RESEARCH ARTICLE

Three-party authenticated key agreements for optimal communication

Tian-Fu Lee¹, Tzonelih Hwang^{2*}

1 Department of Medical Informatics, Institute of Medical Sciences, Tzu Chi University, Hualien, Taiwan, ROC, **2** Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, ROC

* hwangtl@ismail.csie.ncku.edu.tw

Abstract

Authenticated key agreements enable users to determine session keys, and to securely communicate with others over an insecure channel via the session keys. This study investigates the lower bounds on communications for three-party authenticated key agreements and considers whether or not the sub-keys for generating a session key can be revealed in the channel. Since two clients do not share any common secret key, they require the help of the server to authenticate their identities and exchange confidential and authenticated information over insecure networks. However, if the session key security is based on asymmetric cryptosystems, then revealing the sub-keys cannot compromise the session key. The clients can directly exchange the sub-keys and reduce the transmissions. In addition, authenticated key agreements were developed by using the derived results of the lower bounds on communications. Compared with related approaches, the proposed protocols had fewer transmissions and realized the lower bounds on communications.

OPEN ACCESS

Citation: Lee T-F, Hwang T (2017) Three-party authenticated key agreements for optimal communication. PLoS ONE 12(3): e0174473. <https://doi.org/10.1371/journal.pone.0174473>

Editor: Muhammad Khurram Khan, King Saud University, SAUDI ARABIA

Received: December 5, 2016

Accepted: March 9, 2017

Published: March 29, 2017

Copyright: © 2017 Lee, Hwang. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper.

Funding: This research was supported by Ministry of Science and Technology under the grants MOST 105-2221-E-320-003 and by Tzu Chi University under the grants TCRPP105004. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Introduction

Authenticated key agreements (AKA) enable users to exchange confidential and authenticated information over an insecure network, and to establish a common key that can be employed to encrypt all communications over an insecure channel. In an AKA protocol, each communicating entity that wants to determine session keys is assured of the identity of each of the others to provide mutual authentication. In terms of realizing mutual authentication, AKA protocols can be divided into two types—implicit mutual authentication and explicit mutual authentication. An AKA protocol with implicit mutual authentication realizes mutual authentication in later communications. However, it is not possible to be certain how protocol participants will use the session key. In contrast, an AKA protocol with explicit mutual authentication (AKA-MA) realizes mutual authentication while executing the protocol [1].

The AKA protocols mainly focus on providing higher security and developing transmission efficiency. Numerous factors influence transmission efficiency. Aside from the computational complexity of an authentication protocol, message efficiency and round efficiency are two important evaluation criteria. Message efficiency considers the number of messages required

to complete the protocol. A message is a data item sent from one party to a single destination at a particular time. Round efficiency considers the number of rounds required to complete the protocol. A round comprises all of the independent messages that can be sent and received in parallel [2,3].

Three-party authenticated key agreement (3AKA) protocol enables two users to agree a common session key for establishing a secure channel via the help of a trusted server. Recently, several approaches involving 3AKA-MA protocols have been presented. For instance, Gong et al. [2–4] provided lower bounds on communications for 3AKA-MA, which required five messages and four rounds. They also developed 3AKA-MA protocols to realize these lower bounds [2–4]. Kwon et al. [5–8] presented password-based 3AKA-MA protocols. In addition, some 3AKA-MA approaches have modified the structures of session keys to ensure perfect forward secrecy. For instance, the 3AKA-MA protocols in [3–13] based on the Diffie-Hellman problem [14] could provide perfect forward secrecy. Lee et al. [15] developed a 3AKA-MA based on chaotic maps without password table. Amin et al. [16] proposed anonymity preserving three-factor authenticated key exchange protocol for wireless sensor network. With reference to transmission, all of the 3AKA-MA protocols described above and other related secure approaches [14, 17–26] involve at least five messages or four rounds.

For 3AKA-MA protocols, few studies on the lower bounds on communication have been presented up to now, except for the investigation of Gong in [2,3]. However, Gong only considered this issue for conventional 3AKA-MA protocols, without ever completely discussing 3AKA-MA protocols. In 3AKA-MA protocols, two clients do not share any common secret key. Thus, they require the help of the server to authenticate the participants' identities and exchange confidential and authenticated information over an insecure network. In conventional 3AKA-MA protocols, the sub-keys for generating a session key cannot be revealed in the channel. Clients must exchange their sub-keys with the help of the server to establish an authentication key (session key). Accordingly, a conventional 3AKA-MA protocol requires at least five messages and four rounds [2, 3]. However, if the session key is based on asymmetric cryptosystems, such as the Diffie-Hellman key exchange or the Elliptic Curve Diffie-Hellman key exchange, then revealing the sub-keys for generating the session key cannot compromise the session key. The clients can directly exchange the sub-keys without using the server, and thus the number of messages and rounds can be reduced.

This study investigated the rules according to the behavior patterns of AKA-MA protocols, and then derived the lower bounds of communications for 3AKA-MA protocols based on these rules. In addition, we used the derived results to develop communication-efficient 3AKA-MA protocols, including conventional 3AKA-MA protocols whose sub-keys cannot be revealed in the channel and 3AKA-MA protocols in which revealing the sub-keys cannot compromise the session key. The proposed conventional 3AKA-MA protocols require five messages and four rounds of communication and realize the lower bounds on the number of messages and rounds for conventional 3AKA-MA protocols. On the other hand, in the proposed 3AKA-MA protocols, the session key security is based on the Diffie-Hellman problem [14]. Revealing the information $g^x \bmod p$ and $g^y \bmod p$ for generating the session key ($g^{xy} \bmod p$) cannot compromise the session key itself because the session key cannot be determined without a knowledge of x or y , where p is a large prime. Therefore, the clients can publicly exchange the information $g^x \bmod p$ and $g^y \bmod p$ for generating the session key without the help of the server. Using this technique, the proposed protocol reduced the number of messages and rounds and required only four messages and three rounds of communications. Hence, the proposed 3AKA-MA protocol also realized the proposed lower bounds on the number of messages and rounds for 3AKA-MA protocols. Furthermore, the proposed 3AKA-MA protocols were proven secure [27–31] and have AKE security and MA security.

Compared with related 3AKA-MA protocols, the proposed protocols were more efficient in communications, realized the lower bounds on the number of messages and rounds for 3AKA-MA protocols, and were suitable for practical environments.

This study is organized as follows. Section 2 describes the underlying primitives used in this investigation. Section 3 derives and proves the lower bounds on messages and rounds for 3AKA-MA protocols. Section 4 develops communication-efficient 3AKA-MA protocols based on the derived results from Section 3. All of the proposed protocols realize the lower bounds on the number of messages and rounds of communications. Section 5 provides security analyses and compares the performance of the proposed 3AKA-MA protocols with related protocols. Finally, Section 6 draws conclusions.

Preliminaries

This section describes the underlying primitives used in this paper. The underlying primitives include session key security, mutual authentication security, the authenticator, the chosen ciphertext secure symmetric-key encryption, the Diffie-Hellman assumptions, and the cryptographic hash functions.

AKE security (session key security)

In this security definition, the adversary is allowed to ask many *Test* queries as it wants. If a *Test* query is asked to a client instance that has not *accepted*, then return the invalid symbol $\perp$. If a *Test* query is asked to an instance of an honest participant whose intended partner is dishonest or to an instance of a dishonest participant, then returns the real session key. Otherwise, the *Test* query decides to return either the real session key or a random string via an unbiased coin c . The adversary aims to correctly guess the value of the hidden bit c used by the Test oracle. Let E denote the event that the adversary wins this game. The *ake-advantage* of the event that an adversary violates the indistinguishability of the protocol $\mathbf{P}$ $Adv_p^{ake}(\mathcal{A})$. The protocol $\mathbf{P}$ is AKE-secure if $Adv_p^{ake}(\mathcal{A})$ is negligible. [27]

Mutual Authentication (MA) security

In executing protocol $\mathbf{P}$, the adversary $\mathcal{A}$ violates mutual authentication if $\mathcal{A}$ can fake the authenticator μ_A or μ_B . The probability of this event is denoted by $Adv_p^{ma}(\mathcal{A})$. The protocol $\mathbf{P}$ is MA-secure if $Adv_p^{ma}(\mathcal{A})$ is negligible.

Authenticator

Additional information appended to a message to enable the receiver to verify that the message should be accepted as authentic. For AKA-MA protocols, an authenticator is used for the receiver to assure that the sender has the common session key. [32]

Chosen ciphertext secure symmetric-key encryption

For a symmetric-key encryption scheme, the CCA-advantage of the adversary $\mathcal{A}$ is the probability that $\mathcal{A}$ breaks the indistinguishability under Chosen Ciphertext Attacks, and denoted by $Adv_{sk}(\mathcal{A})$. The symmetric-key encryption scheme SE is Chosen Ciphertext Secure if $Adv_{sk}(\mathcal{A})$ is negligible [30].

Decisional Diffie-Hellman (DDH) assumption

Let $G = \langle g \rangle$ be a cyclic group of prime order q and x, y, z are randomly chosen in Z_q . A DDH attacker $\mathcal{A}$, a probabilistic Turing Machine, is defined as follows: Using the value of a random

bit c decides the value of Z , which is $g^{xy} \bmod p$ if $c = 1$ and $g^z \bmod p$ if $c = 0$ where, p is a large prime. Given (X, Y, Z) , $\mathcal{A}$ can correctly guess the bit c with probability $\text{Adv}_G^{\text{ddh}}(\mathcal{A})$ within polynomial time t . The Decisional Diffie-Hellman Assumption states that for every probabilistic polynomial time Turing Machine $\mathcal{A}$, for large enough k , $\text{Adv}_G^{\text{ddh}}(\mathcal{A}) \leq \varepsilon(k)$, where $\varepsilon(k)$ is a negligible function.

Lower bounds on number of messages and rounds for three-party AKA-MA protocols

This section first introduces the rules according to the behavior patterns of AKA-MA protocols, and then derives the lower bounds on the number of messages and rounds for three-party AKA-MA protocols based on these rules. The rules for AKA-MA protocols are as follows.

The rules for AKA-MA protocols

Rule 1. In the AKA-MA protocol, the originator is the only one who initiates a message. The others can issue messages only at the moment they receive one. The protocol will proceed in sequential order.

Rule 2. In the AKA-MA protocol, each participant has to send out a message.

Rule 3. In the AKA-MA protocol, to derive a session key a client has to receive messages from all other clients.

Rule 4. In the AKA-MA protocol, a client cannot issue an authenticator before it derives the session key.

Rule 5. In the AKA-MA protocol, each client must send out an authenticator, and receive and verify authenticators from all other clients.

In the 3AKA-MA protocol, the clients do not share any secret key and thus authenticate each other via the help of the trusted server.

Rule 6. In the 3AKA-MA protocol, client A can only authenticate B via S, and vice versa.

Rule 7. In the 3AKA-MA protocol, if a sub-key cannot be revealed in the channel, then client A can only obtain the sub-key from B via S, and vice versa.

Lower bounds on number of messages and rounds for three-party AKA-MA protocols

This subsection provides the lower bounds on the number of messages and rounds for the 3AKA-MA protocols, based on the rules described in Section 3.1.

Theorem 1. *Every 3AKA-MA protocol is implemented in at least five messages if the sub-keys cannot be revealed in the channel.*

Proof: In a 3AKA-MA protocol, by Rule 1, the protocol originator A initiates a message, the other participants B and S issue messages at the moment of receiving one, and the protocol proceeds in sequential order. If the sub-keys cannot be revealed in the channel, then we have many variable protocols, as shown in Fig 1.

For protocol (a), after the third message, A receives a message sent from B and transmitted by S. A can authenticate B via S by Rule 6, receive a sub-key K_2 from B, and derive the session key SK by Rules 3 and 7. Then, by Rule 4, A can construct and issue an authenticator auth_A in the fourth message. On the other hand, after the fourth message, B receives a

Fig 1. The branches for 3AKA-MA protocols if the sub-keys cannot be revealed in the channel by Rules 1 and 2.

<https://doi.org/10.1371/journal.pone.0174473.g001>

message sent from B and transmitted by S. Similarly, B can authenticate A via S, receive a sub-key K_1 from A, and derive the session key SK. Then, by Rule 4, B can verify $auth_A$ from A, and construct and issue an authenticator $auth_B$ in the fifth message. Finally, A can verify $auth_B$ from B. Hence, for A and B, five messages are required. In addition, for S, three messages are required by Rules 1 and 2. Therefore, the 3AKA-MA protocol can be implemented in five messages in (a).

For protocol (b), after the fourth message, B can receive a sub-key K_1 sent from A and transmitted by S. Then B can derive the session key SK and issue an authenticator $auth_B$ in the fifth message by Rules 3, 4, and 7. However, by Rule 5, A must receive and verify $auth_B$ from B. For A, at least one extra message is required. Thus protocol (b) cannot be implemented in five messages.

For protocol (c), by Rules 3 and 7, B cannot derive the session key SK until it receives a sub-key K_1 sent from A and transmitted by S. Thus, for B, an extra message is required to receive the sub-key K_1 from A. In addition, another extra one is required to issue an authenticator $auth_B$ by Rules 4 and 5. Hence, at least two extra messages are required. Thus protocol (c) cannot be implemented in five messages.

For protocol (d), after the third message, B can authenticate A via S by Rule 6, receive a sub-key K_1 sent from A and transmitted by S, and derive the session key SK by Rules 3 and 7. Then, by Rule 4, B can compute and issue an authenticator $auth_B$ in the fourth message. Similarly, after the fourth message, A can authenticate B via S by Rule 6, receive a sub-key K_2 sent from B and transmitted by S, and derive the session key SK by Rules 3 and 7. Then, by Rule 4, A can verify $auth_B$ from B, and compute and issue an authenticator $auth_A$ in the fifth message. Finally, B can verify $auth_A$ from A. Hence, for A and B, five messages are required. In addition,

for S, three messages are required by Rules 1 and 2. Therefore, the 3AKA-MA protocol can be implemented in five messages in (d).

For protocol (e), after the fourth message, A receives a message sent from B and transmitted by S. A can receive a sub-key K_2 from B and derive the session key SK by Rules 3 and 7. By Rule 4, A can issue an authenticator $auth_A$ in the fifth message. However, B must receive and verify $auth_A$ from A by Rule 5. For B, at least one extra message is required. Thus, protocol (e) cannot be implemented in five messages.

For protocol (f), by Rules 3 and 7, A must receive a message that includes a sub-key K_2 sent from B and transmitted by S. Then A can derive the session key SK. Thus, for A, an extra message is required to receive the sub-key K_2 from B and another extra message is required to issue an authenticator $auth_A$ by Rules 4 and 5. Hence, at least two extra messages are required. Thus the protocol (f) cannot be implemented in five messages.

For protocol (g), from arguments similar to protocol (b), at least two extra messages are required. Thus, protocol (g) cannot be implemented in five messages.

For protocol (h), from arguments similar to protocol (f), at least two extra messages are required. Thus, protocol (h) cannot be implemented in five messages.

For protocol (i), by Rules 3 and 7, A cannot derive the session key SK until it receives a message that includes a sub-key K_2 sent from B and transmitted by S. Thus, for A, two extra messages are required to receive sub-key K_2 from B and another extra one is required to issue an authenticator $auth_A$ by Rules 4 and 5. Hence, at least three extra messages are required. Thus, protocol (i) cannot be implemented in five messages.

For protocol (j), after the second message, B can authenticate A via S by Rule 6, receive a sub-key K_1 sent from A and transmitted by S, and derive the session key SK by Rules 3 and 7. Then, from arguments similar to protocol (d), five messages are required for A and B and four messages are required for S. Therefore, the 3AKA-MA protocol can be implemented in five messages in (j).

For protocol (k), from arguments similar to protocol (e), at least one extra message is required. Thus, protocol (k) cannot be implemented in five messages.

For protocol (l), from arguments similar to protocol (f), at least two extra messages are required. Thus, protocol (l) cannot be implemented in five messages.

For protocols (m) and (n), from arguments similar to protocol (i), at least three extra messages are required. Thus, protocols (m) and (n) cannot be implemented in five messages.

For protocol (o), by Rules 1 and 2, B can send out a message that includes sub-key K_2 only after receiving one, and must send out a message. Therefore, at least two extra messages are required. In addition, A cannot derive the session key SK until it receives sub-key K_2 sent from B and transmitted by S. Thus, for A, an extra message is required to receive the sub-key K_2 transmitted by S. In addition, another extra one is required to issue an authenticator $auth_A$ by Rules 4 and 5. Hence, at least three extra messages are required. Thus, protocol (o) cannot be implemented in five messages.

Table 1 summarizes the analyses of protocols (a), (b), ..., (n) and (o). From these analyses, we can conclude that, with the exceptions of protocols (a), (d), and (j), these 3AKA-MA

Table 1. The messages of 3AKA-MA protocols are required in communications if the sub-keys cannot be revealed in the channel.

Protocols	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
Required messages	5	≥ 6	≥ 6	5	≥ 6	≥ 6	≥ 6	≥ 6
Protocols	(i)	(j)	(k)	(l)	(m)	(n)	(o)	
Required messages	≥ 6	5	≥ 6	≥ 6	≥ 6	≥ 6	≥ 6	

<https://doi.org/10.1371/journal.pone.0174473.t001>

Proof: In a 3AKA-MA protocol, by Rule 1, the protocol originator A initiates messages for authentication and including a sub-key K_1 in the first round. After receiving the message

7 / 25

Table 2. The number of messages for 3AKA-MA protocols are required in communications.

Protocol	(a)	(b)	(c)	(d)	(e)
Required messages at least	4	≥ 5	≥ 5	≥ 5	≥ 5
Protocol	(f)	(g)	(h)	(i)	(j)
Required messages at least	≥ 5	≥ 5	≥ 6	≥ 5	≥ 5

<https://doi.org/10.1371/journal.pone.0174473.t002>

including K_1 , B can randomly select a sub-key K_2 and compute the session key SK by Rule 3 and an authenticator $auth_B$ by Rules 3 and 4. In the second round, S sends the message to B for authenticating A, so that B can authenticate A via S by Rule 6. In addition, B also sends out messages for authentication and including K_2 and $auth_B$. After receiving K_2 and $auth_B$ from B, A can derive a session key SK by Rules 3. Then A can verify $auth_B$, compute an authenticator $auth_A$ by Rule 4, and send a message including $auth_A$ to B in the third round. Simultaneously, S sends out messages so that A can authenticate B via S by Rule 6 and B can verify $auth_A$. Hence, by Rule 5, every 3AKA-MA protocol is implemented in at least four rounds.

This section has provided the lower bounds on the number of messages and rounds for the 3AKA and 3AKA-MA protocols based on the rules described in Section 3.1. In the next section, we will present 3AKA-MA protocols that realize the lower bounds on the number of messages and rounds for 3AKA-MA protocols based on the results of above theorems.

The communication-efficient 3AKA-MA protocols

This section will use the derived communication results of Section III to develop secure and communication-efficient 3AKA-MA protocols. First, we present 3AKA-MA protocols whose sub-keys cannot be revealed in the channel, and then propose 3AKA-MA protocols in which revealing the sub-keys cannot compromise the session key.

Assume that A and B are two communicating parties and that S is a trusted server. Clients A and B share long-lived keys K_{AS} and K_{BS} respectively, with server S. The notation used throughout this section is as follows:

Notation

p, g	A large prime p and a generator g in group Z_p^* , a group in which the Diffie-Hellman problem is considered hard.;
x, y	Random exponents chosen by A and B.;
$\{M\}_K$	Encryption of M using a symmetric encryption scheme with a cryptographically strong shared key K .;
$H(M)$	A one-way hash function H applied to M [32].;
M_1, M_2	M_1 is concatenated with M_2 .;
$A \rightarrow B: M$	A sends message M to B.

The communication-efficient nonce-based 3AKA-MA protocols for cases where the sub-keys cannot be revealed in the channel

In the proposed 3AKA-MA protocols, A and B randomly select sub-keys K_1 and K_2 , respectively. Since the sub-keys cannot be revealed in the channel, A obtains the sub-key K_2 from B via S, and vice versa. Then, they can derive a common session key $SK \equiv f(K_1, K_2)$. Finally, they compute and send out their authenticators μ_A and μ_B . All of the proposed 3AKA-MA protocols

Fig 3. The message-efficient 3AKA1-MA protocol.

<https://doi.org/10.1371/journal.pone.0174473.g003>

were developed based on Theorem 1 and Theorem 2 in Section 3 and executed using five messages and four rounds.

Proposed message-efficient nonce-based 3AKA1-MA protocol. The proposed message-efficient nonce-based 3AKA1-MA protocol was developed based on protocol (a) in Theorem 1 for a case where the sub-keys cannot be revealed in the channel. Fig 3 depicts the proposed 3AKA1-MA protocol, which will now be described in detail.

1. $A \rightarrow B$: $A, B, \{A, S, A, K_1\}_{K_{AS}}$
A selects a random number K_1 as the sub-key, encrypts (A, S, A, K_1) using A's secret key K_{AS} , and sends $A, B, \{A, S, A, g^{N_A}\}_{K_{AS}}$ to B.
2. $B \rightarrow S$: $A, B, \{A, S, A, K_1\}_{K_{AS}}, \{B, S, B, K_2\}_{K_{BS}}$
Similarly, B selects a random number K_2 as the sub-key, encrypts (B, S, B, K_2) using B's secret key K_{BS} . Then it sends $\{B, S, B, K_2\}_{K_{BS}}$ and forwards $A, B, \{A, S, A, g^{N_A}\}_{K_{AS}}$ to S.
3. $S \rightarrow A$: $\{S, A, B, K_2\}_{K_{AS}}, \{S, B, A, K_1\}_{K_{BS}}$
S decrypts $\{A, S, A, K_1\}_{K_{AS}}$ and $\{B, S, B, K_2\}_{K_{BS}}$ with secret keys K_{AS} and K_{BS} , and authenticates A and B by checking A's ID and B's ID, respectively. If it is successful, S then sends $\{S, A, B, K_2\}_{K_{AS}}$ and $\{S, B, A, K_1\}_{K_{BS}}$ to A.
4. $A \rightarrow B$: $\{S, B, A, K_1\}_{K_{BS}}, \mu_A$
A obtains K_2 by decrypting $\{S, A, B, K_2\}_{K_{AS}}$ with K_{AS} and derives the session key $SK \equiv f(K_1, K_2)$. Then it computes an authenticator $\mu_A = H(A, B, SK)$, and sends $\{S, B, A, K_1\}_{K_{BS}}$ and μ_A to B.
5. $B \rightarrow A$: μ_B
Similarly, B obtains K_1 by decrypting $\{S, B, A, K_1\}_{K_{BS}}$ with K_{BS} and derives the session key $SK \equiv f(K_1, K_2)$. If B successfully verifies μ_A from A, then it computes an authenticator $\mu_B = H(B, A, SK)$ and sends μ_B to A. Finally, A can verify μ_B from B. Accordingly, A and B have the common session key $SK \equiv f(K_1, K_2)$.

Fig 4. The message-efficient 3AKA2-MA protocol.

<https://doi.org/10.1371/journal.pone.0174473.g004>

Proposed message-efficient nonce-based 3AKA2-MA protocol. The proposed message-efficient nonce-based 3AKA2-MA protocol was developed based on protocol (d) in Theorem 1 for a case where the sub-keys cannot be revealed in the channel. Fig 4 depicts the proposed 3AKA2-MA protocol, which is described as follows.

1. $A \rightarrow B: A, B, \{A, S, A, K_1\}_{K_{AS}}$
2. $B \rightarrow S: A, B, \{A, S, A, K_1\}_{K_{AS}}, \{B, S, B, K_2\}_{K_{BS}}$
3. $S \rightarrow B: \{S, A, B, K_2\}_{K_{AS}}, \{S, B, A, K_1\}_{K_{BS}}$
4. $B \rightarrow A: \{S, A, B, K_2\}_{K_{AS}}, \mu_B$
5. $A \rightarrow B: \mu_A$

Proposed message-efficient nonce-based 3AKA3-MA protocol. The message-efficient nonce-based 3AKA3-MA protocol was developed based on protocol (j) in Theorem 1 for a case where the sub-keys cannot be revealed in the channel. In fact, this 3AKA3-MA-3 protocol is the same as Gong's nonce-based 3AKA-MA protocol in [2,3]. Fig 5 depicts the proposed 3AKA3-MA protocol, which is described as follows.

1. $A \rightarrow S: A, B, \{A, S, A, K_1\}_{K_{AS}}$
2. $S \rightarrow B: \{S, A, B, K_1\}_{K_{AS}}$
3. $B \rightarrow S: \{B, S, B, K_2\}_{K_{BS}}, \mu_B$
4. $S \rightarrow A: \{S, A, B, K_2\}_{K_{AS}}, \mu_B$
5. $A \rightarrow B: \mu_A$

Fig 5. The message-efficient 3AKA3-MA protocol.

<https://doi.org/10.1371/journal.pone.0174473.g005>

Proposed round-efficient nonce-based 3AKA-MA protocol. The proposed round-efficient nonce-based R3AKA-MA protocol was developed based on Theorem 2 and can be executed in four rounds. The proposed protocol is described as follows.

1. $A \rightarrow S: A, B, \{A, S, A, K_1\}_{K_{AS}}$
 $A \rightarrow B: A, B$
2. $B \rightarrow S: A, B, \{B, S, B, K_2\}_{K_{BS}}$
3. $S \rightarrow A: \{S, A, B, K_2\}_{K_{AS}}$
 $S \rightarrow B: \{S, B, A, K_1\}_{K_{BS}}$
4. $A \rightarrow B: \mu_A$
 $B \rightarrow A: \mu_B$

Proposed communication-efficient 3AKA-MA protocols for a case where revealing the sub-keys cannot compromise the session key

In an authenticated key agreement protocol, if the session key is based on asymmetric cryptosystems, such as the Diffie-Hellman key exchange or the Elliptic Curve (ECC) Diffie-Hellman key exchange [1, 33, 34], Chebyshev chaotic map-based Diffie-Hellman key exchange [15, 35, 36, 37] then revealing the sub-keys used to generate the session key cannot compromise the session key. This subsection will use the Diffie-Hellman key exchange to propose communication-efficient 3AKA-MA protocols in which revealing the sub-keys cannot compromise the session key.

In a Diffie-Hellman-based authentication protocol, revealing the sub-keys, $K_1 = g^x \pmod{p}$ and $K_2 = g^y \pmod{p}$, used to generate the session key ($SK = H(A, B, K_1, K_2, K)$) cannot compromise the session key itself because the session key cannot be determined without a knowledge of x or y , where $K \equiv g^{xy} \pmod{p}$. Therefore, clients A and B can publicly exchange the sub-keys K_1 and K_2 for generating the session key without the help of the server. Upon receiving

Fig 6. The message-efficient DH-3AKA-MA protocol.

<https://doi.org/10.1371/journal.pone.0174473.g006>

K_1 (or K_2), the client can compute the session key and send out its authenticator $\mu_B = H(B, A, K_1, SK)$ (or $\mu_A = H(A, B, K_2, SK)$). Finally, A and B can authenticate each other via S, have a common session key SK, and verify the authenticators μ_B and μ_A .

Proposed message-efficient nonce-based DH-3AKA-MA protocol. The proposed message-efficient nonce-based DH-3AKA-MA protocol is developed according to the protocol (a) in Theorem 3. Fig 6 depicts the proposed DH-3AKA-MA protocol which will now be described in detail.

1. $A \rightarrow B: A, B, \{A, S, A, K_1\}_{K_{AS}}, K_1$
A selects a random number x ; computes the sub-key $K_1 = g^x \pmod{p}$ and encrypts (A, S, A, K_1) using A's secret key K_{AS} . Then A sends $A, B, \{A, S, A, K_1\}_{K_{AS}}, K_1$ to B.
2. $B \rightarrow S: A, B, \{A, S, A, K_1\}_{K_{AS}}, \{B, S, B, K_2\}_{K_{BS}}, K_2, \mu_B$
Similarly, B selects random numbers y ; computes the sub-key $K_2 = g^y \pmod{p}$ and encrypts (B, S, B, K_2) using B's secret key K_{BS} . Simultaneously, B computes $K \equiv (K_1)^y \pmod{p}$, $SK = H(A, B, K_1, K_2, K)$ as the session key shared with A, and an authenticator $\mu_B = H(B, A, K_1, SK)$. Then B sends $\{B, S, B, K_2\}_{K_{BS}}, K_2, \mu_B$ and forwards $A, B, \{A, S, A, K_1\}_{K_{AS}}$ to S.
3. $S \rightarrow A: \{S, A, B, K_2\}_{K_{AS}}, \{S, B, A, K_1\}_{K_{BS}}, K_2, \mu_B$
S decrypts $\{A, S, A, K_1\}_{K_{AS}}$ and $\{B, S, B, K_2\}_{K_{BS}}$ with secret keys K_{AS} and K_{BS} , and authenticates A and B by checking A's ID and B's ID, respectively. If it is successful, S then sends $\{S, A, B, K_2\}_{K_{AS}}$ and $\{S, B, A, K_1\}_{K_{BS}}$ as one-time certificates for A and B, respectively, and forwards K_2, μ_B to A.
4. $A \rightarrow B: \{S, B, A, K_1\}_{K_{BS}}, \mu_A$
If A successfully validates K_2 by decrypting $\{S, A, B, K_2\}_{K_{AS}}$ with K_{AS} , then computes $K \equiv (K_2)^x \pmod{p}$, the session key $SK = H(A, B, K_1, K_2, K)$, and an authenticator $\mu_A = H(A, B, K_2, SK)$, and sends $\{S, B, A, K_1\}_{K_{BS}}, \mu_A$ to B. Finally, B authenticates A and S by decrypting

$\{S, B, A, K_1\}_{K_{BS}}$ with K_{BS} and checking K_1 and verifies the authenticator μ_A from A . Accordingly, A and B have a common session key $SK = H(A, B, K_1, K_2, K)$.

Proposed round-efficient DH-R3AKA-MA protocol. The proposed round-efficient nonce-based DH-3AKA-MA (DH-R3AKA-MA) protocol is developed according to Theorem 4 and can be executed in three rounds. The proposed DH-3AKA-MA protocol is described as follows.

1. $A \rightarrow S: A, B, \{A, S, A, K_1\}_{K_{AS}}$
 $A \rightarrow B: A, B, K_1$
2. $B \rightarrow S: A, B, \{B, S, B, K_2\}_{K_{BS}}$
 $B \rightarrow A: K_2, \mu_B$
3. $S \rightarrow A: \{S, A, B, K_2\}_{K_{AS}}$
 $S \rightarrow B: \{S, B, A, K_1\}_{K_{BS}}$
 $A \rightarrow B: \mu_A$

This section has presented four 3AKA-MA protocols whose sub-keys cannot be revealed in the channel, and two DH-3AKA-MA protocols in which revealing the sub-keys cannot compromise the session key. The proposed 3AKA1-MA, 3AKA2-MA, and 3AKA3-MA protocols require only five messages; the proposed R3AKA-MA protocol requires only four rounds; the proposed DH-3AKA-MA protocol requires only four messages; and the proposed DH-R3AKA-MA protocol requires only three rounds of communication. Thus all of the proposed 3AKA-MA protocols realize the lower bounds on the number of messages and rounds for 3AKA-MA protocols. In addition, we can construct synchronized clocks in a network environment and transform each of the proposed nonce-based 3AKA-MA protocols into a clock-based 3AKA-MA protocol by adding timestamps in ciphertexts. These obtained clock-based 3AKA-MA protocols do not require any extra message or round and thus also realize the lower bounds on the number of messages and rounds for 3AKA-MA protocols.

Security and performance analyses

The proposed 3AKA-MA protocols and compares their performance with that of other related authentication protocols.

Security proofs of proposed 3AKA-MA protocols

Communication model. Protocol participants: Two protocol participants A and B try to authenticate each other and establish an authentication key SK via the help of a trusted third party S in protocol P . A participant may be involved in numerous instances, called oracles, of distinct concurrent executions of P . The instance i of participant U is expressed as Π_U^i . [35]

Long-lived keys: The long-term secret key K_{AS} is shared between A and S , and the long-term secret key K_{BS} is shared between B and S . The long-lived keys K_{AS} and K_{BS} are defined as the symmetric keys of A and B , respectively.

Oracle queries: The following descriptions define oracle queries which model the capabilities of the adversary $\mathcal{A}$.

- **Send**(Π_U^i, M): In this query, the adversary $\mathcal{A}$ can control all communications in protocol P . When $\mathcal{A}$ sends oracle Π_U^i a message M , Π_U^i sends back the response message that is computed by executing P . $\mathcal{A}$ can send a user oracle Π_U^i a query ($\Pi_U^i, \text{"start"}$) as initialization of executing P [35].

- **Corrupt**(U): In this query, the adversary $\mathcal{A}$ who has compromised long-lived keys cannot compromise previous session keys. $\mathcal{A}$ sends a participant U this query, and returns U 's long-lived key.
- **SymEnc** ($\{\mathcal{E}, D\}, k, \{M, C\}$): This query allows that adversary $\mathcal{A}$ accesses to encryption oracle SymEnc defined in previous section. When $\mathcal{A}$ sends SymEnc an encryption query **SymEnc** ($\mathcal{E}, k, M$), SymEnc searches the Γ -table. If a record (k, M, C) has been queried and recorded in the Γ -table, SymEnc sends back the correspondent ciphertext C ; otherwise SymEnc returns a random C , and appends (k, M, C) to the Γ -table. Similarly, on receiving a decryption query **SymEnc** ($\mathcal{D}, k, C$), if a record (k, M, C) has been queried and recorded in the Γ -table, SymEnc sends backs the correspondent plaintext M ; otherwise SymEnc returns a random M , and appends (k, M, C) to the Γ -table.
- **Hash**(M): In this query, the adversary $\mathcal{A}$ receives hash results by sending a random oracle Ω queries. On receiving this query, if a record (M, r) has been queried and recorded in the H -table, Ω send back r ; otherwise Ω send back a random r' , and appends (M, r') to the H -table [35].
- **Reveal** (Π_U^i): This query models known key attacks. The adversary $\mathcal{A}$ who has compromised one authentication key cannot reveal other authentication keys. The **Reveal** query is only available to adversary $\mathcal{A}$ when oracle Π_U^i has accepted [35].
- **Test**(Π_U^i): This query measures the semantic security of the session key SK , which specifies the indistinguishability of the real session key from a random string. During the executing protocol P , adversary $\mathcal{A}$ can ask a single **Test** query at sometime. Upon receiving this query, Π_U^i returns $\mathcal{A}$ the real session key SK or a random string by flipping an unbiased coin c . This query is available only when Π_U^i is Fresh [35].

Security definitions. Partnering: Two user oracles Π_A^i and Π_B^j are partnered if

- oracles Π_A^i and Π_B^j directly exchange messages and
- only oracles Π_A^i and Π_B^j obtain the common session key SK .

Freshness: An oracle Π_U^i is **Fresh** in P if.

- Π_U^i has accepted SK , and
- Π_U^i and its partner have not been sent a **Reveal** query.

Security proofs. The Difference Lemma [30] is used for our sequence of games and is described as follows:

Lemma 5 (Difference Lemma). Let A, B and F be events defined in some probability distribution, and suppose that $A \wedge \neg F \Leftrightarrow B \wedge \neg F$. Then

$$|\Pr[A] - \Pr[B]| \leq \Pr[F].$$

The following theorem shows that the proposed 3AKA1-MA protocol has AKE security and provides mutual authentication by using the logical tool which was defined and presented by Burrows et al. [38] in 1990 and Buttyan et al. [39] in 1998.

Theorem 6. The proposed 3AKA1-MA protocol has AKE security and provides mutual authentication.

The proposed 3AKA1-MA, 3AKA2-MA, 3AKA3-MA, 3AKA4-MA protocols are similar with the proposed 3AKA1-MA protocol. These protocols reveal the same information in the channel. Their security proofs are almost the same, can be obtained by using similar arguments, and thus are not presented here.

In the following, we first prove AKE security of the DH-3PAKA protocol, which is transformed from the DH-3PAKA-MA protocol by removing the authenticators μ_A and μ_B . Then, we use AKE security of the DH-3AKA protocol to prove AKE and MA securities of the DH-3PAKA-MA protocol. The DH-3AKA protocol is described as follow.

1. $A \rightarrow B: A, B, \{A, S, A, K_1\}_{K_{AS}}, K_1$
2. $B \rightarrow S: A, B, \{A, S, A, K_1\}_{K_{AS}}, \{B, S, B, K_2\}_{K_{BS}}, K_2$
3. $S \rightarrow A: \{S, A, B, K_2\}_{K_{AS}}, \{S, B, A, K_1\}_{K_{BS}}, K_2$
4. $A \rightarrow B: \{S, B, A, K_1\}_{K_{BS}}$

The following theorem shows that the proposed DH-3AKA protocol has AKE security if the used long-term secret keys are secure and the Decisional Diffie-Hellman assumptions holds in G .

Theorem 7. Let Adv_{sk} denote the advantage that an adversary breaks the long-term secret key within time t_1 . Let Adv_G^{ddh} be the advantage that a DDH attacker solves the DDH problem within time t_3 . Then, the probability that an adversary breaks the AKE security of the DH-3AKA protocol:

$$Adv_{dh-3aka}^{ake}(t', q_0, q_1, q_2) \leq \frac{q_1^2 + q_2^2}{2^{l-1}} + 4 \cdot Adv_{sk}(t_1, q_0, q_1, q_2) + 2 \cdot Adv_G^{ddh}(t_3, q_0, q_1, q_2),$$

where $t' \leq t_1 + (q_1 + q_2) \cdot \tau_1 + 4 \cdot \tau_3$; q_0 denotes the numbers of the Send queries; q_1 and q_2 denote the numbers of the SymEnc queries involving A and S, and involving B and S, respectively; l is a security parameter [40]; $t' = t_1 + (q_1 + q_2) \tau_1$; and τ_1 is the time to compute a symmetric en/decryption; and τ_3 is the time to perform an exponential computation.

The following theorem shows that the proposed DH-3AKA-MA protocol has AKE security if the used hash function is secure and the DH-3AKA protocol has AKE security.

Theorem 8. Let $Adv_{dh-3aka}^{ake}$ denote the advantage that an adversary breaks the long-term secret key within time t_4 . Then, the probability that an adversary breaks the AKE security of the DH-3AKA-MA protocol:

$$Adv_{dh-3aka-ma}^{ake}(t', q_0, q_1, q_2, q_3) \leq 2Adv_{dh-3aka}^{ake}(t_3, q_0, q_1, q_2, q_3) + \frac{q_3^2}{2^{l-1}},$$

where $t' \leq t_3 + (q_0 + q_1 + q_2) \cdot t_{relay} + 2 \cdot \tau_2 + 4 \cdot \tau_3$; the used parameters are defined as in Theorems 7; q_3 denotes the numbers of the Hash queries involving A and B; t_{relay} is the time of relay a query; τ_2 is the time of generating a random number; and τ_3 is the time to perform an exponential computation.

The following theorem shows that the proposed DH-3AKA-MA protocol has MA security if the used hash function is secure and the DH-3AKA protocol has AKE security.

Theorem 9. Let $Adv_{dh-3aka}^{ake}$ denote the advantage an adversary breaks the AKE security of the DH-3AKA protocol within time t_4 . Let $Adv_{dh-3aka-ma}^{ma}$ denote the advantage in violating the explicit

Table 3. Comparison of the related 3PAKA protocols and the proposed protocols.

Protocols	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆
Gong's 3PAKA-MA [2,3]	N	$8T_S+2T_H$	N	5m/4r	Y	Y
Amin et al.'s 3AKA-MA [16]	N	$32T_H$	N	6m/-	Y	N
Amin-Biswas's 3AKA [40]	N	$20T_H$	N	4m/-	N	Y
Proposed 3PAKA-MA	N	$8T_S+2T_H$	N	5m/4r	Y	Y
Lin et al.'s DH-3AKA-MA [9]	Y	$6T_E+6T_S+4T_H$	Y	5m/-	Y	N
Lu and Cao's DH-3AKA [12]	Y	$10T_E+8T_{MD}+10T_H$	Y	5m/4r	N	N
Lee et al.'s DH-3AKA-MA [15]	Y	$11T_C+8T_S+12T_H$	Y	5m/-	Y	N
Lee et al.'s DH-3AKA-MA [25]	Y	$10T_E+16T_H$	Y	5m/-	Y	N
Lee et al.'s DH-3AKA-MA [35]	Y	$4T_{AS}+4T_C+4T_H$	Y	4m/3r	Y	Y
Lee et al.'s DH-3AKA [36]	Y	$8T_C+11T_H$	Y	4m/-	N	Y
Li et al.'s DH-3AKA-MA [37]	Y	$4T_C+8T_S+11T_H$	Y	5m/-	Y	N
Proposed DH-3AKA-MA	Y	$4T_E+8T_S+4T_H$	Y	4m/3r	Y	Y

P₁: Revealing sub-keys; P₂: Computational cost; P₃: Providing perfect forward secrecy;
P₄: Transmissions; P₅: Providing explicit mutual authentication; P₆: Realizing lower bounds.

<https://doi.org/10.1371/journal.pone.0174473.t003>

mutual authentication of the DH-3AKA-MA protocol. Then, we have

$$Adv_{dh-3aka-ma}^{ma}(t', q_0, q_1, q_2, q_3) \leq 2Adv_{dh-3aka}^{ake}(t_4, q_0, q_1, q_2) + \frac{q_3^2 + 1}{2^{l-1}}$$

where $t' \leq t_4 + (q_0 + q_1 + q_2) \cdot t_{relay} + 2 \cdot \tau_2$, the used parameters are defined as in Theorems 9 and 10.

Performance analyses and comparisons

Table 3 shows a performance comparison of the related 3AKA-MA protocol and the 3AKA-MA protocols proposed here, where T_E denotes the time to execute an exponential operation; T_{AS} denotes the time to execute an asymmetric en/decryption operation; T_C denotes the time required to execute a Chebyshev chaotic map operation; T_S denotes the time to execute a symmetric en/decryption operation, and T_H denotes the time to execute a hash operation.

The first comparison item lists whether or not the sub-keys for generating a session key can be revealed in the channel. In Gong's protocol [2, 3], Amin et al.'s protocol [16], Amin and Biswas's protocol [41] and the proposed protocols, the sub-keys cannot be revealed in the channel and clients must exchange their sub-key via the help of the server. In related DH-3AKA-MA protocols and the proposed DH-3AKA-MA protocols, session key security is based on the Diffie-Hellman problem, and thus clients can directly exchange their sub-keys.

The second comparison item is computational cost. Gong's 3AKA-MA protocol, Amin et al.'s 3AKA-MA [16], Amin and Biswas's protocol [41] and the proposed 3AKA-MA protocols only require symmetric en/decryption and hash processes, but do not provide perfect forward secrecy. The related DH-3AKA-MA protocols [9,12,15,25,35–37] require more exponential computations or Chebyshev chaotic map operations. The proposed DH-3AKA-MA protocols also require eight en/decryption processes and four exponential computations. Although extra modular exponential costs or Chebyshev chaotic map operations are required, the related DH-3AKA-MA protocols and the proposed DH-3AKA-MA protocols provide perfect forward secrecy.

The number of transmissions was also compared. Gong's 3AKA-MA protocol and the proposed 3AKA-MA protocols require five messages and four rounds. These protocols thus realize the lower bounds on communications for 3AKA-MA protocols without revealing sub-keys in the channel. Although Amin and Biswas 3AKA [40] protocol requires fewer messages than other protocols, but does not provide explicit mutual authentication. In addition, the related DH-3AKA-MA protocols in [35] and [36] and the proposed DH-3AKA-MA protocol requires four messages and three rounds. However, Lee et al.'s DH-3AKA-MA protocol [35] require server public keys, thus is inefficient in computations. Lee et al.'s DH-3AKA-MA protocol [36] only requires four messages, but does not provide explicit mutual authentication. Altogether, the proposed protocols involve fewer transmissions than other 3AKA-MA protocols, and realize the lower bounds on communications for 3AKA-MA protocols.

Conclusions

This investigation has provided the lower bounds on communications for 3AKA-MA protocols. In addition, it also considered the lower bounds on communications for the 3AKA-MA protocols whose sub-keys cannot be revealed in the channel and for the 3AKA-MA protocols in which revealing the sub-keys cannot compromise the session key. By using the derived results for the lower bounds on communications, communication-efficient and provably secure 3AKA-MA protocols were developed. As seen in Table 3, the proposed 3AKA-MA protocols involve fewer transmissions than other related 3AKA-MA protocols, but also realize the newly defined lower bounds on communications for 3AKA-MA protocols and are suitable for practical environments. Therefore, a 3AKA-MA protocol, which is developed by using the derived results in this paper, involves fewer transmissions and is efficient in communication.

Appendix A

Proof of Theorem 3: In a 3AKA-MA protocol, by Rule 1, the protocol originator A initiates a message, the other participants B and S issue messages at the moment of receiving one, and the protocol proceeds in sequential order. Then, by using similar arguments in Theorem 2, we have many variable protocols, as shown in Fig 2.

For protocol (a), after the first message, B can receive a sub-key K_1 from A and derive the session key SK by Rule 3. Then, by Rule 4, B can compute and issue an authenticator $auth_B$ in the second message. On the other hand, after the third message, A can authenticate B via S by Rule 6, receive a sub-key K_2 from B, derive the session key SK by Rule 3, and verify $auth_B$ from B. Then, by Rule 4, A can compute and issue an authenticator $auth_A$ in the fourth message. Finally, B can authenticate A via S by Rule 6 and verify $auth_A$ from A. Hence, for A and B, four messages are required. In addition, for S, three messages are required by Rules 1 and 2. Therefore, the 3AKA-MA protocol can be implemented in four messages in (a).

For protocol (b), after the third message, A can receive a sub-key K_2 from B and derive the session key SK by Rule 3. By Rule 4, A can compute and issue an authenticator $auth_A$ in the fourth message. However, by Rule 5, B must receive and verify $auth_A$ from A, and thus requires an extra message at least. Therefore, protocol (b) cannot be implemented in four messages.

For protocol (c), by Rules 3, A cannot derive the session key SK until it receives a sub-key K_2 from B. Thus, for A, an extra message is required to receive the K_2 . In addition, another extra message is required to issue an authenticator $auth_A$. Hence, at least two extra messages are required by Rules 4 and 5. Thus protocol (c) cannot be implemented in four messages.

For protocol (d), by Rule 3, A can obtain a sub-key K_2 from B and derive the session key SK after it receives a message from B in the second message. Then, by Rule 4, A can compute and issue an authenticator $auth_A$ in the third message. However, by Rule 5, B must receive and

verify $auth_A$ from A, and thus requires an extra message at least. Therefore, protocol (d) cannot be implemented in five messages.

For protocol (e), by Rule 6, A cannot authenticate B via S until it receives a message sent from B and transmitted by S. Then, A requires an extra message at least. Therefore, protocol (e) cannot be implemented in four messages.

For protocol (f), by Rules 1 and 2, S can issue messages only while receiving one and must send out a message. Therefore, two extra messages are required at least. Thus protocol (f) cannot be implemented in four messages.

For protocol (g), by Rule 3, A cannot derive the session key SK until it receives a sub-key K_2 from B. Then, A requires an extra message to receive the sub-key K_2 from B. In addition, by Rules 4 and 5, another extra one is required to issue an authenticator $auth_A$. Therefore, protocol (g) requires two extra messages at least, and thus cannot be implemented in four messages.

For protocol (h), by Rule 6, A cannot authenticate B via S since it does not receive a message sent from B and transmitted by S. Therefore, protocol (h) requires two extra messages at least, and thus is implemented in at least six messages.

For protocol (i), by Rule 3, A can receive a sub-key K_2 from B and derive the session key SK after receiving a message from B in the third message. Then, by Rule 4, A can issue an authenticator $auth_A$ in the fourth message. However, by Rule 5, B must receive and verify $auth_A$, and thus requires an extra message at least. Therefore, protocol (i) cannot be implemented in four messages.

For protocol (j), by Rules 1 and 2, B can issue messages only while receiving one and must send out a message including a sub-key K_2 . Then, for B, two extra messages are required at least. On the other hand, A can derive the session key SK after receiving a sub-key K_2 from B. In addition, A requires an extra message to issue an authenticator $auth_A$ by Rules 4 and 5. Hence, three extra messages are required at least. Thus the protocol (j) cannot be implemented in four messages.

Table II summarizes the analyses of protocols (a),(b),...,(j). From these analyses, we can conclude that with the exceptions of protocol (a), these 3AKA-MA protocols cannot be implemented in four messages. Therefore, every 3AKA-MA protocol requires at least four messages for implementation.

Proof of Theorem 6: Assume that P and Q range over principals. C denotes a communicating channel and X and Y are messages. The followings define the notation used for logical analyses.

$C(X)$	The message X is transited via channel C ;
$r(C)$	The set of readers of channel C ;
$w(C)$	The set of writers of channel C ;
$P \triangleleft C(X)$	P sees $C(X)$. The message X is transited via channel C and can be observed by P . P must be a reader of channel C to read message X ;
$P \triangleleft X C$	P sees X via C . The message X is transited via channel C and can be received by P .

The used assumptions and logic rules [38,39,42] and the logical description of the proposed protocol are describes as follows.

The Assumptions used in [38,39,42], where U and V are S, A and B, and $U \neq V$:

(A1) $U \in r(C_{U,V})$: U can read from the channel $C_{U,V}$.

(A2) $U \equiv (w(C_{U,V}) = \{U,V\})$: U believes that U and V can write on $C_{U,V}$.

(A3) $U \equiv (V \parallel \sim \Phi \rightarrow \Phi)$: U believes that V only says what it believes

(A4) $U \equiv \#(N_U)$: U believes that N_U is fresh.

The Inference Rules of the Logic: Seeing rules

(S1) $\frac{P \triangleleft C(X), P \in r(C)}{P \equiv (P \triangleleft X | C), P \triangleleft X}$: If P receives and reads X via C , then P believes that X has arrived on C and P sees X .

(S2) $\frac{P \triangleleft (X, Y)}{P \triangleleft X, P \triangleleft Y}$: If P sees a hybrid message (X, Y) , then P sees X and Y separately. Interpretation rules

(I1) $\frac{P \equiv (w(C) = \{P, Q\})}{P \equiv (P \triangleleft X | C) \rightarrow Q | \sim X}$: If P believes that C can only be written by P and Q , then P believes that if P receives X via C , then Q said X .

(I2) $\frac{P \equiv (Q | \sim (X, Y))}{P \equiv (Q | \sim X), P \equiv (Q | \sim Y)}$: If P believes that Q said a hybrid message (X, Y) , then P believes that Q has said X and Y separately. Freshness rules

(F1) $\frac{P \equiv (Q | \sim X), P \equiv \#(X)}{P \equiv (Q | \sim X)}$: If P believes that another Q said X and P also believes that X is fresh, then P believes that Q has recently said X .

(F2) $\frac{P \equiv \#(X)}{P \equiv \#(X, Y)}$: If P believes that a part of a mixed message X is fresh, then it believes that the whole message (X, Y) is fresh. Rationality rules

(R1) $\frac{P \equiv (\Phi_1 \rightarrow \Phi_2), P \equiv \Phi_1}{P \equiv \Phi_2}$: If P believes that Φ_1 implies Φ_2 and P believes that Φ_1 is true, then P believes that Φ_2 is true.

According to the logic in [38, 39, 42], the proposed protocol is described as follows.

Step 1 $S \triangleleft (A, B, C_{S,A}(A, S, A, K_1))$
 $B \triangleleft (A, B)$

Step 2 $S \triangleleft (A, B, C_{S,B}(B, S, B, K_2))$

Step 3 $A \triangleleft (C_{S,A}(S, A, B, K_2))$
 $B \triangleleft (C_{S,B}(S, B, A, K_1))$

Step 4 $B \triangleleft (C_{A,B}(A, B, K_1, K_2))$
 $A \triangleleft (C_{A,B}(B, A, K_1, K_2))$

According the assumptions and logical analyses, the proposed protocol must realize the goals of authentication and key agreement:

Goal 1: $A \equiv A \xleftrightarrow{f(K_1, K_2)} B$: Participant A believes that $SK = f(K_1, K_2)$ is a symmetric key shared between participants A and B .

Goal 2: $B \equiv A \xleftrightarrow{f(K_1, K_2)} B$: Participant B also believes that $SK = f(K_1, K_2)$ is a symmetric key shared between participants A and B .

Goal 3: $A \equiv B \equiv A \xleftrightarrow{f(K_1, K_2)} B$: Participant A believes that B is convinced of $SK = f(K_1, K_2)$ is a symmetric key shared between participants A and B .

Goal 4: $B \equiv A \equiv A \xleftrightarrow{f(K_1, K_2)} B$: Participant B also believes that A is convinced of $SK =$

$f(K_1, K_2)$ is a symmetric key shared between participants A and B .

For achieving Goal 1, we have that

Sub-goal 1-1: $A \equiv K_1$ by using the interpretation rule (I3)),

Sub-goal 1-2: $A \equiv K_2$ by using the rationality rule (R1),

Sub-goal 1-3: $A \equiv (S || \sim C_{S,B}(B, S, B, K_2) \rightarrow C_{S,B}(B, S, B, K_2))$ by using assumption (A3)) and

Sub-goal 1-4: $A \equiv (S || \sim K_2)$.

Then, we have that

Sub-goal 1-5: $A \equiv \#(K_2)$ by using the freshness rules (F1, F2) and assumption (A11)),

Next, we use the interpretation rule (I1) and the seeing rule (S1), and have that

Sub-goal 1-6: $A \in r(C_{S,A})$ by using assumption (A1),

Sub-goal 1-7: $A \equiv (w(r(C_{S,A}) = \{A, S\}))$ by using assumption (A3) and

Sub-goal 1-8: $A \equiv \triangleleft C_{S,A}(K_2)$ by using the seeing rule (S2).

Thus, the proposed protocol provides Goal 1: $A \equiv A \xleftrightarrow{f(K_1, K_2)} B$.

Similarly, using the same derivation of Goal 1, we have that the proposed protocol provides

Goal 2: $B \equiv A \xleftrightarrow{f(K_1, K_2)} B$.

For achieving Goal 3, we have that

Sub-goal 3-1: $A \equiv ((B || \sim A \xleftrightarrow{f(K_1, K_2)} B) \rightarrow (B \equiv A \xleftrightarrow{f(K_1, K_2)} B))$ by using the rational-

ity rule (R1) assumption (19)) and

Sub-goal 3-2: $A \equiv (B || \sim A \xleftrightarrow{f(K_1, K_2)} B)$.

Then, we have that

Sub-goal 3-3: $A \equiv (B || \sim A \xleftrightarrow{f(K_1, K_2)} B)$ by using the freshness rule (F1) and

Sub-goal 3-4: $A \equiv \#(A \xleftrightarrow{f(K_1, K_2)} B)$ by using the freshness rule (F2) and assumption (A11)).

Next, we use the interpretation rule (I1) and the seeing rule (S1), and have that

Sub-goal 3-5: $A \in r(C_{A,B})$ by using assumption (A15),

Sub-goal 3-6: $A \equiv (w(C_{A,B}) = \{A, B\})$ by using assumption (A17) and

Sub-goal 3-7: $A \triangleleft C_{A,B}(A \xleftrightarrow{f(K_1, K_2)} B)$ by using the seeing rule (S2).

Thus, the proposed protocol provides Goal 3: $A \equiv B \equiv A \xleftrightarrow{f(K_1, K_2)} B$.

Similarly, using the same arguments of Goal 3, the proposed protocol provides Goal 4:

$B \equiv A \equiv A \xleftrightarrow{f(K_1, K_2)} B$.

Then the proof is concluded.

Proof of Theorem 7: Using similar arguments in [35], the proof also consists of a sequence of games starting at the game G_0^{ake} . The first game is the real attack against the DH-3AKA

protocol and the terminal game G_4^{ake} concludes that the adversary has a negligible advantage to break the AKE security of the DH-3AKA protocol.

Game G_0^{ake} : This game corresponds to the real attack. By definition, we have

$$Adv_{dh-3aka}^{ake}(\mathcal{A}_{ake}) = |2Pr[E_0] - 1|. \quad (1)$$

The following games G_1^{ake} and G_2^{ake} can be derived by using similar arguments of Theorem 5.2.

Game G_1^{ake} : This game simulates all oracles as in previous game except for replacing the long-term secret keys with two random numbers. Then, we have

$$|Pr[E_0] - Pr[E_1]| \leq 2 \cdot Adv_{sk}(\mathcal{A}_1). \quad (2)$$

Game G_2^{ake} : This game simulates all oracles as in previous game except for using two table lists to simulate *SymEnc* queries. Then, we have

$$|Pr[E_1] - Pr[E_2]| \leq \frac{q_1^2 + q_2^2}{2^l}, \quad (3)$$

Game G_3^{ake} : This game simulates all oracles as in previous game except for modifying the simulation of *Send* queries refereeing the flows containing g^x in Step 1 and g^y in Step 2 of the DH-3AKA protocol and the simulation of the *Test*(U^i) oracle to avoid relying on the knowledge of x, y and z used to compute the answer to these queries. Assume that $(X = g^x, Y = g^y, Z = g^{xy})$ is a random DDH triple. Using similar arguments in [35], we have that the set of random variables in G_2^{ake} is replaced by another set of identically distributed random variables in G_3^{ake} . G_2^{ake} is equivalent to G_3^{ake} and

$$Pr[E_2] = Pr[E_3]. \quad (4)$$

Game G_4^{ake} : This game simulates all oracles as in previous game except that all rules are computed using a triple (X, Y, Z) sample from a random distribution (g^x, g^y, g^z) , instead of a DDH triple. Using similar arguments in [35], we have

$$|Pr[E_3] - Pr[E_4]| \leq Adv_G^{ddh}(\mathcal{A}_{ddh}), \quad (5)$$

and the probability $Pr[E_4]$ is exactly $\frac{1}{2}$.

Combining Eqs (1), (2), (3), (4) and (5), we have

$$Adv_{dh-3aka}^{ake}(\mathcal{A}_{ake}) \leq \frac{q_1^2 + q_2^2}{2^{l-1}} + 4 \cdot Adv_{sk}(\mathcal{A}_1) + 2 \cdot Adv_G^{ddh}(\mathcal{A}_{ddh}).$$

Then the proof is concluded.

Proof of Theorem 8: The proof also consists of a sequence of games starting at the game G_0^{ake} . Each game G_i^{ake} defines the probability of the event E_i that the adversary wins this game, i.e. $c' = c$. The first game is the real attack against the DH-3AKA-MA protocol and the terminal game G_3^{ake} concludes that the adversary has a negligible advantage to break AKE security of the DH-3AKA-MA protocol. Assume that the challenger $\mathcal{A}_2$ attempts to break AKE security of the DH-3AKA protocol, and the adversary $\mathcal{A}_{ake}$ is constructed to break AKE security of the DH-3AKA-MA protocol. The challenger $\mathcal{A}_2$ returns the real session key SK or a random string to $\mathcal{A}_{ma}$ by flipping an unbiased coin $c \in \{0,1\}$. The adversary $\mathcal{A}_{ma}$ wins if it correctly guesses bit c .

The following game models that $\mathcal{A}_{ake}$ tries to distinguish the real session key from the random string.

Game G_0^{ake} : This game corresponds to the real attack. By definition, we have

$$Adv_{3aka1-ma}^{ake}(\mathcal{A}_{ake}) = |2\Pr[E_0] - 1|. \quad (6)$$

Game G_1^{ake} : This game simulates all oracles as in previous game except for using a table list H to simulate **Hash** queries involving A and B . Then, we have

$$|\Pr[E_0] - \Pr[E_1]| \leq \frac{q_3^2}{2^l}, \quad (7)$$

where $\mathcal{A}_{ma}$ makes q_3 **Hash** queries involving A and B .

Game G_2^{ake} : This game simulates all oracles as in previous game except for replacing the session key SK with a random number. Then, we can use $\mathcal{A}_{ma}$ to build an adversary $\mathcal{A}_2$ against the AKE security of DH-3AKA. First, $\mathcal{A}_2$ sets up the parameters, starts simulating the DH-3AKA-MA protocol and answers the oracle queries made by $\mathcal{A}_{ma}$ as follows.

- When $\mathcal{A}_{ma}$ make **Send** or **SymEnc** queries, $\mathcal{A}_2$ answers what the DH-3AKA protocol says to.
- When $\mathcal{A}_{ma}$ makes **Hash** queries, $\mathcal{A}_2$ answers corresponding authenticators to $\mathcal{A}_{ma}$ by making the same queries to the oracle Hash.
- When $\mathcal{A}_{ma}$ makes **Test** queries, $\mathcal{A}_2$ answers these queries using the bit c that it has previously selected and the session keys that has computed.

Accordingly, the probability that $\mathcal{A}_2$ outputs 1 when its Test oracle returns the real authentication keys is equivalent to the probability that $\mathcal{A}_{ake}$ correctly guesses the hidden bit c in game G_1^{ake} . Similarly, the probability that $\mathcal{A}_2$ outputs 1 when its Test oracle returns the random strings is equivalent to the probability that $\mathcal{A}_{ma}$ correctly guesses the hidden bit c in game G_2^{ake} . Thus, by Lemma 1, we have

$$|\Pr[E_1] - \Pr[E_2]| \leq Adv_{dh-3aka}^{ake}(\mathcal{A}_2) \quad (8)$$

At this time, no information on the hidden bit c is leaked to the adversary. It is straightforward that

$$\Pr[E_2] = \frac{1}{2}. \quad (9)$$

Combining Eqs (6), (7), (8) and (9), we have

$$Adv_{dh-3aka-ma}^{ake}(\mathcal{A}_{ma}) \leq 2Adv_{dh-3aka}^{ake}(\mathcal{A}_2) + \frac{q_3^2}{2^{l-1}}.$$

Then the proof is concluded.

Proof of Theorem 9: The proof also consists of a sequence of games starting at the game G_0^{ma} . The first game is the real attack against the DH-3AKA-MA protocol and the terminal game G_3^{ma} concludes that the adversary has a negligible advantage to break MA security of the DH-3AKA protocol. The challenger $\mathcal{A}_3$ attempts to break MA security for the DH-3AKA protocol and the adversary $\mathcal{A}_{ma}$ is constructed to break MA security for the DH-3AKA-MA protocol. The adversary $\mathcal{A}_{ma}$ wins this game if he successfully fakes the authenticator μ_A or μ_B .

Game G_0^{ma} : This game corresponds to the real attack. By definition, we have

$$Adv_{dh-3aka-ma}^{ma}(\mathcal{A}_{ma}) = 2\Pr[E_0]. \quad (10)$$

Game G_1^{ma} : Similar to G_1^{ake} in Theorem 8, this game simulates all oracles as in previous game except for using a table list $\mathbf{H}$ to simulate **Hash** queries involving A and B . Then, we have

$$|\Pr[E_0] - \Pr[E_1]| \leq \frac{q_3^2}{2^l}, \quad (11)$$

where $\mathcal{A}_{ma}$ makes q_3 **Hash** queries involving A and B .

Game G_2^{ma} : This game simulates all oracles as in previous game except for replacing the session key SK with a random number. Then, we can use $\mathcal{A}_{ma}$ to build an adversary $\mathcal{A}_3$ against the AKE security of 3AKA1. Using similar arguments for G_2^{ake} in Theorem 8, we have

$$|\Pr[E_1] - \Pr[E_2]| \leq Adv_{dh-3aka}^{ake}(\mathcal{A}_3) \quad (12)$$

No information on the authenticator is leaked to the adversary, and thus

$$\Pr[E_2] = \frac{1}{2^l}. \quad (13)$$

Combining Eqs (10), (11), (12) and (13), we have

$$Adv_{dh-3aka}^{ake}(\mathcal{A}_{ma}) \leq 2Adv_{dh-3aka}^{ake}(\mathcal{A}_3) + \frac{q_3^2 + 1}{2^{l-1}}.$$

Then the proof is concluded.

Acknowledgments

In this paper, T.F. Lee found out the problems in AKA protocols, collected related approaches about AKA protocols, developed new AKA protocols, provides security proofs and wrote the manuscript. T. Hwang assisted to develop the AKA protocols, contributed to security and performance analyses, questions discussion, and English language correction. This research was supported by Ministry of Science and Technology under the grants MOST 105-2221-E-320-003 and by Tzu Chi University under the grants TCRPP105004.

Author Contributions

Conceptualization: T-FL TH.

Data curation: T-FL.

Formal analysis: T-FL.

Funding acquisition: T-FL.

Investigation: T-FL TH.

Methodology: T-FL TH.

Project administration: TH.

Resources: T-FL TH.

Supervision: TH.

Validation: T-FL TH.

Writing – original draft: T-FL.

Writing – review & editing: T-FL TH.

References

1. Wen HA, Hwang T. Provably secure password-based authenticated key exchange protocols using bilinear pairing, Dissertation for Doctor of Philosophy. Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan; 2005.
2. Gong L. (1993) Lower bounds on messages and rounds for network authentication Protocols. *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 26–37.
3. Gong L. (1995) Efficient network authentication protocols: Lower bounds and implementations. *Distributed Computing*. 9(3): 131–145.
4. Gong L. Optimal authentication protocols resistant to password guessing attacks. *Proceedings of the 8th IEEE Computer Security Foundation Workshop*; 1995, p. 24–29.
5. Kwon T, Kang M, Song J. An adaptable and reliable authentication protocol for communication networks. *Proc. IEEE INFOCOM* 97; 1997, p. 738–745.
6. Kwon T, Kang M, Jung S, Song J. (1999) An improvement of the password-based authentication protocol (K1P) on security against replay attacks. *IEICE Trans. Commun.* E82-B(7): 991–997.
7. Kwon T, Song J. (1998) Authenticated key exchange protocols resistant to password guessing attacks. *IEE Proc.-Commun.* 145 (5): 304–308.
8. Kwon T, Song J. (1998) Efficient key exchange and authentication protocols protecting weak secrets, *IEICE Trans. Fundamentals*. E81-A(1): 156–163.
9. Lin CL, Sun HM, Hwang T. (2000) Three-party encrypted key exchange: Attacks and a solution. *ACM Operating Syst. Rev.* 34(4): 12–20.
10. Lin CL, Sun HM, Steiner M, Hwang T. (2001) Three-party encrypted key exchange without server public-keys. *IEEE Commun. Letters*. 5(12): 497–499.
11. Lee TF, Hwang T, Lin CL. (2004) Enhanced three-party encrypted key exchange without server public keys. *Computers & Security*. 23(7): 571–577.
12. Lu R, Cao Z. (2007) Simple three-party key exchange protocol. *Computers & Security*. 26(1): 94–97.
13. Steniner M, Tsudik G, Waidner M. (1995) Refinement and extension of encrypted key exchange. *ACM Operating Syst. Rev.* 29(3): 22–30.
14. Diffie W, Hellman M. (1976) New directions in cryptography, *IEEE Trans. Info. Theory*. 22(6): 644–654.
15. Lee CC, Li CT, Chiu ST, Lai YM. (2015) A new three-party-authenticated key agreement scheme based on chaotic maps without password table. *Nonlinear Dyn.* 79:2485–2495.
16. Amin R, Islam SK H, Biswas GP, Khan MK, Lengd L, Kumar N. (2016) Design of anonymity preserving three-factor authenticated key exchange protocol for wireless sensor network. *Computer Networks*,
17. Change CC, Chang YF. (2004) A novel three-party encrypted key exchange protocol. *Computer Standards & Interfaces*. 26(5): 471–476.
18. Chung HR, Ku WC. Impersonation attacks on a simple three-party key exchange protocol. *17th Information Security Conference*; 2007.
19. Chung HR, Ku WC. (2008) Three weaknesses in a simple three-party key exchange protocol. *Information Sciences*. 178(1): 220–229.
20. Lee TF, Hwang T. (2010) Simple password-based three-party authenticated key exchange without server public keys. *Information Sciences* 180(9): 1702–1714.
21. Molva R, Tsudik G, Van Herreweghen E, Zatti S. KryptoKnight authentication and key distribution system. *Proc. 1992 Eur. Symp. on Research in Computer Security—ESORICS*; 1992, p. 1–16.
22. Nam J, Paik J, Kim UM, Won D. (2007) Security weakness in a three-party pairing-based protocol for password authenticated key exchange. *Information Sciences*. 177(6): 1364–1375.
23. Neuman BC, Ts'o' T. (1994) Kerberos: An authentication service for computer networks. *IEEE Commun. Mag.* 32(9): 33–38.
24. Wen HA, Lee TF, Hwang T. (2005) Provably secure three-party password-based authenticated key exchange protocol using Weil pairing. *IEE Proc.-Commun.* 152(2): 138–143.
25. Lee CC, Chen SD, Chen CL. (2012) A computation-efficient three-party encrypted key exchange protocol. *Appl. Math. Inf. Sci.* 6(3): 573–579.

26. Lee CC, Li CT, Chang RX. (2013) An undetectable on-line password guessing attack on Nam et al.'s three-party key exchange protocol. *Journal of Computational Methods in Sciences and Engineering* 13: 455–460.
27. Abdalla M, Pointcheval D. Simple password-based authenticated key protocols. *Topics in Cryptology—CT-RSA 2005, Lecture Notes in Computer Science* 3376; 2005, p. 191–208.
28. Bellare M, Pointcheval D, Rogaway P. Authenticated key exchange secure against dictionary attacks. *Proc. of Advances in Cryptology—Eurocrypt 2000*; 2000, p. 122–138.
29. Bellare M, Rogaway P. Provably secure session key distribution—the three party case. *Proc. 27th ACM Symposium on the Theory of Computing*; 1995, p. 57–66.
30. Shoup V. Sequences of games: A tool for taming complexity in security proofs. manuscript, Available at www.shoup.net; 2005.
31. Steniner M, Buhler P, Eirich T, Waidner M. (2001) Secure password-based cipher suite for TLS. *ACM Trans. Inform. Syst. Security*. 4(2): 134–157.
32. Stallings W. *Cryptography and Network Security: Principles and Practice*, Second Edition. Upper Saddle River, NJ: Prentice Hall; 1999.
33. Joux A. (2000) A One Round Protocol for Tripartite Diffie–Hellman. *Algorithmic Number Theory, LNCS* 1838: 385–393.
34. Amin R, Islam SK H, Biswas GP, Debasis Giri, Khan MK, Kumar N. (2016) A more secure and privacy-aware anonymous user authentication scheme for distributed mobile cloud computing environments. *Security Comm. Networks*,
35. Lee TF, Lin CY, Lin CL, Hwang T. (2015) Provably secure extended chaotic map-based three-party key agreement protocols using password authentication. *Nonlinear Dyn.* 82(1): 29–38.
36. Lee CC, Li CT, Hsu CW. (2013) A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps. *Nonlinear Dyn.* 73:125–132.
37. Li X, Niu J, Kumari S, Khan MK, Liao J, Liang W. (2015) Design and analysis of a chaotic maps-based three-party authenticated key agreement protocol. *Nonlinear Dyn.* 80(3): 1209–1220.
38. Burrows M, Abadi M, Needham R. (1990) A logic of authentication, *ACM Trans. Computer Systems* 8 (1):18–36.
39. Buttyan L, Staamann S, Wilhelm U. (1998) A simple logic for authentication protocol design. *Proceedings of the 11th IEEE Computer Security Foundation Workshop*.
40. https://en.wikipedia.org/wiki/Security_parameter
41. Amin R, Biswas GP. (2016) A secure lightweight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. *Ad Hoc Networks* 36(1): 58–80.
42. Aslan HK. (2004) Logical analysis of AUTHMAC_DH: a new protocol for authentication and key distribution. *Computers & Security* 23: 290–299