Security of Signature Schemes in a Multi-User Setting

Alfred Menezes[1] and Nigel Smart[2]

[1] Dept. of Combinatorics and Optimization, University of Waterloo, Canada
ajmeneze@uwaterloo.ca
[2] Dept. of Computer Science, University of Bristol, UK
nigel@cs.bristol.ac.uk

Abstract. This paper considers the security of signature schemes in the multi-user setting. We argue that the well-accepted notion of security for signature schemes, namely existential unforgeability against adaptive chosen-message attacks, is not adequate for the multi-user setting. We extend this security notion to the multi-user setting and show that signature schemes proven secure in the single-user setting can, under reasonable constraints, also be proven secure in the multi-user setting.

1 Introduction

The security of cryptographic primitives such as public-key encryption schemes, signature schemes, and message authentication code algorithms, has traditionally been studied in the *single-user setting*. That is, there is one legitimate entity who can, respectively, decrypt data, sign data, and authenticate data, and the adversary's goal is to compromise the security of these tasks. Similarly, the security of key establishment protocols has traditionally been studied in the *two-user setting*, where an adversary attempts to obtain information about cryptographic keys established by two communicating entities.

There has been some recent work in defining and proving security of some of the cryptographic primitives in the more realistic *multi-user setting*. Bellare and Rogaway [3] (see also [7, 27, 12]) first gave security definitions for symmetric-key entity authentication schemes and key transport schemes in the multi-user setting, and proved certain protocols secure. More recently, Bellare, Boldyreva and Micali [2] presented security definitions for public-key encryption in the multi-user setting. The motivation for their work was to account for attacks such as Håstad's attacks [19] whereby an adversary can easily recover a plaintext m if the same m (or linearly related m) is encrypted for three legitimate entities using the basic RSA encryption scheme with encryption exponent $e = 3$. Note that Håstad's attacks cannot be considered to defeat the security goals of public-key encryption in the single-user setting for the simple reason that there is only one legitimate entity in this setting. In [2], it is proven that a public-key encryption scheme that is secure (in the sense of being indistinguishable against either chosen-plaintext attacks or chosen-ciphertext attacks) in the single-user setting is also secure in the multi-user setting. The primary benefit of this result is that one can now safely employ in the multi-user setting public-key encryption schemes such as RSA-OAEP [4] and Cramer-Shoup [13] that had been proved secure in the single-user setting without concern for attacks such as Håstad's attacks.

In this paper, we consider the security of signature schemes with appendix in the multi-user setting. By considering the so-called "duplicate-signature key selection" attacks [8] on some commonly-used signature schemes, we argue that the well-accepted security definition for signature schemes (existential unforgeability against adaptive chosen-message attacks) by Goldwasser, Micali and Rivest [17] is not adequate for the multi-user setting. Fortunately, it appears that this definition can easily be extended to the multi-user setting. Furthermore, signature schemes proven secure in the single-user setting can, under reasonable constraints, also be proven secure in the multi-user setting.

The remainder of this paper is organized as follows. We provide further motivation for our work in §2. The notion of a signature scheme in the multi-user setting and our security definition are presented in §3. In §4 we prove that some signature schemes that are provably secure in the single-user setting are also secure in the multi-user setting if appropriate constraints are made on the domain parameters employed. Our conclusions are stated in §5.

2 Motivation

In 1984, Goldwasser, Micali and Rivest (GMR) [16,17] introduced the security notion of *existential unforgeability against adaptive chosen-message attacks* for signature schemes in the single-user setting. Here, the adversary is presented with a legitimate user's public key and a signing oracle with respect to that public key. The adversary's goal is to produce a valid signature for any message that was not previously queried of the oracle; such a message/signature pair is called an *existential forgery*. This security notion has become accepted as the "right" one for signature schemes. For example, [13] states: "This is the strongest type of security for a digital signature scheme that one can expect, and a signature scheme that is secure in this sense can be safely deployed in the widest possible range of applications."

2.1 Key substitution attacks

In [8], *duplicate-signature key selection* attacks on a signature scheme were introduced. We shall henceforth call these attacks *key substitution* (KS) attacks. Here, an adversary is presented with a legitimate user A's public key, and A's signature s_A on some message M. The adversary E's task is to produce a valid public/private key pair for which s_A is also E's signature on M; we call such a key pair a *key substitution*. The requirement that E knows the private key corresponding to her public key is necessary since otherwise E can trivially mount a KS attack by claiming that her public key is the same as A's.

We illustrate this attack on the Gennario-Halevi-Rabin (GHR) signature scheme [15]. In the GHR scheme, A's public key consists of a modulus n that is a product of two safe primes p and q of the same bitlength, and $t \in_R \mathbb{Z}_n^*$; A's corresponding private key is (p, q). To sign a message M, A computes $e = H(M)$ where H is a cryptographic hash function, and solves $s^e \equiv t \pmod{n}$ for $s \in \{1, 2, \ldots, n-1\}$; it is assumed that $\gcd(e, (p-1)(q-1)) = 1$, and hence there is a unique solution s. A's signature on M is s. To verify, one computes $e = H(M)$ and checks that $s^e \equiv t \pmod{n}$. Given n, t, M and s, the adversary E mounting a KS attack selects safe primes $\overline{p}$ and $\overline{q}$ of the same bitlength such that $\overline{n} = \overline{pq} > s$, and then computes $\overline{t} = s^e \bmod \overline{n}$ where $e = H(M)$. Now, $(\overline{n}, \overline{t})$, $(\overline{p}, \overline{q})$ is a valid public/private key pair, and s is also E's signature on M.

In [8], it is shown that KS attacks can also be successfully mounted on the Full Domain Hash (FDH) and Probabilistic Signature Scheme (PSS) versions [3] of the RSA signature scheme when entities are allowed to select arbitrary encryption exponents e, on DSA [21] when entities share domain parameters p and q but are allowed to select arbitrary generators $g \in \mathbb{Z}_p^*$ of order q, and ECDSA [1, 20] when entities share elliptic curve parameters $E/\mathbb{F}_q$ but are allowed to select arbitrary generating points $G \in E(\mathbb{F}_q)$.

The question then is whether a KS attack really constitutes an attack on the signature scheme. Certainly, a successful KS attack does not contradict the GMR security definition since there is only one public key in the single-user setting. This is not unexpected since the GHR scheme has been proven to be GMR-secure under the strong RSA assumption and some reasonable assumptions on the hash function employed, while the KS attack described above does not contradict these assumptions. On the other hand, [8] showed that the station-to-station (STS) key agreement protocol [14] when used with a MAC algorithm for key confirmation succumbs to an unknown key-share attack when the signature scheme employed is susceptible to KS attacks. Since unknown key-share attacks can be damaging in practice, a KS attack in this application can have damaging consequences. Moreover, in the physical world of hand-written signatures, one would certainly judge an adversary to be successful if she can pass off a message signed by A as her own without having to modify the signature.

Our thesis then is that a KS attack should be considered an attack on a signature scheme in the multi-user setting.

2.2 What is a signature scheme?

A digital signature scheme S binds together three things: (i) a public key y; (ii) a message m; and (iii) a signature s.

We assume that a public key uniquely identifies an entity. In practice, this may be accomplished by a Certification Authority (CA) who verifies that each entity knows the private key x corresponding to its public key y, and then conveys this knowledge to other parties via a certificate. A public key is associated with a set of domain parameters to be shared between all entities in a group. A public key y is *valid* if y satisfies some security requirements specified by the designer of the scheme, and if the owner A of y knows the corresponding private key. A triple (y, m, s) is *valid* if the public key is valid and if s is indeed A's signature (with respect to y) on m.

We assume that an adversary E of S in a multi-user setting has a signing oracle for each public key. By querying these oracles, E can gather valid triples $T = \{(y_i, m_i, s_i) : 1 \leq i \leq n\}$ for m_i's of its choice. The adversary's task is to produce a valid triple (y, m, s) not in T. There are several cases to consider—note that these cases are exhaustive but not necessarily mutually exclusive.

1. If $y = y_i$ for some i, but $m \neq m_j$ for all $j \in \{j : y = y_j\}$, then E has produced an existential forgery in the GMR sense.
2. If $(m, s) = (m_i, s_i)$ (and $y \neq y_i$) for some i, then we have an instance of a key substitution attack.
3. If $(y, m) = (y_i, m_i)$ (and $s \neq s_i$) for some i, then E has used the signature on a message to produce without knowledge of the private key a different valid signature on the same message. We cannot think of any damaging consequence of this ability, so we do not consider this an attack on S.
4. If $s = s_i$ for some i, but $y \neq y_j$ and $m \neq m_j$ for all $j \in \{j : s = s_j\}$, then E has generated a valid public key y and a message m that has the same signature s as some message $m_i \neq m$ with respect to another public key $y_i \neq y$. Since signatures by themselves have no meaning and can be considered as random bit strings, we cannot envision a realistic scenario where this ability can have damaging consequences. Hence we do not consider this an attack on S.
5. If for all i, $y \neq y_i$ and $s \neq s_i$, then this cannot be considered an attack on S since anyone can generate a public/private key pair and then sign any message.

It therefore makes sense to say that an attacker of a signature scheme in the multi-user setting is successful if and only if it can generate an existential forgery or can produce a key substitution.

3 Definitions

We formalize our notions of a signature scheme and its security.

Definition 1 A *signature scheme* S is a triple of algorithm pairs $(\mathcal{D}, \mathcal{D}_V)$, $(\mathcal{G}, \mathcal{G}_V)$, (Σ, Σ_V), where, for security parameter k:

(i) $\mathcal{D}$, the *domain parameter generation algorithm*, is a randomized algorithm which takes as input 1^k and outputs a set D of *domain parameters* to be shared between one or more users, and possibly some state information I required to prove that these parameters satisfy some security requirements. It may be the case that $\mathcal{D}$ simply returns its input, namely 1^k.

(ii) $\mathcal{D}_V$, the *domain parameter validation algorithm*, is a deterministic algorithm which on input a set of domain parameters D, and possibly some state information I, outputs a single bit that specifies whether or not the domain parameters conform to the specified security requirements.

(iii) $\mathcal{G}$, the *key pair generation algorithm*, is a randomized algorithm which on input a set of domain parameters D, outputs a public/private key pair (y, x).
(iv) $\mathcal{G}_V$, the *public key validation algorithm*, is a two-party zero-knowledge protocol. Both parties have as inputs a pair (D, y), where D is a set of valid domain parameters and y is a public key. One party, the prover, also has as input a private key x. The protocol $\mathcal{G}_V$ allows the prover to demonstrate to the second party (the verifier) that y is a valid public key corresponding to the private key x.
(v) Σ, the *signature generation algorithm*, is a randomized algorithm which on input a message m and a secret key x associated with domain parameters D, outputs a *digital signature* s.
(vi) Σ_V, the *signature verification algorithm* is a deterministic algorithm which takes as input a message m, a digital signature s, a valid set D of domain parameters, and a valid public key y, and outputs `True` or `False`.

We require that $\Sigma_V(m, s, D, y) =$ `True` whenever D is a set of valid domain parameters generated by $\mathcal{D}$, y is a valid public key (with corresponding private key x) for D generated by $\mathcal{G}$, and $\Sigma(m, D, x) = s$.

Example 2 (*RSA*) In RSA, the domain parameters consist of the hash function employed and the encryption exponent e if a fixed e (e.g., $e = 3$) is used, or just the hash function if entities can select arbitrary e. In both cases, the domain parameter generation and validation algorithms are trivial. The public key validation algorithm may be a zero-knowledge protocol to ensure that the modulus n is a product of two primes (e.g., [18] together with [9] to show that n is the product of two primes p and q with $p \equiv q \equiv 3 \pmod 4$, or [11] to show that n is a product of two safe primes), followed by a request to RSA-sign a message in order to demonstrate knowledge of the private key.

Example 3 (*DSA*) In DSA, the domain parameters may be primes p and q such that $q|(p-1)$, and an element $g \in \mathbb{Z}_p^*$ of order q. Domain parameter validation verifies that p and q are primes with $q|(p-1)$, and that $2 \leq g \leq p-1$ and $g^q \equiv 1 \pmod p$. A DSA public key is $y = g^x \bmod p$, where $x \in_R [1, q-1]$ is the private key. Key validation may verify that $2 \leq y \leq p-1$ and $y^q \equiv 1 \pmod p$, followed by a request to DSA-sign a message in order to demonstrate knowledge of the private key.

In §2.2, we argued that an adversary of a signature scheme in the multi-user setting is successful if it produces an existential forgery or a key substitution. In either case, the attack is against one particular public key. Therefore, it suffices to consider the multi-user setting in which there is initially only one public key. This is without loss of generality because the adversary attacking one entity can simulate the other entities by picking their public keys so that it knows the corresponding private keys.

Definition 4 An *adversary* E of a signature scheme $S = [(\mathcal{D}, \mathcal{D}_V), (\mathcal{G}, \mathcal{G}_V), (\Sigma, \Sigma_V)]$ is a randomized algorithm which takes as input a set D of valid domain parameters generated by $\mathcal{D}$, a valid public key y for D generated by $\mathcal{G}$, and a signing oracle $\mathcal{O}$ for y. The output of D is a public key $\overline{y}$, a bit string $\overline{x}$, a message m, and a signature $\overline{s}$. The adversary is said to be *successful* if its output $(\overline{y}, \overline{x}, m, \overline{s})$ satisfies the following conditions:

(i) If m has not been asked of $\mathcal{O}$, then $\overline{y} = y$ and $\Sigma_V(m, \overline{s}, D, \overline{y}) = \texttt{True}$.
(ii) If m has been asked of $\mathcal{O}$ returning s then $\overline{s} = s$, $\mathcal{G}_V((D, \overline{y}), (D, \overline{y}, \overline{x})) = \texttt{True}$, and $\Sigma_V(m, \overline{s}, D, \overline{y}) = \texttt{True}$.

The signature scheme S is *secure* if no such polynomial-time adversary exists with non-negligible probability of success.

Remark 5 (*notes on Definition 4*)

(i) In Definition 4, case (i) corresponds to an existential forgery, while case (ii) corresponds to a key substitution attack.
(ii) We make the assumption that if the public key $\overline{y}$ output by E is different from y, then $\overline{y}$ is associated with the same domain parameters D as y. This is without loss of generality since otherwise we can interpret the domain parameters as being part of the public key.

4 Some Secure Signature Schemes

We make the elementary observation that signature schemes secure in the single-user setting can be modified so that they are secure in the multi-user setting.

Theorem 6 Any signature scheme that is secure (in the GMR sense) in the single-user setting can be converted to one that is secure (in the sense of Definition 4) in the multi-user setting.

Proof. Let S be a signature scheme that is GMR-secure in the single-user setting. We define a signature scheme S' to be the same as S except that messages are prepended with the signer's public key in some unambiguous way (e.g., a field of fixed length may be reserved for the public key) prior to signing. Since the adversary E must know the private key corresponding to its public key, E's public key must be different from the public key of the legitimate entity A—otherwise E will have learned A's private key and therefore can produce an existential forgery contradicting the assumption that S is secure in the single-user setting. Key substitutions attacks on S' are then doomed to fail because none of A's formatted messages correspond to E's formatted messages. $\square$

Signature schemes used in practice, e.g., as dictated by standards, typically do not require that a message be formatted together with the public key prior to signing. Therefore, it is of considerable importance to show that signature schemes commonly used in practice are indeed secure in the multi-user setting. In §4.1, §4.2 and §4.3, we prove that the Schnorr, DSA and ECDSA signature schemes are secure with respect to Definition 4 under certain constraints on the domain parameters. In §4.4 we prove that the Rabin-Williams signature scheme is secure with respect to Definition 4.

4.1 Schnorr signature scheme

Recall that Schnorr's signature scheme [25] has parameters a prime p, a prime q dividing $p-1$, and an element $g \in \mathbb{Z}_p^*$ of order q. An entity A's public key is $y = g^x \bmod p$ where $x \in_R [1, q-1]$ is the corresponding private key. To sign a message m, A computes $r = g^k \bmod p$, $e = H(m, r)$ and $s = ex + k \bmod q$, where $k \in_R [1, q-1]$. The signature on m is (s, e). To verify, one computes $r' = g^s y^{-e} \bmod p$, and checks that $e = H(m, r')$.

Theorem 7 Suppose that p, q and g are domain parameters. Suppose that the discrete logarithm problem in $\langle g \rangle$ is intractable. Then, in the random oracle model, the Schnorr signature scheme is secure in the multi-user setting.

Proof. Pointcheval and Stern [23] proved that Schnorr's signature scheme is GMR-secure in the random oracle model assuming that the discrete logarithm problem in $\langle g \rangle$ is intractable. Hence all that remains to show is resistance to key substitution attacks.

Assume that an adversary E successfully outputs a public/private key pair $(\overline{y}, \overline{x})$ different from A's such that (s, e) is both A's and E's signature on a message m. Since both signatures are valid, we have $e = H(m, r)$ where $r = g^s y^{-e} \bmod p$, and $e = H(m, \overline{r})$ where $\overline{r} = g^s \overline{y}^{-e} \bmod p$. Since H is a random function, the probability of obtaining a collision is negligible. Thus we can assume that $\overline{r} = r$. If follows that $y^e \equiv \overline{y}^e \pmod p$ and $y = \overline{y}$, which is a contradiction. □

4.2 DSA

Recall that DSA has parameters a prime p, a prime q with $q|(p-1)$, and an element $g \in \mathbb{Z}_p^*$ of order q. An entity A's public key is $y = g^x \bmod p$ where $x \in_R [1, q-1]$ is the corresponding private key. To sign m, A computes $r = f(g^k \bmod p)$ and $s = k^{-1}(H(m) + xr) \bmod q$, where $k \in_R [1, q-1]$ and $f : \langle g \rangle \to \mathbb{Z}_q$ defined by $f(h) = h \bmod q$ is the *conversion function*. The signature on m is (r, s). To verify, one computes $r' = f(g^{s^{-1}H(m)} y^{s^{-1}r} \bmod p)$ and checks that $r = r'$.

At present, there is no security proof known for DSA against existential forgers. The strongest result we can prove on the security of DSA in the multi-user setting is the following.

Theorem 8 Suppose that p, q and g are domain parameters. Suppose that DSA is GMR-secure and that f is collision resistant. Then DSA is secure in the multi-user setting.

Proof. Suppose that an adversary E successfully outputs a key substitution. That is, on input p, q, g, y, it outputs $(m, r, s, \overline{x}, \overline{y})$ such that $\overline{y} = g^{\overline{x}} \bmod p$, $\overline{y} \not\equiv y \pmod p$, $r = f(g^{s^{-1}H(m)} y^{s^{-1}r} \bmod p)$ and $r = f(g^{s^{-1}H(m)} \overline{y}^{s^{-1}r} \bmod p)$. Since $y \not\equiv \overline{y} \pmod p$, we have

$$g^{s^{-1}H(m)} y^{s^{-1}r} \not\equiv g^{s^{-1}H(m)} \overline{y}^{s^{-1}r} \pmod p.$$

Therefore E has produced a collision for f. $\square$

FIPS 186-2 [21] recommends that q is a randomly selected 160-bit prime, p is a randomly selected 1024-bit prime such that q divides $p-1$, and g is a randomly selected element of order q in $\mathbb{Z}_p^*$. For domain parameters selected in this way, it is reasonable to expect that the conversion function f is indeed collision resistant. Indeed, Nguyen and Shparlinski (see §3.2 of [22]) have proven that asymptotically the range of the conversion function f is exponentially large (although they were not able to show that the range of f has size roughly q). It would be useful to prove that DSA is resistant to key substitution attacks under assumptions that are weaker that the collision resistance of f.

4.3 ECDSA

Recall that ECDSA has parameters a finite field $\mathbb{F}_q$, an elliptic curve E defined over $\mathbb{F}_q$ with $\#E(\mathbb{F}_q) = nh$ where n is prime, and a point $G \in E(\mathbb{F}_q)$ of order n. An entity A's public key is $Q = dG$, where $d \in_R [1, n-1]$ is the corresponding private key. To sign a message m, A computes $r = x(kG) \bmod n$ and $s = k^{-1}(H(m) + dr) \bmod n$, where $k \in_R [1, n-1]$ and $x(kG)$ denotes the integer obtained from the binary representation of the x-coordinate of kG. The signature on m is (r, s). To verify, one computes $r' = x(s^{-1}H(m)G + s^{-1}rQ) \bmod n$ and checks that $r = r'$.

We prove that ECDSA is secure in the multi-user setting when certain constraints are made on the domain parameters.

Theorem 9 Let E be an elliptic curve defined over $\mathbb{F}_q$ such that $n = \#E(\mathbb{F}_q)$ is prime and $n > q$. Let $G \in E(\mathbb{F}_q)$ be a point of order n. Suppose that $\mathbb{F}_q$, E, G and n are domain parameters. Suppose that the hash function H is zero-resistant, collision-resistant, and smooth. Then, in the generic group model, ECDSA is secure in the multi-user setting.

Proof. Brown [10] has proven GMR-security of ECDSA in the generic group model under the stated assumptions on H. Hence all that remains to show is resistance to key substitution attacks.

Assume that an adversary E successfully outputs a public/private key pair $(\overline{Q}, \overline{d})$ different from A's such that (r, s) is both A's and E's signature on a message m. Since both signatures are valid, we have

$$x(s^{-1}H(m)G + s^{-1}rQ) \equiv x(s^{-1}H(m)G + s^{-1}r\overline{Q}) \pmod{n}.$$

Now, since $n > q$, the binary representation of any field element in $\mathbb{F}_q$ (with a reasonable encoding) is less than n. Hence

$$x(s^{-1}H(m)G + s^{-1}rQ) = x(s^{-1}H(m)G + s^{-1}r\overline{Q})$$

and so

$$s^{-1}H(m)G + s^{-1}rQ = \pm(s^{-1}H(m)G + s^{-1}r\overline{Q}).$$

From this last equation and her knowledge of $\overline{d}$, E can easily determine d. Therefore, since there is no efficient algorithm for the discrete logarithm problem in the generic group model [26], we conclude that E cannot produce a key substitution. $\square$

4.4 Rabin-Williams signature scheme

We expect that Rabin-PSS [5] and RSA-FDH with encryption exponent $e = 3$ [5] resist key substitution attacks in the random oracle model, although we were unable to prove this. Instead, we present RW-FDH, a variant of the Rabin-Williams [6, 24, 28] signature scheme that is somewhat analogous to RSA-FDH, and prove that RW-FDH is secure in the multi-user setting.

We first establish some notation. A Rabin-Williams (RW) integer is a k-bit integer n that is the product of two randomly selected $(k/2)$-bit primes p and q with $p \equiv 3 \pmod 8$ and $q \equiv 7 \pmod 8$. Let $(\frac{a}{n})$ denote the Jacobi symbol. Let $\mathbb{Z}_n^* = \{1 \le x \le n-1 : \gcd(x, n) = 1\}$, let $J_n = \{x \in \mathbb{Z}_n^* : (\frac{x}{n}) = 1\}$, and let $Q_n = \{x^2 \bmod n : x \in \mathbb{Z}_n^*\}$ be the set of quadratic residues modulo n.

We collect some elementary facts about Rabin-Williams integers.

Lemma 10 Let $n = pq$ be a Rabin-Williams integer.

(i) If $y \in Q_n$, then there are exactly four solutions in $\mathbb{Z}_n^*$ to $s^2 \equiv y \pmod n$. Exactly two of these solutions, say s_1 and s_2, are in J_n. Furthermore, $s_1 \equiv -s_2 \pmod n$, so that exactly one of s_1 and s_2 is even.
(ii) We have $(\frac{1}{p}) = (\frac{1}{q}) = 1$, $(\frac{-1}{p}) = (\frac{-1}{q}) = -1$, and $(\frac{2}{p}) = -(\frac{2}{q})$.
(iii) For any $x \in \mathbb{Z}_n^*$, exactly one of $\pm x \bmod n$, $\pm 2x \bmod n$ is in Q_n.

Let $H : \{0,1\}^* \to \{0,1\}^{k-1}$ be a hash function. In the following, $h = H(M)$ will be interpreted as an integer and we shall always assume that $\gcd(h, n) = 1$ (since factoring n is considered intractable). Thus we shall view H as a function from $\{0,1\}^*$ to R_{k-1}, where $R_{k-1} = \{0,1\}^{k-1} \cap \mathbb{Z}_n^*$.

Remark 11 If H is a random function and k is large (e.g., $k = 1024$), then it is infeasible to find any $h \in \{0,1\}^*$ such that ch is a perfect (integer) square for some $c \in \{\pm 1, \pm 2\}$.

An entity A's public key is a RW integer n. To sign a message m, A computes $h = H(m)$ and finds the unique $c \in \{\pm 1, \pm 2\}$ for which $e = ch \bmod n$ is in Q_n. A then computes the even square root $s \in J_n$ of e modulo n. The signature on m is (c, s). To verify, one checks that $s \in J_n$ and s is even; one then computes $e = s^2 \bmod n$, $h = c^{-1} e \bmod n$, and checks that $h = H(m)$.

Theorem 12 Suppose that the integer factorization problem for RW integers is intractable. Then, in the random oracle model, the RW-FDH signature scheme is secure in the multi-user setting.

Proof. We first prove that RW-FDH is GMR-secure in the random oracle model, i.e., where H is modeled by a random function. Suppose that E is an efficient forger of RW-FDH. We describe an efficient integer factorization algorithm F which uses E. The input to F is an RW integer n. F now runs E on input n. We can assume without loss of generality that if E makes signing queries m, then it has already made hash query m (since otherwise F can make this query on E's behalf). We also assume that the total number of (distinct) hash queries made is exactly T (since if E makes fewer than T queries, then F can make the remaining hash queries on E's behalf). Finally, we assume that if E outputs a message m and a signature (c, s) then E has already made hash query m (since otherwise F can make this query on E's behalf).

F answers E's queries as follows. F first selects $j \in_R \{1, 2, \ldots, T\}$. F answers the ith hash query m as follows:

(a) If $i \neq j$, then select $u \in_R \mathbb{Z}_n^*$ and compute $s = u^2 \bmod n$; if s is odd, then replace s by $n - s$.
If $i = j$, then select $s \in_R \{x \in \mathbb{Z}_n^* : (\frac{x}{n}) = -1\}$.
(b) Compute $e = s^2 \bmod n$.
(c) Let $C = \{c : c \in \{\pm 1, \pm 2\}$ and $c^{-1}e \bmod n$ is in $R_{k-1}\}$.
(d) If $C = \emptyset$ then go to step (a).
(e) Select $c \in_R C$ and compute $h = c^{-1}e \bmod n$.
(f) Output $H(m) = h$ and store (m, c, s) in a table.

F answers E's signing query m as follows:

(a) Suppose that m was E's ith hash query.
(b) If $i = j$ then F terminates with failure.
(c) Find the entry (m, c, s) in the table and output (c, s) as the signature on m.

Let D be the set of pairs (c, e) where $c \in \{\pm 1, \pm 2\}$, $e \in Q_n$, and $c^{-1}e \bmod n$ is in $\{0, 1\}^{k-1}$. Since $|R_{k-1}| \approx 2^{k-1}$, the set $\{e : (c, e) \in D\}$ has cardinality at least 2^{k-3}. And, since, $|Q_n| \approx \frac{n}{4} \leq 2^{k-2}$, it follows that the probability that $C = \emptyset$ in step (d) is at most $\frac{1}{2}$. By Lemma 10(iii), the map $(c, e) \mapsto c^{-1}e \bmod n$ is a 1-1 correspondence between D and R_{k-1}. Hence F's responses to E's hash queries are indistinguishable from the responses of a true random oracle. It follows that F's responses to E's signing queries (except for the jth query) are also indistinguishable from the responses of a real signer.

Now, suppose that E outputs the valid signature $(\overline{c}, \overline{s})$ on a message m. Then with probability at least $1/T$, m was the jth hash query. Since the signature is valid, $(\frac{\overline{s}}{n}) = 1$, and $H(m) = h$ where $h = \overline{c}^{-1}\overline{e} \bmod n$ and $\overline{e} = \overline{s}^2 \bmod n$. And, since m was the jth hash query, we have $h = c^{-1}e \bmod n$ where $e = s^2 \bmod n$ and (m, c, s) is from the stored table; furthermore $(\frac{s}{n}) = -1$. Thus $\overline{c}^{-1}\overline{e} \equiv c^{-1}e \pmod{n}$, whence $c\overline{s}^2 \equiv \overline{c}s^2 \pmod{n}$. By Lemma 10(iii), we have $c = \overline{c}$, and so $\overline{s}^2 \equiv s^2 \pmod{n}$. Finally, since $(\frac{s}{n}) = -1$ and $(\frac{\overline{s}}{n}) = 1$, we have $s \not\equiv \pm\overline{s} \pmod{n}$ and so $\gcd(s - \overline{s}, n) = 1$ is a non-trivial factor of n. We conclude that RW-FDH is GHR-secure in the random oracle model provided that factoring RW integers is hard.

Suppose now that an adversary E can successfully launch a key substitution attack. On input an RW integer n, E outputs $\overline{n}$, $\overline{p}$, $\overline{q}$, m and (c,s) such that $\overline{n} = \overline{pq}$ is an RW integer, and (c,s) is a valid signature for m with respect to both n and $\overline{n}$. Since (c,s) is a valid signature for m with respect to n, we have $0 \le s \le n-1$, and $h = H(m)$ where $h = c^{-1}e \bmod n$ and $e = s^2 \bmod n$. Similarly, we have $0 \le s \le \overline{n}-1$, and $\overline{h} = H(m)$ where $\overline{h} = c^{-1}\overline{e} \bmod \overline{n}$ and $\overline{e} = s^2 \bmod \overline{n}$. Thus we have $h = \overline{h}$ whence $s^2 = ch \pmod{n}$ and $s^2 \equiv ch \pmod{\overline{n}}$. Now, $\gcd(n, \overline{n}) = 1$ since otherwise E has succeeded in factoring n. Thus $n\overline{n}$ divides $s^2 - ch$. Suppose that $n > \overline{n}$. Then since $0 \le s^2 \le (\overline{n}-1)^2$ and $-2\overline{n} < ch < 2\overline{n}$, we have $-2\overline{n} < s^2 - ch \le \overline{n}^2 + 1$. Finally, since $n\overline{n}$ divides $s^2 - ch$, we have $s^2 = ch$; but, as noted in Remark 11, it is infeasible to find such an h. We conclude that RW-FDH is secure in the random oracle against KS attacks provided that factoring RW integers is hard. □

5 Conclusions

We presented a security definition for signature schemes in the multi-user setting. Signature schemes that are secure in the single-user setting can easily be transformed into schemes that are secure in the multi-user setting by hashing the public key together with the message when computing the message digest. We have shown that the Schnorr, DSA and ECDSA signature schemes are secure in the multi-user setting under reasonable constraints on the domain parameters. We have also shown that the Rabin-Williams signature scheme is secure in the multi-user setting.

References

1. ANSI X9.62, *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, 1999.
2. M. Bellare, A. Boldyreva and S. Micali, "Public-key encryption in a multi-user setting: security proofs and improvements", *Advances in Cryptology—Eurocrypt 2000*, LNCS **1807**, 2000, 259-274.
3. M. Bellare and P. Rogaway, "Entity authentication and key distribution", *Advances in Cryptology—Crypto '93*, LNCS **773**, 1993, 232-249.
4. M. Bellare and P. Rogaway, "Optimal asymmetric encryption—how to encrypt with RSA", *Advances in Cryptology—Eurocrypt '94*, LNCS **950**, 1994, 92-111.
5. M. Bellare and P. Rogaway, "The exact security of digital signatures—how to sign with RSA and Rabin", *Advances in Cryptology—Eurocrypt '96*, LNCS **1070**, 1996, 399-416.
6. D. Bernstein, "A secure public-key signature system with extremely fast verification", *Journal of Cryptology*, to appear.
7. S. Blake-Wilson, D. Johnson, and A. Menezes, "Key agreement protocols and their security analysis", *Proc. 6th IMA International Conference on Cryptography and Coding*, LNCS **1355**, 1997, 30-45.
8. S. Blake-Wilson and A. Menezes, "Unknown key-share attacks on the station-to-station (STS) protocol", *Proceedings of PKC '99*, LNCS **1560**, 1999, 154-170.

9. J. Boyar, K. Friedl and C. Lund, "Practical zero-knowledge proofs: giving hints and using deficiencies", *Journal of Cryptology*, **4** (1991), 185-206.
10. D. Brown, "Generic groups, collision resistance, and ECDSA", preprint, 2001.
11. J. Camenisch and M. Michels, "Proving in zero-knowledge that a number is a product of two safe primes", *Advances in Cryptology—Eurocrypt '99*, LNCS **1592**, 1999, 107-122.
12. R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels", *Advances in Cryptology—Eurocrypt 2001*, LNCS **2045**, 2001, 453-474.
13. R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack", *Advances in Cryptology—Crypto '98*, LNCS **1462**, 1998, 13-25.
14. W. Diffie, P. van Oorschot and M. Wiener, "Authentication and authenticated key exchanges", *Designs, Codes and Cryptography*, **2** (1992), 107-125.
15. R. Gennaro, S. Halevi and T. Rabin, "Secure hash-and-sign signatures without the random oracle", *Advances in Cryptology—Eurocrypt '99*, LNCS **1592**, 1999, 123-139.
16. S. Goldwasser, S. Micali and R. Rivest, "A "paradoxical" solution to the signature problem", *Proceedings of the IEEE 25th Annual Symposium on Foundations of Computer Science*, 1984, 441-448.
17. S. Goldwasser, S. Micali and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks", *SIAM J. Computing*, **17** (1988), 281-308.
18. J. van de Graaf and R. Peralta, "A simple and secure way to show the validity of your public key", *Advances in Cryptology—Crypto '87*, LNCS **293**, 1988, 128-134.
19. J. Håstad, "Solving simultaneous modular equations of low degree", *SIAM Journal on Computing*, **17** (1988), 336-341.
20. D. Johnson, A. Menezes and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)", *International J. Information Security*, **1** (2001), 36-63.
21. National Institute of Standards and Technology, *Digital Signature Standard*, FIPS Publication 186-2, 2000.
22. P. Nguyen and I. Shparlinski, "The insecurity of the digital signature algorithm with partially known nonces", *Journal of Cryptology*, to appear.
23. D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures", *Journal of Cryptology*, **13** (2000), 361-396.
24. M. Rabin, 'Digitalized signatures and public-key functions as intractable as factorization", MIT Lab. for Computer Science, Technical Report LCS/TR-212, 1979
25. C. Schnorr, "Efficient signature generation by smart cards", *Journal of Cryptology*, **4** (1991), 161-174.
26. V. Shoup, "Lower bounds for discrete logarithms and related problems", *Advances in Cryptology—Eurocrypt '97*, LNCS **1233**, 1997, 256-266.
27. V. Shoup, "On formal models for secure key exchange", Cryptology ePrint Archive Report 1999/012, 1999. Available from http://eprint.iacr.org/1999/.
28. H. Williams, "A modification of the RSA public-key encryption procedure", *IEEE Transactions on Information Theory*, **26** (1980), 726-729.