Enhancing SVO Logic for Mobile IPv6 Security Protocols

Ilsun You
Korean Bible University
16 Danghyun 2-gil, Nowon-gu
Seoul, Republic of Kroea
isyou@bible.ac.kr

Yoshiaki Hori and Kouichi Sakurai
Kyushu University
Fukuoka-shi 819-0395, Japan
{hori,sakurai}@csce.kyushu-u.ac.jp

Abstract

In order to protect Mobile Internet Protocol Version 6 (MIPv6), considerable researches have been made, consequently followed by various security protocols, which are based on public key cryptography. Especially, depending on a proper address based public key method, these protocols use each node's address as a public key certificate to authenticate its public key because no global public key infrastructure is available in MIPv6 environments. In addition, they execute an appropriate address test to check if a node exists at its claimed address. With such security features, the protocols prevent critical attacks including redirect, man-in-the middle, and denial of service ones. On the other hand, it is clearly of paramount importance to formally evaluate the MIPv6 security protocols to design them without flaws. Unfortunately, there is lack of the formal verification method to precisely reason about their correctness while considering their unique security properties to our best knowledge. In this paper, we propose an extended SVO logic for the thorough verification of the MIPv6 security protocols. Then, we show its effectiveness by applying the proposed logic to four security protocols.

Keywords: MIPv6 security, Formal verification, SVO logic

1 Introduction

Since introduced in 2004, Mobile Internet Protocol Version 6 (MIPv6) has been an important standard protocol for IP mobility management [1]. The main goal of this protocol is to support nodes to stay reachable regardless of their movements and locations in IPv6-based networks. It is supposed in MIPv6 that each *Mobile Node* (*MN*) belongs to a home network, while being assigned two addresses, *Home Address* (*HoA*) and *Care-of Address* (*CoA*) where *HoA* is a permanent address used for identification, *CoA* is a temporal one used for routing, and their relation is called 'binding'. It is necessary for every *MN* to update its binding information whenever changing its location (called 'binding update'). Also, a *MN* is assumed to share a secure connection with its *Home Agent* (*HA*), which is a router in its home network. MIPv6 presents two possible options for communications between a *MN* and its *Corresponding Node* (*CN*). In the first option, called *Bidirectional Tunneling* (BT), *HoA* is used for communication between *CN*s and *MN*s. It means that when a *MN* is located at a foreign network, the packets sent to/from its *HoA* are routed to/from its *CoA* by a *HA*. For this option, each *MN* should inform its *HA* of a new *CoA* whenever moving to a new network. However, due to the triangle routing, this option results in critical inefficiencies. On the other hand, in the second option, called *Route Optimization* (RO), *CoA* instead of *HoA* is used to allows direct relay of packets between a *MN* and its *CN*. Such a direct routing optimizes the performance while excluding *HA*s' involvement. This option needs a *MN* to register its current binding information at both its *HA* and *CN* by performing the binding update processes. However, because the path between a *MN* and its *CN* is not secure, this option can make involved nodes vulnerable to various security threats if it is not protected. Motivated by this, the *Return Routeability* (RR) protocol was proposed as a standard [1]. However, due to its performance and security problems, various security protocols have been proposed based on the public key cryptography [2, 3, 4, 5, 6, 7]. Especially, to protect the binding update process between two previously unknown nodes, *i.e.*, *MN* and *CN*, on the assumption that no global security infrastructure is available, they introduced the novel address based public key methods such as *Cryptographically Generated Address* (CGA) [3, 6], *Statistical Uniqueness and Cryptographic Verifiability* (SUCV) [4], and *Address Based Keys* (ABK) [5]. In those protocols, unlike traditional ones, HoA is used as a certificate to authenticate the public key of its owner. Also, to prevent the redirect attacks [8], *MN*s are checked if they are indeed at their argued address while their binding update messages are authenticated.

In spite of the efforts made by the above protocols, there has been lack of the formal verification method to precisely reason about their correctness while considering their unique security properties to our best knowledge. Clearly, such a method is important and necessary to exactly evaluate security protocols as well as design them without flaws. On the other hand, the formal verification techniques for security protocols can be divided into modal logic, state enumeration and theorem proving [9, 10, 11]. Compared to other techniques, modal logic is decidable and relatively simple while not difficult to apply, thus having been widely applied for security analysis. In this paper, we study the formal analysis on the security protocols for MIPv6 while focusing on modal logic. More importantly, we extend SVO logic [12, 13], which is the most mature and successful modal logic technique, to precisely analyze MIPv6 security protocols. In addition, four security protocols are formally verified with the extended SVO logic. This paper is organized as follows. In section 2, we briefly survey the modal logic approach and describe the MIPv6 security. Section 3 presents the extended SVO logic for MIPv6, with which four security protocols are then analyzed in section 4. Section 5 concludes this paper.

2 Related works

2.1 Mobile IPv6 Security

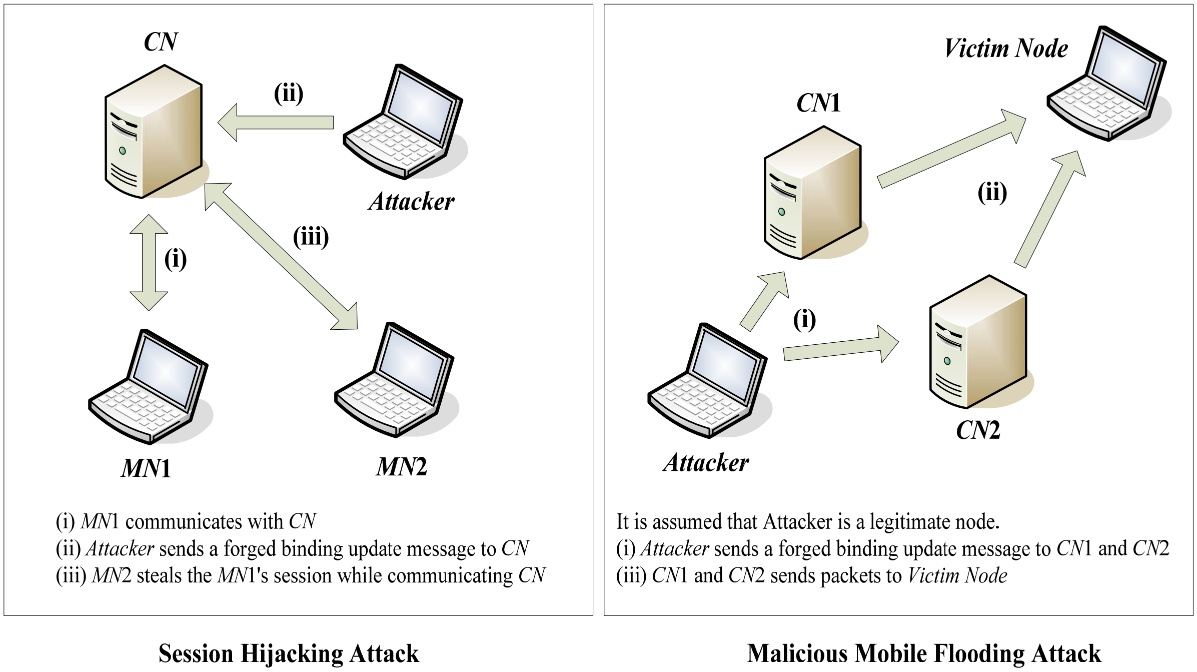
In MIPv6, when moving to a new network, every *MN* should inform both its *HA* and *CN*s of its new location, *i.e.*, *CoA*, through the binding update message. If such a binding update procedure is not secured, MIPv6 is vulnerable to the redirect attacks [8].

As shown in Figure 1, the redirect attacks can be classified into two categories: *Session Hijacking* (SSH) and *Malicious Mobile Node Flooding* (MMF).

- ***Session Hijacking* (SSH)**: The SSH attack is a redirect one, which is launched by intruders masquerading victims, and its main goal is to steal victims' session. In this attack, it is assumed at first that the *MN*1, which is a victim, communicates the *CN*. The $Attacker$ tries to launch this attack by sending the *CN* a forged binding update message or an old one, which claims that the *MN*1 has moved to a new *CoA* owned by the *MN*2. If successful, this attack can cause the *CN* to redirect the *MN*1's traffic to the *MN*2 while resulting in information leakage.

- ***Malicious Mobile Node Flooding* (MMF)**: The MMF attack is a redirect one, which is launched by legitimate but malicious *MN*s, aiming at making victims flooded with a lot of packets. Prior to launching this attack, the $Attacker$ has to communicate with several *CN*s, *i.e.*, *CN*1 and *CN*2. It starts this attack by sending its *CN*s a binding update message arguing the it has moved to the $Victim\ Node$'s location. If the *CN*s approve the message, they redirect the *MN*'s traffic to the $Victim\ Node$ at the same time.

Accordingly, to protect the binding update procedure, the above two redirect attacks should be prevented. In addition, the binding update procedure has to be carefully designed not to be vulnerable to the *Man-In-The-Middle* (MiTM) and *Denial of Service* (DoS) attacks, which are able to happen in public key based protocols.



(i) *MN*1 communicates with *CN*
(ii) *Attacker* sends a forged binding update message to *CN*
(iii) *MN*2 steals the *MN*1's session while communicating *CN*

It is assumed that Attacker is a legitimate node.
(i) *Attacker* sends a forged binding update message to *CN*1 and *CN*2
(iii) *CN*1 and *CN*2 sends packets to *Victim Node*

Session Hijacking Attack

Malicious Mobile Flooding Attack

Figure 1: Redirect attack

In order to prevent the SSH attack, it is necessary to authenticate both *MN*s and their binding update messages. However, it is a big challenge to allow two previously unknown nodes to authenticate each other in MIPv6 environments where there is no global CA or trusted third party. The contemporary MIPv6 security protocols have tried to defend against the SSH attack by adopting novel address based public key methods such as CGA, ABKs, SUCV and so forth. With the help of these methods, the protocols can authenticate both *MN*s and their binding update messages in addition to verifying their address ownership. On the other hand, the MMF attack can be countered by checking if *MN*s exist at their claimed address (called "address test"). For the address test, a nonce-based approach is widely applied. In this approach, a *MN* has to demonstrate that it knows the nonces which have been sent to its claimed address.

Note that it is important to formalize the above countermeasures to support to verify MIPv6 security protocols.

2.2 Modal Logic

Modal logic is composed of diverse statements and inference rules [10]. While the statements express belief in or knowledge about messages in security protocols, the inference rules are used to derive new beliefs from other beliefs and/or new knowledge from other knowledge and beliefs. In this approach, given a target security protocol, its initial assumptions and goals are firstly defined, and its original form is converted for verification. The inference rules are then repeatedly applied to obtain reasonable beliefs or knowledge which satisfy the defined goals.

The most famous and influential modal logic is BAN logic, which was introduced by Burrows, Abadi and Needham [14]. Because of being successful, BAN logic opened the door for substantial researches in the field of formal verification of security protocols. Also, it was extended to various modal logics such as AT, GNY, VO and SVO [15, 16, 17, 12, 13].

- **BAN**: The goal of BAN logic is to formalize reasoning about authentication protocols [14]. This logic consists of a language for expressing the beliefs of the involved parties in a target protocol as well as a set of inference rules used for deriving new beliefs. The language and inference rules will be described in the next section. For verification, BAN logic typically takes the following steps: (i) idealizing the original protocol, (ii) defining assumptions about the initial state (iii) applying inference rules repeatedly until getting the intended results. The advantage of BAN logic is that it is simple, intuitive and easy while providing relatively small proofs and being still useful. However, it has some limitations [10, 11, 13]. First, there is no way to evaluate if the idealized form of a target security protocol is indeed valid. It means that the task of idealization should be manually conducted without any guidance, thus being error-prone. Second, like the problem of the idealization step, this logic lacks a method to check the validity of the initial assumptions. Thus, it can even include ridiculous assumptions, which clearly leads to strange conclusions. Third, due to sticking to the simple structure, BAN logic's syntax and inference rules are not enough to precisely reason about some security protocols. For example, regarding a key, there is only the message meaning rule, which focuses on the belief that the involved parties, P and Q, are only principals who can encrypt a message M with a key K rather than key secrecy. This rule is too strong to support various cases. As another example, BAN logic assumes that all the involved parties are honest. However, such an assumption cannot allow this logic to find some security flaws caused by malicious or dishonest parties.

- **GNY** and **VO**: In 1990, Gong, Needham and Yahalom extended BAN logic to overcome its limitations [15] (this is called GNY logic). In particular, GNY logic distinguishes what one possesses from what one believes in, thereby being able to separately handle the content of a message and its intended information. For example, this logic can express "Q has a key K shared between P and Q", where Q does not have to believe K. Such a separation can differentiate the reasoning about the physical world from the reasoning about other principles' beliefs, thus achieving multiple levels of trust in reasoning. Another contribution of GNY logic is to remove several universal assumptions which BAN logic has by introducing the new notions of *recognizability* and *not-originzted-here*. While the notion of *recognizability* can express that a recipient is able to recognize what she expects, that of *not-originzted-here* can describe that a recipient is able to discern between her own generated messages and others. Moreover, GNY logic modifies and expands the syntax and inference rules of BAN logic. Based on the above extension, GNY logic considerably broadens the scope of application. However, this logic is too complex to be easily applied due to more than 40 inference rules, and still has no way to evaluate if the assumptions and idealized forms are valid. On the other hand, in 1993, van Oorschot introduced VO logic [17], which extends BAN and BAN-like logics to precisely reason about the security protocols based on the Diffie-Hellman key agreement scheme. Such an extension is quite useful because the Diffie-Hellman key agreement scheme has had considerable impact on modern security protocols. Also, this logic provides the six formal authentication goals while formalizing reasoning about corroborated possession of secrets.

- **AT**: In 1991, Abadi and Tuttle reformulated BAN logic to provide a new semantics for it [16] (this is called AT logic). For this goal, they tried to model the concepts the logic is trying to capture. Especially, they introduced the new notions, "P actually possesses the key K" denoted P *has* K and "P has sent X in the present" denoted P *says* X. With help of these notions, AT logics removes some unnecessary mixing of semantic and implementation from BAN logic while providing more direct definitions to dispense with an implicit assumption of honesty. Also, this logic improves the syntax and inference rules of BAN logic as follows. First, AT logic makes distinction between arbitrary expression and formulas by defining a language of messages. Second, AT logic provides all the propositional connectives such as negation, disjunction and implication, which enable the inference rules to be rewritten as axioms. Third, the inference rules are simplified to define all concepts independently, then reformulated as axioms, most of which exclude belief, with modus pones and necessitation as the only inference rules. Compared to BAN, GNY and VO, AT logic looks more like traditional modal logic and its meaning is clearer based on the new robust semantics.

- **SVO**: As mentioned above, several extensions such as GNY, AT and VO were proposed to overcome the limitations of BAN logic. Each of extensions has its own advantages and disadvantages. It is natural that there is a need to combine BAN logic and its extensions to take their all advantages. Motivated by this, SVO logic was introduced by Syverson and van Oorschot [13], then becoming an outstanding successor of BAN logic. For verification, this logic takes the following steps: (i) defining assumptions about the initial state (ii) annotating a target security protocol (iii) asserting comprehensions of the received messages (iv) asserting interpretation of comprehended messages (v) applying inference rules repeatedly until getting the intended results. Especially, it is worth to note that SVO logic splits the idealization of BAN logic into steps (iii) and (iv) to address its problem. This logic gracefully unifies its predecessors by taking their eligible features while being still relatively simple in comparison with GNY, AT and VO. Importantly, it achieves a model-theoretic semantics, to which it is proved to be sound with respect. Nevertheless, the logic does not provide the evaluation method to check the initial assumptions and interpretations.

There have been attempts to formally verify the MIPv6 security protocols based on BAN logic [3, 18, 19, 20]. Note that the protocols make use of the digital signature and public key encryption with the help of the CGA method. However, since BAN logic doesn't support the CGA method, those attempts just assume that each involved node believes its correspondent's public key. Also, they are unable to derive the belief on the address ownership. Moreover, they define the goal of the protocol as "*CN believes MN believes BU*" or "*CN believes MN believes CoA*", where *BU* denotes a binding update message. Note that such a goal cannot show that the target protocol is not vulnerable to the MMF attack because the *CN* depends on only the *MN*'s belief. In other words, the *CN*'s belief can be misused by the legitimate but malicious *MN*. Though the address test is used to count this attack as mentioned above, its verification is not available in BAN logic.

To our best knowledge, other modal logic techniques have the same limitation as BAN logic. Motivated by this, we extend SVO logic to reason about the new security features introduced by the MIPv6 security protocols.

3 Extension of SVO logic

In this section, SVO logic is briefly described, and then extended for formal verification of MIPv6 security protocols.

3.1 SVO logic

As described above, SVO logic successfully combines GNY, AT and VO while keeping simplicity. SVO logic is composed of two inference rules and twenty two axioms [12, 13]. For verification, this logic takes the following steps: (i) defining assumptions about the initial state (ii) annotating a target security protocol (iii) asserting comprehensions of the received messages (iv) asserting interpretation of comprehended messages (v) applying inference rules repeatedly until getting the intended results.

Here, we introduce SVO logic's notations, inference rules and axioms, which are used in our extension and verification. For details on SVO logic, refer to [12, 13].

3.1.1 Notations

The notations used in SVO logic are as follows: (where P and Q are principals, X is a message, and K is a key)

- P *believes* X: P acts as if X is true.
- P *received* X: P has recevied a message including X.
- P *said* X: P sent X at one time.
- P *controls* X: P has jurisdiction on X.
- $fresh(X)$: X is fresh.
- $P \overset{K}{\longleftrightarrow} Q$: K is a shared key between P and Q. It can be shared by a trusted third party of P and Q.
- $\{X\}_K$: X is encrypted with K.
- $PK(P,K)$: K is a public key of P. Also, $PK_\sigma(P,K)$ and $PK_\psi(P,K)$ can be used to denote K as a public signature key and a public ciphering key respectively.
- $\lfloor X \rfloor_K$: X is signed with K.
- $SV(X,K,Y)$: Given a signed message X, applying K to it verifies that X is the result of signing Y with the corresponding private key of K.
- $\langle X \rangle_{*P}$: P does not know or recognize X but P will recognize $\langle X \rangle_{*P}$ if it will receive the message again.
- $X\ from\ P$: X was sent by P.

3.1.2 Inference Rules

SVO logic has the two inference rules: modus ponens and necessitation.

- Modus Ponens (MP):

$$\frac{\varphi \quad \varphi \longrightarrow \psi}{\psi}$$

- Necessitation (NE):

$$\frac{\vdash \varphi}{\vdash\ P\ believes\ \varphi}$$

φ, ψ and $\vdash$ are metalinguistic symbols. While ψ and $\vdash$ are used to refer to arbitrary formulae, '$\Gamma \ \vdash \ \psi$' means the formula ψ can be derived from the set of formulae Γ (and the axioms as stated below). Also, '$\vdash \varphi$' means that φ is a theorem, *i.e.*, deriavable from axioms alone.

3.1.3 Axioms

- Belief Axioms (BA)
  BA1: $(P\ believes\ \varphi\ \bigwedge\ P\ believes\ (\varphi\ \rightarrow\ \psi))\ \rightarrow\ P\ believes\ \psi$
  BA2: $P\ believes\ \varphi \rightarrow \varphi$

  For any principal P and formulae φ and ψ, BA1 means that P believes all that logically follows from its beliefs and BA2 means P can say what it believes. Note that belief is removed from most of other axioms.

- Source Association Axioms (SAA)
  SAA1: $(P \overset{K}{\longleftrightarrow} Q\ \bigwedge\ R\ received\ \{X\ from\ Q\}_K)\ \rightarrow\ (Q\ said\ X\ \bigwedge\ Q\ has\ X)$
  SAA2: $(PK_\sigma(Q,K)\ \bigwedge\ R\ received\ X\ \bigwedge\ SV(X,K,Y))\ \rightarrow\ Q\ said\ Y$

- Receiving Axioms (RA)
  RA1: $P\ received\ (X_1,...,X_n)\ \rightarrow\ P\ received\ X_i,\ for\ i = 1,..,n$

- Saying Axioms (SA)
  SA1: $P\ said\ (X_1,...,X_n)\ \rightarrow\ (P\ said\ X_i\ \bigwedge\ P\ has\ X_i),\ for\ i = 1,..,n$
  SA2: $P\ says\ (X_1,...,X_n)\ \rightarrow\ (P\ said\ (X_1,...,X_n)\ \bigwedge\ P\ says\ X_i),\ for\ i = 1,..,n$

- Freshness Axioms (FA)
  FA1: $fresh(X_i)\ \rightarrow\ fresh(X_1,...,X_n),\ for\ i = 1,..,n$

- Jurisdiction and Nonce-Verification Axioms
  NVA: $(fresh(X)\ \bigwedge\ P\ said\ X)\ \rightarrow\ P\ says\ X$
  JA: $(P\ controls\ \varphi\ \bigwedge\ P\ says\ \varphi)\ \rightarrow\ \varphi$

3.2 Extension

Because no global public key infrastructure is available, the MIPv6 security protocols typically adopt their own address based public key methods such as CGA to verify each node's public key and address ownership. Also, the nonce based address test is used to verify if a node indeed exists at its claimed address. As described above, these security methods are essential to defend against the redirect attacks. Therefore, it is required to formalize them for precise security analysis on the MIPv6 security protocols.

In this subsection, we extend SVO logic to support such a formalization by defining new notations and axioms.

3.2.1 New notations

For new axioms, we firstly define the following notations.

- $ADP(P,A,K)$: The address parameters P (ex. CGA parameters) indicates that the key K is derived from the address A.

- $KA(Q,K,A)$: The principal Q, the key K and the address A are related to each other.

- $OWN(Q,A)$: The principal Q is the owner of the address A.

- $RR(X,Q,A)$: The value X has been sent to the address A to check if the principal Q exists at A.

- $EV(X,K,Q)$: The value X has been encrypted with the public key K and sent to the principal Q.

- $+\{X\}_K$: $(X, MAC(K,X))$, where K is a shared key, X is a message, and $MAC(\cdot)$ is a MAC or $HMAC$ function.

3.2.2 New axioms

Here, we propose new axioms for MIPv6 security protocols, among which the first three ones (*i.e.*, MIP1, MIP2, and MIP3) are added to reason about the verification for the public key and the address ownership, MIP4 is used to reason about the address test, and the last two axioms (*i.e.*, SAA3 and SA3) are presented to support the MAC or HMAC based message verification and the public key encryption.

In the following axioms, it is assumed that P and Q are principals, K is a key, A is an address, and AP denotes address parameters related to a public key method (ex, CGA parameters). Also, X and Y are supposed to be messages.

- Mobile Internet Protocol 1 (MIP1)
  $((R\ received\ AP\ from\ Q) \bigwedge ADP(AP, A, K))$
  $\longrightarrow KA(Q,K,A) \bigwedge PK(Q,K)$,
  where $PK(Q,K)$ can be $PK_\sigma(Q,K)$ or $PK_\psi(Q,K)$ based on its type.

  In address based public key methods such as CGA, public keys are verified through their corresponding address and address parameters. MIP1 formalizes such a public key verification. Its meaning is that if R received from Q AP indicating that A is related to K, Q has K as its public key as well as Q, K and A are related to each other. Note that MIP1 does not say that Q is the owner of A, but that Q is just related to K and A.

- Mobile Internet Protocol 2 (MIP2)
  $(KA(Q,K,A) \bigwedge PK_\sigma(Q,K) \bigwedge (R\ received\ X\ from\ Q) \bigwedge SV(X,K,Y))$
  $\longrightarrow (OWN(Q,A) \bigwedge Q\ said\ Y)$

  In address based public key methods, digital signature is typically used to verify the address ownership. For this, a principal has to show that it has the private key corresponding to its claimed address through digital signature. MIP2 formalizes such an address ownership verification. Its meaning is that Q owns A if any principal R received X from Q and the message is the signature on Y which can be validated with K. For this axiom to be true, there should be the assumption that Q is related to K and A while having K as its public key.

- Mobile Internet Protocol 3 (MIP3)
  $(KA(Q,K,A) \bigwedge PK_\psi(Q,K) \bigwedge Q\ says\ X \bigwedge EV(X,K,Q))$
  $\longrightarrow OWN(Q,A)$

  To verify the address ownership, public key encryption can be used as an alternative instead of digital signature. In this verification, a principal firstly receives a message that is encrypted with the public key related to its claimed address. Once receiving such a message, the principal decrypts it into its original form, the knowledge of which is then announced to the correspondent to demonstrate that that principal has the private key corresponding to its claimed address. This address ownership verification is formalized as MIP3. It means that Q owns A if Q says X, which was encrypted with K and sent to Q. Similarly to MIP2, for this axiom to be true, there should be the assumption that Q is related to K and A while having K as its public key.

- Mobile Internet Protocol 4 (MIP4)
  $(RR(X,Q,A) \bigwedge (Q\ says\ X)) \longrightarrow (Q@A)$

To prevent the MMF attack, it is important to check if a principal exists at its argued addresses (*i.e.*, care-of address and home address). MIPv6 security protocols use the nonce based address test which sends a nonce to a principal's address and checks if that principal knows it. MIP4 formalizes this address test while meaning that Q is at A if X has been sent to Q's A and Q says X. For this axiom, we define a new notion, '$X@A$' meaning 'X exists at A'.

- Source Association Axiom 3 (SAA3)
  $((P \xleftrightarrow{K} Q) \bigwedge (R\ received\ +\{X\ from\ Q\}_K))$
  $\longrightarrow\ ((Q\ said\ X) \bigwedge (Q\ has\ X))$

  SAA3 is presented to make the expression of the message authentication based on the MAC or HMAC method simplified and compact. Its meaning is that Q said X and Q has X if P and Q share K and any principal R received X from Q, which is protected with K based on the MAC or HMAC method.

- Saying Axiom 3 (SA3)
  $((Q\ said\ \{X\}_K\ to\ P) \bigwedge (PK_\psi(P,K))$
  $\longrightarrow\ Q\ said\ X$

  SA3 is presented to reason about the public key encryption used with the 'say' notion. It indicates that Q said X if Q said the X encrypted with K to P and P has K.

4 Analysis with Extensions

In this section, we apply the extended SVO logic to formally verify four security protocols. The first two protocols are the famous MIPv6 security ones while the last two ones were proposed for *Fast Handover for Mobile IPv6* (FMIPv6) [21], which is one of the MIPv6 variants. Especially, with the verification of the last two protocols, we aim at showing that the extended SVO logic can be applied to the MIPv6 variants.

Table 1: Notations for the security protocols for MIPv6 and FMIPv6

Notation	Meaning
$Msg(S,D)$	Msg is sent from S to D, where Msg is a message, and S and D are IPv6 addresses
MN, HA and CN	a mobile node, a home agent, and a corresponding node
HoA, CoA and CNA	MN's home address and care-of address, and a CN's address
$H(M)$	the hash operation on the message M
$SIGN(K,M)$	the digital signature on the message M using the private key K
$HMAC(K,M)$	the HMAC operation on the message M using the shared key K
PU_X and PR_X	X's public key and private key
$CGAP_X$	X's CGA parameters including X's public key

To efficiently express the protocols, we use the notations given in Table 1.

4.1 Analysis on MIPv6 security protocols

Public key based protocols have been proposed for protecting MIPv6 [2, 3, 4, 5, 6, 7] due to the security and efficiency problems in the RR one [1]. Among them, the *Child-proof Authentication for MIPv6* (CAM) protocol was one of the meaningful approaches to introduce how the CGA method can be applied to protect MIPv6 in the initial stage [3] while the *Enhanced Route Optimization for Mobile IPv6* (ERO) protocol was adopted as an alternative standard to the RR protocol (IETF RFC 4866).

Here, we formally verify the correctness of these two protocols with the enhanced SVO logic.

4.1.1 Analysis on the *Child-proof Authentication for MIPv6* (CAM)

As depicted in Figure 2, the CAM protocol provides a simple unilateral authentication based on the CGA method. In this protocol, once receiving a binding update message digitally signed with PR_{MN}, the *CN* firstly authenticate *MN*'s public key by using HoA and $CGAP_{MN}$, and then verify the digital signature included in that message.

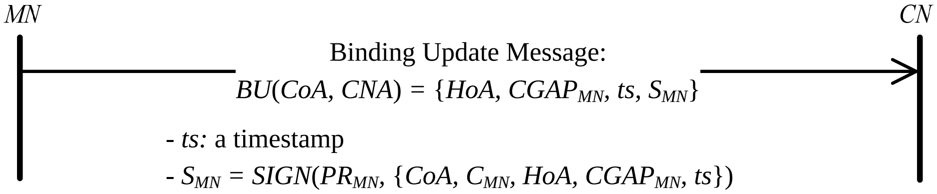

MN
CN
Binding Update Message:
BU(CoA, CNA) = {HoA, CGAP_MN, ts, S_MN}
- ts: a timestamp
- S_MN = SIGN(PR_MN, {CoA, C_MN, HoA, CGAP_MN, ts})

Figure 2: CAM protocol

Here, we precisely analyze this protocol with the enhanced SVO logic. The analysis is composed of the five steps: (i) initial state assumptions, (ii) annotation, (iii) comprehension, (iv) interpretation, and (v) derivation

- **Initial state assumptions**
  As the first step, we define the initial assumptions on about the initial state. Note that the assumption A11 is added because the CGA parameters $CGAP_{MN}$ is related to HoA and PU_{MN}.

  A11. $CN\ believes\ ADP(CGAP_{MN}, HoA, PU_{MN})$
  A12. $CN\ believes\ SV(\lfloor BU \rfloor_{PU_{MN}^{-1}}, PU_{MN}, BU)$,
  where BU is defined in the interpretation step.
  A13. $CN\ believes\ fresh(ts)$

- **Annotation**
  Once the initial assumptions are defined, the CAM protocol is annotated as follows:
  A21. $CN\ received\ (CoA, CNA, HoA, CGAP_{MN}, ts, S_{MN})$

- **Comprehension**
  In this step, we express how the *CN* comprehends the received message as follows:
  A31. $CN\ believes\ CN\ received\ (CoA, CNA, HoA, \langle CGAP_{MN} \rangle_{*CN}, ts, \langle S_{MN} \rangle_{*CN})$

- **Interpretation**
  This step expresses how the *CN* interprets the comprehended message as follows:
  A41. $CN\ believes\ CN\ received\ (CoA, CNA, HoA, \langle CGAP_{MN} \rangle_{*CN}, ts, \langle S_{MN} \rangle_{*CN})$
  $\longrightarrow\ CN\ believes\ CN\ received\ (BU, \langle \lfloor BU \rfloor_{PU_{MN}^{-1}} \rangle_{*CN})$,
  where $BU = (MN@CoA, CNA, MN@HoA, \langle CGAP_{MN} \rangle_{*CN}, ts)$

  From the viewpoint of the CN, $\langle S_{MN} \rangle_{*CN}$ can be interpreted to be $\langle \lfloor BU \rfloor_{PU_{MN}^{-1}} \rangle_{*CN}$. Also, $MN@HoA$ and $MN@CoA$ are added because the binding update message indicates that the *MN* exists at both HoA and CoA.

- **Derivation**

  As the final step, we apply axioms repeatedly until getting the intended results. In order to focus on the authentication reasoning and make the derivation compact and simple, we skip reference to the modus ponens and necessitation rules in our analysis.

  (From A41)
  D1. $CN\ believes\ CN\ received\ (BU, \langle \lfloor BU \rfloor_{PU_{MN}^{-1}} \rangle_{*CN})$
  By A31, A41, and BA1
  D2. $CN\ believes\ CN\ received\ (\langle CGAP_{MN} \rangle_{*CN}\ from\ MN)$
  By D1, RA1, and BA1
  D3. $CN\ believes\ (KA(MN, PU_{MN}, HoA) \bigwedge PK_{\sigma}(MN, PU_{MN}))$
  By D2, A11, MIP1, and BA1
  D4. $CN\ believes\ (OWN(MN, HoA) \bigwedge MN\ said\ BU)$
  By D1, RA1, D3, A12, MIP2, and BA1
  D5. $CN\ believes\ MN\ says\ BU$
  By D4, A13, FA1, NVA, and BA1
  D6. $CN\ believes\ MN\ says\ (MN@HoA, MN@CoA)$
  By D5, SA2, and BA1

- **Discussion**

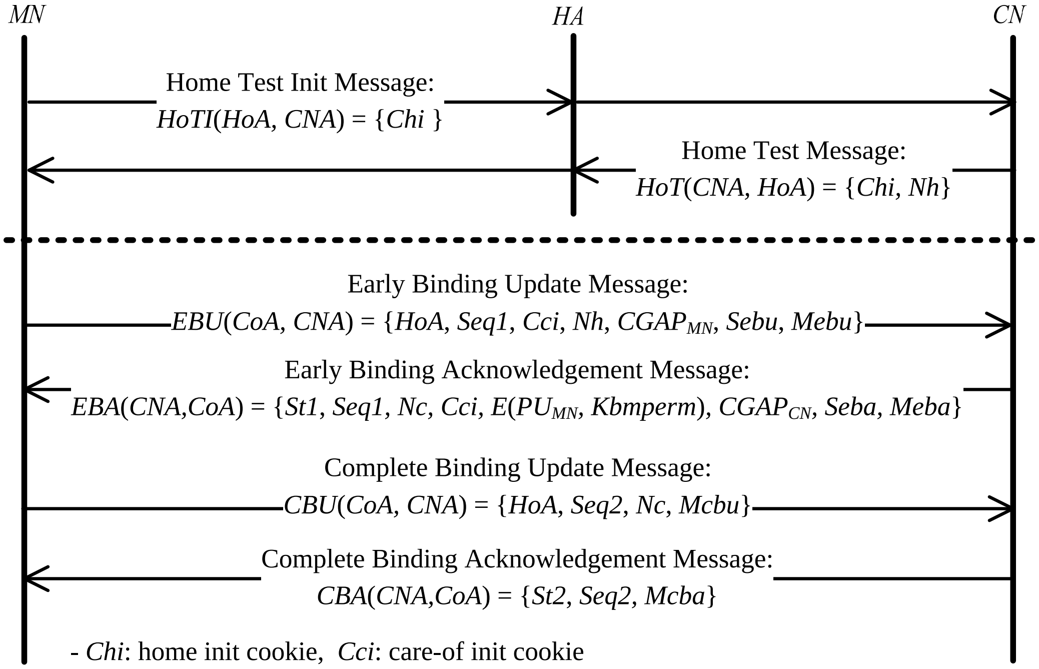
  It is shown from D3 that the *CN* trusts the validity of PU_{MN}, *i.e,* the *MN* is the owner of PU_{MN}. Such a trust allows the *CN* to advance the signature verification with PU_{MN}. Also, the *CN* is sure that the *MN* owns HoA with the help of D4 while believing based on D6 that the *MN* says that it is at both the two addresses, HoA and *CoA*. More importantly, we can conclude from D4 and D6 that the CAM protocol is not vulnerable to the session hijacking attack because they mean that the *MN* and its public key are authenticated by the *CN*. Note that the session hijacking attack can be successful only if its attacker can masquerade as a victim node. Unfortunately, these beliefs cannot convince the *CN* that the *MN* indeed exists at HoA and *CoA* while causing the CAM protocol to be vulnerable to the malicious flooding attack where a malicious but legitimate *MN* can trick its *CN* into redirecting its traffic to a victim node by sending a false binding update message whose new *CoA* is that victim node's address. This is because the *CN* should depend on the *MN* saying it is present at HoA and *CoA*.

  On the other hand, this protocol was formally verified based on BAN logic in [3, 18]. However, due to the limitations of BAN logic, these analysis just provided "*CN believes MN believes BU*" without reasoning about the validity of PU_{MN}.

4.1.2 Analysis on the *Enhanced Route Optimization* (ERO) protocol

The ERO protocol was developed to solve the security and the performance problems of the RR one, then selected as an alternative standard. Especially, it applies the CGA method, the return routeability test, and the early binding update technique to achieve the best security and performance. The ERO protocol is composed of the two phases: the initial one and the subsequent movement one. In the initial phase, the *CN* authenticates the *MN* and its two address while sharing a longterm secret with the *MN*. In the subsequent movement phase, the *CN* uses the shared secret to verify the *MN* and its *CoA* while optimizing the binding update procedure. In this paper, the initial phase, which is shown in Figure 3, is formally verified with SVO logic.

At some point before movement, the *MN* conducts the home test by exchanging the *Home Test Init* ($HoTI$) and *Home Test* (HoT) messages with the *CN*. Once moving to a new network, the *MN* starts the


MN
HA
CN
Home Test Init Message:
$HoTI(HoA, CNA) = \{Chi\}$
Home Test Message:
$HoT(CNA, HoA) = \{Chi, Nh\}$
Early Binding Update Message:
$EBU(CoA, CNA) = \{HoA, Seq1, Cci, Nh, CGAP_{MN}, Sebu, Mebu\}$
Early Binding Acknowledgement Message:
$EBA(CNA,CoA) = \{St1, Seq1, Nc, Cci, E(PU_{MN}, Kbmperm), CGAP_{CN}, Seba, Meba\}$
Complete Binding Update Message:
$CBU(CoA, CNA) = \{HoA, Seq2, Nc, Mcbu\}$
Complete Binding Acknowledgement Message:
$CBA(CNA,CoA) = \{St2, Seq2, Mcba\}$

- *Chi*: home init cookie, *Cci*: care-of init cookie
- *Nh*: a home nonce randomly generated by the *CN*
- *Nc*: a care-of nonce randomly generated by the *CN*
- *SeqX*: The *X*th Sequence number of the binding update message
- *StX*: This value indicates the *X*th result of the binding update
- $K1 = H(Nh|Zero64)$, where *Zero64* is composed of 64 0 bits
- $ebuBody = (CoA, CNA, HoA, Seq1, Cci, CGAP_{MN})$
- $Sebu = SIGN(PR_{MN}, ebuBody)$, $Mebu = HMAC(K1, ebuBody|Sebu)$,
- $ebaBody = (CoA, CNA, St1, Seq1, Nc, Cci, E(PU_{MN}, Kbmperm), CGAP_{CN})$
- $Seba = SIGN(PR_{CN}, ebaBody)$, $Meba = HMAC(K1, ebaBody|Seba)$
- $K2 = H(Nc|Kbmperm)$
- $Mcbu = HMAC(K2, CoA|CNA|HoA|Nc|Seq2)$, $Mcba = HMAC(K2, CoA|CNA|St2|Seq2)$

Figure 3: ERO protocol

early binding update procedure by sending the *Early Binding Update* (*EBU*) message to the *CN*, which then responds with the *Early Binding Acknowledgement* (*EBA*) message. It is worth to note that the *EBU* message is protected with both *Sebu* and *Mebu*. Especially, *Mebu* is computed with *K*1 derived from *Nh*, which means that the *MN* receives the *HoT* message at *HoA*. Moreover, it is used to prevents the DoS attacks on *Sebu*. If the *EBU* message is valid, the *CN* can conclude that the *MN* is at *HoA* and its owner in addition to saying its movement to the new location, *CoA*. While being protected like the *EBU* message, the *EBA* message transmits the care-of nonce *Nc* and the longterm secret *Kbmperm*. *Nc* is used to check the *MN*'s existence at *CoA*, and *Kbmperm* is used to protect the subsequent binding update messages removing public key operations. During the early binding update procedure, in order to minimize the binding update latency, the *MN* starts to send its packets to the *CN* when sending the *EBU* message. Similarly, the *CN*'s packets start to be transmitted when the *EBA* message is sent. Unfortunately, such an optimization can be misused by the attackers because *CoA* is not verified at this point. Even though the amount of the *CN*'s received or sending packets are limited, this security threat remains until the complete binding update procedure is finished.

As the next step, the *MN* and the *CN* conduct the complete binding update procedure by exchanging the *Complete Binding Update* (*CBU*) and *Complete Binding Acknowledgement* (*CBA*) messages, which are protected with $K2$ derived from *Kbmperm* and Nc. If the *CBU* message is valid, the *CN* can be sure of the *MN*'s presence at *CoA* because that message shows the *MN*'s receipt of Nc.

In summary, as the result of this protocol, the *CN* believes that the *MN* is at HoA and *CoA* while sharing with the *MN* a longterm secret, $K2$, which will be used in the subsequent movement phase. In addition, with the help of the CGA method, the *CN* and the *MN* believe each other's address ownership where the *MN*'s address means HoA.

- **Initial state assumptions**:
  We start to analyze the ERO protocol by defining the initial assumptions as follows:
  A11. $CN\ believes\ ADP(CGAP_{MN}, HoA, PU_{MN})$
  A12. $CN\ believes\ SV(\lfloor ebuBody \rfloor_{PU_{MN}^{-1}}, PU_{MN}, ebuBody)$,
  where $ebuBody$ is defined in Comprehension.
  A13. $CN\ believes\ fresh(Nh)$
  A14. $CN\ believes\ RR(Nh, MN, HoA)$
  A15. $MN\ believes\ ADP(CGAP_{CN}, CNA, PU_{CN})$
  A16. $MN\ believes\ SV(\langle \lfloor eb\widehat{aBo}dy \rfloor_{PU_{CN}^{-1}}, PU_{CN}, eb\widehat{aBo}dy)$,
  where $eb\widehat{aBo}dy$ is defined in Comprehension.
  A17. $MN\ believes\ fresh(Seq1)$
  A18. $MN\ believes\ CN\ controls\ St1$
  A19. $MN\ believes\ RR(Seq1, CN, CNA)$
  A1a. $MN\ believes\ CN\ controls\ fresh(MN \xleftrightarrow{K} CN)$
  A1b. $MN\ believes\ PK_{\psi}(MN, PU_{MN})$
  A1c. $MN\ believes\ CN\ controls\ MN \xleftrightarrow{K} CN$
  A1d. $CN\ believes\ MN \xleftrightarrow{K2} CN$
  A1e. $CN\ believes\ fresh(K2)$
  A1f. $CN\ believes\ fresh(Nc)$
  A1g. $CN\ believes\ RR(Nc, MN, CoA)$
  A1h. $MN\ believes\ fresh(Seq2)$
  A1i. $MN\ believes\ CN\ controls\ St2$

  Among the above assumptions, A11 and A15 are added to describe that $CGAP_{MN}$ and $CGAP_{CN}$ are the CGA parameters. Also, A14 and A1g are appended to express that Nh and Nc are used to check if the *MN* is at HoA and *CoA*. These assumptions are applied to the new axioms MIP1 and MIP4.

- **Annotation**
  In this step, the ERO protocol is annotated as follows:
  A21. $CN\ received\ (Chi)$
  A22. $MN\ received\ (Chi, Nh)$
  A23. $CN\ received\ (CoA, CNA, HoA, Seq1, Cci, Nh, CGAP_{MN}, Sebu, Mebu)$
  A24. $MN\ received\ (CoA, CNA, St1, Seq1, Nc, Cci, \{Kbmperm\}_{PU_{MN}}, CGAP_{CN}, Seba, Meba)$
  A25. $CN\ received\ (CoA, CNA, HoA, Seq2, Nc, Mcbu)$
  A26. $MN\ received\ (CoA, CNA, St2, Seq2, Mcba)$

- **Comprehension**
  The annotated protocol is comprehended as follows:
  A31. $CN\ believes\ CN\ received\ (\langle Chi \rangle_{*CN})$

A32. $MN\ believes\ MN\ received\ (Chi, \langle Nh\rangle_{*MN})$

A33. $CN\ believes\ CN\ received\ (ebuBody, \langle Sebu\rangle_{*CN}, \langle Mebu\rangle_{*CN})$,
where $ebuBody = (CoA, CNA, HoA, Seq1, \langle Cci\rangle_{*CN}, Nh, \langle CGAP_{MN}\rangle_{*CN}\ from\ MN)$

A34. $MN\ believes\ MN\ received\ (ebaBody, \{\langle Kbmperm\rangle_{*MN}\}_{PU_{MN}}, \langle Seba\rangle_{*MN}, \langle Meba\rangle_{*MN})$,
where $ebaBody = (CoA, CNA, \langle St1\rangle_{*MN},\ Seq1,\ \langle Nc\rangle_{*MN}, Cci, \langle CGAP_{CN}\rangle_{*MN}\ from\ CN)$

A35. $CN\ believes\ CN\ received\ (CoA, CNA, HoA, Seq2, Nc, \langle Mcbu\rangle_{*CN})$

A36. $MN\ believes\ MN\ received\ (CoA, CNA, \langle St2\rangle_{*MN}, Seq2, \langle Mcba\rangle_{*MN})$

- **Interpretation**

Once comprehended, the ERO protocol is interpreted from the viewpoint of the *CN* and the *MN*. Note that in A43 and A44, the HMAC operations are expresed through the new notation $+\{X_K\}$.

A41. $CN\ believes\ CN\ received\ (ebuBody, \langle Sebu\rangle_{*CN}, \langle Mebu\rangle_{*CN})$
$\longrightarrow\ CN\ believes\ CN\ received\ (ebuBody\ from\ MN, \langle\lfloor ebuBody\rfloor_{PU_{MN}^{-1}}\rangle_{*CN})$

A42. $MN\ believes\ MN\ received\ (ebaBody, \{\langle Kbmperm\rangle_{*MN}\}_{PU_{MN}}, \langle Seba\rangle_{*MN}, \langle Meba\rangle_{*MN})$
$\longrightarrow\ MN\ believes\ MN\ received\ (\widehat{ebaBody}\ from\ CN, \langle\lfloor \widehat{ebaBody}\rfloor_{PU_{CN}^{-1}}\rangle_{*MN})$,
where $\widehat{ebaBody} = (ebaBody, \{MN \overset{\langle K2\rangle_{*MN}}{\longleftrightarrow} CN\}_{PU_{MN}}, fresh(\langle K2\rangle_{*MN}))$

A43. $CN\ believes\ CN\ received\ (CoA, CNA, HoA, Seq2, Nc, \langle Mcbu\rangle_{*CN})$
$\longrightarrow CN\ believes\ CN\ received\ +\{(CoA, CNA, HoA, Seq2, Nc, MN \overset{K2}{\longleftrightarrow} CN)\ from\ MN\ \}_{K2}$

A44. $MN\ believes\ MN\ received\ (CoA, CNA, St2, Seq2, \langle Mcba\rangle_{*MN})$
$\longrightarrow MN\ believes\ MN\ received\ +\{(CoA, CNA, \langle St2\rangle_{*MN}, Seq2)\ from\ CN\ \}_{<K2>_{*MN}}$

Note that we replace $K2$ with *Kbmperm* because $K2$ is derived from *Kbmperm*.

- **Derivation**

(From A41)

D1. $CN\ believes\ CN\ received\ (ebuBody\ from\ MN, \langle\lfloor ebuBody\rfloor_{PU_{MN}^{-1}}\rangle_{*CN})$
By A33, A41, and BA1

D2. $CN\ believes\ CN\ received\ (\langle CGAP_{MN}\rangle_{*CN}\ from\ MN)$
By D1, RA1, and BA1

D3. $CN\ believes\ (KA(MN, PU_{MN}, HoA)\ \bigwedge\ PK_{\sigma}(MN, PU_{MN}))$
By D2, A11, MIP1, and BA1

D4. $CN\ believes\ (OWN(MN, HoA)\ \bigwedge\ MN\ said\ ebuBody)$
By D1, RA1, D3, A12, MIP2, and BA1

D5. $CN\ believes\ MN\ says\ ebuBody$
By D4, A13, FA1, NVA, and BA1

D6. $CN\ believes\ MN\ says\ (HoA, CoA)$
By D5, SA2, and BA1

D7. $CN\ believes\ MN@HoA$
By D5, SA2, A14, MIP4, and BA1

Here, based on D4 and D7, the *CN* can trust that the *MN* is present at *HoA* and its owner. That makes it possible for the *CN* to authenticate the *MN*, thus defending against the session hijacking attack. However, the *CN* still has no belief that the *MN* is at *CoA*. As described above, to minimize the binding update latency, packets start to be exchanged between the *MN* and the *CN* when the early binding update procedure starts. It means that from this point this protocol is vulnerable to the malicious flooding attack until the *MN*'s existence at *CoA* is verified (*i.e.*, the *CBU* message is verified).

(From A42)
D8. $MN\ believes\ MN\ received\ (\widehat{ebaBody}\ from\ CN, \langle \lfloor \widehat{ebaBody} \rfloor_{PU_{CN}^{-1}} \rangle_{*MN})$
By A34, A42, and BA1
D9. $MN\ believes\ (KA(CN, PU_{CN}, CNA) \bigwedge PK\sigma(CN, PU_{CN}))$
By D8, RA1, A15, MIP1 and BA1
D10. $MN\ believes\ (OWN(CN, CNA) \bigwedge CN\ said\ \widehat{ebaBody})$
By D8, RA1, D9, A16, MIP2, and BA1
D11. $MN\ believes\ CN\ says\ \widehat{ebaBody}$
By D10, A17, FA1, NVA, and BA1
D12. $MN\ believes\ \langle St1 \rangle_{*MN}$
By D11, SA2, A18, JA, and BA1
D13. $MN\ believes\ CN@CNA$
By D11, SA2, A19, MIP4, and BA1
D14. $MN\ believes\ fresh(\langle K2 \rangle_{*MN})$
By D11, SA2, A1a, JA, and BA1
D15. $MN\ believes\ MN \overset{\langle K2 \rangle_{*MN}}{\longleftrightarrow} CN$
By D10, SA1, A1b, SA3, D14, NVA, A1c, JA, and BA1

At this point, it is shown from D11 and D12 that the *MN* trusts the *EBA* message as well as the *CN*. Also, D14 and D15 mean that the *MN* believes $K2$ is a good key shared between the *CN* and itself.

(From A43)
D16. $CN\ believes\ CN\ received\ +\{(CoA, CNA, HoA, Seq2, Nc, MN \overset{K2}{\longleftrightarrow} CN)\ from\ MN\ \}_{K2}$
By A35, A43, and BA1
D17. $CN\ believes\ MN\ said\ (CoA, CNA, HoA, Seq2, Nc, MN \overset{K2}{\longleftrightarrow} CN)$
By A1d, D16, SAA3, and BA1
D18. $CN\ believes\ MN\ says\ (CoA, CNA, HoA, Seq2, Nc, MN \overset{K2}{\longleftrightarrow} CN)$
By A1f, D17, FA1, NVA, and BA1
D19. $CN\ believes\ MN\ says\ MN \overset{K2}{\longleftrightarrow} CN$
By D18, SA2, and BA1
D20. $CN\ believes\ MN@CoA$
By D18, SA2, A1g, MIP4, and BA1

Based on D19, the *CN* has the belief that it successfully shares $K2$ with the *MN*. More importantly, D20 shows that the *CN* trusts the *MN* exists at *CoA*. In order words, the ERO protocol is not vulnerable to the malicious flooding attack any more.

(From A44)
D21. $MN\ believes\ MN\ received\ +\{(CoA, CNA, \langle St2 \rangle_{*MN}, Seq2)\ from\ CN\ \}_{<K2>_{*MN}}$
By A36, A44, and BA1
D22. $MN\ believes\ CN\ said\ (CoA, CNA, \langle St2 \rangle_{*MN}, Seq2)$
By D15, D21, SAA3, and BA1
D23. $MN\ believes\ CN\ says\ (CoA, CNA, \langle St2 \rangle_{*MN}, Seq2)$

By A1h, D22, FA1, NVA, and BA1

D24. *MN believes* $\langle St2\rangle_{*MN}$

By D23, SA2, A1i, JA, and BA1

- **Discussion**

  From the above formal analysis, we obtain the following results:

  - It is shown from D4, D7, and D20 that the *CN* believes the *MN* owns *HoA* while being at *HoA* and *CoA*. It means that the ERO protocol can prevent the redirect attacks achieving its important security goal.
  - It is shown from D14, D15, D19, A1d and A1e that *K*2 is a good key shared between the *CN* and the *MN*. Thus, it is enough strong to be used in the subsequent movement phase.

  As a result, we can conclude that the ERO protocol is correct. More importantly, this analysis demonstrates that with the help of the new axioms and notations, we can more precisely reason about the protocol.

4.2 Analysis on FMIPv6 security protocols

FMIPv6 optimizes the handover latency of MIPv6 with the help of link layer triggers and bi-directional tunneling between Access Routers (*AR*) [21]. Similarly to MIPv6, FMIPv6 is required to verify the *MN*'s new *CoA* as well as secure its fast binding update message. Otherwise, it suffers from various attacks such as the SSH, MMF, MiTM and DoS attacks.

As a standard to protect FMIPv6, the Kempf-Koodli's protocol (KKP) was proposed (IETF RFC 5269) [21]. Based on the *SEcure Neighbor Discovery* (SEND) protocol [22], KKP provides the handover key exchange and the message protection without any security infrastructure. However, this protocol is vulnerable to the DoS attack while suffering from high computation cost [19]. In order to improve the drawbacks, You, Hori and Sakurai presented a security protocol (YHSP) [20], which minimizes the public key operations and keeps the KKP's strong security properties with the *Authentication, Authorization, and Accounting* (AAA) infrastructure [23].

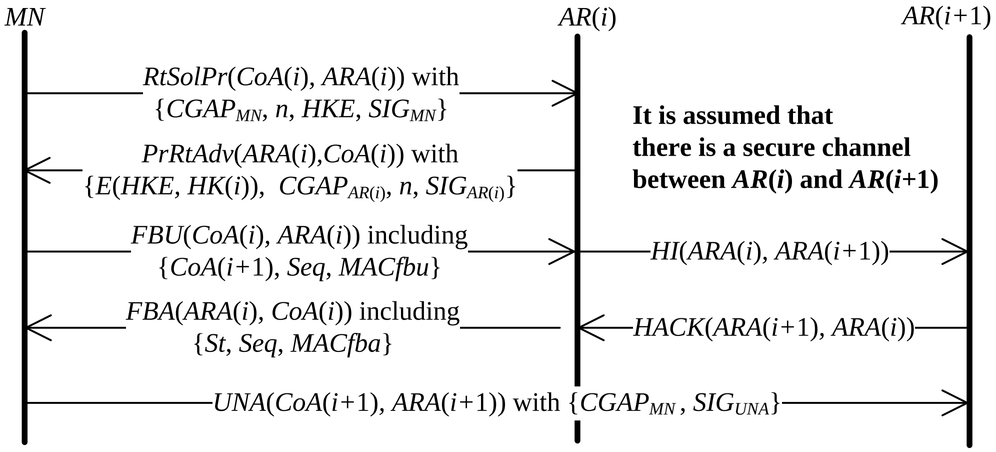
In this section, we formally verify the correctness of the two protocols based on the extended SVO logic.

4.2.1 Analysis on KKP

KKP depends on the SEND protocol, which allows the *MN* and the $AR(i)$ to protect FMIPv6 while sharing a handover key. Especially, through the CGA method which the SEND protocol is based on, the *MN* and the $AR(i)$ can authenticate each other's address and public key.

KKP, illustrated in Figure 4, consists of the handover key negotiation, fast binding update, and new network attachment phases. Note that in KKP every entity (*i.e.*, *MN* and *AR*) is assumed to have a public/private key pair and set its address as a CGA. Also, it is supposed that there is a secure channel between access routers.

Once detecting a link-specific handover event, the *MN* performs the handover key negotiation phase by exchanging the *Router Solicitation for Proxy Advertisement* (*RtSolPr*) and *Proxy Router Advertisement* (*PrRtAdv*) messages with the $AR(i)$. In this phase, the *MN* and the $AR(i)$ establish a handover key, $HK(i)$, through the *MN*'s handover encryption public key HKE. Also, the *RtSolPr* and *PrRtAdv* messages are protected with the digital signatures, SIG_{MN} and $SIG_{AR(i)}$ respectively. In particular, with the help of the CGA method, KKP can verify the *MN*'s and the $AR(i+1)$'s address and public key. After the $HK(i)$ is established, the *MN* conducts the fast binding update phase with the $AR(i)$ by using the


MN
AR(i)
AR(i+1)
RtSolPr(CoA(i), ARA(i)) with
{CGAP_MN, n, HKE, SIG_MN}
PrRtAdv(ARA(i),CoA(i)) with
{E(HKE, HK(i)), CGAP_AR(i), n, SIG_AR(i)}
It is assumed that
there is a secure channel
between AR(i) and AR(i+1)
FBU(CoA(i), ARA(i)) including
{CoA(i+1), Seq, MACfbu}
HI(ARA(i), ARA(i+1))
FBA(ARA(i), CoA(i)) including
{St, Seq, MACfba}
HACK(ARA(i+1), ARA(i))
UNA(CoA(i+1), ARA(i+1)) with {CGAP_MN, SIG_UNA}

- n: nonce, HKE: the MN's public key to be used to encrypt the handover key
- $CoA(i)$: the ith care-of address of MN, $ARA(i)$: the IPv6 address of $AR(i)$
- $SIG_{MN} = SIGN(PR_{MN}, CoA(i)|AR(i)|RtSolPr|CGAP_{MN}|n|\ HKE)$
- $SIG_{AR(i)} = SIGN(PR_{AR(i)}, AR(i)|CoA(i)|PrRtAdv|E(HKE, HK(i)),\ CGAP_{AR(i)}, n)$
- $MACfbu = HMAC(HK(i), CoA(i)|AR(i)|FBU)$
- $MACfba = HMAC(HK(i), AR(i)|CoA(i)|FBA)$
- $SIG_{UNA} = SIGN(PR_{MN}, CoA(i\text{+}1)|AR(i\text{+}1)|UNA|CGAP_{MN})$

Figure 4: Kempf-Koodli's protocol

Fast Binding Update (*FBU*) and *Fast Binding Acknowledge* (*FBA*) messages, which are protected with the handover key, $HK(i)$. If the *FBU* message is valid, the $AR(i)$ starts to redirect the *MN*'s traffic to the $AR(i+1)$ after exchanging the *Handover Initiate* (*HI*) and *Handover Acknowledge* (*HACK*) messages with that new *AR*. As soon as attaching the new network, the *MN* executes the new network attachment phases while informing the $AR(i+1)$ of its attachment. For this goal, it makes use of the *Unsolicited Neighbor Advertisement* (*UNA*) message. In [24], the authors don't mention how the *UNA* message is protected. But, we assume this message is also protected with the SEND protocol. Thus, the *UNA* message is digitally signed with the *MN*'s private key corresponding to its new *CoA*.

Here we verify the correctness of this protocol based on the enhanced SVO logic.

- **Initial state assumptions**:
  As the first step, we define the initial state assumptions as follows:
  A11. $AR(i)\ believes\ ADP(CGAP_{MN}, CoA(i), PU_{MN})$
  A12. $AR(i)\ believes\ SV(\lfloor \widehat{RtSolPr} \rfloor_{PU_{MN}^{-1}}, PU_{MN}, \widehat{RtSolPr})$,
  where $\widehat{RtSolPr}$ is defined in Comprehension.
  A13. $MN\ believes\ ADP(CGAP_{AR(i)}, ARA(i), PU_{AR(i)})$
  A14. $MN\ believes\ SV(\lfloor \widehat{PrRtAdv} \rfloor_{PU_{AR(i)}^{-1}}, PU_{AR(i)}, \widehat{PrRtAdv})$,
  where $\widehat{PrRtAdv}$ is defined in Comprehension.
  A15. $MN\ believes\ fresh(n)$
  A16. $MN\ believes\ RR(n, AR(i), ARA(i))$
  A17. $MN\ believes\ AR(i)\ controls\ fresh(MN \xleftrightarrow{K} AR(i))$
  A18. $MN\ believes\ PK_{\psi}(MN, HKE)$
  A19. $MN\ believes\ AR(i)\ controls\ MN \xleftrightarrow{K} AR(i)$

A1a. $AR(i)\ believes\ MN \overset{HK(i)}{\longleftrightarrow} AR(i)$
A1b. $AR(i)\ believes\ fresh(HK(i))$
A1c. $AR(i)\ believes\ RR(HK(i), MN, CoA(i))$
A1d. $MN\ believes\ fresh(Seq)$
A1e. $MN\ believes\ AR(i)\ controls\ St$
A1f. $AR(i+1)\ believes\ ADP(CGAP_{MN}, CoA(i+1), PU_{MN})$
A1g. $AR(i+1)\ believes\ SV(\lfloor \widehat{UNA} \rfloor_{PU_{MN}^{-1}}, PU_{MN}, \widehat{UNA})$,
  where $\widehat{UNA}$ is defined in Comprehension.

Note that A11, A13, and A1f are the assumptions on the CGP parameters of the MN, the $AR(i)$, and the $AR(i+1)$. They are used to derive the beliefs on the public key ownership and the address ownership. In addition, A16 and A1c are added to express that n and $HK(i)$ play a role of checking the MN's existence at $CoA(i)$ and $CoA(i+1)$.

- **Annotation**
  KKP is annotated as follows:
  A21. $AR(i)\ received\ RtSolPr$
  A22. $MN\ received\ PrRtAdv$
  A23. $AR(i)\ received\ FBU$
  A24. $MN\ received\ FBA$
  A25. $AR(i+1)\ received\ UNA$

- **Comprehension**
  KKP is comprehended as follows:
  A31. $AR(i)\ believes\ AR(i)\ received$
    $(RtSolPr, \langle CGAP_{MN} \rangle_{*AR(i)}, \langle n \rangle_{*AR(i)}, \langle HKE \rangle_{*AR(i)}, \langle SIG_{MN} \rangle_{*AR(i)})$
  A32. $MN\ believes\ MN\ received$,
    $(PrRtAdv, \{\langle HK(i) \rangle_{*MN}\}_{HKE}, \langle CGAP_{AR(i)} \rangle_{*MN}, n, \langle SIG_{AR(i)} \rangle_{*MN})$
  A33. $AR(i)\ believes\ AR(i)\ received\ FBU$,
    where FBU includes $(CoA(i+1), Seq, \langle MACfbu \rangle_{*AR(i)})$
  A34. $MN\ believes\ MN\ received\ FBA$,
    where FBA includes $(\langle St \rangle_{*MN}, Seq, \langle MACfba \rangle_{*MN})$
  A35. $AR(i+1)\ believes\ AR(i+1)\ received$
    $(UNA, \langle CGAP_{MN} \rangle_{*AR(i+1)}, \langle SIG_{UNA} \rangle_{*AR(i+1)})$

- **Interpretation**
  The comprehended version of KKP is interpreted as follows:
  A41. $AR(i)\ believes\ AR(i)\ received$
    $(RtSolPr, \langle CGAP_{MN} \rangle_{*AR(i)}, \langle n \rangle_{*AR(i)}, \langle HKE \rangle_{*AR(i)}, \langle SIG_{MN} \rangle_{*AR(i)})$
  $\longrightarrow\ AR(i)\ believes\ AR(i)\ received\ (\widehat{RtSolPr}\ from\ MN, \langle \lfloor \widehat{RtSolPr} \rfloor_{PU_{MN}^{-1}} \rangle_{*AR(i)})$,
    where $\widehat{RtSolPr} = (RtSolPr, \langle CGAP_{MN} \rangle_{*AR(i)}, \langle n \rangle_{*AR(i)}, PK_{\psi}(MN, \langle HKE \rangle_{*AR(i)}))$
  A42. $MN\ believes\ MN\ received$
    $(PrRtAdv, \{\langle HK(i) \rangle_{*MN}\}_{HKE}, \langle CGAP_{AR(i)} \rangle_{*MN}, n, \langle SIG_{AR(i)} \rangle_{*MN})$
  $\longrightarrow\ MN\ believes\ MN\ received\ (\widehat{PrRtAdv}\ from\ AR(i), \langle \lfloor \widehat{PrRtAdv} \rfloor_{PU_{AR(i)}^{-1}} \rangle_{*MN})$,
    where $\widehat{PrRtAdv} = (PrRtAdv, \{MN \overset{\langle HK(i) \rangle_{*MN}}{\longleftrightarrow} AR(i)\}_{HKE}, fresh(\langle HK(i) \rangle_{*MN}), \langle CGAP_{AR(i)} \rangle_{*MN}, n)$
  A43. $AR(i)\ believes\ AR(i)\ received\ FBU$
  $\longrightarrow\ AR(i)\ believes\ AR(i)\ received\ +\{\widehat{FBU}\ from\ MN\}_{HK(i)}$,

where $\widehat{FBU} = (CoA(i), ARA(i), CoA(i+1), Seq, MN \overset{HK(i)}{\longleftrightarrow} AR(i))$

A44. MN *believes* MN *received* FBA

$\longrightarrow MN$ *believes* MN *received* $+\{\widehat{FBA}$ *from* $AR(i)\}_{HK(i)}$,

where $\widehat{FBA} = (CoA(i), ARA(i), Seq, \langle St\rangle_{*MN})$

A45. $AR(i+1)$ *believes* $AR(i+1)$ *received*

$(UNA, \langle CGAP_{MN}\rangle_{*AR(i+1)}, \langle SIG_{UNA}\rangle_{*AR(i+1)})$

$\longrightarrow AR(i)$ *believes* $AR(i)$ *received* $(\widehat{UNA}$ *from* $MN, \langle\lfloor\widehat{UNA}\rfloor_{PU_{MN}^{-1}}\rangle_{*AR(i+1)})$,

where $\widehat{UNA} = (UNA, \langle CGAP_{MN}\rangle_{*AR(i+1)})$

- **Derivation**

(From A41)

D1. $AR(i)$ *believes* $AR(i)$ *received* $(\widehat{RtSolPr}$ *from* $MN, \langle\lfloor\widehat{RtSolPr}\rfloor_{PU_{MN}^{-1}}\rangle_{*AR(i)})$

By A31, A41, and BA1

D2. $AR(i)$ *believes* $(KA(MN, PU_{MN}, CoA(i)) \bigwedge PK_{\sigma}(MN, PU_{MN}))$

By D1, RA1, A11, MIP1, and BA1

D3. $AR(i)$ *believes* $(OWN(MN, CoA(i)) \bigwedge MN$ *said* $\widehat{RtSolPr})$

By D1, RA1, D2, A12, MIP2, and BA1

D4. $AR(i)$ *believes* MN *said* $(RtSolPr, PK_{\psi}(MN, \langle HKE\rangle_{*AR(i)}))$

By D3, SA1, and BA1

In the above beliefs, D2 shows that the $AR(i)$ trusts the MN's public key and D3 indicates that the $AR(i)$ believes that the MN owns $CoA(i)$. However, because the $RtSolPr$ message does not include any fresh value, we cannot improve D4 anymore. It means that due to this message, KKP is vulnerable to the replay and DoS attacks. More importantly, the $AR(i)$ cannot trust the handover encryption key, HKE even though it believes PU_{MN} is valid. Strictly speaking, this verification cannot be advanced anymore due to the above problems. But, in order to analyze the rest of KKP, we proceed the verification while assuming the $RtSolPr$ message and HKE are authenticated.

(From A42)

D5. MN *believes* MN *received* $(\widehat{PrRtAdv}$ *from* $AR(i), \langle\lfloor\widehat{PrRtAdv}\rfloor_{PU_{AR(i)}^{-1}}\rangle_{*MN})$

By A32, A42, and BA1

D6. MN *believes* $(KA(AR(i), PU_{AR(i)}, ARA(i)) \bigwedge PK_{\sigma}(AR(i), PU_{AR(i)}))$

By D5, RA1, A13, MIP1, and BA1

D7. MN *believes* $(OWN(AR(i), ARA(i)) \bigwedge AR(i)$ *said* $\widehat{PrRtAdv})$

By D5, RA1, D6, A14, MIP2, and BA1

D8. MN *believes* $AR(i)$ *says* $\widehat{PrRtAdv}$

By D7, A15, FA1, NVA, and BA1

D9. MN *believes* $AR(i)@ARA(i)$

By D8, SA2, A16, MIP4, and BA1

D10. MN *believes* $fresh(\langle HK(i)\rangle_{*MN})$

By D8, SA2, A17, JA, and BA1

D11. MN *believes* $MN \overset{\langle HK(i)\rangle_{*MN}}{\longleftrightarrow} AR(i)$

By D7, SA1, A18, SA3, D10, NVA, A19, JA, and BA1

D12. MN *believes* $AR(i)$ *says* $PrRtAdv$

By D8, SA2, and BA1

Unlike the $AR(i)$'s viewpoint, the MN can trust the *PrRtAdv* message from D12. Also, based on D10 and D11, it can be sure that $HK(i)$ is valid.

(From A43)
D13. $AR(i)\ believes\ AR(i)\ received\ +\{\widehat{FBU}\ from\ MN\}_{HK(i)}$
By A33, A43, and BA1
D14. $AR(i)\ believes\ MN\ says\ \widehat{FBU}$
By A1a, D13, SAA3, A1b, FA1, NVA, and BA1
D15. $AR(i)\ believes\ MNsays\ MN \overset{HK(i)}{\longleftrightarrow} AR(i)$
By D14, SA2, and BA1
D16. $AR(i)\ believes\ MN@CoA(i)$
By D14, SA2, A1c, MIP4, and BA1

From the above derivation, D14, D15, and D16 are obtained. While D14 describes the *FBU* message is valid, D15 indicates the *MN*'s trust on $HK(i)$. More importantly, it is shown from D16 that the *MN* is at *CoA*. With such a belief, the $AR(i)$ can start to redirect the *MN*'s traffic to the *MN*'s $CoA(i+1)$ on the $AR(i+1)$'s network.

(From A44)
D17. $MN\ believes\ MN\ received\ +\{\widehat{FBA}\ from\ AR(i)\}_{HK(i)}$
By A34, A44, and BA1
D18. $MN\ believes\ AR(i)\ says\ \widehat{FBA}$
By D11, D17, SAA3, A1d, FA1, NVA, and BA1
D19. $MN\ believes\ \langle St\rangle_{*MN}$
By D18, SA2, A1e, JA, and BA1

(From A45)
D20. $AR(i+1)\ believes\ AR(i+1)\ received\ (\widehat{UNA}\ from\ MN, \langle\lfloor\widehat{UNA}\rfloor_{PU_{MN}^{-1}}\rangle_{*AR(i+1)})$
By A35, A45, and BA1
D21. $AR(i+1)\ believes\ (KA(MN,PU_{MN},CoA(i+1)) \bigwedge PK_{\sigma}(MN,PU_{MN}))$
By D20, RA1, A1f, MIP1, and BA1
D22. $AR(i)\ believes\ (OWN(MN,CoA(i+1)) \bigwedge MN\ said\ \widehat{UNA})$
By D20, RA1, D21, A1g, MIP2, and BA1
D23. $AR(i)\ believes\ MN\ said\ UNA$
By D22, SA1, and BA1

Similar to the *RtSolPr* message, we cannot advance D23 anymore because the *UNA* message does not include any fresh value. Thus, this message also causes KKP to be vulnerable to the replay and DoS attacks.

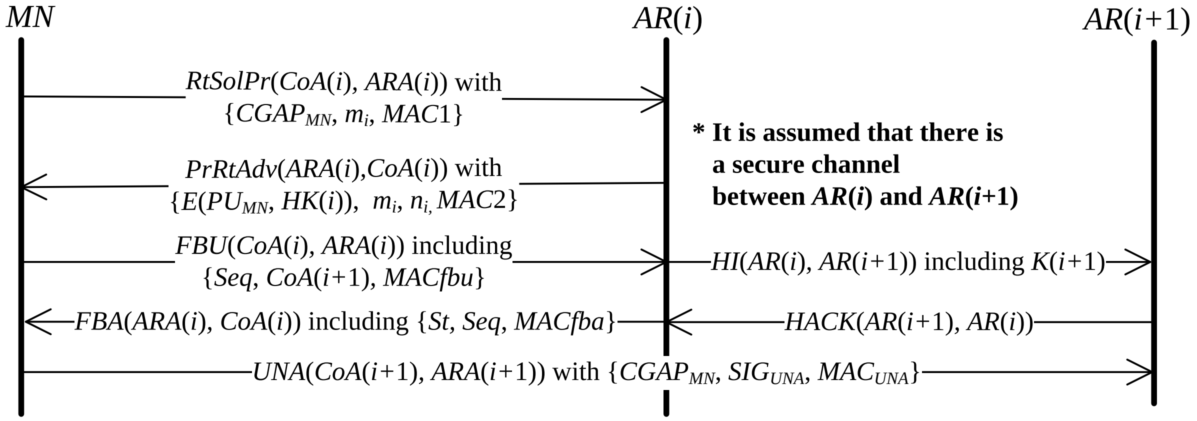
- **Discussion**
As described above, we fail to authenticate the *RtSolPr* and *UNA* messages while not being able to evolve D4 and D23 anymore because those messages are not fresh. That makes this protocol exposed to the replay and DoS attack. Also, from D4, we can just obtain the belief that $AR(i)$ *believes MN said* $PU_{\psi}(MN,HKE)$. This belief cannot guarantee the $AR(i)$ that HKE is strong enough for the handover key exchange. On the other hand, this protocol was formally verified based on BAN logic in [19]. Compared to the verification, we can reason from D2, D3, and D16

that the MN owns PK_{MN} and $CoA(i)$ while being at $CoA(i)$.

4.2.2 Analysis on YHSP

In 2009, You, Hori, and Sakurai introduced a security protocol for FMIPv6, which addresses the drawbacks of KKP [20]. This protocol (called YSHP) minimizes the public key operations as well as prevents the replay and DoS attacks by using the HMAC method, for which the shared secrets between the MN and the AR are used. Especially, it depends on the AAA infrastructure to allow the MN and the AR to share the first message protection secret.

Figure 5 shows YHSP in detail. In addition to the KKP's assumptions, YHSP requires every MN to share the first message protection secret $K(1)$ with the $AR(1)$ based on the AAA infrastructure.


MN
AR(i)
AR(i+1)
RtSolPr(CoA(i), ARA(i)) with {CGAP_MN, m_i, MAC1}
PrRtAdv(ARA(i),CoA(i)) with {E(PU_MN, HK(i)), m_i, n_i, MAC2}
* It is assumed that there is a secure channel between AR(i) and AR(i+1)
FBU(CoA(i), ARA(i)) including {Seq, CoA(i+1), MACfbu}
HI(AR(i), AR(i+1)) including K(i+1)
FBA(ARA(i), CoA(i)) including {St, Seq, MACfba}
HACK(AR(i+1), AR(i))
UNA(CoA(i+1), ARA(i+1)) with {CGAP_MN, SIG_UNA, MAC_UNA}

- m_i: the ith nonce randomly generated by the MN, n_i: the nonce randomly generated by the $AR(i)$
- t: the nonce randomly generated by the $AR(i+1)$, $ARA(i)$: the IPv6 address of $AR(i)$
- $K(i+1) = SHA1(m_i|n_i|HK(i))$, where $i > 1$ and $K(1)$ is initially shared between the MN and the $AR(1)$ during the bootstrapping with the help of the AAA infrastructure
- $MAC1 = HMAC(K(i), RtSolPr|CGAP_{MN}|m_i)$, $MAC2 = HMAC(K(i), PrRtAdv|E(PU_{MN}, HK(i))|m_i|n_i)$
- Seq: It denotes the sequence number for the FBU and FBA messages
- $MACfbu = HMAC(HK(i), CoA(i)|ARA(i)|FBU)$, $MACfba = HMAC(HK(i), ARA(i)|CoA(i)|FBA)$
- $SIGuna = SIGN(PR_{MN}, UNA|K(i+1)|CGAP_{MN})$, $MACuna = HMAC(K(i+1), UNA|CGAP_{MN}|SIGuna)$

Figure 5: You-Hori-Sakurai's protocol

Like KKP, YHSP is composed of the handover key negotiation, fast binding update, and new network attachment phases. When detecting its movement, to start the handover negotiation phase, the MN sends the $RtSolPr$ message to the $AR(i)$, which then replies with the $PrRtAdv$ message. These messages are protected with the HMAC values, $MAC1$ and $MAC2$, instead of the digital signatures. Note that $MAC1$ and $MAC2$ are computed with $K(i)$, which was newly shared between the MN and the $AR(i)$ in the latest handover. Thus, YHSP prevents the replay and DoS attacks while reducing the expensive digital signatures. During the handover negotiation phase, PU_{MN} is verified by $CGAP_{MN}$ and $CoA(i)$, and then is used to help the MN and the $AR(i)$ to negotiate $HK(i)$. The fast binding update phase of YHSP is same as that of KKP except for that $K(i+1)$ is forwarded from the $AR(i+1)$ to the $AR(i)$. That is, $K(i+1)$ is shared between the MN and the $AR(i+1)$ in this phase. When the MN attaches to the new network of the $AR(i+1)$, it conducts the new network attachment phase by sending the UNA message, which is protected with both the digital signature SIG_{UNA} and the HMAC value MAC_{UNA}. Once receiving this message, the $AR(i+1)$ verifies MAC_{UNA} with $K(i+1)$, prior to validating SIG_{UNA}. That makes it for YHSP to prevent the DoS attack. Also, the replay attack can be defended against with the help of the freshness of $K(i+1)$.

Now we start to analyze YHSP with the extended SVO logic.

- **Initial state assumptions**:

  In this step, we define the following assumptions:

  A11. $AR(i)\ \mathit{believes}\ MN \overset{K(i)}{\longleftrightarrow} AR(i)$

  A12. $AR(i)\ \mathit{believes}\ \mathit{fresh}(MN \overset{K(i)}{\longleftrightarrow} AR(i))$

  A13. $AR(i)\ \mathit{believes}\ ADP(CGAP_{MN}, CoA(i), PU_{MN})$

  A14. $MN\ \mathit{believes}\ MN \overset{K(i)}{\longleftrightarrow} AR(i)$

  A15. $MN\ \mathit{believes}\ \mathit{fresh}(m_i)$

  A16. $MN\ \mathit{believes}\ AR(i)\ \mathit{controls}\ \mathit{fresh}(MN \overset{K}{\longleftrightarrow} AR(i))$

  A17. $MN\ \mathit{believes}\ PK_{\psi}(MN, PU_{MN})$

  A18. $MN\ \mathit{believes}\ AR(i)\ \mathit{controls}\ MN \overset{K}{\longleftrightarrow} AR(i)$

  A19. $AR(i)\ \mathit{believes}\ MN \overset{K(i+1)}{\longleftrightarrow} AR(i)$

  A1a. $AR(i)\ \mathit{believes}\ \mathit{fresh}(K(i+1))$

  A1b. $AR(i)\ \mathit{believes}\ EV(K(i+1), PU_{MN}, MN)$

  A1c. $AR(i)\ \mathit{believes}\ RR(K(i+1), MN, CoA(i))$

  A1d. $AR(i)\ \mathit{believes}\ \mathit{fresh}(Seq)$

  A1e. $AR(i)\ \mathit{believes}\ AR(i)\ \mathit{controls}\ St$

  A1f. $AR(i+1)\ \mathit{believes}\ MN \overset{K(i+1)}{\longleftrightarrow} AR(i+1)$

  A1g. $AR(i+1)\ \mathit{believes}\ \mathit{fresh}(K(i+1))$

  A1h. $AR(i+1)\ \mathit{believes}\ ADP(CGAP_{MN}, CoA(i+1), PU_{MN})$

  A1i. $AR(i+1)\ \mathit{believes}\ SV(\lfloor UNA, \langle CGAP_{MN}\rangle_{*AR(i+1)}, MN \overset{K(i+1)}{\longleftrightarrow} AR(i+1)\rfloor_{PU_{MN}^{-1}},$

  $PU_{MN}, (UNA, \langle CGAP_{MN}\rangle_{*AR(i+1)}, MN \overset{K(i+1)}{\longleftrightarrow} AR(i+1)))$

  We add A11, A12, and A14 because $K(i)$ is assumed to be newly shared between the MN and the $AR(i)$ in the latest handover. Also, A13 and A1h are inserted to express the CGA parameters. Especially, A1b and A1c are appended to describe that $K(i+1)$ is used to check if the MN owns PU_{MN} and exists at $CoA(i)$ respectively.

- **Annotation**

  Now, we annotate YHSP as follows:

  A21. $AR(i)\ \mathit{received}\ RtSolPr$

  A22. $MN\ \mathit{received}\ PrRtAdv$

  A23. $AR(i)\ \mathit{received}\ FBU$

  A24. $MN\ \mathit{received}\ FBA$

  A25. $AR(i+1)\ \mathit{received}\ UNA$

- **Comprehension**

  In this step, YHSP is comprehended as follows:

  A31. $AR(i)\ \mathit{believes}\ AR(i)\ \mathit{received}$

  $(RtSolPr, \langle CGAP_{MN}\rangle_{*AR(i)}, \langle m_i\rangle_{*AR(i)}, \langle MAC1\rangle_{*AR(i)})$

  A32. $MN\ \mathit{believes}\ MN\ \mathit{received},$

  $(PrRtAdv, \{\langle HK(i)\rangle_{*MN}\}_{PU_{MN}}, m_i, \langle n_i\rangle_{*MN}, \langle MAC2\rangle_{*MN})$

  A33. $AR(i)\ \mathit{believes}\ AR(i)\ \mathit{received}\ FBU,$

  where FBU includes $(CoA(i+1), Seq, \langle MACfbu\rangle_{*AR(i)})$

A34. *MN believes MN received FBA*,
  where FBA includes $(\langle St\rangle_{*MN}, Seq, \langle MACfba\rangle_{*MN})$
A35. $AR(i+1)$ *believes* $AR(i+1)$ *received*
  $(UNA, \langle CGAP_{MN}\rangle_{*AR(i+1)}, \langle SIG_{UNA}\rangle_{*AR(i+1)}, \langle MAC_{UNA}\rangle_{*AR(i+1)})$

- **Interpretation**

In this step, we make the following interpretation of YHSP:

A41. $AR(i)$ *believes* $AR(i)$ *received*
  $(RtSolPr, \langle CGAP_{MN}\rangle_{*AR(i)}, \langle m_i\rangle_{*AR(i)}, \langle MAC1\rangle_{*AR(i)})$
$\longrightarrow$ $AR(i)$ *believes* $AR(i)$ *received* $+\{\widehat{RtSolPr}\ from\ MN\}_{K(i)}$,
  where $\widehat{RtSolPr} = (RtSolPr, K(i), \langle CGAP_{MN}\rangle_{*AR(i)}, \langle m_i\rangle_{*AR(i)})$
A42. *MN believes MN received*,
  $(PrRtAdv, \{\langle HK(i)\rangle_{*MN}\}_{PU_{MN}}, m_i, \langle n_i\rangle_{*MN}, \langle MAC2\rangle_{*MN})$
$\longrightarrow$ *MN believes MN received* $+\{\widehat{PrRtAdv}\ from\ MN\}_{K(i)}$,
  where $\widehat{PrRtAdv} = (PrRtAdv, \{MN \overset{\langle K(i+1)\rangle_{*MN}}{\longleftrightarrow} AR(i)\}_{PU_{MN}},$
    $fresh(\langle K(i+1)\rangle_{*MN}), m_i, \langle n_i\rangle_{*MN}, \langle CGAP_{AR(i)}\rangle_{*MN})$
A43. $AR(i)$ *believes* $AR(i)$ *received FBU*
$\longrightarrow$ $AR(i)$ *believes* $AR(i)$ *received* $+\{\widehat{FBU}\ from\ MN\}_{K(i+1)}$,
  where $\widehat{FBU} = (CoA(i), ARA(i), CoA(i+1), Seq, MN \overset{K(i+1)}{\longleftrightarrow} AR(i))$
A44. *MN believes MN received FBA*
$\longrightarrow$ *MN believes MN received* $+\{\widehat{FBA}\ from\ AR(i)\}_{K(i+1)}$,
  where $\widehat{FBA} = (CoA(i), ARA(i), Seq, \langle St\rangle_{*MN})$
A45. $AR(i+1)$ *believes* $AR(i+1)$ *received*
  $(UNA, \langle CGAP_{MN}\rangle_{*AR(i+1)}, \langle SIG_{UNA}\rangle_{*AR(i+1)}, \langle MAC_{UNA}\rangle_{*AR(i+1)})$
$\longrightarrow$ $AR(i)$ *believes* $AR(i)$ *received* $+\{\widehat{UNA}\ from\ MN\}_{K(i+1)}$,
  where $\widehat{UNA} = ((UNA, \langle CGAP_{MN}\rangle_{*AR(i+1)})\ from\ MN,$
    $\langle \lfloor (UNA, \langle CGAP_{MN}\rangle_{*AR(i+1)}, MN \overset{K(i+1)}{\longleftrightarrow} AR(i+1)) \rfloor_{PU_{MN}^{-1}} \rangle_{*AR(i+1)})$

- **Derivation**

(From A41)
D1. $AR(i)$ *believes* $AR(i)$ *received* $+\{\widehat{RtSolPr}\ from\ MN\}_{K(i)}$
  By A31, A41, and BA1
D2. $AR(i)$ *believes MN says* $\widehat{RtSolPr}$
  By D1, A11, SAA3, A12, FA1, NVA, and BA1
D3. $AR(i)$ *believes MN says RtSolPr*
  By D2, SA2, and BA1
D4. $AR(i)$ *believes* $(KA(MN, PU_{MN}, CoA(i)) \bigwedge PK_{\psi}(MN, PU_{MN}))$
  By D1, RA1, A13, MIP1, and BA1

In the above derivation, D3 shows that the *RtSolPr* message is authenticated, thus not being misused for the replay and DoS attacks. Also, from D4, we can see that the $AR(i)$ trusts the *MN*'s public key PU_{MN}. With this belief, the *MN* encrypts $HK(i)$ with PU_{MN}.

(From A42)
D5. *MN believes MN received* $+\{\widehat{PrRtAdv}\ from\ AR(i)\}_{K(i)}$

By A32, A42, and BA1

D6. $MN\ believes\ AR(i)\ says\ \widehat{PrRtAdv}$

By D5, A14, SAA3, A15, FA1, NVA, and BA1

D7. $MN\ believes\ AR(i)\ says\ PrRtAdv$

By D6, SA2, and BA1

D8. $MN\ believes\ fresh(\langle K(i+1)\rangle_{*MN})$

By D6, SA2, A16, JA, and BA1

D9. $MN\ believes\ MN \overset{\langle K(i+1)\rangle_{*MN}}{\longleftrightarrow} AR(i)$

By D5, A14, SAA3, SA1, A17, SA3, D8, FA1, A18, JA, and BA1

Here, D7 means that the *MN* authenticates the *PrRtAdv* message. From D8 and D9, it is shown that the *MN* obtains the belief on the new message protection secret $K(i+1)$. This belief plays an important role of allowing $K(i+1)$ to be used between the *MN* and the $AR(i+1)$.

(From A43)

D10. $AR(i)\ believes\ AR(i)\ received\ +\{\widehat{FBU}\ from\ MN\}_{K(i+1)}$

By A33, A43, and BA1

D11. $AR(i)\ believes\ MN\ says\ \widehat{FBU}$

By D10, A19, SAA3, A1a, FA1, NVA, and BA1

D12. $AR(i)\ believes\ MN\ says\ MN \overset{K(i+1)}{\longleftrightarrow} AR(i)$

By D11, SA2, and BA1

D13. $AR(i)\ believes\ OWN(MN, CoA(i))$

By D4, D12, A1b, MIP3, BA1

D14. $AR(i)\ believes\ MN@CoA(i)$

By A1c, D12, MIP4, BA1

In the above derivation, while D11 means that the $AR(i)$ authenticates the FBU message, D12 shows that the $AR(i)$ believes that $K(i+1)$ is well shared between the *MN* and itself. More importantly, D13 and D14 demonstrate that the $AR(i)$ believes that the *MN* owns $CoA(i)$ and exists at that address. These beliefs are enough strong to trigger the $AR(i+1)$ to forward the *MN*'s traffic to the $AR(i+1)$ while proceeding the next step.

(From A44)

D15. $MN\ believes\ MN\ received\ +\{\widehat{FBA}\ from\ AR(i)\}_{K(i+1)}$

By A34, A44, and BA1

D16. $MN\ believes\ AR(i)\ says\ \widehat{FBA}$

By D15, D9, SAA3, A1d, FA1, NVA, and BA1

D17. $MN\ believes\ \langle St\rangle_{*MN}$

By D16, SA2, A1e, JA, and BA1

(From A45)

D18. $AR(i+1)\ believes\ AR(i+1)\ received\ +\{\widehat{UNA}\ from\ MN\}_{K(i+1)}$

By A35, A45, and BA1

D19. $AR(i+1)\ believes\ MN\ says\ \widehat{UNA}$

By D18, A1f, SAA3, A1g, FA1, FA2, FA1, NVA, and BA1

D20. $AR(i+1)\ believes\ MN\ says\ UNA$

By D19, SA2, and BA1

D21. $AR(i+1)$ *believes* $(KA(MN,PU_{MN},CoA(i+1)) \bigwedge PK_{\sigma}(MN,PU_{MN}))$
By D18, RA1, A1h, MIP1, and BA1

D22. $AR(i+1)$ *believes* $(OWN(MN,CoA(i+1)) \bigwedge$
MN *said* $(UNA, \langle CGAP_{MN}\rangle_{*AR(i+1)}, MN \overset{K(i+1)}{\longleftrightarrow} AR(i+1))$
By D18, RA1, D21, A1i, MIP2, and BA1

D23. $AR(i+1)$ *believes* MN *says* UNA
By D22, A1g, FA1, NVA, SA2, and BA1

D24. $AR(i+1)$ *believes* MN *says* $MN \overset{K(i+1)}{\longleftrightarrow} AR(i+1)$
By D22, A1g, FA1, NVA, SA2, and BA1

D20 and D23 show that the $AR(i+1)$ authenticates the UNA message, and D22 demonstrates that the MN owns $CoA(i+1)$. Moreover, according to D24, the $AR(i+1)$ trusts $K(i+1)$ as well as the MN sending the FBU message at $CoA(i)$. That makes the $AR(i+1)$ believe that the MN has just attached to its network and exists at $CoA(i+1)$.

- **Discussion**
  From the above formal analysis, we can provide the following results:
  - It is shown from D3, D7, D11, D16, D20, and D23 that all the FMIPv6 messages are authenticated. Especially, we can see that unlike KKP, the *RtSolPr* and UNA messages are trusted by the $AR(i)$ and the $AR(i+1)$ respectively. Thus, these messages are not misused for the replay and DoS attacks anymore.
  - It is shown from D8, D9, D12, and D24 that $K(i+1)$ is a good key shared among the MN, the $AR(i)$, and the $AR(i+1)$. In addition, D8 and D9 indicate that the MN believes the handover key $HK(i)$ because it is used to derive $K(i+1)$. Moreover, the assumptions A16 and A18 can be applied to obtain this belief as done for $K(i+1)$. As a result, YSHP provides the strong handover key and message protection secret.
  - D14 gives the $AR(i)$ the belief that the MN is present at $CoA(i)$. Note that D11 is not enough for the $AR(i)$ to proceed the rest steps because of not guaranteeing the MN's presence at $CoA(i+1)$. However, such a guarantee is not available in FMIPv6, which sacrifices security for efficiency. Thus, D14 plays an important role in supplementing D11 while triggering the $AR(i)$ to redirect the MN's traffic the $AR(i+1)$.

As a result, it can be concluded from the above analysis that YHSP is correct while improving the drawbacks of KKP.

5 Conclusion

In this paper, we extended SVO logic to achieve the true formal verification on the MIPv6 security protocols. For this extension, we defined the new notations and axioms, which support the new security features typically adopted by the MIPv6 security protocols. The proposed logic was applied for formally analyzing the four security protocols, *i.e.*, CAM, ERO, KKP, and YHSP, while showing its effectiveness in precisely reasoning about their security.

References

[1] D. Johnson, C. Perkins, and J. Arkko, "Mobility support in IPv6," IETF RFC 3775, June 2004, http://www.ietf.org/rfc/rfc3775.txt.

[2] J. Arkko, C. Vogt, and W. Haddad, “Enhanced route optimization for Mobile IPv6,” IETF RFC 4866, May 2007, http://www.ietf.org/rfc/rfc4866.txt.

[3] G. O’Shea and M. Roe, “Child-proof authentication for MIPv6 (CAM),” *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 2, pp. 4–8, April 2001.

[4] G. Montenegro and C. Castelluccia, “Crypto-based identifiers (CBIDs): Concepts and Applications,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 7, no. 1, pp. 97–127, February 2004.

[5] S. Okazaki, A. Desai, C. Gentry, J. Kempf, A. Silverberg, and Y. L. Yin, “Enhanced route optimization for Mobile IPv6,” IETF Internet-draft, October 2002, http://tools.ietf.org/html/draft-okazaki-mobileip-abk-01.

[6] T. Aura, “Cryptographically generated addresses (CGA),” IETF RFC 3972, March 2005, http://www.ietf.org/rfc/rfc3972.txt.

[7] I. You, “Improving the CGA-OMIPv6 protocol for low-power mobile nodes,” in *Proc. of the 6th International Conference on Computational Science and Its Applications (ICCSA’06), Glasgow, UK, LNCS*, vol. 3938. Springer-Verlag, May 2006, pp. 336–343.

[8] R. H. Deng, J. Zhou, and F. Bao, “Defending against redirect attacks in Mobile IP,” in *Proc. of the 9th ACM Conference on Computer and Communications Security (ACM CCS’02), Washington, DC, USA*. ACM, November 2002, pp. 59–67.

[9] L. Ma and J. J. P. Tsai, *Formal Verification Techniques for Computer Communication Security Protocols*. World Scientific, January 2002, pp. 23–46.

[10] C. A. Meadows, “Formal verification of cryptographic protocols: A survey,” in *Proc. of the 4th International Conference on the Theory and Applications of Cryptology: Advances in Cryptologythe (ASIACRYPT’94), Wollongong, Australia, LNCS*, vol. 917. Springer-Verlag, November-December 1994, pp. 133–150.

[11] ——, “Formal methods for cryptographic protocols analysis: Emerging issues and trends,” *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 1, pp. 44–54, January 2003.

[12] P. F. Syverson and P. C. V. Oorschot, “On unifying some cryptographic protocol logics,” in *Proc. of the 1994 IEEE Symposium on Security and Privacy (S&P’94), Oakland, California, USA*. IEEE, August 1994, pp. 14—28.

[13] P. Syverson and I. Cervesato, “The logic of authentication protocols,” *Foundations of Security Analysis and Design, Bertinoro, LNCS*, vol. 2171, pp. 63–136, 2001.

[14] M. Burrows, M. Abadi, and R. Needham, “A logic of authentication,” *ACM Transaction on Computer Systems*, vol. 8, no. 1, pp. 18–36, February 1990.

[15] L. Gong, R. Needham, and R. Yahalom, “Reasoning about belief in cryptographic protocols,” in *Proc. of 2005 IEEE Symposium on Security and Privacy (S&P’04), Berkeley, California, USA*. IEEE, May 1990, pp. 234—248.

[16] M. Abadi and M. R. Tuttle, “A semantics for a logic of authentication,” in *Proc. of the 10th Annual ACM Symposium on Principles of Distributed Computing (PODC’91), Montreal, Quebec, Canada*. ACM, August 1991, pp. 201—216.

[17] P. van Oorschot, “Extending cryptographic logics of belief to key agreement protocols,” in *Proc. of the 1st ACM Conference on Computer and Communications Security (CCS’93), Fairfax, Virginia, USA*. ACM, November 1993, pp. 233—243.

[18] J. xin Li, J. peng Huai, Q. Li, and X. xian Li, “Towards security analysis to binding update protocol in Mobile IPv6 with formal method,” in *Proc. of the 1st International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2005), Wuhan, China, LNCS*, vol. 3794. Springer-Verlag, December 2005, pp. 1073–1080.

[19] I. You, K. Sakurai, and Y. Hori, “A security analysis on Kempf-Koodli’s security scheme for Fast Mobile IPv6,” *IEICE Transaction on Communications*, vol. E92-B, no. 06, pp. 2287–2290, June 2009.

[20] ——, “An enhanced security protocol for Fast Mobile IPv6,” *IEICE Transaction on Information & Systems*, vol. E92-D, no. 10, pp. 1979–1982, October 2009.

[21] R. Koodli, “Mobile IPv6 Fast Handovers,” IETF RFC 5268, June 2008, http://www.ietf.org/rfc/rfc5268.txt.

[22] J. Arkko, J. Kempf, B. Zill, and P. Nikander, “SEcure Neighbor Discovery (SEND),” IETF RFC 3971, March 2005, http://www.ietf.org/rfc/rfc3971.txt.

[23] C. E. Perkins and P. R. Calhoun, “Authentication, Authorization, and Accounting (AAA) registration keys

for Mobile IPv4," IETF RFC 3957, March 2005, http://www.ietf.org/rfc/rfc3957.txt.

[24] J. Kempf and R. Koodli, "Distributing a symmetric FMIPv6 handover key using SEND," IETF RFC 5269, June 2008, http://www.ietf.org/rfc/rfc5269.txt.

Ilsun You received his M.S. and Ph.D. degrees in Computer Science from Dankook University, Seoul, South Korea in 1997 and 2002, respectively. From 1997 to 2004, he worked for the THINmultimedia Inc., Internet Security Co., Ltd. and Hanjo Engineering Co., Ltd. as a Research Engineer. Since March 2005, he has been an Assistant Professor in the School of Information Science at the Korean Bible University, South Korea. Prof. You has served or is currently serving on the organizing or program committees of international conferences and workshops such as IMIS, CISIS, MobiWorld, MIST and so forth. He is the EiC of Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA). He is in the editorial board for International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC), Computing and Informatics (CAI), Journal of Computer Systems, Networks, and Communications (IJSH) and Journal of Korean Society for Internet Information (KSII). His main research interests include Internet security, authentication, access control, MIPv6 and ubiquitous computing.

Yoshiaki Hori received B.E., M.E., and D.E. degrees from Kyushu Institute of Technology, Iizuka, Japan in 1992, 1994, and 2002, respectively. From 1994 to 2003, he was a Research Associate in Common Technical Courses, Kyushu Institute of Design, Fukuoka. From 2003 to 2004, he was a Research Associate in the Department of Art and Information Design, Kyushu University, Fukuoka. Since March 2004, he has been an Associate Professor in the Department of Computer Science and Communication Engineering, Kyushu University. He has been an Associate Professor in the Department of Informatics, Kyushu University. His research interests include network security, network architecture, and performance evaluation of network protocols on various networks. He is a member of IEEE, ACM, and IPSJ.

Kouichi Sakurai received the B.S. degree in mathematics from the Faculty of Science, Kyushu University and the M.S. degree in applied science from the Faculty of Engineering, Kyushu University in 1986 and 1988 respectively. He had been engaged in the research and development on cryptography and information security at the Computer and Information Systems Laboratory at Mitsubishi Electric Corporation from 1988 to 1994. He received D.E. degree from the Faculty of Engineering, Kyushu University in 1993. Since 1994 he has been working for the Department of Computer Science of Kyushu University as Associate Professor, and now he is Full Professor from 2002. His current research interests are in cryptography and information security. Dr. Sakurai is a member of IEICE, IPSJ, Mathematical Society of Japan, IEEE, ACM and the International Association for Cryptologic Research.