Online/Offline Signature Schemes for Devices with Limited Computing Capabilities

Abstract. We propose a family of efficient digital signature schemes, which are proved secure under the strong RSA assumption without requiring a random oracle. The new signature schemes can operate in an online/offline manner, doing most of their work in the offline precomputation phase. The online phase, which is performed after the message to be signed is known, is very efficient, requiring only a single multiplication. Online/offline signatures are useful in settings in which signatures need to be produced with few operations, either when there is a large volume of requests or if the device performing the signature is not computationally powerful. Our schemes have extremely low computation cost so are particularly suitable for devices with limited computing capabilities such as smart cards or mobile devices.

This paper provides three specific contributions. First, we show how to modify the Camenisch-Lysyanskaya signature scheme to operate in an online/offline manner (the modifications also have benefits even when used as a traditional one-phase signature). Secondly, we show that we can use computations over a small subgroup of Z_n^* to further improve the efficiency of our basic signature scheme while retaining the online/offline characteristic. And third, we show that we can use division intractable hash functions to remove the requirement of generating random primes for use in this class of signature schemes.

Keywords: Digital Signature, Strong RSA Assumption, Standard Model, Online/Offline Signing, Devices with Limited Computing Capabilities.

1 Introduction

The digital signature concept is a fundamental cryptographic primitive in modern cryptography. In such schemes, a signer prepares a keypair which includes a signing key and a verification key. The signing key is kept secret by the signer while the verification key is public for potential verifiers. The signer generates a string by signing a message using his signing key. This string is called the signer's signature on this particular message. Later a verifier can check the validity of a signature on a message using the signer's verification key.

The idea of digital signatures was first proposed by Diffie and Hellman [10]. Since then, numerous constructions have been proposed in the literature based on different security assumptions. Many schemes are based on the well-known RSA assumption and a variant known as the strong RSA assumption, including PSS [3] and the Cramer-Shoup scheme [8]. Other schemes are based on variants of the discrete logarithm or computational/decisional Diffie-Hellman assumption, including ElGamal signatures [12], Schnorr signatures [24], and many others.

As a fundamental cryptographic primitive, it is important to clarify the exact requirements for a secure digital signature scheme. In 1988, Goldwasser *et al.* defined a security notion for signature schemes, which is called existential unforgeability under adaptive chosen message attacks [19]. Since then, this notion has been widely used to judge whether a digital signature scheme is strong enough to be deployed in a real application. Many digital signature schemes (e.g., Fiat-Shamir, Schnorr, ElGamal, PSS) can be proved secure under this requirement in the random oracle model, which was first proposed by Fiat and Shamir [14], and formalized by Bellare and Rogaway [2]. In the random oracle model, a cryptographic hash function is abstracted as a random function that can be accessed by all participants in the protocol, including adversaries. However, Canetti *et al.*

constructed a scheme that can be proved secure in the random oracle model, while any real implementation will result in an insecure construction [5]. Therefore, security proofs in the random oracle model do not necessarily imply security in the standard model, so a proof of security in the random oracle model can only be treated as a heuristic argument that a scheme is secure.

The notion of online/offline signatures was first introduced by Even, Goldreich, and Micali in 1989 [13]. An online/offline signature scheme does most of its work in an offline precomputation phase, before the message is known. The online phase, which is performed after the message to be signed is known, should be very efficient and can be completed quickly. Online/offline signing is important for applications when signatures need to be produced quickly. For instance, consider a stock broker's server that has "bursty" requests that need to be signed, where there are periods of low activity and infrequent bursts of rapid transaction requests (e.g., immediately after financial updates or news releases). Another example comes from the area of mobile computing, in which a mobile device with limited computing capabilities needs to authenticate itself by quickly producing a valid signature on a challenge from a server, but can precompute offline results at low speed and power in preparation for authentication requests. In still another example, the authenticating device could be a smart card, with a very weak processor, but which can be loaded with precomputed results of the offline phase from a more powerful device. In these scenarios, using precomputation can enable quick signature generation.

Even *et al.* proposed a generic method to convert any signature scheme into an online/offline one. However, their method is not efficient and practical. In fact, many signature schemes based on the discrete logarithm assumption naturally have online/offline versions without any further effort, while signature schemes based on the strong RSA assumption generally do not have this property. In 2001, Shamir and Tauman proposed another generic method to achieve online/offline signing [25]. Their method is based on a new type of hash function called a trapdoor hash function, which was proposed by Krawczyk and Rabin [21], and allows the use of a "hash-sign-switch" paradigm.

1.1 Related Work

Gennaro, Halevi and Rabin [17] and Cramer and Shoup [8], in 1999 and 2000, respectively, independently proposed practical digital signature schemes secure under adaptive chosen message attacks under the strong RSA assumption in the standard model. Before these two constructions, available schemes secure under an adaptive chosen message attack in the standard model were not practical for real applications [19, 7, 11]. Based on the ideas of Cramer and Shoup, Camenisch and Lysyanskaya [4], Zhu [26, 27], and Fischlin [15] all proposed schemes with similar structure based on the strong RSA assumption. In 2005, Groth extended these results to work over a small subgroup of Z_n^*, improving the efficiency of signature generation [20]. None of these schemes operate in an online/offline manner.

The first signature scheme suitable for devices with limited computing capabilities was devised by Schnorr [24]. Schnorr constructed a three-round identification scheme over a small prime-order subgroup of Z_p^*, which was then converted into a signature scheme using the Fiat-Shamir heuristic [14]. Schnorr's scheme is an efficient online/offline construction in which the signer needs about 160 modular multiplications for the offline phase and one modular multiplication for the online phase[1]. However, Schnorr's scheme can only be proved secure in the random oracle model due to

[1] In the parameters used in the original system, the signer needs only 140 modular multiplications. However, 160 could be considered more appropriate due to the advance of computing technology in the past 20 years.

the reliance on the Fiat-Shamir technique, and in fact Goldwasser and Kalai have recently published a result that casts doubt on the general applicability of the Fiat-Shamir technique [18].

1.2 Our Results

In this paper we propose a family of three efficient digital signature schemes, which we call OOSIG1, OOSIG2, and OOSIG3 ("OOSIG" is for online/offline signature). These signature schemes are progressively more efficient but rely on increasingly strong assumptions. All security proofs are in the standard model, without requiring a random oracle. We summarize the relevant features here:

OOSIG1: This is a modification of the Camenisch-Lysyanskaya signature scheme [4] to operate in an online/offline manner, and is proved secure under the Strong RSA assumption. We note that even when not applied in the online/offline setting, our modifications provide benefits and efficiency improvements over the Camenisch-Lysyanskaya scheme, saving several operations.

OOSIG2: In this scheme we use computations over a small subgroup of Z_n^* to further improve the efficiency of our basic signature scheme while retaining the online/offline capability, and is proved secure under the strong RSA subgroup assumption which was proposed by Groth [20].

OOSIG3: In this scheme, we remove one of the troubling requirements of this class of signature schemes — the necessity of generating a prime number for each signature. Instead, we use a hash function that satisfies certain properties, and so this scheme is proved secure under the assumption that this hash function is division intractable [17] as well as the strong RSA subgroup assumption.

OOSIG3 is very efficient, and the assumptions seem reasonable. Specifically, OOSIG3 needs only about 200 modular multiplications for the offline phase, which is only slightly more costly than the offline phase of Schnorr's scheme. The online phase requires only a few short (non-modular) multiplications, and hence is extremely efficient. To our knowledge, this is the first highly efficient online/offline signature scheme which has security based on variants of the strong RSA assumption in the standard model.

The rest of the paper is organized as follows. Section 2 reviews some cryptographic notations and definitions. Section 3 presents our basic scheme, which can be viewed as an online/offline extension of the Camenisch-Lysyanskaya signature scheme. In Section 4, we discuss ways to improve the efficiency of the basic scheme by using small subgroups of Z_n^*, leading to two new schemes for devices with limited computing capabilities. Finally, we give the conclusions in Section 5.

2 Preliminaries

This section reviews some notations and definitions which are used throughout the paper.

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' also are prime numbers.*

Definition 2 (Quadratic Residue Group QR_n). *Let Z_n^* be the multiplicative group modulo n, which contains all positive integers less than n and relatively prime to n. An element $x \in Z_n^*$ is called a* quadratic residue *if there exists an $a \in Z_n^*$ such that $a^2 \equiv x \pmod{n}$. The set of all quadratic residues of Z_n^* forms a cyclic subgroup of Z_n^*, which we denote by QR_n. If n is the product of two distinct primes, then $|QR_n| = \frac{1}{4}|Z_n^*|$.*

Recently, Groth investigated cryptography over small groups of Z_n^* [20]. Two of our proposed schemes are also constructed using this special kind of group, so we present definitions as used by Groth here.

Definition 3 (Small Subgroup G of Z_n^*). *Let $n = pq$ such that $p = 2p'r_p + 1$ and $q = 2q'r_q + 1$, where p, p', q, q' are all prime. There is a unique cyclic subgroup G of Z_n^* of order $p'q'$. For the purpose of efficient cryptographic construction, the order of G, i.e., $p'q'$, is chosen small and kept private. Let g be a random generator of G, and we call (n, g) an RSA subgroup pair.*

A hash function is a function mapping arbitrary strings of finite length to binary strings of fixed length. For cryptographic purposes, the most basic property that a hash function should satisfy is the collision-intractability property defined by Damgard [9]. One of our schemes in this paper will use another property for a hash function called "division intractability," which was introduced by Gennaro *et al.* [17]. Informally, a hash function is division intractable if it is infeasible to find distinct inputs for this hash function such that the hash value of one input divides the product of hash values of all other inputs.

Definition 4 (Division Intractability [17]). *A hashing family $\mathcal{H}$ is division intractable if it is infeasible to find distinct inputs $X_1, \ldots, X_n, Y$ such that $h(Y)$ divides the product of the $h(X_i)$'s.*

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

$$Pr_{h \in \mathcal{H}_k}\left[\begin{array}{l}\mathcal{A}(h) = \langle X_1, \ldots, X_n, Y\rangle \\ \; s.t.\, Y \neq X_i \; for \; i = 1 \ldots n, \\ \; and \; h(Y) \; divides \; \prod_{i=1}^{n} h(X_i)\end{array}\right] = negl(k).$$

Now we introduce the strongest notion of a secure signature scheme, existential unforgeability under adaptive chosen message attacks, which was proposed by Goldwasser, Micali and Rivest [19]. The definition we give here is due to Gennaro *et al.* [17].

Definition 5 (Secure Signatures [17]). *A signature scheme $S = \langle Gen, Sig, Ver\rangle$ 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 $negl()$ such that

$$Pr\left[\begin{array}{l}\langle pk, sk\rangle \leftarrow Gen(1^k); \\ for \; i = 1 \ldots n \\ \quad M_i \leftarrow \mathcal{F}(pk, M_1, \sigma_1, \ldots, M_{i-1}, \sigma_{i-1}); \; \sigma_i \leftarrow Sig(sk, M_i); \\ \langle M, \sigma\rangle \leftarrow \mathcal{F}(pk, M_1, \sigma_1, \ldots, M_n, \sigma_n), \\ M \neq M_i \; for \; i = 1 \ldots n, \; and \; Ver(pk, M, \sigma) = accept\end{array}\right] = negl(k).$$

The security of the signature schemes presented in this paper is based on a well-accepted cryptographic assumption called the strong RSA assumption, which was first proposed by Baric and Pfitzmann [1] and Fujisaki and Okamoto [16].

Assumption 1 (Strong RSA Assumption) *Let n be an RSA modulus. The* Flexible RSA Problem *is the problem of taking a random element $u \in Z_n^*$ and finding a pair (v, e) such that $e > 1$ and $v^e = u \pmod{n}$. The* Strong RSA assumption *says that no probabilistic polynomial time algorithm can solve the flexible RSA problem for random inputs with non-negligible probability.*

3 The Basic Signature Scheme

In 2002, Camenisch and Lysyanskaya proposed a signature scheme secure in the standard model under the strong RSA assumption [4]. The Camenisch-Lysyanskaya signature scheme produces a triple (v, e, s) as a signature, where e and s are chosen randomly, and v is computed from these values, the message, and the private key. Our basic scheme could be viewed as an online/offline extension of the Camenisch-Lysyanskaya scheme — it produces the same triple (v, e, s) for the signature, and we will show that the distribution of triples from our algorithm is statistically indistinguishable from the distribution of triples from the Camenisch-Lysyanskaya scheme. The key difference is that we randomly select v and e, and compute s, which requires a new key generation and signing algorithm. The benefit is that all but a small part of the computation for s can be done without knowing the message to be signed, so can be done in an offline phase. Since our signing algorithm produces the same signatures as in the Camenisch-Lysyanskaya scheme, our verification algorithm is the same as in the earlier scheme.

3.1 Signature Scheme OOSIG1

In this section, we define our first online/offline signature scheme, called OOSIG1.

Public System Parameters. Let k be the security parameter, and define the following lengths: l_m is the length of the message to be signed, with the restriction that $l_m < k - 2$. l is a security parameter that controls the statistical closeness of distributions, and should be at least polynomial in k (in practice $l = 160$ is sufficient). For convenience, we also define some lengths based on these parameters: $l_e = l_m + 2$ is the length of an exponent in the signature algorithm, $l_n = 2k$ is the length of the public modulus, and $l_s = l_n + l_m + l$ is the length of another exponent used in the signing algorithm.

Key Generation. On input 1^k, pick two k-bit safe primes p and q (so that $p = 2p' + 1$, and $q = 2q' + 1$, where p' and q' are also prime), and let $n = pq$. Select b as a random generator of QR_n. Select $\alpha, \beta \in_R [0, p'q')$ and compute $a = b^\alpha \pmod{n}$ and $c = b^\beta \pmod{n}$. Let $K = \lfloor 2^{l_s}/p'q' \rfloor$. Output public key (n, a, b, c), and private key $(p'q', \alpha, \beta, K)$.

Signing Algorithm. The signing procedure includes two phases.
OFFLINE PHASE: The signer picks a random $\gamma \in_R [0, p'q')$, a random l_e-bit prime number e, and a random $k' \in_R [0, K)$, and then computes

$$v = b^\gamma \pmod{n},\ \lambda = k'p'q' + \gamma e - \beta \pmod{Kp'q'}.$$

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

$$s = \lambda - \alpha m \pmod{Kp'q'}.$$

Note that while this is stated as a modular operation, the ranges of the values ensure that an adjustment to keep the value in range is only needed with negligible probability, and even then this is accomplished with a single addition. 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 e's length is l_e, and

$$v^e \equiv a^m b^s c \mod n. \tag{1}$$

It can be verified that a valid signature can always pass the verification algorithm. Since these operations are being performed in QR_n, we consider operations in the exponent modulo $p'q'$, and get

$$s \equiv \gamma e - \beta - \alpha m \pmod{p'q'},$$

and so

$$a^m b^s c \equiv b^{\alpha m + (\gamma e - \beta - \alpha m) + \beta} \equiv b^{\gamma e} \equiv v^e \mod n.$$

The salient characteristic for the signing algorithm is its online/offline mechanism. Most of the computation can be done before the appearance of a message, and the online phase only needs a single multiplication (where one of the values is short) and a subtraction.

3.2 Comparison with the Camenisch-Lysyanskaya Scheme

OOSIG1 produces signatures that are indistinguishable from those of the Camenisch-Lysyanskaya scheme. To see this, consider equation (1). In their scheme, a, b, c are randomly chosen from QR_n, and v is calculated as

$$v = (a^m b^s c)^{e^{-1}} \mod n, \tag{2}$$

where $s \in_R [0, 2^{l_s})$. In OOSIG1, a, b, c are also random elements of QR_n, and in the proof of Lemma 4 (in the appendix) we show that s in OOSIG1 is uniformly distributed over $[0, Kp'q')$, which is indistinguishable from $[0, 2^{l_s})$. Therefore, the view of an attacker with respect to our scheme is statistically indistinguishable from the view of an attacker with respect to the Camenisch-Lysyanskaya scheme, which gives the following theorem.

Theorem 1. *OOSIG1 is existentially unforgeable under an adaptive chosen message attack, assuming the strong RSA assumption.*

For concrete parameters, we use the recommended parameter settings from the Camenisch-Lysyanskaya scheme, with $k = 512$, 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. As stated earlier, $l = 160$ is sufficient to ensure the statistical closeness of the signature's actual distribution to the simulated distribution in the proof of the scheme [4], so $l_s = 1024 + 160 + 160 = 1344$. For this setting of parameters, the cost of the signing algorithm is about 1022 modular multiplications and the generation of a 162-bit prime number in the offline phase, and one multiplication in the online phase. Our algorithm avoids multiplication related to s in the Camenisch-Lysyanskaya scheme, which is about 1344 modular multiplications. Furthermore, note that OOSIG1 does not require computation of the multiplicative inverse of e as required by the Camenisch-Lysyanskaya scheme (see equation (2)), so has advantages even when not used in the online/offline mode.

The verification algorithm requires about (1344 + 162 + 160) modular multiplications. However this can be expedited by providing precomputed inverses of a and b and then verifying

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

since multi-base exponentiation can be done only slightly slower than single-base exponentiation (Algorithm 15.2 in [23]). Therefore, the verification needs only slightly more than 1344 modular multiplications.

4 Further Efficiency Improvements

The basic scheme can accommodate most application scenarios when online/offline signing is needed. However, it is possible to improve the efficiency further, particularly in the offline phase, which could be useful in situations such as mobile devices precomputing values in the offline phase during idle time. The main costs of the offline phase are due to an exponentiation taking 1022 modular multiplications and the generation of a 162-bit prime e. In this section, we reduce both of these costs: the exponentiation cost is reduced by working over a small subgroup of Z_n^*, and we remove the requirement of generating a prime e by using a division intractable hash function. These improvements also make the online phase more efficient: the online phase of OOSIG1 required the multiplication of a 1022-bit number by a 160-bit number, and the algorithms of this section reduce the size of the first number to 200 bits.

Using computations over a small subgroup of Z_n^* in this class of signature schemes was first investigated by Groth [20]. We incorporate these ideas into our basic scheme OOSIG1, reducing the bit length of $p'q'$ from the 1022 bits suggested for OOSIG1. Following Groth's suggestions, we set the bit length of $p'q'$ to 200, which reduces the number of modular multiplications for calculating v from 1022 to 200. The security of these new schemes is based on Groth's variant of the strong RSA assumption, called strong RSA subgroup assumption over this small subgroup of Z_n^* — we give this definition here, with terminology slightly cleaned up from the original paper [20].

Assumption 2 (Strong RSA Subgroup Assumption) *Let K be a key generation algorithm that produces an RSA subgroup pair (n, g). The flexible RSA subgroup problem is to find $u, w \in Z_n^*$ and $d, e > 1$ such that $g = uw^e \mod n$ and $u^d = 1 \mod n$. The strong RSA subgroup assumption for this key generation algorithm states that it is infeasible to solve the flexible RSA subgroup problem with non-negligible probability for inputs generated by K.*

In Section 4.1 we give our first subgroup-based signature scheme, using an idea of Groth's in order to reduce the cost of finding a prime e: we pick a much smaller prime number e, and a security parameter t such that $e^t \geq 2^{l_m+2}$. However, with a reduced range for e, this method could lead to multiple selections of the same prime number. Therefore, the signature scheme should pick e in an incremental way, and always remember the last prime number used. This way, the signature becomes a stateful construction. In a later section (Section 4.2) we will introduce another method which completely circumvents the cost of prime number generation, and keeps the scheme as a stateless one. The technique is based on the smoothness property of a random integer and the division intractability of a hash function, which have been introduced by Gennaro *et al.* [17], and further investigated by other authors [6, 22]. The security of both signature schemes will be proven in Section 4.3.

4.1 OOSIG2: A Stateful Signature Scheme

In this section, we present our second online/offline signature scheme, called OOSIG2.

Public System Parameters. The main parameters are similar to the OOSIG1 scheme, but with an additional parameter t chosen and l_e reduced subject to $t \times l_e \geq l_m + 2$. We also define a length $l_{p'q'}$ which determines the size of the subgroup used, and use this length to define $l_s = l_{p'q'} + l_m + l$.

Key Generation. On input 1^k, pick two k-bit primes p and q as in Definition 3 (so $p = 2p'r_p + 1$, and $q = 2q'r_q + 1$, where p' and q' are also prime, each with length $l_{p'q'}/2$), and let $n = pq$. Let G be the unique subgroup of Z_n^* of order $p'q'$, and select a random generator b of G. Select $\alpha, \beta \in_R [0, p'q')$ such that $a = b^\alpha \pmod{n}$, and $c = b^\beta \pmod{n}$. Let $K = \lfloor 2^{l_s}/p'q' \rfloor$. Output public key (n, a, b, c), and private key $(p'q', \alpha, \beta, K)$.

Signing Algorithm. The signing procedure includes two phases.
OFFLINE PHASE: Pick a random $\gamma \in_R [0, p'q')$, the next unused prime number e with length l_e, and a random $k' \in_R [0, K)$. Compute

$$v = b^\gamma \mod n,\ \lambda = k'p'q' + \gamma \times e^t - \beta \mod Kp'q'.$$

ONLINE PHASE: For $m \in [0, 2^{l_m})$, compute

$$s = \lambda - \alpha \times m \mod Kp'q'.$$

The signature is (v, e, s).

Verification Algorithm. To verify that (v, e, s) is a signature on message m, check that e's length is l_e, and

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

A concrete example would be $k = 512$ so that $l_n = 1024$. We further set up other system parameters as $l_{p'q'} = 200$, $l_m = 160$, $l_e = 28$, $t = 6$, $l = 120$, and $l_s = 200 + 160 + 120 = 480$. The whole signing cost now is about 200 modular multiplications, and finding the next prime number with bit length of 28, which is significantly easier than finding a 162-bit prime number.

4.2 OOSIG3: A Stateless Signature Scheme

In this section we show how to avoid using prime numbers explicitly for the exponent e. In signature schemes OOSIG1 and OOSIG2, there were two important system requirements: that $e^t \geq 2^{l_m+2}$, and e should not be chosen more than once. If we can somehow generate a random integer that always has a prime factor greater than m, we don't have to use a prime number explicitly. In order to accomplish this, we bring in results from Gennaro *et al.* [17] in a different signature scheme, where the authors introduce the notion of a division intractable hash function and also prove some important properties regarding the smoothness of random integers. In particular, they show that a k-bit random integer (for sufficiently large k) has at least one prime factor greater than $2^{2\sqrt{k}}$ with overwhelming probability. For example, suppose $k = 1024$, then this prime factor is greater than 2^{64}. Unfortunately, this bound is too small compared to our message size which we set $l_m = 160$. To overcome this obstacle, we can split the m into three pieces as $m = m_1||m_2||m_3$ where "||" represents string concatenation, and the bit length of each sub-message is shorter than 64 bits.

For simplicity of notation in this section, we split the message into three pieces as just described, but clearly this generalizes to other numbers of pieces. This technique has also been used in the Camenisch-Lysyanskaya scheme for block message signing, and in Fischlin's scheme for reducing the size of e. Therefore, we have the following stateless scheme.

Public System Parameters. The parameters are similar to the OOSIG2 scheme, but with $l_s = l_{p'q'} + l_m/3 + l$. We define a new length l_h to be the length of message digests produced by a division intractable hash function $h : \{0,1\}^* \rightarrow \{0,1\}^{l_h}$, with the requirement that $2\sqrt{l_h} \geq l_m/3 + 2$.

Key Generation. On input 1^k, pick two k-bit primes p, q such that $p = 2p'r_p + 1$, and $q = 2q'r_q + 1$, where p' and q' are also prime. Let $n = pq$, and let G be the unique subgroup of Z_n^* of order $p'q'$. Select a random generator b of G, select $\alpha_1, \alpha_2, \alpha_3, \beta \in_R [0, p'q')$, and define

$$a_1 = b^{\alpha_1} \bmod n, \;\; a_2 = b^{\alpha_2} \bmod n, \;\; a_3 = b^{\alpha_3} \bmod n, \;\; c = b^{\beta} \bmod n.$$

Finally, let $K = \lfloor 2^{l_s}/p'q' \rfloor$. The public key is (n, a_1, a_2, a_3, b, c), while the private key is $(p'q', \alpha_1, \alpha_2, \alpha_3, \beta, K)$.

Signing Algorithm. The signing procedure includes two phases.
OFFLINE PHASE: Pick a random $\gamma \in_R [0, p'q')$, a random $r \in_R [0, 2^{l_r})$, and a random $k' \in_R [0, K)$. Compute

$$v = b^{\gamma} \mod n, \; \lambda = k'p'q' + \gamma \times h(r) - \beta \mod Kp'q'.$$

ONLINE PHASE: For $m \in [0, 2^{l_m})$, break m into pieces such that $m = m_1 || m_2 || m_3$ and the length of each piece is at most $\lceil l_m/3 \rceil$ bits. Compute

$$s = \lambda - \alpha_1 \times m_1 - \alpha_2 \times m_2 - \alpha_3 \times m_3 \mod Kp'q'.$$

The signature is (v, r, s).

Verification Algorithm. To verify that (v, r, s) is a signature on message m, check that r's length is l_r, and

$$v^{h(r)} \equiv a_1^{m_1} a_2^{m_2} a_3^{m_3} b^s c \mod n.$$

This new scheme is extremely efficient for the signer. Given parameters $l_n = 1024$, $l_{p'q'} = 200$, $l_h = 1024$, $l_r = 256$, $l_m = 180$, $l = 120$, and $l_s = 200 + 180/3 + 120 = 380$, the offline signing cost is about 200 modular multiplications, while the online signing needs three multiplications with small numbers.

4.3 Security of OOSIG2 and OOSIG3

Our construction is similar to the schemes by Camenisch and Lysyanskaya [4], Zhu [26, 27], Fischlin [15], and Groth [20], except none of these schemes operate in the online/offline paradigm, and none except our OOSIG3 scheme make use of a division intractable hash function. However, due to the similarities, our security proof is similar to those in the previous schemes. The proof we give in this section is specifically for the OOSIG3 scheme, and the proof for OOSIG2 is simply a relaxation of the proof for OOSIG3 where we are guaranteed that the exponent e is a prime number, rather than relying on probabilistic results regarding large factors of random integers..

To prove the security of our scheme we use a multiple generator version of the strong RSA subgroup assumption: Given an appropriate modulus n so that Z_n^* has a subgroup G of size $p'q'$, let

$g_1, \ldots, g_k$ be random generators of this subgroup. The problem is to find values $(y, e, e_1, \ldots, e_k)$ such that $y^e = g_1^{e_1} \ldots g_k^{e_k} \mod n$. The following lemma (due to Groth [20]) shows that under the strong RSA subgroup assumption, a probabilistic polynomial time algorithm $\mathcal{A}$ can only reliably find solutions to this problem that have a restricted form.

Lemma 1. *Let $n, g_1, \ldots, g_k$ be as defined as above. If a probabilistic polynomial time algorithm $\mathcal{A}$ produces $(y, e, e_1, \ldots, e_k)$ such that $y^e \equiv g_1^{e_1} \ldots g_k^{e_k} \pmod n$, then with high probability either $e = e_1 = e_2 = \ldots = e_k = 0$, or $e|e_1, \ldots, e|e_k$ and $y = u \prod_{i=1}^{k} g_i^{e_i/e} \mod n$, where $u^e = 1 \mod n$.*

The next lemma addresses the smoothness of a random integer. The proof can be found in the proof of Lemma 6 presented by Gennaro *et al.* [17].

Lemma 2. *Let e be a random k-bit integer. The probability of e being $2^{2\sqrt{k}}$-smooth (i.e., all e's prime factors are smaller than $2^{2\sqrt{k}}$) is no larger than $2^{-2\sqrt{k}}$. In other words, the probability of e having at least one prime factor larger than $2^{2\sqrt{k}}$ is at least $1 - 2^{-2\sqrt{k}}$.*

The following lemma is used directly in the security proof for our stateless digital signature scheme — note that the condition on w is met for sufficiently large k whenever w is polynomial in k. The division intractability property of a hash function is based on this lemma: when a hash function outputs k-bit random integers for arbitrary inputs, it is intractable to find an input whose hash value can divide the product of other hash values. Due to the importance of this lemma, we re-write its proof to facilitate understanding of the subsequent proof. The original proof is presented by Gennaro *et al.* [17].

Lemma 3. *Let $e_1, e_2, \cdots, e_w$ be random k-bit integers, where $w \leq 2^{0.5\sqrt{k}}$. Let j be a randomly chosen index from $[1, w]$, and define $E = (\prod_{i=1}^{w} e_i)/e_j$. Then the probability that e_j divides E is less than $2^{-\sqrt{k}}$.*

Proof. We denote by smooth the event that e_j is $2^{2\sqrt{k}}$-smooth. From Lemma 2, we know that $Pr[\text{smooth}] \leq 2^{-2\sqrt{k}}$.

Consider the case in which e_j is not $2^{2\sqrt{k}}$-smooth. Then e_j has at least one prime factor $p > 2^{2\sqrt{k}}$, so $Pr[e_j \text{ divides } E]$ is bounded by the probability that at least one of the e_i $(i \neq j)$ is divisible by p. Since the e_i's are chosen uniformly, the probability that any specific e_i is divisible by p is at most $1/p < 2^{-2\sqrt{k}}$. Then, the probability that there exists an e_i which is divisible by p is at most $w \times 2^{-2\sqrt{k}}$, and based on the bound on w given in the lemma we get $w \times 2^{-2\sqrt{k}} < 2^{-1.5\sqrt{k}}$. Therefore, $Pr\left[p \text{ divides } E | \neg\text{smooth}\right] < 2^{-1.5\sqrt{k}}$, and since p is a prime factor of e_j, we get $Pr[e_j \text{ divides } E | \neg\text{smooth}] < 2^{-1.5\sqrt{k}}$.

Therefore, the probability that e_j divides E is at most

$$Pr[\text{smooth}] + Pr[e_j \text{ divides } E | \neg\text{smooth}] < 2^{-2\sqrt{k}} + 2^{-1.5\sqrt{k}} < 2^{-\sqrt{k}},$$

which completes the proof. □

Given these lemmas, we can now prove the security of OOSIG3.

Theorem 2. *Under the strong RSA subgroup assumption, OOSIG3 is existentially unforgeable under an adaptive chosen message attack.*

Proof. Suppose there exists a probabilistic polynomial time forgery algorithm $\mathcal{F}$, which can launch an adaptive chosen message attack on OOSIG3 and output a valid signature which has not been produced by the signing algorithm. Then we can construct a probabilistic polynomial time algorithm $\mathcal{A}$ for the multiple generator version of the strong RSA subgroup problem, defined immediately before Lemma 1. $\mathcal{A}$ takes a random input (n, g_1, g_2, g_3), with $g_1, g_2, g_3 \in_R G$, and uses $\mathcal{F}$ as a subroutine. In the following proof, all exponentiation is done modulo n.

Since G has order $p'q'$, where p' and q' are prime, the probability that g_1, g_2, g_3 are generators of G is $(p'-1)(q'-1)/p'q'$, which is an overwhelming probability. So, in the sequel, we assume g_1, g_2, g_3 are generators of G.

Suppose $\mathcal{F}$ asks w signature queries of his choice for messages $m_1, \ldots, m_w$, obtaining signatures $(v_1, r_1, s_1), \ldots, (v_w, r_w, s_w)$ before forging a valid signature (v, r, s) for a message m. We can define three types of forgeries.

Type I: For all $1 \leq i \leq w, r \neq r_i$.
Type II: For some $1 \leq i \leq w, r = r_i, v = v_i$.
Type III: For some $1 \leq i \leq w, r = r_i, v \neq v_i$.

Any forgery algorithm which succeeds in producing forgeries must produce forgeries of at least one of these types with non-negligible probability. Next, we show how to construct three different algorithms for $\mathcal{A}$ such that if $\mathcal{F}$ succeeds in producing a forgery of a particular type, then the corresponding $\mathcal{A}$ will succeed in solving the multiple generator version of the strong RSA subgroup problem. In all three cases we show that such an $\mathcal{A}$ is impossible, so we conclude that no successful forger $\mathcal{F}$ can exist.

Type I: For all $1 \leq i \leq w, r \neq r_i$. $\mathcal{A}$ works as follows: choose according to the signature algorithm distinct l_r-bit integers $r_1, \ldots, r_w$. Set $E = \prod_{i=1}^{w} h(r_i)$. $\mathcal{A}$ selects $t_1, t_2 \in_R [0, 2^{l_s})$, and sets $a_1 = g_1^E$, $a_2 = a_1^{t_1}$, $a_3 = a_1^{t_2}$, $b = g_2^E$, $c = g_3^E$. $\mathcal{A}$ gives (n, a_1, a_2, a_3, b, c) to the forger $\mathcal{F}$. $\mathcal{A}$ can answer the forger $\mathcal{F}$'s signature query $m_i = m_{i1}||m_{i2}||m_{i3}$ by choosing $s_i \in_R [0, 2^{l_s})$ and computing

$$v_i = g_1^{(m_{i1}+t_1 m_{i2}+t_2 m_{i3})E/h(r_i)} g_2^{s_i E/h(r_i)} g_3^{E/h(r_i)} = (a_1^{m_{i1}} a_2^{m_{i2}} a_3^{m_{i3}} b^{s_i} c)^{h(r_i)^{-1}}.$$

$\mathcal{A}$ gives $\mathcal{F}$ the signature (v_i, r_i, s_i) for message m_i, which is statistically indistinguishable from the signature produced by OOSIG3.

Consider that the forger $\mathcal{F}$ outputs (v, r, s) for a message m, so we have

$$v^{h(r)} = a_1^{m_1} a_2^{m_2} a_3^{m_3} b^s c = g_1^{(m_1+t_1 m_2+t_2 m_3)E} g_2^{sE} g_3^E.$$

By Lemma 1, $h(r)$ must divide E, but since E is the product of hash values this would violate the division intractability of the hash function h. Therefore, the $\mathcal{F}$ cannot produce a Type I forgery with non-negligible probability.

Type II: For some $1 \leq i \leq w, r = r_i, v = v_i$. $\mathcal{A}$ follows the same method as in Type I to prepare E, a_1, a_2, a_3, b, c, and answers the forger $\mathcal{F}$'s signature queries.

Consider now that the forger's signature is (v_i, r_i, s) on message m. We have

$$a_1^{m_{i1}} a_2^{m_{i2}} a_2^{m_{i3}} b^{s_i} = a_1^{m_1} a_2^{m_2} a_3^{m_3} b^s,$$

and so $a_1^{m_{i1}-m_1} a_2^{m_{i2}-m_2} a_3^{m_{i3}-m_3} b^{s_i-s} = 1$, which gives

$$g_1^{E((m_{i1}-m_1)+t_1(m_{i2}-m_2)+t_2(m_{i3}-m_3))} g_2^{E(s_i-s)} = 1 = y^0$$

for any non-zero y. Since $m_i \neq m$, $(m_1 - m_{i1}) + t_1(m_2 - m_{i2}) + t_2(m_3 - m_{i3}) \neq 0$. However, this is infeasible by Lemma 1 under the strong RSA subgroup assumption.

Type III: For some $1 \leq i \leq w, r = r_i, v \neq v_i$. $\mathcal{A}$ guesses the forger $\mathcal{F}$ will make the forgery by reusing r_i. $\mathcal{A}$ prepares E as in Type I, and picks at random an l_s-bit long t, and $(l_s - l_m/3 - l/2)$-bit long t_1, t_2, t_3, t_4. Then set up

$$b = g_2^{E/h(r_i)},\ a_1 = b^{t_1},\ a_2 = b^{t_2},\ a_3 = b^{t_3},\ c = b^{h(r_i)t_4 - t}.$$

$\mathcal{A}$ can answer all queries $j \neq i$ as in Type I. For query i, $\mathcal{A}$ computes $s_i = t - t_1 m_{i1} - t_2 m_{i2} - t_3 m_{i3}$, $v_i = b^{t_4}$ such that (v_i, r_i, s_i) is also a valid signature. Due to length restriction over t, t_1, t_2, and t_3, the distribution of s_i is statistically indistinguishable from the uniform distribution over $[0, 2^{l_s})$, which is in turn indistinguishable from the distribution of signatures produced by OOSIG3.

Consider now that the forgery $\mathcal{F}$ outputs a new signature (v, r_i, s) on message m. That is, $v^{h(r_i)} = a_1^{m_1} a_2^{m_2} a_3^{m_3} b^s c$. We can obtain

$$(v/v_i)^{h(r_i)} = g_2^{((m_1-m_{i1})t_1+(m_2-m_{i2})t_2+(m_3-m_{i3})t_3+(s-s_i))E/h(r_i)}.$$

We can assume that $h(r_i)$ has a prime factor $\pi > 2^{2\sqrt{l_h}}$ and that $h(r_i)$ divides $((m_1-m_{i1})t_1+(m_2-m_{i2})t_2+(m_3-m_{i3})t_3+(s-s_i))E/h(r_i)$ — these assumptions fail with negligible probability, due to Lemma 2 and Lemma 1, respectively. Furthermore, by Lemma 3, the probability that E contains π as a factor is negligible, so if $\mathcal{F}$ succeeds with non-negligible probability, it must be the case that

$$((m_1 - m_{i1})t_1 + (m_2 - m_{i2})t_2 + (m_3 - m_{i3})t_3 + (s - s_i)) \tag{3}$$

is divisible by π with non-negligible probability.

Let $t_1 = x_1 p'q' + t_1'$, and note that the forger's view is independent of x_1. Therefore, if the forger succeeds for this value of t_1 it must also succeed for a random $\widehat{t_1} = \widehat{x_1} p'q' + t_1'$ with $\widehat{x_1} \neq x_1$. Thus, π must divide (3) when t_1 is replaced by $\widehat{t_1}$, and so must divide the difference of these two values, leading to the requirement that π divides $(m_1 - m_{i1})(t_1 - \widehat{t_1}) = (m_1 - m_{i1})(x_1 - \widehat{x_1})p'q'$. However, since $2\sqrt{l_h} \geq l_m/3 + 2$, π cannot divide $m_1 - m_{i1}$. Furthermore, the probability that a random factor π divides $p'q'$ is negligible, and since x_1 and $\widehat{x_1}$ are chosen randomly the probability that π divides $(x_1 - \widehat{x_1})$ is also negligible. As we have exhausted all possibilities for $\mathcal{F}$ to succeed with a Type III forgery, we conclude that Type III forgeries are infeasible. □

5 Conclusions

In this paper we have presented a family of new efficient digital signature schemes, which are proved secure under the strong RSA assumption in the standard model. Our construction works in a two-phase offline/online model, so after some offline precomputation that is independent of the message to be signed, the online phase is highly efficient and suitable for devices with very limited computational capabilities. Compared to the well known digital signature scheme for smart cards by Schnorr, our scheme has comparable computational requirements and is provably secure in the standard model, while Schnorr's scheme relies on the Fiat-Shamir heuristic and thus can only be demonstrated secure in the random oracle model.

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. Random oracles are practical: A paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communication Security*, pages 62–73, 1993.
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. J.-S. Coron and D. Naccache. Security analysis of the Gennaro-Halevi-Rabin signature scheme. In *Advances in Cryptology – Eurocrypt'00*, pages 91–101, 2000.
7. R. Cramer and I. Damgard. New generation of secure and practical RSA-based signatures. In *Advances in Cryptology — Crypto'96*, pages 173–185, 1996.
8. 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.
9. I. Damgard. Collision free hash functions and public key signature schemes. In *Advances in Cryptography – Eurocrypt'87*, pages 203–216, 1987.
10. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 11:644–654, Nov. 1976.
11. C. Dwork and M. Naor. An efficient existentially unforgeable signature scheme and its applications. *J. Cryptology*, 11(3):187–208, 1988.
12. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology — Crypto'84*, pages 10–18, 1984.
13. S. Even, O. Goldreich, and S. Micali. On-line/off-line digital signatures. In *Advances in Cryptology — Crypto'89*, pages 263–275, 1990.
14. A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. In *Advances in Cryptology — CRYPTO'86*, pages 186–194, 1987.
15. 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.
16. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology — Crypto'97*, pages 16–30, 1997.
17. 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.
18. S. Goldwasser and Y. T. Kalai. On the (in)security of Fiat-Shamir paradigm. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science — FOCS'03*, pages 102– 114, 2003.
19. S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Computing*, 17:281–308, 1988.
20. J. Groth. Cryptography in subgroups of Z_n^*. In *Theory of Cryptography Conference (TCC 2005)*, pages 50–65, 2005.
21. H. Krawczyk and T. Rabin. Chameleon signatures. In *Symposium on Network and Distributed Systems Security – NDSS'00*, pages 143–154, 2000.
22. K. Kurosawa and K. Schmidt-Samoa. New online/offline signature schemes without random oracles. In *International Workshop on Practice and Theory in Public Key Cryptography 2006 — PKC 2006*, pages 330–346, 2006.
23. W. Mao. *Modern Cryptography: Theory & Practice*. Prentice Hall PTR, 2004.
24. C. Schnorr. Efficient signature generation for smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.
25. A. Shamir and Y. Tauman. Improved online/offline signature schemes. In *Advances in Cryptology — Crypto'01*, pages 355–367, 2001.
26. H. Zhu. New digital signature scheme attaining immunity to adaptive chosen-message attack. *Chinese Journal of Electronic*, 10(4):484–486, 2001.
27. H. Zhu. A formal proof of Zhu's signature scheme, 2003. http://eprint.iacr.org/.

A Appendix: Lemma 4

The following lemma shows s as used in our schemes is statistically indistinguishable from uniform over $[0, 2^{l_s})$.

Lemma 4. *Let $K = \lfloor 2^{l_s}/p'q' \rfloor$, where K is superpolynomial in the security parameter k. Let e be a value that is relatively prime to $p'q'$, α and β be constants in $[0, p'q')$, and m be a constant in $[0, 2^{l_m})$. Then if $k' \in_R [0, K)$ and $\gamma \in_R [0, p'q')$, if we define $s = (k'p'q' + \gamma e - \beta - \alpha m) \bmod Kp'q'$, then s is statistically indistinguishable from uniform over $[0, 2^{l_s})$.*

Proof. First, we prove that s is uniformly distributed over $[0, Kp'q')$. For any $x \in [0, Kp'q')$, since e is relatively prime to $p'q'$ there exists exactly one pair (k', γ) such that $x = (k'p'q' + \gamma e - \beta - \alpha m) \bmod Kp'q'$. Therefore there is a one-to-one mapping between pairs (k', γ) and values in $[0, Kp'q')$, and since pairs (k', γ) are chosen uniformly, the resulting distribution of s over $[0, Kp'q')$ is uniform.

Next, we prove that the uniform distribution over $[0, Kp'q')$ is statistically indistinguishable from uniform over $[0, 2^{l_s})$. Let $Pr_D(x)$ denote the probability of x in distribution D. Then the distance between two distributions D_1 and D_2 is

$$dist(D_1, D_2) = \frac{1}{2} \sum_x \left| Pr_{D_1}(x) - Pr_{D_2}(x) \right|,$$

and note that for any two distributions $dist(D_1, D_2) \leq 1$. Two distributions D_1 and D_2 are *statistically indistinguishable* if $dist(D_1, D_2)$ is negligible.

Distribution D_1 is uniform over $[0, Kp'q')$, and distribution D_2 is uniform over $[0, 2^{l_s})$. Doing the basic algebra, we get

$$\begin{aligned}
dist(D_1, D_2) &= \frac{1}{2} \sum_x |Pr_{D_1}(x) - Pr_{D_2}(x)| \\
&= \frac{1}{2}[(\frac{1}{Kp'q'} - \frac{1}{2^{l_s}})Kp'q' + \frac{1}{2^{l_s}}(2^{l_s} - Kp'q')] \\
&= 1 - \frac{Kp'q'}{2^{l_s}} \\
&= 1 - \frac{\lfloor 2^{l_s}/p'q' \rfloor p'q'}{2^{l_s}} \\
&< 1 - \frac{((2^{l_s} - p'q')/p'q')p'q'}{2^{l_s}} \\
&= 1 - (1 - \frac{p'q'}{2^{l_s}}) \\
&= \frac{p'q'}{2^{l_s}}.
\end{aligned}$$

Thus, the distance between D_1 and D_2 is less than $\frac{p'q'}{2^{l_s}}$, and since $\frac{2^{l_s}}{p'q'}$ is superpolynomial in the security parameter k, this distance is negligible. Therefore, the distribution of s is statistically indistinguishable from uniform over $[0, 2^{l_s})$. □