Efficient Deniably Authenticated Encryption and Its Application to E-Mail

Fagen Li, *Member, IEEE,* Di Zhong, and Tsuyoshi Takagi

***Abstract*— Confidentiality and authentication are two main security goals in secure electronic mail (e-mail). Pretty good privacy (PGP) and secure/multipurpose internet mail extensions (S/MIME) are two famous secure e-mail solutions. Both PGP and S/MIME use digital envelope to provide message confidentiality and digital signature to provide message authentication. However, these methods have the following two weaknesses: 1) digital signature provides non-repudiation evidence of sender that is not desired in some e-mail applications and 2) efficiency is low, since these methods use two kinds of public key cryptographic primitives: public key encryption and digital signature. To overcome the above two weaknesses, we introduce a new concept called deniably authenticated encryption that can achieve confidentiality, integrity, and deniable authentication in a logical single step. We first propose a deniably authenticated encryption scheme and prove its security in the random oracle model. Then, we design a secure e-mail protocol using the proposed deniably authenticated encryption scheme. The deniable authentication property protects senders' privacy.**

***Index Terms*— E-mail, confidentiality, deniable authentication, deniably authenticated encryption.**

I. INTRODUCTION

ELECTRONIC mail (e-mail) has been widely used in modern information society. People send and read e-mails from their personal computers, business workstation and even mobile telephones. While e-mails provide a great convenience for exchanging information, it also brings a lot of research challenges. One of the important issues is the security due to the vulnerability of underlying network. A secure e-mail system should provide the following two security properties.

- Confidentiality: Only the intended receiver can read the transmitted message.
- Authentication: The intended receiver can identify the source of a given message.

We can apply cryptographic techniques to achieve the above two security goals. Concretely, we can use encryption to achieve the confidentiality and digital signature to achieve authentication. Pretty Good Privacy (PGP) [1] and Secure/Multipurpose Internet Mail Extensions (S/MIME) [2] are two famous secure e-mail solutions. In PGP and S/MIME, each user has two public key/private key pairs. One pair is used for message encryption and the other pair is used for digital signature. Both PGP and S/MIME use digital envelopes to provide message confidentiality. First, the sender chooses a session key randomly and encrypts the actual message by using a symmetric cipher with the session key. Then, the sender encrypts the session key by using a public key encryption scheme with the receiver's public key. After receiving the encrypted message and the encrypted session key, the receiver first decrypts the session key with its private key. Then, the receiver decrypts the actual message with the session key. To provide authentication, both PGP and S/MIME use digital signature techniques. The sender signs the message digest by using a signature scheme with its private key. The resulting signature is attached along with the encrypted message. The receiver verifies the validity of the signature with the sender's public key. Since digital signatures provide non-repudiation evidence of the sender, the receiver can prove the source to any third party. This case may violate the privacy of the sender.

To solve the above problem, Harn and Ren [3] proposed a new design to provide deniable authentication in e-mail systems (denoted by HR scheme). In the HR scheme, a sender signs the ciphertext of a session key directly instead of signing the message digest, which makes the signature forgeable to achieve deniability for the authentication. By this new design, the intended receiver can identify the source of a given message, but it cannot prove the source to any third party. That is, the sender can deny its actions. Therefore, deniable authentication is achieved. However, Ki *et al.* [4] showed that the HR scheme is not fully deniable. The transcripts generated by the sender are reasonably distinguishable from those generated by a receiver when the public key encryption scheme is secure against chosen ciphertext attack (CCA). Ki et al. also constructed a privacy-enhanced deniable authentication scheme using the designated verifier signature scheme (denoted by KHNLL scheme). In 2011, Harn *et al.* [5] proposed a fully deniable message authentication protocols preserving confidentiality (denoted by HLLC scheme). However, the HLLC scheme can not be used in the e-mail systems since this scheme is interactive. Another weakness in [3]–[5] is lack of formal security proof that is very important for cryptographic design. In addition, Hwang and Sung [6] proposed a deniable authentication scheme with confidentiality property using promised signcryption (denoted by HS scheme). However, for the confidentiality, the HS scheme is only proved to be indistinguishable against chosen plaintext attack. The chosen

Manuscript received November 21, 2015; revised April 4, 2016 and June 12, 2016; accepted June 18, 2016. Date of publication June 24, 2016; date of current version August 26, 2016. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2013J069 and in part by the National Natural Science Foundation of China under Grant 61073176, Grant 61272525, Grant 61302161, and Grant 61462048. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sen-Ching Samson Cheung.

F. Li and D. Zhong are with the Center for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: fagenli@uestc.edu.cn; 71461758@qq.com).

T. Takagi is with the Institute of Mathematics for Industry, Kyushu University, Fukuoka 819-0395, Japan (e-mail: takagi@imi.kyushu-u.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2016.2585086

plaintext attack is a weak model for cryptanalysis that does not allow the adversary to ask the unsigncryption oracle. In 2013, Hwang *et al.* [7] proposed a deniable authentication scheme that achieves not only confidentiality but also anonymous fair protections (denoted by HSC scheme). However, the HSC scheme is also indistinguishable against chosen plaintext attack since the adversary is not allowed to ask the unsigncryption oracle.

A. Contribution

In this paper, we propose a new concept called deniably authenticated encryption (DAE) that can achieve confidentiality and deniable authentication in a logical single step. We propose a DAE scheme and prove its security in the random oracle model. As compared with existing solutions [3]–[7], our scheme has the following advantages: (1) confidentiality and deniable authentication are achieved simultaneously in a logical single step; (2) the scheme is clearly proved to be indistinguishable against adaptive chosen ciphertext attack; (3) the scheme is very efficient in computational cost and ciphertext size. We also design a secure e-mail protocol using the proposed DAE scheme.

B. Related Work

There are three related works called deniable authentication (DA), authenticated encryption (AE) and secure e-mail protocols.

Authentication assures that the communicating party is the one that it claims to be. Authentication is an important security requirement for many applications, such as smart grid [8], vehicular ad hoc networks [9] and cloud computing [10]. In some cases, we need to design an authentication scheme with some special properties, such as broadcast authentication [11], [12], biometric authentication [13], anonymous authentication [14], group authentication [15], conditional privacy authentication [16] and DA [17]. In this paper, we focus on the DA. Compared with the traditional authentication, the DA has two special properties: (1) an intended receiver can identify the source of a given message; (2) the intended receiver cannot show the source of this given message to any third party. This kind of protocol can be used in some specialized applications. For example, it can supply freedom from coercion in the electronic voting and secure negotiation over the Internet [18]. In 1998, Dwork *et al.* [19] designed a notable DA protocol using on concurrent zero-knowledge proof. However, this protocol needs a timing constraint and this proof is subject to a time delay in this authentication process. Aumann and Rabin [18], [20] designed another DA protocol based on the factoring problem, but it requires a pubic directory trusted by both the sender and receiver. In 2001, Deng *et al.* [21] designed two DA protocols based on the factoring problem and discrete logarithm problem, respectively. Unfortunately, they also need a trusted public directory. To overcome this weakness, Fan *et al.* [22] designed a new DA protocol based on Diffie-Hellman key agreement protocol [23]. Their protocol uses public key certificates to defeat the person-in-middle (PIM) attack and digital signatures to identify the source of a given message. However, Yoon *et al.* [24] showed that [22] suffers from an authentication weakness. In this weakness, an attacker can easily disguise as a receiver to a sender and an inquisitor can easily identify the source of the message. In 2004, Shao [25] designed a non-interactive DA protocol based on the generalized ElGamal signature [26]. If an adversary can generate a forgery in Shao's protocol, we can construct an algorithm that can forge a generalized ElGamal signature. In 2005, Lu and Cao [27] designed a non-interactive DA protocol based on the factoring problem. They also gave a further work to design a formal security model for DA and to prove the proposed protocol in the formal security model. In 2005, Wang *et al.* [28] designed a DA protocol based on the ElGamal cryptosystem [29]. Their protocol makes use of the inverse of the ElGamal to obtain deniability for the authentication. In 2006, Shao *et al.* [30] showed that [28] is not secure under the PIM attack because the ElGamal cryptosystem is malleable. In addition, Shao, Cao and Lu used a hash function to fix this problem. In 2010, Yoon *et al.* [31] showed that both [28] and [30] are not secure under malicious receiver impersonation attack because a sender can not verify the identity of a receiver. In addition, Yoon et al. designed an improved DA protocol based on the ElGamal cryptosystem [29]. They claimed that their protocol satisfies deniable authentication, mutual authentication and confidentiality. However, Li and Takagi [32] showed that [31] does not satisfy the deniable authentication property. The receiver can show the source of a given message to a third party. Lee *et al.* [33] designed a novel DA protocol using generalized ElGamal signature [26]. The characteristic of [33] is that they can replace the underlying signature with a generalized ElGamal-like signature. Wang and Song [34] designed a non-interactive DA protocol using designated verifier proof method. They borrowed the traditional authentication model to prove the security of the proposed protocol. Li *et al.* [35] designed an identity-based DA protocol by combining identity-based cryptography and DA. In [35], a public key is derived from a user's identity information. In addition, public key certificates are not required. From the above analysis, we know that some protocols were broken by later research. An important reason is that these protocols lack formal security proofs.

AE schemes provides two security goals: confidentiality and authenticity. AE can be divided into two types: symmetric AE [36] and asymmetric (public key) AE [37]. In the symmetric-key model, we can use a keyed hash (i.e, a message authentication code (MAC)) with some suitable key K_1 along with a secure encryption scheme with an independent key K_2 to achieve authenticated encryption. But the sender and the receiver need to agree on K_1 and K_2 in advance. In the asymmetric-key model, AE integrates digital signature and public key encryption in a single procedure to reduce the computation and communication cost. The symmetric AE is deniable but the asymmetric AE is not deniable. Many efficient AE schemes have been designed [38], [39]. The main goal of this paper is to find an efficient method which makes public key AE deniable. We give a positive answer to the question.

Although PGP [1] and S/MIME [2] are two good solutions to secure e-mail, they can not achieve perfect forward secrecy.

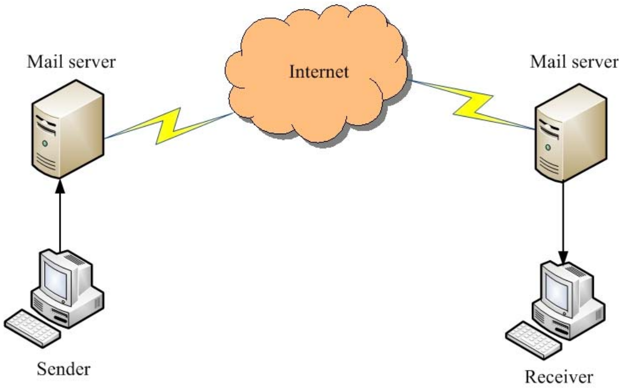


Fig. 1. An e-mail system model.

To solve this problem, Sun *et al.* [40] proposed two secure e-mail protocols. Their protocols used Diffie-Hellman key agreement protocol and certificate of encrypted message being a signature (CEMBS) to provide the forward secrecy. Dent [41] showed that [40] can not provide the forward secrecy since all previous session keys will be disclosed if the receiver's long-term key is exposed. In order to overcome this weakness, Kim *et al.* [42] proposed two e-mail protocols providing forward secrecy in which a short-term key is established between a receiver and an e-mail server using the Diffie-Hellman key agreement protocol [23]. However, Yoon and Yoo [43] showed that the second protocol of [42] is vulnerable to impersonation attack. Aiming to the above analysis, Chang *et al.* [44] proposed a new secure e-mail protocol that provides forward secrecy and is more suitable for mobile devices. In 2013, Zhang and Takagi [45] constructed an anonymous multireceiver encryption scheme and used the scheme to deploy a group e-mail systems. Chen [46] gave a multicast key protocol for e-mail systems that can provide forward secrecy.

C. Organization

The rest of this paper is organized as follows. We introduce the system model, security requirements and complexity assumptions in Section II. An efficient DAE scheme is proposed in Section III. We design a secure e-mail protocol in Section IV. Finally, the conclusions are given in Section V.

II. Preliminaries

In this section, we give the e-mail system model, security requirements and some complexity assumptions that our scheme is based on.

A. System Model

Fig. 1 shows the overview of an e-mail system model. The model consists of a sender, a receiver and mail servers. The sender sends an e-mail by its mail server using simple mail transfer protocol (SMTP). The receiver gets the mail by its mail server using either the post office protocol (POP3) or the Internet message access protocol (IMAP).

B. Security Requirements

A secure e-mail system should satisfy confidentiality, integrity, and deniable authentication. Confidentiality keeps the e-mail content secret from the others except the sender and receiver. Integrity ensures that the e-mail content from the sender has not been altered by unauthorized entities. Deniable authentication enables the receiver to identify the source of a given e-mail and cannot prove the source of the given e-mail to any third party. Deniable authentication protects the privacy of the sender.

C. Complexity Assumptions

Given a group G of prime order q and a generator g of G, the discrete logarithm (DL) problem in G is to find an integer $a \in \mathbb{Z}_q^*$ given y such that $y = g^a \bmod q$.

Definition 1: The (ϵ_{dl}, t)-DL assumption holds if no t-polynomial time adversary $\mathcal{A}$ has advantage at least ϵ_{dl} in solving the DL problem.

Given a group G of prime order q and a generator g of G, the computational Diffie-Hellman (CDH) problem in G is to compute g^{ab} given (g, g^a, g^b) for some unknown $a, b \in \mathbb{Z}_q^*$.

Definition 2: The (ϵ_{cdh}, t)-CDH assumption holds if no t-polynomial time adversary $\mathcal{A}$ has advantage at least ϵ_{cdh} in solving the CDH problem.

Given a group G of prime order q and a generator g of G, the decisional Diffie-Hellman (DDH) problem in G is to decide whether $c = ab \bmod q$ or not given (g, g^a, g^b, g^c) for unknown $a, b, c \in \mathbb{Z}_q^*$. Tuples of the form (g, g^a, g^b, g^{ab}) are called "Diffie-Hellman tuples". There is an important problem called gap Diffie-Hellman (GDH) problem. The GDH problem is to solve a given instance (g, g^a, g^b) of the CDH problem with the help of a DDH oracle that is able to decide whether $c = ab \bmod q$ or not given (g, g^a, g^b, g^c). If (g, g^a, g^b, g^c) is a Diffie-Hellman tuple, we denote it by $\mathrm{DDH}(g, g^a, g^b, g^c) = \top$. Otherwise, we denote it by $\mathrm{DDH}(g, g^a, g^b, g^c) = \perp$.

Definition 3: The $(\epsilon_{gdh}, t, q_{ddh})$-GDH assumption holds if no t-polynomial time adversary $\mathcal{A}$ has advantage at least ϵ_{gdh} in solving the GDH problem after at most q_{ddh} DDH oracle queries.

III. An Efficient DAE Scheme

In this section, we first give the formal definition and security notions for DAE schemes. Then we propose an efficient DAE scheme and analyze its security and performance.

A. Syntax

A generic DAE scheme consists of the following four algorithms.

Setup: This is a probabilistic algorithm that takes as input a security parameter λ to output the system parameters *param*.

KeyGen: This is a key generation algorithm that takes as input the *param* and outputs a public/private key pair (pk_s, sk_s) for a sender and a public/private key pair (pk_r, sk_r) for a receiver.

DA-Encrypt: This is a probabilistic deniably authenticated encryption algorithm run by a sender that takes as input the *param*, a message m, a sender's private key sk_s, a sender's public key pk_s and a receiver's public key pk_r, and outputs a ciphertext σ.

DA-Decrypt: This is a deterministic deniably authenticated decryption algorithm run by the receiver that takes as input the *param*, a ciphertext σ, a sender's public key pk_s, a receiver's private key sk_r and a receiver's public key pk_r, and outputs the plaintext m or an error symbol $\perp$ if σ is an invalid ciphertext between the sender and the receiver.

For consistency, we require that if

$$\sigma = \mathit{DA\text{-}Encrypt}(param, m, sk_s, pk_s, pk_r),$$

then we have

$$m = \mathit{DA\text{-}Decrypt}(param, \sigma, pk_s, sk_r, pk_r).$$

We will omit *param* in the algorithms *DA-Encrypt* and *DA-Decrypt* for simplicity in the following content.

B. Security Notions

A DAE scheme should satisfy confidentiality and deniable authentication.

The standard accepted security notion for the confidentiality is indistinguishability against adaptive chosen ciphertext attack (IND-CCA) [47]. We apply this notion to the DAE schemes. We consider the following game played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Initial: $\mathcal{C}$ runs the *Setup* algorithm to get the system parameters *param* and the *KeyGen* algorithm to get a sender's public/private key pair (pk_s, sk_s) and a receiver's public/private key pair (pk_r, sk_r). $\mathcal{C}$ sends *param*, pk_s and pk_r to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ can perform a polynomially bounded number of deniably authenticated encryption queries and deniably authenticated decryption queries in an adaptive manner. In a deniably authenticated encryption query, $\mathcal{A}$ submits a message m to $\mathcal{C}$. $\mathcal{C}$ runs the deniably authenticated encryption oracle which returns the ciphertext $\sigma = \mathit{DA\text{-}Encrypt}(m, sk_s, pk_s, pk_r)$. Then $\mathcal{C}$ sends σ to $\mathcal{A}$. In a deniably authenticated decryption query, $\mathcal{A}$ submits a ciphertext σ to $\mathcal{C}$. $\mathcal{C}$ runs the deniably authenticated decryption oracle and returns the message $m = \mathit{DA\text{-}Decrypt}(\sigma, pk_s, sk_r, pk_r)$ if it is a valid ciphertext. Otherwise $\mathcal{C}$ returns a rejection symbol $\perp$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ decides when Phase 1 ends. $\mathcal{A}$ chooses two equal length plaintexts m_0 and m_1 and sends these to $\mathcal{C}$. $\mathcal{C}$ takes a random bit β from $\{0, 1\}$ and runs the deniably authenticated encryption oracle which returns a ciphertext $\sigma^* = \mathit{DA\text{-}Encrypt}(m_\beta, sk_s, pk_s, pk_r)$. $\mathcal{C}$ sends σ^* to $\mathcal{A}$ as a challenged ciphertext.

Phase 2: $\mathcal{A}$ can ask a polynomially bounded number of deniably authenticated encryption queries and deniably authenticated decryption queries adaptively again as in Phase 1 with the restriction that it cannot make a deniably authenticated decryption query on the challenged ciphertext σ^*.

Guess: $\mathcal{A}$ produces a bit β' and wins the game if $\beta' = \beta$.

The advantage of $\mathcal{A}$ is defined as

$$Adv(\mathcal{A}) := 2\Pr[\beta' = \beta] - 1,$$

where $\Pr[\beta' = \beta]$ denotes the probability that $\beta' = \beta$.

Definition 4: A DAE scheme is $(\epsilon_{dae}, t, q_e, q_d)$-IND-CCA secure if no probabilistic t-polynomial time adversary $\mathcal{A}$ has advantage at least ϵ_{dae} after at most q_e deniably authenticated encryption queries and q_d deniably authenticated decryption queries in the IND-CCA game.

There is another security notion for the confidentiality is indistinguishability against chosen plaintext attack (IND-CPA). The IND-CPA is similar to the IND-CCA except that $\mathcal{A}$ is not allowed to ask decryption queries in the whole game. Therefore, the IND-CCA represents a stronger security model since the power of the adversary in the IND-CCA is stronger than in the IND-CPA. The IND-CCA security is very important for a public key encryption scheme because it can defend against an active adversary who may modify a transmitted message. However, the IND-CPA can not defend against the active adversary. In addition, the IND-CCA security allows a public key encryption scheme to be securely plugged in a higher level protocol that may be run in arbitrary environments.

Deniable authentication in DAE schemes is different to unforgeability in digital signature schemes. In a digital signature scheme, only the signer can produce a valid signature. That is, no one except the signer can produce a valid signature for a message. The standard accepted security notion for digital signature is existential unforgeability against adaptive chosen messages attack (EUF-CMA) [48]. However, in DAE schemes, we require that only the sender and the receiver can produce a valid ciphertext. Here we modify the EUF-CMA security notion to adapt the requirement for DAE schemes and we call it deniable authentication against adaptive chosen messages attack (DA-CMA). We consider the following game played between a challenger $\mathcal{C}$ and an adversary $\mathcal{F}$.

Initial: $\mathcal{C}$ runs the *Setup* algorithm to get the system parameters *param* and the *KeyGen* algorithm to get a sender's public/private key pair (pk_s, sk_s) and a receiver's public/private key pair (pk_r, sk_r). $\mathcal{C}$ sends *param*, pk_s and pk_r to $\mathcal{F}$.

Attack: $\mathcal{F}$ can perform a polynomially bounded number of queries just like in the IND-CCA game.

Forgery: At the end of the game, $\mathcal{F}$ produces a ciphertext σ' and wins the game if the following conditions hold:

1) $\mathit{DA\text{-}Decrypt}(\sigma', pk_s, sk_r) = m'$. Here m' is an output of *DA-Decrypt*.
2) $\mathcal{F}$ has not made a deniably authenticated encryption query on message m'.

The advantage of $\mathcal{F}$ is defined as the probability that it wins.

Definition 5: A DAE scheme is $(\epsilon_{dae}, t, q_e, q_d)$-DA-CMA secure if no probabilistic t-polynomial time adversary $\mathcal{F}$ has advantage at least ϵ_{dae} after at most q_e deniably authenticated encryption queries and q_d deniably authenticated decryption queries in the DA-CMA game.

Notice that the adversary is not allowed to learn the receiver's private key sk_r in the above definition. This requirement is necessary to obtain the deniability property. The sender can deny its action because the receiver also can produce a valid ciphertext. This is the main difference between deniable authentication and digital signature.

C. Our Scheme

Our scheme consists of the following four algorithms.

Setup: Let λ be a security parameter. Let p be a large prime such that $|p| = \lambda$, q be a large prime factor of $p-1$ and g be a generator with order q in $\mathbb{Z}_p$ such that $q > 2^{l_q(\lambda)}$. Here $l_q : \mathbb{N} \to \mathbb{N}$ is a function deciding the length of q. $H_1 : \{0,1\}^* \to \{0,1\}^n$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_q^*$ are two hash functions. Here n is the length of a message. The system parameters *param* is $\{n, p, q, g, H_1, H_2\}$.

KeyGen: A sender chooses a random number $x_s \in \mathbb{Z}_q^*$ as its private key and sets its public key $y_s = g^{x_s} \bmod p$. Similarly, a receiver chooses a random number $x_r \in \mathbb{Z}_q^*$ as its private key and sets its public key $y_r = g^{x_r} \bmod p$.

DA-Encrypt: Given a message m, a sender's private key x_s, a sender's public key y_s and a receiver's public key y_r, this algorithm works as follows.

1) Choose x from $\mathbb{Z}_q^*$ randomly.
2) Compute $w = y_r^x \bmod p$ and $k = H_1(w)$.
3) Compute $c = m \oplus k$.
4) Compute $e = H_2(m \parallel y_s \parallel y_r \parallel w)$. Here $\parallel$ represents the message concatenation.
5) Compute $v = ex_s + x \bmod q$.
6) Compute $z = g^v \bmod p$ and $s = y_r^v \bmod p$.

The ciphertext is $\sigma = (c, e, z, s)$.

DA-Decrypt: Given a ciphertext σ, a sender's public key y_s, a receiver's private key x_r and a receiver's public key y_r, this algorithm works as follows.

1) Compute $w = (z/y_s^e)^{x_r} \bmod p$.
2) Compute $k = H_1(w)$.
3) Recover $m = c \oplus k$.
4) Accept the message if and only if $e = H_2(m \parallel y_s \parallel y_r \parallel w)$ and $z^{x_r} = s \bmod p$, return $\perp$ otherwise.

We can replace bitwise exclusive OR with a symmetric cipher (E, D) (such as AES [49]) with a key of length n. That is, $c = m \oplus k$ is changed into $c = E_k(m)$ and $m = c \oplus k$ is changed into $m = D_k(c)$. The symmetric cipher scheme only needs to satisfy the very weak requirement to be semantically secure against passive attack.

Both the HS scheme [6] and the HSC scheme [7] are only proved to satisfy the IND-CPA security since they can not overcome the difficulty to construct the decryption oracle in the security proof. This difficulty stems from their construction method. However, our scheme has the following congruence

$$w = y_r^x \bmod p = (z/y_s^e)^{x_r} \bmod p$$

This congruence implies that $z/y_s^e = g^x \bmod p$ since $y_r = g^{x_r} \bmod p$. If we set $\tau = z/y_s^e \bmod p$, we find that (g, τ, y_r, w) is a Diffie-Hellman tuple (here $\tau = g^x$, $y_r = g^{x_r}$ and $w = g^{xx_r}$). In addition, (g, z, y_r, s) is also a Diffie-Hellman tuple (here $z = g^v$, $y_r = g^{x_r}$ and $s = g^{vx_r}$). Therefore, we can use the DDH oracle to construct the decryption oracle in the security proof. So our scheme overcomes the difficulty to construct the decryption oracle and achieves the IND-CCA security.

D. Consistency and Security

We discuss the consistency, deniability, security of the proposed DAE scheme.

1) Consistency: The consistency can be easily verified by the following equations.

$$\begin{aligned} w &= \left(\frac{z}{y_s^e}\right)^{x_r} \bmod p = \left(\frac{g^v}{y_s^e}\right)^{x_r} \bmod p \\ &= \left(\frac{g^{ex_s} g^x}{y_s^e}\right)^{x_r} \bmod p = \left(g^x\right)^{x_r} \bmod p = y_r^x \bmod p \end{aligned}$$

2) Deniability: The receiver with private key x_r may produce a ciphertext which is indistinguishable from that produced by the sender with private key x_s. To simulate the transcripts on a given message m, the receiver does the steps below.

1) Choose $\bar{x}$ from $\mathbb{Z}_q^*$ randomly.
2) Compute $\bar{w} = y_r^{\bar{x}} \bmod p$ and $\bar{k} = H_1(\bar{w})$.
3) Compute $\bar{c} = m \oplus \bar{k}$.
4) Compute $\bar{e} = H_2(m \parallel y_s \parallel y_r \parallel \bar{w})$.
5) Compute $\bar{z} = y_s^{\bar{e}} g^{\bar{x}} \bmod p$ and $\bar{s} = \bar{z}^{x_r} \bmod p$.

$\bar{\sigma} = (\bar{c}, \bar{e}, \bar{z}, \bar{s})$ produced by the receiver is indistinguishable from $\sigma = (c, e, z, s)$ that is produced by the sender according to the *DA-Encrypt* algorithm. Let $\hat{\sigma} = (\hat{c}, \hat{e}, \hat{z}, \hat{s})$ be a ciphertext that is randomly selected in the set of all valid sender's ciphertext intended to receiver. The probability $\Pr[(\bar{c}, \bar{e}, \bar{z}, \bar{s}) = (\hat{c}, \hat{e}, \hat{z}, \hat{s})]$ is $1/(q-1)$ because $(\bar{c}, \bar{e}, \bar{z}, \bar{s})$ is produced from a randomly chosen value $\bar{x} \in \mathbb{Z}_q^*$. Likewise, the probability $\Pr[(c, e, z, s) = (\hat{c}, \hat{e}, \hat{z}, \hat{s})]$ is also $1/(q-1)$ because it is produced from $x \in \mathbb{Z}_q^*$. That is, both distributions of probability are the same.

3) Security: We prove that our scheme satisfies confidentiality and deniable authentication by Theorems 1 and 2.

Theorem 1: In the random oracle model, we assume we have an IND-CCA adversary called $\mathcal{A}$ that is able to distinguish ciphertexts during the IND-CCA game with an advantage ϵ_{dae} when running in a time t and asking at most q_{h_1} H_1 queries, q_{h_2} H_2 queries, q_e deniably authenticated encryption queries and q_d deniably authenticated decryption queries. Then, there exists an algorithm $\mathcal{C}$ that can solve the GDH problem in a time t' and q_{ddh} DDH queries with an advantage

$$\epsilon_{gdh} \geq \epsilon_{dae} - \frac{q_e(q_{h_1} + q_{h_2}) + q_d}{2^{l_q(\lambda)}},$$

where $t' = O(t + t_{h_1} + t_{h_2} + t_e + t_d)$ and $q_{ddh} = O(q_{h_1} + q_{h_2} + q_d)$. Here t_{h_1}, t_{h_2}, t_e and t_d denote the simulation time for the random oracle H_1, the random oracle H_2, the deniably authenticated encryption oracle and the deniably authenticated decryption oracles, respectively.

Proof: $\mathcal{C}$ receives a random instance (g, g^a, g^b) of the GDH problem and attempts to compute $w^* = g^{ab}$. The general idea of this proof is that $\mathcal{C}$ runs $\mathcal{A}$ as a subroutine and plays $\mathcal{A}$'s challenger in the IND-CCA game. $\mathcal{A}$ can ask $\mathcal{C}$ the deniably authenticated encryption queries and deniably authenticated decryption queries. In addition, $\mathcal{A}$ may consult $\mathcal{C}$ for answers to the random oracles H_1 and H_2. Roughly speaking, these answers are randomly produced, but are consistently maintained to avoid collision. $\mathcal{C}$ keeps lists $L_{H_1}^1$ and $L_{H_1}^2$ for the simulation of the random oracle H_1 and keeps lists $L_{H_2}^1$ and $L_{H_2}^2$ for the simulation of the random oracle H_2. If $\mathcal{A}$ wins this game, $\mathcal{C}$ will use $\mathcal{A}$'s queries to compute $w^* = g^{ab}$. This point contradicts the GDH problem assumption.

Initial: At the beginning of the game, $\mathcal{C}$ runs the *Setup* algorithm to get the system parameters *param*. In addition, $\mathcal{C}$ chooses a random number $k^* \in \{0,1\}^n$ for $H_1(w^*)$. Note that w^* is unknown to $\mathcal{C}$ at this stage. $\mathcal{C}$ also chooses e^* and v^* from $\mathbb{Z}_q^*$ and sets the sender's public key $y_s = (g^{v^*}/g^a)^{\frac{1}{e^*}} \bmod p$ and the receiver's public key $y_r = g^b$. $\mathcal{C}$ gives *param*, y_s and y_r to $\mathcal{A}$.

Phase 1: $\mathcal{C}$ deals with $\mathcal{A}$'s queries as follows.

H_1 queries: we use the list $L^1_{H_1}$ to store simple input/output entries for H_1 of the form (w_i, k_i) and list $L^2_{H_1}$ to store special input/output entries for H_1 which are of the form $(\tau_i, ?, k_i)$ and implicitly represents the input/output relation $H_1(\tau_i^{x_r} \bmod p) = k_i$. We denote $\tau_i^{x_r}$ by "?" since it is not explicitly stored. Here $i \in \{1, 2, \ldots, q_{h_1}\}$. For a $H_1(w)$ query, $\mathcal{C}$ does the following:

- If $\mathrm{DDH}(g, g^a, y_r, w) = \top$, then stop and output w as the solution of the GDH problem.
- Else if the oracle $\mathrm{DDH}(g, \tau_i, y_r, w) = \top$ for some $(\tau_i, ?, k_i)$ in $L^2_{H_1}$, then return k_i.
- Else if $w = w_i$ for some (w_i, k_i) in $L^1_{H_1}$, then return k_i.
- Else choose randomly $k_i \in \{0,1\}^n$, put (w, k_i) into $L^1_{H_1}$ and return k_i.

H_2 queries: Similarly to H_1 queries, we use list $L^1_{H_2}$ to store simple input/output entries for H_2 of the form $(m_i \parallel y_s \parallel y_r \parallel w_i, e_i)$ and list $L^2_{H_2}$ to store special input/output entries for H_2 which are of the form $(\tau_i, m_i \parallel y_s \parallel y_r \parallel ?, e_i)$ and implicitly represents the input/output relation $H_2(m_i \parallel y_s \parallel y_r \parallel \tau_i^{x_r} \bmod p) = e_i$. We denote $\tau_i^{x_r}$ by "?" since it is not explicitly stored. For a query $H_2(m \parallel y_s \parallel y_r \parallel w)$, $\mathcal{C}$ does the following:

- If $\mathrm{DDH}(g, g^a, y_r, w) = \top$, then stop and output w as the solution of the GDH problem.
- Else if the oracle $\mathrm{DDH}(g, \tau_i, y_r, w) = \top$ for some $(\tau_i, m_i \parallel y_s \parallel y_r \parallel ?, e_i)$ in $L^2_{H_2}$, then return e_i.
- Else if $(m \parallel y_s \parallel y_r \parallel w, e_i)$ is in $L^1_{H_2}$, return e_i.
- Else choose randomly $e_i \in \mathbb{Z}_q^*$, put $(m \parallel y_s \parallel y_r \parallel w, e_i)$ into $L^1_{H_2}$ and return e_i.

Deniably authenticated encryption queries: when $\mathcal{A}$ makes a deniably authenticated encryption query on a message m, $\mathcal{C}$ first chooses a random $k \in \{0,1\}^n$ and computes $c = m \oplus k$. Then $\mathcal{C}$ chooses randomly $e, v \in \mathbb{Z}_q^*$ and computes $\tau = g^v/y_s^e \bmod p$. $\mathcal{C}$ puts $(\tau, ?, k)$ into $L^2_{H_1}$ and $(m \parallel y_s \parallel y_r \parallel ?, e)$ into $L^2_{H_2}$. Finally, $\mathcal{C}$ computes $z = g^v \bmod p$ and $s = y_r^v \bmod p$, and sends $\sigma = (c, e, z, s)$ to $\mathcal{A}$.

Deniably authenticated decryption queries: when $\mathcal{A}$ makes a deniably authenticated decryption query on a ciphertext $\sigma = (c, e, z, s)$. $\mathcal{C}$ does the following:

- Compute $\tau = z/y_s^e \bmod p$.
- If $\tau = g^a$, terminate.
- If there exists (w_i, k_i) in $L^1_{H_1}$ such that the oracle $\mathrm{DDH}(g, \tau, y_r, w_i) = \top$ or $(\tau_i, ?, k_i)$ in $L^2_{H_1}$ such that $\tau = \tau_i$, set $k' = k_i$.
- Else choose randomly $k' \in \{0,1\}^n$, put $(\tau, ?, k')$ into $L^2_{H_1}$.
- Compute $m = c \oplus k'$.
- If there exists $(m_i \parallel y_s \parallel y_r \parallel w_i, e_i)$ in $L^1_{H_2}$ such that $\mathrm{DDH}(g, \tau, y_r, w_i) = \top$ or there exists $(\tau_i, m_i \parallel y_s \parallel y_r \parallel ?, e_i)$ in $L^2_{H_2}$ such that $\tau = \tau_i$ and $m = m_i$ for some e_i, set $e' = e_i$.
- Else choose randomly $e' \in \mathbb{Z}_q^*$ and put $(\tau, m \parallel y_s \parallel y_r \parallel ?, e')$ in $L^2_{H_2}$.
- If $e = e'$ and $\mathrm{DDH}(g, z, y_r, s) = \top$, then return m.
- Else terminate.

Challenge: $\mathcal{A}$ picks two plaintexts m_0 and m_1. $\mathcal{C}$ takes a random bit β from $\{0,1\}$ and encrypts m_β. To do so, it computes $c^* = m_\beta \oplus k^*$, $z^* = g^{v^*} \bmod p$ and $s^* = y_r^{v^*} \bmod p$. Finally, $\mathcal{C}$ gives the ciphertext $\sigma^* = (c^*, e^*, z^*, s^*)$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ then performs a second series of queries which is treated in the same way as the first one. The only restriction is that it cannot make a deniably authenticated decryption query on the challenged ciphertext σ^*.

Guess: at the end of the simulation, $\mathcal{A}$ produces a bit β' as its guess. Then $\mathcal{C}$ outputs w^* which is a guess for $g^{ab} \bmod p$ and is a preimage of k^*.

We now analyze $\mathcal{C}$'s probability of success. Let us denote by E_0 the event that $\mathcal{A}$ asks $H_1(w^*)$ during the simulation. As done in [50] and [51], as long as the simulation of the attack's environment is perfect, the probability for E_0 to happen is the same as in a real attack. In a real attack, we have

$$\begin{aligned}\Pr[\beta = \beta'] &\le \Pr[\beta = \beta' \mid \neg E_0]\Pr[\neg E_0] + \Pr[E_0] \\ &= \frac{1}{2}(1 - \Pr[E_0]) + \Pr[E_0] \\ &= \frac{1}{2} + \frac{1}{2}\Pr[E_0].\end{aligned}$$

So we have $\epsilon_{dae} = 2\Pr[\beta = \beta'] - 1 \le \Pr[E_0]$. In addition, we note that the simulation only fails in providing a consistent simulation because one of the following independent events:

E_1: $\mathcal{C}$ aborts in a deniably authenticated encryption query because of a collision on H_1 and H_2.

E_2: $\mathcal{C}$ rejects a valid ciphertext in a deniably authenticated decryption query.

We know that

$$\Pr[E_1] \le \frac{q_e(q_{h_1} + q_{h_2})}{2^{l_q(\lambda)}}$$

and

$$\Pr[E_2] \le \frac{q_d}{2^{l_q(\lambda)}}.$$

Therefore, we have

$$\epsilon_{gdh} \ge \epsilon_{dae} - \frac{q_e(q_{h_1} + q_{h_2}) + q_d}{2^{l_q(\lambda)}}.$$

The running time can be readily checked. □

Theorem 2: In the random oracle model, we assume we have a DA-CMA adversary called $\mathcal{F}$ that is able to forge a ciphertext during the DA-CMA game with an advantage ϵ_{dae} when running in a time t and asking at most q_{h_1} H_1 queries, q_{h_2} H_2 queries, q_e deniably authenticated encryption queries and q_d deniably authenticated decryption queries. Then, there exists an algorithm $\mathcal{C}$ that can solve the GDH problem in a time t' and q_{ddh} DDH queries with an advantage

$$\epsilon_{gdh} \geq \epsilon_{dae} - \frac{q_e(q_{h_1} + q_{h_2}) + q_d + 1}{2^{l_q(\lambda)}},$$

where $t' = O(t + t_{h_1} + t_{h_2} + t_e + t_d)$ and $q_{ddh} = O(q_{h_1} + q_{h_2} + q_d)$. Here t_{h_1}, t_{h_2}, t_e and t_d denote the simulation time for the random oracle H_1, the random oracle H_2, the deniably authenticated encryption oracle and the deniably authenticated decryption oracles, respectively.

Proof: $\mathcal{C}$ receives a random instance (g, g^a, g^b) of the GDH problem and attempts to compute g^{ab}. The general idea of this proof is that $\mathcal{C}$ runs $\mathcal{F}$ as a subroutine and plays $\mathcal{F}$'s challenger in the DA-CMA game. $\mathcal{F}$ can adaptively perform H_1 queries, H_2 queries, deniably authenticated encryption queries and deniably authenticated decryption queries. $\mathcal{C}$ also keeps lists $L^1_{H_1}$ and $L^2_{H_1}$ for the simulation of the random oracle H_1 and keeps lists $L^1_{H_2}$ and $L^2_{H_2}$ for the simulation of the random oracle H_2. If $\mathcal{F}$ wins this game, $\mathcal{C}$ will use $\mathcal{F}$'s forgery to compute g^{ab}. This point contradicts the GDH problem assumption.

Initial: At the beginning of the game, $\mathcal{C}$ runs the *Setup* algorithm to get the system parameters *param*. In addition, $\mathcal{C}$ sets the sender's public key $y_s = g^a$ and the receiver's public key $y_r = g^b$. $\mathcal{C}$ gives *param*, y_s and y_r to $\mathcal{F}$.

Attack: $\mathcal{C}$ handles H_1, H_2, deniably authenticated encryption and deniably authenticated decryption queries in the following ways.

H_1 queries: we use list $L^1_{H_1}$ to store simple input/output entries for H_1 of the form (w_i, k_i) and list $L^2_{H_1}$ to store special input/output entries for H_1 which are of the form $(\tau_i, ?, k_i)$ and implicitly represents the input/output relation $H_1(\tau_i^{x_r} \bmod p) = k_i$. We denote $\tau_i^{x_r}$ by "?" since it is not explicitly stored. Here $i \in \{1, 2, \ldots, q_{h_1}\}$. For a $H_1(w)$ query, $\mathcal{C}$ does the following:

- If $\mathrm{DDH}(g, \tau_i, y_r, w) = \top$ for some $(\tau_i, ?, k_i)$ in $L^2_{H_1}$, then return k_i.
- Else if $w = w_i$ for some (w_i, k_i) in $L^1_{H_1}$, then return k_i.
- Else choose randomly $k_i \in \{0, 1\}^n$, put (w, k_i) into $L^1_{H_1}$ and return k_i.

H_2 queries: Similarly to H_1 queries, we use list $L^1_{H_2}$ to store simple input/output entries for H_2 of the form $(m_i \parallel y_s \parallel y_r \parallel w_i, e_i)$ and list $L^2_{H_2}$ to store special input/output entries for H_2 which are of the form $(\tau_i, m_i \parallel y_s \parallel y_r \parallel ?, e_i)$ and implicitly represents the input/output relation $H_2(m_i \parallel y_s \parallel y_r \parallel \tau_i^{x_r} \bmod p) = e_i$. We denote $\tau_i^{x_r}$ by "?" since it is not explicitly stored. For a query $H_2(m \parallel y_s \parallel y_r \parallel w)$, $\mathcal{C}$ does the following:

- If $\mathrm{DDH}(g, \tau_i, y_r, w) = \top$ for some $(\tau_i, m_i \parallel y_s \parallel y_r \parallel ?, e_i)$ in $L^2_{H_2}$, then return e_i.
- Else if $(m \parallel y_s \parallel y_r \parallel w, e_i)$ is in $L^1_{H_2}$, return e_i.
- Else choose randomly $e_i \in \mathbb{Z}_q^*$, put $(m \parallel y_s \parallel y_r \parallel w, e_i)$ into $L^1_{H_2}$ and return e_i.

Deniably authenticated encryption queries: when $\mathcal{F}$ makes a deniably authenticated encryption query on a message m, $\mathcal{C}$ first chooses a random $k \in \{0, 1\}^n$ and computes $c = m \oplus k$. Then $\mathcal{C}$ chooses randomly $e, v \in \mathbb{Z}_q^*$ and computes $\tau = g^v / y_s^e \bmod p$. $\mathcal{C}$ puts $(\tau, ?, k)$ into $L^2_{H_1}$ and $(m \parallel y_s \parallel y_r \parallel ?, e)$ into $L^2_{H_2}$. Finally, $\mathcal{C}$ computes $z = g^v \bmod p$, $s = y_r^v \bmod p$ and sends $\sigma = (c, e, z, s)$ to $\mathcal{F}$.

Deniably authenticated decryption queries: when $\mathcal{F}$ makes a deniably authenticated decryption query on a ciphertext $\sigma = (c, e, z, s)$. $\mathcal{C}$ does the following:

- Compute $\tau = z / y_s^e \bmod p$.
- if there exists (w_i, k_i) in $L^1_{H_1}$ such that $\mathrm{DDH}(g, \tau, y_r, w_i) = \top$ or $(\tau_i, ?, k_i)$ in $L^2_{H_1}$ such that $\tau = \tau_i$, set $k' = k_i$.
- Else choose randomly $k' \in \{0, 1\}^n$, put $(\tau, ?, k')$ into $L^2_{H_1}$.
- Compute $m = c \oplus k'$.
- If there exists $(m_i \parallel y_s \parallel y_r \parallel w_i, e_i)$ in $L^1_{H_2}$ such that $\mathrm{DDH}(g, \tau, y_r, w_i) = \top$ or there exists $(\tau_i, m_i \parallel y_s \parallel y_r \parallel ?, e_i)$ in $L^2_{H_2}$ such that $\tau = \tau_i$ and $m = m_i$ for some e_i, set $e' = e_i$.
- Else choose randomly $e' \in \mathbb{Z}_q^*$ and put $(\tau, m \parallel y_s \parallel y_r \parallel ?, e')$ in $L^2_{H_2}$.
- If $e = e'$ and $\mathrm{DDH}(g, z, y_r, s) = \top$, then return m.
- Else terminate.

Forgery: At the end of the game, $\mathcal{F}$ produces a ciphertext $\sigma' = (c', e', z', s')$.

If the hash value $H_2(m' \parallel y_s \parallel y_r \parallel w')$ was not asked by $\mathcal{F}$ during the simulation, $\mathcal{C}$ fails and stops. Otherwise, $\mathcal{C}$ searches $L^1_{H_2}$ and $L^2_{H_2}$ to find w' corresponding to e'. Then $\mathcal{C}$ can solve the GDH problem by computing $(w' s'^{-1})^{\frac{-1}{e'}}$. Since $w' = y_r^{x'} \bmod p$, $s' = y_r^{v'} \bmod p$ and $v' = e' x_s + x' \bmod q$, we have

$$(w' s'^{-1})^{\frac{-1}{e'}} = (y_r^{x'} y_r^{-v'})^{\frac{-1}{e'}} = (y_r^{x'} y_r^{-e' x_s - x'})^{\frac{-1}{e'}} = y_r^{x_s} = g^{ab}.$$

We now analyze $\mathcal{C}$'s probability of success. Let us denote by E_0 the event that $\mathcal{F}$ succeeds in producing a forged ciphertext $\sigma' = (c', e', z', s')$ without asking the query $H_2(m' \parallel y_s \parallel y_r \parallel w')$. We know that

$$\Pr[E_0] \leq \frac{1}{2^{l_q(\lambda)}}.$$

We note that it only fails in providing a consistent simulation because of one of the following events:

E_1: $\mathcal{C}$ aborts in a deniably authenticated encryption query because of a collision on H_1 and H_2.

E_2: $\mathcal{C}$ rejects a valid ciphertext in a deniably authenticated decryption query.

We know that

$$\Pr[E_1] \leq \frac{q_e(q_{h_1} + q_{h_2})}{2^{l_q(\lambda)}}$$

and

$$\Pr[E_2] \leq \frac{q_d}{2^{l_q(\lambda)}}.$$

Therefore, we have

$$\epsilon_{gdh} \geq \epsilon_{dae} - \frac{q_e(q_{h_1} + q_{h_2}) + q_d + 1}{2^{l_q(\lambda)}}.$$

The running time can be readily checked. □

TABLE I

PERFORMANCE COMPARISON

Schemes	Computational cost		Ciphertext size	Security		Formal	Non-inte-
	Sender	Receiver		IND-CCA	DA-CMA	proof	ractive
HR [3]	$T_h+3T_e+3T_m+T_i$	$T_h+4T_e+2T_m+T_i$	$\|m\|+4\|p\|+\|h\|+\|T\|$	?	×	×	√
KHNLL [4]	$2T_h+5T_e+2T_m$	$2T_h+4T_e+3T_m+T_i$	$\|m\|+5\|p\|+\|h\|+\|T\|$	?	?	×	√
HLLC [5]	$2T_h+4T_e$	$2T_h+3T_e$	$2\|m\|+2\|p\|+2\|h\|$	?	?	×	×
HS [6]	$2T_h+3T_e+T_m$	$2T_h+2T_e+T_m+T_i$	$\|m\|+2\|q\|+\|p\|$	?	√	√	√
HSC [7]	$6T_h+8T_e+3T_m$	$3T_h+8T_e+4T_m+T_i$	$\|m\|+6\|q\|+\|p\|$	?	√	√	√
Ours	$2T_h+3T_e+T_m$	$2T_h+3T_e+T_m+T_i$	$\|m\|+\|q\|+2\|p\|$	√	√	√	√

E. Comparison

We compare the major computational cost, ciphertext size, security, formal proof and non-interactive characteristic of our scheme with those of related works [3]–[7] in Table I. For convenience, the following notation is used: T_h is the time for executing a hash function; T_e is the time for executing a modular exponentiation operation; T_m is the time for executing a modular multiplication operation; T_i is the time for executing a modular inverse operation; $|\chi|$ is the size of message χ; √ denotes that this scheme satisfies this property; × denotes that this scheme does not satisfy this property; and ? denotes that this scheme is not clearly showed to satisfy this property. Note that the time for computing addition and exclusive OR (or symmetric encryption and decryption) is ignored because they are much smaller than T_h, T_e, T_m and T_i.

For the HR scheme [3], we use ElGamal's encryption and signature scheme [29] as an example. For the KHNLL scheme [4], we use ElGamal encryption and their designated verifier signature scheme. Although the security of their designated verifier signature was proved, the combined security of designated verifier signature and encryption has not been proved. An inappropriate combination of signature and encryption will result in an insecure system. So we think their scheme does not provide formal security. For the HLLC scheme [5], we use the protocol based on Diffie-Hellman key exchange as an example. Note that $|h|$ is the size of hash function $H_k(m \parallel T)$ used in [3]. Here T is a timestamp. In [5], they used MAC instead of hash function. We assume that the computational cost and size of MAC are the same as those of hash function.

From Table I, we know that the HR, KHNLL and HLLC schemes can not achieve formal security proof (this point can be found in [3]–[5]). In addition, the HLLC scheme is an interactive protocol that can not be used in e-mail systems. Both the HS scheme and the HSC scheme are only proved to satisfy the IND-CPA in [6] and [7], respectively. The IND-CPA is a weaker model than the IND-CCA. In the IND-CPA, the adversary can make encryption queries but can not make decryption queries. In the IND-CCA, the adversary can make both encryption queries and decryption queries. That is, the adversary obtains more power and training in the IND-CCA model than in the IND-CPA model. Therefore, a scheme that is secure in the IND-CPA model does not mean that it is also secure in the IND-CCA model. Note that the IND-CCA security has been widely accepted as the standard security concept for a public key encryption scheme. In the HS and HSC schemes, the adversary can not make decryption queries. So both the HS scheme and the HSC scheme may be broken by a CCA adversary in the future. For a real application, we require that a scheme should satisfy the IND-CCA security. An IND-CPA scheme can not be used in the real world. In our scheme, the adversary can make decryption queries. That is, our scheme is clearly proved to satisfy the IND-CCA security. This point is an important difference between our scheme with previous related works, the HR, KHNLL, HLLC, HS and HSC schemes. In addition, our scheme is also proved to satisfy the DA-CMA security. From efficiency, our scheme is similar to the HS and HLLC schemes and is higher than the HR, KHNLL and HSC schemes.

TABLE II

SPECIFICATIONS FOR DIFFERENT SECURITY LEVEL OF OUR IMPLEMENTATION (BITS)

Security level	Size of p	Size of q
80-bit	1024	160
112-bit	2048	224
128-bit	3072	256

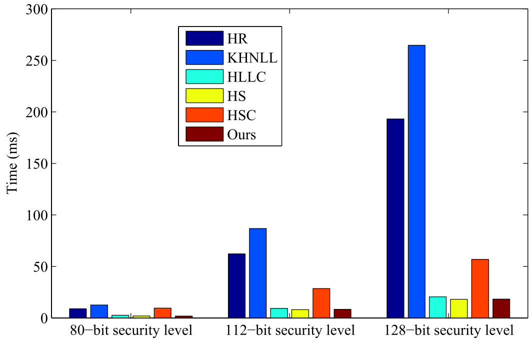


Fig. 2. The sender's computational time.

We implement the six schemes using MIRACL library [52] on an Intel Core i7 4770S 3.10 GHz machine with 4G RAM. The MIRACL library is the gold standard among cryptographic software development kit for easily implementing big number cryptography. In this implementation, we use three types of parameters that represents 80-bit, 112-bit and 128-bit AES [49] key sizes security level, respectively. Table II gives the concrete specifications for different security level of our implementation.

Fig. 2 and Fig. 3 respectively gives the computational time (average time of running 3000 times algorithm) of the sender and the receiver for the six schemes at the 80-bit, 112-bit and 128-bit security level. The implementation result is consistent with the theoretical analysis. The computational time of the HR and KHNLL schemes is obviously higher than the other four schemes. The reason is that the HR and KHNLL

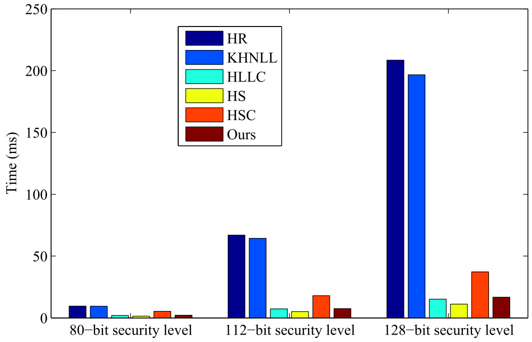


Fig. 3. The receiver's computational time.

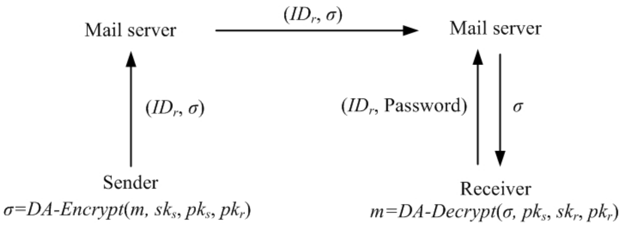


Fig. 4. A secure e-mail protocol.

schemes choses random number in $\mathbb{Z}_p^*$, not in $\mathbb{Z}_q^*$. From Fig. 2 and Fig. 3, we know that our scheme only needs 1.75 ms to encrypt a message and 2.14 ms to decrypt a ciphertext at the 80-bit security level. This time is sound for practical applications. If we adopt higher security level, we consume more computational cost.

IV. A Secure e-Mail Protocol

In this section, we design a secure e-mail protocol using the proposed deniably authenticated encryption scheme. This protocol is described in Fig. 4.

In this secure e-mail protocol, the sender first runs *DA-Encrypt*(m, sk_s, pk_s, pk_r) to obtain the ciphertext σ. The sender transmits the receiver's identity ID_r and the ciphertext σ to its mail server. Then the sender's mail server transfers the (ID_r, σ) to the receiver's mail server. The receiver's mail server stores (ID_r, σ) and waits for the receiver. When the receiver wants to receive its mails, it sends its identity ID_r and password to its mail server for identity authentication. If the receiver passes the identity authentication, the mail server sends the ciphertext σ to the receiver. Finally, the receiver runs *DA-Decrypt*$(\sigma, pk_s, sk_r, pk_r)$ to obtain the message m.

Different to PGP and S/MIME, the designed e-mail protocol can efficiently protect the privacy of the sender since this protocol uses our DAE scheme. The receiver can identify the source of a given e-mail but cannot prove the source of the given e-mail to any third party. The sender is more willing to use our protocol for sending e-mails.

V. Conclusions

In this paper, we proposed a new concept called deniably authenticated encryption that can achieve confidentiality and deniable authentication in a logical single step. We also proposed a deniably authenticated encryption scheme and proved its security in the random oracle model. The confidentiality and deniable authentication of this scheme are based on the gap Diffie-Hellman problem. As compared with existing five related schemes, our scheme is clearly proved to satisfy both the IND-CCA security and the DA-CMA security. We also design a secure e-mail protocol using the proposed deniably authenticated encryption scheme.

References

[1] S. Garfinkel, *PGP: Pretty Good Privacy*. Sebastopol, CA, USA: OReilly, 1994.

[2] B. Ramsdell, *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification, RFC 3851*. Reston, VA, USA: The Internet Society, 2004.

[3] L. Harn and J. Ren, "Design of fully deniable authentication service for e-mail applications," *IEEE Commun. Lett.*, vol. 12, no. 3, pp. 219–221, Mar. 2008.

[4] J. Ki, J. Y. Hwang, D. Nyang, D. H. Lee, and J. Lim, "Privacy-enhanced deniable authentication E-mail service," in *Digital Enterprise and Information Systems* (Communications in Computer and Information Science), vol. 194. Berlin, Germany: Springer-Verlag, 2011, pp. 16–29.

[5] L. Harn, C.-Y. Lee, C. Lin, and C.-C. Chang, "Fully deniable message authentication protocols preserving confidentiality," *Comput. J.*, vol. 54, no. 10, pp. 1688–1699, Oct. 2011.

[6] S.-J. Hwang and Y.-H. Sung, "Confidential deniable authentication using promised signcryption," *J. Syst. Softw.*, vol. 84, no. 10, pp. 1652–1659, Oct. 2011.

[7] S.-J. Hwang, Y.-H. Sung, and J.-F. Chi, "Deniable authentication protocols with confidentiality and anonymous fair protections," in *Advances in Intelligent Systems and Applications* (Smart Innovation, Systems and Technologies). Berlin, Germany: Springer-Verlag, 2013, pp. 41–51.

[8] H. Nicanfar, P. Jokar, K. Beznosov, and V. C. M. Leung, "Efficient authentication and key management mechanisms for smart grid communications," *IEEE Syst. J.*, vol. 8, no. 2, pp. 629–640, Jun. 2014.

[9] M. C. Chuang and J. F. Lee, "TEAM: Trust-extended authentication mechanism for vehicular ad hoc networks," *IEEE Syst. J.*, vol. 8, no. 3, pp. 749–758, Sep. 2014.

[10] S. Ruj, M. Stojmenovic, and A. Nayak, "Decentralized access control with anonymous authentication of data stored in clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 384–394, Feb. 2014.

[11] T. Kwon and J. Hong, "Secure and efficient broadcast authentication in wireless sensor networks," *IEEE Trans. Comput.*, vol. 59, no. 8, pp. 1120–1133, Aug. 2010.

[12] A. A. Yavuz, "An efficient real-time broadcast authentication scheme for command and control messages," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 10, pp. 1733–1742, Oct. 2014.

[13] B. Sayed, I. Traoré, I. Woungang, and M. S. Obaidat, "Biometric authentication using mouse gesture dynamics," *IEEE Syst. J.*, vol. 7, no. 2, pp. 262–274, Jun. 2013.

[14] J. Liu, Z. Zhang, X. Chen, and K. S. Kwak, "Certificateless remote anonymous authentication schemes for wireless body area networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 332–342, Feb. 2014.

[15] L. Harn, "Group authentication," *IEEE Trans. Comput.*, vol. 62, no. 9, pp. 1893–1898, Sep. 2013.

[16] D. He, J. Bu, S. Chan, and C. Chen, "Handauth: Efficient handover authentication with conditional privacy for wireless networks," *IEEE Trans. Comput.*, vol. 62, no. 3, pp. 616–622, Mar. 2013.

[17] A. C.-C. Yao and Y. Zhao, "Privacy-preserving authenticated key-exchange over Internet," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 1, pp. 125–140, Jan. 2014.

[18] Y. Aumann and M. O. Rabin, "Authentication, enhanced security and error correcting codes," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 1462. Berlin, Germany: Springer-Verlag, 1998, pp. 299–303.

[19] C. Dwork, M. Naor, and A. Sahai, "Concurrent zero-knowledge," in *Proc. 30th ACM Symp. Theory Comput.*, Dallas TX, USA, 1998, pp. 409–418.

[20] Y. Aumann and M. Rabin, "Efficient deniable authentication of long messages," in *Proc. Int. Conf. Theor. Comput. Sci. Honor Professor Manuel Blum's 60th Birthday*, 1998. [Online]. Available: http://www.cs.cityu.edu.hk/dept/video.html

[21] X. Deng, C. H. Lee, and H. Zhu, "Deniable authentication protocols," *IEE Proc.-Comput. Digit. Techn.*, vol. 148, no. 2, pp. 101–104, Mar. 2001.

[22] L. Fan, C. X. Xu, and J. H. Li, "Deniable authentication protocol based on Deffie-Hellman algorithm," *Electron. Lett.*, vol. 38, no. 14, pp. 705–706, Jul. 2002.

[23] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[24] E.-J. Yoon, E.-K. Ryu, and K.-Y. Yoo, "Improvement of Fan et al.'s deniable authentication protocol based on Diffie–Hellman algorithm," *Appl. Math. Comput.*, vol. 167, no. 1, pp. 274–280, Aug. 2005.

[25] Z. Shao, "Efficient deniable authentication protocol based on generalized ElGamal signature scheme," *Comput. Standards Interfaces*, vol. 26, no. 5, pp. 449–454, Sep. 2004.

[26] L. Harn and Y. Xu, "Design of generalized ElGamal type digital signature schemes based on discrete logarithm," *Electron. Lett.*, vol. 30, no. 24, pp. 2025–2026, Nov. 1994.

[27] R. Lu and Z. Cao, "Non-interactive deniable authentication protocol based on factoring," *Comput. Standards Interfaces*, vol. 27, no. 4, pp. 401–405, Apr. 2005.

[28] Y. Wang, J. Li, and L. Tie, "A simple protocol for deniable authentication based on ElGamal cryptography," *Networks*, vol. 45, no. 4, pp. 193–194, Jul. 2005.

[29] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.

[30] J. Shao, Z. Cao, and R. Lu, "An improved deniable authentication protocol," *Networks*, vol. 48, no. 4, pp. 179–181, Dec. 2006.

[31] E.-J. Yoon, K.-Y. Yoo, S.-S. Yeo, and C. Lee, "Robust deniable authentication protocol," *Wireless Pers. Commun.*, vol. 55, no. 1, pp. 81–90, Sep. 2010.

[32] F. Li and T. Takagi, "Cryptanalysis and improvement of robust deniable authentication protocol," *Wireless Pers. Commun.*, vol. 69, no. 4, pp. 1391–1398, Apr. 2013.

[33] W.-B. Lee, C.-C. Wu, and W.-J. Tsaur, "A novel deniable authentication protocol using generalized ElGamal signature scheme," *Inf. Sci.*, vol. 177, no. 6, pp. 1376–1381, Mar. 2007.

[34] B. Wang and Z. Song, "A non-interactive deniable authentication scheme based on designated verifier proofs," *Inf. Sci.*, vol. 179, no. 6, pp. 858–865, Mar. 2009.

[35] F. Li, P. Xiong, and C. Jin, "Identity-based deniable authentication for ad hoc networks," *Computing*, vol. 96, no. 9, pp. 843–853, Sep. 2014.

[36] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," *J. Cryptol.*, vol. 21, no. 4, pp. 469–491, Oct. 2008.

[37] H. Petersen, "Authenticated encryption schemes with low communication costs," *Electron. Lett.*, vol. 30, no. 15, pp. 1212–1213, Jul. 1994.

[38] P. Sarkar, "A simple and generic construction of authenticated encryption with associated data," *ACM Trans. Inf. Syst. Secur.*, vol. 13, no. 4, 2010, Art. no. 33.

[39] F. Li, J. Deng, and T. Takagi, "An improved authenticated encryption scheme," *IEICE Trans. Inf. Syst.*, vol. E94-D, no. 11, pp. 2171–2172, Nov. 2011.

[40] H.-M. Sun, B.-T. Hsieh, and H.-J. Hwang, "Secure E-mail protocols providing perfect forward secrecy," *IEEE Commun. Lett.*, vol. 9, no. 1, pp. 58–60, Jan. 2005.

[41] A. Dent, "Flaws in an e-mail protocol of Sun, Hsieh, and Hwang," *IEEE Commun. Lett.*, vol. 9, no. 8, pp. 718–719, Aug. 2005.

[42] B. H. Kim, J. H. Koo, and D. H. Lee, "Robust E-mail protocols with perfect forward secrecy," *IEEE Commun. Lett.*, vol. 10, no. 6, pp. 510–512, Jun. 2006.

[43] E. J. Yoon and K. Y. Yoo, "Cryptanalysis of robust e-mail protocols with perfect forward secrecy," *IEEE Commun. Lett.*, vol. 11, no. 5, pp. 372–374, May 2007.

[44] C.-C. Chang, C.-Y. Lee, and Y.-C. Chiu, "An efficient e-mail protocol providing perfect forward secrecy for mobile devices," *Int. J. Commun. Syst.*, vol. 23, no. 12, pp. 1463–1473, Dec. 2010.

[45] M. Zhang and T. Takagi, "Efficient constructions of anonymous multireceiver encryption protocol and their deployment in group e-mail systems with privacy preservation," *IEEE Syst. J.*, vol. 7, no. 3, pp. 410–419, Sep. 2013.

[46] H.-C. Chen, "Secure multicast key protocol for electronic mail systems with providing perfect forward secrecy," *Secur. Commun. Netw.*, vol. 6, no. 1, pp. 100–107, Jan. 2013.

[47] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," *SIAM J. Comput.*, vol. 33, no. 1, pp. 167–226, 2003.

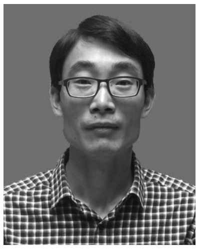
[48] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM J. Comput.*, vol. 17, no. 2, pp. 281–308, 1988. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/021701

[49] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*. Berlin, Germany: Springer, 2002.

[50] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.

[51] X. Boyen, "Multipurpose identity-based signcryption: A Swiss army knife for identity-based cryptography," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 2729. Berlin, Germany: Springer-Verlag, 2003, pp. 383–399.

[52] *MIRACL-Multiprecision Integer and Rational Arithmetic C Library*, accessed on Sep. 2015. [Online]. Available: http://www.certivox.com/miracl

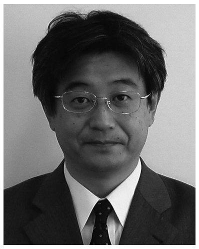


Fagen Li (M'14) received the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2007.

From 2008 to 2009, he was a Postdoctoral Fellow with Future University-Hakodate, Hokkaido, Japan, which is supported by the Japan Society for the Promotion of Science. He was a Research Fellow with the Institute of Mathematics for Industry, Kyushu University, Fukuoka, Japan, from 2010 to 2012. He is currently an Associate Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. He has authored more than 70 papers in international journals and conferences. His recent research interests include cryptography and network security.

Di Zhong is currently pursuing the M.S. degree with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China.

His research interests are cryptography and information security.



Tsuyoshi Takagi received the B.S. and M.S. degrees in mathematics from Nagoya University, Nagoya, Japan, in 1993 and 1995, respectively, and the Dr.rer.nat. degree from Technische University Darmstadt in 2001.

He was an Assistant Professor with the Department of Computer Science, Technische University Darmstadt until 2005. He is currently a Professor with the Institute of Mathematics for Industry, Kyushu University. He engaged in research on network security with NTT Laboratories from 1995 to 2001. His current research interests are information security and cryptography.