STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **XLVII**, Number 1, 2002

A MODIFICATION OF THE CRAMER-SHOUP DIGITAL SIGNATURE SCHEME

CONSTANTIN POPESCU

Abstract. Digital signatures have been used in Internet applications to provide data authentication and non-repudiation services. Digital signatures will keep on playing an important role in future Internet applications. In this paper we propose a secure digital signature scheme based on the Strong RSA Assumption. Compared with the recent signature scheme by Cramer and Shoup, public keys in our scheme are a bit smaller but the two schemes have about the same computational efficiency.

Keywords: Signature schemes, efficiency, security, adaptively chosen message attack, hash functions

1. Introduction

In 1976 Diffie and Hellman [4] devised the concept of public key cryptography and showed that secret communication is possible without a prior exchange of a secret key, as was necessary previously. Their ingenious idea was to use two different keys, a public key for encryption and a private key for decryption. Based on this asymmetry, they further devised the concept of digital signatures. There are two most well-known public key cryptosystems, the RSA scheme and the ElGamal scheme, which can provide both digital signature and data encryption. In the following years, others realizations of digital signature schemes were proposed [2], [3], [9], [15], [17]. The RSA scheme [16] can be used to provide both digital signatures and public key encryption. Its security relies on the difficulty of factorizing a modulus which is the product of two large primes. The algorithms of ElGamal [5] can also provide digital signatures and public key encryption. These rely on the difficulty of finding discrete logarithms in the field of integers modulo a large prime p. Subsequent refinements have been made to the original ElGamal schemes, particularly to the signature scheme. For example, the Digital Signature Standard (DSS) algorithm [6] combines ElGamal signatures with a idea of Schnorr [17] to increase efficiency and provide short signatures. More recently, the Elliptic

2000 *Mathematics Subject Classification.* 94A60.

1998 *CR Categories and Descriptors.* E.3 [**Data**]: Data Encryption.

Curve Cryptosystems (ECC) [11], [12], [13], in which the difficulty of breaking the system is based on the difficulty of computing a discrete logarithm over an elliptic curve, has also been considered to become a standard in the IEEE P1363 project.

Since digital signature has one of the unique features associated with the public key cryptography, digital signature has been used in security services to provide non-repudiation services. For example, digital signature has been used in the Secure Electronic Transactions (SET) standard [18] to provide security of electronic transfers of credit and payment information over the Internet. Digital signature has been adopted by many security protocols, such as SSL [19], to provide data authentication and non-repudiation services.

In this paper we propose a digital signature scheme which is provably secure against adaptive chosen message attacks [9]. This improves on recent results by Gennaro et al. [8] in that we do not require that the involved hash function is division intractable. Compared with the recent signature scheme by Cramer and Shoup [3], public keys in our scheme are a bit smaller but the two schemes have about the same computational efficiency.

2. The Model of a Signature Scheme

A user's signature on a message m is a string which depends on m, on public and secret data specific to the user and, possibly on randomly chosen data, in such a way that anyone can check the validity of the signature by using public data only. The user's public data are called the *public key*, whereas his secret data are called the *secret key*. Obviously we would like to prevent the forgery of a user's signature without knowledge of his secret key. In this section we give a more precise definition of signature schemes and of the possible attacks against them.

Definition 1. *A digital signature scheme consists of three algorithms:*

***Gen**: On input of a security parameter* 1^l *this probabilistic algorithm output the signer's secret and public keys,* x *and* y*, respectively.*

***Sign**: On input of the signer's secret and public keys and a message* $m \in \{0,1\}^*$ *this algorithm outputs a signature* σ *on* m*.*

***Verify**: On input of a message* m*, a signature* σ *and the public key* y *of a signer, the algorithm **Verify** outputs true or false.*

A signature scheme must satisfy the following properties:

(1) **Correctness**: Signatures produced by the signer with **Sign** must be accepted by **Verify**.
(2) **Unforgeability**: A signature scheme must be existentially unforgeable under an chosen message attack. That is, we require that every attacker

has a negligible probability of success in the following game. The attacker is allowed to sequentially obtain signature on polynomially many messages of his choosing (i.e., messages are allowed to depend on signatures that the adversary has seen). He is then required to produce as output a message m for which he did not receive a signature and a second string σ. If $\mathbf{Verify}(m, \sigma, y) = \mathit{true}$ then the attacker is successful and, hence, the signature scheme is vulnerable to existential forgery.

There are two specific kinds of attacks against signature schemes: the *no-message attack* and the *known-message attack.* In the first scenario the attacker only knows the public key of the signer. In the second one the attacker has access to a list of message-signature pairs. According to the way this list was created, we distinguish four subclasses of known-message attacks:

(1) The *plain known-message attack*: the attacker has access to a list of signed messages, but he has not chosen them.
(2) The *generic chosen-message attack*: the attacker can choose the list of messages to be signed. However, this choice must be made before accessing the public key of the signer. We call attack generic because the choice is independent of the signer.
(3) The *oriented chosen-message attack*: as above, the attacker chooses the list of messages to be signed, but the choice is made once the public key of the signer has been obtained. This attack is oriented against a specific signer.
(4) The *adaptively chosen-message attack*: having knowledge of the public key of the signer, the attacker can ask the signer to sign any message that he wants. He can then adapt his queries according to previous message-signature pairs.

We now classify the expected results of an attack:

- Disclosing the secret key of the signer. It is the most serious attack. This attack is termed *total break.*
- Constructing an efficient algorithm which is able to sign any message. This is called *universal forgery.*
- Providing a new message-signature pair. This is called *existential forgery.* In many cases this attack is not dangerous, because the output message is likely to be meaningless. Nevertheless, a signature scheme which is not existentially unforgeable does not guarantee by itself the identity of the signer. For example, it cannot be used to certify randomly looking elements, such as keys.

Definition 2. *A signature scheme is secure if an existential forgery is computationally impossible, even under an adaptively chosen-message attack.*

The first secure signature scheme was proposed by Goldwasser et al. [10] in 1984.

3. Number Theoretic Assumptions

This section reviews some cryptographic assumptions necessary in the subsequent design of our signature scheme.

The Strong RSA Assumption was independently introduced by Baric and Pfitzmann [1] and by Fujisaki and Okamoto [7]. It strengthens the widely accepted RSA assumption that finding e^{th}-roots modulo n, where e is the public and thus fixed exponent, is hard to the assumption that finding an e^{th}-roots modulo n for any $e > 1$ is hard.

Definition 3 (Strong RSA Problem)**.** *Let $n = pq$ be an RSA-like modulus and let G be a cyclic subgroup of $\mathbb{Z}_n^*$ of order l_g. Given n and $z \in G$, the Strong RSA Problem consists of finding $u \in G$ and $e \in \mathbb{Z}_{>1}$ satisfying $z \equiv u^e (mod\ n)$.*

Assumption 1 (Strong RSA Assumption)**.** *There exists a probabilistic polynomial time algorithm K which on input 1^{l_g} outputs a pair (n, z) such that for all probabilistic polynomial-time algorithms P, the probability that P can solve the Strong RSA Problem is negligible.*

Consequently, if n is a safe RSA-modulus (i.e., $n = pq$ with $p = 2p' + 1$, $q = 2q' + 1$ and p, q, p', q' all prime), it is more cautions to work in the subgroup of quadratic residues modulo n, that is, in the cyclic subgroup $QR\,(n)$ generated by an element of order $p'q'$.

The next corollary shows that it is easy to find a generator g of $QR\,(n)$: it suffices to choose an element $a \in \mathbb{Z}_n^*$ satisfying $\gcd\,(a \pm 1, n) = 1$ and then to take $g = a^2\ mod\ n$. We then have $QR\,(n) =< g >$.

Proposition 1. *Let $n = pq$, where $p \neq q, p = 2p' + 1$, $q = 2q' + 1$ and p, q, p', q' all prime. The order of the elements in $\mathbb{Z}_n^*$ are one of the set $\{1, 2, p', q', 2p', 2q', p'q', 2p'q'\}$. Moreover, the order of $a \in \mathbb{Z}_n^*$ is equal to $p'q'$ or $2p'q'$ if and only if* $\gcd\,(a \pm 1, n) = 1$.

Corollary 1. *Let $n = pq$, where $p \neq q, p = 2p' + 1$, $q = 2q' + 1$ and p, q, p', q' all prime. Then, for any $a \in \mathbb{Z}_n^*$ such that* $\gcd\,(a \pm 1, n) = 1$, *$< a^2 >\subset \mathbb{Z}_n^*$ is a cyclic subgroup of order $p'q'$.*

The security of our digital signature scheme is based on the Strong RSA Assumption.

4. Our Secure Digital Signature Scheme

This section describes a secure digital signature scheme based on the Strong RSA Assumption. Let $\varepsilon > 1$ be a security parameter and let $l_p, l_{\lambda_1} > l_{\lambda_2}, l_{\gamma_1} > l_{\gamma_2}$ denote lengths. Define the integral ranges $\Lambda = \left[2^{l_{\lambda_1}} - 2^{l_{\lambda_2}}, 2^{l_{\lambda_1}} + 2^{l_{\lambda_2}}\right]$ and $\Gamma = \left[2^{l_{\gamma_1}} - 2^{l_{\gamma_2}}, 2^{l_{\gamma_1}} + 2^{l_{\gamma_2}}\right]$ such that for all $(x, e) \in \Lambda \times \Gamma$, we have $0 < x + 2^{2l_p} < e$. Finally, let $H : \{0,1\}^* \to \Lambda$ be a collision-resistant hash function [14].

4.1. **Key Generation.**

To generate his public and secret key, a signer runs the following algorithm (**Gen**):

(1) Select random secret l_p-bit primes p', q' such that both $p = 2p' + 1$ and $q = 2q' + 1$ are also prime. Set the modulus $n = pq$.
(2) Chose two random elements $a, a_0 \in QR(n)$.
(3) The public key consists of the tuple (n, a, a_0, H) .
(4) The corresponding secret key consists of (p', q').

4.2. **Signature Generation.**

To sign a message $m \in \{0,1\}^*$ a signer uses the following algorithm (**Sign**):

(1) Choose a prime $e \in \Gamma$ that was not used before.
(2) Choose a random integer $r \in \Lambda$.
(3) Compute $x = H(m \parallel e \parallel r)$ and $u = (a^x a_0)^{1/e} \pmod n$.
(4) Output the signature (u, e, r).

4.3. **Signature Verification.**

Checking whether a tuple (u, e, r) is a valid signature on a message $m \in \{0,1\}^*$ with respect to the public key n can be done as follow (the algorithm **Verify**):

(1) Check whether $(u, e, r) \in \mathbb{Z}_n^* \times \Gamma \times \Lambda$.
(2) Compute $x' = H(m \parallel e \parallel r)$.
(3) Check whether $u^e \equiv a^{x'} a_0 \pmod n$.
(4) Output the signature true if none of the checks failed.

5. Efficiency and Security Analysis

The cost of the **Sign** algorithm can be broken down into three components:

(1) Generation of a random prime e from the interval $\left[2^{l_{\gamma_1}} - 2^{l_{\gamma_2}}, 2^{l_{\gamma_1}} + 2^{l_{\gamma_2}}\right]$.
(2) Computation of its inverse e^{-1}.
(3) Computation of u which requires 2 exponentiations: one full with e^{-1} and one small with x.

The cost of the last step can be reduced by amending **Gen** to generate a_0 as a power of a, i.e., choose a random $r' \in \Lambda$ and compute $a_0 = a^{r'}$. The value r' would then become part of the secret key. This amendment allows us to avoid the small exponentiation in the last step above, i.e., the signer would perform only one full exponentiation with the exponent $(x + r')\, e^{-1}$.

The only possible drawback is the potential loss in the range of a_0 since it is no longer generated independently from a. However, we note that a similar speedup was proposed by Cramer and Shoup [3] where it was claimed that, since, a is highly likely a generator of $QR\,(n)$, the distribution of the resultant public key does not change significantly.

The cost of signature verification in our scheme is dominated by step 3 which requires two exponentiations: one full to compute u^e and one small to compute $a^{x'}$. However, the verification equation can be changed to $u^e \left(a^{-1}\right)^{x'} \equiv a_0\, (mod\ n)$ and hence the computation gets reduced to about one full exponentiation.

Next, we show that our signature scheme indeed satisfies the requirement for a secure signature scheme according to Definition 1. The correctness property follows from inspection of the scheme. It remains to prove the schemes security against an adaptively chosen message attack. Similar to [8] we require that:

- For every H a collision-resistant hash function, all primes $e \in \Gamma$ and every two messages m_1 and m_2 the distribution $H\,(m_1 \parallel e \parallel r)$ and $H\,(m_2 \parallel e \parallel r)$ induced be the random choice of r are statistically close.
- The Strong RSA Assumption holds in a world where there exists an oracle that on input a message m, a prime $e \in \Gamma$ and an $x \in \Lambda$ outputs an $r \in \Lambda$ such that $x = H\,(m \parallel e \parallel r)$.

Theorem 1. *The signature scheme presented above is secure against adaptively chosen messages attack under the Strong RSA Assumption and the further assumption that there exists a family of hash functions $\{\mathcal{H}\}$ satisfying the above requirements.*

Proof. Assume that the attacker A queries signature for K messages and then outputs a signature (u', e') on the message m'. We now show that if we take control over the hash function, then we can use this attacker to break the Strong RSA Assumption, i.e., we are given a z and an n and must find an w and v such that $w^v \equiv z\,(mod\ n)$.

Let $((u_1, e_1, r_1), m_1), ..., ((u_K, e_K, r_K), m_K)$ denote the signature-message pairs that are constructed during the interaction with A. In order for A to be successful its output $((u', e', r'), m')$ must satisfy $(u', e') \neq (u_i, e_i)$ for $1 \leq i \leq K$. Depending of whether $e_i \nmid e'$ for some i, there are two games to calculate a pair $(w, v) \in \mathbb{Z}_n^* \times \mathbb{Z}_{>1}$ satisfying $w^v \equiv z\ (mod\, n)$ from which we randomly chose one each time then play with the attacker. As mentioned before, we are assuming that there is an oracle that input a message m, a prime $e \in \Gamma$ and an $x \in \Lambda$ outputs an $r \in \Lambda$ such that $x = H(m \parallel e \parallel r)$. The adversary is allowed to query this oracle as well. The first of the two game goes as follows:

(1) Select $x_1, ..., x_K \in \Lambda$ and $e_1, ..., e_K \in \Gamma$.
(2) Set $a = z^{\prod_{1 \le l \le K} e_l} \; mod \; n$.
(3) Choose a random $r \in \{0,1\}^{2l_p}$ and set $a_0 = a^r \; mod \; n$.
(4) For all $1 \le i \le K$, compute $u_i = z^{(x_i + r) \prod_{1 \le l \le K; l \ne i} e_l} \; mod \; n$.
(5) Start A, feed it the (u_i, e_i, r_i), where we get r_i from the oracle, and eventually obtain $\left(x'; \left[u' = \left(a^{x'} a_0\right)^{1/e'} \; mod \; n, e', r'\right]\right)$ with $x', r' \in \Lambda$ and $e' \in \Gamma$.
(6) If $\gcd(e', e_j) \ne 1$ for all $1 \le j \le K$ output fail and stop. Otherwise, let $\widetilde{e} = (x' + r) \prod_{1 \le l \le K} e_l$. Since $\gcd(e', e_j) = 1$ for all $1 \le j \le K$, we have $\gcd(e', \widetilde{e}) = \gcd(e', (x' + r))$. Hence, by the extended Euclidean algorithm, there exist $\alpha, \beta \in \mathbb{Z}$ such that $\alpha e' + \beta \widetilde{e} = \gcd(e', (x' + r))$. Therefore, letting $w = z^{\alpha} (u')^{\beta} \; mod \; n$ and $v = e' / \gcd(e', (x' + r)) > 1$ since $e' > (x' + r)$ we have $w^v \equiv z \; (mod \, n)$.

The previous game is only successful if A returns a new signature with $\gcd(e', e_j) = 1$ for all $1 \le j \le K$. We now present a game that solves the Strong RSA Problem in the other case, that is, when $\gcd(e', e_j) \ne 1$ for some $1 \le j \le K$. Note that $\gcd(e', e_j) \ne 1$ means $\gcd(e', e_j) = e_j$ since e_j is prime.

(1) Select $x_1, ..., x_K \in \Lambda$ and $e_1, ..., e_K \in \Gamma$.
(2) Choose a random $j \in \{1, ..., K\}$ and set $a = z^{\prod_{1 \le l \le K; l \ne j} e_l} \; mod \; n$.
(3) Choose a random $r \in \{0,1\}^{2l_p}$ and set $u_j = a^r \; mod \; n$ and $a_0 = u_j^{e_j} / a^{x_j}$ $mod \; n$.
(4) For all $1 \le i \le K$, $i \ne j$, compute $u_i = z^{(x_i + e_j r - x_j) \prod_{1 \le l \le K; l \ne i,j} e_l} \; mod$ n.
(5) Start A, feed it the (u_i, e_i, r_i), where we get r_i from the oracle, and eventually obtain $\left(x'; \left[u' = \left(a^{x'} a_0\right)^{1/e'} \; mod \; n, e', r'\right]\right)$ with $x', r' \in \Lambda$ and $e' \in \Gamma$.
(6) If $\gcd(e', e_j) \ne e_j$ output fail and stop. Otherwise, we have $e' = t e_j$ for some t and can define $b = (u')^t / u_j \, mod \; n$ if $x' \ge x_j$ and $b = u_j / (u')^t \; mod \; n$ otherwise. Hence $b \equiv \left(a^{\left|x' - x_j\right|}\right)^{1/e_j} \equiv \left(z^{|\widetilde{e}|}\right)^{1/e_j} \; (mod \, n)$ with $\widetilde{e} = (x' - x_j) \prod_{1 \le l \le K; l \ne j} e_l$. Since $\gcd\left(e_j, \prod_{1 \le l \le K; l \ne j} e_l\right) = 1$ it follows that $\gcd(e_j, |\widetilde{e}|) =$

$\gcd(e_j, |x' - x_j|)$. Hence, by the extended Euclidean algorithm, there exist $\alpha, \beta \in \mathbb{Z}$ such that $\alpha e_j + \beta |\tilde{e}| = \gcd(e_j, |x' - x_j|)$. Therefore, letting $u = z^{\alpha} b^{\beta}\ mod\ n$ and $e = e_j / \gcd(e_j, |x' - x_j|) > 1$ since $e_j > |x' - x_j|$, we have $u^e \equiv z\ (mod\, n)$.

Consequently, by playing randomly one of game 1 or game 2 with A one can solve the Strong RSA Problem. Since the latter is assumed to be infeasible, it follows that no such attacker can exist. ∎

We now compare our signature scheme with some recent results. The scheme due to Gennaro et al. [8] is simpler and seemingly more efficient than our scheme. The scheme is simpler since it appears as a true hash-and-sign scheme very close to RSA. It uses a similar variation of the Strong RSA Assumption for the proof of security as we do. However, their requirements for a suitable hash function are non-standard, e.g., it is required to be division intractable.

An interesting sidenote is that the only practical realization of a suitable hash function presented in [8] is the so-called chameleon hashing which outputs primes. This yields a signature scheme that requires the signer to generate a random looking prime. The cost of signing thus becomes roughly the same as in our present scheme: generation of a large prime, computation of its inverse and a single exponentiation. The cost of verification is one exponentiation plus the cost of a message hash which is quite expensive due to the special hash function used.

Comparing our signature scheme to the one by Cramer and Shoup, we find that are similar in many aspects of security properties and associated costs. The public key size in our scheme is somewhat smaller than its counterpart in Cramer and Shoup. The latter consists of a tuple (n, h, x, e'), where n is a modulus, h and x are elements of $QR(n)$ and e' is a prime. In contrast, our scheme's public key is a tuple (n, a, a_0, H), where n is a modulus, a and a_0 are elements of $QR(n)$ and H is a hash function which is, incidentally, also needed in a Cramer and Shoup public key. Thus, the size difference is due to the prime e' in the latter. A Cramer-Shoup signature is a tuple (y, y', e) where e is a small prime, $y' \in QR(n)$ and $y \in \mathbb{Z}_n^*$, i.e., both are n-bit integers. This is about the same as for our scheme. The cost of signing in [3] amounts to generating a prime, computing its inverse and three exponentiations of which two are small (each with an exponent from the range of the underlying hash function) and one is full. Hence, the cost of signing is somewhat higher in [3] than in our scheme. Signature verification in the Cramer and Shoup translates into two small exponentiations which is a bit more efficient than in our scheme.

6. Conclusion

In this paper we proposed a digital signature scheme which is provably secure against adaptive chosen message attacks. Compared with the recent signature scheme by Cramer and Shoup [3], public keys in our scheme are a bit smaller but the two schemes have about the same computational efficiency.

References

[1] N. Baric, B. Pfitzmann, Collision-free accumulators and fail-stop signature schemes without trees, In Advances in Cryptology-EUROCRYPT'97, Lecture Notes in Computer Science, vol. 1233, Springer-Verlag, pp. 480-494, 1997.

[2] R. Cramer, I. Damgaard, New generation of secure and practical RSA-based signatures, In Advances in Cryptology-Crypto'96, pp. 173-185, 1996.

[3] R. Cramer, V. Shoup, Signature Schemes Based on the Strong RSA Assumption, IBM Research Report RZ 3083, 1998.

[4] W. Diffie, M. Hellman, New Directions in Cryptography, IEEE Transaction Information Theory, IT-22, 6, pp. 644-654, 1976.

[5] T. ElGamal, A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, IEEE Transaction Information Theory, IT-31, 4, pp. 469-472, 1985.

[6] FIPS 186, Digital Signature Standard, US Department of Commerce/NIST, 1994.

[7] E. Fujisaki, T. Okamoto, A practical and provably secure scheme for publicly verifiable secret sharing and its applications, In Advances in Cryptology-EUROCRYPT'98, Lecture Notes in Computer Science, vol. 1403, Springer-Verlag, pp. 32-46, 1998.

[8] R. Gennaro, S. Halevi, T. Rabin, Secure hash-and-sign signatures without the random oracle, Proc. of Eurocrypt'99, LNCS, Springer-Verlag, 1999.

[9] S. Goldwasser, S. Micali, R. Rivest, A digital signature scheme secure against adaptive chosen-message attacks, SIAM Journal on Computing, 17(2), pp. 281-308, 1988.

[10] S. Goldwasser, S. Micali, R. Rivest, A "Paradoxical" solution to the signature problem, Proc. of the 25th FOCS, pp. 441-448, 1984.

[11] N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation, 48, pp. 203-209, 1987.

[12] N. Koblitz, CM-Curves with Good Cryptographic Properties, Proceedings of Crypto'91, 1992.

[13] V. Miller, Uses of elliptic curves in cryptography, Advances in Cryptology, Proceedings of Crypto'85, Lecture Notes in Computer Sciences, 218, Springer-Verlag, pp. 417-426, 1986.

[14] National Institute of Standards and Technology. Secure Hash Standard (SHS). FIPS Publication 180-1, April 1995.

[15] C. Popescu, Signature Schemes Based on the Discrete Logarithm Problem, Anal. of University of Oradea, pp. 25-32, 2001..

[16] R. Rivest, A. Shamir, L. Adleman, A Method for Obtaining Digital Signatures and Public Key Cryptosystems, Communications of the ACM, 21, pp. 120-126, 1978.

[17] C.P. Schnorr, Efficient Signature Generation by Smart Cards, Journal of Cryptology, Vol. 4, No. 3, pp. 161-174, 1991.

[18] SET Specification, http://www.visa.com/cgi-bin/vee/ht/ecomm/set/ downloads.

[19] The Secure Sockets Layer Protocol, http://www.netscape.com/info/security-doc.html.

University of Oradea, Department of Mathematics, Str. Armatei Romane 5, Oradea, Romania

E-mail address: cpopescu@math.uoradea.ro