Security Verification for Authentication and Key Exchange Protocols

Haruki Ota[†] **Shinsaku Kiyomoto**[†]**, and Toshiaki Tanaka**[†]

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

Summary

In a ubiquitous environment, it is preferable for authentication and key exchange protocols to be optimized automatically in accordance with security requirements. In this paper, we propose a security verification method for authentication and key exchange protocols that is based on Bellare et al.'s model. In particular, we show the verification points of one security property for authentication protocols and five security properties for key exchange protocols. We show that this method is valid by verifying the security of four typical examples of the authentication and key exchange protocols and the 87 authentication and key exchange protocols which were generated automatically.

Key words:

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

1. Introduction

1.1 Motivation

In a ubiquitous environment, in which various terminals, devices, and networks are used, it is preferable for security protocols, such as authentication and key exchange protocols, to be provided according to the environment. However, in such a ubiquitous environment, the security protocol used in a high-performance PC cannot be applied to a low-power device. Also, security protocols are required to modify various security levels according to a variety of services that it is assumed will be available in future. These protocols cannot correspond to such various services flexibly at present, since they are implemented individually for every terminal or service.

On the other hand, for a considerable period, the existing security protocols were designed by trial and error, based on the designer's understanding of the security and cryptographic techniques. Therefore, it is necessary to deal with compromised protocols quickly. However, the process of designing security protocols by specialists is a time consuming one, and it takes a considerable amount of time to modify the protocol specification, or design and verify a new security protocol. Thus, there were neither the methods to evaluate the security protocols formally nor the mechanisms to deal with the compromised protocols quickly.

1.2 Related Work

Two 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. Bellare et al. formulated models that were secure against an off-line dictionary attack and forward secrecy. We call the model proposed in the papers [1], [2], and [3] the "BPR model" hereinafter. The BPR model became the basis of a considerable number of subsequent works in this area, such as those on a simulation paradigm and a universally composable framework. However, there is a problem that the security of the protocols still needs to be proved, respectively.

On the other hand, the 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. As examples of methods based on state-machine approaches, there are the Dolev-Yao model [6, 7], Interrogator [8], NRL (Naval Research Laboratory) Protocol Analyzer [9, 10], Longley-Rigby tool, and the strand space model [11]. As examples of methods using model checkers, there are FDR (Failures Divergences Refinement) / CSP (Communicating Sequential Processes) [13, 14] and Murφ [15]. As examples of methods using algebraic systems, there are spi calculus [16], LOTOS (Language of Temporal Ordering

Manuscript received March 5, 2009
Manuscript revised March 20, 2009

Specification) [17], STA (Symbolic Trace Analyzer) [18], TRUST [19], and CryptoVerif [20, 21]. As examples of methods based on modal logic, there are BAN (Burrows-Abadi-Needham) logic [22], GNY (Gong-Needham-Yahalom) logic [23], and SVO (Syverson-van Oorschot) logic [24]. As examples of methods based on inductive approaches, there are Isabelle/HOL (Higher Order Logic) [25, 26, 27, 28] and CafeOBJ [29, 30]. 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.

1.3 Contributions

In this paper, we propose a security verification method for authentication and key exchange protocols, based on the BPR model. First, we show the procedure for the proposed method, and set up each item with respect to cryptographic primitives and flow data used in the authentication and key exchange protocols. Next, we show verification points for each security requirement introduced for security reasons and proof techniques for the BPR model. In particular, we show the verification points of one security property for authentication protocols and five security properties for key exchange protocols. The proposed method is characterized by the fact that it can verify the security of authentication and key exchange protocols automatically more quickly than methods based on formal verification, since only these verification points are checked for the above protocols. We also show the validity of the proposed method by verifying the security of the authentication and key exchange protocols.

1.4 Organization

The rest of this paper is organized as follows. We introduce the BPR model in Sect. 2. The security verification method is proposed for authentication and key exchange protocols in Sect. 3, and the verification points of the security properties for these protocols are presented in Sect. 4. We show verification results using the proposed method in Sect. 5. Finally, our conclusions are in Sect. 6. We show flows of protocol examples in Appendix A and the relations of the verification of the proposed method and the proof for these examples in Appendix B.

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

(1) Matching conversation (MC)

In an authentication protocol, an adversary cannot alter messages, send other messages, intercept messages, or deliver messages out of order.

(a) Security against an active attack (MC-AAS)

An adversary cannot break an authentication protocol even when he/she controls all communications between parties. [Impersonation attack]

(2) Semantic security (SS)

In a key exchange protocol, an adversary cannot distinguish between the session key and random key.

(a) Security against an active attack (SS-AAS)

An adversary cannot break a key exchange protocol even when he/she controls all communications between parties. [Impersonation attack]

(b) Security against a passive attack (SS-PAS)

An adversary cannot break a key exchange protocol even when he/she eavesdrops on all communications between parties. [Eavesdropping attack]

(c) Security against an off-line dictionary attack (SS-DAS)

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]

(d) Security against a known key attack (SS-KAS)

An adversary cannot obtain a target session key even when he/she obtains session keys in other sessions. [Known key attack]

(e) Forward secrecy (SS-FS)

An adversary cannot obtain the past session key even when he/she obtains a long-lived key such as secret key of secret key encryption, password, or private key of public key encryption. [Corruption attack]

Authentication and key exchange protocols are provably secure in the BPR model when the matching conversation and semantic security are entirely achieved with respect to the above security properties. If the long-lived key is not used in the key exchange protocol, (2-e) is not required. In particular, if the password is not used in the key exchange protocol, (2-c) is not required either.

3. Security Verification Method

This section proposes a security verification method for authentication and key exchange protocols that is based on the BPR model.

3.1 Outline

This subsection provides an outline of the security verification method for authentication and key exchange protocols based on the BPR model.

In the BPR model, the security of protocols is reduced to that of some cryptographic primitives. We focus on these cryptographic primitives and their input and output values. The security reasons and types and states of arguments of the cryptographic primitives are analyzed in accordance with the flows and the data that are related to each attack, and then the verification points of each security property are introduced. In the proposed method, we check only these verification points for the authentication and key exchange protocols. Therefore, the proposed method can verify the security of protocols automatically more quickly than methods based on formal verification.

3.2 Procedure

This subsection describes the procedure of the security verification method for authentication and key exchange protocols.

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 above protocols:

- In the authentication and key exchange protocols, two parties share a secret key or password in a secure manner beforehand when the secret key or password is used.
- In the authentication and key exchange protocols, 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 the certificate authority, when the public key is used.
- The verification program (VG) can obtain and record the information of the compromised cryptographic primitives in real time.

The VG verifies the security of the authentication and key exchange protocols in the following manner:

(1) The VG enumerates all cryptographic primitives used in the authentication and key exchange protocols. The VG checks whether these cryptographic primitives are compromised or not. If compromised cryptographic primitives exist, then the VG judges that these protocols are not secure. Principal cryptographic primitives are classified as follows:

- Secret key encryption group
  - Secret key encryption (SKE)
  - Encryption using password (PWE)
- Public key encryption group
  - Public key encryption (PKE)
  - Diffie-Hellman family (DH)
  - Digital signature scheme (SIG)
- Hash function group
  - Hash function (HF)
  - Message authentication code scheme (MAC)

(2) The VG sets up the following among the cryptographic primitives enumerated in step (1) in the authentication and key exchange protocols:

- Cryptographic primitives required for authenticator generation in the authentication protocols (AGF)
- Cryptographic primitives required for key generation in the key exchange protocols (KGF)
- Cryptographic primitives included in the arguments of AGF or KGF (ACP)
- Cryptographic primitives that appear in flows (OCP)

The VG sets up the security reasons for the above cryptographic primitives. Table 1 shows the security reasons of principal cryptographic primitives, as follows:

- Indistinguishability (IND):
  An adversary cannot distinguish between two target events (without knowing the secret data).
- One-wayness (OW):
  An adversary cannot compute the input value from the output value (without knowing the secret data).
- Unforgeability (UF):
  An adversary cannot forge the required value without knowing the secret data.

Also, Symbols "Y", "y", and "N" show that the cryptographic primitives "have", "may have according to the situation", and "do not have" the characteristic of the corresponding security reasons, respectively.

Table 1: Security reasons for cryptographic primitives.

	SKE	PWE	PKE	DH	SIG	HF	MAC
IND	Y	y	Y	Y	y	y	y
OW	Y	Y	Y	y	N	Y	Y
UF	y	N	y	N	Y	N	Y

(3) The VG enumerates all flows in the authentication and key exchange protocols. It sets up the following elements about these flows in accordance with the protocol specifications:

- Types of flow data and arguments of cryptographic primitives
  - General public data (PUB)
  - Particular public data (ID)
  - Temporary public data (TPK)
  - Long-lived complete secret data (LLK)
  - Long-lived incomplete secret data (PW)

- Temporary secret data (TSK)

- States of flow data and arguments of cryptographic primitives
  - First appearance with the public state (PFT)
  - Existing appearance with the public state in the same session (PSS)
  - Existing repeated appearance with the public state in other sessions (PRS)
  - First appearance with the secret state (SFT)
  - Existing appearance with the secret state in the same session (SSS)
  - Existing repeated appearance with the secret state in other sessions (SRS)

Table 2 shows the relations between types and states of flow data and arguments of cryptographic primitives except for SS-FS. If there are incorrect relations except for SS-FS, then the VG judges that the authentication and key exchange protocols are not secure.

(4) The VG selects the security properties required for the authentication and key exchange protocols, as described in Sect. 2. Then, it sets up the security parameter required for these protocols, and confirms whether sizes of flow data and each data with respect to the cryptographic primitives are larger than or equal to this security parameter or not. If there are data sizes smaller than this security parameter, then the VG judges that the authentication and key exchange protocols are not secure.

Table 2: Relations between types and states of flows and arguments of cryptographic primitives.

	PUB	ID	TPK	LLK	PW	TSK
PFT	Y	Y	Y	N	N	N
PSS	Y	Y	Y	N	N	N
PRS	Y	Y	N	N	N	N
SFT	Y	Y	Y	Y	Y	Y
SSS	Y	Y	Y	Y	Y	Y
SRS	Y	Y	N	Y	Y	N

(5) The VG checks the verification points shown in Sect. 4 using the elements of step (3) for the security properties of step (4) in the authentication and key exchange protocols. Then, the VG sets up the data that are related to each attack. It sets up the elements of step (3) and the security properties of step (4) in accordance with the order of the protocol flows for these data. Here, the states of flow data and arguments of cryptographic primitives 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 Authentication Protocol

This subsection shows the verification points of the security property for the authentication protocol.

4.1.1 Security Against an Active Attack (MC-AAS)

The security reason of MC-AAS for the authentication protocol in order that an adversary cannot generate a valid authenticator by launching an impersonation attack is as follows:

(a_1) AGF has UF or OW.

The VG sets up all flows due to the impersonation attack. Then, the following are the verification points of the security against an active attack, since the authentication protocol needs to have the security reason (a_1) of MC-AAS:

(a_{11}) AGF is SKE, PKE, SIG or MAC, and the arguments of AGF contain TPK-PFT, TPK-PSS, TSK-SFT, or TSK-SSS.

(a_{12}) AGF is SKE, PWE, PKE, DH, or HF, and the arguments of AGF contain TSK-SFT or TSK-SSS.

4.2 Key Exchange Protocol

This subsection shows the verification points of the security properties for the key exchange protocol.

4.2.1 Security Against an Active Attack (SS-AAS)

The security reasons of SS-AAS for the key exchange protocol in order that an adversary cannot distinguish between the session key and random key by launching an impersonation attack are as follows:

(a_1) KGF has IND or OW.

(a_2) ACP of KGF has IND or OW.

The VG sets up all flows due to the impersonation attack. Then, the following are the verification points of the security against an active attack, since the key exchange protocol needs to have the security reasons (a_1) or (a_2) of SS-AAS:

(a_{11}) KGF is SKE, PWE, PKE, SIG, HF, or MAC, and the arguments of KGF contain (TPK-PFT or TSK-SFT) and (LLK-SRS or PW-SRS).

(a_{12}) KGF is SKE, PWE, or PKE, and the arguments of KGF contain (TPK-PFT or TPK-PSS) and TSK-SFT.

(a_{21}) ACP of KGF is SKE, PWE, PKE, HF, or MAC, and the arguments of ACP of KGF contain (TPK-PFT or TSK-SFT) and (LLK-SRS or PW-SRS).

(a_{22}) ACP of KGF is DH, and the arguments of ACP of KGF contain TSK-SFT.

(a_{23}) ACP of KGF is SKE, PWE, or PKE, and the arguments of ACP of KGF contain (TPK-PFT or TPK-PSS) and TSK-SFT.

4.2.2 Security Against a Passive Attack (SS-PAS)

The security reasons of SS-PAS for the key exchange protocol in order that an adversary cannot distinguish between the session key and random key by launching an eavesdropping attack are as follows:
(b_1) KGF has IND.
(b_2) KGF has OW, and ACP of KGF has IND.
The VG sets up all flows due to the eavesdropping attack. Then, the following are the verification points of the security against a passive attack, since the key exchange protocol needs to have the security reasons (b_1) or (b_2) of SS-PAS:

(b_{11}) KGF is SKE, PKE, or DH, and the arguments of KGF contain TSK-SFT.

(b_{12}) KGF is SKE, PWE, HF, or MAC, and the arguments of KGF contain (TPK-PFT or TSK-SFT) and (LLK-SRS or PW-SRS).

(b_{21}) KGF is SKE, PWE, PKE, HF, or MAC, ACP of KGF is SKE, PKE, or DH, and the arguments of ACP of KGF contain TSK-SFT or TSK-SSS.

4.2.3 Security Against an Off-line Dictionary Attack (SSDAS)

The security reason of SS-DAS for the key exchange protocol in order that an adversary cannot search for a password of a party by launching an off-line dictionary attack is as follows:
(c_1) OCP whose arguments contain PW has OW.
The VG sets up all flows due to the off-line dictionary attack. Then, the following are the verification points of the security against an off-line dictionary attack, since the key exchange protocol needs to have the security reason (c_1) of SS-DAS:

(c_{11}) OCP whose arguments contain PW-SRS is SKE or MAC.

(c_{12}) OCP whose arguments contain PW-SRS is PWE, PKE, or HF, and the arguments of OCP contain TSK-SFT or TSK-SSS.

4.2.4 Security Against a Known Key Attack (SS-KAS)

The security reason of SS-KAS for the key exchange protocol in order that an adversary cannot obtain a target session key by launching a known key attack is as follows:
(d_1) KGF has IND or OW.
The VG sets up session keys in all other sessions due to the known key attack. Then, the following are the verification points of the security against a known key attack, since the key exchange protocol needs to have the security reason (d_1) of SS-KAS:

(d_{11}) KGF is SKE, PWE, PKE, DH, HF, or MAC, and the arguments of KGF contain TSK-SFT or TSK-SSS.

(d_{12}) KGF is SKE, PWE, HF, or MAC, and the arguments of KGF contain (TPK-PFT or TSK-SFT) and (LLK-SRS or PW-SRS).

4.2.5 Forward Secrecy (SS-FS)

The security reasons of SS-FS for the key exchange protocol in order that an adversary cannot obtain the past session key by launching a corruption attack are as follows:
(e_1) KGF has IND or OW.
(e_2) KGF has OW, and ACP of KGF has IND.
The VG sets up long-lived keys due to the corruption attack. Then, the following are the verification points of the forward secrecy, since the key exchange protocol needs to have the security reasons (e_1) or (e_2) of SS-FS:
(e_{11}) KGF is SKE, PWE, PKE, DH, HF, or MAC, and the arguments of KGF contain TSK-SFT or TSK-SSS.
(e_{21}) KGF is HF, ACP of KGF is DH, and the arguments of ACP of KGF contain TSK-SFT or TSK-SSS.

Remark 1: We showed the verification points of one security property 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. Evaluation

This section shows the verification results using the method proposed in Sect. 3 and 4.

5.1 Verification Results for Typical Protocols

This subsection shows the verification results for four typical examples of the authentication and key exchange protocols.

We verify the security of the following authentication, key exchange, and authenticated key exchange protocols, using the security verification method:

- Authentication protocol, MAP1 [1]
- Authenticated key exchange protocol, AKEP1 [1]
- Key exchange protocol, EKE2 [3]
- Authenticated key exchange protocol, AddMA(EKE2) [3]

We show the above protocol flows in Appendix A. Then, the cryptographic primitives and the types and states of flow data and arguments of cryptographic primitives are set up for each protocol as follows, respectively, where the classifications of cryptographic primitives are shown in brackets:

- MAP1
  - AGF = {f_a [MAC]}
  - OCP = {f_a [MAC]}
  - ID-PRS = {A, B}
  - LLK-SRS = {a}
  - TPK-PSS = {R_A, R_B}
- AKEP1
  - AGF = {f_{a1} [MAC]}
  - KGF = {$f'_{a2}(r)$ XOR y [SKE]}
  - OCP = {f_{a1} [MAC], $f'_{a2}(r)$ XOR y [SKE]}
  - ID-PRS = {A, B}
  - LLK-SRS = {a_1, a_2}
  - TPK-PSS = {R_A, R_B}
  - TSK-SFT = {α}
- EKE2
  - KGF = {H [HF]}
  - ACP of KGF = {g^{xy} [DH]}
  - OCP = {E_{pw} [PWE]}

  Except for SS-FS,
  - ID-PRS = {A, B}
  - PW-SRS = {pw}
  - TSK-SFT = {x, y, g^x, g^y}

  For SS-FS,
  - ID-PRS = {A, B}
  - PW-PRS = {pw}
  - TPK-PFT = {g^x, g^y}
  - TSK-SFT = {x, y}
- MAEKE2
  - AGF = {H [HF]}
  - ACP of AGF = {g^{xy} [DH]}
  - KGF = {H [HF]}
  - ACP of KGF = {g^{xy} [DH]}
  - OCP = {E_{pw} [PWE], g^{xy} [DH], H [HF]}

  Except for SS-FS,
  - PUB-PRS = {0, 1, 2}
  - ID-PRS = {A, B}
  - PW-SRS = {pw}
  - TSK-SFT = {x, y}
  - TSK-SSS = {g^x, g^y}

  For SS-FS,
  - PUB-PRS = {0, 1, 2}
  - ID-PRS = {A, B}
  - PW-PRS = {pw}
  - TPK-PSS = {g^x, g^y}
  - TSK-SFT = {x, y}

Here, we briefly explain the relation of the security verification of the proposed method and the actual security proof by taking as examples the protocols MAP1 and AKEP1 from among the above four protocols.

First, the following theorem with respect to MAP1 was proven by Bellare et al.

Theorem 1: Suppose f is a pseudorandom function family. Then protocol MAP1 described above and based on f is a secure mutual authentication protocol. □

Refer to the literature [1] for details of the proof. We explain the relation of the security verification of the proposed method and the security proof of MAP1. First, the function f_a has unforgeability and the key a must not be known by the adversary. Therefore, it is required that f_a has the role of a secure MAC function and a is LLK-SRS. Next, the random numbers R_A and R_B must not be known and used by the adversary beforehand. That is, R_A and R_B must be neither TPK-PRS nor TSK-SRS that can be used repeatedly in other sessions. Thus, they need to be elements except for the above, such as TPK-PSS. Consequently, MAP1 is a secure authentication protocol against an active attack, since the above corresponds to the verification point (a_{11}) of MC-AAS. In addition, the sizes of R_A and R_B are larger than or equal to the set-up security parameter, as described in step (4) of Sect. 3.2.

Next, the following theorem with respect to AKEP1 was proven by Bellare et al.

Theorem 2: Let $S = \{S_k\}_k$ be samplable, and suppose f; f' are pseudo-random function families with the parameters specified in the literature [1]. Then the protocol AKEP1 based on f; f' is a secure authenticated key exchange protocol over S. □

Refer to the literature [1] for details of the proof. We explain the relation of the security verification of the proposed method and the security proof of AKEP1. First, the function f'_{a2} has unforgeability and the key $a2$ must not be known by the adversary. Therefore, it is required that f'_{a2} has the role of a secure secret key encryption and $a2$ is LLK-SRS. Next, the session key α must not be known and used repeatedly by the adversary. That is, α must not be TPK-PFT, TPK-PSS, or TPK-PRS, which are public states, or TSK-SRS, which can be used repeatedly in other sessions. Thus, they need to be elements except for the above, such as TSK-SFT. Consequently, AKEP is a secure authentication protocol against an active attack, a passive attack, and a known key attack, since the above corresponds to the verification points (a_{11}) of SS-AAS, (b_{11}) of SS-PAS, and (d_{11}) of SS-KAS. In addition, from Theorem 1, AKEP1 is a secure authentication protocol against an active attack.

Table 3 shows the verification results and processing time for the protocol examples, where the alphabetic symbols denote the verification points that guarantee the security for each protocol, and the unit of processing time is the millisecond ([ms]). Symbols “Y” and “---” denote that the protocol “meets” and “does not require” the corresponding security property, respectively. These results completely coincide with the security requirements for each protocol. Also, the processing time is within 15 [ms] in the four authentication and key exchange protocols, using a PC with an Intel Core 2 Duo 3.0-GHz processor and 2.0-Gbyte RAM.

Table 3: Verification results and processing time for each protocol.

	MC	SS					Time
	AAS	AAS	PAS	DAS	KAS	FS	
MAP1	Y (a_{11})	---	---	---	---	---	3.098
AKEP1	Y (a_{11})	Y (a_{11})	Y (b_{11})	---	Y (d_{11})		5.761
EKE2	---	Y (a_{22})	Y (b_{21})	Y (c_{12})	Y (d_{11})	Y (e_{21})	4.667
AddMA (EKE2)	Y (a_{12})	Y (a_{22})	Y (b_{21})	Y (c_{12})	Y (d_{11})	Y (e_{21})	11.682

5.2 Verification Results for Automatically Generated Protocols

This subsection shows the verification results for the authentication and key exchange protocols which were generated automatically.

An automatic generation technique of the authentication and key exchange protocols was proposed in the literature [31], in relation to this paper. Eighty-seven types of authentication and key exchange protocols (15 authentication, 22 key exchange, and 50 authenticated key exchange protocols) were automatically generated using the automatic generation tool shown in the literature. Then, we verified the security of the above authentication, key exchange, and authenticated key exchange protocols, using the proposed method.

Tables 4, 5, and 6 show the verification results and average processing time for the authentication, key exchange, and authenticated key exchange protocols, respectively, where the unit of the average processing time is the millisecond. 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 automatically generated protocols. The items “F”, “D”, and “I” denote the numbers of flows, data, and function instances used in each protocol, respectively. Also, the item “P” denotes the number of protocols in which all the numbers of flows, data, and function instances are the same. Furthermore, the processing time is within 10 [ms] in the 87 authentication and key exchange protocols, using the PC shown in Sect.5.1.

Table 4: Verification results and average processing time in authentication protocols.

MC	SS					The Number				Time
AAS	AAS	PAS	DAS	KAS	FS	F	D	I	P	
Y	---	---	---	---	---	2	3	2	2	1.423
						2	5	2	2	1.716
						2	6	2	1	1.986
						2	8	2	1	2.125
						3	3	2	2	1.519
						3	3	4	1	2.120
						3	5	2	2	2.040
						3	5	6	1	3.121
						3	6	2	2	1.925
						3	8	2	1	2.535

Table 5: Verification results and average processing time in key exchange protocols.

MC	SS					The Number				Time
AAS	AAS	PAS	DAS	KAS	FS	F	D	I	P	
---	Y	Y	---	N	N	1	2	3	2	0.765
						1	3	4	1	1.538
---	Y	Y	---	Y	N	2	3	2	1	1.510
						2	3	3	3	1.350
						2	4	3	1	1.608
---	Y	Y	---	Y	Y	2	5	3	1	2.232
						2	5	4	1	1.737
						2	5	5	2	3.011
						2	6	3	1	2.912
						2	6	5	2	3.422
						2	7	3	1	3.099
						2	7	5	1	3.668
						2	8	3	1	2.997
						2	8	4	1	2.206
						2	8	5	1	3.613
---	Y	Y	Y	Y	N	2	3	5	1	2.144
---	Y	Y	Y	Y	Y	2	5	7	1	2.988

Table 6: Verification results and average processing time in authenticated key exchange protocols.

MC	SS					The Number				Time
AAS	AAS	PAS	DAS	KAS	FS	F	D	I	P	
Y	Y	Y	---	Y	N	2	4	4	2	2.479
						2	5	5	1	3.855
						2	6	4	2	3.385
						2	7	5	1	5.139
						3	4	3	1	2.900
						3	4	4	4	2.652
						3	5	4	2	3.115
						3	6	3	1	3.579
						3	6	4	5	3.278
						3	7	4	1	3.652
Y	Y	Y	---	Y	Y	3	6	4	1	3.727
						3	6	6	2	4.707
						3	7	4	1	4.448
						3	7	5	1	3.567
						3	7	6	2	5.384
						3	8	4	2	4.562
						3	8	6	4	5.268
						3	9	4	1	4.920
						3	9	5	1	4.271
						3	9	6	2	6.079
						3	10	4	1	5.138
						3	10	6	2	5.794
						3	11	6	1	5.576
						3	13	5	1	6.761
						3	13	6	1	6.127
						3	13	7	1	7.613
						3	14	6	1	6.038
						3	16	5	1	8.191
						3	16	6	1	6.885
						3	16	7	1	9.339
Y	Y	Y	Y	Y	N	3	5	7	1	4.751
Y	Y	Y	Y	Y	Y	3	5	9	1	4.626

6. Conclusions

In this paper, we proposed a security verification method for authentication and key exchange protocols, based on the BPR model. We showed the verification points of one security property for authentication protocols and five security properties for key exchange protocols. We verified the security of four typical examples of the authentication and key exchange protocols and the 87 authentication and key exchange protocols, which were generated automatically. Then, we confirmed that the verification results completely coincide with the security requirements for each protocol. On the other hand, the methods based on formal verification, such as STA [18] and TRUST [19], take forty [ms] at the fastest to the best of our knowledge [32]. 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, we confirmed that the proposed method can verify the security of each protocol automatically and more quickly than most existing methods.

Acknowledgments

A part of this work was supported by the National Institute of Information and Communications Technology (NICT).

References

[1] 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.

[2] 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 1996.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] M. Boreale, “Symbolic trace analysis of cryptographic protocols,” Proc. 28th International Colloquium on Automata, Languages and Programming (ICALP 2001), LNCS 2076, pp.667--681, Springer-Verlag, Crete, Greece, July 2001.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] L. Paulson, “Proving properties of security protocols by induction,” Computer Laboratory Technical reports, No.409, Dec. 1996.

[26] L. Paulson, “Mechanized proofs of security protocols: Needham-Schroeder with public keys,” Computer Laboratory Technical reports, No.413, Jan. 1997.

[27] L. Paulson, “Inductive analysis of the internet protocol TLS,” Computer Laboratory Technical reports, No.440, Dec. 1997.

[28] L. Paulson, “The inductive approach to verifying cryptographic protocols,” Journal of Computer Security, Vol.6, No.1/2, pp.85--128, 1998.

[29] 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.

[30] 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.

[31] 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.

[32] 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. Protocol Flows

This Appendix show the protocol flows of MAP1, AKEP1, EKE2, and AddMA(EKE2).

Figures 1, 2, 3, and 4 show the protocol flows of MAP1, AKEP1, EKE2, and AddMA(EKE2), respectively. In MAP1 of Fig. 1, $[x]_a$ denotes $x \parallel f_a(x)$ and f is a pseudorandom function family. In AKEP1 of Fig. 2, $\sigma(k)$ is some polynomial, $[x]_{a1}$ denotes $x \parallel f_{a1}(x)$, $\{y\}_{a2}$ denotes $r \parallel f'_{a2}(r)$ XOR y with r selected at random, and f and f' are pseudorandom function families. In EKE2 of Fig. 3 and AddMA(EKE2) of Fig. 4, $E_{pw}(a)$ denotes encryption of a by a password pw. Refer to the literature [1] and [3] for the details of the protocol specifications.

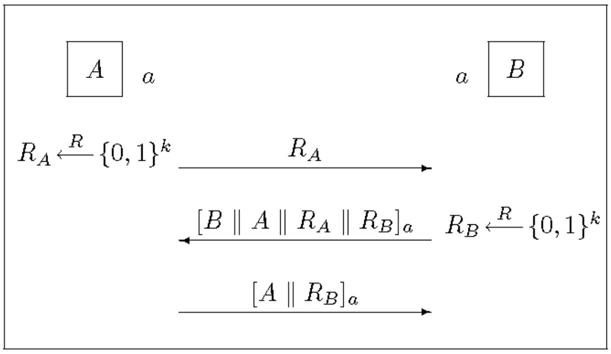

A
a
a
B
$R_A \xleftarrow{R} \{0,1\}^k$
R_A
$[B \parallel A \parallel R_A \parallel R_B]_a$
$R_B \xleftarrow{R} \{0,1\}^k$
$[A \parallel R_B]_a$

Fig. 1: Protocol MAP1 [1].

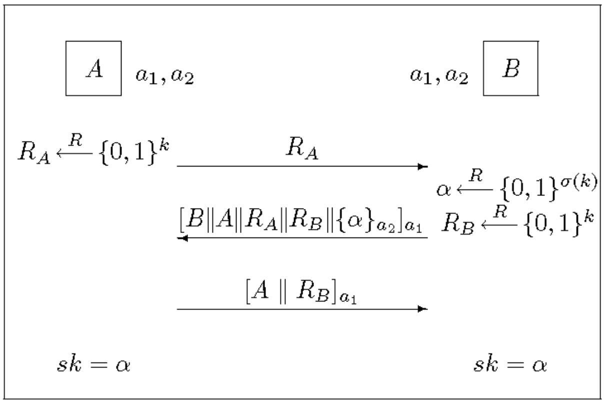

A
a_1, a_2
a_1, a_2
B
$R_A \overset{R}{\longleftarrow} \{0,1\}^k$
R_A
$\alpha \overset{R}{\longleftarrow} \{0,1\}^{\sigma(k)}$
$[B \| A \| R_A \| R_B \| \{\alpha\}_{a_2}]_{a_1}$
$R_B \overset{R}{\longleftarrow} \{0,1\}^k$
$[A \parallel R_B]_{a_1}$
$sk = \alpha$
$sk = \alpha$

Fig. 2: Protocol AKEP1 [1].

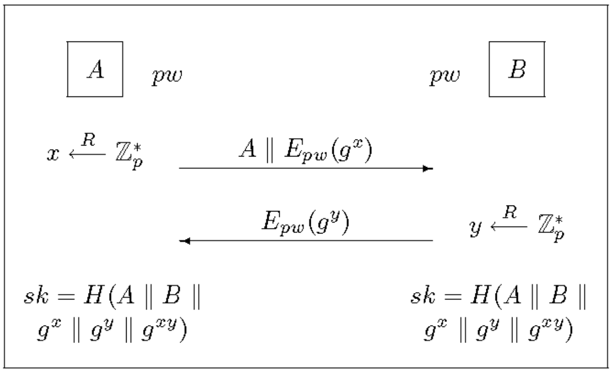

A
pw
pw
B
$x \overset{R}{\longleftarrow} \mathbb{Z}_p^*$
$A \parallel E_{pw}(g^x)$
$E_{pw}(g^y)$
$y \overset{R}{\longleftarrow} \mathbb{Z}_p^*$
$sk = H(A \parallel B \parallel g^x \parallel g^y \parallel g^{xy})$
$sk = H(A \parallel B \parallel g^x \parallel g^y \parallel g^{xy})$

Fig. 3: Protocol EKE2 [3].

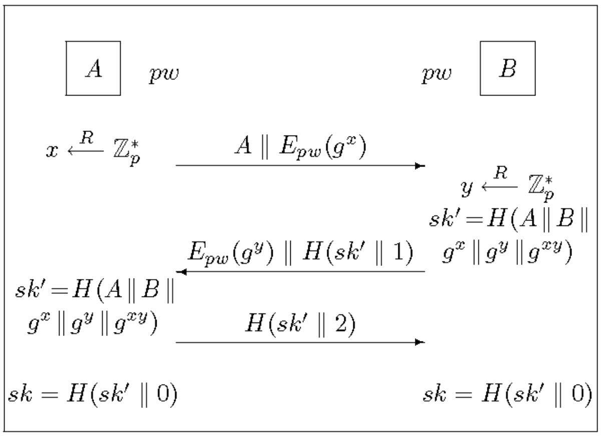

A
pw
pw
B
$x \overset{R}{\longleftarrow} \mathbb{Z}_p^*$
$A \parallel E_{pw}(g^x)$
$y \overset{R}{\longleftarrow} \mathbb{Z}_p^*$
$sk' = H(A \parallel B \parallel g^x \parallel g^y \parallel g^{xy})$
$E_{pw}(g^y) \parallel H(sk' \parallel 1)$
$sk' = H(A \parallel B \parallel g^x \parallel g^y \parallel g^{xy})$
$H(sk' \parallel 2)$
$sk = H(sk' \parallel 0)$
$sk = H(sk' \parallel 0)$

Fig. 4: Protocol AddMA(EKE2) [3].

Appendix B. Relations of Verification and Proof

This Appendix show the relations of the security verification of the proposed method and the actual security proof with respect to each protocol.

Tables 7, 8, and 9 show the relations of the security verification of the proposed method and the security proof for each verification point with respect to MAP1, AKEP1, and EKE2 and AddMA(EKE2), respectively, where MC-AAS is not required in EKE2. The items “Points”, “Proof”, and “Verification” denote the verification points of the proposed method, the summary of the security proof by Bellare et al., and the elements for the security verification of the proposed method.

Table 7: Relation of proof and verification for MAP1.

Points	Proof	Verification
MC-AAS	An adversary cannot forge $[B \parallel A \parallel R_A \parallel R_B]_a$ and $[A \parallel R_B]_a$ without knowing R_A and R_B beforehand as well as a.	f_a: UF[MAC] R_A: TPK-PSS R_B: TPK-PSS a: LLK-SRS

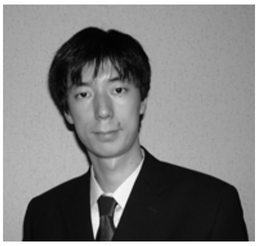
Table 8: Relation of proof and verification for AKEP1.

Points	Proof parts	Verification
MC-AAS	An adversary cannot forge $[B \parallel A \parallel R_A \parallel R_B \parallel \{\alpha\}_{a2}]_{a1}$ and $[A \parallel R_B]_{a1}$ without knowing R_A and R_B beforehand as well as a_1.	f_{a1}: UF[MAC] R_A: TPK-PSS R_B: TPK-PSS a_1: LLK-SRS
SS-AAS	An adversary cannot distinguish α and a random key from $\{\alpha\}_{a2}$ and the information about the impersonation attack without knowing α and a_2.	f_{a2}: IND[SKE] a_2: LLK-SRS α: TSK-SFT
SS-PAS	An adversary cannot distinguish α and a random key from $\{\alpha\}_{a2}$ and flow data without knowing α and a_2.	f_{a2}: IND[SKE] α: TSK-SFT
SS-KAS	An adversary cannot distinguish α and a random key from $\{\alpha\}_{a2}$ and session keys in all other sessions without knowing α and a_2.	f_{a2}: IND[SKE] α: TSK-SFT

Table 9: Relation of proof and verification for EKE2 and AddMA(EKE2).

Points	Proof parts	Verification
MC-AAS	An adversary cannot forge $H(sk' \| 1)$ and $H(sk' \| 2)$ without knowing *pw*, *x* and *y*.	*H*: OW[HF] *x*: TSK-SFT *y*: TSK-SFT
SS-AAS	An adversary cannot distinguish *sk* and a random key from and the information about impersonation attack without knowing *pw*, *x*, and *y*.	g^{xy}: IND[DH] *x*: TSK-SFT *y*: TSK-SFT
SS-PAS	An adversary cannot distinguish *sk* and a random key flow data without knowing *pw*, *x*, and *y*.	*H*: OW[HF] g^{xy}: IND[DH] *x*: TSK-SFT *y*: TSK-SFT
SS-DAS	An adversary cannot obtain *pw* from flow data without knowing *pw*, *x*, and *y*.	E_{pw}: OW[PWE] *x*: TSK-SFT *y*: TSK-SFT
SS-KAS	An adversary cannot distinguish *sk* and a random key from session keys in all other sessions without knowing *pw*, *x*, and *y*.	*H*: IND[HF] *x*: TSK-SFT *y*: TSK-SFT
SS-FS	An adversary cannot distinguish the past *sk* and a random key from the information about the corruption attack without knowing *x* and *y* even if he/she knows *pw*.	*H*: OW[HF] g^{xy}: IND[DH] *x*: TSK-SFT *y*: TSK-SFT

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 Information Security Laboratory in KDDI R&D Laboratories Inc. He received the Young Engineer Award from IEICE 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 researcher of the Information Security Laboratory in KDDI R&D Laboratories Inc. He received his doctorate of engineering from Kyushu University in 2006. He received the Young Engineer Award from IEICE in 2004. He is a member of JPS, IEICE, and IPSJ.

Toshiaki Tanaka received B.E. and M.E. degrees in communication engineering from Osaka University, Japan, in 1984 and 1986 respectively. He joined KDD (now KDDI) and has been engaged in research on cryptographic protocols, mobile security, digital rights management, and intrusion detection. He is currently a senior manager of the Information Security Laboratory in KDDI R&D Laboratories Inc. He received his doctorate of engineering from Kyushu University in 2007. He is a member of IEICE and IPSJ.