Short One-Time Signatures

Gregory M. Zaverucha* and Douglas R. Stinson†
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo ON, N2L 3G1, Canada
{gzaveruc, dstinson}@uwaterloo.ca

July 26, 2010

Abstract

We present a new one-time signature scheme having short signatures. Our new scheme is also the first one-time signature scheme that supports aggregation, batch verification, and which admits efficient proofs of knowledge. It has a fast signing algorithm, requiring only modular additions, and its verification cost is comparable to ECDSA verification. These properties make our scheme suitable for applications on resource-constrained devices such as smart cards and sensor nodes.

Keywords: one-time signatures, short signatures, cover-free families
AMS Classifications: 94A60, 05B40

1 Introduction

A one-time signature (OTS) scheme is a digital signature scheme that can be used to sign one message per key pair. More generally, we consider w-time signatures, which allow w signatures to be signed securely with each key pair (signing more than w messages breaks the security of the scheme). One-time signatures are an old idea: the first digital signature scheme invented was an OTS (Rabin/Lamport [43, 29]). The two main advantages of OTS is that they may be constructed from *any* one-way function, and the signing and verification algorithms are very fast and cheap to compute (when compared to regular public-key signatures). Common drawbacks, aside from the signature limit, are the signature length and the size of the public and private keys.

Despite the limitations of OTS, they have found many applications. On the more practical side, OTS can be used to authenticate messages in sensor networks [18] and to provide source authentication for multicast (also called broadcast) authentication [39]. One-time signatures are also used in the construction of other primitives, such as online/offline signatures [22] and CCA-secure public-key encryption [14].

With respect to signature length, designing conventional signature schemes with short signatures is not a new problem, and is motivated by applications with strong bandwidth constraints. For

*Supported by an NSERC post-graduate scholarship
†Research supported by NSERC discovery grant 203114-06

example, signatures which are bar-coded for postage stamps, or which must be entered manually by users as a part of a product registration system, must be as short as possible while maintaining security.

There is a significant gap in signature length between regular public-key signatures and OTS. While signatures in conventional schemes can be very short, e.g., as small as 160 bits for 80-bit security (in the BLS scheme [9]), one-time signatures are usually many times longer (for typical examples, see [39, 44], where signature lengths are over one thousand bits). This motivates the following questions: Are short OTS possible? Can we retain the advantages of OTS but reduce the signature size?

We give a positive answer to these questions. In particular, we make the following contributions:

- We give the first one-time signature scheme with short signatures (constant size, about 180 bits long for 80-bit security) and a tight security reduction based on the difficulty of the discrete logarithm problem.
- We give a unified description of five previous schemes and improve parameter selection for these schemes.
- Our new scheme supports aggregation and batch verification, admits efficient proofs of knowledge, and is fail-stop. Ours is the first OTS to have the first two of these properties.
- As a corollary, we give a fail-stop signature scheme with the shortest signatures to date.
- Our new OTS retains fast signing, but has slower verification than most OTS. Nonetheless, verification in our new scheme is only about as expensive as verification of an ECDSA signature.

Informal description of our solution. We consider a general class of OTS schemes based on cover-free families, and make the observation that the one-way function in an OTS scheme is being used as a commitment function. The signer creates commitments during key generation that form the public key, and the openings of these commitments make up the signature. By replacing the one-way function with Pedersen commitments (of the form $g^s h^r$) [38], we can use the algebraic properties of this commitment scheme to compress a number of openings into a single opening. We also show that it is sufficient for security to have the value r in the commitment be very small, leading to short signatures. We make further use of the algebraic structure to prove security, and provide additional features: batch verification, aggregation, and proofs of knowledge.

Paper organization. First we describe a general construction of OTS based on cover-free families (§2), then we review relevant related work on one-time signatures and their applications (§3). In §4 we present our new scheme and discuss parameter selection, in §5 we describe additional features of our scheme, and then we conclude with a discussion of its applications (§6).

2 General Construction of OTS from Cover-Free Families

A number of existing OTS schemes may be described as special cases of a (unified) general construction based on cover-free families. Our new scheme will also be a variant of this general construction. We start with the definition of a cover-free family (which is somewhat specialized for this paper).

Definition 2.1. *A* w-cover-free family $(X, \mathcal{B})$ *is a set* X *of* m *elements, and a set* $\mathcal{B}$ *of* 2^n *subsets of* X *called blocks, with the following property. For any* w *blocks* $B_{i_1}, \ldots, B_{i_w} \in \mathcal{B}$*, and all other blocks* $B \in \mathcal{B}$*, it holds that*

$$B \not\subseteq \bigcup_{j=1}^{w} B_{i_j}$$

We say that $B_{i_1}, \ldots, B_{i_w}$ *does not* cover *any other* $B \in \mathcal{B}$*. We will use the notation* w-CFF$(m, 2^n)$ *for cover-free families.*

We now describe the general construction of a w-time signature scheme based on a cover-free family. Throughout this paper, the message space is $\{0,1\}^n$.

Setup(n)**:** Let $(X, \mathcal{B})$ be a w-CFF with $|X| = m$ elements and $|\mathcal{B}| = 2^n$ sets. Let $f : \{0,1\}^\ell \to \{0,1\}^\ell$ be any one-way function (where ℓ is a security parameter). Also, to each message $M \in \{0,1\}^n$ we associate a unique $B_M \in \mathcal{B}$ (this correspondence is public).

Key Generation: Choose m random values $s_1, \ldots, s_m \in \{0,1\}^\ell$ and compute $v_i = f(s_i)$ for $i = 1, \ldots, m$. Output the public key $PK := (v_1, \ldots, v_m)$ and the secret key $SK := (s_1, \ldots, s_m)$.

Sign: To sign $M \in \{0,1\}^n$, compute $B_M \in \mathcal{B}$, the subset corresponding to M. Then output the signature $\sigma = \{(s_i, i) : i \in B_M\}$.

Verify: To verify (σ, M) using PK, compute B_M, then check that $f(s_i) = v_i$ for all $i \in B_M$. If so, output 1; otherwise, output 0.

It is easy to see that security is provided by the one-wayness of f and the cover-free property: given w signatures, all other messages require knowledge of at least one s_i value that has not been revealed (a more detailed analysis appears in [40]). The correspondence between M and B_M depends on the CFF; we will describe an efficient algorithm for our scheme in §4.2.

3 Related work

We will first describe five existing schemes that are special cases of the general CFF scheme described above. Then we discuss other work related to OTS and fail-stop signatures. The OTS literature is vast, and a complete survey is impractical, given space constraints. For a more comprehensive survey, see Menezes et al. [31, Ch. 11.6] and Dods et al. [21]. All of the schemes considered in [21, 31] have signatures that are longer than the new scheme we present in §4. For details of the signature length in conventional signature schemes, see [9].

3.1 Schemes Based on the CFF Model

The descriptions assume we want to sign n-bit messages. We write M_i for the i-th bit of message M.

In the Lamport scheme [29], the public key consists of $v_{i,b} = f(s_{i,b})$ for $i = 1, \ldots, n$, and for $b = 0, 1$. To sign M, reveal $s_{1,M_1}, \ldots, s_{n,M_n}$. The verifier checks that $v_{i,M_i} = f(s_{i,M_i})$, for $i = 1, \ldots, n$. This can be interpreted as a 1-CFF with $2n$ points, where $X = \{(i, b) : i = 1, \ldots, n \text{ and } b = 0, 1\}$ and $B_M = \{(i, b) : M_i = b\}$ for all $M \in \{0,1\}^n$.

The Bos and Chaum [10] scheme has m secrets and can be used to sign all weight $\lfloor m/2 \rfloor$ binary vectors. Therefore, we require $\binom{m}{\lfloor m/2 \rfloor} > 2^n$ in order to sign n-bit messages. When these vectors are viewed as the incidence matrix of a CFF, this forms an *optimal* 1-CFF with m points (optimality is proven by Sperner [41]).

The Reyzin-Reyzin schemes [44] use random structures instead of explicitly constructed CFFs (under the name "subset-resilient functions"). Their security analysis considers two probabilities. First, the probability that a random matrix is a w-CFF is used in the security analysis for chosen-message attacks. The adversary has the description of the CFF, and finding w blocks that cover another block allows the adversary to sign the covered message after observing w signatures. Second, the probability that a randomly chosen set of messages covers another message is used to analyze security when the adversary is passive and observes w signatures on random messages. Some example parameters are given in [44]. To sign two messages with 80-bit security and a forgery probability of 2^{-53}, the public key size is ≈ 82 Kb and the signatures are 1600 bits in length.

The scheme of Pieprzyk, Wang and Xing [40] (the PWX scheme) is the first to explicitly use CFFs, and it uses some constructions based on polynomials and error-correcting codes. All of the constructions in [40] are constructions for w-CFF for general w. The special case $w = 1$ is not singled out (having already been considered in [10]).

Katz [27] defines the same scheme as the PWX scheme, but uses a CFF with a stronger property, namely, that the union of w blocks misses λ (or more) points of any other block. The stronger property is required to provide leakage resilience (a property that guarantees security even if a bounded amount of the signer's secret information is leaked). As in [40], going from signing one message to signing multiple messages is done via a w-CFF.

Parameter Selection In Section 4.3 we show that better parameters (shorter signatures and smaller keys) are obtained by using w 1-CFFs rather than a single w-CFF. This improves the schemes above that use w-CFF.

3.2 Other Work Related to One-Time Signatures

Fail-Stop Signatures. In a fail-stop signature scheme, the signer may efficiently prove that he did not create a given, valid signature (when the signature is a legitimate forgery). The scheme of van Heyst and Pedersen [49] uses Pedersen commitments in the public key to create a fail-stop signature scheme. For this reason, our schemes have some similarities: both schemes can sign w messages, the public key is a list of commitments, and the secret key is their openings. The signature and verification algorithms differ, however: in [49], the messages are used in the signature directly (rather than being encoded with a CFF), verification is slower (at least twice as slow) and signatures are about twice as long.

Other schemes. Goldwasser, Micali and Rivest [24] present a one-time signature scheme based on trapdoor claw-free permutations. The scheme requires n evaluations of the permutation to sign n-bit messages and the signature is the size of the output. For all known trapdoor claw-free permutations, this means that the signature is at least as long as an RSA modulus (i.e., 1024 bits at the 80-bit security level).

Groth [25] gives an alternate construction of an OTS scheme with the same properties as the van Heyst and Pedersen scheme (though it is not fail-stop).

Bellare and Shoup [4] present a general construction of OTS from three-move identification protocols. Specific instantiations of their construction yield an OTS scheme with short signatures (as short as ours) based on the *one more discrete log* problem [36], while another gives a scheme with signatures about twice as large as ours having discrete log security. Both schemes presented in [4] also require a collision resistant hash function (CRHF), even when signing short messages, i.e., the hash function is part of the signing algorithm, and not just applied to the input message in the usual way.

A popular approach to convert an OTS scheme to a w-time signature scheme is to use a Merkle tree to authenticate w one-time public keys [21, 31, 32]. Since the signature must include a path through the authentication tree, the signature length is typically thousands of bits. Some examples of multiple-time signatures using Merkle trees are given in [6, 11, 34]. To our knowledge, the Merkle-like scheme with the shortest signatures is due to Dahmen and Krauß [18]. At the 80-bit security level, the signatures produced by their scheme are 330 bits long (in general the signature length is about 4κ to achieve κ-bit security).

A recent paper by Mohassel gives a black-box construction of OTS schemes from chameleon hash functions [33]. This result leads to new OTS schemes based on the hardness of factoring, the discrete logarithm problem, and the short integer solution problem on lattices. The instantiation based on the discrete log assumption has the shortest signatures, which are the same length as the van Heyst and Pedersen OTS, about twice as long as in the new scheme we present.

3.3 Applications of One-Time Signatures

Here we review a few example applications of one-time signatures. In §6 we discuss using our new OTS for these applications.

Smart cards and sensor networks. Resource-constrained devices that are not capable of using public-key cryptography (RSA, for example) often have sufficient resources for symmetric-key operations and w-time signatures. Rohde et al. [42] show that the Merkle signature scheme is practical on smart cards without a cryptographic coprocessor. Nodes in sensor networks have similar limitations, and one-time signatures were applied in this case by Dahmen and Krauß [18]. Their scheme uses the fact that most messages in sensor networks are short (8–24 bits), for example, basic commands or simple measurements (such as temperature).

Bicakci et al. [7] introduce the concept of *one-time sensors*. In this application, nodes in a wireless sensor network are given enough cryptographic material to produce only one (or a few) authentic messages. This is motivated by nodes with a short lifespan, for instance, due to limited battery life. Also, the nodes might not be strictly disposable, e.g., they must return to the central authority periodically to obtain new cryptographic keys. Signing must be fast on a sensor node, while verification is done at a central repository by a more powerful computer.

Broadcast authentication. Using OTS to authenticate a stream of broadcasted data was initiated by Gennaro and Rohatgi [23, 45]. For streams of data that are unknown in advance (e.g., live broadcasts) their solution uses an OTS to authenticate each block of the stream against the public key transmitted in the previous block.

The BiBa OTS scheme was designed by Perrig [39] as the main component of the BiBa broadcast authentication protocol. In broadcast authentication, a sender wishes to send authenticated

packets to multiple receivers that may have limited resources. Fast signing and verification are important in this application to allow high-throughput, low-latency communication. Substituting the Reyzin-Reyzin scheme (see §3.1) for the BiBa OTS improves the efficiency of the BiBa broadcast authentication protocol (see [44]).

4 A New OTS Scheme with Short Signatures

In this section we describe our new scheme and discuss parameter selection. We also give a set of concrete parameters for 80-bit security. The scheme, given in Figure 1, signs only one message, since in §4.3 we show that w public keys that each sign one message will be smaller than one public key that signs w messages (for CFF-based OTS).

4.1 Scheme Description

We first briefly review Pedersen's commitment scheme [38]. Let G be a group of order q, where q is prime. Let $g, h \in G$ be system parameters. To commit to a message $m \in \mathbb{Z}_q^*$, choose $r \in \mathbb{Z}_q$ at random, and output $C = g^m h^r$ as the commitment. To open C, reveal (m, r). Also note that, given two distinct openings to a Pedersen commitment using distinct bases g and h, it is possible to recover $\log_g h$.

The complete scheme is presented in Figure 1.

Remark 4.1. *For the scheme in Figure 1 to be fail-stop,* $\log_g h$ *must be unknown to the signer. In practice, g and h may be chosen by a trusted authority, or verifiably at random. A forgery will allow the signer to recover* $\log_g h$ *(with probability at least* $1 - 2^{-\ell_r}$*), and use it as a proof of forgery (under the assumption that the signer cannot compute* $\log_g h$*). See the proof of Theorem 4.4 for details. If the fail-stop property is not desired, the signer may use any distinct g and h, provided* $\log_g h$ *is not publicly known.*

Our scheme is secure if the discrete log problem is hard in G.

Definition 4.2. *Let G be a cyclic group of prime order q, and let g be a generator of G. The* discrete log problem (DLP) *is, when given $g, h \in G$, to compute $x = \log_g h$, i.e., $h = g^x$.*

We say that an adversary A (t, ϵ)-solves the DLP in G if after time t, A outputs a correct solution to a DLP instance with probability ϵ.

The definition of security we use is *strong unforgeability under chosen message attacks*. Strong unforgeability against an adaptive adversary is modelled by the following game between a challenger C and an adversary A.

1. C publishes $Params := \mathsf{Setup}(n)$.
2. C runs the key generation algorithm with $Params$ w times, and publishes $PK_1, \ldots, PK_w$.
3. A adaptively requests up to w signatures, at most one per public key, which C provides. Let Q be the set of (message, signature, public key index) triples queried by A.
4. A outputs (M, σ, i).

Setup(n)**:** Choose a group G of order q, where q is an ℓ_q-bit prime (ℓ_q is a security parameter). Let $(X, \mathcal{B})$ be an optimal 1-CFF$(m, 2^n)$, and write B_M for the block corresponding to the message $M \in \{0,1\}^n$. Let g and h be generators of G (see Remark 4.1 below on choosing g and h).

Key Generation: For $i = 1, \ldots, m$, generate random values $s_i \in_R \mathbb{Z}_q$ and $r_i \in_R \{0,1\}^{\ell_r}$, where ℓ_r is a parameter. The secret key is $SK := (s_i, r_i)_{i=1}^m$. For $i = 1, \ldots, m$, compute $v_i := g^{s_i} h^{r_i}$. The public key is $PK := (v_1, \ldots, v_m)$.

Sign: To sign M, compute and output

$$(\sigma, \rho) := \left(\sum_{i \in B_M} s_i \pmod q, \sum_{i \in B_M} r_i \right) \in \mathbb{Z}_q \times \mathbb{Z}.$$

More precisely, ρ is an integer in $[0, m(2^{\ell_r} - 1)/2]$, since $|B_M| = \lfloor m/2 \rfloor$ in the optimal 1-CFF.

Verify: To verify the signature (σ, ρ), check that $0 \leq \rho \leq m(2^{\ell_r} - 1)/2$ and

$$g^\sigma h^\rho \stackrel{?}{=} \prod_{i \in B_M} v_i.$$

Output 1 if both conditions hold, and output 0 otherwise.

To sign w messages, simply create w public keys as above, and include a counter with each signature to indicate which of the w public keys is being used.

Figure 1: Our new one-time signature scheme.

We say that A (t, ϵ)-*wins* the game if

$$\Pr[\mathsf{Verify}(PK_i, M, \sigma) = 1 \wedge (M, \sigma, i) \notin Q] = \epsilon \,,$$

and Step 4 takes time t. Note that A can win by outputting a triple (M, σ, i) where M appears in Q, but with a different σ. This is the *strong* unforgeability property: a new signature on a previously signed message is considered a forgery.

For our proof of security, we will require the following technical lemma. In what follows, for an integer x we write $[x]$ to denote $\{0, \ldots, x-1\}$.

Lemma 4.3. *Let $\mathcal{X}_n$ be the probability distribution on $[n2^\ell]$ defined as $\mathcal{X}_n = X_1 + \ldots + X_n$, where X_i is the uniform distribution on $[2^\ell]$. Then the min-entropy of $\mathcal{X}_n$, $H_\infty(\mathcal{X}_n)$, is at least ℓ bits.*

Proof. Let $P_n(k) = \Pr[\mathcal{X}_n = k]$ for $k \in [n2^\ell]$. Clearly, $P_1(k) = 2^{-\ell}$ for all $k \in [2^\ell]$ and zero otherwise. By applying repeated convolution to P_1,

$$\begin{aligned} P_n(k) &= \sum_{i \in \mathbb{Z}} P_1(i) P_{n-1}(k-i) \\ &= 2^{-\ell}(P_{n-1}(k-1) + \ldots + P_{n-1}(k-2^\ell)) \,. \end{aligned}$$

because $P_1(i) = 0$ for values of $i \notin [2^\ell]$. Since the sum of the terms $P_{n-1}(k-i)$ are at most one (P_{n-1} is a probability distribution), it follows that $P_n(k) \leq 2^{-\ell}$ for all n, k. Now we compute the min-entropy of $\mathcal{X}_n$:

$$H_\infty(\mathcal{X}_n) = -\log\left(\max\left\{\Pr[\mathcal{X}_n = x]\right\}_{x\in[n2^\ell]}\right) \geq -\log 2^{-\ell} = \ell,$$

which proves the lemma. □

We now prove the security of our scheme.

Theorem 4.4. *Let A be an adversary who (t,ϵ)-wins the strong unforgeability security game above for the w-time signature scheme in Figure 1. Then A can be used to $(t+c, \epsilon(1-2^{-\ell_r}))$-solve the DLP in G, where ℓ_r is a parameter of the signature scheme, and c is a small constant.*

Proof. B will be a new algorithm that uses A to solve an instance of the DLP in G. Let the DLP instance given as input to B be (g,h). B creates $PK_1,\ldots,PK_w$ and $SK_1,\ldots,SK_w$ as above, using g and h, and gives $PK_1,\ldots PK_w$ and the system parameters to A. Since B knows $SK_1,\ldots,SK_i$, the w adaptive queries A makes may all be answered correctly. (Recall that each message is signed with one of the w keys, as the w-time key pair consists of w one-time key pairs.) After seeing signatures on a set of Q messages, A outputs a signature (σ,ρ,i) on a message M, which verifies using PK_i. Let $(\overline{\sigma},\overline{\rho})$ be the signature on M created by B using SK_i and the signing algorithm above. Now define the Pedersen commitment

$$C := \prod_{j\in B_M} v_j = g^{\overline{\sigma}} h^{\overline{\rho}} = g^\sigma h^\rho \ .$$

There are two cases to consider: (i) $M \notin Q$ or, (ii) $M \in Q$, but $(\sigma,\rho) \neq (\overline{\sigma},\overline{\rho})$.

Since the ρ-value of a signature is in $[0, m(2^{\ell_r}-1)/2]$, there are $m(2^{\ell_r}-1)/2$ valid openings of C. We must analyze the probability that A outputs the same opening as $(\overline{\sigma},\overline{\rho})$. A does have some information about $\overline{\rho}$; he knows that $\overline{\rho}$ is the sum of uniformly random values from $\{0,1\}^{\ell_r}$. By Lemma 4.3, the min-entropy of $\overline{\rho}$ as a random variable defined on $\left\{0,\ldots,m(2^{\ell_r}-1)/2\right\}$ is at least ℓ_r bits. Therefore, in case (i) the forgery equals $(\overline{\sigma},\overline{\rho})$ with probability not more than $2^{-\ell_r}$, and differs (giving distinct openings of C) with probability at least $1-2^{-\ell_r}$. In case (ii), $(\sigma,\rho) \neq (\overline{\sigma},\overline{\rho})$ by definition, so we always have distinct openings of C.

Thus, with probability at least $\epsilon(1-2^{-\ell_r})$ A succeeds *and* B has two distinct openings of C, which allows B to recover $\log_g h$, solving the DLP instance. The time required by B is t (the time required by A) plus the time required to: generate w key pairs, sign w messages, and solve for $\log_g h$, i.e, compute $\log_g h = (\overline{\rho}-\rho)/(\sigma-\overline{\sigma})$. □

Remark 4.5. *Note that non-repudiation is provided, despite the signer's ability to choose SK such that two messages have the same signature (the signer can do this by ensuring that the sum of two openings is the same). Suppose Bob has a signature (σ,ρ) from Alice on the message M, and Alice later claims that (σ,ρ) is a signature on M' (and the verification equation holds). If Alice is telling the truth, and she did not give Bob a signature on M, then Bob has produced a forgery on M, which is not possible by Theorem 4.4. Therefore, Alice must have signed M.*

A natural question to ask is whether we can change the commitments used in the public key to simple discrete log commitments, i.e., a commitment to m is computed as g^m. This simplifies the scheme and reduces signature size (by 10 bits using our parameters from §4.3). Unfortunately, we were not able to find a security proof for this modified scheme with a tight security reduction, and it remains an open question.

4.2 Encoding a message M as B_M

Encoding messages is considered in detail by Bos and Chaum [10] as well as Reyzin and Reyzin [44]. Pieprzyk et al. [40] discuss coding-theoretic approaches to the problem (encoding using generator matrices). In both [10] and [44], algorithms to encode a message as a weight $\lfloor m/2 \rfloor$ vector are given. Both require a nontrivial amount of computation (at least $m^2 n$ multiplications). Here we mention a simpler approach.

A bijective function $S : \left\{1, \ldots, \binom{a}{b}\right\} \leftrightarrow \{b\text{-subsets of } [a]\}$ is called a *ranking algorithm* from integers to subsets and an *unranking algorithm* from subsets to integers. The well-known ranking algorithm described in [15, 16] requires $\lfloor m/2 \rfloor$ subtractions and $\log d$ comparisons where $d = \binom{n}{\lfloor m/2 \rfloor}$, using precomputed values. The algorithm is as follows:

Input M: an n-bit integer, k: weight of the output
Output y: length d binary vector of weight k

for $i = 1, \ldots, d$

  if $M > \binom{n-i}{k}$
    $y_i = 1$
    $M = M - \binom{n-i}{k}$
    $k = k - 1$
  else $y_i = 0$

return y

This is called the *co-lex ranking* [28]. Bicakci et al. [6] use the *lex ranking* in their implementation of the Bos-Chaum scheme.

4.3 Parameter Selection

We first show that signing w messages using w instances of a 1-CFF will require less storage than using a single w-CFF. This applies to all of the CFF schemes described in Section 3.1.

Suppose $w = \ell t$ for integers ℓ and t. In any w-CFF$(m, 2^n)$, we have $m = \Omega(\frac{w^2}{\log w} n)$ (see Stinson et al. [46]). Combining ℓ public keys allowing t signatures each to create a public key permitting $w = \ell t$ signatures therefore has $m = \Omega(\ell \frac{t^2}{\log t} n)$ (where m in this case is the *total* number of secrets needed by the signer). It is easy to see that this bound on m is smallest when $\ell = w$ and $t = 1$, i.e., when we use w public keys each allowing one signature. It is also easy to see that an intermediate choice (e.g., using $\sqrt{w}$ $\sqrt{w}$-CFFs) is not an improvement. Thus, by using w 1-CFF$(m, 2^n)$ instead of one w-CFF$(m, 2^n)$, we can potentially reduce storage by a factor of $w/\log w$. A minor drawback of this approach is that the signature must include a counter, to tell the verifier which part of the public key to use.

Constructing an optimal 1-CFF$(m, 2^n)$ is simple: choose all length m binary vectors with weight $\lfloor m/2 \rfloor$ (there are $\binom{m}{\lfloor m/2 \rfloor}$ such vectors). This is a 1-CFF$(m, 2^n)$ provided that

$$\binom{m}{\lfloor m/2 \rfloor} > 2^n, \tag{1}$$

in fact, it is the same CFF used in the Bos-Chaum scheme [10].[1]

Next we consider ℓ_q. We choose G to be a group of points on an elliptic curve, since its elements have a compact representation. Therefore, ℓ_q must be large enough so that the DLP is hard in G. We would probably use the standard NIST curves [35] for performance reasons.

In most applications, $\ell_r = 10$ will be sufficient. This means that the security reduction succeeds with probability $\epsilon - \epsilon/1024$, or put another way, an instance of the DLP may be solved on average, given a forgery, 1023 out of 1024 times. A probability of proving a forgery of 1023/1024 should also be sufficient for most applications requiring fail-stop signatures.

Parameter sizes for 80-bit security and arbitrary-length messages. We assume that messages will be hashed with a collision-resistant hash function which produces a 160-bit output, and so $n = 160$. We need $\ell_q = 160$ for security, therefore group elements may be represented using 160 bits (recall our choice of G above). In this case the public key size is $160 \cdot 165w$ bits, which is about 26Kb per signature (we take $m \geq 165$ in order to satisfy (1)). When $w = 10$, 100, and 1000, the public key is 264Kb, 2.64Mb and 26.4Mb, respectively. Since $\ell_r = 10$ and $\log_2 \rho \leq \log_2(\lfloor m/2 \rfloor 2^{\ell_r})$ the signature length is not more than $160 + 17 + \log_2 w = 177 + \log_2 w$ bits. (The $\log_2 w$ bits are required for the counter.) Next, the encoding algorithm requires $\log_2 \binom{165}{82} \approx 160$ comparisons, and we must store about 162 160-bit values, requiring about 26Kb space.

Computational costs. Recall that $m = 165$ for 80-bit security and G is a prime-order elliptic curve group. Key generation requires m multi-exponentiations in G. Verification requires a single multi-exponentiation (where one exponent is small) and 82 multiplications, which is comparable to ECDSA (which requires one multi-exponentiation with full-length exponents). Signing requires 82 additions mod q. Note that, when computing the multi-exponentiations during key generation and verification, fast exponentiation techniques are applicable [5].

The signing and verification times of our scheme were compared to ECDSA using version 5.6.0 of the Crypto++ library [17] on a Pentium D 3.0GHz processor. The parameter sizes and elliptic curve group implementation used in both schemes was the same. For signing, our OTS is 47.14 times faster than ECDSA, while verification was 1.13 times slower.

Parameter sizes for 80-bit security and short messages. We now consider the parameter sizes when our scheme is used to sign 16-bit messages. Recall (§3) that short messages are common in sensor networks [18]. To construct the optimal 1-CFF$(m, 2^{16})$ requires $m = 19$. Therefore the public key contains $3040w$ bits to sign w 16-bit messages. While the key sizes are larger than the Dahmen and Krauß scheme, the signature overhead is halved.

Some possible tradeoffs. First, to reduce the signature size even further, we can avoid including the counter at the cost of either (i) increased verification time, by simply omitting it, which may be acceptable when w is small, or (ii) larger public and private parameters, by using a w-CFF. Second, using randomized hashing, we may reduce the public and private storage requirement. The idea is to use a *target collision-resistant hash function*, which is a type of keyed hash function with a short digest (i.e., a κ-bit digest for κ-bit security, instead of a 2κ-bit digest). Since the digest

[1] The 1-CFF with the stronger property required to construct Katz' leakage-resilient scheme also has a simple (but not optimal) construction – a direct generalization of the Sperner construction (see Stinson et al. [47, Lemma 3.2]). The reduction in required storage is even more pronounced here, as m depends more strongly on w.

Scheme	Security	PK size	SK size	Sig. size	Sign	Verify
vHP [49]	DLP	2κ	4κ	2κ	2 mult.	3 exp.
Groth [25]	DLP	3κ	2κ	2κ	3 mult.	3 exp.
BS [4]	DLP + CRHF	κ	2κ	2κ	2 mult.	1 exp.
BS [4]	omDLP + CRHF	κ	3κ	κ	1 mult	1 exp.
Generic CFF [10, 29, 40, 44]	OWF	$O(\gamma n)$	$O(\gamma n)$	$O(\gamma n)$	Encode (§4.2)	$O(n)$ OWF computations
This paper	DLP	$O(\kappa n)$	$O(\kappa n)$	$\kappa + c$	Encode and $O(n)$ add.	1 exp. + $O(n)$ mult.

Table 1: Comparison of various OTS schemes. The DL security parameter is denoted by κ, the one-way function (OWF) security parameter is denoted by δ and n is the number of bits in the message to sign, and c is a small constant ($c = 10$ bits in the examples given above). The "Generic CFF" construction is given in §2.

is half the size, the public and private parameters are roughly halved. However, the signer must commit to w hash keys during key generation, and include the hash key with the signature. This idea is discussed in [34] and [45]. Another possible way to reduce signer storage is to store only a short seed and generate SK using a pseudorandom generator (PRG), which increases computation during signing since the PRG must be invoked $\lfloor m/2 \rfloor$ times. Finally, we may reduce computation by precomputing and storing values of h^r for all $r \in \{0,1\}^{\ell_r}$, since ℓ_r is small.

Comparison. In Table 1, we compare the size of the keys and signatures, and the computation required to create signatures and verify them, for several OTS. For schemes providing security under the DLP, our scheme has the shortest signatures. This is traded off against the large public key size. The scheme of Bellare and Shoup (BS) has the best all-around performance, but it requires stronger assumptions, namely, the one more DLP (omDLP) and the existence of a collision-resistant hash function (CRHF).

5 Additional Features of the OTS Scheme

In this section, we describe four additional features of our OTS scheme. Batch verification and aggregation are techniques for handling multiple signatures, in order to efficiently verify or compress many signatures. Proving knowledge of a signature and verifiably encrypting a signature are useful properties when using OTS to build more complex cryptographic protocols (e.g., certified email [2]).

5.1 Batch Verification

Our new OTS provides batch verification under the most general definition: verification of k signatures on k (possibly different) messages created under k (possibly different) public keys. Given a batch of signatures, $(\sigma_1, \rho_1, \ldots, \sigma_k, \rho_k, M_1, \ldots, M_k, PK_1, \ldots, PK_k)$, we can efficiently verify them

using the *small exponents tests* of Bellare, Garay and Rabin [3], as follows:

$$g^{\sigma_1 d_1 + \ldots + \sigma_k d_k} h^{\rho_1 d_1 + \ldots \rho_k d_k} \stackrel{?}{=} \left(\prod_{i \in B_{M_1}, v_i \in PK_1} v_i \right)^{d_1} \times \cdots \times \left(\prod_{i \in B_{M_k}, v_i \in PK_k} v_i \right)^{d_k}$$

where $d_1, \ldots, d_k$ are randomly chosen small exponents. Using ℓ-bit exponents, there is a $2^{-\ell}$ probability of error during verification.

5.2 Aggregation

In this section we prove that k signatures may be securely aggregated by simply computing their sum. We consider the most general definition of aggregation: signatures from k (possibly different) signers on k (possibly different) messages are combined to produce a single signature. Here are the aggregation functions.

$\mathsf{Aggregate}(\sigma_1, \rho_1, \ldots, \sigma_k, \rho_k)$: Output the aggregate signature

$$(\sigma, \rho) = \left(\sum_{i=1}^{k} \sigma_i \pmod q, \sum_{i=1}^{k} \rho_i \pmod q \right) \in \mathbb{Z}_q \times \mathbb{Z}_q \, .$$

Note that, when aggregating a large number of signatures, ρ may be reduced mod q.

$\mathsf{VerAgg}(PK_1, \ldots, PK_k, M_1, \ldots, M_k, \sigma, \rho)$: Output 1 if

$$g^{\sigma} h^{\rho} \stackrel{?}{=} \left(\prod_{i \in B_{M_1}, v_i \in PK_1} v_i \right) \times \cdots \times \left(\prod_{i \in B_{M_k}, v_i \in PK_k} v_i \right) \, ,$$

holds, and output 0 otherwise.

Here we present the security game for aggregate signatures from Boneh et al. [8]. Let C be the challenger and A be the adversary.

1. C runs $\mathsf{Setup}(n)$ to generate $params$, computes $PK_1 = \mathsf{Keygen}(params)$, and gives $params$ and PK_1 to A.
2. A adaptively queries a set of messages Q, and C provides signatures using SK_1.
3. A outputs $PK_2, \ldots, PK_k$, (σ, ρ) and $\mathcal{M} = \{M_1, \ldots, M_k\}$. A's success probability is the probability that $\mathsf{VerAgg}(PK_1, \ldots, PK_k, \mathcal{M}, \sigma, \rho) = 1$ and $M_1 \notin Q$.

In order for $\mathsf{Aggregate}$ to be secure, we must augment the public key PK_i with a zero-knowledge (ZK) proof of knowledge Z_i of the secret key (using standard techniques, e.g., [12, 19]). This proves that each public key is well formed. We naturally require that A must output valid public keys to win the above game.

Sketch of the security proof. We show that an adversary who produces a forgery in the above game can be used to solve the DLP in G. The challenger creates PK_1, SK_1, and Z_1 and sends PK_1 and Z_1 to A. A makes adaptive queries, then responds as in Step 3. C then extracts $SK_2, \ldots, SK_k$ from $Z_2, \ldots, Z_k$ (which is possible using the knowledge extractor since the proofs Z_i are valid, see [19]). Given (σ, ρ), and $SK_2, \ldots, SK_k$, the challenger computes $(\sigma', \rho') = \mathsf{Aggregate}(PK_2, \ldots, PK_k, M_2, \ldots, M_k)$. Now, $\sigma'' := \sigma - \sigma'$, $\rho'' := \rho - \rho'$ is a valid signature on M_1 under PK_1 (a forgery). We now proceed as in the proof of Theorem 4.4.

Remark 5.1. *No ZK proofs are necessary when all signatures are created by a single signer, or by a group of trusted signers. In this case, since the participants are honest (not under A's control), the above game should be modified to have the challenger choose $PK_2, \ldots, PK_k$, instead of the adversary.*

5.3 Proving Knowledge of a Signature on the Message M

The following ZK proof of knowledge allows a prover to convince a verifier that he possesses a signature on M under a known public key without revealing the signature.[2]

$$\mathrm{PK}\{(\sigma, \rho) : \prod_{i \in B_M} v_i = g^\sigma h^\rho\}$$

Note that prover only needs to know g, h, σ, ρ and can remain ignorant of the whole public key. This is easily generalized to proving knowledge of aggregate signatures. The computation of the prover is a single multi-exponentiation in both cases. The size of the proof is about $4\ell_q$ bits.

5.4 Verifiably Encrypting a Signature

Since proving knowledge of a signature amounts to proving knowledge of a discrete log, we can verifiably encrypt signatures by combining our OTS with the Camenisch-Shoup verifiable encryption scheme [13]. The ability to verifiably encrypt signatures allows them to be exchanged fairly using an optimistic fair exchange protocol [1]. If we encrypt using an additively homomorphic encryption scheme with efficient proofs of plaintext knowledge (such as the Paillier scheme [20]), encrypted signatures may also be aggregated.

6 Impact on Applications

In this section we discuss using our scheme in the applications mentioned in Section 3.3. For two of the applications mentioned in the introduction, bar-coded postage and product registration systems, our scheme may be used, but it does not offer any immediate advantages over conventional signatures. So we do not discuss these applications further.

Smart cards and sensor networks. It is important that authentication for smart cards and sensor networks have low overhead, due to the high cost of communication [30]. This is the biggest advantage of using our OTS: the number of signature bits per message bit is smaller than alternative schemes. A specific application where short signatures are especially important is when sensors are

[2]The "PK" (proof of knowledge) notation [12] is a short-hand for the various Schnorr-like proof of knowledge of a discrete logarithm protocols which exist for types of statements such as knowledge of, relations between, and the length of discrete logarithms.

used in vehicular networks. Since the vehicles pass roadside access points at high speed, only a limited amount of data can be transferred [37]. Key management can be handled offline, i.e., when the vehicle is parked.

The computation costs of our scheme favour applications where the card or nodes produce the signatures and verification is performed by a device with more resources. For example, in heterogeneous sensor networks, where signatures are created by low-resource nodes and verified by nodes with greater resources, or in sensor networks where the sensors return authenticated data to a central authority (e.g., weather sensors reporting to a meteorological agency). In the latter example, aggregation can reduce authentication overhead even further, and batch verification can reduce verification time. We stress that that resource-constrained devices are still able to efficiently verify signatures in our scheme. ECDSA has been shown practical for sensor nodes and smart cards [26, 48], and verification in our scheme requires equivalent computational resources.

Broadcast authentication. While our scheme could be used with BiBa, it will not be as efficient as the Reyzin-Reyzin scheme because of the size of the public parameters; we cannot use the hash chain technique to re-use a single public key for many signatures. However, our observation on parameter selection (i.e., to use w 1-CFFs instead of one w-CFF) gives a variant of the Reyzin-Reyzin scheme with smaller public keys and signatures, and thus gives a version of the BiBa protocol with reduced communication and computation, and improved security.

Acknowledgements

The authors would like to thank the members of the CrySP lab of the university of Waterloo for their feedback on earlier drafts of this paper (J. Clark, I. Goldberg, R. Henry, U. Hengartner, A. Kate, F. Olumofin and C. Swanson). This research was supported by NSERC discovery grant 203114-06, and an NSERC PGS scholarship. We also thank Payman Mohassel for pointing out [4].

References

[1] N. Asokan, V. Shoup and M. Waidner. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications* **18** (2000), 593–610.

[2] G. Ateniese. Verifiable encryption of digital signatures and applications. *ACM Trans. on Inf. and Systems Security (TISSEC)* **7** (2004), 1–20.

[3] M. Bellare, J. Garay and T. Rabin. Fast batch verification for modular exponentiation and digital signatures. *Proceedings of EUROCRYPT '98, LNCS* **1403** (1998), 236–250.

[4] M. Bellare and S. Shoup. Two-Tier Signatures, Strongly Unforgeable Signatures, and Fiat-Shamir Without Random Oracles *In Public Key Cryptography (PKC'07), LNCS* 4450 (2007) 201–216.

[5] D. J. Bernstein. Pippenger's exponentiation algorithm. Manuscript. `http://cr.yp.to/papers.html#pippenger`.

[6] K. Bicakci, G. Tsudik and B. Tung. How to construct optimal one-time signatures. *Computer Networks* **43** (2003), 339–349.

[7] K. Bicakci, C. Gamage, B. Crispo and A.S. Tanenbaum. One-time sensors: A novel concept to mitigate node-capture attacks. *Proceedings of Security and Privacy in Ad-hoc and Sensor Networks (ESAS'05), LNCS* **3813**, 80–90.

[8] D. Boneh, C. Gentry, B. Lynn and H. Shacham. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. *Proceedings of EUROCRYPT '03, LNCS* **2656** (2003), 416–432.

[9] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. *Journal of Cryptology* **17** (2004), 297–319.

[10] J. Bos and D. Chaum. Provably unforgeable signatures. *Proceedings of CRYPTO '92, LNCS* **740** (1992), 1–14.

[11] J. Buchmann, E. Dahmen, E. Klintsevich, K. Okeya and C. Vuillaume. Merkle signatures with virtually unlimited signature capacity. *Proceedings of Applied Cryptography and Network Security (ACNS'07), LNCS* **4521** (2007), 31–45.

[12] J. Camenisch and M. Stadler. Proof systems for general statements about discrete logarithms. Technical Report **TR 260** (1997), Institute for Theoretical Computer Science, ETH Zürich.

[13] J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. *Proceedings of CRYPTO '03, LNCS* **2729** (2003), 126–144.

[14] R. Canetti, S. Halevi and J. Katz. Chosen-ciphertext security from identity-based encryption. *Proceedings of EUROCRYPT '04, LNCS* **3027** (2004), 207–222.

[15] T.M. Cover. Enumerative source coding. *IEEE Transactions on Information Theory* **19** (1973), 73–77.

[16] B. Chor and R. Rivest. A knapsack-type public key cryptosystem based on arithmetic in finite fields. *IEEE Transactions on Information Theory* **34** (1988), 901–909.

[17] W. Dai. Crypto++: A free C++ class library of cryptographic schemes. Accessed January 2010, `http://www.cryptopp.com/`.

[18] E. Dahmen and C. Krauß. Short hash-based signatures for wireless sensor networks. *Proceedings of Cryptology and Network Security (CANS'09), LNCS* **5888** (2009), 463–476.

[19] I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. *Proceedings of EUROCRYPT'00, LNCS* **1807** (2000), 418–430.

[20] I. Damgård and M. Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. *Proceedings of PKC 2001*, LNCS **1992**, 119–136.

[21] C. Dods, N.P. Smart and M. Stam. Hash based digital signature schemes. *Proceedings of Cryptography and Coding 2005, LNCS* **3796** (2005), 96–115.

[22] S. Even, O. Goldreich and S. Micali. On-line/off-line digital signatures. *Journal of Cryptology* **9** (1996), 35–67.

[23] R. Genarro and P. Rohatgi. How to sign digital streams. *Proceedings of CRYPTO '97, LNCS* **1294** (1997), 180–197.

[24] S. Goldwasser, S. Micali and R.L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Computing* **17** (1988), 281–308.

[25] J. Groth Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. *In proceedings of ASIACRYPT'06, LNCS* 4284 (2006) 444–459.

[26] N. Gura, A. Patel, A. Wander, H. Eberle and S.C. Shantz. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. *Proceedings of CHES '04, LNCS* **3156** (2004), 118–132.

[27] J. Katz. Signature schemes with bounded leakage resilience. IACR ePrint Archive Report 2009/220, `http://eprint.iacr.org/2009/220`.

[28] D.L. Kreher and D.R. Stinson. *Combinatorial Algorithms: Generation, Enumeration and Search*, CRC Press, Boca Raton FL, 1999.

[29] L. Lamport. Constructing digital signatures from a one-way function. Technical Report CSL-98, SRI International, Palo Alto, 1979.

[30] M. Luk, A. Perrig and B. Whillock. Seven cardinal properties of sensor network broadcast authentication. In: *SASN '06: Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*, ACM Press (2006), 147–156.

[31] A.J. Menezes, P.C. van Oorschot and S.A. Vanstone. *Handbook of Applied Cryptography.* CRC Press LLC, Boca Raton, FL, 1996.

[32] R. Merkle. A certified digital signature. *Proceedings of CRYPTO '89, LNCS* **435** (1989), 218–238.

[33] P. Mohassel. One-time signatures and chameleon hash functions. To appear at *Selected Areas in Cryptography (SAC) 2010.*

[34] D. Naor, A. Shenhav and A. Wool. One-time signatures revisited: Have they become practical? IACR ePrint Archive Report 2005/442, `http://eprint.iacr.org/2005/442`.

[35] National Institute of Standards and Technology. Digital signature standard (DSS). *FIPS PUB* **186-2** (2000).

[36] P. Paillier and D. Vergnaud. Discrete-log-based signatures may not be equivalent to discrete log. *Proceedings of ASIACRYPT'05, LNCS* **3788** (2005), 1–20.

[37] F. Kargl, P. Papadimitratos, L. Buttyan, M. Müter, E. Schoch, B. Wiedersheim, T.-V. Thong, G. Calandriello, A. Held, A. Kung, J.-P. Hubaux. Secure vehicular communication systems: implementation, performance, and research challenges. *IEEE Communications Magazine* **46** (2008), 110–118.

[38] T.P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. *Proceedings of CRYPTO'91, LNCS* **1440** (1992), 129–140.

[39] A. Perrig. The BiBa one time signature and broadcast authentication protocol. *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS '01)*, ACM Press, New York, 2001, 28–37.

[40] J. Pieprzyk, H. Wang and C. Xing. Multiple-time signature schemes against chosen message attacks. *Proceedings of SAC '03, LNCS* **3006** (2003), 88–100.

[41] E. Sperner. Ein Satz Uber Untermengen einer endliche Menge. *Math. Zeit.* **27** (1928) 544–548.

[42] S. Rohde, T. Eisenbarth, E. Dahmen, J. Buchmann and C. Paar. Fast hash-based signatures on constrained devices. *Proceedings of CARDIS'08, LNCS* **5189** (2008), 104–117.

[43] M.O. Rabin. Digitalized Signatures. Foundations of Secure Computation, New York: Academic Press, 1978, pp.155–168.

[44] L. Reyzin and N. Reyzin. Better than BiBa: Short one-time signatures with fast signing and verifying. *Proceedings of ACISP '02, LNCS* **2384** (2002), 144–153.

[45] P. Rohatgi. A compact and fast hybrid signature scheme for multicast packet authentication. *Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS '99)*, ACM Press, New York, 1999, 93–100.

[46] D.R. Stinson, R. Wei and L. Zhu. Some new bounds for cover-free families. *Journal of Combinatorial Theory Series A.* **90** (2000), 224–234.

[47] D.R. Stinson and R. Wei. Generalized cover-free families. *Discrete Mathematics* **279** (2004), 463–477.

[48] P. Szczechowiak, L.B. Oliveira, M. Scott, M. Collier, R. Dahab. NanoECC: Testing the limits of elliptic curve cryptography in sensor networks. *Proceedings of EWSN '08, LNCS* **4913** (2008), 305–320.

[49] E. van Heyst and T.P. Pedersen. How to make efficient fail-stop signatures. *Proceedings of EUROCRYPT '92, LNCS* **658** (1993), 366–377.