Toward Real-life Implementation of Signature Schemes from the Strong RSA Assumption

Ping Yu and Rui Xue

State Key Laboratory of Information Security
Institute of Software, Chinese Academy of Sciences
Beijing, China 100190, yuping,rxue@is.iscas.ac.cn

Abstract. This paper introduces our work on performance improvement of signature schemes based on the strong RSA assumption for the purpose of real-life implementation and deployment. Many signature schemes based on the strong RSA assumption have been proposed in literature. The main advantage of these schemes is that they have security proofs in the standard model, while the traditional RSA scheme can only be demonstrated secure in the Random Oracle Model. However, the downside is the loss of efficiency among these schemes. Almost all these schemes double the computational cost of signature generation in the RSA scheme. So far the research in this area is more focusing on theoretical aspect. In this paper, we introduce techniques which greatly improve the performance of available schemes, and obtain a state-of-the-art signature scheme in the strong RSA family. In a typical setting where the RSA modulus is 1024 bits, it needs only one exponentiation calculation at the cost of about 160 modular multiplications, and a 162-bit prime number generation. This cost is even lower than the RSA signature scheme. Our work brings the current theoretical results into real-life implementation and deployment.

Keywords: Digital Signature, Efficiency, Real-life Implementation, Strong RSA Assumption.

1 Introduction

The digital signature concept is a fundamental primitive in modern cryptography. A digital signature scheme is a triple of algorithms: $(\mathtt{Gen}, \mathtt{Sig}, \mathtt{Ver})$. $\mathtt{Gen}(1^k)$ is called the key generation algorithm, which generates a pair of verification and signing keys (vk, sk) based on the security parameter k. $\mathtt{Sig}(sk, m)$ is called the signing algorithm, which produces a signature σ on message m. $\mathtt{Ver}(vk, m, \sigma)$ is called the verification algorithm, which checks if σ is a valid signature of message m. A basic requirement for a signature scheme is that a valid signature can only be produced by the signer who knows the signing key.

It is a challenging task to demonstrate the security of cryptographic schemes, including signature schemes. A popular method for carrying out security analysis is the Random Oracle Model, in which a public random oracle is set up to be

accessed by all parties. Since random oracles are a mathematical convenience for the sake of analysis, when such an algorithm is implemented in practice the random oracle is typically replaced by a cryptographic hash function. The random oracle methodology facilitates design and analysis of many cryptographic schemes. For example, the RSA scheme with the Optimal Asymmetric Encryption Padding (OAEP), which is one way the RSA scheme is used for encryption in practice, has been proved secure in the Random Oracle Model [2, 15]. Unfortunately, Canetti *et al.* showed that there exist schemes which can be proved secure in the Random Oracle Model, but any instantiation of the random oracle will result in a broken construction [5]. Their work shows the Random Oracle Model fundamentally has some issues.

Another model is called the real world model, or the standard model, in which the behaviors of all involved parties in the environment of a proof are the same as or indistinguishable from those in the real protocol in the view of attackers. No additional assumptions are needed to carry out the proof. It is always desirable for a scheme to be secure in the standard model since the Random Oracle Model is in essence a heuristic method for security proofs.

1.1 Signature Schemes from the Strong RSA Assumption

The first signature scheme is the well known RSA scheme [14]. The RSA scheme combined with a padding technique (e.g., the technique due to Bellare and Rogaway [3]) can be proved secure in the Random Oracle Model.

In 2000, Cramer and Shoup proposed the first practical signature scheme from the RSA family which has a security proof in the standard model [6]. It is based on a stronger assumption called the strong RSA assumption. Later, many schemes have been proposed in the strong RSA family with different types of enhancement for the purpose of efficiency and simplicity. Among those are the Camenisch-Lysyanskaya scheme [4], Zhu's scheme [17, 18], Fischlin's scheme [7], the Yu-Tate scheme [16], Joye's scheme [11], and others.

The major advantage of these schemes is that they all have security proofs in the standard model. The discouraging side is that they have much higher computational cost compared to the RSA scheme. For example, the computational cost for the Camenisch-Lysyanskaya scheme is about three times higher than the RSA scheme. It has been a continuous effort in this area to adjust design to improve efficiency with the hope of obtaining a scheme at least as efficient as the standard RSA scheme.

The performance issue in these schemes hampers people's interest to implement and deploy them in the real world. So far, we are not aware of any of these schemes being deployed in practice. Even though the RSA scheme can only be demonstrated secure in the Random Oracle Model, it has been working so well

in the real world for more than twenty years and people are satisfied with the current situation. To encourage real-life deployment of signature schemes in the strong RSA family which have better security property, it is critical for a candidate being at least as efficient as the RSA scheme. Otherwise, people might not show much interest in these theoretical results.

1.2 Contributions

In this paper, we propose a new signature scheme, which is the state-of-the-art signature scheme in the strong RSA family. We discuss techniques on parameter tuning on current schemes in the family. Even though these tricks are not theoretically significant, the performance improvement is not marginal. The new scheme is the first construction in the strong RSA family that is even more efficient than the standard RSA scheme.

In a typical setting in which the RSA modulus is 1024 bits, to produce a signature, the new scheme only needs one modular exponentiation whose cost is about 160 modular multiplications, plus the cost of a 162-bit prime number generation. This cost is the lowest one among all signature schemes in the strong RSA family, even lower than the standard RSA signature scheme which needs about 1024 modular multiplications. In addition, the new scheme can produce signatures in an online/offline manner. The majority of computation can be done before a message appears, and the online computation only needs a multiplication of two 160-bit integers. This is the best online performance which can be achieved so far. Joye's scheme has already achieved such a level of online performance, but its offline computation is about six times more expensive than the new scheme.

Our work brings the current theoretical results into real-life implementation and deployment. The rest of the paper is organized as follows. Section 2 reviews some cryptographic notations and definitions. We analyze the Camenisch-Lysyanskaya scheme in Section 3, and propose a method to improve its performance. Section 4 analyzes the Yu-Tate scheme, and discusses another way for performance improvement. We introduce the new scheme in Section 5. Section 6 gives a brief comparison on some typical signature schemes from the strong RSA assumption. Finally, we give the conclusions in Section 7.

2 Preliminaries

This section reviews some notations, definitions and security assumptions which are related to the discussion in this paper.

One of the first public key cryptographic systems published was the RSA scheme [14], which uses computations over modular group $\mathbb{Z}_n^*$, where $n = pq$,

p, q are both prime, and n should be constructed in a way such that the factorization of n is infeasible. This type of n is called an RSA modulus. Many different ways exist to construct n such that the resulting modular groups exhibit different properties which can be used in cryptographic constructions. Many constructions adopt a special RSA modulus which is defined as follows.

Definition 1 (Special RSA Modulus). *An RSA modulus $n = pq$ is called special if $p = 2p' + 1$ and $q = 2q' + 1$ where p' and q' are also prime numbers.*

Most signature schemes in the RSA family which have security proofs in the standard model rely on a well-accepted complexity assumption called the strong RSA assumption. This assumption was first introduced by Baric and Pfitzmann [1] and Fujisaki and Okamoto [8].

Assumption 1 (Strong RSA Assumption) *(SRSA Assumption) Let n be an RSA modulus. The* flexible RSA problem *is the problem of taking a random element $u \in \mathbb{Z}_n^*$ and finding a pair (v, e) such that $e > 1$ and $v^e = u \mod n$.*

The strong RSA assumption *says that no probabilistic polynomial time algorithm can solve the flexible RSA problem for random inputs with non-negligible probability.*

Even though the first signature scheme, i.e., the RSA scheme, was proposed back in 1977, a formal definition of a secure signature scheme appeared much later. The well accepted definition is called existential unforgeability under adaptive chosen message attacks, which was proposed by Goldwasser, Micali and Rivest in 1988 [10]. The definition we give here is due to Gennaro *et al.* [9].

Definition 2 (Secure Signatures [9]). *A signature scheme $S = \langle \mathtt{Gen}, \mathtt{Sig}, \mathtt{Ver} \rangle$ is existentially unforgeable under an adaptive chosen message attack if it is infeasible for a forger who only knows the public key to produce a valid (message, signature) pair, even after obtaining polynomially many signatures on messages of its choice from the signer.*

Formally, for every probabilistic polynomial time forger algorithm $\mathcal{F}$, there exists a negligible function $\mathtt{negl}(\cdot)$ such that

$$\Pr\left[\begin{array}{l} \langle vk, sk\rangle \leftarrow \mathtt{Gen}(1^k); \\ for\ i = 1 \ldots n \\ \quad m_i \leftarrow \mathcal{F}(vk, m_1, \sigma_1, \ldots, m_{i-1}, \sigma_{i-1});\ \sigma_i \leftarrow \mathtt{Sig}(sk, m_i); \\ \langle m, \sigma\rangle \leftarrow \mathcal{F}(vk, m_1, \sigma_1, \ldots, m_n, \sigma_n), \\ s.t.\ m \neq m_i\ for\ i = 1 \ldots n,\ and\ \mathtt{Ver}(vk, m, \sigma) = accept \end{array}\right] = \mathtt{negl}(k).$$

Intuitively speaking, an adaptive chosen message attack for a signature scheme is that a signature forger is allowed to adaptively choose any messages a polynomial number of times, asking the signer to produce signatures for these messages. If the forger can create a signature which is not produced by the signer before, the attack succeeds and the scheme is broken. Otherwise, we say that the signature scheme is secure.

3 Analysis of the Camenisch-Lysyanskaya Signature Scheme

In 2002, Camenisch and Lysyanskaya proposed a signature scheme secure in the standard model under the strong RSA assumption [4], which is referred to as the CL scheme in the rest of the paper. We discuss a way to improve the CL scheme in this section.

3.1 The CL Scheme

Like all digital signature schemes, the CL scheme has three procedures: key generation, signing and verification algorithms.

Key Generation. On input 1^k, choose a special RSA modulus $n = pq$, $p = 2p' + 1$, $q = 2q' + 1$ of length $l_n = 2k$. Choose uniformly at random, $a, b, c \in QR_n$. Output public key (n, a, b, c), and private key (p', q').

Signing Algorithm. On input message $m \in [0, 2^{l_m})$, choose a random prime number e of length $l_e \geq l_m + 2$, and a random number s of length $l_s = l_n + l_m + l$, where l is a security parameter. Compute the value v as

$$v = (a^m b^s c)^{e^{-1}} \mod n.$$

Verification Algorithm. To verify that the triple (e, s, v) is a signature on message m in the message space, check that $v^e \equiv a^m b^s c \mod n$, and $2^{l_e} > e > 2^{l_e - 1}$.

As specified in their paper, one parameter setting for the CL scheme is $k = 512$, so n is 1024 bits long. l_m can be chosen as 160, and messages longer than 160 bits can first be sent through a collision-resistant hash function (e.g., SHA-1) to produce a 160-bit message digest, which is then signed. The security parameter $l = 160$ so $l_s = 1024 + 160 + 160 = 1344$. For this setting of parameters, the cost of the signing algorithm is about $(160 + 1022 + 1344)$ modular multiplications and the generation of a 162-bit prime number. The verification requires $(1344 + 162 + 160)$ modular multiplications. Notice in the CL scheme s is required to be a very large integer, which contributes to a large portion of computational cost for signature generation.

3.2 Improvement of the CL scheme

If we look at the CL scheme carefully, one interesting observation would be uncovered. A valid CL signature satisfies

$$v^e \equiv a^m b^s c \mod n.$$

Notice, since $s > e$, s can always be represented as $s = k'e + s'$ for some k', and $s' < e$. Then we have

$$v^e \equiv a^m b^s c \equiv a^m b^{k'e+s'} c \equiv a^m b^{k'e} b^{s'} c \mod n.$$

Subsequently we obtain $v^e b^{-k'e} \equiv v'^e \equiv a^m b^{s'} c \mod n$, with $s' < e$, and $v' = vb^{-k'} \mod n$. This transformation shows that from a valid CL signature, we can always obtain a new valid signature with much shorter s. Therefore, we are able to obtain a variant of the CL scheme with $s < e$, which is obviously equivalent to the CL scheme in terms of security properties, since both schemes can be converted into each other by a trivial transformation. This implies that the length of s actually has no impact on the security properties of the scheme. Following the similar analysis, we can observe that s can be a random integer with the length between l_e and l_s.

We summarize our analysis as the following lemma.

Lemma 1. *The length requirement of s in the CL scheme has no impact on the security properties of the scheme. That is, s can be any random integer with the length between l_e and l_s as defined in the scheme. Therefore, we can adjust the length of s as needed to improve computational efficiency.*

4 Analysis of the Yu-Tate Signature Scheme

In 2008, Yu and Tate proposed an online/offline signature scheme which is referred to as the YT scheme [16]. Their scheme is similar to the CL scheme at the structural level. Both of them have the same verification algorithm and similar parameter choices. For example, the YT scheme also requires s being 1344 bits long for a typical setting where the RSA modulus is 1024 bits long. The major difference is that the YT scheme takes a different approach on signature generation. In the YT scheme, v is first calculated as $v = b^\gamma \mod n$, then s is computed out based on the relationship among exponents of a, b, c. More specifically, in the YT scheme, $a = b^\alpha \mod n$, and $c = b^\beta \mod n$. Therefore

$$\gamma \times e \equiv \alpha \times m + s + \beta \mod p'q'.$$

s can be calculated out based on this equation, and the verification algorithm is

$$v^e = a^m b^s c \mod n,$$

which is the same as the CL scheme.

The YT scheme only needs one exponentiation of 1022-bit exponent for signature generation in a typical setting. In comparison, the CL scheme needs three exponentiation calculations: one with 160-bit exponent, one with 1344-bit exponent, and one with 1022-bit exponent. Therefore the YT scheme is much more efficient than the CL scheme.

Our consideration is how to further improve the YT scheme. In the YT scheme, computing operations are conducted in the group of QR_n, since all parameters in the scheme are randomly chosen in QR_n. For example, when computing v, γ is picked up as a 1022-bit integer. We can consider to choose a smaller exponent to reduce computational cost. For example, we could pick a 160-bit integer instead of a 1022-bit integer. However, we need to address one issue for this consideration. The security proofs for the CL scheme and the YT scheme require v being randomly distributed in QR_n for the purpose of simulation. We should argue that this change will not affect an attacker's view in the proof.

The soundness of this consideration relies on the fact that it is infeasible to distinguish elements with short exponent and those with full size exponent. That is, informally speaking, if we have two elements $(a = g^x, b = g^y)$, where g is a generator of a group, $x \in_R (0, order(g))$, $y \in_R (0, 2^l)$, and $l < l_{order(g)}$, it is assumed impossible to make a decision whether a and b are generated based on different sizes of exponents. There is a well-known assumption called the discrete logarithm assumption with short exponent (DLSE) [13], which states that no efficient algorithms can calculate the exponent of an element if the exponent is larger than a threshold value for a large group. For example, it is assumed impossible to calculate the exponent r of a random element v such that $v = g^r$ where the length of r is longer than certain threshold length (e.g., 160 bits). Many secure problems related to the short exponent problem have been proposed, such as short exponent Diffie-Hellman problem, short exponent DDH problem, etc. Interested readers may refer to [13, 12] for detailed discussion.

Koshiba and Kurosawa proved that, based on the DLSE assumption, it is infeasible to distinguish elements with short exponent and those with full size exponent [12]. Their result is applied to groups whose order is known to attackers. For constructions in the strong RSA family, the order of the underlying group is not known. However, a simple observation shows this indistinguishable property still holds for a group with unknown order. Suppose we have two elements, one has a short exponent, while the other has a full size exponent.

Since the order of the group is not known to the attacker, we can simply tell the attacker the order of the group, which at least provides more information for him to use. Using the same proof by Koshiba and Kurosawa, we can show even when the order is known to the attacker, he still cannot distinguish these elements. Now, without the knowledge of the order of the group, this certainly makes the attacker's strategy more stringent. Therefore, we have the following lemma.

Lemma 2 (Indistinguishability between Short Exponent and Full Exponent). *Let g be a generator of G where the discrete logarithm problem with short exponent is assumed hard. Let l_f be the bit length of $order(g)$. Let $l_s < l_f$ and is greater than a threshold value so the DLSE assumption holds (e.g., $l_s > 160$). Let $(a = g^x, b = g^y)$, where $x \in_R (0, order(g)), y \in_R (0, 2^{l_s})$. Under the DLSE assumption, no probabilistic polynomial time algorithm can distinguish a, b with non-negligible probability.*

5 The New Signature Scheme

In this section, we introduce the new signature scheme based on the analysis in the previous sections.

5.1 The Scheme

- **Public System Parameters.** Let k be the security parameter. l is the length of a specific exponent used in the signing algorithm, which ensures the DLSE assumption holds (in practice, $l = 160$ is sufficient). l_m is the bit length of messages. l_e is the bit length of parameter e which is a prime number. It is required $l_e > l_m$.
- **Key Generation.** On input 1^k, pick two k-bit safe RSA primes p and q (so $p = 2p' + 1$, and $q = 2q' + 1$, where p' and q' are also prime), and let $n = pq$. Let $l_n = 2k$ be the length of the public modulus, Let QR_n be the quadratic residue group of $\mathbb{Z}_n^*$, and select a random generator b of QR_n. Select $\alpha, \beta \in_R [0, 2^l)$ and compute $a = b^\alpha \mod n$, $c = b^\beta \mod n$. Output public key (n, a, b, c), and private key $(p'q', \alpha, \beta)$.
- **Signing Algorithm.** The signing procedure includes two steps.
  STEP ONE: The signer picks a random $\gamma \in_R [0, 2^l)$, and a random prime e with length l_e, then computes

$$v = b^\gamma \mod n,\ \lambda = \gamma \times e - \beta.$$

STEP TWO: When a message $m \in [0, 2^{l_m})$ appears, the signer computes

$$s = \lambda - \alpha \times m.$$

The signature is (v, e, s) for the message m.

– **Verification Algorithm.** To verify that (v, e, s) is a signature on message m, check that

$$v^e \equiv a^m b^s c \mod n.$$

5.2 Performance Analysis

This new scheme is very efficient. A typical setting is that $l_n = 1024$, $l = 160$, $l_e = 162$, and $l_m = 160$. The major computation happens at STEP ONE, which needs about 160 modular multiplications and the cost of a 162-bit prime number generation. STEP TWO needs one multiplication of two 160-bit integers and an addition. Notice in the new scheme, s is about 322 bits long, and is much shorter than that in the CL ans YT schemes which is 1344 bits long.

The experiments in [6] show in the CS scheme the cost of generating a 161-bit prime is roughly one third of total cost of signature generation, and the CS scheme is 1.4 times slower than standard RSA scheme. In addition to prime number generation, the new scheme needs roughly 160 modular multiplications, where the CS scheme needs 1342 modular multiplications[1]. A simple calculation shows the new scheme runs faster than the RSA scheme ($1.4 \times (\frac{160}{1342} \times \frac{2}{3} + \frac{1}{3}) = 0.58$). This is the first scheme from the strong RSA family that is more efficient than the RSA scheme.

The new signature scheme produces signature in two steps. The first step does not need to know a message, so can be done offline. The second step can be done when the message is known, which only needs a multiplication of two 160-bit integers. So far the best online performance among online/offline signature schemes is due to Joye's scheme [11]. The new scheme achieves the same level of online performance as Joye's scheme. However, the offline computation of Joye's scheme is about six times more expensive than the new scheme.

In summary, the new scheme is the state-of-the-art signature scheme from the strong RSA assumption, with best online and offline performance.

5.3 Security Property

Based on our analysis on the CL scheme and the YT scheme, we have the following theorem for the security of the new scheme.

[1] The basic CS scheme needs 1502 modular multiplications. However, the implementation technique in Section 3 of [6] can reduce this cost to 1342 modular multiplications.

Theorem 1. *The new scheme is existentially unforgeable under an adaptive chosen message attack, assuming the strong RSA assumption and the DLSE Assumption, in the standard model.*

Proof. The CL scheme has been proved secure in the standard model based on the strong RSA assumption (Theorem 1 in [4]). As showed in Lemma 1, we can reduce the length of s to the setting in the new scheme without any impact on the security of the scheme. In the new scheme, b is a random generator of QR_n, and v, a, c are produced by choosing short exponents. By Lemma 2, v, a, c in the new scheme are indistinguishable from those in the CL scheme.

As a result, the new scheme is also secure in the standard model based on the strong RSA assumption and the DLSE assumption. □

6 A Brief Comparison among Signature Schemes from the Strong RSA Family

In this section, we give a brief comparison among signature schemes from the strong RSA family. We also use the RSA scheme as the base for comparison. The parameter choices are based on a typical setting in the RSA based schemes in which n is chosen as a 1024-bit integer. For simplicity, we can use the number of modular multiplications in a scheme to estimate computational cost. For example, for a modular exponentiation $g^x \mod n$, if x is a 160-bit integer, we can estimate its cost as 160 modular multiplications. All strong RSA based schemes need to produce a large prime number, and we consider that all schemes have the same cost for prime number generation. In the following table, we use "+ e" to represent the cost of prime number generation. For schemes which need hash computation, we take the bit length of hash value as 160. The comparison is showed in the following table. Clearly, the new scheme is the most efficient scheme so far.

Signature Scheme	Cost of Signature Generation	Support Online/Offline
RSA	1024	No
the CS scheme	1342 + e	No
the CL scheme	2526 + e	No
Zhu's scheme	1342 + e	No
Joye's scheme	1342 + e	Yes
the YT scheme	1022 + e	Yes
the new scheme	160 + e	Yes

Table 1. Comparison of Signature Schemes from the Strong RSA Family

7 Conclusions

In this paper, we discussed techniques on performance improvement of signature schemes based on the strong RSA assumption, and proposed a new signature scheme, which is state-of-the-art among constructions in the strong RSA family. It is the first signature scheme based on the strong RSA assumption that outperforms the standard RSA scheme. Before that, all available schemes in this family have low computational performance compared to the RSA scheme. Moreover, the new scheme can be proved secure in the standard model, while the standard RSA construction can only be demonstrated secure in the Random Oracle Model. Furthermore, the new scheme supports online/offline signing, and online performance stands in line with the best online/offline scheme so far (Joye's scheme).

Our work brings the current theoretical results into real life practice. Our next work will be implementation, field testing and verification, and future standardization.

Acknowledgments

This work is supported partially by NSFC No. 60903210 and No. 60873260, China national 863 project No. 2009AA01Z414, and China national 973 project No. 2007CB311202.

References

1. N. Baric and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology — Eurocrypt'97*, pages 480–494, 1997.
2. M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *A. de Santis, editor, Advances in Cryptology —Eurocrypt'94, LNCS 950*, pages 92–111. Springer-Verlag, 1995.
3. M. Bellare and P. Rogaway. The exact security of digital signatures — how to sign with RSA and Rabin. In *Advances in Cryptology — Eurocrypt'96*, pages 399–416, 1996.
4. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *Third Conference on Security in Communication Networks (SCN'02)*, pages 268–289, 2002.
5. R. Canetti, O. Goldreich, and S. Halevi. The random oracle model, revisited. In *30th Annual ACM Symposium on Theory of Computing*, pages 209–218, 1998.
6. R. Cramer and V. Shoup. Signatures schemes based on the strong RSA assumption. In *ACM Transaction on Information and System Security*, pages 161–185, 2000.
7. M. Fischlin. The Cramer-Shoup strong-RSA signature scheme revisited. In *International Workshop on Practice and Theory in Public Key Cryptography (PKC 2003)*, pages 116–129, 2003.
8. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology — Crypto'97*, pages 16–30, 1997.
9. R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signatures without the random oracle. In *Advances in Cryptology — Eurocrypt'99*, pages 123–139, 1999.

10. S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Computing*, 17:281–308, 1988.
11. M. Joye. An efficient on-line/off-line signature scheme without random oracles. In *CANS 2008, LNCS 5339*, pages 98–107, 2008.
12. T. Koshiba and K. Kurosawa. Short exponent diffie-hellman problems. In *PKC 2004 LNCS 3027*, pages 173–186. Springer-Verlag, 2004.
13. P. C. V. Oorshot and M. J. Wiener. On diffie-hellman key agreement with short exponents. In *Advances in Cryptology–Eurocrypt'96, LNCS 1070*, pages 332–343, 1996.
14. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. In *Commnuications of the ACM*, volume 21, pages 120–126, Feb. 1978.
15. V. Shoup. OAEP reconsidered. In *Advances in Cryptology — Crypto'01 LNCS 2139*, pages 239–259, 2001.
16. P. Yu and S. R. Tate. Online/offline siganture schemes for devices with limited computing capabilites. In *CT-RSA 2008, LNCS 4964*, pages 301–317, 2008.
17. H. Zhu. New digital signature scheme attaining immunity to adaptive chosen-message attack. *Chinese Journal of Electronic*, 10(4):484–486, 2001.
18. H. Zhu. A formal proof of Zhu's signature scheme, 2003. http://eprint.iacr.org/.