Universal Encrypted Deniable Authentication Protocol

Zhenfu Cao
Department of Computer Science and Engineering, Shanghai Jiao Tong University
No.800, Dongchuan Rd., Minhang District, Shanghai, 200240, China (Email: zfcao@cs.sjtu.edu.cn)

(Received Jan. 2, 2007; accepted June 4, 2007)

Abstract

The notion of deniable authentication protocol was introduced in 1998 by Dwork et al. [11] and Aumann and Rabin [1, 2] independently. As a new cryptographic authentication protocol, a deniable authentication protocol enables an intended receiver to identify the source of a given message without being able to prove the identity of the sender to a third party. Over the past years, many deniable authentication protocols have been proposed. In this paper, we would like to present a universal encrypted deniable authentication protocol from any public key encryption scheme. Provided that the public key encryption scheme is secure, a secure deniable authentication protocol can be directly derived.

Keywords: Authentication protocol, deniable authentication protocol, public key encryption

1 Introduction

The notion of deniable authentication protocol was introduced in 1998 by Dwork et al. [11] and Aumann and Rabin [1, 2] independently. As a new technique compared with the traditional authentication protocols, it has the following two characteristics:

1) It enables an intended receiver to identify the source of a given message just like the traditional authentication protocol.

2) However, the intended receiver cannot prove to any third party the identity of the sender.

Just due to these two characteristics, deniable authentication protocol can help the sender control who is able to authenticate his messages and who cannot, which therefore becomes a solution to the special requirements for practical applications, such as being used to solve the coercion problem in electronic voting systems and constructing a secure electronic negotiation system in electronic commerce [1, 2, 11].

Over the past years, many researchers have done deep researches on deniable authentication protocol [1, 7, 10, 11, 14, 16, 17, 18, 19, 23, 27, 28, 29, 30, 31]. Dwork et al. [11] developed a deniable authentication protocol based on concurrent zero-knowledge proof. However, their protocol suffers from a time constraint, and the proof of knowledge is subject to a time delay during the authentication process. Aumann and Rabin [1] proposed another deniable authentication scheme based on the factoring problem. Lately, Deng et al. [10] also developed two deniable authentication protocols based on factoring and discrete logarithm problems, respectively. However, these schemes of Deng et al. [10] and Aumann-Rabin [1] both need a public directory which is trusted by the sender and the receiver. In 2002, to solve this problem, Fan et al. [14] proposed a simple deniable authentication protocol based on the Diffie-Hellman [9] key distribution protocol. More recently, other deniable authentication protocols have been presented [18, 28, 30, 31]. In addition to these interactive deniable authentication protocols, several non-interactive deniable authentication protocols also have been proposed [7, 16, 17, 19, 23, 27, 29].

In this paper, we would like to present a universal deniable authentication protocol from any public key encryption scheme. Only if these exists a secure public key encryption scheme Π, such as RSA [25], ElGamal [12], Identity-based encryption [4], we can directly derive a secure encrypted deniable authentication protocol from Π.

The rest of this paper is organized as follows. In the next section, we first briefly review some preliminarily work, namely some used notations and public key encryption. Then, we formalize the definition of the universal encrypted deniable authentication protocol and its security notions in Section 3. In Sections 4 and 5, we present the universal encrypted deniable authentication protocol (UEDAP) from any public key encryption with a precise security treatment and efficiency analysis. And in Section 6, we give some examples of UEDAP. Finally, we draw our conclusion in Section 7.

2 Preliminaries

2.1 Notations

Throughout this paper, if x is a string then $|x|$ denotes its length, and if S is a set then $|S|$ denotes its size. We denote by $\mathbb{N}$ the set of positive integers, and the integer $k \in \mathbb{N}$ denotes the security parameter. We say a function $\epsilon(k) : \mathbb{N} \mapsto [0,1]$ is *negligible* if for all $\alpha > 0$, $\epsilon(k) < 1/k^{\alpha}$ for all sufficiently large k [20]. Assume that $\mathcal{A}$ is a probabilistic algorithm that runs in polynomial time with respect to the security parameter k. Then we denote $z \leftarrow \mathcal{A}(x, y, \cdots)$ as the operation of running $\mathcal{A}$ with inputs $x, y, \cdots$ and output z, $z \leftarrow \mathcal{A}(x, y, \cdots, \mathcal{O}_1, \mathcal{O}_2, \cdots)$ as the operation of running $\mathcal{A}$ with inputs $x, y, \cdots$, accesses to oracle $\mathcal{O}_1, \mathcal{O}_2, \cdots$ and output z. Furthermore, we denote $\perp$ as no oracle.

2.2 Public Key Encryption

In this subsection, we recall the formal definition for public key encryption schemes, together with the security notions.

Definition 1. *A public key encryption scheme $\Pi = (\mathbf{KGen}, \mathbf{Enc}, \mathbf{Dec})$ consists of the following three polynomial-time (in k) algorithms:*

- *The key generation algorithm – KGen: On input 1^k (unary representation of k), the algorithm KGen produces a pair (pk, sk) of matching public and private keys. Algorithm KGen is probabilistic.*
- *The encryption algorithm – Enc: Given a message m and a public key pk, Enc produces a ciphertext $c = \Pi(m)$ of m. This algorithm may be probabilistic.*
- *The decryption algorithm – Dec: Given a ciphertext $c = \Pi(m)$ and the private key sk. Dec(sk, c) gives back the plaintext m. This algorithm is necessarily deterministic.*

In addition, for every pair (pk, sk) generated by $\mathbf{KGen}(1^k)$, algorithms Enc and Dec satisfy

$$\Pr[\mathbf{Dec}(sk, \mathbf{Enc}(pk, m)) = m] = 1,$$

where the probability is taken over the internal coin tosses of algorithm Enc and Dec.

Adversarial goals. The basic security notion required from a public key encryption scheme is the one-wayness (OW), which roughly means that one can't recover the whole plaintext from a given ciphertext.

Definition 2 (One-Wayness). *A public key encryption scheme $\Pi = (\mathbf{KGen}, \mathbf{Enc}, \mathbf{Dec})$ is said to be one-way if for all probabilistic polynomial time algorithms $\mathcal{A}$, for every $\alpha > 0$ and sufficiently large k,*

$$\Pr[\mathcal{A}(pk, c) = \mathbf{Dec}(sk, c) = m] < \frac{1}{k^{\alpha}},$$

where $c = \Pi(m) \leftarrow \mathbf{Enc}(pk, m)$, $(pk, sk) \leftarrow \mathbf{KGen}(1^k)$ and m is any message in message space.

A stronger security notion for a public key encryption scheme is the so-called semantic security (a.k.a. indistinguishability (IND) of encryption) [15]. This security notion requires computational impossibility to distinguish between two messages chosen by an adversary, which one has been encrypted, with a non-negligible probability better than 1/2.

Definition 3 (Semantic Security). *A public key encryption scheme $\Pi = (\mathbf{KGen}, \mathbf{Enc}, \mathbf{Dec})$ is said to be semantic security if for all probabilistic polynomial time algorithms $\mathcal{A}$, for every $\alpha > 0$ and sufficiently large k,*

$$\Pr[\mathcal{A}(pk, m_0, m_1, c) = m] < \frac{1}{2} + \frac{1}{k^{\alpha}},$$

where (m_0, m_1) is chosen by $\mathcal{A}$, $m \leftarrow \{m_0, m_1\}$, $c = \Pi(m) \leftarrow \mathbf{Enc}(pk, m)$, $(pk, sk) \leftarrow \mathbf{KGen}(1^k)$.

Adversarial models. Currently, there are several types of attacks models for public key encryption, namely the chosen-plaintext attack (CPA), non-adaptive chosen-ciphertext attacks (CCA1) [21] and adaptive chosen-ciphertext attacks (CCA2) [24]. In a CPA, an adversary can access an encryption oracle. This scenario clearly cannot be avoided. In a CCA1, an adversary also can access a decryption oracle before being given the challenge ciphertext. While in a CCA2, an adversary can access a decryption oracle before and after being challenged; and the only restriction for him is that he cannot feed the oracle with the challenge ciphertext himself. This is the strongest known attack scenario.

Security levels are usually defined by pairing each goal (OW, IND) with an attack model (CPA, CCA1 or CCA2); *i.e.*, OW-CPA, OW-CCA1, OW-CCA2; IND-CPA, IND-CCA1 and IND-CCA2. Among each security level, the following relations are satisfied.

$$\begin{tikzcd}
\text{OW-CPA} & \text{OW-CCA1} \arrow[l] & \text{OW-CCA2} \arrow[l] \\
\text{IND-CPA} \arrow[u] & \text{IND-CCA1} \arrow[l] \arrow[u] & \text{IND-CCA2} \arrow[l] \arrow[u]
\end{tikzcd}$$

Definition 4 (OW-ATK). *Let $\Pi = (\mathbf{KGen}, \mathbf{Enc}, \mathbf{Dec})$ be a public key encryption scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be any probabilistic polynomial time algorithm. For ATK $\in$ {CPA, CCA1, CCA2}, under sufficiently large k, let define*

$$\mathbf{Succ}_{\mathcal{A},\Pi}^{\mathsf{OW\text{-}ATK}} := \Pr\left[\begin{array}{c} (pk, sk) \leftarrow \mathbf{KGen}(1^k); \\ s \leftarrow \mathcal{A}_1(pk, \mathcal{O}_1) \\ c = \mathbf{Enc}(pk, m) : \\ \mathcal{A}_2(s, c, \mathcal{O}_2) = m \end{array}\right],$$

where s is $\mathcal{A}$'s inner statement information, $\mathcal{O}_1, \mathcal{O}_2$ are oracles that $\mathcal{A}$ can access. According to each attack, $\mathcal{O}_1, \mathcal{O}_2$ are defined as follows:

- *If* $\mathsf{ATK} = \mathsf{CPA}$ *then* $\mathcal{O}_1(\cdot) = \perp$ *and* $\mathcal{O}_2(\cdot) = \perp$*;*
- *If* $\mathsf{ATK} = \mathsf{CCA1}$ *then* $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ *and* $\mathcal{O}_2(\cdot) = \perp$*;*
- *If* $\mathsf{ATK} = \mathsf{CCA2}$ *then* $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ *and* $\mathcal{O}_2(\cdot) = \mathcal{D}_{sk}(\cdot)$*.*

Here a limitation is that $\mathcal{A}_2$ *is not allowed to make access to decryption oracle with the challenge c itself as a query. We say that* Π *is* (t, q_D, ϵ)*-secure if for every adversary* $\mathcal{A}$ *that runs at most in time t, achieving* $\mathbf{Succ}^{\mathsf{OW\text{-}ATK}}_{\mathcal{A},\Pi}(k) < \epsilon$*, where* q_D *is the query times on decryption oracle* $\mathcal{D}_{sk}(\cdot)$*.*

Definition 5 (IND-ATK). *Let* $\Pi = (\mathbf{KGen}, \mathbf{Enc}, \mathbf{Dec})$ *be a public key encryption scheme and let* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be any probabilistic polynomial time algorithm. For* $\mathsf{ATK} \in \{\mathsf{CPA, CCA1, CCA2}\}$*, under sufficiently large k, let define*

$$\mathbf{Adv}^{\mathsf{IND\text{-}ATK}}_{\mathcal{A},\Pi} := 2 \times \Pr\left[\begin{array}{c} (pk, sk) \leftarrow \mathbf{KGen}(1^k); \\ (m_0, m_1, s) \rightarrow \mathcal{A}_1(pk, \mathcal{O}_1); \\ b \xleftarrow{R} \{0, 1\}; \\ c = \mathbf{Enc}(pk, m_b) : \\ \mathcal{A}_2(m_0, m_1, s, c, \mathcal{O}_1) = b \end{array}\right] - 1,$$

where s is $\mathcal{A}$*'s inner statement information,* $\mathcal{O}_1, \mathcal{O}_2$ *are oracles that* $\mathcal{A}$ *can access. According to each attack,* $\mathcal{O}_1, \mathcal{O}_2$ *are defined as follows:*

- *If* $\mathsf{ATK} = \mathsf{CPA}$ *then* $\mathcal{O}_1(\cdot) = \perp$ *and* $\mathcal{O}_2(\cdot) = \perp$*;*
- *If* $\mathsf{ATK} = \mathsf{CCA1}$ *then* $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ *and* $\mathcal{O}_2(\cdot) = \perp$*;*
- *If* $\mathsf{ATK} = \mathsf{CCA2}$ *then* $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ *and* $\mathcal{O}_2(\cdot) = \mathcal{D}_{sk}(\cdot)$*.*

Here a limitation is that $\mathcal{A}_2$ *is not allowed to make access to decryption oracle with the challenge c itself as a query. We say that* Π *is* (t, q_D, ϵ)*-secure if for every adversary* $\mathcal{A}$ *that runs at most in time t, achieving* $\mathbf{Adv}^{\mathsf{IND\text{-}ATK}}_{\mathcal{A},\Pi}(k) < \epsilon$*, where* q_D *is the query times on decryption oracle* $\mathcal{D}_{sk}(\cdot)$*.*

Remark. The above security notion is defined in the standard model. In the random oracle model [5], one should think $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is also allowed to make access to random oracle $\mathcal{O}_H$. To date, the strongest security notion for public key encryption is IND-CCA2[1]. In the standard model, the typical IND-CCA2 public key encryption scheme is Crame-Shoup scheme [8]; and the typical IND-CCA2 public key encryption schemes in the random oracle model include OAEP [6] and others [13, 22]. Identity-based public key cryptography is a paradigm introduced by Shamir to simplify key management and remove the necessity of public key certificates [26]. To achieve this, the user's public key should be an information which can directly identify him in a non ambiguous way, such as e-mail address, IP address. The first practical identity based encryption scheme (IBE) was found by Boneh and Franklin in 2001 [4]. Using Fujisaki-Okamoto transformation [13], the IBE can be converted to IND-CCA2 secure under adaptive chosen identity attack.

3 Model for Universal Encrypted Deniable Authentication Protocol

In this section, we model the universal encrypted deniable authentication protocol. First, we define the protocol participants as follows.

Protocol Participants. We fix a nonempty set $\mathcal{P} = \{\mathcal{P}_0, \mathcal{P}_1, \cdots, \mathcal{P}_n\}$ of principals. Each principal $\mathcal{P}_i \in \mathcal{P}$ is a peer entity and named by the fixed length string $\mathcal{P}_i$. Assume that a secure public key encryption $\Pi = (\mathbf{KGen}, \mathbf{Enc}, \mathbf{Dec})$ is employed in the system, then each principal $\mathcal{P}_i$ is armed with a pair of public key and private key (pk_i, sk_i) conformed to Π.

Executing The Protocol. Formally, an encrypted deniable authentication protocol is just a probabilistic algorithm taking strings to strings. This algorithm determines how instances of the principals behave in response to messages from the system. We should take notice that each principal $\mathcal{P}_i$ may be running many instances. Then, we call each instance j of principal $\mathcal{P}_i$ as an oracle, and we denote it as $\Pi^j_{\mathcal{P}_i}$. In addition, we denote $\mathbf{sid}^j_{\mathcal{P}_i}$ as the session identifier of a particular instance $\Pi^j_{\mathcal{P}_i}$. If two session identifers $\mathbf{sid}^j_{\mathcal{P}_i}$ and $\mathbf{sid}^v_{\mathcal{P}_u}$ are equal, we say two instances $\Pi^j_{\mathcal{P}_i}$ and $\Pi^v_{\mathcal{P}_u}$ are *partnered.*

During the encrypted deniable authentication protocol, assume that two principals $\mathcal{P}_0$, $\mathcal{P}_1$ are *partnered*, who act as the sender and the intended receiver, respectively. An instance $\Pi^i_{\mathcal{P}_0}$ of $\mathcal{P}_0$ speaks first, producing some encrypted information on message m, Flow-1. The *parntered* instance $\Pi^j_{\mathcal{P}_1}$ of $\mathcal{P}_1$ responds with a message of its own, Flow-2, intended for $\mathcal{P}_0$'s instance $\Pi^i_{\mathcal{P}_0}$ which sent Flow-1. Finally, $\mathcal{P}_0$'s instance $\Pi^i_{\mathcal{P}_0}$ returns Flow-3, intended for Flow-2. As a result, the mutual authentication between $\mathcal{P}_0$ and $\mathcal{P}_1$ is executed, and the deniable authentication message m securely flows to the intended receiver $\mathcal{P}_1$.

Definition of Security. The definition of security for encrypted deniable authentication protocol requires

- **Deniablity:** Consider the following two games. In one game, a normal protocol execute is run between $\mathcal{P}_0$ and $\mathcal{P}_1$. The output of this game is the transcripts (Flow-1, Flow-2, Flow-3). In the other game, only the intended receiver $\mathcal{P}_1$ involves in the protocol, and the output is the transcripts (Flow-1', Flow-2', Flow-3'). By deniablity, (Flow-1', Flow-2', Flow-3') should be computationally indistinguishable from (Flow-1, Flow-2, Flow-3).
- **Confidentiality:** Consider the following game. Assume that $\mathcal{A}$ is an adversary, who is sitting in

[1] Non-malleability against adaptive chosen-ciphertext attacks (NM-CCA2) is another strongest security notions, which has been proved to be equivalent to IND-CCA2 in [3].

the middle of $\mathcal{P}_0$ and $\mathcal{P}_1$, and can freely intercept all transcripts between $\mathcal{P}_0$ and $\mathcal{P}_1$ in the system. The security level of confidentiality is relying on its employed public key encryption scheme Π. If Π is $\mathsf{G\text{-}ATK}$ secure, where $\mathsf{G} \in \{\mathsf{OW}, \mathsf{IND}\}$, $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA1}, \mathsf{CCA2}\}$, then the protocol is also $\mathsf{G\text{-}ATK}$ secure. For example, if the public key encryption scheme Π is $\mathsf{IND\text{-}CCA2}$ secure, then an adversary $\mathcal{A}$ intercepts all transcripts between $\mathcal{P}_0$ and $\mathcal{P}_1$ and queries the decryption oracle $\mathcal{O}_{dk}(\cdot)$ on some transcripts in the system. In the end, for some non-queried transcript Flow-1, which is encrypted one message m_b, $b \in \{0,1\}$, from two messages (m_0, m_1) chosen by $\mathcal{A}$, $\mathcal{A}$ guesses $b' \in \{0,1\}$. If $b' = b$, $\mathcal{A}$ wins the game. We define the advantage of $\mathcal{A}$ as $\mathbf{Adv}(\mathcal{A}) := 2\Pr[b = b'] - 1$. If $\mathbf{Adv}(\mathcal{A})$ is negligible, then the encrypted deniable authentication protocol satisfies the confidentiality.

- **Mutual Authentication:** After the transcripts (Flow-1, Flow-2, Flow-3) have successfully flowed, both $\mathcal{P}_0$ and $\mathcal{P}_1$ can authenticate each other, but can't prove the identity to a third party.

4 Universal Encrypted Deniable Authentication Protocol

In this section, we present our universal encrypted deniable authentication protocol UEDAP. The description of UEDAP is as follows.

Let $\Pi = (\mathbf{KGen}, \mathbf{Enc}, \mathbf{Dec})$ be a secure public key encryption scheme and $H(\cdot)$ be a cryptographic hash functions. Assume that the sender $\mathcal{P}_0$ wants to send a deniable authentication message m to an intended receiver $\mathcal{P}_1$, they execute the UEDAP by running the following steps. See Figure 1.

- $\mathcal{P}_0 \to \mathcal{P}_1$: $\mathcal{P}_0$ first chooses a random number r_0, then uses $\mathcal{P}_1$'s public key pk_1 to encrypt $\Pi(\mathcal{P}_0\|m\|r_0) = \mathbf{Enc}(pk_1, \mathcal{P}_0\|m\|r_0)$. In the end, $\mathcal{P}_0$ sends Flow-1=$[\Pi(\mathcal{P}_0\|m\|r_0)]$ to the intended receiver $\mathcal{P}_1$.
- $\mathcal{P}_1 \to \mathcal{P}_0$: After receiving Flow-1=$[\Pi(\mathcal{P}_0\|m\|r_0)]$, $\mathcal{P}_1$ uses his private key sk_1 to recover $\mathcal{P}_0\|m\|r_0 = \mathbf{Dec}(sk_1, \Pi(\mathcal{P}_0\|m\|r_0))$. Then, $\mathcal{P}_1$ chooses another random number r_1 and uses $\mathcal{P}_0$'s public key pk_0 to encrypt $\Pi(\mathcal{P}_1\|r_1\|r_0) = \mathbf{Enc}(pk_0, \mathcal{P}_1\|r_1\|r_0)$. In the end, $\mathcal{P}_1$ sends Flow-2=$[\Pi(\mathcal{P}_1\|r_1\|r_0)]$ to the sender $\mathcal{P}_0$.
- $\mathcal{P}_0 \to \mathcal{P}_1$: Upon receiving Flow-2=$[\Pi(\mathcal{P}_1\|r_1\|r_0)]$, $\mathcal{P}_0$ uses his private key sk_0 to recover $\mathcal{P}_1\|r_1\|r_0 = \mathbf{Dec}(sk_0, \Pi(\mathcal{P}_1\|r_1\|r_0))$. $\mathcal{P}_0$ checks the validity of r_0. If so, $\mathcal{P}_1$ is authenticated; otherwise terminates the protocol. Once $\mathcal{P}_1$ is valid, $\mathcal{P}_0$ computes $h = H(\mathcal{P}_0\|r_1)$ and sends Flow-3=$[h]$ to $\mathcal{P}_1$. $\mathcal{P}_1$ checks whether $h \overset{?}{=} H(\mathcal{P}_0\|r_1)$. If so, the deniable authentication message m is accepted; otherwise rejected.

5 Security and Efficiency

In this section, we mainly discuss the security of the universal encrypted deniable authentication protocol UEDAP and analyze its efficiency.

5.1 Security

Theorem 1. *The universal encrypted deniable authentication protocol UEDAP is deniable.*

Proof. To prove that the proposed UEDAP is deniable, we should show that the intended receiver $\mathcal{P}_1$ can create indistinguishable transcripts (Flow-1', Flow-2', Flow-3') himself.

TRANSCRIPTS SIMULATION: The intended receiver $\mathcal{P}_1$ can produce the transcripts (Flow-1', Flow-2', Flow-3') intended for himself, by performing the followings: Choose two random numbers r_0, r_1; compute Flow-1', Flow-2' and Flow-3' as

$$\begin{aligned}
\text{Flow-1'} &: \Pi(\mathcal{P}_0\|m\|r_0) = \mathbf{Enc}(pk_1, \mathcal{P}_0\|m\|r_0);\\
\text{Flow-2'} &: \Pi(\mathcal{P}_1\|r_1\|r_0) = \mathbf{Enc}(pk_0, \mathcal{P}_1\|r_1\|r_0);\\
\text{Flow-3'} &: h = H(\mathcal{P}_0\|r_1).
\end{aligned}$$

Clearly, (Flow-1', Flow-2', Flow-3') is indistinguishable from the transcripts (Flow-1, Flow-2, Flow-3) interactively generated by the sender $\mathcal{P}_0$ and the intended receiver $\mathcal{P}_1$. Therefore, the proposed UEDAP is deniable. □

Theorem 2. *The universal encrypted deniable authentication protocol UEDAP satisfies the confidentiality, provided that the employed public key encryption scheme* Π *is secure.*

Proof. In our proposed universal encrypted deniable authentication protocol UEDAP, the transcripts Flow-1 and Flow-2 are encrypted by using a $\mathsf{G\text{-}ATK}$ secure public key encryption scheme Π, where $\mathsf{G} \in \{\mathsf{OW}, \mathsf{IND}\}$, $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA1}, \mathsf{CCA2}\}$. Therefore, based upon the employed $\mathsf{G\text{-}ATK}$ secure public key encryption scheme Π, the corresponding security level of the confidentiality of the proposed UEDAP follows. □

Theorem 3. *The universal encrypted deniable authentication protocol UEDAP provides the mutual authentication.*

Proof. In our proposed UEDAP, since the public key encryption Π is secure, after $\mathcal{P}_0$ encrypts a random number r_0 to Flow-1, only the intended receiver $\mathcal{P}_1$ can know the random number r_0. Therefore, when $\mathcal{P}_1$ returns r_0 to $\mathcal{P}_0$ by using Flow-2, $\mathcal{P}_0$ can authenticate the intended receiver $\mathcal{P}_1$. For analogous reasons, $\mathcal{P}_1$ also can authenticate the sender $\mathcal{P}_0$. Therefore, the mutual authentication between $\mathcal{P}_0$ and $\mathcal{P}_1$ is provided. □

Sender $\mathcal{P}_0$ (pk_0, sk_0)		Receiver $\mathcal{P}_1$ (pk_1, sk_1)
nonce r_0		
$\Pi(\mathcal{P}_0\Vert m\Vert r_0) \leftarrow \mathbf{Enc}(pk_1, \mathcal{P}_0\Vert m\Vert r_0)$	$\xrightarrow{\Pi(\mathcal{P}_0\Vert m\Vert r_0)}$	
		$\mathcal{P}_0, m, r_0 \leftarrow \mathbf{Dec}(sk_1, \Pi(\mathcal{P}_0\Vert m\Vert r_0))$
		nonce r_1
	$\xleftarrow{\Pi(\mathcal{P}_1\Vert r_1\Vert r_0)}$	$\Pi(\mathcal{P}_1\Vert r_1\Vert r_0) \leftarrow \mathbf{Enc}(pk_0, \mathcal{P}_1\Vert r_1\Vert r_0)$
$\mathcal{P}_1, r_1, r_0 \leftarrow \mathbf{Dec}(sk_0, \Pi(\mathcal{P}_1\Vert r_1\Vert r_0))$		
check r_0		
$h = H(\mathcal{P}_0\Vert r_1)$	$\xrightarrow{h}$	
		check $h \overset{?}{=} H(\mathcal{P}_0\Vert r_1)$

Figure 1: Universal encrypted deniable authentication protocol

5.2 Efficiency

In this subsection, we briefly discuss the computation overhead and communication overhead in the proposed UEDAP. To estimate the computation overhead, the following notations are used: T_{E} denotes the required time for encryption operation; T_{D} denotes the required time for decryption operation; and T_{H} denotes the required time for hash function operation. Then, we can see the sender $\mathcal{P}_0$ requires $T_{\mathsf{E}}+T_{\mathsf{D}}+T_{\mathsf{H}}$, and the receiver $\mathcal{P}_1$ also requires $T_{\mathsf{E}}+T_{\mathsf{D}}+T_{\mathsf{H}}$. If the public key encryption scheme Π can be online/offline executed, then the efficiency can be improved by pre-computation. Assume that the length of ciphertext is $|\Pi(\cdot)|$ and the length of hash value is $|H(\cdot)|$. Then, the sender $\mathcal{P}_0$ totally sends $|\Pi(\cdot)| + |H(\cdot)|$ bits and the intended receiver sends $|\Pi(\cdot)|$ bits.

We summarise in Table 1 the detailed computation overhead and communication overhead of the proposed UEDAP. From the table, we can see that the efficiency of UEDAP relies on the employed public key encryption scheme Π.

6 Examples of UEDAP

In this section, we exemplify three universal encrypted deniable authentication protocol based on the RSA cryptosystem [25], the ElGamal cryptosystem [12] and the IBE [4].

6.1 RSA Cryptosystem Based Deniable Authentication Protocol

Initialization:

- Let $H(\cdot)$ be a cryptographic hash function. Given security parameter k, the sender $\mathcal{P}_0$ initializes his public key and private key pair $(pk_0 := (n_0, e_0), sk_0 := (n_0, p_0, q_0, d_0))$, where $n_0 = p_0 q_0$, $e_0 \cdot d_0 \equiv 1 \bmod (p_0 - 1)(q_0 - 1)$, $|p_0| = |q_0| = k/2$, and p_0, q_0 are Blum integers (i.e., $p_0, q_0 \equiv 3 \bmod 4$).
- The intended receiver $\mathcal{P}_1$ initializes his public key and private key pair $(pk_1 := (n_1, e_1), sk_1 := (n_1, p_1, q_1, d_1))$, where $n_1 = p_1 q_1$, $e_1 \cdot d_1 \equiv 1 \bmod (p_1 - 1)(q_1 - 1)$, $|p_1| = |q_1| = k/2$, and p_1, q_1 are Blum integers (i.e., $p_1, q_1 \equiv 3 \bmod 4$).

Protocol Running:

- $\mathcal{P}_0 \rightarrow \mathcal{P}_1$: $\mathcal{P}_0$ first chooses a random number r_0, then uses $\mathcal{P}_1$'s public key pk_1 to encrypt

  $$c_0 = (\mathcal{P}_0\|m\|r_0)^{e_1} \bmod n_1.$$

  In the end, $\mathcal{P}_0$ sends Flow-1=$[c_0]$ to the intended receiver $\mathcal{P}_1$. We assume that $\mathcal{P}_0||m||r_0$ is in $\mathbb{Z}_{n_1}$.

- $\mathcal{P}_1 \rightarrow \mathcal{P}_0$: After receiving Flow-1=$[c_0]$, $\mathcal{P}_1$ uses his private key sk_1 to recover

  $$c_0^{d_1} = (\mathcal{P}_0\|m\|r_0)^{e_1 d_1} = \mathcal{P}_0\|m\|r_0 \bmod n_1.$$

  Then, $\mathcal{P}_1$ chooses another random number r_1 and uses $\mathcal{P}_0$'s public key pk_0 to encrypt

  $$c_1 = (\mathcal{P}_1\|r_1\|r_0)^{e_0} \bmod n_0.$$

  In the end, $\mathcal{P}_1$ sends Flow-2=$[c_1]$ to the sender $\mathcal{P}_0$.

- $\mathcal{P}_0 \rightarrow \mathcal{P}_1$: Upon receiving Flow-2=$[c_1]$, $\mathcal{P}_0$ uses his private key sk_0 to recover

  $$c_1^{d_0} = (\mathcal{P}_1\|r_1\|r_0)^{e_0 d_0} = \mathcal{P}_1\|r_1\|r_0 \bmod n_0.$$

  $\mathcal{P}_0$ checks the validity of r_0. If so, $\mathcal{P}_1$ is authenticated; otherwise terminates the protocol. Once $\mathcal{P}_1$ is valid, $\mathcal{P}_0$ computes $h = H(\mathcal{P}_0\|r_1)$ and sends Flow-3=$[h]$ to $\mathcal{P}_1$. $\mathcal{P}_1$ checks whether

  $$h \overset{?}{=} H(\mathcal{P}_0\|r_1).$$

  If so, the deniable authentication message m is accepted; otherwise rejected.

Table 1: Computation overhead and communication overhead

	Computation overhead		Communication overhead	
	$\mathcal{P}_0$	$\mathcal{P}_1$	$\mathcal{P}_0$	$\mathcal{P}_1$
$\mathcal{P}_0 \rightarrow \mathcal{P}_1$	T_{E}		$\|\Pi(\cdot)\|$ bits	
$\mathcal{P}_1 \rightarrow \mathcal{P}_0$		$T_{\mathsf{D}} + T_{\mathsf{E}}$		$\|\Pi(\cdot)\|$ bits
$\mathcal{P}_0 \rightarrow \mathcal{P}_1$	$T_{\mathsf{D}} + T_{\mathsf{H}}$	T_{H}	$\|H(\cdot)\|$ bits	

6.2 ElGamal Cryptosystem Based Deniable Authentication Protocol

Initialization:

- Let $H(\cdot)$ be a cryptographic hash function. Given security parameter k, a trusted authority generates system parameters (p, g, q) where $|p| = k$, $q|p-1$, and $\# < g >= q$.
- The sender $\mathcal{P}_0$ initializes his public key and private key pair $(pk_0 := (y_0), sk_0 := (x_0))$, where $y_0 = g^{x_0} \bmod p$, $x_0 \in \mathbb{Z}_q^*$.
- The receiver $\mathcal{P}_1$ initializes his public key and private key pair $(pk_1 := (y_1), sk_1 := (x_1))$, where $y_1 = g^{x_1} \bmod p$, $x_1 \in \mathbb{Z}_q^*$.

Protocol Running:

- $\mathcal{P}_0 \rightarrow \mathcal{P}_1$: $\mathcal{P}_0$ first chooses two random number r_0, r_0', then uses $\mathcal{P}_1$'s public key pk_1 to encrypt

$$c_0 = \left\langle A_0 = g^{r_0'} \bmod p, B_0 = y_1^{r_0'} \cdot (\mathcal{P}_0 \| m \| r_0) \bmod p \right\rangle.$$

In the end, $\mathcal{P}_0$ sends Flow-1=$[c_0]$ to the intended receiver $\mathcal{P}_1$. We assume $\mathcal{P}_0||m||r_0$ is in group $\mathbb{Z}_p$.

- $\mathcal{P}_1 \rightarrow \mathcal{P}_0$: After receiving Flow-1=$[c_0]$, $\mathcal{P}_1$ uses his private key sk_1 to recover

$$\begin{aligned} \frac{B_0}{A_0^{x_1}} &= \frac{y_1^{r_0'} \cdot (\mathcal{P}_0 \| m \| r_0)}{g^{r_0' x_1}} \\ &= \frac{g^{r_0' x_1} \cdot (\mathcal{P}_0 \| m \| r_0)}{g^{r_0' x_1}} = \mathcal{P}_0 \| m \| r_0 \bmod p. \end{aligned}$$

Then, $\mathcal{P}_1$ chooses another random number r_1, r_1' and uses $\mathcal{P}_0$'s public key pk_0 to encrypt

$$c_1 = \left\langle A_1 = g^{r_1'} \bmod p, B_1 = y_0^{r_1'} \cdot (\mathcal{P}_1 \| r_1 \| r_0) \bmod p \right\rangle.$$

In the end, $\mathcal{P}_1$ sends Flow-2=$[c_1]$ to the sender $\mathcal{P}_0$. We assume $\mathcal{P}_1||r_1||r_0$ is in group $\mathbb{Z}_p$.

- $\mathcal{P}_0 \rightarrow \mathcal{P}_1$: Upon receiving Flow-2=$[c_1]$, $\mathcal{P}_0$ uses his private key sk_0 to recover

$$\begin{aligned} \frac{B_1}{A_1^{x_0}} &= \frac{y_0^{r_1'} \cdot (\mathcal{P}_1 \| r_1 \| r_0)}{g^{r_1' x_0}} \\ &= \frac{g^{r_1' x_0} \cdot (\mathcal{P}_1 \| r_1 \| r_0)}{g^{r_1' x_0}} = \mathcal{P}_1 \| r_1 \| r_0 \bmod p. \end{aligned}$$

$\mathcal{P}_0$ checks the validity of r_0. If so, $\mathcal{P}_1$ is authenticated; otherwise terminates the protocol. Once $\mathcal{P}_1$ is valid, $\mathcal{P}_0$ computes $h = H(\mathcal{P}_0 \| r_1)$ and sends Flow-3=$[h]$ to $\mathcal{P}_1$. $\mathcal{P}_1$ checks whether

$$h \overset{?}{=} H(\mathcal{P}_0 \| r_1).$$

If so, the deniable authentication message m is accepted; otherwise rejected.

6.3 IBE Based Deniable Authentication Protocol

Initialization:

- Let $H(\cdot)$ be a cryptographic hash function. Given security parameters k, l, the PKG chooses groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q > 2^k$, a generator P of $\mathbb{G}_1$, a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and hash functions $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $\mathcal{G} : \mathbb{G}_2 \rightarrow \{0,1\}^l$. It chooses a master secret $s \in_R \mathbb{Z}_q^*$ and computes $P_{pub} = sP \in \mathbb{G}_1$ that is made public.
- By the PKG, the sender $\mathcal{P}_0$ obtains his public key and private key pair $(pk_0 := \mathcal{P}_0, sk_0 := s\mathcal{H}(\mathcal{P}_0))$. Similarly, the receiver $\mathcal{P}_1$ obtains his public key and private key pair $(pk_1 := \mathcal{P}_1, sk_1 := s\mathcal{H}(\mathcal{P}_1))$.

Protocol Running:

- $\mathcal{P}_0 \rightarrow \mathcal{P}_1$: $\mathcal{P}_0$ first chooses two random number r_0, r_0', then uses $\mathcal{P}_1$'s public key pk_1 to encrypt

$$c_0 = \left\langle \begin{array}{l} A_0 = r_0' P, \\ B_0 = \mathcal{G}(e(P_{pub}, \mathcal{H}(\mathcal{P}_1))^{r_0'}) \oplus (\mathcal{P}_0 \| m \| r_0) \end{array} \right\rangle.$$

In the end, $\mathcal{P}_0$ sends Flow-1=$[c_0]$ to the intended receiver $\mathcal{P}_1$. We assume $\mathcal{P}_0||m||r_0$ is in Z_{2^l}.

- $\mathcal{P}_1 \rightarrow \mathcal{P}_0$: After receiving Flow-1=$[c_0]$, $\mathcal{P}_1$ uses his private key sk_1 to recover

$$B_0 \oplus \mathcal{G}(e(A_0, s\mathcal{H}(\mathcal{P}_1))) = \mathcal{P}_0 \| m \| r_0.$$

Then, $\mathcal{P}_1$ chooses another random number r_1, r_1' and uses $\mathcal{P}_0$'s public key pk_0 to encrypt

$$c_1 = \left\langle \begin{array}{l} A_1 = r_1' P, \\ B_1 = \mathcal{G}(e(P_{pub}, \mathcal{H}(\mathcal{P}_0))^{r_1'}) \oplus (\mathcal{P}_1 \| r_1 \| r_0) \end{array} \right\rangle.$$

In the end, $\mathcal{P}_1$ sends Flow-2=$[c_1]$ to the sender $\mathcal{P}_0$.

- $\mathcal{P}_0 \to \mathcal{P}_1$: Upon receiving Flow-2=$[c_1]$, $\mathcal{P}_0$ uses his private key sk_0 to recover

$$B_1 \oplus \mathcal{G}(e(A_1, s\mathcal{H}(\mathcal{P}_0))) = \mathcal{P}_1 \| r_1 \| r_0.$$

$\mathcal{P}_0$ checks the validity of r_0. If so, $\mathcal{P}_1$ is authenticated; otherwise terminates the protocol. Once $\mathcal{P}_1$ is valid, $\mathcal{P}_0$ computes $h = H(\mathcal{P}_0 \| r_1)$ and sends Flow-3=$[h]$ to $\mathcal{P}_1$. $\mathcal{P}_1$ checks whether

$$h \overset{?}{=} H(\mathcal{P}_0 \| r_1).$$

If so, the deniable authentication message m is accepted; otherwise rejected.

Remark. Using OAEP technique [6] or the Fujisaki-Okamoto transformation [13], the above three protocols can be easily converted to IND-CCA2 secure.

7 Conclusions

A deniable authentication protocol allows a sender to transfer an authenticated message to an intended receiver in such a way the intended receiver can identify the source of a message without being able to prove the identity of the sender to a third party. In this paper, we investigate the construction of unverisal encrypted deniable authentication protocol from *any* public key encryption scheme, such as RSA [25], ElGamal [12], IBE [4]. By considering both security and efficiency, we conclude that the universal encrypted deniable authentication protocol is practical.

Acknowledgements

The author would like to thank his former PhD student, Rongxing Lu, for his discussions. This work is supported in part by the National NaturalScience Foundation of China under Grant Nos. 60572155, 60773086, the National Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20060248008, and the National 863 Project of China under Grant No. 2006AA01Z424.

References

[1] Y. Aumann, and M. O. Rabin, "Authentication, enhanced security and error correcting codes," *Proceedings of Crypto '98*, LNCS 1462, pp. 299-303, Springer-Verleg, 1998.

[2] Y. Aumann, and M. O. Rabin, "Efficient deniable authentication of long messages," *International Conference on Theoretical Computer Science in honor of Professor Manuel Blums 60th birthday, 1998*, http://www.cs.cityu.edu.hk/dept/video.html.

[3] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes," *Advances in Cryptology - Crypto '98*, LNCS 1462, pp. 26-45, Springer Verlag, Berlin , 1998.

[4] D. Boneh, and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM. Journal of Computing*, vol. 32, no. 3, pp. 586-615, 2003. Extended abstract *Advances in Crptology-Crypto '01*, LNCS 2139, pp. 213-229, Berlin, Springer-Verlag, 2001.

[5] M. Bellare, and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," *First ACM Conference on Computer and Communications Security*, pp. 62-73, Fairfax, ACM, 1993.

[6] M. Bellare, and P. Rogaway, "Optimal asymmetric encryption - How to encrypt with RSA," *Advance in Cryptology - Eurocrypt '94*, LNCS 950, pp. 92-111, Springer-Verlag, Berlin, 1995.

[7] T. Cao, D. Lin, and R. Xue, "An efficient ID-based deniable authentication protocol from pairings," *19th International Conference on Advanced Information Networking and Applications, 2005*, vol. 1, pp. 388-399, Mar. 2005,

[8] R. Cramer, and V. Shoup, "A Practical public key cryptosystem provably secure against adaptive chosen ciphertext attacks," *Advances in Cryptology - Crypto'98*, LNCS 1462, pp. 13-25, Springer-Verlag, 1998.

[9] W. Diffie, and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644-654, 1976.

[10] X. Deng, C. H. Lee, and H. Zhu, "Deniable authentication protocols," *IEE Proceedings of Computer Digital Technology*, vol. 148, no. 2, pp. 101-104, Mar. 2001.

[11] C. Dwork, M. Naor, and A. Sahai, "Concurrent zero-knowledge," *Proceedings of 30th ACM Symposium on Theory of Computing*, pp. 409-418, 1998.

[12] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithm," *IEEE Transactions on Information Theory*, vol. IT-31, no. 4, pp. 469-472, Jul. 1985.

[13] E. Fujisaki, and T. Okamoto, "Secure integeration of asymmetric and symmetric encryption schemes," *Advances in Cryptology - Crypto '99*, LNCS 1666, pp. 537-554, Berlin, Springer-Verlag, 1999.

[14] L. Fan, C. Xu, and J. Li, "Deniable authentication protocol based on Diffie-Hellman algorithm," *Electronics Letters*, vol. 38, no. 4, pp. 705-706, 2002.

[15] S. Goldwasser, and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, pp. 270-299, 1984.

[16] R. Lu, and Z. Cao, "Non-interactive deniable authentication protocol based on factoring," *Computer Standards and Interfaces*, vol. 27, pp. 401-405, 2005.

[17] R. Lu, and Z. Cao, "A new deniable authentication protocol from bilinear pairings," *Applied Mathematics and Computation*, vol. 168, pp. 954-961, 2005.

[18] R. Lu, Z. Cao, X. Dong, and R. Su, "Group oriented deniable authentication protocol," *First International Multi-Symposiums on Computer and Computational Sciences*, vol. 2. pp. 89-92, 2006.

[19] R. Lu, Z. Cao, S. Wang, and H. Bao, "A new id-based deniable authentication protocol," *Informatica*, vol. 18, no. 1, pp. 67-78, 2007.

[20] J. Manuel, G. Nieto, C. Boyd, and E. Dawson, "A public key cryptosystem based on a subgroup membership problem," *Designs, Codes and Cryptography*, vol. 36, pp. 301-316, 2005.

[21] M. Naor, and M. Yung, "Public-key cryptosystem provably secure against chosen ciphertext attacks," *Proceedings of the 22nd STOC*, pp.427-437, ACM Press, New York, 1990.

[22] T. Okamoto, and D. Pointcheval, "REACT: Rapid enhanced-security asymetric cryptosystem transform," *CT-RSA '2001*, LNCS 2020, pp.159-175, Springer-Verlag, Berlin, 2001.

[23] H. Qian, Z. Cao, L. Wang, and Q. Xue, "Efficient non-interactive deniable authentication protocols," *The Fifth International Conference on Computer and Information Technology*, pp. 673-679, 2005.

[24] C. Rackoff, and D. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," *Advances in Cryptology - CRYPTO '91*, LNCS 576, pp.433-444, Springer-Verlag, 1992.

[25] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystem," *Communications of the ACM*, vol. 21, pp. 120-126, 1978.

[26] A. Shamir, "Identity-based cryptosystems and signature schemes," *Advances in Cryptology - Crypto' 84*, LNCS 196, pp. 47-53, Berlin, Springer-Verlag, 1984.

[27] Z. Shao, "Efficient deniable authentication protocol based on generalized ElGamal signature scheme," *Computer Standards and Interfaces*, vol. 26, pp. 449-454, 2004.

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

[29] Y. Shi, and J. Li, "Identity-based deniable authentication protocol," *Electronics Letters*, vol. 41, no. 5, pp. 241-242, 2005.

[30] 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, 2005.

[31] R. W. Zhu, D. S. Wong, and C. H. Lee, "Cryptanalysis of a suite of deniable authentication protocols," *IEEE Communications Letters*, vol. 10, no. 6, pp. 504-506, 2006.

Zhenfu Cao received his B.S. degree in computer science and technology from Harbin Institute of Technology, China, in 1983, and his Ph.D. degree in mathematics from the same university. Currently, he is a professor and a doctoral supervisor in the Department of Computer Science and Engineering of Shanghai Jiao Tong University. His main research areas are number theory, modern cryptography, information security and trusted computing etc.. Now, he is the director of Trusted Digital Technology Laboratory of Shanghai Jiao Tong University (http://tdt.sjtu.edu.cn).