Fast and Automatic Verification of Authentication and Key Exchange Protocols

Haruki Ota, Shinsaku Kiyomoto and Toshiaki Tanaka
Information Security Laboratory
KDDI R&D Laboratories, Inc.
Saitama, Japan
{*haruki, kiyomoto, toshi*}*@kddilabs.jp*

***Abstract*—Security in a peer-to-peer (P2P) system is not considered, although it has many potential threats. In order to make the whole P2P system secure, various security functions require to be taken into consideration for these threats, respectively. We take up authentication and key exchange protocols as a target of the security functions in the P2P system. These protocols can bear one duty in order to realize the secure P2P system. It is preferable for authentication and key exchange protocols to be verified automatically and rapidly in accordance with security requirements. In order to meet this requirement, we proposed the security verification method for the aforementioned protocols based on Bellare et al.'s model and showed the verification points of security properties to verify their security efficiently. However, there are three weaknesses in the aforementioned paper. In this paper, (1) we describe the relations of the six verification points, (2) explain how the proposed method verifies the aforementioned protocols by providing one example and (3) show the validity of the proposed method by verifying the security of 87 authentication and key exchange protocols that were generated automatically.**

***Keywords*-security verification method; authentication and key exchange protocols; verification points;**

I. Introduction

A. Motivation

Recently, a peer-to-peer (P2P) system, which is one of the distributed network architectures, has been developed. However, the security in the P2P system is not considered, although it has potential threats, such as viruses and worms, illegal uses of data with copyrights, impersonation, privacy issues and unauthorized access. For these threats, the security functions, such as virus detection techniques, digital rights management, cryptographic protocols, privacy protection mechanisms and intrusion detection techniques, require to be taken into consideration, respectively, in order to make the whole P2P system secure. Then, we take up the cryptographic protocols, in particular, authentication and key exchange protocols as a target of the security functions in the P2P system. In the authentication protocol, pair of users can communicate with each other, while each user knows who his/her communication partner is. In the key exchange protocol, they can send and receive secret data over an unreliable channel. Therefore, these protocols can bear one duty in order to realize the secure P2P system.

For a considerable period, existing authentication and key exchange protocols were designed by trial and error, based on the designer's understanding of security and cryptographic techniques. Therefore, it is vital to be able to deal with compromised protocols quickly. However, the process of specialists designing authentication and key exchange protocols is a time-consuming one and designing a new protocol or modifying an existing protocol and verifying its security are a lengthy process. As a result, there were neither the methods to evaluate the authentication and key exchange protocols formally nor the mechanisms to deal with compromised protocols quickly.

B. Related Work

Two different types of methods have been proposed as ways of verifying the security of authentication and key exchange protocols: those based on a computational complexity approach and those based on formal verification. As methods based on the computational complexity approach, Bellare, Pointcheval and Rogaway introduced the first indistinguishability-based formal model of security for authentication and key exchange protocols [1], [2], [3]. Specifically, Bellare and Rogaway first proposed 2-party mutual authentication and authenticated key exchange protocols in 1993 [1], and subsequently extended this to a 3-party setting via the key distribution center with respect to key exchange protocols in 1995 [2]. In 2000, Bellare, Pointcheval and Rogaway proposed provably secure password-based key exchange and authenticated key exchange protocols, based on the Bellare-Rogaway model [3]. Bellare et al. formulated models that were secure against an off-line dictionary attack and forward secrecy. Hereinafter, we call the model proposed in [1], [2], [3] the "BPR model". The BPR model became the basis of a considerable number of subsequent research studies in this area, such as those that investigated a simulation paradigm [4] and a universally composable framework [5]. However, the problem remained that the security of the protocols still needed to be proved. That is, there was not the automatic verification method based on the BPR model since it is very difficult to implement the notations of the provable security in the BPR model.

On the other hand, methods based on formal verification are classified into the following: methods based on state machine approaches (e.g., the Dolev-Yao model [6]), methods using model checkers (e.g., FDR (Failures Divergences Refinement)/CSP (Communicating Sequential Processes) [7]), methods using algebraic systems (e.g., spi calculus [8]), methods based on modal logic (e.g., BAN (Burrows-Abadi-Needham) logic [9]) and methods based on inductive approaches (e.g., Isabelle/HOL (Higher Order Logic) [10]). However, these methods are less than optimal as it takes a considerable amount of time to verify the security of protocols and/or they cannot always verify the security of protocols automatically.

In order to resolve the aforementioned problems, we proposed a security verification method for authentication and key exchange protocols based on the BPR model [11], [12], [13]. We generalized the process of the security proofs based on the BPR model to implement it as a tool. In particular, we showed the complete verification points of security properties for authentication and key exchange protocols so that the security of each protocol could be verified rapidly and automatically [13]. The verification points have the characteristic that the authentication and key exchange protocols are determined to be secure if they are satisfied with at least one verification point item of the security property. However, there are the following three weaknesses in [13].

1) The relations of the verification points are not clear.
2) It is not clear how the proposed method verifies the authentication and key exchange protocols.
3) The verification results for concrete protocols are not shown using the proposed method.

C. Contributions

In this paper, we provide the following contributions in order to improve the aforementioned weaknesses.

1) We describe the relations of the six verification points by considering the attack models and the security targets.
2) We explain how the proposed method verifies the aforementioned protocols by providing one verification example, which is satisfied with the six security properties.
3) We show the validity of the proposed method by verifying the security of the concrete authentication and key exchange protocols and confirming the verification results and verification time.

D. Organization

The rest of this paper is organized as follows. We introduce the BPR model in Section II. We review the proposed security verification method for authentication and key exchange protocols and describe the relations of the verification points in Section III. We explain the verification example and the verification results using the proposed method in Section IV. Our conclusions are presented in Section V and we present detailed tables of the verification points for the aforementioned protocols in Appendix.

II. BPR Model

This section introduces the security properties of the authentication and key exchange protocols in the BPR model.

In the BPR model, Bellare et al. introduced new notions of security: “matching conversation” of the authentication protocol and “semantic security” of the key exchange protocol [1]. They formulated the following security properties from real attacks, which are shown in brackets, for each notion in accordance with the security requirements.

- Matching conversation (MC) [1]
  In an authentication protocol, an adversary cannot alter messages, send other messages, intercept messages or deliver messages out of order.
  - Security against an impersonation attack (MC-SIA) [1]
    An adversary cannot break an authentication protocol even when he/she controls all communications between parties. [Impersonation attack]
- Semantic security (SS) [1]
  In a key exchange protocol, an adversary cannot distinguish between the session key and random session key.
  - Security against a passive attack (SS-SPA) [1], [2]
    An adversary cannot break a key exchange protocol even when he/she eavesdrops on all communications between parties. [Eavesdropping attack]
  - Security against an active attack (SS-SAA) [1], [2]
    An adversary cannot break a key exchange protocol even when he/she controls all communications between parties. [Active attack (e.g., replay attack, man-in-the-middle attack and so on)]
  - Known key security (SS-KKS) [1], [2]
    An adversary cannot obtain a target session key even when he/she obtains session keys in other sessions. [Known key attack]
  - Weak forward secrecy (SS-WFS) [2], [3]
    An adversary cannot obtain the past session key even when he/she obtains long-lived keys such as the secret keys used in secret key encryption, passwords or private keys used in public key encryption. [Corruption attack]
- Common item
  - Resistance to an off-line dictionary attack (RODA) [3]
    An adversary cannot search for a password of a party that corresponds to the recorded communication off-line from the dictionary.
    [Off-line dictionary attack]

III. Security Verification Method

This section reviews the proposed security verification method for authentication and key exchange protocols based on the BPR model.

A. Procedure

This subsection describes the procedure of the proposed method.

The verification program (VP) verifies the security of the authentication and key exchange protocols in the following manner.

1) The VP enumerates all cryptographic primitives and data used in the authentication and key exchange protocols. Principal cryptographic primitives are classified as functions that are equivalent to the following definitions.
   - Secret key encryption (SKE)
     Function for the purpose of encryption using the pre-shared key.
   - Encryption using password (EPW)
     Function for the purpose of encryption using the pre-shared password.
   - Public key encryption (PKE)
     Function for the purpose of encryption using the public key.
   - Diffie-Hellman family (DH)
     Function for the purpose of key exchange using the Diffie-Hellman method.
   - Digital signature scheme (SIG)
     Function for the purpose of generating the signature using the signing key.
   - Hash function (HF)
     Function for the purpose of generating the digest without using the pre-shared key.
   - Message authentication code scheme (MAC)
     Function for the purpose of generating the digest using the pre-shared key.
2) The VP sets up the following roles among the cryptographic primitives enumerated in step 1 in the authentication and key exchange protocols.
   - Cryptographic primitives required for authenticator generation in the authentication protocol (PAG).
   - Cryptographic primitives required for key generation in the key exchange protocol (PKG).
   - Cryptographic primitives that appear in flows and include the password (PAF).
   - Cryptographic primitives included in the arguments of other cryptographic primitives (PAO).
   - Cryptographic primitives that are not PAG, PKG, PAF or PAO (PNA).

   Here, we define a framework as $g(f(A, B), C)$ with respect to the aforementioned roles without loss of generality. f and g denote the aforementioned roles and A, B and C denote the values of the cryptographic primitives or data enumerated in step 1, where other arguments of f and g that are not related to the verification are ignored. In this case, the combinations of f and g are as follows.
   - g is the PNA and f is the PAG, PKG or PAF, namely, $f(A, B)$.
   - g is the PAG, PKG or PAF and f is the PAO, namely, $g(f(A, B), C)$.

   There are no other variants, since the VP sets up not only the data but also the values of the cryptographic primitives, as described in step 3.
3) The VP sets up the following elements in respect of the values of the cryptographic primitives and data enumerated in step 1 in accordance with the protocol specifications.
   - Data types
     - General data (GD)
     - Identity data (ID)
     - Temporary data (TD)
     - Long-lived key (LLK)
     - Password (PW)
   - Values types
     - Fixed value (FV)
     - Temporary value (TV)
   - Values and data states
     - Public state (PS)
     - Secret state (SS)
4) The VP sets up the security properties defined in Section II according to the user's requirement for the authentication and key exchange protocols.
5) The VP checks the verification points shown in Appendix, using the elements of step 3 for the security properties of step 4 in the authentication and key exchange protocols. If the authentication and key exchange protocols are satisfied with at least one verification point item of the security property, then the VP determines that these protocols are secure against this security property. Then, the VP sets up these elements and security properties in accordance with the order of the protocol flows for the values of the cryptographic primitives and data that are related to each attack. Here, the values and data states are renewed, where public states are given priority over secret states.

We provide the verification example of the proposed method in Section IV-A in order to show how to verify a protocol.

B. Relations of Verification Points

This subsection describes the relations of the six verification points.

We explain the relations of the six verification points. The VP sets up the data that are related to each attack in the proposed method. Table I denotes the corresponding data and the combinations of f and g. The security properties are roughly classified into three as can be seen from the combinations of f and g in Table I: MC-SIA, SS group (SS-SPA, SS-SAA, SS-KKS and SS-WFS) and RODA. MC-SIA and RODA are independent of the other security properties since the former's target is the authenticator and the latter's target is the flows that include the password, respectively.

On the other hand, there are some relations in the SS group since it has the same target as the key generation function. First, SS-SPA is the weakest security level in the SS group, that is, SS-SPA has the most verification point items. The verification points of the remaining SS-SAA, SS-KKS and SS-WFS are derived from that of SS-SPA. Second, SS-SAA implies SS-SPA from the security properties, that is, the verification point of SS-SPA completely includes that of SS-SAA. Third, the known key attack is equivalent to the active attack, except that the adversary can obtain session keys in other sessions. The verification point of SS-KKS is the same as that of SS-SAA since the data and values with respect to the session keys in other sessions are only set up in accordance with the known key attack. Finally, the adversary can obtain the long-lived keys in the corruption attack, which is different from the eavesdropping attack. That is, the long-lived keys in the verification point of SS-SPA are modified into the public state from the secret state in the verification point of SS-WFS. Then, the inapplicable items in the verification point of SS-WFS need to be deleted.

We show the verification points of MC-SIA, SS-SPA, SS-SAA, SS-KKS, SS-WFS and RODA in Tables III – VII of Appendix. See [13] with respect to how to derive the verification point of each security property.

IV. Evaluation

This section shows the evaluation of the method proposed in Section III.

A. Verification Example

This subsection shows the verification example of the proposed method.

Table I
SETUP DATA AND COMBINATIONS OF f AND g FOR EACH SECURITY PROPERTY.

	MC-SIA		SS-SPA		SS-SAA	
Data	All flows		All flows		All flows	
g	PNA	PAG	PNA	PKG	PNA	PKG
f	PAG	PAO	PKG	PAO	PKG	PAO
	SS-KKS		SS-WFS		RODA	
Data	All flows Other session keys		All flows Long-lived keys		All flows	
g	PNA	PKG	PNA	PKG	PNA	PAF
f	PKG	PAO	PKG	PAO	PAF	PAO

We verify the security of the authenticated key exchange protocol using the proposed method as the example. This protocol, which is satisfied with the six security properties: MC-SIA, SS-SPA, SS-SAA, SS-KKS, SS-WFS and RODA, is one of the authenticated key exchange protocols that were automatically generated using an automatic generation technique [14], as described in Section IV-B. Figure 1 shows the protocol flow. Parties P_1 and P_2 share a password pw beforehand. The party P_1 generates a random number x and sends $\mathcal{E}_{pw}(g^x)$ to the party P_2, where $\mathcal{E}_{pw}$ is the encryption using the password pw and g^x is the Diffie-Hellman-based public value. The party P_2 generates a random number y and sends $\mathcal{E}_{pw}(H(g^x \parallel g^y) \parallel g^y) \parallel H(g^x)$ to the party P_1, where H is the hash function and g^y is the Diffie-Hellman-based public value. The party P_1 sends $H(g^y)$ to the party P_2. Finally, the parties P_1 and P_2 share a session key $sk = H(g^{xy})$.

Then, the roles of cryptographic primitives and types and states of data and values are set up for this protocol as items 1 and 2, respectively. Note that the states of data and values are different for the case of SS-WFS and cases other than SS-WFS in item 2. Also, the VP determines that this protocol is secure against each security property, since f and g take the corresponding cryptographic primitives and A, B and C take the corresponding values of the cryptographic primitives and data for the framework $g(f(A, B), C)$ in item 3, where "null" denotes empty.

1) Roles of cryptographic primitives:
   - Cryptographic primitives $= \{g_1, g_2, g_3, g_4, H_1, H_2, H_3, H_4, H_5, E_1, E_2\}$
     - $g_1 = g^x$ [DH], $g_2 = g^y$ [DH]
     - $g_3 = (g^y)^x$ [DH], $g_4 = (g^x)^y$ [DH]
     - $H_1 = H(g_1 \parallel g_2)$ [HF]
     - $H_2 = H(g_3)$ [HF], $H_3 = H(g_4)$ [HF]
     - $H_4 = H(g_1)$ [HF], $H_5 = H(g_2)$ [HF]
     - $E_1 = \mathcal{E}_{pw}(g_1)$ [EPW]
     - $E_2 = \mathcal{E}_{pw}(H_1 \parallel g_2)$ [EPW]

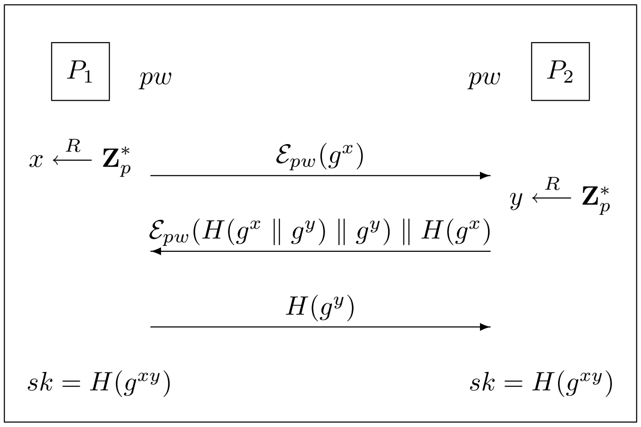
P_1
pw
pw
P_2
$x \xleftarrow{R} \mathbf{Z}_p^*$
$\mathcal{E}_{pw}(g^x)$
$y \xleftarrow{R} \mathbf{Z}_p^*$
$\mathcal{E}_{pw}(H(g^x \parallel g^y) \parallel g^y) \parallel H(g^x)$
$H(g^y)$
$sk = H(g^{xy})$
$sk = H(g^{xy})$

Figure 1. Protocol example, which is satisfied with the six security properties: MC-SIA, SS-SPA, SS-SAA, SS-KKS, SS-WFS and RODA.

- PAG $= \{H_5 \,(\text{of}\, P_1), H_4 \,(\text{of}\, P_2)\}$
- PKG $= \{H_2 \,(\text{of}\, P_1), H_3 \,(\text{of}\, P_2)\}$
- PAO for PKG $= \{E_2 \,(\text{for}\, H_2), E_1 \,(\text{for}\, H_3)\}$
- PAF $= \{E_1, E_2\}$

2) Types and states of data and values:

- Types and states $= \{pw, x, y, g_1, g_2, g_3, g_4, H_1, H_2, H_3, H_4, H_5, E_1, E_2\}$
  - $pw =$ PW-PS (when SS-WFS)
  - $pw =$ PW-SS (except for SS-WFS)
  - $x =$ TD-SS, $y =$ TD-SS
  - $g_1 =$ TV-PS (when SS-WFS)
  - $g_1 =$ TV-SS (except for SS-WFS)
  - $g_2 =$ TV-PS (when SS-WFS)
  - $g_2 =$ TV-SS (except for SS-WFS)
  - $g_3 =$ TV-SS, $g_4 =$ TV-SS
  - $H_1 =$ TV-PS (when SS-WFS)
  - $H_1 =$ TV-SS (except for SS-WFS)
  - $H_2 =$ TV-SS, $H_3 =$ TV-SS
  - $H_4 =$ TV-PS, $H_5 =$ TV-PS
  - $E_1 =$ TV-PS, $E_2 =$ TV-PS

3) Reasons that meet each security property:

- MC-SIA
  - P_1: $\text{null}(H_5\,[\text{HF}](g_2\,[\text{TV-SS}], \text{null}), \text{null})$
  - P_2: $\text{null}(H_4\,[\text{HF}](g_1\,[\text{TV-SS}], \text{null}), \text{null})$
- SS-SPA
  - P_1: $\text{null}(H_2\,[\text{HF}](g_3\,[\text{TV-SS}], \text{null}), \text{null})$
  - P_2: $\text{null}(H_3\,[\text{HF}](g_4\,[\text{TV-SS}], \text{null}), \text{null})$
- SS-SAA and SS-KKS
  - P_1: $H_2\,[\text{HF}](E_2\,[\text{EPW}](pw\,[\text{PW-SS}], \text{null}), g_3\,[\text{TV-SS}])$
  - P_2: $H_3\,[\text{HF}](E_1\,[\text{EPW}](pw\,[\text{PW-SS}], \text{null}), g_4\,[\text{TV-SS}])$
- SS-WFS
  - P_1: $H_2\,[\text{HF}](E_2\,[\text{EPW}](pw\,[\text{PW-PS}], \text{null}), g_3\,[\text{TV-SS}])$
  - P_2: $H_3\,[\text{HF}](E_1\,[\text{EPW}](pw\,[\text{PW-PS}], \text{null}), g_4\,[\text{TV-SS}])$
- RODA
  - 1st flow: $\text{null}(E_1\,[\text{EPW}](pw\,[\text{PW-SS}], g_1\,[\text{TV-SS}]), \text{null})$
  - 2nd flow: $\text{null}(E_2\,[\text{EPW}](pw\,[\text{PW-SS}], H_1\,[\text{TV-SS}]), \text{null})$

We explain the verification process of P_1 in MC-SIA as an example. The VP sets up the items 1 and 2 by steps 1 $\sim$ 3 of the proposed method. Here, PAG of P_1 is $H_5 = H(g_2)$ and its PAO is null. Thus, the authenticator of P_1 has the form of "$\text{null}(H_5\,[\text{HF}](g_2\,[\text{TV-SS}], \text{null}), \text{null})$" for the framework $g(f(A, B), C)$, as described in item 3. In this case, the aforementioned form is satisfied with the item of the third row in Table III. That is, f is HF of PDH, A is TV-SS of T*-SS and g, B and C are null.

B. Verification Results

This subsection describes the verification results using the method proposed in Section III.

An automatic generation technique of the authentication and key exchange protocols was proposed in [14], in relation to this paper. In [14], eighty-seven types of authentication and key exchange protocols, which are composed of 15 authentication (Auth), 22 key exchange (KE) and 50 authenticated key exchange (AKE) protocols, were automatically generated using this automatic generation technique. In the automatic generation technique, the optimal protocol is generated automatically when the following items are set up.

- Types: Auth, KE and AKE
- Cryptographic algorithms: algorithms that correspond to SKE, EPW, PKE, DH, SIG, HF and MAC
- Security properties: MC-SIA, SS-SPA, SS-SAA, SS-KKS, SS-WFS and RODA
- The numbers of flows: 1, 2 and 3

Then, we verified the security of the aforementioned authentication, key exchange and authenticated key exchange protocols, using the proposed method. Table II shows the verification results, best, worst and average verification time, minimal, maximal and average protocol definition file size for the authentication, key exchange and authenticated key exchange protocols, respectively, where the unit of the verification time is the millisecond and the unit of the protocol definition file size is the kilobyte. Symbols "Y", "N" and "—" denote that the protocol "meets", "does not meet" and "does not require" the corresponding security property, respectively.

These results completely coincide with the security requirements for the automatically generated protocols. The verification time is within 110 [ms] in the 87 authentication and key exchange protocols, using a PC with an Intel Pentium 4 2.6-GHz processor and 2.0-Gbyte RAM. On the other hand, TRUST [15] takes 40 [ms] $\sim$ 1.8 [s] at the fastest among the methods based on formal verification [16]. We cannot make a precise comparison between the proposed method and the existing methods, since the performance of the PC and the verified protocols are different from ours. However, the proposed method can verify the security of each protocol automatically and more quickly than most existing methods, since our method takes 4.6 [ms] $\sim$ 110 [ms] from Table II. Furthermore, the size of the protocol definition file is within 14.2 [KB] in the aforementioned protocols and the program size is 1.25 [MB].

V. Conclusion

Various security functions require to be taken into consideration for many potential threats, respectively, in order to realize the secure P2P system. Then, we took up the authentication and key exchange protocols as a target of the security functions in the P2P system. These protocols can

Table II
VERIFICATION RESULTS IN AUTHENTICATION AND KEY EXCHANGE PROTOCOLS.

Types	MC	SS				RODA	Numbers	Verification Time [ms]			Protocol Definition File Size [KB]		
	SIA	SPA	SAA	KKS	WFS			Best	Worst	Average	Minimum	Maximum	Average
Auth	Y	—	—	—	—	—	13	4.648	9.256	6.597	6.34	8.42	7.18
	Y	—	—	—	—	Y	2	8.235	11.988	10.112	7.11	7.93	7.52
KE	—	Y	Y	N	N	—	3	8.948	16.451	11.590	4.79	5.66	5.09
	—	Y	Y	Y	N	—	5	12.352	15.091	13.071	5.51	6.28	5.82
	—	Y	Y	Y	Y	—	12	20.035	32.445	27.058	6.27	8.10	7.20
	—	Y	Y	Y	N	Y	1	23.424	23.424	23.424	6.36	6.36	6.36
	—	Y	Y	Y	Y	Y	1	39.138	39.138	39.138	7.69	7.69	7.69
AKE	Y	Y	Y	Y	N	—	20	30.215	67.539	40.519	7.27	9.68	8.39
	Y	Y	Y	Y	Y	—	28	41.864	109.054	73.821	8.23	14.20	10.72
	Y	Y	Y	Y	N	Y	1	64.928	64.928	64.928	8.77	8.77	8.77
	Y	Y	Y	Y	Y	Y	1	88.700	88.700	88.700	9.90	9.90	9.90

bear one duty in order to realize the secure P2P system. So far, we proposed the security verification method for the aforementioned protocols based on the BPR model and showed the verification points of security properties to verify their security efficiently.

In this paper, we described the relations of the six verification points and explained the verification example using the proposed method. We also verified the security of 87 authentication and key exchange protocols, which were generated automatically. Then, we confirmed that the verification time was within 110 [ms] and that the security properties of the verification results completely coincided with the security requirements for the aforementioned protocols.

REFERENCES

[1] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Advances in Cryptology — CRYPTO'93*, vol. 773. Santa Barbara, CA: Springer-Verlag, Aug. 1993, pp. 232–249.

[2] M. Bellare and P. Rogaway, "Provably secure session key distribution — The three party case," in *Proc. 27th Annual ACM Symposium on Theory of Computing*. Philadelphia, PA: ACM Press, May/Jun. 1995, pp. 57–66.

[3] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology — EUROCRYPT 2000*, vol. 1807. Bruges, Belgium: Springer-Verlag, May 2000, pp. 139–155.

[4] M. Bellare, R. Canetti, and H. Krawczyk, "A modular approach to the design and analysis of authentication and key exchange protocols," in *Proc. 30th Annual ACM Symposium on the Theory of Computing*. Dallas, TX: ACM Press, May 1998, pp. 419–428.

[5] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *Proc. 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2001)*. Las Vegas, NV: IEEE Computer Society Press, Oct. 2001, pp. 136–145.

[6] D. Dolev and A. Yao, "On the security of public key protocols," in *Proc. 22nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 1981)*. Nashville, TN: IEEE Computer Society Press, Oct. 1981, pp. 350–357.

[7] A. Roscoe, "Modelling and verifying key-exchange protocols using CSP and FDR," in *Proc. Eighth IEEE Computer Security Foundations Workshop*. County Kerry, Ireland: IEEE Computer Society Press, Mar. 1995, pp. 98–107.

[8] M. Abadi and A. Gordon, "A calculus for cryptographic protocols: The spi calculus," *Information and Computation*, vol. 148, no. 1, pp. 1–70, 1999.

[9] M. Burrows, A. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[10] L. Paulson, "Proving properties of security protocols by induction," Univ. of Cambridge, Cambridge, UK, Computer Laboratory Tech. Rep. 409, 1996.

[11] H. Ota, S. Kiyomoto, and T. Tanaka, "Security Verification for Authentication and Key Exchange Protocols," in *Proc. 2008 International Symposium on Information Theory and its Applications (ISITA 2008)*. Auckland, New Zealand: Computer Society Press, Dec. 2008, pp. 507–512.

[12] H. Ota, S. Kiyomoto, and T. Tanaka, "Security Verification for Authentication and Key Exchange Protocols," *International Journal of Computer Science and Network Security*, vol. 9, no. 3, pp. 1–11, 2009.

[13] H. Ota, S. Kiyomoto, and T. Tanaka, "Security verification for authentication and key exchange protocols, revisited," in *Proc. 2010 IEEE 24th IEEE International Conference on Advanced Information Networking and Applications Workshops*. Perth, Australia: IEEE Computer Society Press, Apr. 2010, pp. 226–233.

[14] S. Kiyomoto, H. Ota, and T. Tanaka, "Security protocol dynamic generation and modification mechanisms for ubiquitous services," in *Proc. 11th International Conference on Wireless Personal Multimedia Communications (WPMC'08)*, Lapland, Finland, 2008.

[15] R. Amadio, D. Lugiez, and V. Vancackere, "On the symbolic reduction of processes with cryptographic functions," *Theoretical Computer Science*, vol. 290, no. 1, pp. 695–740, 2003.

[16] A. Bracciali, G. Baldi, G. Ferrari, and E. Tuosto, "A coordination-based methodology for security protocol verification," in *Proc. 2nd International Workshop on Security Issues with Petri Nets and other Computational Models (WISP2004)*, vol. 121. Bologna, Italy: Elsevier Science, 2004, pp. 23–46.

APPENDIX

This appendix presents detailed tables of the verification points referred to in Section III-B.

Tables III – VII show the verification points of MC-SIA, SS-SPA, SS-SAA, SS-KKS, SS-WFS and RODA. Table IV shows the common verification point of SS-SPA, SS-SAA and SS-KKS and Table V shows the remaining verification point of SS-SPA, where the verification points of SS-SAA and SS-KKS coincide with Table IV. In addition, each abbreviation symbol denotes the following.

- ALL denotes SKE, EPW, PKE, DH, SIG, HF or MAC.
- 6-SIG denotes SKE, EPW, PKE, DH, HF or MAC.
- SSM denotes SKE, SIG or MAC.
- S/S denotes SKE or SIG.
- SM denotes SKE or MAC.
- EPDH denotes EPW, PKE, DH or HF.
- PDHM denotes PKE, DH, HF or MAC.
- PDH denotes PKE, DH or HF.
- T*-*S denotes TD-PS, TD-SS, TV-PS or TV-SS.
- T*-SS denotes TD-SS or TV-SS.
- T*-SS+ denotes TD-SS, TV-SS or FV with LLK-SS.
- EXC denotes elements except for PW-PS and PW-SS.

Table III
VERIFICATION POINTS OF MC-SIA.

g	f	A	B	C
—	SSM	LLK-SS	T*-*S	—
—	EPW	PW-SS	T*-*S	—
—	PDH	T*-SS	—	—
SSM	SSM	LLK-SS	T*-*S	LLK-SS
ALL	SSM	LLK-SS	—	T*-*S
SSM	EPW	PW-SS	T*-*S	LLK-SS
ALL	EPW	PW-SS	—	T*-*S
SSM	PDH	T*-*S	—	LLK-SS
EPW	SSM	LLK-SS	T*-*S	PW-SS
EPW	EPW	PW-SS	T*-*S	PW-SS
EPW	PDH	T*-*S	—	PW-SS
PDH	SSM	LLK-SS	T*-*S	—
PDH	EPW	PW-SS	T*-*S	—
PDH	PDH	T*-SS	—	—
PDH	PDH	—	—	T*-SS

Table IV
COMMON VERIFICATION POINTS OF SS-SPA, SS-SAA AND SS-KKS.

g	f	A	B	C
SM	SSM	LLK-SS	T*-*S	LLK-SS
6-SIG	SSM	LLK-SS	—	T*-*S
SSM	EPW	PW-SS	T*-*S	LLK-SS
6-SIG	EPW	PW-SS	—	T*-*S
SM	PDH	T*-*S	—	LLK-SS
EPW	SSM	LLK-SS	T*-*S	PW-SS
EPW	EPW	PW-SS	T*-*S	PW-SS
EPW	PDH	T*-*S	—	PW-SS
PDH	SSM	LLK-SS	T*-*S	—
PDH	EPW	PW-SS	T*-*S	—
SIG	SM	LLK-SS	T*-*S	LLK-SS

Table V
REMAINING VERIFICATION POINTS OF SS-SPA.

g	f	A	B	C
—	SM	LLK-SS	T*-*S	—
—	EPW	PW-SS	T*-*S	—
—	PDH	T*-SS	—	—
EPW	PDH	—	—	T*-*S
PDH	PDH	T*-SS	—	—
PDH	PDH	—	—	T*-SS
SIG	PDH	T*-SS	—	LLK-SS

Table VI
VERIFICATION POINTS OF SS-WFS.

g	f	A	B	C
SSM	PDH	T*-SS	—	LLK-PS
S/S	MAC	LLK-PS	T*-SS	LLK-PS
EPW	PDH	T*-SS	—	PW-PS
EPW	MAC	LLK-PS	T*-SS	PW-PS
PDHM	SSM	LLK-PS	—	T*-SS
PDHM	EPW	PW-PS	—	T*-SS
PDH	PDH	T*-SS	—	—
PDHM	PDH	—	—	T*-SS
PDH	MAC	LLK-PS	T*-SS	—
MAC	SSM	LLK-PS	T*-SS	LLK-PS
MAC	EPW	PW-PS	T*-SS	LLK-PS

Table VII
VERIFICATION POINTS OF RODA.

g	f	A	B	C
—	SM	PW-SS	LLK-SS	—
—	EPDH	PW-SS	T*-SS+	—
SM	SSM	PW-SS	LLK-SS	LLK-SS
SM	EPDH	PW-SS	—	LLK-SS
EPW	SSM	PW-SS	LLK-SS	PW-SS
EPW	EPDH	PW-SS	T*-SS+	PW-SS
EPDH	EPDH	PW-SS	—	T*-SS+
PDH	SSM	PW-SS	LLK-SS	—
PDH	EPDH	PW-SS	T*-SS+	—
SIG	SM	PW-SS	LLK-SS	EXC
SIG	EPDH	PW-SS	T*-SS+	EXC