New Public Key Authentication Frameworks with Lite Certification Authority

Xiaolei Dong, Licheng Wang, and Zhenfu Cao*

Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200030, P. R. China
{xldong,wanglc,zfcao}@sjtu.edu.cn

Abstract. Two variants of CA-based public key authentication framework are proposed in this paper. The one is termed as public key cryptosystem without certificate management center (PKCwCMC) and the other is termed as proxy signature based authentication framework (PS-based AF). Moreover, we give an implementation of the former based on quadratic residue theory and an implementation of the latter from RSA. Both of the two variants can be looked as lite-CA based authentication frameworks since the workload and deployment of CAs in these systems are much lighter and easier than those of in the traditional CA-based PKC.

1 Introduction

In the usual sense of public-key cryptography (PKC), a key generation procedure invariantly contains the following step:

$$public\_key = \mathcal{F}(secret\_key) \tag{1}$$

Here, $\mathcal{F}$ is some efficient and one-way function which maps from the private key space to the public-key space [Mao04]. To use PKC in real-world applications, we need a mechanism which enables a ready verification of the association between a public key and a principal's identity. Such a mechanism is usually realized in an authentication framework (AF): it enables the owner of a public key to authenticate toward the system [Mao04]. At present, the mainly developed authentication frameworks can be divided into three categories: CA-based, ID-based and some middle courses, such as self-certified, certificated-based, as well as the so-called certificateless PKC.

1.1 CA-based PKC and Certificate Management Center

The term of "public key certificate" is used firstly by Kohnfelder [Koh78]. In general, a public key certificate is a structured data record with a number of data entries which include a uniquely identifiable identity of the holder and

* Corresponding author. Tel.: +86-21-62835602

her/his public key parameter. Certificate is digitally signed by a certification authority (CA) who is a special principal and trusted directly by all principals in the domain it serves. The CA's signature of a certificate provides a cryptographic binding between the holder's identity and her/his public key[Mao04]. Before issuing a certificate, CA should validate the identity of the applicant. The validation should of course involve some physical (i.e., non-cryptographic) procedures. The applicant should also prove that she/he knows the private component of the public key to be certified. The proof can either be in the form of[Mao04]:

- A signature of a challenge message verifiable using the public key
- A zero-knowledge proof protocol between the user and the CA
- Some application requires the private key

The certificate management center (CMC) is a set of components, which are responsible for the management of infrastructure supporting certificates, including policies, softwares, procedures, revocation, storage, distribution, and so forth.

1.2 ID-Based PKC and Key Generator Center

In 1984, Shamir [Sha84] introduced the concept of identity-based (ID-based) cryptosystems where a user's private key is generated by a trusted private key generator (PKG) and the user's public key could be easily derived from his identity by any party. ID-based cryptosystems, which simplify the key and certificate management procedure of CA-based PKI, are good alternatives for CA-based systems, especially when efficient key management and moderate security are required.

However, there are three main problems which incur a lot of complaint against to apply ID-based cryptosystems to open and large range networks with high security requirements:

- Inherent key escrow problem.
- Inefficiency for pairing computation.
- Private key losing problem and the corresponding identifier revocation problem.

1.3 Trust Levels and Other Authentication Framework of Public Key Cryptography

At EuroCrypt'91, Mirault [Gir91] suggested that the public key cryptosystem can be classified into three trust levels according to the trust assumption of the TTP (Trusted Third Party):

- Level 1: The TTP knows the users' private keys and therefore can impersonate any user at any time without being detected.

– Level 2: The TTP does not know the users' private keys. Nevertheless, the TTP can still impersonate a user by generating a false public key (or a false certificate).
– Level 3: The TTP does not know the users' private keys and the frauds of the authority are detectable. More precisely, a public-key scheme will be said of level 3 if the authority cannot compute users' secret keys, and if it can be proven that it generates false guarantees of users if it does so.

> *"Clearly, the level 3 is the most desirable one, and is achieved by CA-based schemes. Indeed only the authority is able to produce certificates. As a consequence, the existence of two (or more) different certificates for the same user is in itself a proof that the authority has cheated."* [Gir91]

However, ID-based schemes only achieves trust level 1, since the PKG knows all the secret keys held by the users and these users must have every confidence in it. This may be highly insufficient in some applications [Gir91].

Therefore, Girault [Gir91] proposed self-certified cryptosystems in the earlier of 1990s. Recently, two other authentication frameworks of public key cryptography have been proposed: One is the concept of certificate-based encryption proposed by Gentry [Gen03]; Another is the certificateless cryptosystems proposed by Al-Riyami and Paterson [AP03]. All of their proposals are instructive and interesting. To some extent, these solutions are neither CA-based nor identity-based, while inherent some merits from both sides of CA-based and ID-based cryptosystems.

However, none of these solutions is too perfect to be criticizable. One the one hand, we hold a different opinion on the needless of authentication for public keys in these schemes. On the other hand, none of them achieves trust level 3 in a perfect mode. Just as Girault's suggestion, it would be significant to design schemes which are not CA-based, while they still achieve level 3 [Gir91].

1.4 Our Contribution and Organization

Two variants of CA-based public key authentication framework are proposed in this paper: The one removes the centralized certificate management center — so it is termed as public key cryptosystem without certificate management center (PKCwCMC); The other further alleviates CA's burden by employing a proxy-protected signature pattern which enables the principals can authenticated their public keys by themselves. Both of the two variants achieve the highest trust level, i.e., level 3. We call these two variants as lite-CA based authentication frameworks since the workload and deployment of CAs in these systems are much lighter and easier than those of in the traditional CA-based PKC.

The rest of the paper is organized as follows: In section 2, we introduce some necessary preliminaries for the following designing; In section 3, we propose a public key cryptosystem without certificate management center based on quadratic residue problems; Further analysis and discussion on the first type of lite-CA based authentication framework are presented in section 4; In section 5, we propose the second type of lite-CA based systems based on proxy-protected signature pattern. Finally, concluding remarks are made in section 6.

2 Preliminaries: The Dependent Quadratic Residue Problems

Similar to the dependent RSA problems defined in [Po99], the dependent quadratic residue problem can be defined by the following steps.

Definition 1 (Dependent Quadratic Residue Problems). *Let N be the product of two unknown secure Blum primes and $P \in \mathbb{Z}_N$. We can define a function $f : \mathbb{Z}_N^* \to \mathbb{Z}_N^* \times \{0,1\}^2$ as $f(x) = (c, d_1, d_2)$ where $c = x(x+P)$* mod N *and*

$$\begin{cases} d_1 = 0, x < \frac{N}{2} \\ d_1 = 1, otherwise \end{cases} \quad and \quad \begin{cases} d_2 = 0, \left(\frac{x}{N}\right) = -1 \\ d_2 = 1, otherwise \end{cases} \tag{2}$$

Then the dependent quadratic residue problem (DQRP) is: Given $f(a) = (c, d_1, d_2) \in \mathbb{Z}_N^* \times \{0,1\}^2$ for some unknown a, find $f(a+1)$. For convenience, the problems is denoted by DQRP(N, P, f).

Apparently, in the above definition, f is an one-way function under the factoring assumption. Moreover, we have

Theorem 1. *The dependent quadratic residue problem (DQRP) is intractable if and only if the computational square root problem (SRP) is intractable.*

Proof. At first, suppose one can solve SRP problem effectively. Then, he can obtain a from $f(a)$. Thus, he can compute $f(a+1)$ easily.

Then, suppose one can work out DQRP problem, i.e., he can get $f(a+1)$ from $f(a)$. Let $f(a) = (c, d_1, d_2)$, $f(a+1) = (c', d_1', d_2')$. Then he can also extract a by $a \equiv (c' - c - 1 - P)/2 \mod N$. □

We denote $Succ(\mathcal{A})$ the success probability of an adversary $\mathcal{A}$ for solving the DQRP problems, i.e.,

$$Succ(\mathcal{A}) = \Pr[\mathcal{A}(f(a)) = f(a+1) | a \xleftarrow{R} \mathbb{Z}_N^*]. \tag{3}$$

The *DQRP assumption* says that for any p.p.t. adversary $\mathcal{A}$, the maximal $Succ(\mathcal{A})$ is negligible w.r.t. the systems' security parameter, i.e., the binary length of N. Since the square root problem (SRP) is hard enough to be used as a cryptography assumption [Will80]. Therefore, according the above theorem, the DQRP assumption is also a useful building block in designing of cryptosystems.

Note that the decisional version of dependent quadratic residue problem is not intractable. This situation is much different from the decisional dependent RSA (D-DRSA) problem in [Po99].

3 Public Key Cryptosystems without Certificate Management Center

In the original security model of certificateless PKC[AP03], there are two kinds of adversaries being taken into consideration. Type I adversaries can amount public key substitution attacks while they can not access to PKG's private key. Type II adversaries can access to PKG's private key while they can not amount public key substitution attacks. Since in CL-PKC, public keys are no longer CA's signature, the verification on users' public keys is meaningless. However, the convenience of certificateless is at the cost of losing some robustness of the cryptosystems. For example, if Bob's public key has been substituted, while Alice needs to send Bob an encrypted message, which is very important and urgent. What would happen in this situation? Therefore, also enlightened by the thought of needless of certificate management center in CL-PKC, we remove the management center of CA-based PKC but keep the explicit authentication process, and then obtain another authentication framework, dominated as public key encryption without certificate management center, referring to PKEwCMC for abbreviation.

3.1 Security Model of PKEwCMC

Definition 2. *A public key encryption without certificate management center is a 6-ary tuple $\Pi = (\mathcal{G_{CA}}, \mathcal{G_U}, \mathcal{E_P}, \mathcal{S_P}, \mathcal{E}, \mathcal{D})$ defined as follows:*

- `CA-Setup`, *$\mathcal{G_{CA}}$, is a probabilistic polynomial time (ppt) algorithm that takes as input k, the system's security parameter, and outputs the public/private keys pair (pk_{CA}, sk_{CA}).*
- `User-Setup`, *$\mathcal{G_U}$, is also a ppt algorithm that takes as input k, the system's security parameter, and outputs the public/private keys pair (pk_U, sk_U).*
- `Extract-Partial-Public-Key`, *$\mathcal{E_P}$, is also a ppt algorithm that takes the system's security parameter k, CA's private key sk_{CA} and the user's public key pk_U and identity ID_U as input and outputs P_U as partial public key.*
- `Set-Public-Key`, *$\mathcal{S_P}$, is a deterministic algorithm that takes the system's security parameter k, CA's public key pk_{CA} and the user's public key pk_U and identity ID_U as input, and outputs (pk_U, P_U) as the user's extended public key if P_U is a valid partial public key.*
- `Encrypt`, *$\mathcal{E}$, is also a ppt algorithm that takes a plaintext $M \in \mathcal{M}$, pk_U, P_U and pk_{CA} as input and outputs a ciphertext $C \in \mathcal{C}$ or $\perp$ which means that pk_U is invalid.*
- `Decrypt`, *$\mathcal{D}$, is a deterministic algorithm that takes a ciphertext $C \in C$ and a private key sk_U as input and outputs the corresponding plaintext $M \in M$ or $\perp$ which means that C is not a valid ciphertext.*

Among the above definition, the algorithms $\mathcal{G_{CA}}$ and $\mathcal{E_P}$ are executed by CA while the algorithms $\mathcal{G_U}$, $\mathcal{S_P}$ and $\mathcal{D}$ are executed by the user itself. Of course, the algorithm $\mathcal{E}$ is executed by who wants to send ciphertext to the user.

Since in the above scheme, partial public key P_U is, in essential, CA's signature on user's public key pk_U and identity ID_U, no adversary except CA can substitute user's public key pk_U of partial public key P_U without being detected. Therefore, we assume that the adversary can never amount public key substitution attacks. Similar to the classical security notion of encryptions[GM84], three types attacks — chosen plaintext attack (cpa), chosen ciphertext attack (cca1) and adaptive chosen ciphertext attack (cca2), should be taken into consideration.

Definition 3. *Let $\Pi = (\mathcal{G}_{\mathcal{CA}}, \mathcal{G}_{\mathcal{U}}, \mathcal{E}_{\mathcal{P}}, \mathcal{E}, \mathcal{D})$ be a public key encryption without certificate management center, and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a ppt adversary. For $atk \in \{cpa, cca1, cca2\}$ and $1^k \in \mathbb{N}$ let*

$$Adv_{\mathcal{A},\Pi}^{ind-atk}(1^k) = \left| \Pr\left[\begin{array}{c} (pk_{CA}, sk_{CA}) \leftarrow \mathcal{G}_{\mathcal{CA}}(1^k); \\ (pk_U, sk_U) \leftarrow \mathcal{G}_{\mathcal{U}}(1^k); \\ P_U \leftarrow \mathcal{E}_{\mathcal{P}}(1^k, sk_{CA}, pk_U, ID_U); \\ (m_0, m_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk_U, sk_{CA}, pk_{CA}); \\ b \leftarrow \{0,1\}; \\ c^* \leftarrow \mathcal{E}(m_b, pk_U, P_U, pk_{CA}) : \\ \mathcal{A}_2^{\mathcal{O}_2}(m_0, m_1, c^*) = b \end{array} \right] - \frac{1}{2} \right| \tag{4}$$

where $|\cdot|$ is the absolute-value function and

$$\begin{cases} \mathcal{O}_1 = \epsilon \text{ and } \mathcal{O}_2 = \epsilon, & if\ atk = cpa, \\ \mathcal{O}_1 = \mathcal{D}_{sk_U}(\cdot) \text{ and } \mathcal{O}_2 = \epsilon, & if\ atk = cca1, \\ \mathcal{O}_1 = \mathcal{D}_{sk_U}(\cdot) \text{ and } \mathcal{O}_2 = \mathcal{D}_{sk_U}(\cdot), & if\ atk = cca2. \end{cases} \tag{5}$$

We insist that $\mathcal{A}_1$ outputs m_0 and m_1 with the same length. Also, $\mathcal{A}_2$ is not permitted to make the query $\mathcal{O}_2(c^*)$.

The encryption scheme Π is ind-atk secure if $Adv_{\mathcal{A},\Pi}^{ind-atk}(1^k)$ is negligible. The advantage function of the scheme is also defined by a similar way:

$$Adv_{\mathcal{A},\Pi}^{ind-atk}(1^k, t, q_d) = \max\{Adv_{\mathcal{A},\Pi}^{ind-atk}(1^k)\} \tag{6}$$

where the maximum is taken over all adversaries that run for time t and make at most q_d queries to the decryption oracle.

3.2 New Encryption Scheme from Quadratic Residues

Now, it is time to describe a concrete implementation of PKEwCMC scheme. Our construction is based on quadratic residue theory. The main idea comes from [DC06]. The new encryption scheme consists of six algorithms as follows:

– `CA-Setup`: This algorithm takes security parameter k as input and returns a secure RSA modulus and parameters $N = p \cdot q$, e and d, where p and q are two large secure primes and d the inverse of e modula $\varphi(N) = (p-1)(q-1)$. Usually, this algorithm is run by the third trusted party, referred as CA in our scheme. The pair $< N, e >$ and the tuple $< p, q, d >$ should be looked as CA's public key and private key, respectively. Suppose $H : \{0,1\}^* \times \{0,1\}^{2k} \rightarrow \{0,1\}^{2k}$ and $H_1 : Z_N \rightarrow Z_N$ be two working cryptographic hash functions[1].

- `User-Setup`: This algorithm takes security parameter k and an identifer for entity $A, ID_A \in \{0,1\}^*$, as input and returns another secure RSA modulus and parameters $N_A = p_A \cdot q_A$, where p_A and q_A are two secure Blum primes (i.e., $p_A \equiv q_A \equiv 3 \pmod 4$). Usually, This algorithm is run by a user with the identifier ID_A. The pair $< p_A, q_A >$ should be looked as A's private key, while the pair $< ID_A, N_A >$ should be sent to CA for registration.
- `Extract-Partial-Public-Key`: After receiving $< ID_A, N_A >$ from a user with identifer ID_A, CA extracts A's partial public key $P_A = H(ID_A, N_A)^d \mod N$ and sends it back to A via the public channel.
- `Set-Public-Key`: After receiving P_A from CA, the user A validates it by checking whether $P_A^e \equiv H(ID_A, N_A) \pmod N$ holds. If not, the user A broadcasts a "Complaint Message" against CA; otherwise, A publishes the tuple $< ID_A, N_A, P_A >$ as its public key.
- `Encryption`: Suppose the plaintext be M, any entity who want to send the ciphertext $C = E(M)$ to the user A does the following steps:
  - Validate P_A by checking whether $P_A^e \equiv H(ID_A, N_A) \pmod N$ holds. If not, send to A a "Alert Message: Your public key has been juggled!" and abort.
  - Pick a random number $k \in \mathbb{Z}_{N_A}^*$, compute $c_1 = k(k + P_A) \mod N_A$ and send $C = (c_1, c_2; d_1, d_2)$ to A via the public channel, where $c_2 = M \oplus H_1((k+1)(k+1+P_A) \mod N_A)$ and

$$\begin{cases} d_1 = 0, k < \frac{N_A}{2} \\ d_1 = 1, otherwise \end{cases} \quad \text{and} \quad \begin{cases} d_2 = 0, \left(\frac{k}{N_A}\right) = -1 \\ d_2 = 1, otherwise \end{cases} \tag{7}$$

- `Decryption`: Taking the ciphertext $C = (c_1, c_2; d_1, d_2)$ and the private key $< p_A, q_A >$ as input, the user A can extract the corresponding plaintext M as follows:
  - Pre-computation. Let $a = \frac{P_A^2}{4} \mod N_A$ and $b = \frac{N_A - p_A - q_A + 5}{8} \mod \varphi(N_A)$. Employing Chinese Remainder Theorem to the equation $x^2 \equiv 1 \pmod{N_A}$, obtain four quadratic roots of 1, denoted by ± 1 and $\pm i$ respectively.
  - Let $d = c_1 + a \mod N_A$ and $r = d^b \mod N_A$. Among four possible numbers $\pm r$ and $\pm i \cdot r$, only one, denoted by k', satisfies

$$\begin{cases} k' < \frac{N_A}{2}, d_1 = 0 \\ k' \geq \frac{N_A}{2}, d_1 = 1 \end{cases} \quad \text{and} \quad \begin{cases} \left(\frac{k'}{N_A}\right) = -1, d_2 = 0 \\ \left(\frac{k'}{N_A}\right) = 1, \quad d_2 = 1 \end{cases} \tag{8}$$

  Now, let $k = k' - \frac{P_A}{2} \mod N_A$. Then, $M = c_2 \oplus H_1((k+1)(k+1+P_A))$ is just the desired plaintext.

[1] Meanwhile, H_1 should be modelled as a random oracle.

3.3 Correctness and Security

Theorem 2 (Correctness). *The proposed scheme in section 3.2 is correct.*

Proof. Since

$$\left(k + \frac{P_A}{2}\right)^2 \equiv k^2 + k \cdot P + \frac{P_A^2}{4} \equiv c_1 + \frac{P_A^2}{4} \equiv c_1 + a \equiv d \pmod{N_A} \tag{9}$$

and

$$r^2 \equiv d^{\frac{N_A - p_A - q_A + 5}{4}} \equiv d^{\frac{\varphi(N_A)+4}{4}} \equiv d^{\frac{\varphi(N_A)}{4}} \cdot d \equiv d \pmod{N_A}, \tag{10}$$

the proposed scheme is consistent. □

Theorem 3 (IND-CPA). *The encryption scheme defined above is secure against chosen plaintext attacks under the assumption that the dependent quadratic residue problem (DQRP) is intractable.*

Proof. The proof is very similar to the theorem 9 in [Po99]. □

The above scheme, denoted by `BasicIdent`, achieves only CPA security. For practical applications, an improved one, denoted by `FullIdent`, which achieves CCA or CCA2 security, is much more desired. On the one hand, we know that in [FO99], Fujisaki and Okamoto described a method to transform a cryptosystem with IND-CPA security into one with IND-CCA2 security. The method uses a hash function which is modelled as a random oracle in the security analysis. Thus, it is easy to extend our scheme from IND-CPA secure to IND-CCA2 secure by employing their method. On the other hand, in [Po99], Pointcheval gave a more systematic way to construct encryption scheme with CCA2 security under the dependent RSA (DRSA) assumption. Enlightened by Pointcheval's idea, we also describe the `FullIdent` scheme, here. The main modification made to the `BasicIdent` lie in the following two aspects:

- During the encryption process, let $C = E(M) = (c_1, c_2, c_3; d_1, d_2)$ where c_1, d_1 and d_2 are worked out just as those in the `BasicIdent` scheme, while $c_2 = m \oplus H_1(g(k+1))$ and $c_3 = H_2(m, k)$. Here, the function g is defined as $g(x) = x(x + P_A) \mod N_A$ while $H_1(\cdot)$ and $H_2(\cdot, \cdot)$ are two new introduced hash functions which should be modelled as random oracles. Both g and $H_i (i = 1, 2)$ can be viewed as parts of system's public parameters.
- During the decryption process, when the salt number k is worked out by the same way defined in the `BasicIdent`, $m = c_2 \oplus H_1(g(k + 1))$ is an accepted plaintext if and only if $c_3 = H_2(m, k)$ holds.

Theorem 4 (IND-CCA2). *The `FullIdent` encryption scheme is semantically secure against adaptive chosen-ciphertext attacks under the assumption that the dependent quadratic residue problem (DQRP) is intractable.*

Proof. The proof is very similar to the theorem 11 in [Po99]. □

4 Further Analysis of the Proposed Scheme

4.1 Efficiency Comparisions

Comparing with CL-PKE schemes in [AP03] and [AP05], our proposal has the following advantages:

- Higher Trust Level. In our proposed PKE scheme, the entity A itself is totally in charge of private key generation. Therefore, just as traditional CA-based PKE, our scheme reaches trust level 3.
- More robust. In our scheme, if the public key has been substituted by some adversaries, except PKG, it is detectable before the ciphertext has been sent. Thus, if an entity A receives a ciphertext, it can extract corresponding plaintext without fail. However, in CL-PKE schemes, without necessary authentication for public keys, adversaries can make trouble in A's decryption process as aforemention.
- More efficient. The proposed scheme in section 3.2 is more efficient than the original CL-PKE scheme in [AP03] and improved scheme in [AP05]. Table 1 shows the comparison of our proposal and the schemes in [AP03] and [AP05]. The computational cost of the following operations are measured in our comparison:
  - `M`: The operation taking the form of $a \cdot b \mod N$ and $\left(\frac{a}{N}\right)$ for $a, b \in Z_N^*$ or $p \cdot q$ for large primes p and q;
  - `E`: The operation taking the form of $a^b \mod N$ for $a, b \in Z_N^*$ or g^r for $g \in G_2^*$ and $r \in Z_q^*$;
  - `P`: The operation taking the form of aP for $a \in Z_N^*$ and $P \in G_1^*$;
  - `e`: The operation taking the form of $e(P, Q)$ for $P, Q \in G_1^*$;
  - `X`: The operation taking the form of $(a \pm b) \mod N$ or $a \oplus b$ for $a, b \in Z_N^*$.

Table 1. Efficiency Comparison

	CL-PKE in [AP03][a]	CLK-PKE in [AP05]	Our PKE Scheme
Setup	1P	1P	1M
Ext-Partial-Key[b]	1P+2e	1P+2e	1E
Set-Private-Key	1P	--	1M
Set-Public-Key	2P	1P	1E
Encrypt	1E+1P+3e+2X	1E+2P+1e+3X	2M+1E+1X
Decrypt[c]	1e+2X	1e+3X	2M+1E+1X
Total	1E+6P+6e+4X	1E+5P+4e+6X	6M+4E+2X

[a] We take the Full CL-PKE scheme, instead of the basic scheme into account.
[b] In CL-PKE schemes, this item means partial-private-key-extract algorithm, while in our proposal it means partial-public-key-extract algorithm.
[c] The pre-computation cost takes only once, thus it is neglected.

4.2 Similarities and Differences

Now, let us give more elaborate comparisons of our proposal and well-known authentication frameworks (AF) from the following 7 perspectives:

– `who_gen_sk`: Who is in charge of private key generation?
– `who_gen_pk`: Who is in charge of public key generation?
– `who_man_pk`: Who is in charge of public key maintenance?
– `where_pub_pk`: Where does public key publish to?
– `pk_man_mode`: Which is the public key maintaining mode, centralized or decentralized?
– `pk_auth_mode`: Which is the public key authentication mode, explicit or implicit?
– `ttp_tl`: Which trust level can achieve for the TTP?

Table 2. Authentication Framework Comparison

	CA-based AF	ID-based AF	Certificateless AF	Our Scheme
`who_gen_sk`	user	PKG	user and PKG	user
`who_gen_pk`	user	user	user	user and PKG
`who_man_pk`	CA	indifferent, such as users themselves		
`where_pub_pk`	directory	indifferent, such as bulletin board, etc.		
`pk_man_mode`	centralized	decentralized		
`pk_auth_mode`	explicit	implicit		explicit
`ttp_tl`	3	1	2	3

Apparently, from the table 2, we can see that the proposed scheme enlightens an authentication framework which is different from CA-based, ID-based and cerfiticateless cryptosystems. If the certificateless authentication framework can be looked as intermediate solution between CA-based and ID-based schemes, then our proposal can also be looked as intermediate solution between CA-based and certificateless schemes.

Although our scheme still needs explicit public key authentication, just as that of in CA-based schemes, there are many different aspects between them. In addition, in traditional CA-based PKI, CA's signature on a public keys, i.e., certificate, is not the public key itself, while in our scheme, the partial public key, P_A, itself is a part of the public key.

4.3 Rethinking of Certificate Management Center in PKIs

In traditional CA-based PKI, a certificates management center is required. Thus, the management of infrastructure supporting certificates, including revocation, storage, distribution and the computation cost of certificate verification, incurs the main complaint against traditional PKCs. These situations are particularly acute in processor or bandwidth limited environments.

We think the most significant merits of certificateless schemes lies in the needless of a certificates management center, instead of needless of public key authentication process. In fact, there are implicit certificates in certificateless cryptosystems. However, in order to reach the highest trust level for the TTP, we think the explicit authentication process is necessary. What is more, explicit authentication makes encryption schemes more robust than those with implicit authentication, since an incorrect ciphertext will not be formed if the public key has been juggled.

At this juncture, we may rethink the necessary of certificate management centers in traditional CA-based cryptosystems. Maybe, centralized PKIs can be removed safely. Of course, this view is so abrupt that we cannot address anymore in this paper. More systematic and elaborate research is required before we draw more affirmatory conclusions on this topic.

5 New Authentication Framework from Proxy Signatures

We known that the certificates are, in essential, CA's signatures on the users' public keys. In order to alleviate CA's burden for signature generation. One may ask the users to generate the signatures by themselves – in a proxy manner. Further, if the proxy signature scheme used here is proxy protected, i.e., the signatures can only be generated by the users themselves, this authentication framework is likely to reach the highest trust level, i.e., trust level 3. Therefore, in this section, we will construct a new authentication framework from proxy signatures and denominate it as the proxy signature-based authentication framework (PS-based AF).

5.1 Diagram Semantic and Channel Symbols

The semantic of description diagram (All unspecified elements are picked randomly.) and the channel symbols used in the following contents are defined as:

$$Roles : \frac{Input}{Output}, \text{Additional actions.} \tag{11}$$

Symbols	Channels
$\xrightarrow{x}$	Public channel, directed
$\bullet \xrightarrow{x}$	Sender authenticated channel, directed
$\xrightarrow{x} \bullet$	Receiver authenticated channel, directed
$\bullet \xrightarrow{x} \bullet$	Mutual authenticated channel, i.e., secure channel, directed

For example, suppose that Alice takes as input x, computes y and z, and then sends z to Bob via a secure channel. We can depicted this semantic as the following diagrams:

$$Alice : \frac{x}{y, z}, \bullet \xrightarrow{z} \bullet Bob. \tag{12}$$

5.2 Definition of PS-based Authentication Framework

Definition 4 (PS-based AF). *Suppose Π_{PS} be a proxy signature with proxy protection, a proxy signature-based authentication framework consists of the following 9 diagrams:*

- *CA's keys generation, $\mathcal{G}_{\mathcal{CA}}$:*

$$CA : \frac{1^k}{(sk_{CA}, pk_{CA})}, \xrightarrow{pk_{CA}} * \tag{13}$$

  where $*$ denotes arbitrary receiver, i.e., making pk_{CA} public.
- *User's long term keys generation, $\mathcal{G}_{\mathcal{U}}$:*

$$User : \frac{1^k}{(sk_U, pk_U)}, \xrightarrow{pk_U} CA. \tag{14}$$

- *Proxy signing key generation, $\mathcal{G}_{\mathcal{P}}$:*

$$CA : \frac{1^k, sk_{CA}, pk_U, ID_U}{S_U}, \bullet \xrightarrow{S_U} \bullet User \tag{15}$$

  where S_U is the proxy signing key generated by CA (under the proxy signature scheme Π_{PS}) and it is sent to the user via a secure channel.
- *User's temporary keys generation, $\mathcal{G}_{\mathcal{T}}$:*

$$User : \frac{1^k}{(sk, pk)}, \xrightarrow{pk} * \tag{16}$$

- *User's temporary keys authentication, $\mathcal{A}_{\mathcal{T}}$:*

$$User : \frac{1^k, S_U, sk_U, pk}{Cert}, \xrightarrow{Cert} * \tag{17}$$

  where $Cert$ is the proxy signature generated by the user with the proxy signing key S_U (under the proxy signature scheme Π_{PS}) and its long term private key sk_U.
- *Encryption, $\mathcal{E}$:*

$$* : \frac{1^k, M \in \mathcal{M}, (pk, Cert), pk_{CA}, r}{\begin{cases} C \in \mathcal{C}, & \text{if } pk \text{ is valid;} \\ \perp, & \text{otherwise.} \end{cases}}, \xrightarrow{C/\perp} User. \tag{18}$$

- *Decryption, $\mathcal{D}$:*

$$User : \frac{1^k, C \in \mathcal{C} \cup \{\perp\}, sk}{\begin{cases} M \in \mathcal{M}, & \text{if } C \text{ is a valid ciphertext;} \\ \perp, & \text{otherwise.} \end{cases}} \tag{19}$$

– *Signature, $\mathcal{S}$:*

$$User : \frac{1^k, M \in \mathcal{M}, sk, r}{\sigma \in \mathcal{S}}, \xrightarrow{(M,\sigma)} Receiver. \tag{20}$$

– *Verification, $\mathcal{V}$:*

$$Receiver : \frac{1^k, (M,\sigma), (pk, Cert), pk_{CA}}{\begin{cases} 0, & if\ pk\ is\ valid\ while\ \sigma\ is\ invalid; \\ 1, & both\ pk\ and\ \sigma\ are\ valid; \\ \perp, & if\ pk\ is\ invalid. \end{cases}} \tag{21}$$

5.3 Security Notion of PS-based PKC

The security notion of PS-based encryption and signature can be defined by a similar way.

Definition 5. *Let $\Pi_E = (\mathcal{G_{CA}}, \mathcal{G_U}, \mathcal{G_P}, \mathcal{G_T}, \mathcal{A_T}, \mathcal{E}, \mathcal{D})$ be a PS-based encryption scheme, and let $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ be a ppt adversary. For $atk \in \{cpa, cca1, cca2\}$ and $1^k \in \mathbb{N}$ let*

$$Adv_{\mathcal{B},\Pi}^{ind-atk}(1^k) = \left| \Pr \begin{bmatrix} (sk_{CA}, pk_{CA}) \leftarrow \mathcal{G_{CA}}(1^k); \\ (sk_U, pk_U) \leftarrow \mathcal{G_U}(1^k); \\ S_U \leftarrow \mathcal{G_P}(1^k, sk_{CA}, pk_U, ID_U); \\ (sk, pk) \leftarrow \mathcal{G_T}(1^k); \\ Cert \leftarrow \mathcal{A_T}(1^k, S_U, sk_U, pk); \\ (m_0, m_1) \leftarrow \mathcal{B}_1^{\mathcal{O}_1}(pk, Cert, pk_U, sk_{CA}, pk_{CA}); \\ b \leftarrow \{0,1\}; \\ c^* \leftarrow \mathcal{E}(m_b, pk, Cert, pk_{CA}) : \\ \mathcal{B}_2^{\mathcal{O}_2}(m_0, m_1, c^*) = b \end{bmatrix} - \frac{1}{2} \right| \tag{22}$$

where $|\cdot|$ is the absolute-value function and

$$\begin{cases} \mathcal{O}_1 = \epsilon \ and\ \mathcal{O}_2 = \epsilon, & if\ atk = cpa, \\ \mathcal{O}_1 = \mathcal{D}_{sk}(\cdot) \ and\ \mathcal{O}_2 = \epsilon, & if\ atk = cca1, \\ \mathcal{O}_1 = \mathcal{D}_{sk}(\cdot) \ and\ \mathcal{O}_2 = \mathcal{D}_{sk}(\cdot), & if\ atk = cca2. \end{cases} \tag{23}$$

we insist that $\mathcal{B}_1$ outputs m_0 and m_1 with $|m_0| = |m_1|$. Also, $\mathcal{B}_2$ is not permitted to make the query $\mathcal{O}_2(c^*)$.

The encryption scheme Π_E is ind-atk secure if $Adv_{\mathcal{B},\Pi_E}^{ind-atk}(1^k)$ is negligible. The advantage function of the scheme is also defined by a similar way:

$$Adv_{\mathcal{B},\Pi_E}^{ind-atk}(1^k, t, q_d) = \max\{Adv_{\mathcal{B},\Pi_E}^{ind-atk}(1^k)\} \tag{24}$$

where the maximum is taken over all adversaries that run for time t and make at most q_d queries to the decryption oracle.

Definition 6. *Let $\Pi_S = (\mathcal{G}_{\mathcal{CA}}, \mathcal{G}_{\mathcal{U}}, \mathcal{G}_{\mathcal{P}}, \mathcal{G}_{\mathcal{T}}, \mathcal{A}_{\mathcal{T}}, \mathcal{S}, \mathcal{V})$ be a PS-based signature scheme, and let $\mathcal{F}$ be a ppt forger. For $1^k \in \mathbb{N}$, let*

$$Adv_{\mathcal{F},\Pi}^{EUF-CMA}(1^k) = \Pr\left[\begin{array}{c} (sk_{CA}, pk_{CA}) \leftarrow \mathcal{G}_{\mathcal{CA}}(1^k); \\ (sk_U, pk_U) \leftarrow \mathcal{G}_{\mathcal{U}}(1^k); \\ S_U \leftarrow \mathcal{G}_{\mathcal{P}}(1^k, sk_{CA}, pk_U, ID_U); \\ (sk, pk) \leftarrow \mathcal{G}_{\mathcal{T}}(1^k); \\ Cert \leftarrow \mathcal{A}_{\mathcal{T}}(1^k, S_U, sk_U, pk); \\ (m, \sigma) \leftarrow \mathcal{F}^{\mathcal{O}}(pk, Cert, pk_U, sk_{CA}, pk_{CA}) : \\ \mathcal{V}(m, \sigma, pk, Cert, pk_{CA}) = 1, M \notin \mathcal{M}_{\mathcal{O}} \end{array}\right] \tag{25}$$

where $\mathcal{O}$ denotes signing oracle and necessary Hash oracles used in the signing algorithm $\mathcal{S}$ while $\mathcal{M}_{\mathcal{O}}$ is the set of all messages that has been submitted to the signing oracle during $\mathcal{F}$'s whole querying process.

The signature scheme Π_S is EUF-CMA secure if $Adv_{\mathcal{F},\Pi_S}^{EUF-CMA}(1^k)$ is negligible. The advantage function of the scheme is also defined by a similar way:

$$Adv_{\mathcal{F},\Pi_S}^{EUF-CMA}(1^k, t, q_s) = \max\{Adv_{\mathcal{F},\Pi_S}^{EUF-CMA}(1^k)\} \tag{26}$$

where the maximum is taken over all adversaries that run for time t and make at most q_s queries to the signing oracle.

5.4 Concrete PS-based Encryption and Signature From RSA

The first and also the simplest proxy signature scheme based on RSA assumption is due to Shao [Shao03]. Now, let us take Shao's scheme as the building block to design a concrete PS-based authentication framework, including an encryption scheme and a signature scheme.

The concrete PS-based AF consists of 9 diagrams as follows:

– CA's keys generation, $\mathcal{G}_{\mathcal{CA}}$:

$$CA : \frac{1^k}{sk_{CA} = (p_0, q_0, d_0), pk_{CA} = (n_0, e_0)}, \xrightarrow{pk_{CA}=(n_0,e_0)} * \tag{27}$$

where p_0 and q_0 are two large secure primes, $n_0 = p_0 \cdot q_0$ the RSA-modulus, while d_0 and e_0 are the corresponding decryption exponent and encryption exponent, respectively, i.e., $e_0 d_0 \equiv 1 \pmod{\varphi(n_0)}$. Moreover, e_0 should not larger than the output of $h(\cdot,\cdot,\cdot)$, where h is an one-way hash function, which can be viewed as a public parameter.

– User's long term keys generation, $\mathcal{G}_{\mathcal{U}}$:

$$User\ i : \frac{1^k}{sk_i = (p_i, q_i, d_i), pk_i = (n_i, e_i)}, \xrightarrow{pk_i=(n_i,e_i)} * \tag{28}$$

where p_i and q_i are two large secure primes, $n_i = p_i \cdot q_i$ the RSA-modulus, while d_i and e_i are the corresponding decryption exponent and encryption exponent, respectively, i.e., $e_i d_i \equiv 1 \pmod{\varphi(n_i)}$. Moreover, e_i should not larger than the output of $h(\cdot,\cdot,\cdot)$.

– Proxy signing key generation, $\mathcal{G}_{\mathcal{P}}$:

$$CA : \frac{1^k, sk_{CA} = (p_0, q_0, d_0), pk_i = (n_i, e_i)}{s_i = (m_{w_i}, v_i)}, \bullet \xrightarrow{s_i=(m_{w_i}, v_i)} \bullet User\ i \quad (29)$$

where $v_i = h(m_{w_i}, n_i, e_i)^{-d_0} \mod n_0$ while m_{w_i} is a warrant, which records the delegation policy including limits of authority, valid periods of $pk_u = (n_e, e_i)$, and the identity and the public key of CA. Both m_{w_i} and v_i are generated by CA and sent to the user i via a secure channel. After receiving the (m_{w_i}, v_i), the user i checks whether

$$v_i^{e_0} h(m_{w_i}, n_e, e_i) \equiv 1 \pmod{n_0} \quad (30)$$

holds. If not, the user i broadcasts a "Complaint" message against CA.

– User's temporary keys generation, $\mathcal{G}_{\mathcal{T}}$:

$$User\ i : \frac{1^k}{sk_i = (p, q, d), pk = (n, e)}, \xrightarrow{pk=(n,e)} * \quad (31)$$

where p and q are two large secure primes, $n = p \cdot q$ the RSA-modulus, while d and e are the corresponding decryption exponent and encryption exponent, respectively, i.e., $ed \equiv 1 \pmod{\varphi(n)}$. Moreover, e should not larger than the output of $h(\cdot, \cdot)$.

– User's temporary keys authentication, $\mathcal{A}_{\mathcal{T}}$:

$$User : \frac{1^k, s_i = (m_{w_i}, v_i), pk = (n, e)}{\begin{array}{l} t \leftarrow_R [1, n_0], r \leftarrow t^{e_0} \mod n_0; \\ k \leftarrow h(r, n, e), u \leftarrow k^{d_i} \mod n_i; \\ y \leftarrow t \cdot v_i^k \mod n_0; \\ Cert = (m_{w_i}, y, u) \end{array}}, \xrightarrow{Cert} * \quad (32)$$

– Encryption, $\mathcal{E}$:

$$* : \frac{1^k, M \in \mathcal{M}, (pk = (n, e), Cert), pk_{CA}, r}{\begin{array}{l} \text{1. If } pk \text{ is invalid, output } \perp \text{ and then abort.} \\ \text{2. } r \leftarrow_R \{0,1\}^{k_0}, w \leftarrow h_3(M||r); \\ \text{3. } s \leftarrow g(w) \oplus (M||r), y \leftarrow (w||s); \\ \text{4. If } (y \geq n) \text{ goto 2;} \\ \text{5. } \sigma = y^e \mod n \end{array}}, \xrightarrow{C/\perp} User\ i. \quad (33)$$

where $h : \{0,1\}^* \to \{0,1\}^{k_1}$ and $g : \{0,1\}^{k_1} \to \{0,1\}^{k-k_1-1}$ are two working hash functions while $k = k_0 + k_1$ such that 2^{-k_0} and 2^{-k_1} are both negligible.

– Decryption, $\mathcal{D}$:

$$User\ i : \frac{1^k, C \in \mathcal{C} \cup \{\perp\}, sk = (n, d)}{\begin{array}{l} \text{1. } y \leftarrow \sigma^d \mod n; \\ \text{2. Parse } y \text{ as } w||s; \\ \text{3. Parse } g(w) \oplus s \text{ as } M||r; \\ \text{4. If } h_3(M||r) = w \text{ output 1 and } M; \\ \text{5. Otherwise, output 0 and } Null. \end{array}} \quad (34)$$

– Signature, $\mathcal{S}$:

$$User\ i : \frac{1^k, M \in \mathcal{M}, sk = (d, n), r}{\left\downarrow \begin{array}{l} 1.\ r \leftarrow_R \{0,1\}^{k_0}, w \leftarrow h_3(M||r); \\ 2.\ s \leftarrow g(w) \oplus (M||r), y \leftarrow (w||s); \\ 3.\ \text{If } (y \geq n) \text{ goto } 1; \\ 4.\ \sigma = y^d \mod n \end{array}\right.}, \xrightarrow{\sigma} Receiver. \tag{35}$$

– Verification, $\mathcal{V}$:

$$Receiver : \frac{1^k, \sigma, (pk = (n, e), Cert), pk_{CA}}{\left\downarrow \begin{array}{l} 1.\ \text{If } pk \text{ is invalid, output } \perp \text{ and then abort.} \\ 2.\ y \leftarrow \sigma^e \mod n; \\ 3.\ \text{Parse } y \text{ as } w||s; \\ 4.\ \text{Parse } g(w) \oplus s \text{ as } M||r; \\ 5.\ \text{If } h_3(M||r) = w \text{ output } 1 \text{ and } M; \\ 6.\ \text{Otherwise, output } 0 \text{ and } Null. \end{array}\right.} \tag{36}$$

Apparently, the system described above is secure enough, i.e., the encryption achieves IND-CCA2 security while the signature achieves EUF-CMA security, because

– On the one hand, the implementation of proxy signature is an equivalent variant of Shao's scheme in [Shao03], which is proxy protected and caters to the highest security requirements.
– On the other hand, the implementation of encryption and signature is not other than PSS-R scheme, which is a well-known secure signature with capability of message recovering. Here, we take it as both encryption scheme and signature scheme. By the way, one basic principal which must be taken into our mind at any time is that we should not use the same the working public/private key pair both for encryption and for signature simultaneously.

5.5 Further Discussions about the PS-based Authentication Framework

The advantages of the new authentication framework are very apparent. One the one hand, CA's burden for signature generation is alleviated greatly. In a PS-based scheme, the user's public key pk is authenticated by the user itself; meanwhile, the user has no chance to practice fraud since the validity of its public key can be checked by any entity with CA's public key pk_{CA}. One the other hand, whenever a user wants to revoke an old key pair (pk, sk), the only necessary step is to run $\mathcal{G}_{\mathcal{T}}$ and $\mathcal{A}_{\mathcal{T}}$ algorithms provided that the user's long term private key sk_U has not been compromised.

To construct a concrete and efficient PS-based scheme, a proxy signature, Π_{PS} with the property of proxy-protected, a general encryption Π_E and a general signature Π_S are necessary. Of course, these building blocks should achieve the desired securities. One the one hand, we think these tasks are not very difficult, considering that there are many secure proxy signature schemes, general encryption schemes and general signature schemes which cater to our requirements for building PS-based schemes. One the other hand, we also think these tasks are meaningful, since efficiency is must be taken into consideration. In particular, it is not trivial to construct a PS-based scheme using only *one* proper suite of parameters, i.e., both the proxy signature (Π_{PS}) and the destination schemes (Π_E and Π_S) are defined by *one* suite of parameters. The existence of such a construction is manifested by our implementation described in the previous subsection. Another interesting implementation of PS-based system could be to replace the basic building block – RSA scheme – with the improved RSA scheme proposed by Cao [Cao01], of which the security has been proved to be equivalent to the intractability of the factoring problem. Similarly, one can also build a secure PS-based system based on the multi-dimension RSA assumption proposed by Cao [Cao00]. Such a system would be even more efficient because that low exponents are permitted without any discounting in security.

6 Conclusions

The true meaning of certificateless schemes lies in the needless of a certificates management center, instead of needless of public key authentication process. In order to enhance the robustness of the certificateless schemes, we at first proposed a public key cryptosystem without certificate management center (PKCwCMC) based on quadratic residue theory. Then, we develop this idea further and proposed the proxy signature based authentication framework (PS-based AF) and describe a concrete implementation from RSA. Both of the PKCwCMC and PS-based AF can be looked as lite-CA based authentication frameworks since the workload and deployment of CAs in these systems are much lighter than those of in the traditional CA-based PKC.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China for Distinguished Young Scholars under Grant No. 60225007 and 60572155, the Science and Technology Research Project of Shanghai under Grant Nos. 04JC14055 and 04DZ07067, and the Special Research Funds of Huawei.

References

[AP03] S.S. Al-Riyami and K.G. Paterson. Certificateless Public Key Cryptography. At AsiaCrypt'03.

[AP05] S.S. Al-Riyami and K.G. Paterson. CBE from CL-PKE: A Generic Construction and Efficient Schemes. At PKC'05.

[Cao00] Zhenfu Cao. The multi-dimension RSA and its low exponent security. Science in China Series E, 43(4): 349-354, 2000.

[Cao01] Zhenfu Cao. A threshold key escrow scheme based on public key cryptosystem. Science in China Series E , 44(4): 441-448, 2001.

[DC06] Xiaolei Dong and Zhenfu Cao. $\mathbb{Z}_N$-trees and public-key cryptosystems. Discrete Mathematics, 2006 (to appear).

[FO99] E. Fujisaki and T. Okamoto. How to Enhance the Security of Public Key Encryption at Minimum Cost. In Public Key Cryptography PKC'99, LNCS 1560, pp. 53-68, Springer Verlag.

[Gen03] C. Gentry. Certificate-based encryption and certificate revocation problem. In E. Biham (Ed.): EUROCRYPT 2003, LNCS 2656, pp. 272-293.

[Gir91] M. Girault. Self-certified public keys, D.W. Davies (Ed.): Advances in Cryptology - EUROCRYPT'91, LNCS 547, pp. 490-497, 1991.

[GM84] S. Goldwasser and S. Micali. Probabilistic Encryption. Journal of Computer and System Sciences (1984) 270-299.

[Koh78] L.M. Kohnfelder. Towards a practical public key cryptosystem. MIT B.S. Thesis, MIT Department of Electrical Engineering, May 1978.

[Mao04] Wenbo Mao. Modern Cryptography: Theory and Practice. Publishing House of Electronics Industry. Beijing, 2004.

[Po99] D. Pointcheval. New Public Key Cryptosystems based on the Dependent-RSA Problems. EUROCRYPT'99, LNCS 1592, pp. 239-254.

[Sha84] A. Shamir. Identity-based cryptosystems and signature schemes, Advances in Cryptology – CRYPTO'84, LNCS 196, 1985, pp. 47-53.

[Shao03] Zuhua Shao. Proxy signature schemes based on factoring. Information Processing Letters 85 (2003) 137C143.

[Will80] H.C. Williams. A Modification of the RSA Public-Key Encryption Procedure. IEEE Transactions on Information Theory, IT No.6(26):726-729, November 1980.