Turkish Journal of Electrical Engineering and Computer Sciences

Volume 23 | Number 2 Article 17

1-1-2015

Efficient ID-based authentication and key agreement protocols for the session initiation protocol

HACI HAKAN KILINÇ

YOLGULY ALLABERDIYEV

TUĞRUL YANIK

SERDAR SÜER ERDEM

Follow this and additional works at: https://journals.tubitak.gov.tr/elektrik

 Part of the Computer Engineering Commons, Computer Sciences Commons, and the Electrical and Computer Engineering Commons

Recommended Citation

KILINÇ, HACI HAKAN; ALLABERDIYEV, YOLGULY; YANIK, TUĞRUL; and ERDEM, SERDAR SÜER (2015) "Efficient ID-based authentication and key agreement protocols for the session initiation protocol," *Turkish Journal of Electrical Engineering and Computer Sciences*: Vol. 23: No. 2, Article 17. https://doi.org/10.3906/elk-1207-102

Available at: https://journals.tubitak.gov.tr/elektrik/vol23/iss2/17

This Article is brought to you for free and open access by TÜBİTAK Academic Journals. It has been accepted for inclusion in Turkish Journal of Electrical Engineering and Computer Sciences by an authorized editor of TÜBİTAK Academic Journals. For more information, please contact academic.publications@tubitak.gov.tr.

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2015) 23: 560 – 579
© TÜBİTAK
doi:10.3906/elk-1207-102

Efficient ID-based authentication and key agreement protocols for the session initiation protocol

Hacı Hakan KILINÇ[1,*], **Yolguly ALLABERDİYEV**[2], **Tuğrul YANIK**[3], **Serdar Süer ERDEM**[1]

[1]Department of Electronics Engineering, Gebze Institute of Technology, Gebze, Kocaeli, Turkey
[2]Department of Computer Engineering, Fatih University, Büyükçekmeçe, İstanbul, Turkey
[3]Department of Computer Engineering, Celal Bayar University, Muradiye, Manisa, Turkey

Received: 25.07.2012 • **Accepted:** 18.04.2013 • **Published Online:** 23.02.2015 • **Printed:** 20.03.2015

Abstract: In a widely deployed VoIP system tens of thousands of clients compete for the SIP proxy server's authentication service. SIP protocol implementations have to meet certain QoS and security requirements. In this study new ID-based protocols are proposed for the SIP authentication and key agreement protocols. These protocols minimize the use of expensive pairing functions but still resist notable attacks. The security of the proposed protocols are analyzed and demonstrated with security proofs based on the BJM security model. Finally, the performance overhead of the proposed protocols are compared to ID-based SIP authentication and key agreement protocols given in the literature.

Key words: SIP, ID-based cryptography, authentication, key agreement

1. Introduction

Session Initiation Protocol (SIP) is designed to manage communication sessions such as audio and video transmissions through the Internet. The protocol can be used to create, modify, and terminate sessions among multiple parties. SIP is accepted as a 3GPP signaling protocol and a part of the IP Multimedia Subsystem (IMS) architecture [1].

Despite its benefits, SIP is subject to various security threats [2, 3]. Dantu et al. [4] studied the security problems of the VoIP infrastructure and presented features necessary for high level security at different levels. This study demonstrated that an effective authentication protocol is indispensable for almost all network components. Therefore, fortifying the SIP proxy with a secure and efficient SIP authentication and key agreement mechanism is of key importance.

SIP uses an authentication scheme that relies on HTTP Digest Authentication[5]. The HTTP Digest Authentication is vulnerable to server spoofing and offline password guessing attacks. To avoid these attacks, various authentication and key agreement schemes were proposed. These schemes can be categorized as Password Authenticated Key Exchange (PAKE)-based schemes [6, 7, 8, 9], hash and encryption-based shemes [10, 11, 12], public key cryptosystem (PKC)-based schemes that include RSA and elliptic curve cryptography (ECC) [13, 14], and ID-based schemes[15, 16, 17, 18, 19]. Each category presents different tradeoffs between performance and security.

Hash and encryption-based schemes offer high performance but can be vulnerable to a number of security attacks such as spoofing and offline password guessing attacks. Usually PAKE-based schemes have slightly

*Correspondence: hkilinc@gyte.edu.tr

higher computational costs than hash and encryption-based schemes. When designed properly they can resists spoofing and offline password guessing attacks. PKC-based schemes have much higher computational costs but eliminate the need of a preshared password. When fortified with a robust public key infrastructure (PKI), secure authentication schemes can be designed. Compared to the previous categories ID-based schemes have the highest computational cost, but they eliminate the costly PKI infrastructure and offer comparable security, which can be preferable to PKC-based schemes.

This study focuses on authentication protocols that employ ID-based cryptography. In an ID-based cryptosystem, the public keys can be chosen as arbitrary strings, which can be unique identifiers such as email addresses. There is no need to associate a user's identity to a public key using a certificate. The private key generator (PKG), a trusted third party, authorizes the public key and calculates the associated private key for each user. The PKG publishes a set of authenticated public system parameters as well. To verify a signature or send an encrypted message, only the signer's identity or receiver's identity and the system parameters are required. For a user, obtaining and authenticating the system parameters is a one-time process. In traditional PKCs each user needs to validate a certificate, which can involve the verification of a certificate path if multiple certificate authorities are present. Furthermore, in a traditional PKC, each user needs to maintain a public key directory for other users, which is not necessary in ID-based cryptosystems.

The heavy computational cost of the bilinear pairing function, the fact that the PKG knows the private keys of the users, and the need for a secure channel to transfer private keys to the users are drawbacks of ID-based SIP authentication schemes. The proposed ID-based SIP authentication and key agreement schemes in the literature greatly suffer from high computational costs caused by the multiple use of bilinear pairing functions. These schemes can resist most attacks, but not collusion attacks. Since the private keys are known by the PKG, it can share sensitive information with other parties causing collusion attacks.

In this study, we propose specific ID-based authentication and key agreement schemes for client-to-server and client-to-client communication that have significantly better performances compared to previously proposed ID-based authentication and key agreement schemes. In addition, we show that the schemes we propose are provably secure according to the BJM security model.

The next section explains the related work. Section 3 gives background information about the proposed protocols. Section 4 explains proposed protocols and security proofs. Section 5 evaluates the performance and security of the proposed protocols and compares them to the ID-based SIP authentication protocols proposed in the literature respectively. Section 6 concludes the paper. The Appendix includes the message flow diagrams of the schemes compared in this work.

2. Related work

In the literature, several schemes have been proposed for SIP authentication and key agreement. Authentication and key agreement schemes based on ID-based cryptosystems are more recent and provide some advantages compared to the PKC-based and ECC-based schemes that were mentioned in Section 1. In this section, we will examine the proposed SIP authentication schemes based on ID-based cryptography.

Ring et al. [15] proposed an authentication scheme that relies on ID-based signatures and time-dependent nonce values. This scheme consists of 2 parts: the authentication part and the key agreement part. For the key agreement part the modified ID-based protocol of Chen and Kudla[20] was proposed. To realize this scheme an additional signature field in the standard existing authentication protocol is needed. The ID-based signature scheme requires the computation of multiple pairing functions and point multiplication, which cause significant computational overhead and delay.

Ring et al. did not specify the signature scheme for the authentication, whereas Han et al. [16] proposed to use a combination of Hess's ID-based signature scheme[21]. To decrease the delay for session key generation in the key establishment part, the one-way authenticated ID-based key agreement protocol of Okamoto et al. [22] was proposed. This protocol reduces the computational overhead compared to Ring et al.'s 2-way key agreement scheme. Each party can compute the session key simultaneously using bilinear pairing rules. Due to multiple pairing computations, the computational cost is significant. A collusion attack is also applicable.

Patil and Willis [23] proposed authentication protocols using the ID-based signature and signcryption schemes. The authors use the "Identity" and "Identity-Info" fields that are defined in RFC 4474 for authentication purposes. Although an RSA-based signature scheme is recommended in RFC 4474, the authors propose ID-based signature and signcryption schemes and compare the proposed schemes to an RSA-based scheme. The purpose of the signcryption schemes is to achieve authenticity and confidentiality together. The authors propose different ID-based signature and signcryption schemes for single and hierarchical domain environments.

Wang and Zhang [17] proposed a secure mutual authentication and key agreement (SAKA) protocol that relies on certificateless public key cryptography (CL-PKC), which was established by Al-Riyami and Paterson[24]. This scheme consists of the system initiation and the authentication and key agreement stages. The authors acknowledged their contribution as the removal of the key escrow feature and support for interdomain peer-to-peer connections where clients in different domains can communicate directly. In this scheme, the key generation center (KGC), which is a variation of the PKG, does not know the private keys of the clients. A security proof based on the CK-Model was provided.

Kilinc et al. [18] replaced the standard HTTP Digest Authentication with the ID-based signature schemes of Choon and Cheon[25]. In this comparative study, real performance data of the proposed schemes were obtained by using the Pairing Based Cryptography (PBC) Library[26] to implement the proposed schemes into an SIP proxy server (OpenSIPs[27]), which is open source.

Ni et al. [19] proposed an ID-based authenticated key agreement mechanism relying on a signature scheme. The scheme is based on ECC and does not require the computation of a pairing function. The proposed mechanism employs the CL-PKC method to construct a secret key, which is only known by the client side, avoiding the key escrow problem. To sign and verify, a set of parameters are needed, where the identity is one of them. To calculate the client's and server's public key an identity-based public key building parameter is sent by the related party. This scheme and its performance falls between the traditional PKC and the ID-based cryptography.

3. Background information

3.1. SIP summary

SIP is a signaling and application-layer control protocol commonly used for VoIP communication. The traditional SIP architecture is a client-server architecture. The general idea of SIP is to establish sessions among different user agents on any Internet platform. SIP networks includes 5 kinds of logical entities such as user agents, registrar servers, proxy servers, location servers, and redirect servers. The user agents are the end users, which generate or receive SIP messages. The SIP proxy server is an intermediate entity that can hold session information and redirect SIP messages to other proxy servers or user agents.

SIP security mechanisms can be categorized as hop-by-hop and end-to-end mechanisms. TLS, which is a transport layer security mechanism, and IPsec, which is a network layer security mechanism, are hop-by-hop security mechanisms. S/MIME and the HTTP Digest Authentication are end-to-end security mechanisms. For

media security, the Secure Real-Time Transport Protocol (SRTP) and the Datagram Transport Layer Security (DTLS) protocol are used.

3.2. Identity-based cryptosystem

The ID-based cryptography concept was first suggested by Shamir in 1984[28]. The ID-based cryptosystem includes ID-based encryption (IBE) and ID-based signature (IBS). The essential idea is to set the public key to an arbitrary string. An IP address, an e-mail address, or different types of identities can be assigned as a public key for a user or a node. Public keys are derived from a user identifier. The main purpose is the elimination of the public key distribution infrastructure and certificate management. The private key based on the public key is calculated by the PKG, which is recognized as a trusted third party. After that, the PKG uses a secure channel to submit the private key to the client.

The first usable IBE schemes were realized by Boneh and Franklin in 2001. They used elliptic curve cryptography and Weil pairings on elliptic curves[29]. Following this work many IBE and IBS schemes such as Hess and Cha-Cheon algorithms were proposed.

Weil pairing is realized by mapping 2 points of group G_1 on elliptic curve $E(F_q)$ to the finite field F_q. Some of the very practical features of Weil pairing can be summarized as follows:

1. Bilinearity: $\forall P, Q, R \in G_1$,

   $e : G_1 \times G_1 \rightarrow G_2$

   $e(aP, bQ) = e(bP, aQ) = e(P, Q)^{ab}$,

   $e(P + Q, R) = e(P, R)e(Q, R)$, and

   $e(P, Q + R) = e(P, Q)e(P, R)$;

2. Nondegeneracy:

   $e(P, Q) = 1$, for all $Q \in G_1$, iff $P = O$, where O is point at infinity

3. Computability:$\forall P, Q \in G_1$, the computation of the function $e(P, Q)$ is efficient.

3.3. The BJM security model

The work of Chen and Kudla adapted the security model proposed by Blake-Wilson et al.[30] (BJM security model) to the identity-based setting. The security model proposed by Blake-Wilson et al. was adapted to the public key setting from the security model initially proposed by Bellare and Rogaway [31]. In this work the security proofs of the identity-based authentication protocols will rely on Chen and Kudla's BJM security model adaptation.

The BJM security model consists of a set of participants denoted by $\mathcal{U}$ and an adversary denoted by $\mathcal{E}$. The participants are modeled by oracles denoted by $\Pi^n_{I,J}$, where participant I believes that it is conducting a protocol session with participant J for the nth time. The oracles answer queries by receiving and sending messages, which are recorded to transcripts. The adversary is modeled by a probabilistic polynomial time Turing machine and can conduct certain queries on all oracles, which consist of the participant's oracles and random oracles. The adversary has control over the communications and can rely, modify, delay, interleave, or delete messages. The participant oracles can communicate only through the adversaries' queries.

To define the groups G_1, G_2 and the bilinear map e, and to assign a long-term master key to the PKG, a *setup* algorithm is deployed. For each participant the PKG will calculate the private key based on the public key, which is related to the participant's identifier.

The adversary can conduct the following queries:

- Create: Using this query adversary $\mathcal{E}$ can create and setup a new participant. The public key of the participant will be obtained from the identity. The PKG will generate a private key using this public key. An oracle that will model the participant will be created as well. The adversary will obtain the public key of the participant.
- Send: The adversary $\mathcal{E}$ can use this query to send a message to an oracle. If the receiving oracle is $\Pi^n_{I,J}$, it will assume that the message was sent by participant J. The adversary can send a special message λ, which will instruct the oracle to initiate a session with participant J. An oracle becomes an initiator or a responder oracle according to the first message it receives.
- Reveal: The adversary uses this query to ask an oracle to reveal the session key it holds.
- Corrupt: Using this query, the adversary asks an oracle to reveal the long-term private key it is holding.

Before explaining the *test* query, which is the last query, the states of an oracle should be explained. According to the BJM security model an oracle can be in the *accepted* state, *rejected* state, * state, *opened* state, or *corrupted* state. To reach the *accepted* state an oracle has received proper messages, is holding a session key, and decides to accept. The *rejected* state is reached when an oracle has decided to abort the session without establishing a session key. The * state means that no decision is reached yet. The *opened* state is reached when the oracle has answered a reveal query. Finally, the *corrupted* state is reached when the oracle has answered a corrupt query.

When 2 oracles reach the accepted state receiving proper messages generated by the communicating oracle, where one of the oracles is an initiator oracle, they have had a matching conversation[31].

- Test: A test query is actually used to model an attack. After creating participants and initiating and conducting sessions among participants, the adversary can ask a single test query to an oracle $\Pi^n_{I,J}$ that has accepted and is unopened where none of the participants I or J have been corrupted. In addition, there should be no oracle $\Pi^t_{J,I}$ that had a matching conversation and is opened. The oracle receiving the test query should flip a fair coin denoted by $b \in 0, 1$. If $b = 0$ the oracle should return the session key it is holding; otherwise, it should return a k-bit random key. The adversary $\mathcal{E}$ should output b' as its guess to b. The function $Advantage^{\mathcal{E}}(k)$ is the probability the adversary has in distinguishing the session key from a random string. It is defined as:

  $Advantage^{\mathcal{E}}(k) = |Pr[b^{'} = b]1/2|.$

As stated in [30] an authenticated key agreement protocol (AK protocol) is a key agreement protocol where both parties are assured that no other party can possibly compute the key agreed upon. However, if both parties want to make sure that each party actually computed the key agreed upon, a key confirmation part should be included. Such a protocol is called an authenticated key agreement with key confirmation (AKC protocol).

According to the BJM security model, an AK protocol is secure if *Definition 1* given below is met.

- **Definition 1**: A protocol is an AK protocol if:

  1. In the presence of the benign adversary on $\prod_{i,j}^{n}$ and $\prod_{j,i}^{t}$ both oracles always accept holding the same session key, and this key is distributed uniformly at random on $\{0,1\}^k$; and if for every adversary $\mathcal{E}$.

  2. If uncorrupted oracles $\prod_{i,j}^{n}$ and $\prod_{j,i}^{t}$ have matching conversations then both oracles accept and hold the same session key.

  3. $Advantage^{\mathcal{E}}(k)$ is negligible.

According to the BJM security model an AKC protocol is secure if *Definition 2* given below is met.

- **Definition 2**: A protocol is an AKC protocol if the first 3 conditions of Definition 1 and the below condition are satisfied.

  1. The probability of $No-Matching^{\mathcal{E}}(k)$ is negligible.

$No-Matching^{\mathcal{E}}(k)$ as the additional condition in *Definition 2* denotes the event where an uncorrupted oracle $\Pi_{I,J}^{n}$ has accepted under the attack of adversary $\mathcal{E}$ but there is no other uncorrupted oracle $\Pi_{J,I}^{t}$ that had a matching conversation with it.

3.4. Security features

AK and AKC protocols should have the following security features:

- Known-key security: A protocol is secure against known-key attack when the adversary cannot compromise a shared key knowing several shared keys of past sessions.

- Forward secrecy: A protocol has the forward secrecy feature if the adversary cannot obtain the previous shared keys by compromising the long-term private keys of multiple parties. This definition can be expressed as partial forward security if the compromise of some parties' long-term private keys cannot reveal any previously shared key or extended to perfect forward secrecy if the compromise of all parties' long-term private keys cannot reveal any previously shared key.

- Ephemeral key reveal resilience: The ephemeral key reveal can be realized when a revealed short-term key (random key) can be used to obtain the shared key of that session.

- Key compromise impersonation resilience: The key compromise impersonation attack is applicable if the long-term key of a valid entity is compromised and with that entity the adversary can establish a session key by masquerading as another party [32].

- Unknown key share resilience: A party should not be persuaded into calculating a shared key with another party that masquerades as another party. A man-in-the-middle attack is conducted to achieve such a goal with both parties that want to communicate with each other.

4. Proposed protocols and their security analysis

4.1. Assumptions

There are some assumptions and common parameters for the proposed protocols.

- All parties use the same elliptic curve parameters, and all private keys are stored in secure environments.
- SIP servers have the functionality of a PKG, which produces public/private key pairs and is trusted by all entities in the system.
- The bilinear function e, H_1, H_2, P_{pub}, P, and some group parameters are published by the PKG.
- Both the client and server should agree on the following system parameters:
  - Cyclic groups $(G_1, +)$ and $(G_2, .)$, which have the same prime order l.
  - $P \in E(Fq)$ is the generator point of the group G_1.
  - ID_S is identity of the server.
  - s is the private key of the PKG.
  - P_{pub} is the public key of the PKG.
  - $e : G_1 \times G_1 \rightarrow G_2$ is a pairing function with bilinearity and nondegeneracy properties.
  - ID_A is the identity of Alice.
  - Q_A $(= H_1(ID_A))$ is the public key of Alice.
  - S_A $(= sQ_A)$ is the private key of Alice.
  - ID_B is the identity of Bob.
  - Q_B $(= H_1(ID_B))$ is the public key of Bob.
  - S_B $(= sQ_B)$ is the private key of Bob.
  - $F(.)$ is a one-way hash function.
  - $F^*(.)$ is another one-way hash function used to obtain the shared secret key.

4.2. Protocols and security analysis

We propose 3 ID-based authentication and key agreement protocols that target client-to-server (Protocol I and Protocol II) and client-to-client (Protocol III) communications. The main difference between Protocol I and II is the use of the pairing function. The proposed protocols provide various security features. In Section 4.3, we will give a formal security proof based on the BJM security model for the proposed protocols. The BJM model does not allow the adversary to ask test queries to corrupted oracles. Therefore, the security model cannot model the key compromise impersonation attack and the forward secrecy properties. In this section we will explain the proposed protocols and conduct their security analysis based on heuristic arguments. We will

Table 1. Protocol I: Client-to-server authentication with pairing.

Step 1. *Alice* $\rightarrow$ *Server* :	Alice generates a random number a and computes $T_{A1} = aP + aS_A$ and $T_{A2} = aQ_A$ using her private key. Then she sends to the server a REQUEST message with session identifier *sid*. $REQUEST$ $\{sid, T_{A1}, T_{A2}\}$	$a \in Z_r$ $T_{A1} = aP + aS_A$ $T_{A2} = aQ_A$
Step 2. *Server* $\rightarrow$ *Alice* :	Receiving the REQUEST message, the server computes $a^{'}P = T_{A1} - sT_{A2}$. Then it derives a random number b and computes $T_B = bP$, a key $K = e(sa^{'}P, bQ_A)$, and the $F(K, ID_S)$ values by using its identity ID_S. The CHALLENGE message with session identifier *sid* is sent to Alice. The shared session key K_S will be derived by $K_S = F^*(K)$. $CHALLENGE$ $\{sid, T_B, F(K, ID_S)\}$	$a^{'}P = T_{A1} - sT_{A2}$ $b \in Z_r$ $T_B = bP$ $K = e(sa^{'}P, bQ_A)$ $K_S = F^*(K)$ $F(K, ID_S)$
Step 3. *Alice* $\rightarrow$ *Server* :	Upon receiving the CHALLENGE message, Alice calculates the key $K^* = e(aT_B, S_A)$ using her random number and private key. After that, she calculates the hash value $F(K^*, ID_S)$ and compares it with $F(K, ID_S)$. If they are equal, Alice authenticates the server and sends a RESPONSE message. The shared session key K_S will be derived by $K_S = F^*(K^*)$. $RESPONSE$ $\{sid, F(K^*, ID_S, ID_A)\}$	$K^* = e(aT_B, S_A)$ $F(K^*, ID_S)$ $F(K, ID_S)$?= $F(K^*, ID_S)$ $K_S = F^*(K^*)$ $F(K^*, ID_S, ID_A)$
Step 4. *Server:*	After receiving the RESPONSE message, the server computes $F(K, ID_S, ID_A)$, and compares it with $F(K^*, ID_S, ID_A)$. If they match, the server authenticates Alice.	$F(K, ID_S, ID_A)$ $F(K, ID_S, ID_A)$?= $F(K^*, ID_S, ID_A)$

consider the man-in-the-middle attack, the replay attack, forward secrecy, the key compromise impersonation attack, the ephemeral key reveal, and the known-key attack.

Protocol I: Protocol I in Table 1 relies on the decisional bilinear Diffie–Hellman (DBDH) problem. A known-key attack or a reply attack cannot be realized because each session key relies on random ephemeral keys a and b. Each protocol will generate a new shared key that does not leak any information about other session keys. The man-in-the-middle attack cannot be realized because the adversary must calculate $e(aT_B, S_A)$ in order to share the same key with Alice. To calculate this value, the adversary has to know s and b or S_A and a.

Protocol I provides perfect forward secrecy because the compromise of long-term secret keys of all parties does not reveal the past session keys where each session key relies on random ephemeral keys a and b.

It can resist the ephemeral key reveal attack unless both ephemeral keys are revealed. To calculate $e(abP, sQ_A) = e(abP_{pub}, Q_A)$ it can be assumed that the adversary captured the ephemeral key a and conducted a man-in-the-middle attack. The adversary will choose an a' to replace a, but to calculate $e(a'bP_{pub}, Q_A)$, one needs to know the value of b. The assumption that the server side ephemeral key is revealed is a strong one.

Although by definition the key compromise impersonation attack is pointed toward client-to-client protocols, it can be applied to this protocol. Even if the adversary compromises a client's long-term secret key, this protocol can resist the key compromise impersonation attack because the adversary does not know aP, which is required to calculate the session key.

Table 2. Protocol II: Client-to-server authentication without pairing.

Step 1. *Alice* $\rightarrow$ *Server* :	Alice generates a random number a and computes $T_{A1} = aP + aS_A$ and $T_{A2} = aQ_A$ using her private key. Then she sends to the server a REQUEST message with session identifier *sid*. $REQUEST$ $\{sid, T_{A1}, T_{A2}\}$	$a \in Z_r$ $T_{A1} = aP + aS_A$ $T_{A2} = aQ_A$
Step 2. *Server* $\rightarrow$ *Alice* :	Receiving the REQUEST message, the server computes $a^{'}P = T_{A1} - sT_{A2}$. Then it derives a random number b and calculates $T_B = bP + a'P + sQ_A$, a key $K = ba^{'}P$, and the $F(K, ID_S)$ values by using its identity ID_S. The shared session key K_S will be derived by $K_S = F^*(K)$. The CHALLENGE message is sent to Alice. $CHALLENGE$ $\{sid, T_B, F(K, ID_S)\}$	$a^{'}P = T_{A1} - sT_{A2}$ $b \in Z_r$ $T_B = bP + a'P + sQ_A$ $K = ba^{'}P$ $K_S = F^*(K)$ $F(K, ID_S)$
Step 3. *Alice* $\rightarrow$ *Server* :	Upon receiving the CHALLENGE message, Alice calculates $b'P = T_B - aP - S_A$ and the key $K^* = ab'P$ using her random number. Then she calculates the hash value $F(K^*, ID_S)$ and compares it with $F(K, ID_S)$. If they are equal, Alice authenticates the server and sends a RESPONSE message to the server. The shared session key K_S will be derived by $K_S = F^*(K^*)$. $RESPONSE$ $\{sid, F(K^*, ID_S, ID_A)\}$	$b'P = T_B - aP - S_A$ $K^* = ab'P$ $F(K^*, ID_S)$ $F(K, ID_S)$?= $F(K^*, ID_S)$ $K_S = F^*(K^*)$ $F(K^*, ID_S, ID_A)$
Step 4. *Server:*	After receiving the RESPONSE message, the server computes $F(K, ID_S, ID_A)$, and compares it with $F(K^*, ID_S, ID_A)$. If they match, the server authenticates Alice.	$F(K, ID_S, ID_A)$ $F(K, ID_S, ID_A)$?= $F(K^*, ID_S, ID_A)$

Protocol II: Protocol II in Table 2 relies on the elliptic curve decisional Diffie–Hellman (ECDDHP) problem. Similar to Protocol I, a known key attack or a replay attack could not be realized because of random ephemeral keys a and b. A man-in-the-middle attack could not be realized because the adversary does not know a or S_A. It provides perfect forward secrecy because the compromise of long-term secret keys of all parties does not reveal the past session keys where each session key relies on random ephemeral keys a and b.

Protocol II can resist the ephemeral key reveal attack. When the client's ephemeral key is revealed the adversary cannot calculate S_A even knowing aQ_A and aS_A. When the server side ephemeral key is revealed the adversary cannot calculate aP. The protocol is secure against the key compromise impersonation attack as well. Knowing S_A, the adversary cannot calculate aP and bP, which are necessary to calculate K or K^*.

Protocol III: Protocol III in Table 3 relies on the DBDH problem. A replay attack is not possible because of the random ephemeral keys. The man-in-the-middle attack cannot be realized because an adversary needs S_A or S_B to calculate a session key. It provides perfect forward secrecy because the session keys rely on random ephemeral keys.

The proposed protocol can resist an ephemeral key reveal attack. Even if the adversary captures a ephemeral private key, it is clear that the shared key K cannot be calculated from the demonstrated key calculation. Without S_A or S_B, capturing a and b is not enough to calculate K.

The key-compromise impersonation attack is not applicable in here. Suppose an adversary captures the long-term private key S_A of Alice and pretends to be Bob to establish communication with her. However, he

Table 3. Protocol III: Client-to-client authentication.

Step 1. *Alice* $\rightarrow$ *Bob* :	Alice generates a random number a and computes $T_A = aP + aQ_A$ using her public key. Then she sends to Bob a REQUEST message with session identifier *sid*. $REQUEST\ \{sid, T_A\}$	$a \in Z_r$ $T_A = aP + aQ_A$
Step 2. *Bob* $\rightarrow$ *Alice* :	After Bob receives the REQUEST message, he derives a random number b and computes $T_B = bP + bQ_B$ and a key $K = e(bP + bQ_A + T_A, P_{pub} + S_B)$, and the $F(K, ID_B)$ values by using his identity ID_B. The shared session key K_S will be derived by $K_S = F^*(K)$. After that, he sends the CHALLENGE message with session identifier *sid* given below to Alice. $CHALLENGE\ \{sid, T_B, F(K, ID_B)\}$	$b \in Z_r$ $T_B = bP + bQ_B$ $K = e(bP + bQ_A + T_A, P_{pub} + S_B)$ $K_S = F^*(K)$ $F(K, ID_B)$
Step 3. *Alice* $\rightarrow$ *Bob* :	Upon receiving the CHALLENGE message, using her random number Alice calculates the session key $K^* = e(P_{pub} + S_A, T_B + aQ_B + aP)$. Then she calculates the hash value $F(K^*, ID_B)$ and compares it with $F(K, ID_B)$. If they are equal, Alice authenticates Bob and sends a RESPONSE message to him. The shared session key K_S will be derived by $K_S = F^*(K^*)$. $RESPONSE\ \{sid, F(K^*, ID_B, ID_A)\}$	$K^* = e(P_{pub} + S_A, T_B + aQ_B + aP)$ $F(K^*, ID_B)$ $F(K, ID_B)\ ?= F(K^*, ID_B)$ $K_S = F^*(K^*)$ $F(K^*, ID_B, ID_A)$
Step 4. *Bob:*	After receiving the RESPONSE message, Bob computes $F(K, ID_B, ID_A)$ and compares it with $F(K^*, ID_B, ID_A)$. If they match, Bob authenticates Alice.	$F(K, ID_B, ID_A)$ $F(K, ID_B, ID_A)\ ?=$ $F(K^*, ID_B, ID_A)$

cannot calculate $aQ_B + aP$. If the adversary captures the long-term private key S_B of Bob and pretends to be Alice to establish communication with him, he will not be able to do so because he cannot calculate $bP + bQ_A$.

The proposed protocol has the key escrow property where the PKG is capable of recovering the session keys by capturing the message exchanges and using the master secret key. To disable this property, both parties should send the extra points aP_{pub} and bP_{pub} separately and the shared key should be calculated as $F^*(K, abP_{pub})$.

$$
\begin{aligned}
K &= e(P_{pub} + S_A, T_B + aQ_B + aP) \\
&= e(P_{pub} + S_A, T_B)e(P_{pub} + S_A, aQ_B + aP) \\
&= e(sP + sQ_A, bP + bQ_B)e(sP + sQ_A, aQ_B + aP) \\
&= e(bP + bQ_A, P_{pub} + S_B)e(aP + aQ_A, S_B + P_{pub}) \\
&= e(bP + bQ_A + T_A, P_{pub} + S_B) \\
&= K^*
\end{aligned}
$$

4.3. Security proofs of Protocols I and III

In the proposed authentication and key agreement protocols the key confirmation part is realized within the last 2 steps. We can assume that the previous steps work as an AK protocol. The formal proofs given will be for the AK protocols. It is supposed that the DBDH problem [33] is hard. The proof of the security of Protocols I and III relies on this notion.

Definition 3 - The DBDH problem: Let G_1, G_2 be 2 groups of prime order q. Let P be a generator of G_1 and $e : G_1 \times G_1 \rightarrow G_2$ be an admissible bilinear mapping of 2 elements in G_1 to an element in G_2.

Instance: (P, xP, yP, zP, r) for some $x, y, z, r \in Z_n$.

Output: Yes if $r \equiv e(P,P)^{xyz} \mod p$.

Adversary $\mathcal{E}$ in solving decisional Diffie–Hellman (DDH) is defined by:

$Adv_{\mathcal{E}}^{DDH} = Pr[\mathcal{E}(P, xP, yP, zP, r) = 1]$ - $Pr[\mathcal{E}(P, xP, yP, zP, e(P,P)^{xyz}) = 1]$: $x, y, z, r \in Z_n$.

The DBDH assumption: $Adv_{\mathcal{E}}^{DBDH}$ is negligible.

Theorem I: Protocols I and III are secure AK protocols when the DBDH assumption is given. It is also assumed that the adversary $\mathcal{E}$ does not make any reveal queries and the hash functions used are random oracles.

Proof: Similar to the proof given by Chen and Kudla [20], the first 2 conditions in *Definition 1* are satisfied when the oracles follow the protocol and accept with matching conversations. They will hold the same session key K_S due to the bilinear pairing function. In addition, the key will be distributed uniformly at random on $\{0,1\}^k$ because the hash functions deployed are random oracles.

For the third condition we will assume that the adversary $\mathcal{E}$ can conduct a successful test query with nonnegligible advantage $\eta(k)$ in time $\tau(k)$. We further assume that during this attack the adversary makes at most T_F queries to the random oracle H_F and T_C create queries.

Using $\mathcal{E}$, we will build an algorithm $\mathcal{D}$ that solves the DBDH problem by calculating the output $r \equiv e(P,P)^{xyz} \mod p$ given the input parameters G_1, G_2,e,xP,yP,zP explained in *Definition 3* with nonnegligible probability. $\mathcal{D}$ will simulate the oracles for all parties and maintain 2 additional random oracles H_C and H_F. H_F can be queried by the adversary at any time whereas H_C is not directly available to the adversary. The H_C oracle will only answer create queries. $\mathcal{D}$ will run a setup algorithm assigning the master key of the PKG to xP.

Before the test query $\mathcal{D}$ will choose 2 distinct random values I and J, which are equal to or smaller than T_C, and a value l, which is equal to or smaller than T_F. $\mathcal{D}$ will start the adversary $\mathcal{E}$ and answer its queries as explained below.

Queries to H_C and H_F: A random oracle answers queries in a consistent way. The H_C oracle is simulated with an H_C-list that holds $< ID_i, Q_i >$ tuples. If the H_C oracle is queried with ID_i and the tuple $< ID_i, Q_i >$ already exists the oracle answers with Qi. If the tuple does not exist and the query is the Jth query the oracle answers with $Q_i = yP$ and adds the tuple $< ID_i, Q_i >$ to its H_C-list. Otherwise, the oracle answers with $Q_i = r_iP$ where $r_i \in Z_n$ is selected randomly and adds the tuple $< ID_i, Q_i >$ to its H_C-list. Similar to H_C, the H_F oracle is simulated with an H_F-list. The difference is that it answers distinct queries randomly.

Create queries: The H_C oracle can only be queried via a create query. The create query will deliver an ID_i and $\mathcal{D}$ will simulate this query by querying H_C to set up the public key $Q_i = r_iP$ and the private key $S_i = r_ixP$. The public key will be delivered to $\mathcal{E}$. If the query to H_C is the Jth query, or in other words the Jth participant is created, the public key will be $Q_J = yP$. In Protocol I, $\mathcal{D}$ will be able to calculate the secret key for this participant whereas in Protocol III it will not.

Corrupt queries: If $\mathcal{E}$ queries $\mathcal{D}$ for I or J, $\mathcal{D}$ gives up. Otherwise, $\mathcal{D}$ answers by revealing the long-term private keys.

Send queries for Protocol I: For a normal oracle, $\mathcal{D}$ answers all send queries as specified. Protocol I is a client-server protocol. For the first send query to an initiator oracle, $\mathcal{D}$ generates a random value to calculate

the oracle's contribution. If $\mathcal{E}$ asks $\Pi^t_{J,I}$ for its first send query where $\Pi^t_{J,I}$ is an initiator, $\mathcal{D}$ will generate a random $j \in Z_n$ and answer with $jP + jS_j$. If $\mathcal{E}$ asks $\Pi^n_{I,J}$ for its first send query where $\Pi^n_{I,J}$ is an responder oracle, $\mathcal{D}$ will generate a random $s_n \in Z_n$ and answer with s_nzP.

Send queries for Protocol III: For a normal oracle, $\mathcal{D}$ answers all send queries as specified. For the first send query to an initiator oracle, $\mathcal{D}$ generates a random value to calculate the oracle's contribution. If $\mathcal{E}$ asks $\Pi^t_{J,I}$ for its first send query where $\Pi^t_{J,I}$ is an initiator, $\mathcal{D}$ will generate a random $j \in Z_n$ and answer with $jP + jQ_j$. If $\mathcal{E}$ asks $\Pi^n_{I,J}$ for its first send query where $\Pi^n_{I,J}$ is a responder oracle, $\mathcal{D}$ will generate a random $s_n \in Z_n$ and answer with $s_nzP + s_nr_izP$.

Reveal queries: The adversary is not allowed to ask reveal queries. Therefore, $\mathcal{D}$ will not answer any reveal queries.

Test queries for Protocol I: If $\mathcal{E}$ asks a test query to an oracle other than the $\Pi^n_{I,J}$ oracles, $\mathcal{D}$ aborts. Otherwise, the $\Pi^n_{I,J}$ must have accepted. I and J must be uncorrupted. Assuming that $\Pi^n_{I,J}$ received $jP+jS_j$ and calculated $jP + jS_j - jxyP = jP + jS_j - jS_j^{'} = jP$ before accepting, the oracle will hold the session key $F^*(e(xjP, s_nzyP))$. Because $\mathcal{D}$ cannot calculate this session key, it will not be able to simulate the test query in a correct way. $\mathcal{D}$ will give a random answer in this case. If $\mathcal{D}$ does not abort for some reason and $\mathcal{E}$ does not detect the random answer, $\mathcal{E}$'s probability of success is still $\eta(k)$. $\mathcal{E}$ can distinguish the session key from a random value only if it has queried oracle H_F for the value of $F^*(e(xjP, s_nzyP))$ with nonnegligible probability $\eta(k)^{'}$. The number of H_F queries is bounded by T_F.

$\mathcal{E}$'s state can become undefined if it detects $\mathcal{D}$'s inconsistency. In this undefined state, $\mathcal{E}$ may not terminate. Therefore, $\mathcal{D}$ should terminate $\mathcal{E}$'s attack if it lasts longer than time $\tau(k)$.

At the end of the test query, if $\mathcal{E}$ has made fewer than l H_F queries, $\mathcal{D}$ aborts. If not, $\mathcal{D}$ uses $\mathcal{E}$'s lth H_F query (on some value h) guessing that its value is $e(xjP, s_nzyP) = e(P,P)^{js_nxyz} = e(P,P)^{xyz\gamma}$ where $\gamma = js_n$. $\mathcal{D}$ will output $h^{1/\gamma}$ as its guess for $e(P,P)^{xyz}$. In this case, the probability that $\mathcal{D}$ outputs the correct output is at least $\frac{\eta(k)^{'}}{T_C^2 T_H}$, which is nonnegligible.

Test queries for Protocol III: The test query for Protocol III progresses very similarly to the test query for Protocol I. We will point out the differences. Assuming that $\Pi^n_{I,J}$ received $jP + jQ_j$ and calculated $s_nzP+s_nr_izP$ before accepting, the oracle will hold the session key $F^*(e(s_nzP+s_nzyP+jP+jQ_j, xP+xr_iP))$. $\mathcal{D}$ uses $\mathcal{E}$'s lth H_F query (on some value h) guessing that its value is $e(s_nzP+s_nzyP+jP+jQ_j, xP+xr_iP) = e(P,P)^{xyz(s_n+s_nr_i)+(xyj+s_nxz+xj)(1+r_i)} = \delta e(P,P)^{xyz\gamma}$ where $\delta = e(P,P)^{(xyj+s_nxz+xj)(1+r_i)}$ and $\gamma = s_n + s_nr_i$. $\mathcal{D}$ will output $(h/\delta)^{1/\gamma}$ as its guess for $e(P,P)^{xyz}$. The rest is the same as for the test query for Protocol I explained above.

As a result, at the end of a test query if adversary $\mathcal{E}$ is able to guess the value of b correctly with a nonnegligible advantage, $\mathcal{D}$ can estimate $e(P,P)^{xyz}$ with nonnegligible probability. However, this situation contradicts the DBDH assumption.

We will not give a formal proof for Protocol II. We note that the formal proof would be similar to the proof given for the other protocols and based on the ECDH problem, [34] which is hard.

4.4. Proof explanation of protocols

The formal proof based on the BJM security model does not allow the adversary to ask reveal queries because the adversary could alter the messages exchanged so that the communicating parties agree on a session key without achieving a matching conversation. The adversary can ask a test query to one of the parties and ask a reveal query to the other party. This is valid because the parties are uncorrupted, the test query is asked to an oracle that is not revealed, and the reveal query is asked to an oracle that does not have a matching conversation with the tested oracle. The work of Chen and Kudla includes a useful example for the case explained above [20].

As explained in the work of Chen and Kudla, the known key security and the unknown key share resilience properties implied by the definitions of the AK and AKC protocols. On the other hand, the BJM model does not allow the adversary to ask test queries to corrupted oracles. Therefore, the security model cannot model the key compromise impersonation attack and the forward secrecy properties. The attacks not covered by the security model are discussed based on heuristic arguments in Section 4.2.

To extend the proof to an AKC protocol we would follow a similar approach to the proof of Theorem 9 in the work of Blake-Wilson et al. [30]. However, we prefer to leave the proof for the interested reader.

5. Performance and security evaluation

In this section, we evaluate the performance and security features of the ID-based protocols that are proposed in this study and compare them to some known ID-based protocols found in the literature. According to the primitive function timings given in Table 4, Table 5 demonstrates the computational costs and the related performance numbers for each protocol. Table 6 evaluates each protocol against a number of security threats and lists the security features.

5.1. Performance evaluation

We will evaluate the performance of the proposed ID-based protocols and compare them to the ID-based SIP authentication schemes given in the literature with respect to the arithmetic and cryptographic operations. We used the PBC Library (version 0.5.12) to obtain the primitive function timings given in Table 4. The PBC Library is built on the GMP Library (version 5.0.5). Table 4 shows the arithmetic mean and the standard deviation of the following primitive operations for 1000 executions of each operation. The timings were obtained on a personal computer that had an Intel Pentium Dual CPU E2200 2.20GHz processor, 2048 MB of RAM, and the Ubuntu 12.04.1 LTS 32-bit operating system.

The symbols used for the primitive functions are given in the first column of Table 4. A short explanation for each symbol, their arithmetic means, and their standard deviations are given respectively in columns 2, 3, and 4. The ID-based signature and ID-based verification operations are denoted by (ID-SIGN) and (ID-SVER). These timings were obtained by implementing the signature scheme of Hess using the PBC Library. Due to the fact that only arithmetic operations are involved, the standard deviations are quite low.

For ID-based operations, we used the Type A curves defined within the PBC Library because they are fast and efficient. In the PBC Library, the Type A curve is chosen as $E(F_q) : y^2 = x^3 + x$, where q is some prime. The group G_1 is a subgroup of $E(F_q)$, while G_2 is a subgroup of F_q^2. The group order of G_1 is 160 bits, and the order of the base field is 512 bits. The embedding degree k is 2. Since the bilinear function is $e : G_1 \times G_1 \rightarrow G_2$, this pairing is symmetric.

A detailed performance evaluation of the proposed protocols and protocols given in the literature is

Table 4. PBC Library-based primitive timings.

Symbol	Operation	Arithmetic mean (ms)	Standard deviation (ms)
RNG	Select random number Z_r	0.539	0.0000106
H	String to number (hash) Z_r	0.0023	0.0000006
H_1	String to point (hash) G_1	12.418	0.0000442
H_2	String to point (hash) G_2	0.947	0.0000260
PM	Point multiplication G_2	2.226	0.0000733
PA	Point addition G_1	0.0288	0.0000025
PAIRING	Pairing $G_1 \times G_1 \rightarrow G_2$	5.811	0.0002854
ID-SIGN	ID-based signature (Hess)	23.8662	0.0003236
ID-SVER	ID-based verification (Hess)	5.87147	0.0001007
EXPO	Modular exponentiation (1024 bit)	3.8500	0.0000464

demonstrated in Table 5. The arithmetic and cryptographic operations performed on both the client side and the server side are given in column four and five respectively. The estimated timings for these operations can be found in the last two columns. We assumed that the signature scheme of Hess is used for the signature and verification operations.

The proposed ID-based schemes of Wang and Zhang [17], Ring et al. [15], and Han et al. [16] rely on ID-based signature schemes that use multiple pairing operations. The server side cost is relatively less in Ring et al.'s protocol, which can increase the performance of the server. In general, the overhead for these 3 protocols is considerably high for SIP authentication where tens of thousands of clients compete for the SIP proxy servers' authentication service. SIP protocol implementations have to meet certain QoS requirements. When these

Table 5. Performance evaluations of the discussed schemes and proposed protocols.

Performance properties	Scheme	Problem	Operations: client side	Operations: server side	Est. cost: client side (ms)	Est. cost: server Side (ms)
Protocol I	ID-based	BDH	3H + 1RNG + 3PM + 1PAIRING	3H + 1RNG + 4PM + 1PAIRING	13.0343	15.2603
Protocol II	ID-based	ECDLP	3H + 1RNG + 4PM + 1PA	3H + 1RNG + 4PM + 1PA	9.4787	9.4787
Protocol III	ID-based	BDH	3H + 1RNG + 2PM + 3PA + 1PAIRING	NA	10.8953	NA
Ring et al. [15]	ID-based	BDH	1H + 2RNG + 2PM + 2PAIRING + 2ID-SIGN + 2ID-SVER	1RNG + 1ID-SIGN + 1ID-SVER	76.6297	30.2766
Han et al. [16]	ID-based	BDH	$2H_1$ + $1H_2$ + 2PM + 1RNG + 2PA + 3PAIRING + 2EXPO	$1H_1$ + 1PM + 4PAIRING + 1EXPO	55.9646	41.738
Wang and Zhang [17]	ID-based	BDH	2H + 1PM + 1RNG + 2PAIRING + 1ID-SIGN + 1ID-SVER	3H + 1PM + 1RNG + 2PAIRING + 1ID-SIGN + 1ID-SVER	44.1293	44.1316
Ni et al. [19]	ECC-based	ECDLP	6H + 6PM + 3PA + 3RNG	6H + 6PM + 3PA + 2RNG	15.0732	14.5342

requirements are not satisfied, the requests are retransmitted, further increasing SIP traffic. Compared to these protocols, Ni et al.'s scheme achieves better performance results, but it cannot be exactly classified as an ID-based protocol. It uses elliptic curve point operations instead of pairing functions.

The first 3 rows in Table 5 exhibit the computational costs of the new protocols proposed in this work. Protocol I is comparable to the protocols proposed by Wang and Zhang [17], Ring et al. [15], and Han et al. [16] because it aims to achieve the client-to-server authentication by using bilinear pairing functions. The overhead is considerably less because the protocol does not rely on a signature scheme and deploys only one pairing function. The second protocol is comparable to Ni et al.'s scheme [19]. It originates from the ID-based cryptosystem but eliminates the pairing function and uses elliptic curve point multiplications. Although it requires immediate disposal of the client's ephemeral key, its overhead is significantly lower than the other protocols listed.

The protocols proposed by Wang and Zhang, Ring et al., Han et al., and Ni et al. can establish session keys between the clients. Similarly, the proposed third protocol is an authenticated key agreement protocol that uses only one bilinear pairing function. The performance achieved is significantly better than the comparable protocols, which is preferable for the SIP protocol.

Table 6. Security attack and feature evaluations of the discussed schemes and proposed protocols.

Security attacks and features	Prot. I	Prot. II	Prot. III	Ring et al. [15]	Han et al. [16]	Wang and Zhang [17]	Ni et al. [19]
Security attacks							
Replay attack	No	No	No	No	No	No	No
Man-in-the-middle attack	No	No	No	No	No	No	No
Known-key attack	No	No	No	No	No	No	No
Key compromise impersonation attack	No	No	No	No	No	No	No
Ephemeral key reveal attack	No	No	No	No	No	No	No
Security features							
Forward secrecy	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Session key is used	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mutual authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Different domain app.	No	No	No	No	No	Yes	Yes
Key escrow feature	Yes	Yes	Yes/No	Yes	Yes	No	No

5.2. Security evaluation

Investigating Table 6, we can state that the evaluated ID-based authentication schemes provide strong security against some well-known attacks such as the man-in-the-middle and reply attacks. The main reasons for this strength are the ephemeral keys and the public keys, which are chosen as unique identifiers.

The session keys of the ID-based schemes evaluated in Table 6 are calculated using the bilinear pairing function. Because the pairing function uses the ephemeral keys in addition to long-term private keys, the compromise of long-term private keys does not provide enough information to calculate the previous session keys. Based on this fact, we can state that ID-based authentication schemes provide forward secrecy.

The evaluated ID-based schemes can resist the known-key attack. An adversary that compromises a session key cannot compromise other session keys because, as described above, the calculation of each session

key includes ephemeral keys, as well. The schemes were also evaluated against the ephemeral key reveal attack. For each scheme it was verified that an adversary obtaining an ephemeral key does not have enough information to calculate the session key.

Eliminating the need for certificates, ID-based cryptography has become an important alternative to public key cryptography. However, ID-based protocols can suffer from the key escrow problem where the PKG can use the parties' secret information to obtain the session keys. Depending on the purpose, the key escrow can be defined as a feature where authorities such as government agencies can conduct surveillance. Among the evaluated protocols, Wang and Zhang's and Ni et al.'s schemes do not have the key escrow feature. These schemes use the certificateless public key cryptography concept to hide the client's private key from the trusted third party and avoid the key escrow problem.

In practical implementations we observed that SIP proxy servers conducted the authentication process accessing the required secret information. Therefore, for the proposed protocols it was assumed that the SIP servers include the functions of the PKG. Among the proposed protocols, Protocol III is a client-to-client protocol where the key escrow feature can be disabled by sending an extra elliptic curve point and changing the session key calculation slightly. Protocols I and II are client-to-server authentication protocols where the key escrow concept is unrelated due to the PKG functionality of the SIP server.

On the other hand, there is an important disadvantage of architectures where SIP servers pose PKG functionalities. This type of architectures cannot be supported within a multidomain structure. As noted in Table 6, only Wang and Zhang's and Ni et al.'s schemes can be applied to a multidomain structure.

In the protocols of Wang and Zhang, Ring et al., and Han et al., the authenticity is established by using ID-based signature schemes. When the signature cannot be verified, the protocol will fail at that point. The proposed protocols use implicit key authentication and key confirmation, which avoids the use of expensive signature schemes. The authentication fails when the key confirmation fails at the end of the protocol. Although the proposed protocols fail one step later, avoiding the signature generation and verification reduces the overhead significantly.

6. Conclusion

ID-based cryptography presents convenient features for authentication and key agreement protocols. In the literature, various ID-based protocols are proposed, but their computational overhead is not very suitable for SIP. In this study, new ID-based protocols are proposed for the SIP authentication and key agreement protocols. These protocols minimize the use of expensive pairing functions but still resist notable attacks.

The security of the proposed protocols were analyzed and demonstrated with security proofs based on the BJM security model. Finally, the performance overhead and security of the proposed protocols were compared to other ID-based SIP authentication and key agreement protocols. Once the computational overhead of calculating the pairing function is reduced, ID-based cryptography will become more attractive.

A. Appendix

Parameters and meanings used in schemes			
C	Client	$(G_1, +)$ and $(G_2, .)$	Cyclic groups of the same prime order l
S	Server	$P \in E(Fq)$	The generator point of the group G_1
A	Alice	H_1	$\{0,1\}^* \rightarrow G_1^*$ where $G_1^* := G_1 \backslash \{O\}$ (map-to-point)
B	Bob	H_2	$\{0,1\}^* \times G_2 \rightarrow (Z/lZ)^*$ (map-to-number)
F()	One-way hash function	$e : G_1 \times G_1 \rightarrow G_2$	Pairing function that has bilinear and nondegenerate properties
K	Session key	S_A	Private key of Alice.
$\oplus$	XOR operation	$Q_A = H_1(ID_A)$	Public key of Alice.
s	Private key of the PKG	S_B	Private key of Bob
P_{pub}	Public key of the PKG	$Q_B = H_1(ID_B)$	Public key of Bob

Ring et al.'s scheme.		
Step	**Message**	**Calculation**
The authentication scheme		
$C \rightarrow S$	*REQUEST (REGISTER or INVITE)*	
$S \rightarrow C$	*CHALLENGE* {*Sign(nonce,realm,opaque,time, uname)*} Also, the server responds "401 Unauthorized" message for REGISTER message or "407 Proxy Authentication Required" message for INVITE message with CHALLENGE message.	*realm string*: An identifier of the security domain *opaque string*: A session identifier *Sign(nonce,realm,opaque,time,uname)*
$C \rightarrow S$	*RESPONSE* {*Sign(nonce,realm,opaque,uname, response)*}	Verify *Sign(nonce,realm,opaque,time, uname)* *Sign(nonce,realm,opaque,uname,response)*
$S \rightarrow C$	200 OK	Verify *Sign(nonce,realm,opaque,uname, response)*
The key agreement scheme		
$A \rightarrow B$	$INVITE$ $\{Sign(T_A, To, From, ...)\}$	$a \in Z_r$ $T_A = aP$ $Sign(T_A, To, From, ...)$
$B \rightarrow A$	$200OK$ $\{Sign(T_B, To, From, ...)\}$	Verify $Sign(T_A, To, From, ...)$ $b \in Z_r$ $T_B = bP$ $Sign(T_B, To, From, ...)$ $K_{BA} = e(S_B, T_A)e(Q_A, bP_{Pub(PKG_A)})$ $K = F(A\|\|B\|\|T_A\|\|T_B\|\|K_{BA})$: The session key
$A \rightarrow B$	ACK	Verify $Sign(T_B, To, From, ...)$ $K_{AB} = e(S_A, T_B)e(Q_B, aP_{Pub(PKG_B)})$ $K = F(A\|\|B\|\|T_A\|\|T_B\|\|K_{AB})$

Han et al.'s scheme.		
Step	**Message**	**Calculation**
$A \rightarrow B$	$REQUEST\{(u, v)\}$	$P_1 \in G_1$ $k \in (Z/lZ)^*$ $r = e(P_1, P)^k$ $t = H_1^*(r)Q_A$ $v = H_2(m, t)$, m can be time-based nonce $u = vS_A + kP_1$
$B \rightarrow A$	$200OK$ (Note: if the values are same, the signature is verified and Alice is authenticated.)	$t = H_1^*(r)Q_A = H_1^*(e(u, P)e(Q_A, -P_{pub})^v)Q_A$ $v \stackrel{?}{=} H_2(m, t)$
A And B	They can calculate the session key simultaneously.	$K_A = e(S_A, Q_B)^{H_1^*(r)} \oplus e(S_A, Q_B)$: Alice's calculation $K_B = e(t, S_B) \oplus e(Q_A, S_B)$: Bob's calculation K $= K_A = K_B$: The session key

Wang and Zhang's scheme.		
Step	**Message**	**Calculation**
$A \rightarrow B$ or $C \rightarrow S$	$REQUEST\ \{ID_A, P_A, s, T_A\}$	$a \in {Z_q}^*$ $T_A = aP$ ID_A: Alice's identity $P_A =< X_A, Y_A >$: Alice's public key s: The session identifier
$B \rightarrow A$ or $S \rightarrow C$	$CHALLENGE\ \{nonce, realm, P_B, T_B, signature_B\}$	$b \in {Z_q}^*$ $T_B = bP$ $nonce = F(realm, time)$ $signature_B = Sign(nonce, realm, ID_A, T_A, T_B, P_B)$: The signature of Bob $P_B =< X_B, Y_B >$: Bob's public key
$A \rightarrow B$ or $C \rightarrow S$	$RESPONSE\ \{nonce, realm, ID_A, signature_A\}$	Verify $signature_B = Sign(nonce, realm, ID_A, T_A, T_B, P_B)$ $signature_A = Sign(nonce, realm, ID_A, T_A, T_B, P_A)$ $K_{AB} = e(S_A, T_B)e(Q_B, aY_B)$ $K = K_A = H(K_{AB})$: The session key
$B \rightarrow A$ or $S \rightarrow C$	$200OK$	Verify $signature_B = Sign(nonce, realm, ID_A, T_A, T_B, P_B)$ $signature_A = Sign(nonce, realm, ID_A, T_A, T_B, P_A)$ $K_{BA} = e(S_B, T_A)e(Q_A, bY_A)$ $K = K_A = K_B = H(K_{BA})$: The session key

References

[1] Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Sparks R, Handley M, Schooler E. SIP: Session Initiation Protocol. Internet Engineering Task Force 2002; RFC 3261.

[2] Geneiatakis D, Dagiouklas A, Kambourakis G, Lambrinoudakis C, Gritzalis S, Ehlert S, Sisalem D. Survey of security vulnerabilities in session initiation protocol. IEEE Communications Surveys and Tutorials 2006; 8: 68–81.

[3] Salsano S, Veltri L, Papalilo D. SIP security issues: the SIP authentication procedure and its processing load. IEEE Network 2002; 16: 38–44.

[4] Dantu R, Fahmyb S, Schulzrinne H, Cangussu J. Issues and challenges in securing VoIP. Computers & Security 2009; 28: 743–753.

[5] Franks J, Hallam-Baker P, Hostetler J, Lawrence S, Leach P, Luotonen A, Stewart L. HTTP authentication: basic and digest access authentication. Internet Engineering Task Force 1999; RFC 2617.

[6] Yang CC, Wang RC, Liu WT. Secure authentication scheme for session initiation protocol. Computers & Security 2005; 24: 381–386.

[7] Durlanik A, Sogukpinar I. SIP authentication scheme using ECDH. Enformatika 2005; 8: 350–353.

[8] Choi J, Jung S, Bae K, Moon H. A lightweight authentication and hop-by-hop security mechanism for sip network. In: Advanced Technologies for Communications;6-9 Oct 2008; Hanoi,Vietnam; pp. 235–238.

[9] Yoon EJ, Yoo KY. A new authentication scheme for session initiation protocol. In: International Conference on Complex, Intelligent and Software Intensive Systems;2009; Fukuoka, Japan.

[10] Geneiatakis D, Lambrinoudakis C. A lightweight protection mechanism against signaling attacks in a sip-based voip environment. Telecommunication Systems 2007; 36: 153–159.

[11] Tsai JL. Efficient nonce-based authentication scheme for session initiation protocol. International Journal of Network Security 2009; 8: 312–316.

[12] Dacosta I, Traynor P. Proxychain: Developing a robust and efficient authentication infrastructure for carrier-scale VoIP networks. In: Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference; 2010; pp. 10–10.

[13] Srinivasan R, Vaidehi V, Harish K, LakshmiNarasimhan K, LokeshwerBabu S, Srikanth V. Authentication of signaling in VoIP applications. In: 11th Asia Pacific Conference on Communication(APCC);2005; Perth, Australia.

[14] Nodooshan AM, Darmani Y, Jalili R, Nourani M. A robust and efficient SIP authentication scheme. Communications in Computer and Information Science 2009; 6: 551–558.

[15] Ring J, Choo KKR, Foo E, Looi M. A new authentication mechanism and key agreement protocol for SIP using identity-based cryptography. In: AusCERT Asia Pacific Information Technology Security Conference; 23 May 2006; Gold Coast, Australia.

[16] Han K, Yeun C, Kim K. Design of secure VoIP using Id-based cryptosystem. In: The Symposium on Cryptography and Information Security (SCIS2008); 22-25 January 2008; Miyazaki,Japan.

[17] Wang F, Zhang Y. A new provably secure authentication and key agreement mechanism for SIP using certificateless public-key cryptography. Computer Communication 2008; pp. 2142–2149.

[18] Kilinc HH, Allaberdiyev Y, Yanik T. Performance evaluation of Id based authentication methods in the SIP protocol. In: Application of Information And Communication Technologies 3rd IEEE International Conference (AICT 2009); 2009.

[19] Ni L, Chen G, Li J. A pairing-free identity-based authenticated key agreement mechanism for sip. In: Proceedings of the 2011 International Conference on Network Computing and Information Security - Volume 01(NCIS 2011); 2011; Washington, DC, USA; pp. 209–217.

[20] Chen L, Kudla C. Identity based authenticated key agreement protocols from pairings. CSFW; 2003; pp. 219–233.

[21] Hess F. Efficient identity based signature schemes based on pairings. In: Proceedings of the 9thWorkshop on Selected Areas in Cryptography; 2003; pp. 310–324.

[22] Okamoto T, Tso R, Okamoto E. One-way and two-party authenticated id-based key agreement protocols using pairing. Modeling Decisions for Artificial Intelligence 2005; 3558/2005: 122–133.

[23] Patil HK, Chen TM, Willis D, Nguyen N. Authentication in SIP using identity based signature scheme. In: NIST workshop on Applications of Pairing-Based Cryptography: Identity-Based Encryption and Beyond; 3-4 June 2008; Gaithersburg, Maryland, USA.

[24] Al-Riyami S, Paterson K. Certificateless public key cryptography. Advances in Cryptology-Asiacrypt 2003; 2894: 452–473.

[25] Choon JC, Cheon JH. An identity-based signature from Gap Diffie-Hellman Groups. In: 6th International Workshop on Practice and Theory in Public Key Cryptography (PKC2003); January 2003; Miami, FL, USA; pp. 18–30.

[26] Lynn B. Pairing-based cryptography library, available at http://crypto.stanford.edu/pbc/, accessed: 14 February 2013.

[27] OpenSIPs, available at http://www.opensips.org/, accessed: 14 February 2013.

[28] Shamir A. Identity-based cryptosystems and signature schemes. In: Advances in Cryptology: Proceedings of CRYPTO 84; 1984; pp. 47–53.

[29] Boneh D, Franklin MK. Identity-based encryption from the Weil Pairing. In: Advances in Cryptology: Proceedings of CRYPTO 01; 2001; pp. 213–229.

[30] Blake-Wilson S, Johnson D, Menezes A. Key agreement protocols and their security analysis. In: Proceedings of the 6th IMA International Conference on Cryptography and Coding; 1997; London, UK; pp. 30–45.

[31] Bellare M, Rogaway P. Entity authentication and key distribution. In: Proceedings of the 13th annual international cryptology conference on Advances in cryptology (CRYPTO'93); 1994; New York, NY, USA; pp. 232–249.

[32] Strangio MA. On the resilience of key agreement protocols to key compromise impersonation. Cryptology ePrint Archive, Report 2006/252, 2006, available at http://eprint.iacr.org/2006/252, accessed: 14 February 2013.

[33] Cheon JH, Lee DH. Diffie-hellman problems and bilinear maps. Cryptology ePrint Archive, Report 2002/117, 2002, available at http://eprint.iacr.org/2002/117, accessed: 14 February 2013.

[34] Hankerson D, Menezes AJ, Vanstone S. Guide to Elliptic Curve Cryptography. New York, USA: Springer-Verlag, 2004.