A Post-Quantum One Time Signature Using Bloom Filter

Masoumeh Shafieinejad, Reihaneh Safavi-Naini
University of Calgary, Calgary, Canada

***Abstract*—Today's commonly used digital signatures will not be secure if a quantum computer exists. One time signatures (OTS) base security on the one way property of hash functions and will stay secure against an adversary with access to a quantum computer. These schemes however suffer from large public and private keys, as well as large signature size. We propose an OTS that uses Bloom filters to enhance the efficiency without sacrificing security, and show the required sizes of public/private keys, as well as the signature size will all reduce for the same security level.**

***Keywords*—Post-Quantum Digital Signature, One time Signature Scheme, Hash-based Signatures, Bloom Filter.**

I. Introduction

Digital signatures are one of the most widely used cryptographic primitive in electronic communication, and are used in commonly used secure protocols such as SSL(Secure Socket Layer) and TLS(Transport Layer Security). Security of widely use signatures including RSA[9], DSA[2], and ECDSA[4] rely on the hardness of factoring and discrete logarithm problems. Shor [11] proposed a polynomial time quantum algorithm for solving both problems, rendering all these signature schemes insecure. Two main categories of post-quantum signatures are those that rely on computational assumptions that do not have an efficient quantum algorithm, and One-time signature(OTS) schemes[1,7,8]. Example of assumptions in the former categories, are lattice-based and code-based assumptions[3,13,12]. In this paper we consider OTS schemes. These schemes are constructed from general one-way functions can be efficiently computed but hard to find a preimage for an element in the range of the function. A cryptographic hash function is considered as a good representation of one-way hash function. One attractive property of OTS is their flexibility in replacing the one-way function (hash function) with a new one if the older function become insecure. OTS scheme was first introduced by Lamport[7] and require large signature length as well as large secret and public key sizes. To enhance efficiency two important approaches are (i) proposed by Bos and Chaum[1] who used cover-free families to reduce the signature size and the number of required keys, and (ii) introduced by Winternitz[8] providing time and space trade-off. In this paper, follow the former approach and show to further reduce the key size by using Bloom filters. A Bloom filter[10] is a compact probabilistic data structure that represents a set of elements, and supports set membership queries. These structures have been widely used in distributed systems. We show that by using Bloom filters the public and private key and signature sizes of a cove-free based OTS can be substantially reduced. This poster is organized as follows. Section II gives an overview of OTS and relevant building blocks of the scheme, namely cover free families and Bloom filters. Section III describes our OTS signature scheme and provides security proof for it. Section IV evaluates time and space complexity of our scheme and compares it with other schemes for concrete parameters.

II. Preliminaries

We introduce the building blocks for the proposed signature scheme. One time signature(OTS), introduced by Lamport[7] is the main building block of our work. OTS is enhanced by using cover free families by Bos and Chaum[1]. Cover free families enable us to sign more messages than the original one time signature would allow. The third block is Bloom filter[10], applying which improves efficiency of the OTS.

A. One Time Signature

Lamport introduced the first OTS scheme[7]. The scheme proceeds as follows to sign a 1-bit message b. Two secrets, x_0, x_1, are chosen randomly as signing keys, and their corresponding images, computed under the one way function f, $y_0 = f(x_0)$ and $y_1 = f(x_1)$, form the public verification keys. The signature for message b is then x_b. Any party can verify the signature by evaluating f on x_b and comparing the result with y_b in the public verification key. The scheme is secure, fast and simple; however, it requires long signatures and twice secret/public keys as the message length in bits.

B. 1-Cover-Free Families

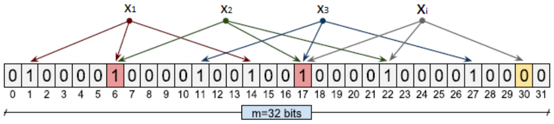
Cover-free families first used implicitly by Bos and Chaum[1] are described as follows. An n-uniform 1-cover-free family, *1-CFF(E,B)*, indicates a finite set of E elements and a collection(B) of subsets of size n. Any two distinct subsets in B differ on at least one element.

1-CFF in OTS: To use an optimal $n-$uniform *1-CFF(E,B)* for signing an $l-$bit message M, the parameters E and n are chosen such that: $\binom{E}{n} \geq 2^l$. The *1-CFF* provides a set of E secret keys(public keys are evaluations of the one way function on these secrets) and a collection, B, of subsets of size"n". We assume there is a bijection that maps a message $0 \leq M \leq \binom{E}{n}$ into the M^{th} subset in B denoted by B_M. To sign a message M, the secret keys in B_M are revealed. To verify a signature, verifier evaluates the one way function on the revealed secrets and compares them with the public keys.

C. Bloom Filter

Bloom filter[10] is a data structure, a binary array of size *m*, used for storing elements of a set. The array is filled with 0's first. To store a set as $\{x_1, x_2, x_3\}$, each element is mapped to (using hash function as H_1, H_2, H_3 mod m) *k* positions in the bloom filter. The bits in the bloom filter which are in the positions corresponding to hash outputs are then switched to 1, as shown in figure 1. We associate this mapping procedure with the function BF_store. BF_store takes an element of the set, applies *k* hash functions separately on the element, and stores it in the Bloom filter. The BF_check function on the other hand, is used to query for an element (test whether it is in the set). BF_check feeds the query to each of the *k* hash functions to get *k* array positions. If any of the bits at these positions is 0, the BF_check outputs 0. If all the bits would have been set to 1, the function outputs 1.

Figure 1. Bloom Filter of size 32 storing 3 elements [10]



$H_1(x_1)||H_2(x_1)||H_3(x_1) = 01000010000000100000000000000000$

$H_1(x_2)||H_2(x_2)||H_3(x_2) = 00000010000000000100001000000000$

$H_1(x_3)||H_2(x_3)||H_3(x_3) = 00000000000100000000001000010000$

Bloom filter $= 01000010000100100100001000010000$

False Positive Probability: In membership query application, false positive probability is the probability of an element(like x_i in figure above) not belonging to the the set but being mapped to positions in the bloom filter that are already filled with 1. For a bloom filter with desired false positive rate p, number of elements inserted *E*, and optimal number of hash functions, the array size must fulfill $m = \frac{-E \ln p}{(\ln 2)^2}$.

III. Signature Scheme

We describe three phases of our signature scheme; key generation & system setup, signing and verification for signing an l bit message M. We also define the security in digital signatures and provide a model to analyze the security of the proposed scheme.

A. Key Generation and System Setup

An optimal n-uniform *1-CFF(E, B)* is created, with parameters E and n such that: $\binom{E}{n} \geq 2^l$. The set of randomly chosen elements $X = \{x_1, x_2, \cdots, x_E\}$ form the secret signing keys. These keys are mapped by the BF_store function to the corresponding bloom filter and form the public verification key, $PK = BF(X)$.

B. Signing Messages

We assume there is a bijection that maps the message M, to the M^{th} subset in our *1-CFF(E,B)* denoted by B_M. To sign a message M using SK, the n-subset $B_M \in B$ is computed as $\{x_{i_1}, x_{i_2}, \cdots, x_{i_n}\}$ and revealed. Hence, the signature is: $\sigma = \{x_{i_1}, x_{i_2}, \cdots, x_{i_n}\}$.

C. Signature Verification

Given (M, σ, PK) as input, to verify σ's validity for M using PK, the verifier confirms whether all the elements in the set $\sigma = \{x_{i_1}, x_{i_2}, \cdots, x_{i_n}\}$ match the given public key, PK. In other words verification outputs 1 if and only if BF_check's output is 1 for all x_i's in σ.

IV. Performance Evaluation

We evaluate the scheme in time and space complexity as well as providing comparisons in concrete parameters.

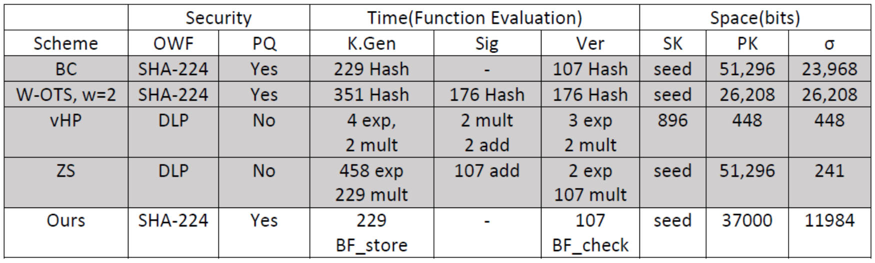
A. Time Complexity

Time complexity is measured in 3 signature's phases.
Key generation:In the generation time, a set of e random numbers, $X = \{x_1, x_2, \cdots, x_E\}$, is generated and BF_store maps each of them to the corresponding bloom filter to form the public key $PK = BF(X)$. So, the time complexity is E BF_store calculations, each BF_store requires *k* hash function computations.
Signing: Signing a message M is revealing the secrets in B_M, i.e. $\sigma = \{x_{i_1}, x_{i_2}, \cdots, x_{i_n}\}$, therefore it imposes no time complexity to the signature scheme.
Verification: During verification, the verifier applies BF_check to all secret keys in the given signature, $\sigma = \{x_{i_1}, x_{i_2}, \cdots, x_{i_n}\}$, to check if they are valid members of the public key $PK = BF(X)$, where: $X = \{x_1, x_2, \cdots, x_E\}$. So, the complexity is n BF_check, while each BF_check requires *k* hash function computations.

B. Space Complexity

Space complexity of the scheme is calculated in terms of secret key size, public key size, and the signature size.
Secret Key Size: As a set of E secret keys is used for a message of size l, the secret key size is then E times the secret key size, $E \times s$ bits.
Public Key Size: The public verification key is the

Figure 2. Comparison of the proposed signature and other(form [5]) schemes



	Security		Time(Function Evaluation)			Space(bits)		
Scheme	OWF	PQ	K.Gen	Sig	Ver	SK	PK	σ
BC	SHA-224	Yes	229 Hash	-	107 Hash	seed	51,296	23,968
W-OTS, w=2	SHA-224	Yes	351 Hash	176 Hash	176 Hash	seed	26,208	26,208
vHP	DLP	No	4 exp, 2 mult	2 mult 2 add	3 exp 2 mult	896	448	448
ZS	DLP	No	458 exp 229 mult	107 add	2 exp 107 mult	seed	51,296	241
Ours	SHA-224	Yes	229 BF_store	-	107 BF_check	seed	37000	11984

Bloom filter used in the scheme, it is of size m bits.

Signature Size: Signature is a subset of secret signing keys that is revealed to authenticate a message. while the system uses n-uniform *1-CFF*, n secrets are used to sign each message. Hence, signature size is $n \times s$ bits.

C. Concrete Parameters & Comparison

We select parameters to provide security level of $s = 112$ bits for signing messages of size $l = 224$ bits, to use SHA-224 as a suitable collision-resistant hash function for digital signatures. To sign 224-bit messages, we need a *1-CFF*'s with total number of $E = 229$ secret(and public) keys with subsets of size $n = 107$. Since: $\binom{229}{107} \geq 2^{224}$, the *1-CFF* enables us to sign all messages of length 224. Table above compares the proposed scheme with four other signature schemes; BC[1] W-OTS[8], ZS[13], vHP[3]. The scheme of van Heyst et al.[3] (vHP) essentially provides the best balanced performance using DLP. Zaverucha et al. [13] provide the shortest signatures using only DLP, but on the expense of much slower key generation phase and longer public-keys. Neither of the two is post-quantum. For our Bloom filter based scheme, we have to determine the filter parameters. As stated in our security analysis, the false positive probability of the filter should be less than 2^{-s}. To support that, we set $m = 37000$ for Bloom filter size and $k = 112$ for the number of hash functions.Our scheme provides signatures of 11,984bits (1.46KB), which is the shortest among all schemes based on general one-way functions, while keeping the similar time efficiency. Our signing is faster than W-OTS, as it does not require hash calculations. Our PK is better than BC, but not W-OTS. However, the public key size is not a main concern and can be reduced using a standard technique. Comparing with DLP-based schemes, our scheme has much better time complexity, but longer signatures. The ZS signature size is short; however, the key generation time takes 458 exponentiations.

V. Conclusion

In this paper we introduced a one time signature scheme that improves the hash-based OTS. The performance enhancement comes from using Bloom filter as the one way function embedded in the OTS scheme instead of hash functions. This change targets the main concern in hash-based OTS, which is memory requirements, while keeping the scheme quantum-safe. The proposed scheme reduces both the signature and the public verification key size in comparison with the original hash-based OTS.

References

[1] Bos J.N.E., Chaum D., *Provably unforgeable signatures*, In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 1–14. Springer, Heidelberg (1993)

[2] Elgamal T., *A public key cryptosystem and a signature scheme based on discrete logarithms*, IEEE Trans. Inf. Theor. 31(4), 469–472 (1985)

[3] van Heyst E., Pedersen T.P.*How to make efficient fail-stop signatures*, In: Rueppel, R.A. (ed.) EUROCRYPT 1992. LNCS, vol. 658, pp. 366–377. Springer, Heidelberg (1993)

[4] Johnson D., Menezes A., Vanstone S., *The elliptic curve digital signature algorithm (ECDSA)*, Int. J. Inf. Secur. 1(1), 36–63 (2001)

[5] Kalach K., Safavi-Naini Rei, *An Efficient Post-Quantum One-Time Signature Scheme*, SAC 2015, LNCS, volume 9566, pp. 331-351, Springer, Canada(2015)

[6] Katz J., Lindell Y., *Introduction to Modern Cryptography*, Chapman & Hall/CRC, 2007, ISBN:1584885513

[7] Lamport, L.: Constructing digital signatures from a one-way function. Technical report, SRI International Computer Science Laboratory (1979)

[8] Merkle R.C., *A certified digital signature*, In: Brassard, G. (ed.) CRYPTO 1989, LNCS, vol. 435, pp. 218–238. Springer, Heidelberg

[9] Rivest R.L., Shamir A., Adleman L., *A method for obtaining digital signatures and public-key cryptosystems*, Commun. ACM 21(2), 120–126 (1978)

[10] Sasu Tarkoma, Christian Esteve Rothenberg, and Eemil Lagerspetz, *Theory and practice of bloom filters for distributed systems*, IEEE Communications Surveys and Tutorials 14 (2012), no. 1, 131–155

[11] Shor, P.W., *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput. 26(5), 1484–1509 (1997)

[12] Wang X.M, *Digital signature scheme based on error-correcting codes*, Electronics Letters, (13):898–899, 1990

[13] Zaverucha G.M., Stinson D.R., *Short one-time signatures*, Adv. Math. Commun, 5, 473–488 (2011)