{ij}. Therefore, p_i believes that p_j is in lively correspondence with p_i in this session. The explanation of the liveness axiom A4(b) is omitted.

A5 (Association Rule)

(a) $\pm\{\ldots N\ldots\}_{k_{ij}} \wedge B_{p_i,t}(<11k_{ij}p_ip_j>) \wedge B_{p_i,t}(<\ldots 1N\ldots>)$
$\rightarrow B_{p_i,t}(<\ldots 1Np_ip_j>)$

(b) $\pm\{\ldots N, p_j\ldots\}_{k_{ij}} \wedge B_{p_i,t}(<11k_{ij}p_iS>) \wedge B_{p_i,t}(<\ldots 1N\ldots>)$
$\rightarrow B_{p_i,t}(<\ldots 1Np_ip_j>)$

(c) $-\{\ldots N\ldots\}_{k^{-1}} \wedge B_{p_i,t}(<01kp>) \wedge B_{p_i,t}(<11k_j^{-1}p_j>) \wedge B_{p_i,t}(<\ldots 1N\ldots>)$
$\rightarrow B_{p_i,t}(<\ldots 1Np_j>)$

(d) $-\{\ldots N, p_i\ldots\}_{k^{-1}} \wedge B_{p_i,t}(<01kp>) \wedge B_{p_i,t}(<11k_j^{-1}p_j>) \wedge B_{p_i,t}(<\ldots 1N\ldots>)$
$\rightarrow B_{p_i,t}(<\ldots 1Np_ip_j>)$

(e) $-\{\ldots N\ldots\}_k \wedge B_{p_i,t}(<01kp>) \wedge B_{p_i,t}(<11k_i^{-1}p_i>) \wedge B_{p_i,t}(<\ldots 1N\ldots>)$
$\rightarrow B_{p_i,t}(<\ldots 1Np_i>)$

(f) $+\{\ldots p_i, N\ldots\}_{k_j} \wedge B_{p_i,t}(<01kp>) \wedge B_{p_i,t}(<11k_j^{-1}p_j>) \wedge B_{p_i,t}(<\ldots 1N\ldots>)$
$\rightarrow B_{p_i,t}(<\ldots 1Np_i>)$

The association axioms A5(a) to A5(f) refer to the association beliefs held by the principal p_i. That is, p_i believes that the freshness identifier N is fresh (not an old or compromised one), and associated with the protocol participant p_i and p_j, hence N is associated with a

particular protocol run.

The association axiom A5(a) states: the principal p_i receives a term $-\{...N...\}_{k_{ij}}$ including a trusted freshness identifier N, since p_i believes that k_{ij} is the shared long term key between p_i and p_j, hence p_i asserts that it must be p_j who has encrypted the message including the trusted freshness identifier N using the key k_{ij}. Therefore, p_i believes that this session related to N is associated with the protocol run between p_i and p_j. The explanations of the association axioms A5(b) to A5(f) are omitted.

A6 (Exp-deduction Rule)

(a) $B_{pi,t}(\prec\{N,p_j\}) \wedge -\{...N...\}_k \wedge B_{p_i,t}(key(p_i,k^{-1})) \wedge B_{p_i,t}(<...1N...>) \rightarrow B_{p_i,t}(<1p_j>)$

(b) $B_{p_i,t}(\prec\{N,p_i,p_j\}) \wedge -\{...N...\}_k \wedge B_{p_i,t}(key(p_i,k^{-1})) \wedge B_{p_i,t}(<...1N...>) \rightarrow B_{p_i,t}(<1p_j>)$

(c) $B_{pi,t}(\prec\{N,p_i,p_j\}) \wedge -\{...N...\}_k \wedge B_{p_i,t}(key(p_i,k^{-1})) \wedge B_{p_i,t}(<...1N...>) \rightarrow B_{p_i,t}(<...1Np_j>)$

The axiom A6(a) to A6(b) state: the principal p_i has the expectation $B_{pi,t}(\prec\{N,p_j\})$ or $B_{pi,t}(\prec\{N,p_i,p_j\})$ which implies that only the intended opposite participant p_j can read the freshness identifier N . If p_i receives the term $-\{...N...\}_k$ including the trusted freshness identifier N, then p_i can assert that it must be p_j who has encrypted the message including the trusted freshness identifier N. Hence, p_i believes that p_j is in lively correspondence with p_i in this session.

The axiom A6(c) states: the principal p_i has the expectation $B_{pi,t}(\prec\{N,p_i,p_j\})$ which implies that only the intended opposite participant p_j can read the freshness identifier N. If p_i receives the term $-\{...N...\}_k$ including the trusted freshness identifier N, then p_i can assert that N is related to a particular protocol run between p_j and another principal. Therefore, p_i believes that N is associated with p_j.

A7 (Frag-deduction Rule)

$B_{p_i,t}(\Rightarrow\{...N,N'...\}_k) \wedge B_{p_i,t}(<...1Np_ip_j>) \rightarrow B_{p_i,t}(<...1N'p_ip_j>)$

The axiom A7 states: the principal p_i believes the freshness identifier N' is bound to a trusted freshness identifier N, since N is fresh in a particular run between p_i and p_j, then N' is also fresh in the same particular run between p_i and p_j.

3. Stepwise Analysis Based on Belief Multisets

A. Specify the security goals to be reached based on table 1 and the belief multisets

(1) Unilateral entity Authentication Secure (**UA-Secure**)

Authenticate the liveness of p_j for p_i: $b_{p_i,t_a} = \lfloor <1p_j> \rfloor$ or

Authenticate the liveness of p_i for p_j: $b_{p_j,t_a} = \lfloor <1p_i> \rfloor$.

(2) Mutual entity Authentication Secure (**MA-Secure**)

p_i and p_j authenticate each other $b_{p_i,t_a} = \lfloor <1p_j> \rfloor$ and $b_{p_j,t_a} = \lfloor <1p_i> \rfloor$

(3) Unilateral authenticated Key Secure (**UK-Secure**)

Authenticate p_j for p_i where the new session key k is generated by p_j: $b_{p_i,t_a} = \lfloor <11kp_ip_j><1p_j> \rfloor$or

Authenticate p_i for p_j where the new session key k is generated by p_i: $b_{p_j,t_a} = \lfloor <11kp_ip_j><1p_i> \rfloor$

(4) Mutual authenticated Key Secure (**MK-Secure**) (for authenticated key transport protocol)

p_i and p_j authenticate each other; and the new session key k is generated by one of the principals or the trusted third party: $b_{p_i,t_a} = \lfloor <11kp_ip_j><1p_j> \rfloor$and $b_{p_j,t_a} = \lfloor <11kp_ip_j><1p_i> \rfloor$.

(5) Mutual authenticated Key Secure (**MK-Secure**) (for authenticated key agreement protocol)

p_i and p_j authenticate each other; N_{pi} and N_{pj} are the random inputs of a function to produce the new session key: $b_{p_i,t_a} = \lfloor <11N_{p_i}p_ip_j>,<11N_{p_j}p_ip_j><1p_j> \rfloor$and $b_{p_j,t_a} = \lfloor <11N_{p_i}p_ip_j>,<11N_{p_j}p_ip_j><1p_i> \rfloor$.

B. Specify the premise before the start of the protocol

(1) Public key case

$B_{pi,t0}(<11ki^{-1}p_i>,<11kj^{-1}p_j>,<01kip>,<01kjp>)$ and

$B_{p_i,t_a}(<11k_j^{-1}p_j>,<11k_i^{-1}p_i>,<01k_ip>,<01k_jp>)$

(2) Shared key case without trusted third party

$B_{p_i,t_0}(<11k_{ij}p_ip_j>)$ and $B_{p_j,t_0}(<11k_{ij}p_ip_j>)$

(3) Shared key case with trusted third party

$B_{p_i,t_0}(<11k_{is}p_iS>)$ and $B_{p_j,t_0}(<11k_{js}p_jS>)$

C. Establish the security properties of a cryptographic protocol based on the belief multisets while sending or receiving a one-way transformation that includes a trusted freshness.

D. Compare with the security goals established in step A.

4. Application of the Belief Multisets Formalism

4.1 Analysis of the Original N-S public key protocol

We illustrate the efficiency of the belief multisets via the same N-S protocol as in subsection III.5:

A. Specify the security goals

$b_{A,t_a} = \lfloor <11N_aAB>,<11N_bAB>,<1B> \rfloor$ and

$b_{B,t_a} = \lfloor <11N_aAB>,<11N_bAB>,<1A> \rfloor$

B. Specify the premise before the start of the protocol

$b_{A,t_0} = \lfloor <11k_a^{-1}A>,<11k_b^{-1}B>,<01k_ap>,<01k_ap> \rfloor$ and

$b_{B,t_0} = \lfloor <11k_b^{-1}B>,<11k_a^{-1}A>,<01k_ap>,<01k_bp> \rfloor$

C. Establish the security properties of N-S protocol

Step I. Upon receiving Message 1 $A \rightarrow B$: $\{N_a, A\}_{Kb}$

No.	Security properties	Term	Applied
(1)	$+\{\ldots N_a\ldots\} \rightarrow B_{A,t1}(<\cdots 1N_a\cdots>)$	$+\{\ldots N_a\ldots\}$	Axiom A0
(2)	$B_{A,t0}(<1 1k_b^{-1} B>) \rightarrow B_{A,t0}(<1 1k_b^{-1}\ldots>)$		Definition of $<\ldots_1\ldots_2 N\ldots_3>$
(3)	$B_{A,t0}(<1 1k_b^{-1} B>) \rightarrow B_{A,t0}(\neg key(I,k_b^{-1}))$		Def. of $<\ldots_1\ldots_2 N\ldots_3>$
(4)	$+\{\ldots N_a\ldots\}_{kb} \wedge B_{A,t0}(\neg key(I,k_b^{-1})) \wedge B_{A,t0}(<1 1k_b^{-1}\ldots>) \rightarrow B_{A,t1}(<1\ldots N_a\ldots>)$	$+\{\ldots N_a\ldots\}_{kb}$	(2),(3),A3(b)
(5)	$+\{\ldots A,N_a\ldots\}_{kb} \wedge B_{A,t0}(<0 1k_b p>) \wedge B_{A,t0}(<1 1k_b^{-1} B>) \wedge B_{A,t1}(<\cdots 1N_a\cdots>) \rightarrow B_{A,t1}(<\ldots 1N_a A>)$	$+\{\ldots A,N_a\ldots\}_{kb}$	(1),A5(f)
(6)	$+\{\ldots N_a\ldots\}_{kb} \wedge B_{A,t0}(<1 1k_b^{-1} B>) \wedge B_{A,t1}(<1 1N_a\ldots>) \rightarrow B_{A,t1}(\prec\{N_a,B\})$	$+\{\ldots N_a\ldots\}_{kb}$	(1), (4),A 2(b)
(7)	$B_{B,t0}(<1 1k_b^{-1} B>) \rightarrow B_{B,t0}(<1 1k_b^{-1}\ldots>)$		Def. of $<\ldots_1\ldots_2 N\ldots_3>$
(8)	$B_{B,t0}(<1 1k_b^{-1} B>) \rightarrow B_{B,t0}(\neg key(I,k_b^{-1}))$		Def. of $<\ldots_1\ldots_2 N\ldots_3>$
(9)	$-\{\ldots N_a\ldots\}_{kb} \wedge B_{B,t0}(\neg key(I,k_b^{-1})) \wedge B_{B,t0}(<1 1k_b^{-1}\ldots>) \rightarrow B_{B,t1}(<1\ldots N_a\ldots>)$	$-\{\ldots N_a\ldots\}_{kb}$	(7),(8),A3(a)

Step II. Upon receiving Message 2 $B \rightarrow A$: $\{N_a, N_b\}_{Ka}$

No.	Security properties	Term	Applied
(10)	$+\{\ldots N_b\ldots\} \rightarrow B_{B,t2}(<\cdots 1N_b\cdots>)$	$+\{\ldots N_b\ldots\}$	A0
(11)	$+\{\ldots N_b\ldots\}_{ka} \wedge B_{B,t0}(\neg key(I,k_a^{-1})) \wedge B_{B,t0}(<1 1k_b^{-1}\ldots>) \rightarrow B_{B,t2}(<1\ldots N_b\ldots>)$	$+\{\ldots N_b\ldots\}_{ka}$	(7),(8),A3(b)
(12)	$+\{\ldots N_b\ldots\}_{ka} \wedge B_{B,t0}(<1 1k_b^{-1} B>) \wedge B_{B,t0}(<1 1N_b\ldots>) \rightarrow B_{B,t2}(\prec\{N_b,A\})$	$+\{\ldots N_b\ldots\}_{ka}$	(10),(11) ,A2(b)
(13)	$B_{A,t0}(<1 1k_a^{-1} A>) \rightarrow B_{A,t0}(<1 1k_a^{-1}\ldots>)$		Def. of $<\ldots_1\ldots_2 N\ldots_3>$
(14)	$B_{A,t0}(<1 1k_a^{-1} A>) \rightarrow B_{A,t0}(\neg key(I,k_a^{-1}))$		Def. of $<\ldots_1\ldots_2 N\ldots_3>$
(15)	$-\{\ldots N_b\ldots\}_{ka} \wedge B_{A,t0}(\neg key(I,k_a^{-1})) \wedge B_{A,t0}(<1 1k_a^{-1}\ldots>) \rightarrow B_{A,t2}(<1\ldots N_b\ldots>)$	$-\{\ldots N_b\ldots\}_{ka}$	(13),(14),A3(a)
(16)	$-\{\ldots N_a,N_b\ldots\}_{ka} \wedge B_{A,t0}(<0 1k_b p>) \wedge B_{A,t0}(<1 1k_a^{-1} A>) \wedge B_{A,t1}(<\ldots 1N_a\ldots>) \rightarrow B_{A,t2}(\Rightarrow\{\ldots N_a,N_b\ldots\}_{ka})$	$-\{\ldots N_a,N_b\ldots\}_{ka}$	(1),A1(c)
(17)	$B_{A,t0}(<1 1k_a^{-1} A>) \rightarrow B_{A,t0}(key(A,k_a^{-1}))$		Def. of $<\ldots_1\ldots_2 N\ldots_3>$
(18)	$-\{\ldots N_a\ldots\}_{ka} \wedge B_{A,t1}(\prec\{N_a,B\}) \wedge B_{A,t0}(key(A,k_a^{-1})) \wedge B_{A,t1}(<\ldots 1N_a\ldots>) \rightarrow B_{A,t2}(<1B>)$	$-\{\ldots N_a\ldots\}_{ka}$	(1), (6),(17) ,A6(a)
(19)	$-\{\ldots N_a\ldots\}_{ka} \wedge B_{A,t1}(\prec\{N_a,B\}) \wedge B_{A,t0}(key(A,k_a^{-1})) \wedge B_{A,t1}(<\ldots 1N_a\ldots>) \rightarrow B_{A,t2}(<\ldots 1N_a B>)$	$-\{\ldots N_a\ldots\}_{ka}$	(1), (6),(17) ,A6(b)
(20)	$B_{A,t2}(<1 1N_a AB>)$		(1),(4),(5),(19) R3,R4,R5
(21)	$B_{A,t2}(\Rightarrow\{\ldots N_a,N_b\ldots\}_{ka}) \wedge B_{A,t2}(<\ldots 1N_a AB>) \rightarrow B_{A,t2}(<\ldots 1N_b AB>)$		(16),(20),A7
(22)	$B_{A,t2}(<1 1N_b AB>)$		(15),(21) ,R5

Step III. Upon receiving Message 3 $B \rightarrow A$: $\{N_b\}_{Kb}$

No.	Security properties	Term	Applied
(23)	$B_{B,t0}(<1 1k_b^{-1} B>) \rightarrow B_{B,t0}(key(B,k_b^{-1}))$		Def. of $<\ldots_1\ldots_2 N\ldots_3>$
(24)	$-\{\ldots N_b\ldots\}_{kb} \wedge B_{B,t2}(\prec\{N_b,A\}) \wedge B_{B,t0}(key(B,k_b^{-1})) \wedge B_{B,t2}(<\ldots 1N_b\ldots>) \rightarrow B_{B,t3}(<1A>)$	$-\{\ldots N_b\ldots\}_{kb}$	(10), (12),(23),A6(a)
(25)	$-\{\ldots N_b\ldots\}_{kb} \wedge B_{B,t2}(\prec\{N_b,A\}) \wedge B_{B,t0}(key(B,k_b^{-1})) \wedge B_{B,t2}(<\ldots 1N_b\ldots>) \rightarrow B_{B,t3}(<\ldots 1N_b A>)$	$-\{\ldots N_b\ldots\}_{kb}$	(10), (12),(23),A6(b)
(26)	$-\{\ldots N_b\ldots\}_{kb} \wedge B_{B,t0}(<1 1k_b^{-1} B>) \wedge B_{B,t}(<0 1k_b p>) \wedge B_{B,t2}(<\ldots 1N_b\ldots>) \rightarrow B_{B,t3}(<\ldots 1N_b B>)$	$-\{\ldots N_b\ldots\}_{kb}$	(10),A5(e)
(27)	$B_{B,t3}(<1 1N_b AB>)$		(10),(11),(25),(26),R3,R4,R5
(28)	$b_{A,t_0} = \lfloor <1 1N_a AB>, <1 1N_b AB>, <1B> \rfloor$		(20),(22),(18)
(29)	$b_{B,t_0} = \lfloor <1\ldots N_a\ldots>, <1 1N_b AB>, <1A> \rfloor$		(9) ,(27),(24)

D. Compare with the security goals established in step A.

(1) Absence of the association of freshness identifier *Na* with *A* and *B* for principal *B*

→ Cheat *B* by confusing *N'a* between the attacker *I* and *A* with *Na* between *A* and *B*.

(2) The liveness of the principal *A* under the condition (1)

→ Suggest that A must be alive to act as an oracle in this attack, so it is a concurrent run attack.

From above analysis, we can get the attack structure: an interleave attack, confusing *Na* to cheat *B*. We are glad to find that the constructed not just discovered attack is the same as the well-known flaw discovered by Lowe using FDR [14].

4.2 Analysis of the N-S-L public key protocol

In [14], Lowe gives a correction of the original N-S public key protocol (say, N-S-L protocol) in Message 2 $B \rightarrow A$: $\{B, N_a, N_b\}_{Ka}$. Low proves the correctness of the revised protocol using FDR. We will give a different proof using the belief multisets approach. We omit the similar analysis in IV.4.1.

C. Establish the security properties of N-S-L protocol

Step II. Upon receiving Message 2 $B \rightarrow A$: $\{B, N_a, N_b\}_{Ka}$

No.	Security properties	Term	Applied
(29)	$+\{\ldots B,N_b\ldots\}_{ka} \wedge B_{B,t0}(<1 1k_a^{-1} A>) \wedge B_{B,t0}(<0 1k_b p>) \wedge B_{B,t2}(<\ldots 1N_b\ldots>) \rightarrow B_{B,t2}(<\ldots 1N_b B>)$	$+\{\ldots B,N_b\ldots\}_{ka}$	(10),A5(f)

(30) $+\{\ldots N_a, N_b \ldots\}_{k_{ab}} \wedge B_{B,t0}(<11k_a^{-1}A>) \wedge B_{B,t0}(<01k_BP>) \wedge B_{B,t2}(<\ldots 1N_bB\ldots>) \rightarrow B_{B,t2}(\Rightarrow\{\ldots N_a, N_b\ldots\}_{k_{ab}})$ $+\{\ldots B, N_a, N_b\ldots\}_{k_{ab}}$ (10),A1(e)

Step III. Upon receiving Message 3 $B \rightarrow A$: $\{N_b\}_{Kb}$

No.	Security properties	Term	Applied
(31)	$B_{B,t2}(\Rightarrow\{\ldots N_a, N_b\ldots\}_{k_{ab}}) \wedge B_{B,t1}(<\ldots 1N_bAB>) \rightarrow B_{A,t1}(<\ldots 1N_aAB>)$		(30),(27)
(32)	$B_{A,t1}(<11N_aAB>)$		(9),(31)
(33)	$b_{B,t_4} = \lfloor <11N_aAB>, <11N_bAB>, <1A> \rfloor$		(27),(32),(24)

Up to now, both principal A and B have achieved the security goals of the protocol, hence N-S-L protocol is MK-secure.

5. Automation

As we have seen, the analysis process of the belief multisets formalism is completely rigorous and amenable for automation. First, the premise set of a cryptographic protocol is clear. For example, $B_{pi,t0}(<11ki^{-1}pi>, <11kj^{-1}pj>, <01kip>, <01kjp>)$ and $B_{pi,t0}(<11kj^{-1}pj>, <11ki^{-1}pi>, <01kip>, <01kjp>)$ for public key case; $B_{pi,t0}(<11kijpipj>)$ and $B_{pj,t0}(<11kijpipj>)$ for shared key case without trusted third party; $B_{pi,t0}(<11kispiS>)$ and $B_{pj,t0}(<11kjspjS>)$ for shared key case with trusted third party. Second, the security goals of a cryptographic protocol can be accurately specified. For example, to achieve MK secure of a key agreement protocol, the security goals are $b_{pi,t_4} = \lfloor <11N_{pi}p_ip_j>, <11N_{pj}p_ip_j><1p_j> \rfloor$ and $b_{pj,t_4} = \lfloor <11N_{pj}p_ip_j>, <11N_{pj}p_ip_j><1p_i> \rfloor$. Third, the security properties of a cryptographic protocol, from the point of view of each legitimate participant, can be inferred rigorously via the premise set, axioms and rules in the belief multisets, as we have done in subsection IV.4.1 and IV.4.2. Finally, the construction of an attack can be directly deduced from the absence of the protocol security.

We contribute an accomplishment of the belief multisets formalism similar to SPEAR II Version 1.0.0.25 via prolog language. Fig. 2 is a fragment of the analysis procedure on the security of N-S-L protocol. We have to emphasize that the introduced automation of the belief multisets formalism is still in its first place, and we'll do our best to improve it for future practical application.

```
[16] Proof for B believes that A associate with Na:
1. B believes that -Ka is secret. {Assumption}
2. B believes that -Ka is fresh. {Assumption}
3. B believes that -Kb is secret. {Assumption}
4. B believes that -Kb is fresh. {Assumption}
5. B believes that +Kb is fresh. {Assumption}
6. B believes that +Ka is fresh. {Assumption}
7. B believes that A associate with -Ka. {Assumption}
8. B believes that B associate with -Kb. {Assumption}
9. B believes that A associate with +Ka. {Assumption}
10. B believes that B associate with +Ka. {Assumption}
11. B believes that A associate with +Kb. {Assumption}
12. B believes that B associate with +Kb. {Assumption}
13. B once said E(+Ka : B, Na, Nb). {Step}
14. B was told E(+Kb : Nb). {Step}
15. B once said (B, Na, Nb). {13, 10, Said5}
16. B believes that (B, Na, Nb) is secret. {13, 10, Secret3}
17. B believes that (Na, Nb) is secret. {16, Secret9}
18. B once said (Na, Nb). {15, Said2}
19. B believes that Nb is secret. {17, Secret9}
20. B once said Nb. {18, Said2}
21. B believes that Nb is fresh. {20, Generate3}
22. B believes that (B, Na, Nb) are bound in the session.
{13, 1, 6, 2, 7, 10, 9, 21, Fragment5}
23. B believes that A receives Nb. {13, 1, 2, 7, 21, 19, Expect2}
24. B believes that B associate with Nb. {14, 3, 4, 5, 8, 12, 11, 21, Associate5}
25. B believes that A associate with Nb. {23, 14, 8, 21, ExpectInference3}
26. B believes that B and A associate with Nb. {24, 25, Associate7}
27. B believes that B and A associate with Na. {22, 21, 26, FragmentInference1}
28. B believes that A associate with Na. {27, Associate9}
```

Fig. 2 Illustration of protocol analysis via the belief multisets automation

V. Conclusion

A novel idea of protocol security analysis is presented based on trusted freshness. The idea has been efficiently implemented not only by hand but also by a belief multisets formalism for automation. The key of the security analysis based on trusted freshness is the freshness principle: for each participant of a cryptographic protocol, the security of the protocol depends only on the sent or received one-way transformation of a message, which includes a trusted freshness. A manual freshness analysis method and a belief multisets formalism are also established on the basis of the freshness principle.

Security analysis based on trusted freshness can efficiently distinguish whether a message is fresh or not, and the analysis result suggests the correctness of a protocol or the way to construct attacks intuitively from the absence of security properties. Furthermore, the security analysis based on trusted freshness is independent of the idealization of a protocol, the concrete formalization of attackers' possible behaviors, and the concurrent runs of protocols.

The security analysis method described in this paper is useful in our own work and it becomes an exciting and interesting job to analyze cryptographic protocols. In this

paper we have confined ourselves to the authentication properties. We will extend our work to include other security properties in the future and we hope the security analysis method based on trusted freshness will help more researchers to improve their security communication protocols.

Acknowledgement(s) This work was supported by the National High Technology Development 863 Program of China under Grant No. 2006AA01Z422, the National Natural Science Foundation of China under Grant No.60573030 and No. 90704004.

References

[1] R. Needham and M.D.Schroeder. "Using encryption for authentication in large network of computers". Communication of the ACM, vol.21, no.12, pp. 993-999, 1978.

[2] G. Lowe. "An attack on the needham-schroeder public key authentication protocol". Information Processing letters, vol.56, no.3, pp. 131-133, 1995.

[3] M. Abadi and R. Needham. "Prudent engineering practice for cryptographic protocols". IEEE Transactions on Software Engineering, vol.21, no.1, pp. 6-15, Jan. 1996.

[4] D.Dolev and A.C.Yao. "On the security of public key protocols". IEEE Transactions on Information Theory, vol.29, no.2, 1983.

[5] M. A. M. Burrows and R. Needham. "A logic of authentication". Proc. Royal Soc. London A, vol. 426, pp. 233-271, 1989.

[6] P. Syverson and P. van Oorschot. "On unifying some cryptographic protocol logics". IEEE Computer Society Symposium on Research in Security and Privacy, 16-18 May 1994, pp. 14-28.

[7] G.lowe. "Toward a completeness result for model checking of security protocol". pp. 1-48, June 1999.

[8] F. Fabrega, J. Herzog, and J. Guttman. "Strand spaces: why is a security protocol correct? ". Proc. 1998 IEEE Symposium on Security and Privacy, 3-6 May 1998, pp. 160-171.

[9] J. Guttman and F. Thayer. "Authentication tests". Proc.IEEE Symposium on Security and Privacy, 14-17 May 2000, pp. 96-109.

[10] M. Bellare, P. Rogaway. "Random oracles are practical: a paradigm for designing efficient protocols". In First ACM Conference on Computer and Communications Security New York: ACM Press , 1993, 62-73.

[11] M. Bellare, P. Rogaway. "Entity authentication and key distribution".' Proc. of CRYPTO'93, LNCS 773, 1993, pp. 232-249.

[12] W. Mao, Modern Cryptography: Theory and Practice. English reprint Copyright by PEARSON EDUCATION NORTH ASIA LIMITED and Publishing House of Electronic Industry, 2004.

[13] S. Goldwasser and S. Micali. "Probabilistic encryption." JCSS, vol.28, no.2, pp. 270-299, April 1984.

[14] G. Lowe. "Breaking and fixing the needham-schroeder public key protocol using FDR". Proc. of TACAS, vol. 1055, Springer Verlag, 1996, pp. 147-166.

[15] R. Canetti and H. Krawczyk. "Analysis of key-exchange protocols and their use for building secure channels". Proc. of EUROCRYPT 2001, LNCS 2045, 2001, pp. 453-474.

[16] L. Dong, K. Chen, X. Lai. "Belief Multisets for Cryptographic Protocol Analysis". Journal of Software (To Appear).

[17] L. Dong, K. Chen, Y. Zheng, X. Hong. "The Guarantee of Authentication Protocol Security ". Journal of Shanghai JiaoTong University (in Chinese), vol. 42, no.4, pp.518- 522, 2008.

[18] Otway D, and O. Rees. "Efficient and Timely Mutual Authentication". Operating Systems Review, vol.21, pp. 8-10, 1987.

[19] D. Denning and G. Sacco. "Timestamps in key distribution protocols". Communication of the ACM, vol.24, no.8, pp. 533-536, 1978.

Authors

Kefei CHEN received the B.S. and the M.S. degree in applied mathematics from Xidian University, Xi'an, in 1982 and 1985, respectively, and the Ph.D. degree from Justus-Liebig University, Gießen, Germany, in 1994.

From 1982 to 1990, he served as a lecturer in the Department of Applied Mathematics and Information Security Institute. He got the German Konrad Adenauer Scholarship and visited 1990-1991 in Goethe-Institute Mannheim, spent 1991-1995 as a PhD candidate and researcher assistant at the security group in the mathematics Institute, Gießen, Germany, respectively. He came to Shanghai Jiaotong University as associate professor in 1996, and assumed his present position as professor in 1998.

Dr. Chen has been concentrated his work in cryptography and information security in the past years, he has been awarded many projects such as National Natural Science Foundation of China and National High-Tech (863) Programs of China etc, he is also the author of more than 100 research papers and 9 books.

He is invited as a member of Standing Council of Chinese Association for Cryptologic Research, the Expert Panel Committee in the Department of Information Sciences of NSFC (2004-2007), he also is a member of the academic committee both of the State Key Laboratory of Information Security at the Chinese Academy of Sciences and the State Key Laboratory for Novel Software Technology at Nanjing University, member of Editorial Board of the Journal of Software and IET Electronic Letters (Chinese Edition).

Ling DONG received the Ph.D. degree of computer architecture from Shanghai Jiaotong University, Shanghai in 2008. Her research interest lies in information security and applied cryptosystems, especially in the design and analysis of practical cryptographic communication protocols, and practical cryptosystems.

XueJia LAI received Ph.D. of sc. techn in 1992 from the Swiss Federal Institute of Technology, Zurich. He is a professor of Shanghai Jiao Tong University since 2004. His work has been concentrated in cryptography during the past 20 years, especially in the design and analysis of practical cryptosystems (including block ciphers and stream ciphers), differential cryptanalysis of block ciphers, and analysis and hash functions. He is co-inventor (together with J. L. Massey) of the IDEA cipher. In 1994, he joined r3 security engineering ag which became Entrust Technologies Switzerland since June 1998. He was at S.W.I.S. GROUP, Switzerland since 2001. He has published a book "On the Design and Security of Block Ciphers" (Hartung-Gorre Verlag, 1992) and more than 50 papers.