Fast and Automatic Verification of Authentication and Key Exchange Protocols [1]

Haruki Ota[†] Shinsaku Kiyomoto[†], and Yutaka Miyake[†]

[†]KDDI R&D Laboratories, Inc., 2-1-15 Ohara, Fujimino-shi, Saitama, 356-8502 Japan

Summary
It is preferable for authentication and key exchange protocols to be verified automatically and rapidly in accordance with security requirements. In order to meet these requirements, we proposed the security verification method (OKT method) for the aforementioned protocols based on Bellare et al.'s model (BPR model) and showed the verification points of security properties to verify their security efficiently. However, there is an estrangement between the security of the OKT method and the BPR model. In this paper, we reconsider the OKT method and propose an updated security verification method for authentication and key exchange protocols based on the BPR model. In particular, we revise the procedure of the OKT method to address the aforementioned issue. We show the novel verification points for each security property in the authentication and key exchange protocols in accordance with the aforementioned revisions. In addition, we describe the relations among the six verification points, explain how the proposed method verifies the aforementioned protocols by providing one example and show the validity of the proposed method by verifying the security of 87 authentication and key exchange protocols that were generated automatically.

Key words:
Security Verification Method, Authentication and Key Exchange Protocols, Verification Points, Bellare et al.'s Model

1. Introduction

1.1 Motivation

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. Furthermore, designing a new protocol or modifying an existing protocol and then 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.

1.2 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 one example 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 [3, 4, 5]. Specifically, Bellare and Rogaway first proposed 2-party mutual authentication and authenticated key exchange protocols in 1993 [3], and subsequently extended this to a 3-party setting via a key distribution center with respect to key exchange protocols in 1995 [4]. In 2000, Bellare, Pointcheval and Rogaway proposed provably secure password-based key exchange and authenticated key exchange protocols based on the Bellare-Rogaway model [5]. Bellare et al. formulated models that were secure against an off-line dictionary attack and forward secrecy. Hereinafter, we call the model proposed in [3], [4] and [5] 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 [6] and a universally composable framework [7]. However, the problem remained that the security of the protocols still needed to be proved. That is, there was no automatic verification method based on the BPR model since it is very difficult to implement as algorithms the formulations of provable security in the BPR model.

On the other hand, methods based on formal verification are classified into the following: those based on state-machine approaches, those using model checkers, those using algebraic systems, those based on modal logic and those based on inductive approaches. Methods based on state-machine approaches include the Dolev-Yao model [8, 9], Interrogator [10], NRL (Naval Research Laboratory) Protocol Analyzer [11, 12], Longley-Rigby tool [13] and the strand space model [14]. Methods using model checkers include FDR (Failures Divergences

[1] Preliminary versions of this paper appeared in [1, 2].

Manuscript received April 5, 2011
Manuscript revised April 20, 2011

Refinement) / CSP (Communicating Sequential Processes) [15, 16] and Murφ [17]. Methods using algebraic systems include spi calculus [18], LOTOS (Language of Temporal Ordering Specification) [19], TRUST [20] and CryptoVerif [21, 22]. Methods based on modal logic include BAN (Burrows-Abadi-Needham) logic [23], GNY (Gong-Needham-Yahalom) logic [24] and SVO (Syverson-van Oorschot) logic [25]. Methods based on inductive approaches include Isabelle/HOL (Higher Order Logic) [26, 27, 28, 29] and CafeOBJ [30, 31]. 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 [32, 33], which we hereinafter call the "OKT method." We generalized the process of the security proofs based on the BPR model to implement it as a tool. In particular, we showed the verification points of security properties for authentication and key exchange protocols so that the security of each protocol could be verified rapidly and automatically. The verification points have the characteristic that the authentication and key exchange protocols are determined to be secure if they satisfy at least one verification point item of the security property. However, while the verification points of the OKT method may be sufficient conditions, they may not be necessary conditions. In the BPR model, the security of a specific protocol is proven individually. Meanwhile, it is necessary for the OKT method to be applicable to any protocol. Therefore, there is an estrangement between the security of the OKT method and the BPR model. This is because the OKT method is associated with the following three issues.

- The roles of cryptographic primitives configured in the OKT method are not restricted to their original roles.
- The OKT method may be unable to deal effectively with more deeply nested data included inside the arguments of the cryptographic primitives.
- Definitions of the types and states for the values of cryptographic primitives and data include redundant parameters.

1.3 Contributions

In this paper, we reconsider the OKT method and propose an updated security verification method for authentication and key exchange protocols based on the BPR model. First, we review the security properties of the BPR model and the procedures of the OKT method. In particular, we revise the procedure of the OKT method to address the aforementioned three issues as follows.

- We explicitly characterize the roles of the cryptographic primitives.
- We reconsider the treatment for more deeply nested data included inside the arguments of the cryptographic primitives.
- We redefine the types and states for the values of cryptographic primitives and data.

We show the novel verification points for each security property in the authentication and key exchange protocols in accordance with the aforementioned revisions. Our proposed method is characterized by the fact that it can verify the security of authentication and key exchange protocols automatically and rapidly on a par with the OKT method, since the basic concept upon which both methods are based is the same. In addition, we provide the following in order to make our method clear.

- We describe the relations among the six verification points by considering attack models and security targets.
- We explain how the proposed method verifies the aforementioned protocols by providing one verification example.
- 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.

1.4 Organization

The rest of this paper is organized as follows. We introduce the BPR model in Sect. 2. We propose an updated security verification method for authentication and key exchange protocols in Sect. 3 and present the novel verification points of the security properties for these protocols in Sect. 4. We consider the proposed method by comparing it with the OKT method in Sect. 5. We explain the verification example and the verification results using the proposed method in Sect. 6. Our conclusions are presented in Sect. 7 and we present detailed tables of the verification points for the aforementioned protocols in Appendix A.

2. 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 [3]. They formulated the following security properties from real attacks, which are shown in brackets, for each notion in accordance with security requirements.

- Matching conversation (MC) [3]
  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) [3]
    An adversary cannot break an authentication protocol even when he/she controls all communications between parties.
    [Impersonation attack]
- Semantic security (SS) [3]
  In a key exchange protocol, an adversary cannot distinguish between the session key and a random session key.
  - Security against a passive attack (SS-SPA) [3, 4]
    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) [3, 4]
    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) [3, 4]
    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) [4, 5]
    An adversary cannot obtain a 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) [2] [5]
    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]

[2] In [30], we described RODA as being confined to the security properties of the key exchange protocol. However, in this paper, we deal with RODA as including the common security properties of the both protocols.

3. Security Verification Method

In this section, the procedures of the OKT method are revised and an updated security verification method for authentication and key exchange protocols based on the BPR model is proposed.

We deal with only two-party authentication and key exchange protocols in this paper. Here, we assume the following when verifying the security of the aforementioned protocols.

- Two parties share a secret key or password in a secure manner beforehand when the secret key or password is used.
- Each party can confirm the validity of the other party's public key certificate in a secure manner by means of a trusted third party, such as a certificate authority, when the public key is used.
- The cryptographic primitives are not compromised. If compromised cryptographic primitives are used, then the verification program (VP) determines that the aforementioned protocols are not secure.

The 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 a pre-shared key.
- Encryption using password (EPW)
  Function for the purpose of encryption using a pre-shared password.
- Public key encryption (PKE)
  Function for the purpose of encryption using a 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 a signature using a signing key.
- Hash function (HF)
  Function for the purpose of generating a digest without using a pre-shared key.
- Message authentication code scheme (MAC)
  Function for the purpose of generating a digest using a 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 a 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)

These elements used later mean the following.

- TD-PS denotes the public temporary data.
- TD-SS denotes the secret temporary data.
- TV-PS denotes the public temporary value.
- TV-SS denotes the secret temporary value.
- FV-PS denotes the public fixed value.
- FV-SS denotes the secret fixed value.
- LLK-PS denotes the public long-lived key.
- LLK-SS denotes the secret long-lived key.
- PW-PS denotes the public password.
- PW-SS denotes the secret password.

(4) The VP sets up the security properties defined in Sect. 2 in accordance with the user's requirement for the authentication and key exchange protocols. Then, it sets up the security parameters required for these protocols and confirms whether the sizes of the respective data and values are larger than or equal to these parameters or not. If there are data sizes smaller than these parameters, then the VP determines that the authentication and key exchange protocols are not secure. This is because the aforementioned protocols are not secure against an exhaustive search attack when data sizes are smaller than the parameters.

(5) The VP checks the verification points shown in Sect. 4 and Appendix A, 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 satisfy 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.

4. Verification Points

This section shows the verification points of the security properties for each protocol.

4.1 Relations Among Verification Points

This subsection describes the relations among the six verification points.

We explain the relations among the six verification points. The VP sets up the data that are related to each attack in the proposed method. Table 1 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 1: 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 targets are the flows that include the password.

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, a known key attack is equivalent to an 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 a known key attack. Finally, the adversary can obtain the long-lived keys in a corruption attack, which is different from an 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.

Table 1: 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

4.2 Derivations of Verification Points

This subsection describes how to derive the verification points.

First, we consider the case where the framework is $f(A, B)$, that is, g is the PNA and f is the PAG, PKG or PAF. Next, we consider the case where the framework is $g(f(A, B), C)$, that is, g is the PAG, PKG or PAF and f is the PAO. In this case, the verification points of each security property are basically obtained through the combination of those in the case of $f(A, B)$. However, there are some exceptions in some of their combinations. Therefore, we describe the case of $f(A, B)$ and the exception of $g(f(A, B), C)$ for each verification point after the next subsection.

4.2.1 Security Against an Impersonation Attack (MC-SIA)

First, we consider the case where f is the PAG in $f(A, B)$. An adversary has only to be able to forge the authenticator or reuse a previous authenticator in order to impersonate some party. In other words, the authentication protocols are secure against an impersonation attack if he/she can neither forge the authenticator nor reuse a previous authenticator. The requirements for the verification points of MC-SIA are to include the temporary data or values and secret data or values in the PAG. That is, the following two requirements are obtained.

- f represents the cryptographic primitives that have the secret long-lived key (LLK-SS or PW-SS) and temporary data (TD) or values (TV), such as the SKE, EPW, SIG or MAC.
- f represents the cryptographic primitives for which arguments include the secret temporary data (TD-SS) or values (TV-SS), such as the PKE, DH or HF, if they do not have the long-lived key.

Next, we consider the exception where g is the PAG in $g(f(A, B), C)$. The secret temporary data must be included in the arguments of f when f is the PKE, DH or HF in $f(A, B)$. However, the following exception is allowed for the verification points of MC-SIA.

- The PKE, DH and HF can include the public temporary data (TD-PS) or values (TV-PS) as the verification points of MC-SIA if they can be protected by the SKE, EPW, SIG or MAC.

4.2.2 Security Against a Passive Attack (SS-SPA)

First, we consider the case where f is the PKG in $f(A, B)$. By eavesdropping, an adversary has only to be able to obtain the session key from all communications between parties. In other words, the key exchange protocols are secure against an eavesdropping attack if he/she cannot obtain any information with respect to the session key from the eavesdropped communications. The requirements for the verification points of SS-SPA are to include the temporary data or values and secret data or values in the PKG. That is, the following two requirements are obtained.

- f represents the cryptographic primitives that have the secret long-lived key (LLK-SS or PW-SS) and temporary data (TD) or values (TV), such as the SKE, EPW or MAC.
- f represents the cryptographic primitives for which arguments include the secret temporary data (TD-SS) or values (TV-SS), such as the PKE, DH or HF, if they do not have the long-lived key.

The SIG is also the cryptographic primitive that has the secret long-lived key. However, it is excluded from the verification points of SS-SPA, since each party knows only his/her signing key and cannot generate the same session key by his/her signature.

Next, we consider the exception where g is the PKG in $g(f(A, B), C)$. The secret temporary data must be included in the arguments of f when f is the PKE, DH or HF in $f(A, B)$. In addition, f cannot take the SIG in $f(A, B)$ for the aforementioned reason. However, the following two exceptions are allowed for the verification points of SS-SPA.

- The SIG can be included in the verification points of SS-SPA if it can be used as PAO.
- The PKE, DH and HF can include the public temporary data (TD-PS) or values (TV-PS) as the verification points of SS-SPA if they can be protected by the SKE, EPW, SIG or MAC.

4.2.3 Security Against an Active Attack (SS-SAA)

First, we consider the case where f is the PKG in $f(A, B)$. An adversary has only to be able to obtain the session key by controlling all communications between parties. In other words, the key exchange protocols are secure against an active attack if he/she cannot obtain any information with respect to the session key even though he/she controls the communications. In the framework of $f(A, B)$, there are no verification points for which the key exchange protocols are secure against an active attack. However, the verification points in the case of $g(f(A, B), C)$ are basically obtained through the combinations of those in the case of $f(A, B)$. Then, we consider the provisional requirements in the case of $f(A, B)$. The requirements for the verification points of SS-SAA are to include the secret long-lived key and the temporary data or values in the PKG. That is, the following requirement is obtained.

- f represents the cryptographic primitives that have the secret long-lived key (LLK-SS or PW-SS) and temporary data (TD) or values (TV), such as the SKE, EPW or MAC.

The SIG is also the cryptographic primitive that has the secret long-lived key. However, it is excluded from the verification points of SS-SAA, since each party knows only his/her signing key and cannot generate the same session key by his/her signature. Furthermore, the PKE, DH and HF are excluded from the verification points of SS-SAA, since the elements of the session key may be altered by an active adversary even if they are secret data.

Next, we consider the exception where g is the PKG in $g(f(A, B), C)$. f can take neither the PKE, DH, SIG nor HF in $f(A, B)$ for the aforementioned reasons. However, the following exception is allowed for the verification points of SS-SAA.

- The SIG can be included in the verification points of SS-SAA if it can be used as PAO.
- The PKE, DH and HF can be included in the verification points of SS-SAA if they can be protected by other cryptographic primitives.

4.2.4 Known Key Security (SS-KKS)

First, we consider the case where f is the PKG in $f(A, B)$. An adversary has only to be able to generate the target session key from the other session keys or reuse the other session keys. In other words, the key exchange protocols achieve the known key security if he/she can neither generate the target session key from the other session keys nor reuse the other session keys. In the framework of $f(A, B)$, there are also no verification points for which the key exchange protocols achieve the known key security. However, the verification points in the case of $g(f(A, B), C)$ are basically obtained through the combinations of those in the case of $f(A, B)$. Then, we consider the provisional requirements in the case of $f(A, B)$. The requirements for the verification points of SS-KKS are to include the secret long-lived key and the temporary data or values in the PKG. That is, the following requirement is obtained.

- f represents the cryptographic primitives that have the secret long-lived key (LLK-SS or PW-SS) and temporary data (TD) or values (TV), such as the SKE, EPW or MAC.

The SIG is also the cryptographic primitive that has the secret long-lived key. However, it is excluded from the verification points of SS-KKS, since each party knows only his/her signing key and cannot generate the same session key by his/her signature. Furthermore, the PKE, DH and HF are excluded from the verification points of SS-KKS, since the elements of the session key may be altered by an active adversary even if they are secret data.

Next, we consider the exception where g is the PKG in $g(f(A, B), C)$. f can take neither the PKE, DH, SIG nor HF in $f(A, B)$ for the aforementioned reasons. However, the following exception is allowed for the verification points of SS-KKS.

- The SIG can be included in the verification points of SS-KKS if it can be used as PAO.
- The PKE, DH and HF can be included in the verification points of SS-KKS if they can be protected by other cryptographic primitives.

4.2.5 Weak Forward Secrecy (SS-WFS)

First, we consider the case where f is the PKG in $f(A, B)$. Here, an adversary can corrupt the long-lived keys by launching a corruption attack (LLK-PS and PW-PS). He/she has only to be able to obtain a past session key by using the corrupted long-lived keys. In other words, the key exchange protocols achieve weak forward secrecy if he/she cannot obtain a past session key even when he/she uses the corrupted long-lived keys. In the framework of $f(A, B)$, there are also no verification points for which the key exchange protocols achieve weak forward secrecy. However, the verification points in the case of $g(f(A, B), C)$ are basically obtained through the combinations of those in the case of $f(A, B)$. Then, we consider the provisional requirements in the case of $f(A, B)$. The

requirements for the verification points of SS-WFS are to include the secret temporary data or values in the PKG. That is, the following two requirements are obtained.

- f represents the cryptographic primitives for which arguments include the secret temporary data (TD-SS) or values (TV-SS), such as the PKE, DH or HF, if they do not have a corrupted long-lived key.
- f represents the cryptographic primitives for which arguments include the secret temporary data (TD-SS) or values (TV-SS), such as the MAC, even if it has a corrupted long-lived key.

The SKE and EPW are also the cryptographic primitives that may each have a corrupted long-lived key. However, they are excluded from the verification points of SS-WFS, since the secret data may be transformed into public data by the corrupted key or password. Similarly, the SIG is the cryptographic primitive that has the corrupted long-lived key. However, it is excluded from the verification points of SS-WFS, since each party knows only his/her signing key and cannot generate the same session key by his/her signature.

Next, we consider the exception where g is the PKG in $g(f(A, B), C)$. f can take neither the SKE, EPW nor SIG in $f(A, B)$ for the aforementioned reasons. However, the following exception is allowed for the verification points of SS-WFS.

- The SKE and EPW can be included in the verification points of SS-WFS if they can be protected by the other cryptographic primitives.
- The SIG can be included in the verification points of SS-WFS if it can be used as PAO.

4.2.6 Resistance to an Off-line Dictionary Attack (RODA)

First, we consider the case where f is the PAF in $f(A, B)$. Here, the targets of an adversary are only the flows that include the secret password (PW-SS) which can be recovered by launching an off-line dictionary attack. He/she has only to be able to search for the password of some party from past communications off-line. In other words, the authentication and key exchange protocols are resistant to an off-line dictionary attack if he/she cannot obtain the password of the party off-line. The requirements for the verification points of RODA are to include the secret data or values in the PAF for which sizes satisfy the security parameters. That is, the following two requirements are obtained.

- f represents the cryptographic primitives that have the secret long-lived key (LLK-SS), such as SKE or MAC.
- f represents the cryptographic primitives for which arguments include the secret data (TD-SS) or values (TV-SS or FV-SS with LLK-SS), such as EPW, PKE, DH or HF, if they do not have the long-lived key except for the password.

The SIG is also the cryptographic primitive that has the secret long-lived key. However, it is excluded from the verification points of RODA since the signature may be transformed into the corresponding message by the verification key.

Next, we consider the exception where g is the PAF in $g(f(A, B), C)$. f cannot take the SIG in $f(A, B)$ for the aforementioned reasons. However, the following exception is allowed for the verification points of RODA.

- The SIG for which an argument does not have the password can be included in the verification points of RODA if it can be protected by other cryptographic primitives.

We show the verification points of MC-SIA, SS-SPA, SS-SAA, SS-KKS, SS-WFS and RODA in Tables 3 -- 7 of Appendix A.

Remark 1: We showed the verification points of two security properties for authentication protocols and five security properties for key exchange protocols, as described above. Note that checking the verification points of security properties for authentication and key exchange protocols separately means checking those for an authenticated key exchange protocol.

5. Considerations

This section considers the proposed method by comparing it with the OKT method.

In the OKT method, there are three issues, as described in Sect. 1. Now, we reconsider the OKT method in relation to issues and obtain the proposed method by resolving the issues as follows.

(1) We make clear the roles of the cryptographic primitives.
(2) We add a framework for the roles of the cryptographic primitives.
(3) We explicitly redefine the types and states of the cryptographic primitives and data.

5.1 Codification of the Roles of Cryptographic Primitives

This subsection explains the codification of the roles of the cryptographic primitives with respect to item (1).

The classification of the cryptographic primitives in the proposed method is the same as that in the OKT method. The cryptographic primitives are set up in accordance with the protocol specification in the OKT

method. In this method, we distribute the cryptographic primitives to the following three properties: indistinguishability, one-wayness and unforgeability (see [32, 33] for details). However, the roles of cryptographic primitives configured in the OKT method are not restricted to their original roles depending on the protocols. For example, when the hash function takes the pre-shared key as its argument, this function plays the role of the MAC scheme. The verification points in the OKT method are redundant since it can be deduced that they correspond to another role of the cryptographic primitives.

Here, we consider the following key exchange protocol as an example of the aforementioned application of the hash function. The party P_2 sends $E_K(R)$ to the party P_1, where E is the secret key encryption, K is the secret long-lived key and R is the secret temporary data. The parties P_1 and P_2 obtain $H(K, R)$ as the session key, respectively, where H is the hash function. In this case, the role of $H(K, R)$ is the same as the MAC scheme. For example, this protocol satisfies the second verification point of SS-SPA in the OKT method (see [32, 33] for details). However, the hash function should be excluded from this verification point due to the aforementioned reason. On the other hand, this protocol satisfies the verification point of SS-SPA in the proposed method only when the PKG is the MAC scheme. Therefore, the verification points in the proposed method are not redundant since we make it possible to set up the roles of the cryptographic primitives depending on the protocols.

5.2 Addition of Framework

This subsection explains the reasons for adding a framework for the roles of the cryptographic primitives with respect to item (2).

The number and depth of the arguments of the cryptographic primitives can be set without any limit in the OKT method. Hence, the arguments of the cryptographic primitives may include deeper data inside the nested arguments, but it is not clear as to how such data are dealt with. There are also many data that are not related to the verification inside the arguments. Therefore, we add the framework $g(f(A, B), C)$ for the roles of the cryptographic primitives and explicitly determine the form of the cryptographic primitives. For example, when the form of the cryptographic primitive is $\delta(\gamma(\beta(\alpha(v, w), x), y), z)$, where α, β, γ and δ denote the cryptographic primitives and v, w, x, y and z denote the data, each element of the framework is $g = \delta$, $f = \gamma$, $A = \beta(\alpha(v, w), x)$, $B = y$ and $C = z$. In this case, $A = \beta(\alpha(v, w), x)$ is dealt with as the value of the cryptographic primitive and is set up as FV-PS, FV-SS, TV-PS or TV-SS in accordance with the roles of α and β and the types and states of v, w and x.

Here, we consider the authentication protocol in which the authenticator is $S_{sk}(H((g^x)^y))$, where S is the signature scheme, sk is the signing key and $(g^x)^y$ is the Diffie-Hellman key. In this case, this protocol satisfies the verification point of MC-SIA in the OKT method. However, the arguments of the signature scheme include deeper secret temporary data y inside the nested hash function and Diffie-Hellman function. On the other hand, each element of the framework is $g = S$, $f = H$, $A = (g^x)^y$, B = null and $C = sk$ and $A = (g^x)^y$ is set up as TV-SS in the proposed method. Therefore, the treatment of the deeper data is obvious in the proposed method.

5.3 Redefinition of Types and States

This subsection explains the redefinition of the types and states for the values of the cryptographic primitives and data with respect to item (3).

In the OKT method, there are the following three problems with respect to the types and states for the values of the cryptographic primitives and data.

(1) It is redundant to include the public and secret states in the elements of the types and states.
(2) The segment between the first and existing appearances in the states causes an estrangement between the security of the OKT method and the BPR model.
(3) The states of the deeper data inside the nested arguments are undetermined, as described in Sect. 5.2.

Therefore, we first of all remove the public and secret states from each element with respect to problem (1). Next, we remove the original six states and instead add the types for the values of the cryptographic primitives with respect to problems (2) and (3). Consequently, problem (3) is resolved by setting up the type and state for the values of the cryptographic primitives, instead of the deeper data. Problem (2) is also resolved since the ambiguous segment between the first and existing appearances is removed by combining the types and states for the values of the cryptographic primitives and data.

Here, we consider the same protocol as that of Sect. 5.1 except for the following. Party P_2 sends R and $M_K(R)$ to party P_1 instead of $E_K(R)$, where M is the MAC scheme. This protocol does not satisfy the verification points of SS-SPA in the OKT method although it is secure. This is because the temporary data R is dealt with as the existing appearance in the same session. On the other hand, this protocol satisfies with one of the verification points in the proposed method. Therefore, the security of the proposed method is closer to that of the BPR model than that of the OKT method.

6. Evaluation

The proposed method is evaluated in this section.

6.1 Verification Example

This subsection provides a verification example of the proposed method.

We verify the security of the authenticated key exchange protocol using the proposed method. This protocol, which satisfies 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 [34], as described in Sect. 6.2. Figure 1 shows the protocol flow. Parties P_1 and P_2 share a password pw beforehand. Party P_1 generates a random number x and sends $E_{pw}(g^x)$ to party P_2, where E_{pw} is the encryption using the password pw and g^x is the Diffie-Hellman-based public value. Party P_2 generates a random number y and sends $E_{pw}(H(g^x \| g^y) \| g^y) \| H(g^x)$ to party P_1, where H is the hash function and g^y is the Diffie-Hellman-based public value. Party P_1 sends $H(g^y)$ to party P_2. Finally, parties P_1 and P_2 share a session key $sk = H(g^{xy})$.

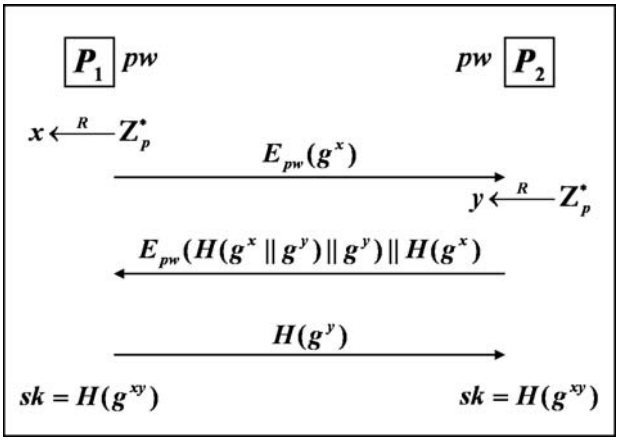


Fig. 1: Protocol example, which satisfies the six security properties: MC-SIA, SS-SPA, SS-SAA, SS-KKS, SS-WFS and RODA.

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]
  - $H_1 = H(g_1 \| g_2)$ [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]
  - $E_1 = E_{pw}(g_1)$ [EPW]
  - $E_2 = E_{pw}(H_1 \| g_2)$ [EPW]
- PAG = {H_5 (of P_1), H_4 (of P_2)}
- PKG = {H_2 (of P_1), H_3 (of P_2)}
- PAO for PKG = {E_2 (for H_2), E_1 (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: null(H_5 [HF](g_2 [TV-SS], null), null)
  - P_2: null(H_4 [HF](g_1 [TV-SS], null), null)
- SS-SPA
  - P_1: null(H_2 [HF](g_3 [TV-SS], null), null)
  - P_2: null(H_3 [HF](g_4 [TV-SS], null), null)
- SS-SAA and SS-KKS
  - P_1:
    H_2 [HF](E_2 [EPW](pw [PW-SS], null), g_3 [TV-SS])
  - P_2:
    H_3 [HF](E_1 [EPW](pw [PW-SS], null), g_4 [TV-SS])
- SS-WFS
  - P_1:
    H_2 [HF](E_2 [EPW](pw [PW-PS], null), g_3 [TV-SS])
  - P_2:
    H_3 [HF](E_1 [EPW](pw [PW-PS], null), g_4 [TV-SS])
- RODA
  - 1st flow:
    null(E_1 [EPW](pw [PW-SS], g_3 [TV-SS]), null)
  - 2nd flow:
    null(E_2 [EPW](pw [PW-SS], H_1 [TV-SS]), null)

We explain the verification process of P_1 in MC-SIA as an example. The VP sets up items (1) and (2) using steps (1) -- (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 "null(H_5 [HF](g_2 [TV-SS], null), null)" for the framework $g(f(A, B), C)$, as described in item (3). In this case, the aforementioned form satisfies with the item of the third row in Table 3. That is, f is HF of PDH, A is TV-SS of T*-SS and g, B and C are null.

6.2 Verification Results

This subsection describes the verification results using the proposed method.

An automatic generation technique for the authentication and key exchange protocols was proposed in [34], in relation to this paper. In [34], 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 number 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 2 shows the verification results, best, worst and average (avg.) verification time, minimal (min.), maximal (max.) 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 "---" indicate 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. Using a PC with an Intel Pentium 4 2.6-GHz processor and 2.0-Gbyte RAM, the verification time is 110 [ms] or less for the 87 authentication and key exchange protocols. On the other hand, the fastest among the methods based on formal verification is TRUST [20], which takes 40 [ms] -- 1.8 [s] [35]. As examples of other methods, Murφ takes 1.7 [s] -- 5761.1 [s] [17], and Isabelle/HOL takes 52 [s] [27] and 150 [s] [28], respectively. We cannot make a precise comparison between the proposed method and other 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] -- 110 [ms], as can be seen from Table 2. Furthermore, the size of the protocol definition file is less than 14.2 [KB] in the aforementioned protocols and the program size is 1.25 [MB].

Table 2: Verification results in authentication and key exchange protocols.

Types	MC	SS				RODA	Num.	Verification Time [ms]			Protocol Definition File Size [KB]		
	SIA	SPA	SAA	KKS	WFS			Best	Worst	Avg.	Min.	Max.	Avg.
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

7. Conclusion

In this paper, we reconsidered the OKT method and proposed an updated security verification method for authentication and key exchange protocols based on the BPR model. We showed the novel verification points of one security property for authentication protocols, four security properties for key exchange protocols and a common security property for both protocols. We obtained the proposed method by resolving the three issues affecting the OKT method. Furthermore, we described the relations among the six verification points and explained a verification example that used 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 did not exceed 110 [ms] and that the security properties of the verification results completely coincided with the security requirements for the aforementioned protocols.

References

[1] H. Ota, S. Kiyomoto, and T. Tanaka, "Security verification for authentication and key exchange protocols, revisited," Proc. 2010 IEEE 24th IEEE International Conference on Advanced Information Networking and Applications Workshops (The Sixth International Symposium on Frontiers of Information Systems and Network Applications), pp.226--233, IEEE Computer Society Press, Perth, Australia, Apr. 2010.

[2] H. Ota, S. Kiyomoto, and T. Tanaka, "Fast and automatic verification of authentication and key exchange protocols," Proc. The Second International Conference on Advances in P2P Systems, pp.7--13, Xpert Publishing Services, Firenze, Italy, Oct. 2010.

[3] M. Bellare and P. Rogaway, "Entity authentication and key distribution," Advances in Cryptology --- CRYPTO'93, LNCS 773, pp.232--249, Springer-Verlag, Santa Barbara, CA, USA, Aug. 1993.

[4] M. Bellare and P. Rogaway, "Provably secure session key distribution --- The three party case," Proc. 27th Annual ACM Symposium on Theory of Computing, pp.57--66, ACM Press, Philadelphia, PA, USA, May 1995.

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

[6] M. Bellare, R. Canetti, and H. Krawczyk, "A modular approach to the design and analysis of authentication and key exchange protocols," Proc. 30th ACM Symposium on the Theory of Computing, pp.419--428, Dallas, TX, USA, May 1998.

[7] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," Proc. 42nd Symposium on Foundations of Computer Science (FOCS 2001), Las Vegas, NV, USA, Oct. 2001.

[8] D. Dolev and A. Yao, "On the security of public key protocols," Proc. IEEE 22nd Annual Symposium on Foundations of Computer Science, pp.350--357, Nashville, TN, USA, Oct. 1981.

[9] D. Dolev and A. Yao, "On the security of public key protocols," IEEE Trans. on Information Theory, Vol.29, No.2, pp.198--208, Mar. 1983.

[10] J. Millen, S. Clark, and S. Freeman, "The interrogator: Protocol security analysis," IEEE Trans. on Software Engineering, Vol.13, No.2, pp.274--288, Feb. 1987.

[11] C. Meadows, "Applying formal methods to the analysis of a key management protocol," Journal of Computer Security, Vol.1, No.1, pp.5--36, 1992.

[12] R. Kemmerer, C. Meadows, and J. Millen, "Three systems for cryptographic protocol analysis," Journal of Cryptology, Vol.7, No.2, pp.79--130, 1994.

[13] D. Longley and S. Rigby, "An automatic search for security flaws in key management schemes," Computers and Security, Vol.11, No.1, pp.75--89, Mar. 1992.

[14] J. Thayer, J. Herzog, and J. Guttman, "Strand spaces: Proving security protocols correct," Journal of Computer Security, Vol.7, No.2/3, pp.191--230, 1999.

[15] A. Roscoe, "Modelling and verifying key-exchange protocols using CSP and FDR," Proc. Eighth IEEE Computer Security Foundations Workshop, County Kerry, Ireland, pp.98--107, June 1995.

[16] G. Lowe, "Breaking and fixing the Needham-Schroeder public-key protocol using FDR," Software --- Concepts and Tools, Vol.17, No.3, pp.93--102, 1996.

[17] J. Mitchell, M. Mitchell, and U. Stern, "Automated analysis of cryptographic protocols using Murφ," Proc. 1997 IEEE Symposium on Security and Privacy, pp.141--151, IEEE Computer Society, Oakland, CA, USA, May 1997.

[18] M. Abadi and A. Gordon, "A calculus for cryptographic protocols: The spi calculus," Information and Computation, Vol.148, No.1, pp.1--70, Jan. 1999.

[19] G. Leduc and F. Germeau, "Verification of security protocols using LOTOS-method and application," Computer Communications, Vol. 23, No.12, pp.1089--1103, July 2000.

[20] 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, Elsevier Science, Jan. 2003.

[21] B. Blanchet, "A computationally sound mechanized prover for security protocols," Proc. 2006 IEEE Symposium on Security and Privacy, pp.140--154, IEEE Computer Society, Oakland, CA, USA, May 2006.

[22] B. Blanchet, "Computationally sound mechanized proofs of correspondence assertions," Proc. 20th IEEE Computer Security Foundations Symposium (CSF-20), pp.97--111, Venice, Italy, July 2007.

[23] M. Burrows, A. Abadi, and R. Needham, "A logic of authentication," ACM Trans. on Computer Systems, Vol.8, No.1, pp.18--36, Feb. 1990.

[24] L. Gong, R. Needham, and R. Yahalom, "Reasoning about belief in cryptographic protocols," Proc. 1990 IEEE Symposium on Security and Privacy, pp.234--248, IEEE Computer Society, Oakland, CA, USA, May 1990.

[25] P. Syverson and P. van Oorschot, "On unifying some cryptographic protocol logics," Proc. 1994 IEEE Computer

Society Symposium on Research in Security and Privacy, pp.14--28, IEEE Computer Society, Oakland, CA, USA, May 1994.
[26] L. Paulson, "Proving properties of security protocols by induction," Computer Laboratory Technical reports, No.409, Dec. 1996.
[27] L. Paulson, "Mechanized proofs of security protocols: Needham-Schroeder with public keys," Computer Laboratory Technical reports, No.413, Jan. 1997.
[28] L. Paulson, "Inductive analysis of the internet protocol TLS," Computer Laboratory Technical reports, No.440, Dec. 1997.
[29] L. Paulson, "The inductive approach to verifying cryptographic protocols," Journal of Computer Security, Vol.6, No.1/2, pp.85--128, 1998.
[30] K. Ogata and K. Futatsugi, "Formal verification of the Horn-Preneel micropayment protocol," Proc. 4th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI 2003), LNCS 2575, pp.238--252, Springer-Verlag, New York, NY, USA, Jan. 2003.
[31] K. Ogata and K. Futatsugi, "Equational approach to formal analysis of TLS," Proc. 25th IEEE International Conference on Distributed Computing Systems (ICDCS2005), pp.795--804, IEEE Computer Society, Columbus, OH, USA, June 2005.
[32] H. Ota, S. Kiyomoto, and T. Tanaka, "Security verification for authentication and key exchange protocols," Proc. 2008 International Symposium on Information Theory and its Applications, pp.507--512, Auckland, New Zealand, Dec. 2008.
[33] 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.
[34] S. Kiyomoto, H. Ota, and T. Tanaka, "Security protocol dynamic generation and modification mechanisms for ubiquitous services," Proc. 11th International Conference on Wireless Personal Multimedia Communications (WPMC'08), Lapland, Finland, Sept. 2008.
[35] A. Bracciali, G. Baldi, G. Ferrari, and E. Tuosto, "A coordination-based methodology for security protocol verification," Proc. 2nd International Workshop on Security Issues with Petri Nets and other Computational Models (WISP2004), Electronic Notes in Theoretical Computer Science, Vol.121, pp.23--46, Elsevier Science, June 2004.

Appendix A. Verification Points

This appendix presents detailed tables of the verification points described in Sect. 4.

Tables 3 -- 7 show the verification points of MC-SIA, SS-SPA, SS-SAA, SS-KKS, SS-WFS and RODA. Table 4 shows the common verification point of SS-SPA, SS-SAA and SS-KKS and Table 5 shows the remaining verification point of SS-SPA, where the verification points of SS-SAA and SS-KKS coincide with Table 4. The following abbreviations are used:

- 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.
- 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-SS with LLK-SS.
- EXC denotes elements except for PW-PS and PW-SS.

Table 3: 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 4: 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
SM	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	---

Table 5: Remaining verification points of SS-SPA.

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

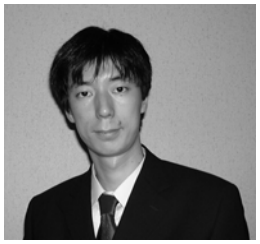
Table 6: Verification points of SS-WFS.

g	*f*	*A*	*B*	*C*
SM	PDH	T*-SS	---	LLK-PS
SKE	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	---	---
PDH	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 7: 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

Haruki Ota received his B.E. Department of Computer Science and M.E. Department of Communications and Integrated Systems from Tokyo Institute of Technology, Japan, in 2000 and 2002, respectively. He joined KDDI and has been engaged in research on cryptographic protocols, biometrics, and information security. He is currently a research engineer of the Information Security Laboratory in KDDI R&D Laboratories Inc. He received the IEICE Young Engineer Award in 2008. He is a member of IEICE and IPSJ.



Shinsaku Kiyomoto received his B.E. in Engineering Sciences and M.E. in Materials Science from Tsukuba University, Japan, in 1998 and 2000, respectively. He joined KDD (now KDDI) and has been engaged in research on stream ciphers, cryptographic protocols and mobile security. He is currently a senior research engineer at the Information Security Laboratory of KDDI R&D Laboratories Inc. He is a visiting researcher of the Information Security Group, Royal Holloway University of London from 2008. He received his Doctorate in Engineering from Kyushu University in 2006. He received the IEICE Young Engineer Award in 2004. He is a member of JPS, IEICE, and IPSJ.

Yutaka Miyake received his B.E. and M.E. degrees of Electrical Engineering from Keio University, Japan, in 1988 and 1990, respectively. He joined KDD (now KDDI) in 1990, and has been engaged in the research on high-speed communication protocol and secure communication system. He received his Dr. degree in engineering from the University of Electro-Communications, Japan, in 2009. He is currently a senior manager of Information Security Laboratory in KDDI R&D Laboratories Inc. He received IPSJ Convention Award in 1995 and the Meritorious Award on Radio of ARIB in 2003.