

Identity-based Cryptography for Admissible and Anonymous Communication

Extended Version

Dijiang Huang
Computer Science and Engineering
Arizona State University
Tempe, AZ 85287-8809 USA

Abstract

Key management for anonymous communication in mobile ad-hoc network is a critical but unsolved problem. Many current anonymous mobile ad-hoc routing protocols assume that mobile users share pairwise secrets before they start an anonymous communication session. This assumption is impractical for many scenarios where pairwise shared keys are difficult to set up in advance. Public-key based solution, such as identity-based cryptographic solutions have been proposed for anonymous communications. This approach assumes that a centralized trust authority is in charge of the private key generation. Using this approach, the anonymous communications are not blind to the trust authority.

To solve above discussed anonymity problem by using identity-based cryptography, we present a pairing-based encryption and certificate scheme. Our approach provides the following properties compared to traditional approaches: (1) the user's *id* (i.e., a pseudonym) can be used as his public key (this is the same as traditional identity-based solution); however, each anonymous user can self-derive his private key based on a set of publicly known system parameters and his chosen pseudonym (this is the difference from the traditional identity-based solution); (2) the self-chosen pseudonym can be blindly signed by a CA or an anonymous group leader and thus only the pseudonym with a verifiable certificate (verified by using a set of publicly known system parameters) can be used during the anonymous communications. Our approach reduces the key management complexity and it is suitable for large scale and ad-hoc anonymous services.

1 Introduction

In wireless environment, such as Ad-hoc networks, preserving privacy is a difficult task. Due to the broadcasting feature of wireless networks, if transmitted data is not encrypted, adversaries can sit anywhere and eavesdrop all transmitted data. Even though the transmitted data is encrypted, the underlying protocol can still expose the wireless users' identities, such as a unique IP address or MAC address. If we hide the packet's header information in both MAC layer and network layer, every wireless user needs to decrypt all the received broadcasting packets and to check the IP payload to see if he can decrypt the packet and then discover the receiver. Though this method

is desirable for the purpose of anonymity, it will involve too much cryptographic computational overhead.

Recent research [9] proposed to use *identity-based encryption* (IBE) [4] for anonymous communication. In their approach, a unique *trust authority* (TA) administrates an *anonymous communication group* (ACG) in broadcasting wireless environment. The TA's duties include admission control of the ACG and key distribution for ACG members. ACG members use each other's identity (i.e., a pseudonym) as the public key to set up anonymous communication sessions¹. Before starting an ACG session, the anonymous participants (APs) are required to register at the TA in order to derive their private keys. They self-generate a set of pseudonyms and submit them to TA, and then the TA generates corresponding private keys and sends them back to the APs. Note that all these operations are processed offline or via secure channels. In this way, the APs' identities are protected by using pseudonyms. Within an anonymous communication session, it requires that the APs' pseudonyms must be transmitted in plaintext. Using this method, the key management is easier. This is because that a pseudonym serves for two purposes simultaneously: (a) public key and (b) identity. However, the drawback of this approach is obvious; that is the underlying anonymous communications are not blind to the TA.

Based on above discussions, we note that the challenge arises between admissibility and anonymity. On one hand, we want to protect users' identities; on the other hand, we expect to create a manageable and admissible communication environment for anonymous users. *The challenge is that if we can achieve both admissibility and anonymity for wireless ad-hoc network applications by using identity-based cryptography.*

The challenge can be illustrated as follows. It is desirable to have a controller who is in charge of the admission control of an ACG. To admit an AP, we need to fulfill the following requirements: (1) each AP must rely on a set of publicly known system parameters to self-generate his pseudonym (a public key) and then self-derive corresponding private key; (2) each AP must rely on the ACG controller to derive his pseudonym certificate (to get the admission to the ACG, the AP must use traditional authentication method; thus the admission is not blind to the controller.); (3) however, the controller cannot disclose the contents of the pseudonym and corresponding private key; and (4) the anonymous user cannot replicate new pseudonym and corresponding certificates based on his pseudonym and derived certificate from the ACG controller.

In this paper, we present such a scheme that fulfill above mentioned requirements ((1) to (4)). We propose a pairing-based cryptographic solution to construct an Anonymity-Based Cryptography (ABC). Our approach provides the protection to user's identity as well as the basic cryptographic functions, such as encryption and signatures (certificates). Under the formal PKI assumption, the proposed ABC works for the following scenario: *a certificate authority (CA) issues the certificates for anonymous users; a certificate is a pseudonym signed by the CA; based on the certificate, the anonymous user can be verified if he possesses a valid certificate, i.e., only the anonymous user with a verifiable signature by using the certificate authority's public key is a valid participant in the anonymous group.*

We can relax the restriction that each AP must rely on a wellknown CA to issue his certificate. Then, we assume that each AP can self-generate anonymous communication parameters and organize his own ACGs. This feature is very attractive to ad-hoc organized anonymous group.

¹In all our following discussions, without special notation, we use the terms "pseudonym" and "public key" interchangeably

In the following sections, we propose an admissible pairing-based solution for anonymous applications. Our solution is especially useful in the wireless broadcasting environments, in which each AP uses the broadcasting network and hardware addresses to blind the sender and receiver. As a result, the user's pseudonym is the only method to identify an anonymous user. Our pairing-based approach is suitable for anonymous applications from small to large scale.

The rest of the paper is organized as follows: In Section 2, the system models and mathematical background of our approach are presented; In Section 3, we propose the anonymity-based encryption scheme; In Section 4, a certificate scheme for anonymity-based cryptography is presented; In section 5, we present the existing challenges for anonymity-based cryptography; Finally, in Section 6, we summarize our work.

2 Backgrounds and System Models

According to [2], the construction of a workable and provably secure *identity-based encryption* (IBE) scheme was, until recently, an open problem dating back to Shamir's 1984 paper [8]. Two solutions appeared in rapid succession in early 2001 – the pairing-based approach proposed by Boneh and Franklin [3] and Cocks' scheme based on the Quadratic Residuosity problem [6]. Our approach is based on IBE scheme in [4] and BLS signature scheme in [5]. In this section, we present the system models and the preliminary mathematical backgrounds of pairing.

2.1 Anonymous System Models

An ACG is formed by multiple APs. The AP's identity serves as his public key as well as his pseudonym. During an anonymous communication, the AP A broadcasts his pseudonym PD_A . Other APs can encrypt a message by using A 's pseudonym PD_A as the message-encrypting key. Thus, the AP A will be able to use his private key to decrypt the ciphertext. In above described setting, the ACG can be arranged in a self-managed manner or arbitrarily controlled by a trusted authority. For ad-hoc anonymous communication environment, it is desirable that APs can self-manage their ACG without restrictions by a fixed infrastructure, such as the restriction to have a online trust authority or a offline pre-registered certificate authority. In this scenario, an AP can be elected as the *ACG leader* (AL) and the AL determines the publicly known parameters (denoted by *params*) for his anonymous communication group. The AL can issue the certificates for the ACG members in the self-and-ad-hoc-managed ACG. There are two requirements imposed on above described anonymous system. These requirements ensure the anonymous users to achieve both anonymity and security.

1. The AP can self-generate his pseudonym and corresponding private key. However, the pseudonym must have a valid certificate derived from the AL. The certificate is verifiable by using the *params* published by the AL.
2. The AL admits an AP based on traditional authentication methods, such as a certificate generated by a CA or a correct password. Once passing the authentication, the AL blindly generates the certificate for the AP's pseudonym without disclosing the pseudonym and its signature (a certificate).
3. The AP cannot replicate new pseudonyms (with private keys) and corresponding certificates based on the known pseudonym and certificate pair.

Table 1: System parameters: *params*

$\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_3$	Mapping from addition group \mathbb{G}_1 to multiplicative group \mathbb{G}_3
n	Bit length of plaintext
$P \in \mathbb{G}_1$	$P \in E(\mathbb{F}_q^*)$, P is the generator of \mathbb{G}_1
$Q_0 = [s]P$	Public key of the ACG, $s \in \mathbb{Z}_{\delta_P}^*$ is only known by the <i>params</i> generator
$\delta_P = \text{ord}(P)$	Order of point P
H	$\mathbb{G}_1 \times \mathbb{G}_3 \rightarrow \mathbb{G}_1$
H_2	$\mathbb{G}_3 \rightarrow \{0, 1\}^n$
H_3	$\{0, 1\}^{2n} \rightarrow \mathbb{Z}_{\delta_P}^*$
H_4	$\{0, 1\}^n \rightarrow \{0, 1\}^n$

2.2 Pairing

The known mathematical models of pairings (see [2] Chapter IX) – the Weil and Tate pairings– involve fairly complex mathematics. Fortunately, they can be dealt with abstractly, using only the group structure and mapping properties. Many interesting schemes have been built based purely on abstract bilinear maps.

The major pairing-based construction is the bilinear map. Considering two groups \mathbb{G}_1 and \mathbb{G}_3 , we denote \mathbb{G}_1 using additive notation and \mathbb{G}_3 using multiplicative notation. The bilinear mapping can be denote by $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_3$ and the mapping have three properties:

Bilinearity:

$$\hat{e}([a]P, [b]Q) = \hat{e}(P, Q)^{ab}, \quad \forall P, Q \in \mathbb{G}_1, \forall a, b \in \mathbb{Z}_{\delta_P}^*.$$

Non-Degeneracy: if $\hat{e}(P, Q) = 1$ for all $Q \in \mathbb{G}_1$, then P must be the identity element in \mathbb{G}_1 .

Computability: the bilinear map \hat{e} is efficiently computable.

2.3 Multiplicative mask

The order of P (denoted by $\delta_P = \text{ord}(P)$ and $P \in \mathbb{G}_1$) is the smallest positive integer such that $[\delta_P]P = \mathcal{O}$, where \mathcal{O} is the point of infinity. The multiplicative mask property is described as follows: let $\delta_P = \text{ord}(P)$, if r is a unit in the multiplicative group $\mathbb{Z}_{\delta_P}^*$ and r^{-1} is the inverse, then $k = r(r^{-1}k) \pmod{\delta_P}$ for any $k \in \mathbb{Z}_{\delta_P}^*$. This means that calling $k' = r^{-1}k \pmod{\delta_P}$, for any group element P , we can recover the desired points $[k]P$ by first computing $Q' = [k']P$ and then $[k]P = [r]Q'$. Let $\delta_{\mathbb{G}_1}$ be the group order. We know $\text{ord}(P)$ divides $\text{ord}(\mathbb{G}_1)$, thus, $k' = r^{-1}k \pmod{\text{ord}(\mathbb{G}_1)} \Rightarrow k' = r^{-1}k \pmod{\text{ord}(P)}$.

2.4 System Parameters – *params*

A set of system parameters, denoted as *params*, is publicly known by all anonymous users. There are many ways to publish the *params*. For example, it can be published on some trusted website, and thus every anonymous user can download it. Some publicly well-known trusted party can generate a certificate for the *params*. Thus, during the anonymous communication, the certificate

can be broadcasted and every anonymous user can verify the *params*. In this paper, the system parameters are represented as:

$$params = \langle \mathbb{G}_1, \mathbb{G}_3, \hat{e}, n, P, Q_0, \delta_P, H, H_2, H_3, H_4 \rangle.$$

The detailed explanation of *params* is shown in Table 1.

In traditional IBE scheme [4], the function H_1 is a random oracle which is used to map a identity from $\{0, 1\}^n$ to a point in \mathbb{G}_1 . However in anonymous service, the identity is a pseudonym, which is not necessarily to be meaningful. An anonymous user can randomly select a point Q_A in the group \mathbb{G}_1 as his pseudonym. Thus, we remove the H_1 from *params*. Instead, we introduce another random function H , which maps a point in \mathbb{G}_1 and a value in \mathbb{G}_3 to a random point in the group \mathbb{G}_1 .

2.5 Identity-based Encryption

Boneh and Franklin proposed an Identity-based encryption (IBE) scheme (or **FullIdent** scheme) in [4], which provides strong security guarantees. Their scheme can be related to the hardness of Bilinear Diffie-Hellman Problem (BDH problem, see Appendix B) in a model that naturally extends the widely-accepted adaptive chosen cipher attack (IND-CCA2) model [1] for public-key encryption to the identity-based setting. We present the detailed **FullIdent** algorithms of the IBE scheme in Appendix A.

2.6 Identity-based Signature Scheme

In [5], Boneh, Lynn and Shacham used pairings to construct a signature scheme (noted as BLS). Our certificate approach is based on the BLS scheme. As usual, we work with system parameters *params* and assume P of prime order δ_P generates \mathbb{G}_1 . The TA's private key is a value $s \in \mathbb{Z}_{\delta_P}$, and the matching public key is $[s]P \in \mathbb{G}_1$. The signature on a message $M \in \{0, 1\}^*$ is simply $\sigma = [s]H_1(M) \in \mathbb{G}_1$ (the hash function $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1$ is required in the BLS signature scheme but not in our scheme). To verify a purported signature σ on message M , the verifier checks that the 4-tuple: $\langle P, [s]P, H_1(M), \sigma \rangle$ is a Diffie-Hellman tuple. This can be done by checking that the equation:

$$\hat{e}(\sigma, P) = \hat{e}(H_1(M), [s]P)$$

3 Anonymity-Based Encryption

In this section, we describe the proposed Anonymity-Based Encryption (ABE) scheme in details.

3.1 ABE with Trusted PKG

To enable an AP to self-generate his pseudonym and corresponding private key without relaying on the PKG, we can simply remove the $H_1 : \{0, 1\}^n \rightarrow \mathbb{G}_1$ function in the original IBE scheme (see IBE scheme in Appendix A). In IBE scheme, the hash function H_1 maps an identity ID_A to a point $Q_A = H_1(ID_A)$ in group \mathbb{G}_1 . As we know, the point P is the generator of group \mathbb{G}_1 , i.e., any point in group \mathbb{G}_1 can be represented as $[k']P$, where $k' \in \mathbb{Z}_{\delta_P}$. Due to the hash function H_1 , the adversary cannot derive the value k' (to find k' with the known points P and $[k']P$ is equivalent to

solve the ECDLP problem which is shown in Appendix B). If we remove the H_1 operation from the IBE scheme, an AP can randomly select a value $k \in \mathbb{Z}_{\delta_P}$ and then compute point $[k]P$ as his pseudonym. Based on the publicly known system parameter $[s]P$, the AP can self-derive the private key $[ks]P$ by simply multiplying the point $[s]P$ with the value k . In this way, the AP can self-generate a valid pseudonym and private key pair $\langle [k]P, [sk]P \rangle$.

Using above presented modified IBE scheme, the AP can generate pseudonym and private key pairs without restrictions. However, using above approach, the drawback is that the APs must trust the PKG. Although there is no need to rely on the PKG to generate the private key, the underlying anonymous communications are not blind to the PKG. This is because that the PKG can simply multiply $[k]P$ by s to derive the private key $[sk]P$.

3.2 ABE with Un-trusted PKG

The same as it is specified in IBE scheme (see Appendix A), the ABE scheme includes four steps:

Setup, Extract, Encryption, and Decryption. The ABE scheme is represented as follows:

Setup: System parameters $params = \langle \mathbb{G}_1, \mathbb{G}_3, \hat{e}, n, P, Q_0, \delta_P, H, H_2, H_3, H_4 \rangle$ is published². The description of $params$ is shown in Table 1.

Extract: An anonymous user does follows:

1. Randomly select two numbers $k_1, k_2 \in \mathbb{Z}_{\delta_P}^*$.
2. Compute k_2^{-1} , where $1 \equiv k_2 k_2^{-1} \pmod{\delta_P}$.
3. Compute points $Q_A = [k_1 k_2]P$ and $Q'_A = [k_2^{-1} - 1]Q_0$.
4. Compute pairing $c = \hat{e}(Q_A, Q'_A)$.
5. Output tuple $PD_A = \langle Q_A, c \rangle$ as the pseudonym.
6. Compute $S_{PD_A} = [sk_1]P$ as the private key.

Encrypt: To encrypt the plaintext $M \in \{0, 1\}^n$ for entity A with pseudonym PD_A , perform the following steps:

1. Choose a random $r \in \{0, 1\}^n$.
2. Set $t = H_3(r, M)$.
3. Compute and output the ciphertext:

$$C = \langle [t]P, r \oplus H_2(\hat{e}(Q_A, Q_0)^t \cdot c^t), M \oplus H_4(r) \rangle \in \mathbb{G}_1 \times \{0, 1\}^{2n}.$$

Decrypt: Suppose $C = \langle U, V, W \rangle \in \mathbb{G}_1 \times \{0, 1\}^{2n}$ is a ciphertext encrypted for A . To decrypt C using the private key S_{ID_A} :

1. Compute $r' := V \oplus H_2(\hat{e}(S_{PD_A}, U))$.
2. Compute $M' := W \oplus H_4(r')$.

²The hash function H is not used in ABE scheme. its usage will be described in our certificate scheme in the Section 4.

3. Set $t' = H_3(r', M')$ and test if $U = [t']P$. If not, reject the ciphertext.
4. Otherwise, output M' as the decryption of C .

The ABE scheme is a modified **FullIdent** scheme of IBE scheme [4]. However, they are fundamentally different. We describe their differences as follows:

- First, the basic constructions of ABE scheme and IBE scheme are different. In ABE scheme, there is no PKG. All anonymous users use a set of publicly known *params*. Even the master secret s can be publicly known (only if no certificate is required). The point Q_A is indeed a masked point and the masker is c . The private key is only known to the user himself. Thus, on one except the AP himself is able to decrypt the ciphertext and knows the real identity. In the IBE scheme, all users rely on a trusted third party (i.e., a PKG) to generate their private keys. Thus, the underlying communications are not blind to the PKG.
- Second, the methods to generate an identity (or a pseudonym) are different. In ABE scheme, since the pseudonym can be a random string, it is not necessarily to be meaningful. In IBE scheme, the user's identity is publicly known and it should be unique, such as the user's email address. The conversion from an identity to a point on the curve is one-way, such as $H_1 : \{0, 1\}^n \rightarrow \mathbb{G}_1$.
- Third, the ABE scheme is more scalable and flexible in terms of the private key generation. In ABE scheme, the anonymous user computes his private key. It is not necessarily to contact a trusted third party to start an anonymous conversation and there is no need to derive a private key from the PKG in order to decrypt a ciphertext. Thus, the ABE scheme is suitable for large-scale anonymous system. Whereas in the IBE scheme, the PKG generates the user's private key and the private key must be securely delivered to the user.
- Fourth, the IBE scheme is relatively computational efficient. The ABE scheme introduces several additional operations: one inverse operation, three point multiplications (i.e., the computations of $[k_1]P$, $[k_1k_2]P$, and $[k_2^{-1} - 1]P$), and one pairing operation in the **Extract** algorithm. The compensation is the reduction of one mapping operation (by using H_1 in IBE scheme). In **Encryption** algorithm, one multiplication operation in \mathbb{G}_3 is added. Since the **Extract** algorithm is only performed for the pseudonym and private key generation, the computational overhead due to one multiplication operation in the **Encryption** algorithm is not significant.

3.3 Analysis of ABE

Here, we analyze the ABE scheme presented in Section 3.2. The ABE scheme uses two random numbers to determine a pseudonym, i.e., k_1 and k_2 . In addition, the anonymous user needs to compute the inverse of k_2 , where $1 \equiv k_2k_2^{-1} \pmod{\delta_P}$. Note that both k_1 and k_2 are secretly reserved by the anonymous user. The private key of the anonymous user is $S_{PD_A} = [k_1]Q_0 = [sk_1]P$. The $c = \hat{e}(Q_A, Q'_A) = \hat{e}([k_1k_2]P, [k_2^{-1} - 1]Q_0)$ is a masker. It plays a crucial role in ABE scheme. c masks the real pseudonym $[k_1]P$ of the anonymous user. In this way, the receiver will be able to use the private key $[sk_1]P$ to decrypt the ciphertext. In other words, the ABE scheme is a masked version of IBE scheme and it is equivalent to IBE scheme that uses the point $[k_1]P$ as the public key and the point $[sk_1]P$ as the private key without exposing the pseudonym $[k_1]P$ to the ciphertext.

sender. Thus, even if the adversary knows the master secret s , she cannot derive the private key $[sk_1]P$.

To see how it works, we demonstrate the correctness of the H_2 operation in the **Encrypt** algorithm:

$$\begin{aligned}
H_2(\hat{e}(Q_A, Q_0)^t \cdot c^t) &= H_2(\hat{e}(Q_A, Q_0)^t \cdot \hat{e}(Q_A, Q'_A)^t) \\
&= H_2(\hat{e}(Q_A, Q_0 + Q'_A)^t) \\
&= H_2(\hat{e}(Q_A, [k_2^{-1}]Q_0)^t) \\
&= H_2(\hat{e}([sk_1 k_2 k_2^{-1}]P, [t]P)) \\
&= H_2(\hat{e}(S_{PD_A}, U))
\end{aligned}$$

The proposed ABE scheme is based on the **FullIdent** scheme proposed by Boneh and Franklin in [4]. The **FullIdent** scheme is obtained from the basic scheme (also presented in [4]) by applying the Fujisaki-Okamoto hybridization techniques [7]. To analyze the security of **FullIdent**, Boneh and Branklin constructed an **IND-ID-CCA Security Game**, which is presented in Appendix C.

To prove our ABE scheme is secure, we first assume the IBE scheme is secure, and then we present how to securely convert ABE scheme to IBE scheme.

Theorem 1 *The ABE scheme is also against **IND-ID-CCA** attack.*

Proof of Theorem 1: In [4], the authors prove the IBE scheme is against adaptive chosen ciphertext attack in the random oracle model under the BDH (presented in Appendix B) assumption. The chosen ciphertext attack allows the adversary to access private keys of arbitrary entities (except the challenge identity, of course) as well as giving the adversary access to a decryption oracle. This attack model also applies to our ABE scheme. The difference is that in ABE scheme the challenger and the adversary can be the same entity (see chosen ciphertext attack in Appendix C). Based on above analysis, we argue that the attack models of IBE scheme and ABE scheme are the same.

The security proof for ABE scheme is subtle. We assume that the IBE scheme is secure due to the proof presented in [4]. To prove the proposed ABE scheme is secure, we evaluate the security due to the conversion from ABE scheme to IBE scheme. Our strategy is to prove the modifications introduced by ABE scheme will not affect the security of the original IBE scheme. The detailed IBE scheme is presented in Appendix A. To summarize, two differences exist between ABE scheme and IBE scheme:

1. In ABE scheme, $Q_A = [k_1 k_2]P$ is publicly known, where $k_1, k_2 \in \mathbb{Z}_{\delta_P}^*$. The private key is computed based on the masked pseudonym $[k_1]P$. To find $[k_1]P$, (in other words, to find k_2^{-1} by given P and $[k_1 k_2]P$), it is at least as hard as to solve ECDLP problem (see ECDLP definition in Appendix B), which is considered to be a hard problem.
2. In ABE scheme, the masker $c = \hat{e}(Q_A, Q'_A)$ is given, where $Q'_A = [k_2^{-1} - 1]Q_0$. To find Q_A and Q'_A and satisfy $c = \hat{e}(Q_A, Q'_A)$ is believed to be a DBDH problem (see Appendix B for the definition of the DBDH problem). Even if the adversary \mathcal{A} derives the point Q'_A , to find $k_2^{-1} - 1$ by given Q_0 and $[k_2^{-1} - 1]Q_0$ is equivalent to solve the ECDLP problem. Note that both DBDH and ECDLP problems are hard problems.

It is proved in [4] that IBE scheme is chosen ciphertext attack secure in the Random Oracle model, provided that there is no polynomially bounded algorithm having a non-negligible advantage

in solving the BDH problem. Based on above analysis, we claim that the modifications and the introduced parameters by ABE scheme will not affect the security of IBE algorithms in [4]. Thus, the ABE scheme is secure.

4 Certificate Scheme for Anonymity-Based Cryptography

In this section, we present an anonymity-based signature (ABS) scheme that can be used to create certificates for ACG members. The ABS scheme is designed based on the BLS scheme proposed in [5].

It is desirable that an ACG organizer, such as an AL, is able to grant the admission to ACG members. An easy way to do this is to generate certificates for the pseudonyms that are self-generated by the APs. In this way, during the anonymous communications, a pseudonym can be validated by verifying its certificate, i.e., only the pseudonym with valid certificate will be used during the anonymous communication sessions. Due to the anonymity feature of the ACG, several requirements of our certificate scheme are given as follows:

1. Each AP can self-generate his pseudonym and corresponding private key.
2. The AL is responsible for the certificate generation. However, both the pseudonym and the corresponding private key are blind to the AL
3. The AP cannot generate new and valid certificate based on the certificate already derived from the AL.
4. Other APs can verify the pseudonym based on the provided certificate and publicly known *params*.

To provide such a certificate scheme, we propose an anonymity-based signature scheme that is based on the BLS scheme presented in [5]. Our scheme includes four steps: **KeyGen**, **Sign**, **Recover**, and **Verify**. The ABS scheme for AOCs model is presented as follows:

KeyGen: $params = \langle \mathbb{G}_1, \mathbb{G}_3, \hat{e}, n, P, Q_0, \delta_P, H, H_2, H_3, H_4 \rangle$ is published. The description of $params$ is shown in Table 1 (not all parameters are used in ABS scheme). The AP generates a masked point $Q'_B = [\ell]H(PD_A)$, where $\ell \in \mathbb{Z}_{\delta_P}$ is randomly selected and $1 \equiv \ell\ell^{-1} \pmod{\delta_P}$. AP then sends Q'_B to the AL.

Sign: The AL computes the signature $\sigma' = [s]Q'_B$, then sends it to the AP.

Recover: On receiving the σ' , the AP recovers the signature by computing $\sigma = [\ell^{-1}]\sigma'$, where $1 \equiv \ell\ell^{-1} \pmod{\delta_P}$. Thus, the σ is the signature of PD_A .

Verify: To verify the signature, the AP performs the following test:

$$\hat{e}(H(PD_A), Q_0) = \hat{e}(\sigma, P),$$

In ABS scheme, the AP generates the hash value of PD_A . This will allow the AP to mask the point Q_B . Thus, the AL will not know the real pseudonym of the AP and the corresponding signature σ .

4.1 Analysis of ABS

To verify a signature, the ABS scheme tests two pairing operations. To see how it works, we demonstrate the correctness of the testing operations in the **Verify** algorithm as follows:

$$\begin{aligned} \hat{e}(\sigma, P) &= \hat{e}([s]Q_B, P) \\ &= \hat{e}(Q_B, [s]P) \\ &= \hat{e}(H(PD_A), Q_0) \end{aligned}$$

Our ABS scheme is based on the BLS scheme proposed by Boneh et al. in [5]. In [5], the authors prove that the BLS scheme is securely against existential forgery under adaptive chosen message attack in the random oracle model assuming **CDH** problem (presented in Appendix B) is hard in \mathbb{G}_1 . Based on the proof of BLS scheme, we present the security of ABS scheme as follows.

Theorem 2 *The ABS scheme is securely against forgery under adaptive chosen-message attack.*

Proof of Theorem 2: To prove the our ABS scheme is secure, we first assume that the BLS scheme is secure, and then we present how to securely convert ABS scheme to BLS scheme. In BLS scheme, the user's identity ID_A is mapped to the point $Q_A = H_1(ID_A) = [k']P$ in \mathbb{G}_1 by using a random function $H_1 : \{0, 1\}^n \rightarrow \mathbb{G}_1$. H_1 prevents the adversary from determining k' by knowing the point Q_A . To solve k' is believed to solve **ECDLP** problem, which is a hard problem. Thus the adversary can not forge the signature $[k's]P$ for arbitrarily selected user identity ID_A by knowing the public key $[s]P$. The ABS scheme introduces the random mapping (one-way) function $H : \mathbb{G}_1 \times \mathbb{G}_3 \rightarrow \mathbb{G}_1$ in the **Verify** algorithm, in which the function H serves the same purpose of H_1 in BLS scheme. Although the adversary knows the public key $[s]P$ and a point Q_A in \mathbb{G}_1 , she cannot forge the signature $[s]H(Q_A, c) = [sk']P$ without knowing the k' . Thus, the ABS scheme is another form of BLS scheme with different parameter setting and assumptions.

Based on above analysis, the ABS scheme is a modified version of BLS scheme and the ABS scheme does not change the security strength of BLS scheme. Thus, it is securely against forgery under adaptive chosen-message attack.

5 Challenges of Anonymity-Based Cryptography

Our ABE and ABS schemes ensure both security and anonymity for ACG members. However, the ABS scheme allows a certificate to be used only for one pseudonym. For anonymous communications, the AP will change his pseudonym frequently to prevent the adversaries from identifying his involved anonymous sessions. Thus, it is highly desired: (1) a certificate can be used for multiple pseudonyms; (2) the change of pseudonyms is traceable, i.e., both and only the anonymous communication peers can trace the change of the peer's pseudonyms (in this way, the anonymous communication peers will not lost the tracks of established anonymous communication sessions); and (3) the security requirement for conditions (1) and (2) is that the adversary cannot forge certificates for known pseudonyms, in other words, the adversary cannot impersonate the APs.

Above discussed traceable pseudonym and certificate feature is very useful for anonymity applications. We expect that further and deeper investigations in pairing-based cryptography will enable us to achieve these goals.

6 Conclusion

In this paper, we propose an anonymity-based encryption (ABE) scheme and anonymity-based signature (ABS) schemes for anonymous communications. Using ABE scheme, an anonymous user can self-generate his pseudonym and corresponding private key based on a set of publicly known system parameters. During the anonymous communication, a pseudonym uniquely identifies an anonymous user and serves as his/her public key. The ABE scheme is an anonymous version of IBE scheme. For anonymous communication, the ABE scheme is more flexible and scalable since no PKG is required and it is more secure due to the self-generated private key. In addition to ABE scheme, we also proposed ABS schemes for anonymous communications. Together with ABE scheme, the ABS schemes ensure the admissibility for an anonymous communication group. Only the pseudonym with a valid certificate is admissible to the anonymous communication group.

References

- [1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. *Advances in Cryptology -CRYPTO '98*, pages 26–45. Springer-Verlag, LNCS 1462, 1998.
- [2] I. F. Blake, G. Seroussi, and N. P. Smart. *Advances in Elliptic Curve Cryptography*. Cambridge University Press, London mathematical Society Lecture Note Series 317, 2005.
- [3] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Proceedings of the CRYPTO 01, Springer-Verlag*, 2001.
- [4] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. *SIAM J. of Computing*, (3):586–615, 2003.
- [5] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In *Proceedings of the Asiacrypt 2001, volume 2248 of LNCS*, pages 514–532, 2001.
- [6] C. Cocks. *An Identity Based Encryption Scheme Based on Quadratic Residues*. Springer-Verlag, 2001.
- [7] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Proceedings of the CRYPTO 99, Springer-Verlag*, pages 537–554, 1999.
- [8] A. Shamir. Identity-based cryptosystems and signature schemes. In *In Proceedings of Crypto '84, LNCS Volumn 196*, pages 47–53, 1985.
- [9] Y. Zhang, W. Liu, and W. Lou. Anonymous communications in mobile ad hoc networks. In *Proceedings of IEEE Information Communications Conference (INFOCOM)*, March 2005.

Appendix

A Identity-based Encryption

The following algorithms are presented as **FullIdent** algorithms in [4].

Setup: $params = \langle \mathbb{G}_1, \mathbb{G}_3, \hat{e}, n, P, Q_0, H_1, H_2, H_3, H_4 \rangle$. The description of $params$ is shown in Table 1.

Extract: This algorithm takes as input an identity string ID and returns the corresponding private key $[s]H_1(ID)$.

Encrypt: To encrypt the plaintext $M \in \{0, 1\}^n$ for entity A with identity ID_A , perform the following steps:

1. Compute $Q_A = H_1(ID_A) \in \mathbb{G}_1$.
2. Choose a random $\sigma \in \{0, 1\}^n$.
3. Set $t = H_3(\sigma, M)$.
4. Compute and output the ciphertext:
 $C = \langle [t]P, \sigma \oplus H_2(\hat{e}(Q_A, Q_0)^t), M \oplus H_4(\sigma) \rangle \in \mathbb{G}_1 \times \{0, 1\}^{2n}$.

Decrypt: Suppose $C = \langle U, V, W \rangle \in \mathbb{G}_1 \times \{0, 1\}^{2n}$ is a ciphertext encrypted for A . To decrypt C using the private key $[s]Q_A$:

1. Compute $\sigma' := V \oplus H_2(\hat{e}([s]Q_A, U))$.
2. Compute $M' := W \oplus H_4(\sigma')$.
3. Set $t' = H_3(\sigma', M')$ and test if $U = [t']P$. If not, reject the ciphertext.
4. Otherwise, output M' as the decryption of C .

B Some Hard Problems

Elliptic Curve Discrete Logarithm Problem (ECDLP problem): given P and $[m]P$ in \mathbb{G}_1 with $m \in \mathbb{Z}_{\delta_P}^*$, compute m .

Computational Diffie-Hellman Problem (CDH Problem): given P , $[a]P$, and $[b]P$ in \mathbb{G}_1 with $a, b \in \mathbb{Z}_{\delta_P}^*$, compute $[ab]P$.

Bilinear Diffie-Hellman Problem (BDH Problem): given P , $P_1 = [a]P$, $P_2 = [b]P$ and $P_3 = [c]P$ in \mathbb{G}_1 with a, b , and c selected uniformly at random from $\mathbb{Z}_{\delta_P}^*$, compute $\hat{e}(P, P)^{abc}$.

Decisional Bilinear Diffie-Hellman Problem (DBDH Problem): given P , $P_1 = [a]P$ and $P_2 = [b]P$ in \mathbb{G}_1 with a and b selected uniformly at random from $\mathbb{Z}_{\delta_P}^*$, and $\ell \in \mathbb{G}_3$, output **Yes** if $\ell = \hat{e}(P, P)^{ab}$ and output **No** otherwise.

C IND-ID-CCA Security Game

The attacker \mathcal{A} plays against a challenger \mathcal{C} in the following game:

IND-ID-CCA Security Game: The game runs in five steps:

Setup: \mathcal{C} runs algorithm **Setup** on input some value ℓ , gives \mathcal{A} the system parameters $params$ and keeps the master secret s to itself.

Phase 1: \mathcal{A} issues a series of queries, each of which is either an **Extract** query on an identity, in which case \mathcal{C} responds with the appropriate private key, or a **Decrypt** query on an identity/ciphertext combination, in which case \mathcal{C} responds with an appropriate plaintext (or possibly a fail message).

Challenge: Once \mathcal{A} decides to end Phase 1, it selects two plaintexts M_0, M_1 , and an identity ID_{ch} on which it wishes to be challenged. We insist that ID_{ch} not be the subject of an earlier **Extract** query. Challenger \mathcal{C} then chooses b at random from $\{0, 1\}$ and runs algorithm **Encrypt** on M_b and ID_{ch} to obtain the challenge ciphertext C^* ; \mathcal{C} then gives C^* to \mathcal{A} .

Phase 2: \mathcal{A} issues another series of queries as in Phase 1, with the restriction that no **Extract** query be on ID_{ch} and that no **Decrypt** query be on the combination $\langle ID_{ch}, C^* \rangle$. \mathcal{C} responds to these as before.

Guess: Finally, \mathcal{A} outputs a guess b' and wins the game if $b' = b$.