

Factorization-based Fail-Stop Signatures Revisited

Katja Schmidt-Samoa

Technische Universität Darmstadt, Fachbereich Informatik,
Hochschulstr. 10, D-64283 Darmstadt, Germany
`samoa@informatik.tu-darmstadt.de`

June 23, 2004

Abstract. Fail-stop signature (FSS) schemes are important primitives because in a fail-stop signature scheme the signer is protected against unlimited powerful adversaries as follows: Even if an adversary breaks the scheme's underlying computational hard problem and hence forges a signature, then with overwhelming probability the signer is able to prove that a forgery has occurred (i.e. that the underlying hard problem has been broken). Although there is a practical FSS scheme based on the Discrete Logarithm problem, no provable secure FSS scheme is known that is based on the pure factorization problem (i.e. the assumption that integer factoring for *arbitrary* integers is hard). To be more concrete, the most popular factorization based FSS scheme relies on the assumption that factoring a special kind of Blum integers is intractable. All other FSS schemes related to integer factoring are based on even stronger assumptions or insecure. In this paper, we first cryptanalyze one of those schemes and show how to construct forged signatures that don't enable the signer to prove forgery. Then we repair the scheme at the expense of a reduced message space. Finally, we develop a new provable secure scheme based on the difficulty of factoring integers of the shape p^2q for primes p, q .

Keywords: Fail-stop Signature schemes, Provable Security, Cryptanalysis of Fail-stop Signature schemes, Bundling Homomorphisms

1 Introduction

Digital signatures were introduced to replace handwritten signatures in the electronic world. The security of traditional signature schemes relies on a computational assumption. Provided that this assumption holds, no one but the owner of a secret key should be able to produce signatures that can be verified using the corresponding public key. But an adversary who breaks this assumption is able to sign any message of his choice such that the signatures pass the verification test, and the signer Alice has no chance to convince the recipient Bob (or a judge) that a forged signature has not been created by herself. To overcome this problem, fail-stop signature (FSS) schemes were invented. In a FSS scheme, the signer is protected against computationally unbounded adversaries in the following sense: If the signer sees a forged signature (i.e. a signature passing the verification test but not created by herself), then with overwhelming probability the signer is able to prove that the underlying computational assumption has been broken and the protocol is stopped (hence

the name fail-stop signature). Of course, a signer Alice who breaks the underlying problem herself may exploit this mechanism to produce signatures which she later proves to be forgeries, i.e. she can sign messages and disavow the signatures later. Therefore, the security of the recipients of fail-stop signatures against a cheating signer is still based on computational assumptions, whereas the signer is unconditionally secure¹. As a consequence, FSS schemes are particularly suitable in asymmetric constellations, where the recipient (e.g. a bank) is assumed to be much more powerful than the signer (e.g. a single customer).

1.1 Previous Work

In this paper, we focus on FSS schemes where the underlying problem is related to the integer factorization problem. Unfortunately, while there is an efficient FSS scheme based on the discrete logarithm problem [vHP93], the situation regarding factorization based FSS schemes is less satisfying. In 1991, the first factorization based FSS scheme has been published [BPW91] (see [PP97] for a revised version). This scheme – called the quadratic residue scheme in the following – is based on the intractability of factoring integers $n = pq$ for primes p, q with $p = q = 3 \pmod 4$ (i.e. Blum integers) and $p \not\equiv q \pmod 8$ (see Appendix A for a review of this FSS scheme). Until today, it is unknown if factoring integers of this special form is as hard as factoring arbitrary RSA-moduli. In addition, this scheme is quite complicated and the structure is not a “natural” one (the construction is defined in a way that the proofs work, but it looks cumbersome at the first sight). Nevertheless, the quadratic residue scheme is the only previously known provable secure FSS scheme that is based on the factorization assumption only. All other factorization related FSS schemes are based on stronger assumptions or insecure. The first of these is [SSNP99], it is based on the factorization assumption but unfortunately turned out to be not provable secure (see [SSN03]).

The second scheme [SSNGS00] – referred to as the order scheme in the following – is based on the so-called strong factorization assumption, which states that it is hard to factor $n = pq$ even if an element $g \in (\mathbb{Z}/P\mathbb{Z})^\times$ (with P prime and $n|P - 1$) of multiplicative order p is known. We will show that it is insecure.

The most recent one is [SSN03], which is in fact an analogon of the discrete logarithm scheme [vHP93]. The basis of the scheme from [vHP93] is formed by two primes p, q with $q|p - 1$, and its security is based on the subgroup discrete logarithm problem related to the field $(\mathbb{Z}/p\mathbb{Z})^\times$ and the subgroup (of the multiplicative group) generated by g . In [SSN03], the only difference is that the field $(\mathbb{Z}/p\mathbb{Z})^\times$ is replaced by $(\mathbb{Z}/n\mathbb{Z})^\times$, where n is an RSA modulus. Consequently, this scheme is based on the subgroup discrete logarithm problem, too, and not on factoring as stated. In particular, there is no reduction that breaking this scheme enables to factor n . The only connection to factoring is that the knowledge of the factors of n may weaken the schemes security, but this is of course not the meaning of “factorization based”. Therefore we exclude the scheme [SSN03] from subsequent considerations.

¹ Note that this situation is dual to ordinary signature schemes, where the recipients are unconditionally secure and the signer’s security relies on a computational assumption.

1.2 Our contributions

In this paper, first we cryptanalyze the order scheme [SSNGS00] and show that it is not secure for the signer. Then we show how to repair the scheme at the expense of a smaller message space. But our major aim is the development of a new factorization-based FSS scheme. The proposed scheme is the first scheme based on the intractability of factoring integers of p^2q type. Moduli of this special have become more and more important during the last years [FOM91,FKM⁺,OU98,Tak98,Tak04]. The basis of our quite simple construction is a group homomorphism with the property that the problem of collision-finding is reduced to the factoring problem, which may be of interest on its own. Thus our new scheme provides a good alternative to the quadratic residue scheme. We will show a complete security proof for the new scheme.

2 Preliminaries

2.1 Notations

Let n be a positive integer. We write $\mathbb{Z}/n\mathbb{Z}$ for the ring of residue classes modulo n , and $(\mathbb{Z}/n\mathbb{Z})^\times$ for its multiplicative group. For $x \in (\mathbb{Z}/n\mathbb{Z})^\times$, the term $\text{ord}_n(x)$ denotes the multiplicative order of x modulo n .

As usual, a probability $\Pr(k)$ is called *negligible* if $\Pr(k)$ decreases faster than the inverse of any polynomial in k , i.e. $\forall c \exists k_c (k > k_c \Rightarrow \Pr(k) < k_c^{-c})$. In contrast, a probability $\Pr(k)$ is called *overwhelming*, if $1 - \Pr(k)$ is negligible.

Finally, $|n|_2$ denotes the bit-length of n .

2.2 Definitions

In this section, we briefly review the basic definitions related to FSS schemes. For a comprehensive treatment (including formal details that are omitted in this paper for better readability) see [PP97,PW90,vHP93]. As ordinary signature schemes, FSS schemes consist of algorithms for key-generation, signing and signature testing. A signature passing the signature test is called *acceptable* signature in the following. Furthermore, to achieve the above mentioned extended security for the signer, there is an algorithm for proving that a forgery has been occurred and an algorithm for verifying forgery proofs. In addition to the usual correctness requirements – e.g. that each correctly generated signature passes the verification test – a secure FSS scheme has to fulfill two different security properties:

- If an adversary creates an acceptable signature, then with overwhelming probability the signer is able to present a valid proof of forgery. This property is referred to as *security for the signer* and it is not based on a computational assumption.
- A computationally bounded signer should not be able to create signatures that she later can prove to be forgeries. This property is referred to as *security for the recipients* and it relies on the scheme’s underlying hard problem.

In this paper, we call a signature that can be proven to be a forgery a *provable forgery*. The two security requirements have to be understood as independent and hence there are two different security parameters σ (related to the signer’s security) and k (related to the recipient’s security). The success probability for an unbounded adversary to create non-provable forgeries is upper-bounded by $2^{-\sigma}$, whereas the success probability for a cheating signer is a negligible function in k . Note that besides the possibility of proving forgeries, a signature scheme where forging signatures is easy does not make sense. Fortunately, it can be proven [PP97] that the above security requirements already imply that FSS schemes meet the strongest notion of security related to traditional signature schemes: existential unforgeability under adaptive chosen message attacks. This proof can be sketched as follows: Assume that \mathcal{A} is a successful attacker. Then Alice runs \mathcal{A} , responds all signature queries using her secret key, and finally \mathcal{A} returns a new acceptable message/signature pair. The security for the signer implies that Alice can construct a proof of forgery for this pair. Hence Alice is able to publish this signature and to disavow it later, contradicting the security for the recipients.

Another consequence of the signer’s ability of disavowing forged signatures is that the key generation becomes slightly more complex than in ordinary signature schemes. In ordinary signature schemes, the key generation usually is a two-party protocol between the signer and a center. In FSS schemes, a good key must guarantee both, the signer’s and the recipient’s security. Therefore it is necessary that the recipient (or a third party trusted by the recipient) is involved in the key generating process. For simplicity, we only speak of a *center* in the following, capturing the cases that the center is a trusted third party, a recipient or a risk-bearer like an insurance that suffers damages if the recipient accepts invalid signatures. It is also possible to extend this model to multiple recipients (see [PP97]). Hence in general the key generation is again a two party protocol between the signer and a center. To simplify the situation if there are several signers, we only consider *FSS schemes with pre-key*. In this case, first the center generates a pre-key on his own and publishes it. Then each signer carries out a two-party protocol with the center to verify that the pre-key is “good” and finally, each signer creates her key-pair consisting of public/private key individually and publishes the public key. In the basic variant, FSS schemes with pre-key are one-time signature schemes, i.e. for each message to be signed, the signer has to generate a fresh key-pair. However, it is possible to extend this variant to sign multiple messages [vHP93,BP97].

Thus, a FSS scheme with pre-key is composed of the following components:

- KeyGen** is a triple (PreKeyGen, PreKeyVerify, MainKeyGen) consisting of
- a probabilistic polynomial time algorithm PreKeyGen, that is executed by the center and creates a pre-key $prek$ with the security parameters σ, k as inputs,
 - a two-party protocol PreKeyVerify that is carried out between the center and each signer to convince the signer that $prek$ indeed fulfills the properties necessary for the signer’s security. The probability that a signer accepts a bad pre-key must be smaller than $2^{-\sigma}$, and
 - a polynomial time algorithm MainKeyGen that is executed by the signer to create a pair (sk, pk) of secret and public key on the input $prek$.

Sign is a polynomial time algorithm that creates signatures using the secret key sk .

Test is a polynomial time algorithm that can be executed by anyone knowing the public key pk for testing if a signature is acceptable.

Prove is a polynomial time algorithm that on input (sk, m, s) , where s is an acceptable signature on m , outputs “not a forgery” or a bit-string *proof*. The output “not a forgery” indicates that it is impossible to construct a proof of forgery. The probability that an adversary knowing $prek, pk$ and one acceptable signature/message pair (s^*, m^*) is able to create an acceptable signature/message pair (s, m) with $s \neq s^*, m \neq m^*$ and $\text{Prove}(sk, (m, s)) = \text{“not a forgery”}$ must be smaller than $2^{-\sigma}$.

Verify is a polynomial time algorithm that on input $(prek, m, s, proof)$, where s is an acceptable signature on m , outputs “accept” or “reject”. The output “accept” indicates that s is regarded as a forged signature and that the scheme should be stopped. The probability that an adversary knowing $prek, pk$ and one acceptable signature/message pair (s^*, m^*) is able to create an acceptable signature/message pair (s, m) with $s \neq s^*, m \neq m^*$ and $\text{Verify}(prek, (m, s), \text{Prove}(sk, (m, s))) = \text{“reject”}$ must be smaller than $2^{-\sigma}$. The probability that a polynomially bounded signer knowing $prek$ is able to construct a key-pair sk, pk , an acceptable signature/message pair (s, m) and a string *proof* with $\text{Verify}(prek, (m, s), proof) = \text{“accept”}$ must be negligible in k .

Note that there are general methods of verifying a pre-key that work independent from particular FSS schemes [PW90]. Therefore, a description of the protocol PreKeyVerify may be omitted when specifying a concrete FSS scheme.

3 A General Construction using Bundling Homomorphisms

In this section, we review a method of constructing FSS schemes with pre-key from any family of bundling homomorphisms. Bundling homomorphisms can be understood as a special kind of collision-resistant hash functions.

Definition 1 (Bundling homomorphism). *Let $(G, +, 0)$ and $(H, *, 1)$ be two Abelian groups and τ, k natural numbers. The function $h : G \rightarrow H$ is called a (τ, k) -bundling homomorphism iff the following three properties are fulfilled:*

1. *h is a group homomorphism.*
2. *Each $y \in \text{im}_h(G) \subseteq H$ has at least 2^τ pre-images. 2^τ is called the bundling degree of h .*
3. *It is hard (measured in k) to find collisions, i.e. for each polynomial time probabilistic algorithm A the probability that A on input G, H and h outputs $x, y \in G$ with $h(x) = h(y)$ is a negligible function in k .*

Loosely speaking, a family of bundling homomorphisms is a collection of “computational friendly” bundling homomorphisms that is indexed by a key.

Definition 2 (Family of Bundling Homomorphisms). *A family of bundling homomorphisms is a quadruple $\mathcal{B} = (g, h, \mathcal{G}, \mathcal{H})$ where g – the key generator – is a polynomial time algorithm and for each pair of natural numbers (τ, k) holds the following: g on the input (τ, k) outputs a key K that determines Abelian groups $(G_K, +, 0) \in \mathcal{G}, (H_K, *, 1) \in \mathcal{H}$. The restriction h_K of h to G_K is a (τ, k) -bundling homomorphism on H_K . Furthermore, there must be polynomial time algorithms for*

- computing the group operations in G_K and H_K ,
- testing membership in G_K and H_K , and
- selecting elements of G_K uniformly at random.

For better readability, we omit the subscript K whenever it is clear from the context. It was first pointed out by Pedersen and Pfitzmann [PP97] that families of bundling homomorphisms can be used to construct provable secure FSS schemes with pre-key as follows:

General construction using bundling homomorphisms

Let \mathcal{B} be a family of bundling homomorphisms with key generating function g and let σ, k be two FSS security parameters. Then the components of a FSS scheme that is provable secure according to σ and k are given as follows:

KeyGen: Define τ according to σ (details are given later). Then run g on τ, k to obtain a (τ, k) -bundling homomorphism $h : G \rightarrow H$. The groups G, H and h will serve as the pre-key.

For pre-key verification, the signer has to be convinced that h is a group homomorphism with bundling degree 2^τ (e.g. using a zero-knowledge proof^a).

Finally, the signer chooses two elements sk_1, sk_2 uniformly at random from G and computes $pk_i = h(sk_i), i = 1, 2$. This determines the secret key $sk = (sk_1, sk_2)$ and the public key $pk = (pk_1, pk_2)$.

Sign: The message space \mathcal{M} is a suitable subset of \mathbb{Z} . To sign a message $m \in \mathcal{M}$, the signer computes

$$\text{sign}(sk, m) := sk_1 + msk_2,$$

where msk_2 has to be understood as applying m times the group operation in G on sk_2 .

Test: An element $s \in G$ is an acceptable signature on $m \in \mathcal{M}$ iff $h(s) = pk_1 * pk_2^m$ holds, where pk_2^m has to be understood as applying m times the group operation in H on pk_2 .

Prove: Assume that s^* is an acceptable signature on m that the signer wants to prove to be a forgery. To do so, the signer computes her own signature $s = sk_1 + msk_2$ on m . If $s = s^*$ holds, then the proof of forgery fails, otherwise (s, s^*) is the proof of forgery.

Verify: A pair $(x, x') \in G \times G$ forms a valid proof of forgery iff $x \neq x'$ and $h(x) = h(x')$ hold.

^a Note that there is no need to prove the collision-resistance of h , because the signer's security does not depend on this property.

Note that because of the homomorphic properties of h , each correctly generated signature passes the signature test. Following the above construction, the security for the recipients is reduced on the problem of finding collisions of the bundling homomorphism.

Next, we try to explain the idea behind this construction. Assume that a signer Alice and a center follow the general construction. The crucial point is that because of property 2, Definition 1, there are at least $2^{2\tau}$ possible secret keys $sk' = (sk'_1, sk'_2)$ matching Alice's public key pk (in the sense that $h(sk'_i) = pk_i, i = 1, 2$). Each of these keys produces acceptable signatures. Therefore, an adversary \mathcal{A} with unbounded computational power may be able to invert h and to find secret keys matching Alice's public key, but \mathcal{A} does not know which of the $2^{2\tau}$ possibilities is in fact

Alice's secret key. However, the knowledge of a signature/message pair (s, m) correctly generated by Alice provides \mathcal{A} with some extra information. In particular, as the equation $sk_1 = s - msk_2$ must hold in G , the number of possible secret keys reduces to 2^τ . Alice is able to present a valid proof of forgery if the forged signature on a message m^* differs from her own signature on m^* . Consequently, to measure \mathcal{A} 's probability of generating an unprovable forgery, we must analyze the number of pairwise different signatures on m^* that can be produced using these 2^τ possible secret keys. As some easy implications show (see [PP97]), this number is upper-bounded by the magnitude of the set

$$T := \{d \in G \mid h(d) = 1 \wedge (m - m^*)d = 0\} \quad (1)$$

In order to upperbound \mathcal{A} 's success probability, we have to consider the worst case, i.e. we must find the maximum number taken over all possible messages $m^* \neq m$. Hence we obtain the following set

$$T_{max} := \max_{0 \neq m' \in \mathcal{M}} \{d \in G \mid h(d) = 1 \wedge m'd = 0\} \quad (2)$$

Indeed, we have the following theorem (see [PP97] for a proof):

Theorem 1 (Security of the general construction). *Let σ, k be security parameters and let \mathcal{B} be a family of bundling homomorphisms. Let \mathcal{F} be a FSS scheme following the general construction above. Then we have*

- a) \mathcal{F} is k -secure for the recipients.
- b) If the bundling degree 2^τ is chosen such that $T_{max}/2^\tau \leq 2^{-\sigma}$, then \mathcal{F} is σ -secure for the signer.

Consequently, the general construction offers a convenient tool for designing FSS schemes. Actually, to describe a FSS scheme based on the general construction, it is sufficient to specify a family of bundling homomorphisms and to determine the bundling degree and the number $T_{max}/2^\tau$. In particular, the underlying hard problem of the scheme equals the problem of collision-finding in the family of bundling homomorphisms. The above construction is the basis of all previously known provable secure FSS schemes [PP97,SSNGS00,SSN03].

4 Cryptanalysis of the Order Scheme

In this section, we focus on the FSS scheme from [SSNGS00] – referred to as the order scheme in this paper – and show that it is not secure for the signer. The reason for this is that the general construction was not applied carefully enough. A similar problem occurred in [SSNP99], that was already shown to be not provable secure in [SSN03], but for a different reason. Hence the aim of this section is to warn the interested reader not to make the same mistake.

4.1 Review of the Order scheme

The order scheme is an instance of the general constructions. We describe the key generating function g that is used in the order scheme to determine the bundling homomorphism: On the input (τ, k) , g chooses two primes q, p with $|p| \approx |q| \approx k/2$, a prime P such that $n = pq$ divides $P - 1$ and an element $\alpha \in (\mathbb{Z}/P\mathbb{Z})^\times$ of multiplicative order p . Consider the Abelian groups $G = (\mathbb{Z}/n\mathbb{Z}, +, 0)$ and $H = ((\mathbb{Z}/P\mathbb{Z})^\times, *, 1)$. The bundling homomorphism h is defined as

$$\begin{aligned} h : \mathbb{Z}/n\mathbb{Z} &\longrightarrow (\mathbb{Z}/P\mathbb{Z})^\times \\ x &\mapsto \alpha^x \bmod P \end{aligned}$$

It is shown in [SSNGS00] that this defines a family of bundling homomorphisms with bundling degree $2^{k/2}$ under the so-called *Strong Factorization Assumption*:

Given n as a product of two nearly equal-sized primes p and q , $P = nt + 1$ (where $t \in \mathbb{N}$ and P is also prime) and α (where $\text{ord}_P(\alpha) = p$), it is hard to find a non-trivial of n .

Note that this assumption is in fact stronger than the factoring assumption, because there is no proof that knowledge of α does not weaken the hardness of factoring. In particular, this assumption is broken if it is possible to find the order of elements $\alpha \in (\mathbb{Z}/P\mathbb{Z})^\times$ with the additional informations that this order is prime and that it is a factor of the 2-factor number n . Indeed, this assumption is considered as “quite unnatural” by Victor Shoup [Sho99].

The definitions of sign, test, proof, and verify follow the general construction. The message space \mathcal{M} equals $\{0, 1, \dots, n - 1\}$.

4.2 How to break the Order Scheme

In [SSNGS00] there is a proof that the order scheme is secure for the signer by showing that $\#T = 1$. However, in order to evaluate the signer’s security, we have to take into account T_{max} instead of $\#T$, i.e. we have to find the *maximum* size of T taken over all messages an adversary could try to forge. Therefore the security proof from [SSNGS00] is not sound. Indeed, consider the following attack on the signer’s security: Assume that Alice’s secret key equals $(sk_1, sk_2) \in \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$. The corresponding public key $(pk_1, pk_2) \in (\mathbb{Z}/P\mathbb{Z})^\times \times (\mathbb{Z}/P\mathbb{Z})^\times$ is defined as

$$pk_1 = \alpha^{sk_1} \bmod P, \quad pk_2 = \alpha^{sk_2} \bmod P. \quad (3)$$

Let (s, m) be a signature/message pair that Alice has created using her secret keys (sk_1, sk_2) , i.e.

$$s = sk_1 + msk_2 \bmod n.$$

We construct a computationally unbounded adversary \mathcal{A} who is able to compute unprovable forged signatures. Remember that an acceptable signature s^* on a message $m^* \neq m$ is unprovable if s^* equals Alice’s own signature on m^* . Let $m^* \in \mathcal{M}, m^* \neq m$ be any message with $q|m - m^*$, i.e.

$$m^* = m + qx \text{ for a suitable integer } x. \quad (4)$$

First, \mathcal{A} solves the discrete logarithm problem (3) in $(\mathbb{Z}/P\mathbb{Z})^\times$ and obtains $sk'_2 \in \mathbb{Z}/n\mathbb{Z}$ such that $pk_2 = \alpha^{sk'_2} \bmod P$ holds. This is feasible because \mathcal{A} has unlimited computational power. As the multiplicative order of α equals p , we have

$$sk'_2 = sk_2 \bmod p. \quad (5)$$

In the same manner and with help of the Chinese Remainder Theorem, \mathcal{A} constructs $sk'_1 \in \mathbb{Z}/n\mathbb{Z}$ with

$$sk'_1 = sk_1 \bmod p \quad (6)$$

and

$$sk'_1 = s - msk'_2 \bmod q. \quad (7)$$

The key-pair (sk'_1, sk'_2) can be used to construct signatures that Alice cannot prove to be forgeries:

Lemma 1. *Define $s^* = sk'_1 + m^*sk'_2 \bmod n$. Then s^* is an unprovable forgery on m^* .*

Proof. First note that (5),(6) and (7) imply

$$s = sk_1 + msk_2 = sk'_1 + msk'_2 \bmod n. \quad (8)$$

We show that s^* equals Alice's signature on m^* (namely $sk_1 + m^*sk_2 \bmod n$):

$$\begin{aligned} s^* &= sk'_1 + m^*sk'_2 \stackrel{(4)}{=} sk'_1 + msk'_2 + qxsk'_2 \stackrel{(8)}{=} sk_1 + msk_2 + qxsk'_2 \\ &\stackrel{(5)}{=} sk_1 + msk_2 + qxsk_2 \stackrel{(4)}{=} sk_1 + m^*sk_2 \bmod n \end{aligned}$$

Thus Alice's signature on m^* equals the forged signature and Alice can't construct a valid proof of forgery. \square

Consequently, we have the following theorem:

Theorem 2. *The order scheme is not secure for the signer.*

4.3 How to repair the Order Scheme

A possible countermeasure is to reduce the message space \mathcal{M} to $\{0, 1, \dots, q-1\}$. In this case, the security proof provided in [SSNGS00] becomes sound. Unfortunately, this reduction deprives the order scheme of its merits, namely that it has been the most efficient scheme with respect to the ratio of message length to signature length.

5 A New Factorization-based Scheme

In this section, we introduce a new factorization-based FSS scheme and provide a complete security proof. We claim that the proposed construction is more simple and elemental than the artificial construction defining the quadratic residue scheme. For the purpose of enabling the reader to form his own opinion, we review the quadratic residue scheme in Appendix A.

5.1 Some Ideas that don't work

As shown in Section 3, the main thing to do is finding a family of bundling homomorphisms. As we are interested in factorization-based FSSs, we have to search for a bundling homomorphism with the property that finding collisions enables to factor. The most obvious homomorphism in that context is squaring modulo n , because it is well-known that knowledge of $x, y \in (\mathbb{Z}/n\mathbb{Z})^\times$ with $x^2 = y^2 \pmod n$ leads to nontrivial factors of n , provided $x \neq \pm y \pmod n$ holds. Trivial collisions (i.e. $x = -y \pmod n$) can be eliminated by defining G not as $(\mathbb{Z}/n\mathbb{Z})^\times$ but as $(\mathbb{Z}/n\mathbb{Z})^\times$ modulo the subgroup generated by -1 . But unfortunately, the bundling degree in this case is only 2, therefore by far too small. Nevertheless, the quadratic residue scheme from [PP97] is based on the principle factoring with congruent squares. The bundling degree is enlarged by clever exploiting the concept of claw-free permutation pairs [GMR88].

A further idea (used by the order scheme) is to find a multiplicative group that contains elements of multiplicative order p and to define the homomorphism on input $x \in (\mathbb{Z}/n\mathbb{Z})^\times$ as raising such an element to the power x . The bundling degree for this construction is $n/p = q$. But the problem with this attempt is that in general one gets stronger assumptions than the pure factorization assumption, because knowledge of an element of order p may reduce the hardness of factoring. As a demonstrative example, consider the group $(\mathbb{Z}/n^2\mathbb{Z})^\times$. This group is of order $\varphi(n^2) = n\varphi(n)$ and contains elements of order p , say g_p . But designing a FSS scheme based on the bundling homomorphism $x \mapsto g_p^x \pmod{n^2}$ is actually a quite poor idea, because each element of order p in $(\mathbb{Z}/n^2\mathbb{Z})^\times$ reveals q , the second secret factor². Thus this scheme would be completely insecure. Similar considerations hold if we choose $G = (\mathbb{Z}/n\mathbb{Z})^\times$, where n is of the shape p^2q for primes p and q . There are elements of order p in $(\mathbb{Z}/n\mathbb{Z})^\times$, but the knowledge of such an element enables the signer to factor n . But in this case, we can make a virtue of necessity: If we define the bundling homomorphism as exponentiation with n , then a collision leads to an element of order p , and hence to the factorization of n . This is the basis of our proposed scheme.

5.2 The Proposed Scheme

Our proposed scheme is an instance of the general construction. For the sake of completeness, we give the full description of our proposed scheme in the one recipients model. For the extensions to multiple recipients see [PP97].

KeyGen: On the input σ, k the center chooses two equally sized primes p, q of approximate bit-length $\tau := \max(\sigma, k/3)$. The Abelian groups according to $n = p^2q$ are given as

$$G = H = ((\mathbb{Z}/n\mathbb{Z})^\times, *, 1).$$

The bundling homomorphism h is defined by

$$\begin{aligned} h : (\mathbb{Z}/n\mathbb{Z})^\times &\longrightarrow (\mathbb{Z}/n\mathbb{Z})^\times \\ x &\mapsto x^n \pmod n \end{aligned}$$

² It is easy to show that g_p must be of the shape $1 + kqn$ for a suitable $0 < k < p$.

The groups G, H and the homomorphism h will serve as the pre-key.

For pre-key verification, it is sufficient to assure the signer that n possesses a squared factor of approximate bit-length σ (e.g. using a zero-knowledge proof).

Finally, the signer chooses two elements $sk_1, sk_2 \in (\mathbb{Z}/n\mathbb{Z})^\times$ uniformly at random and computes $pk_i = sk_i^n \bmod n, i = 1, 2$. This determines the secret key $sk = (sk_1, sk_2)$ and the public key $pk = (pk_1, pk_2)$.

Sign: The message space is defined as $\mathcal{M} = \{0, 1, \dots, p-1\}$. To sign a message $m \in \mathcal{M}$, the signer computes

$$\text{sign}(sk, m) := sk_1 * sk_2^m \bmod n.$$

Test: An element $s \in (\mathbb{Z}/n\mathbb{Z})^\times$ is an acceptable signature on $m \in \mathcal{M}$ iff $s^n = pk_1 * pk_2^m \bmod n$ holds.

Prove: Assume that s^* is an acceptable signature on m that the signer wants to prove to be a forgery. To do so, the signer computes her own signature $s = sk_1 * sk_2^m \bmod n$ on m . If $s = s^*$ holds, then the proof of forgery fails, otherwise (s, s^*) is the proof of forgery.

Verify: A pair $(x, x') \in (\mathbb{Z}/n\mathbb{Z})^\times \times (\mathbb{Z}/n\mathbb{Z})^\times$ forms a valid proof of forgery iff $x \neq x'$ and $x^n = x'^n \bmod n$ hold.

The underlying assumption is the p^2q Factorization Assumption:

Given $n = p^2q$ where p and q are equally sized primes, it is hard to factor n .

In Appendix B, we discuss the hardness of the p^2q factorization problem.

To prove that g indeed generates a family of bundling homomorphisms, we need the following lemma:

Lemma 2. Let p, q be primes and $n = p^2q$. Define the set \mathcal{S} as

$$\mathcal{S} := \{x \in (\mathbb{Z}/n\mathbb{Z})^\times \mid x = 1 + kpq \text{ for an integer } k, 0 < k < p\}.$$

Then \mathcal{S} consists of exactly the elements of multiplicative order p in $(\mathbb{Z}/n\mathbb{Z})^\times$.

Proof. Let x be an element of multiplicative order p in $(\mathbb{Z}/n\mathbb{Z})^\times$. Then we have

$$\begin{aligned} x^p = 1 \bmod n &\Rightarrow (x^p = 1 \bmod p \wedge x^p = 1 \bmod q) \\ &\Rightarrow (x = 1 \bmod p \wedge x = 1 \bmod q). \end{aligned}$$

Hence $pq \mid x - 1$ must hold, and we conclude $x \in \mathcal{S}$.

On the other hand, it is obvious that for all $x \in \mathcal{S}$ we have $x^p = 1 \bmod n \wedge x \neq 1$, thus the assertion follows. \square

We have the following theorem:

Theorem 3 (Factoring bundling homomorphisms). Under the p^2q Factorization Assumption, the construction above is a family of bundling homomorphisms with bundling degree 2^τ .

Proof. It is obvious that h is a homomorphism. To analyze the bundling degree, we determine the kernel $\ker(h)$. Note that as p is the only non-trivial common factor of n and $\varphi(n) = n(p-1)(q-1)$, we must have

$$x^n = 1 \pmod n \iff x = 1 \vee \text{ord}_n(x) = p \quad (9)$$

Hence the kernel of h consists of 1 and exactly the elements of multiplicative order p in $(\mathbb{Z}/n\mathbb{Z})^\times$, i.e the elements of \mathcal{S} as defined in Lemma 2. Consequently, we have $\#\ker(h) = p$, which equals the bundling degree because h is a homomorphism.

It remains to show that h is collision resistant under the p^2q Factorization Assumption. Assume that \mathcal{A} is a polynomial time algorithm that determines $x, y \in (\mathbb{Z}/n\mathbb{Z})^\times$ with $x \neq y$ and $h(x) = h(y)$. In particular, we have $x^n = y^n \pmod n$, leading to $(xy^{-1})^n = 1 \pmod n$. As $x \neq y \pmod n$ holds, from eq. (9) we conclude $\text{ord}_n(xy^{-1}) = p$, and thus Lemma 2 tells us $\text{gcd}((xy^{-1} \pmod n) - 1, n) = pq$, which completely reveals the factorization of n . \square

Theorem 3 implies the first part of the security proof for the new scheme:

Corollary 1 *Under the p^2q Factorization Assumption, the proposed scheme as defined above is secure for the recipients.*

To complete the security proof, we show the following theorem:

Theorem 4. *The proposed scheme as defined above is secure for the signer.*

Proof. According to Theorem 1, we have to determine the number

$$T_{max} := \max_{0 \neq m' \in \mathcal{M}} \{d \in G \mid h(d) = 1 \wedge m'd = 0\}$$

and show that $T_{max}/2^\tau \leq 2^{-\sigma}$ is fulfilled. Putting in the parameters of the proposed scheme, we obtain

$$T_{max} = \max_{0 < m' < p} \{d \in (\mathbb{Z}/n\mathbb{Z})^\times \mid d^n = 1 \pmod n \wedge d^{m'} = 1 \pmod n\} = 1.$$

Hence we conclude $T_{max}/2^\tau \leq 2^{-\sigma}$, because τ was chosen as the maximum of σ and k . \square

Note that unfortunately it is not possible to construct an analogue FSS scheme for general n . Though in $(\mathbb{Z}/n^2\mathbb{Z})^\times$ there are elements of order p and knowledge of these elements reveals the factorization of n , a collision of the homomorphism $x \mapsto x^n \pmod{n^2}$ not necessary leads to an element of order p . The problem is that $(\mathbb{Z}/n^2\mathbb{Z})^\times$ contains elements of order n , too, and a collision leading to an element of order n is useless for factoring.

Table 1 provides a detailed comparison of the quadratic residue scheme and the proposed one.

Due to the interaction of the different parameters, a general evaluation is difficult. As a rough guideline, in case of $k > \sigma$, the proposed scheme outperforms the quadratic residue scheme in most points, whereas in case of $k < \sigma$ the quadratic residue scheme is more advantageous. But in both cases, the differences are not large.

Concluding we remark that the new scheme is conceptually easier and more natural than the quadratic residue scheme. As it requires roughly the same amount of computing and storage, it provides a good alternative to the quadratic residue scheme (based on a different factorization assumption).

	Quadratic Residue Scheme	Proposed Scheme
Signer's security	σ	σ
Recipient's security	k	k
Message length	ρ	$\rho = \max(\sigma, k/3)$
Sig. length	$\sigma + \rho + k$	$3\rho = \max(3\sigma, k)$
Length of pk	$2k$	$6\rho = \max(6\sigma, 2k)$
Length of sk	$2(\sigma + \rho + k)$	$6\rho = \max(6\sigma, 2k)$
Sign (# Mod. Mult.)	ρ	ρ
Test (# Mod. Mult.)	$< 2\rho + \sigma$	$< 4\rho = \max(4\sigma, 4k/3)$
Underlying problem	Factorization of $n = pq$ $p = q = 3 \bmod 4, p \neq q \bmod 8$	Factorization of $n = p^2q$

Table 1. Comparison of Several Parameters

6 Conclusion and Further Work

In this paper, we revisited some FSS schemes based on factorization related assumptions. First we cryptanalyzed a scheme based on a rather strong assumption and showed how to repair it. Then we introduced a new FSS scheme based on a well established factorization assumption (i.e. the hardness of factoring p^2q type integers) and provided a complete security proof for it. The new bundling homomorphism construction is more elemental and artless than the previous factoring bundling homomorphism from the quadratic residue scheme [PP97] and it promises to be of interest on its own. The new scheme's efficiency compares to the quadratic residue scheme, that is based on the hardness of factoring a special kind of Blum integers. Unfortunately, the efficiency of both schemes is not optimal (i.e. compared to discrete logarithm based schemes), although they are practical. Therefore important further work in this field is the development of a FSS scheme that is either: based on a fairly weak factorization assumption and as efficient as the discrete logarithm scheme.

References

- [AM94] Leonard Adleman and Kevin S. McCurley. Open problems in number-theoretic complexity ii. In *Algorithmic Number Theory - ANTS 94*, number 877 in *Lecture Notes in Computer Science*, pages 291–322, 1994.
- [BDHG99] Dan Boneh, Glenn Durfee, and Nick Howgrave-Graham. Factoring $N = p^r q$ for large r . In *Advances in Cryptology - CRYPTO 99*, volume 1666 of *Lecture Notes in Computer Science*, pages 326–337, Berlin, 1999. Springer-Verlag.
- [BP97] Niko Baric and Birgit Pfizmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology - EUROCRYPT 97*, volume 1233 of *Lecture Notes in Computer Science*, pages 366 – 377, Berlin, 1997. Springer-Verlag.
- [BPW91] Gerrit Bleumer, Birgit Pfizmann, and Michael Waidner. A remark on signature scheme where forgery can be proved. In *Advances in Cryptology - EUROCRYPT 90*, volume 473 of *Lecture Notes in Computer Science*, pages 441 – 445, Berlin, 1991. Springer-Verlag.
- [FKM⁺] Eiichiro Fujisaki, Tetsutaro Kobayashi, Hikaru Morita, Hiroaki Oguro, Tatsuaki Okamoto, Satomi Okazaki, David Pointcheval, and Shigenori Uchiyama. EPOC: Efficient probabilistic public-key encryption.

- [FOM91] Atsushi Fujioka, Tatsuaki Okamoto, and Shoji Miyaguchi. ESIGN: An efficient digital signature implementation for smart cards. In *Advances in Cryptology - EUROCRYPT 91*, volume 547 of *Lecture Notes in Computer Science*, pages 446–457, Berlin, 1991. Springer-Verlag.
- [GMR88] Shafi Goldwasser, Silvio Micali, and Ron L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.
- [Len87] H.W. Lenstra, Jr.. Factoring integers with elliptic curves. *Ann. of Math.* **126**, pages 649–673, 1987.
- [LL93] A.K. Lenstra and H.W. Lenstra, Jr., editors. *The Development of the Number Field Sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, 1993.
- [OU98] Tatsuaki Okamoto and Shigenori Uchiyama. A new public-key cryptosystem as secure as factoring. In *Advances in Cryptology - EUROCRYPT 98*, volume 1403 of *Lecture Notes in Computer Science*, pages 308–317, Berlin, 1998. Springer-Verlag.
- [PO96] René Peralta and Eiji Okamoto. Faster factoring of integers of a special form. *TIEICE: IEICE Transactions on Communications/Electronics/Information and Systems*, 1996.
- [PP97] Torben Pryds Pedersen and Birgit Pfitzmann. Fail-stop signatures. *SIAM Journal on Computing*, 26(2):291–330, 1997.
- [PW90] Birgit Pfitzmann and Michael Waidner. Formal aspects of fail-stop signatures. Technical report, Universität Karlsruhe, 1990.
- [Sho99] Victor Shoup. On the security of a practical identification scheme. *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, 12(4):247–260, 1999.
- [SSN03] Willy Susilo and Rei Safavi-Naini. An efficient fail-stop signature scheme based on factorization. In *Information Security and Cryptology ICISC 2002*, volume 2587 of *Lecture Notes in Computer Science*, pages 62–74, Berlin, 2003. Springer-Verlag.
- [SSNGS00] Willy Susilo, Rei Safavi-Naini, Marc Gysin, and Jennifer Seberry. A new and efficient fail-stop signature scheme. *The Computer Journal*, 43(5):430–437, 2000.
- [SSNP99] Willy Susilo, Rei Safavi-Naini, and Josef Pieprzyk. RSA-based fail-stop signature schemes. In *ICPP Workshop*, pages 161–166, 1999.
- [Tak98] Tsuyoshi Takagi. Fast RSA-type cryptosystem modulo p^kq . In *Advances in Cryptology - CRYPTO 98*, volume 1462 of *Lecture Notes in Computer Science*, pages 318–326, Berlin, 1998. Springer-Verlag.
- [Tak04] Tsuyoshi Takagi. A fast RSA-type public-key primitive modulo p^kq using hensel lifting. *IEICE Transactions*, Vol.E87-A(1):94–101, 2004.
- [vHP93] Eugène van Heyst and Torben Pryds Pedersen. How to make efficient fail-stop signatures. In *Advances in Cryptology - EUROCRYPT 92*, volume 1070 of *Lecture Notes in Computer Science*, pages 366 – 377, Berlin, 1993. Springer-Verlag.

A Review of the Quadratic Residue Scheme

In this section, we give a short account of the previous factorization based FSS scheme from [PP97]. The foundation of this scheme is the concept of claw-free permutations, that was introduced by Goldwasser, Micali and Rivest in 1988 [GMR88]. As the quadratic residue scheme is an instance of the general construction from Section 3, we only give the description of the family of bundling homomorphisms used.

Let σ, k be the security parameters related to the signer’s and the recipient’s security, respectively. Define the bundling degree $\tau := \sigma + \rho$, where $\mathcal{M} := \{0, 1, \dots, 2^\rho - 1\}$ is the message space. On

the input σ, τ , the key generating function g chooses two primes p, q with $p = q = 3 \pmod 4$ and $p \not\equiv q \pmod 8$, such that $n := pq$ has bit-length k . Let $QR(n)$ be the group of quadratic residues modulo n , i.e. $QR(n) := \{x \in (\mathbb{Z}/n\mathbb{Z})^\times \mid \exists y : y^2 = x \pmod n\}$. Then the Abelian groups G and H are defined as follows:

$$G := (\mathbb{Z}/2^p\mathbb{Z} \times (\pm QR(n))/\{1, -1\}, \circ, (0, 1)), \quad H := ((\pm QR(n))/\{1, -1\}, *, 1),$$

where the group operation \circ on G is given as

$$(a, x) \circ (b, y) := ((a + b \pmod{2^\tau}, xy4^{(a+b) \div 2^\tau}).$$

Each element of H is a coset $\{x, -x\}$, which is identified with its smaller member (i.e. with x , if $0 \leq x \leq (n-1)/2$, and with $-x$, otherwise).

Finally, the bundling homomorphism h is defined by:

$$h : \mathbb{Z}/2^p\mathbb{Z} \times (\pm QR(n))/\{1, -1\} \longrightarrow (\pm QR(n))/\{1, -1\} \\ (a, x) \mapsto \pm(4^a x^{2^\tau}) \pmod n,$$

where again the notation $\pm x$ in the image indicates that the coset $\{x, -x\}$ is identified with its smaller member.

It can be shown that the above construction is a family of bundling homomorphisms under the assumption that factoring Blum integers $n = pq$ with $p \not\equiv q \pmod 8$ is infeasible [PP97, BPW91]. Note that in contrast to our proposed scheme, the above construction is quite artificial, namely the cumbersome group operation \circ in G is only chosen in order to provide h with homomorphic properties. Concerning the groups G and H , there are two reasons for considering the factor group modulo $\{1, -1\}$ instead of $QR(n)$. On one hand, this choice anticipates the trivial collisions $x^2 = (-x)^2 \pmod n$, and on the other hand, it makes testing membership in H (and hence in G) efficient³.

B The Hardness of the p^2q Factoring Problem

Recently, the use p^2q type moduli (or more general p^kq) attracted much attention in cryptography. For example, the modulus p^2q is used in the famous EPOC cryptosystem [FKM⁺, OU98] and in the signature scheme ESIGN [FOM91], whereas moduli p^kq can be utilized to enhance the decryption speed in RSA-type encryption schemes [Tak98, Tak04]. Numerous researchers tried to exploit the special form of those integers to find faster factorization methods [AM94, PO96, BDHG99]. But unless the exponent k in p^kq is not too large, the most efficient methods for factoring $n = p^kq$ are still Lenstra's elliptic curve method (ECM) [Len87], its improvements [PO96], and the number field sieve (NFS) [LL93]. More precisely, if the size of the smallest prime factor of n exceeds some bound (about 200 bits), the NFS is the method of choice. Consequently, if n is sufficiently large (i.e. 1024 bits), the special form $n = p^2q$ causes no problem, because in contrast to ECM the runtime of the NFS depends only on the size of n , not on the size of its smallest prime factor. Concluding, although it is not known if factoring $n = p^2q$ is more tractable than factoring $n = pq$ or not, the p^2q Factorization Assumption is well-investigated and therefore can be regarded as fairly weak.

³ A number $0 \leq x \leq (n-1)/2$ belongs to H iff the Jacobi symbol $(\frac{x}{n})$ equals 1.