QIC 891 Topics in Quantum Safe Cryptography

Module 1: *Post-Quantum Cryptography*

Lecture 2

Lecturer: Fang Song *May 12, 2016*

Review.: 1) candidate problems: one-way& homomorphic; 2) MQ: schemes usually named after the center polynomials, such as (OV,UOV,HFE). Modifiers (e.g., +, -, s) are often used as ad-hoc tricks to add (+), discard (-), etc. a few polynomials.
Today. We will disucss main proposals for post-quantum (public-key) signature schemes. This lecture will cover two generic approaches that are based on hash functions possibly with additional algebraic properties. We will see another generic framework using functions with trapdoors in the next lecture.

1 Hash-based Signature

Hash-based signature scheme is essentially instantiating a generic approach of constructing signature schemes based on one-way functions with efficient cryptographic hash functions such as the SHA-2 family (SHA-256, SHA-512) and the latest standard SHA-3 (Keccak).

Cryptographic hash functions are efficient algorithms that map (usually compress) input message into fixed length output strings. They are designed in the hope that they behave similar to a totally random function (though this is impossible formally). In particular, finding a preimage for an output value as well as finding collisions (i.e. a pair of input messages that get mapped to the same output) are deemed difficult. Namely, they are assumed to be *one-way* (preimage resistant) and *collision-resistant*. In fact a common heuristic is to treat hash functions as a random oracle. This is the so called *random-oracle* model, which we will encounter frequently later on.

The main goal of this section is showing the following.

Theorem 1. *Assuming the existence of one-way functions (e.g. assuming SHA-2 is one-way), then there exists a secure signature scheme.*

The proof basically goes through two steps:

$$\mathsf{OWF}\text{ (One-way functions)} \xrightarrow{a} \mathsf{OTS}\text{ (One-time signature)} \xrightarrow{b} \text{full-fledged signature}$$

(a) Building a one-time signature scheme from one-way functions.

(b) Using the Merkle-tree technique to convert a one-way signature scheme to a full-fledged scheme that can sign unbounded many messages. The resulting scheme is stateful (some internal configuration needs to be recorded and updated dynamically with every signature). This can be removed using a pseudorandom function (which can be constructed based on OWF), getting a stateless signature scheme.

For simplicity, we will look at Lamport's OTS construction and a simple version of the Merkle-tree construction. Read [Kat10, Chapter 2] for more details and other discussion on the generic construction. In recent years many variants have been developed that are more efficient in terms of time complexity and sizes of verification key and signatures. For instance, Winternitz-OTS and some optimized Merkle-tree constructions have gain popularity (see e.g., [BDH11] and its followup works).
Definition of signature & security.

[...]

1.1 One-time Signature from OWF

Lamport's OTS construction [Lam79].

[standard materials...]

1.2 Getting full signature using Merkle-tree

Let $\Sigma = (G, S, V)$ be a secure OTS which can sign messages that are twice as long as its public key, i.e. $|m| = 2|pk|$. (Note that Lamport's scheme does not immediately provide this feature. But this can be achieved by compressing a long message with a universal-one-way hash function. UOWHF in turn can be constructed from a OWF.)

[Picture of a Merkle tree]

We construct a new signature scheme $\Sigma' = (G', S', V')$ that will be secure for signing multiple messages from Σ. Basically we maintain a tree of height h to sign all h-bit messages:

- we lable every left edge 0 and every right edge 1, and each node of the tree is labeled with the prefix of the path from the root. The root is denoted by ε. Each leaf (or rather path from root to leaf) corresponds to a message. For example the left-most leaf node corresponds to string $\underbrace{0\ldots0}_{h}$.
- each node is associated with a OTS key-pair (pk_p, sk_p) indexed by the path from the root to itself. Denote it $(pk_\varepsilon, sk_\varepsilon)$ at the root. They are generated independently and adaptively, which is part of the *state* that the signing algorithm maintains and keeps updating whenever producing a new signature.
- Signing a message m consists of
  1) $\sigma_0 := S(sk_m, m)$, signing m using the OTS signing algorithm and the leaf secret key.
  2) $\sigma_1 := (\mathsf{auth}_1^{(0)}, \ldots, \mathsf{auth}_1^{(h-1)})$, an "authentication" list that signs the two public keys of the children of each node on the path from root to leaf m. Specifically, each auth_1^j is associated with the node of m_j (jth prefix of the message m) and contains the public keys at m_j and its two children (i.e., pk_{m_j} and (pk_{m_j0}, pk_{m_j1})) as well as the signature $S(sk_{m_j}, (pk_{m_j0}, pk_{m_j1}))$. The Signer generates new key pairs of OTS Σ when necessary, and they are appended in the state that the Signer maintains.
- To verify $(m, (\sigma_0, \sigma_1))$, we first verify the authentication path specified by σ_1. Namely for every $j = 0, \ldots, h-1$, we check if $V(pk_{m_j}, (pk_{m_j0}, pk_{m_j1}), \sigma_1^j)$. If this passes, we accept if $V(pk_m, m, \sigma_0) = 1$.

See a complete description of $\Sigma' = (G', S', V')$ in Figure 1.

Figure 1: Full signature scheme from OTS using Merkle tree

We can show that Σ' is a secure signature scheme. Intuitively, this is because at any time a secret key at any node signs at most one message, which is either an actual message at the leaf node or a pair of public keys at two child nodes. The intuition can be turned into a formal reduction that breaks the one-time security of OTS Σ from any adversary that breaks Σ'.

Remark 1. In principle all the candidate problems from Lecture 1 give rise to one-way functions which can instantiate this generic construction. But people use SHA family for efficiency as well as security concerns. SHA family has resisted extensive cryptanalysis and there is no clear structure that seems feasible for quantum algorithms to exploit other than a generic quadratic speedup by Grover's search algorithm. Formal security analysis of this approach against quantum attacks can be found in [Son14].

2 Signature from "Homomorphic" hash functions

The second generic "trapdoor-less" approach for signatures follows the Fiat-Shamir paradigm realized by hash functiions that are homomorphic in certain sense. Basically

$$\text{“homomorphic” hash functions} \xrightarrow{a} \text{Identification (ID)} \xrightarrow{b} \text{signatures}$$

(a) Using hash functions with some homomorphic property (e.g., some functions from Lecture 1), we can construct an identification (ID) scheme. An ID scheme is basically an interactive protocol where a prover convinces a verifier that s/he is whoever s/he claims to be.

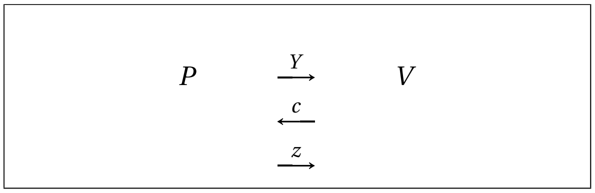
(b) Converting ID into a signature using the Fiat-Shamir transformation, which is a common heuristic to remove interaction in protocols. The transformation uses another hash function, which is treated as a random oracle.

For more information about identification and the Fiat-Shamir transformation, such as a formal definition of ID and other ID schemes based on factoring and discrete logarithm, see [Kat10, Chapter 8].

2.1 Identification from "Homomorphic" Hash

An ID scheme consists of (G, P, V): a key generating algorithm G and an interactive protocol described by two interactive machines P and V, called the prover and verifier respectively.

- $(pk, sk) \leftarrow G(1^\lambda)$. Secret sk is given to the prover, and pk is handed to verifier V.
- (P, V) executes the interactive protocol and in the end V output one bit indicating accepting or rejecting. We will consider atypical 3-round protocol.



Roughly we'd need an ID scheme to satisfy

- **Correctness**: A honest prover with the secret key will make the verifier accept with high probability.
- **Security**: we want that no one can impersonate a prover without knowing a secret key. But this is too weak and we need that the protocol is somewhat "zero-knowledge": even if an adversary $\mathscr{A}$ (e.g., a malicious verifier) has see transcripts of executing the protocol with a honest prover, $\mathscr{A}$ still cannot pass the verification by impersonating the prover and convincing another person.

  [a toy example to motivate security]

A template using "homomorphic" one-way hash functions.

This is motivated by some existing ID schemes based on factoring or discrete logarithm (e.g., Schnorr's ID scheme). Let $h : D \to R$ be a function that is homormorphic: $h(a + b) = h(a) + h(b)$. The

Sample $x \leftarrow D$ from domain. Set $pk = X = h(x), sk = x$.

- P samples $y \leftarrow D$, and sends $Y = h(y)$ to V.
- V picks $c \in_R C$ uniformlly at random. C is the challenge space which is $\{0,1\}$ here.
- P computes $z := y + cx$ and sends z to V.
- V accepts iff. $h(z) = Y + cX$.

Correctness follows from the homomorphic property of h since $h(z) = h(y) + h(cx) = Y + cX$. Security may be based on one-wayness of h. This protocol can be repeated in parallel while preserving the security. This would give rise to an ID scheme with large challenge space.

Instantiating based on lattices (e.g.,SIS).

One subtle issue arises when implementing the template using e.g., the SIS problem. The one-wayness holds for small-norm inputs only, therefore $y + x$ may leak some information about the secret x. To address this, some technique for "safety" check and aborting was introduced. This line of work started from [Lyu08] based on SIS, with further developments in [KTX08, Lyu09, Lyu12, DDLL13]. Efficiency has been improved both due to using ring-versions as well as more refined techniques for safety check. The most efficient scheme to date probably goes to called BLISS [DDLL13].

Remark 2. Classical security proof for ID schemes uses a technique of *rewinding*, which faces severe challenges against quantum attackers. We will say more in a future lecture.

Instantiating based on coding problems. Stern's scheme [Ste96] is the only wide-accepted one. It differs slightly from the template. The lattice-based ID scheme [KTX08] was actually inspired by Stern's scheme.

[Exercise: is the deviation from the template essential? Is it possible to get a variant?]

Instantiating with MQ. N/A.

2.2 Signature from Identification: Fiat-Shamir Paradigm

Let $\mathcal{O} : D \to R$ be uniformly drawn from all possible functions from its domain D to codomain R. We assume that $\mathcal{O}$ is provided as a black-box so that everyone can only query $\mathcal{O}$ to evaluate $x \mapsto \mathcal{O}(x)$. This is the so called random-oracle model.

Basically the Fiat-Shamir transform lets the prover run an instance of the identification protocol by itself, generating the challenge c by applying a hash function $\mathcal{O}$ to the first message Y and then computing an appropriate response z using the secret key. The signature.

Let (Gen, P, V) be a secure ID scheme.

- G: using the KeyGen algorithm of ID, i.e., $(sk, pk) \leftarrow Gen(1^\lambda)$.
- $S(sk, m)$: to sign a message m, run P and get $y, Y = h(y)$. Let $c := \mathcal{O}(Y, m)$. Then produce response $z = P(y, c)$. Output the signature being $\sigma := (Y, z)$.
- $V(pk, \sigma = (Y, z))$: evaluate $c := \mathcal{O}(m, Y)$ and accept iff. $V(Y, c, z)$ accepts.

Remark 3. The FS transformation is provably secure in the random oracle model. However in the quantum setting, it is natural to give the adversary quantum *superposition* access to the (hash function) oracle. Classically proofs fail at large in this new setting (so called *quantum random-oracle* model). We will come back to this issue in a future lecture.

3 Signing with trapdoors

[The following will be discussed next time.]

3.1 Full-Domain Hash

3.2 Instantiations

(Incomplete) summary: PQ-Sign schemes

Approach	**Instantiation**	**Remarks**
Without trapdoors		
Hash-based $\mathsf{OWF} \to \mathsf{OTS} \xrightarrow{*}$ Sign *: Merkle tree [Mer90]	SHA family [BDH11] and many variants, stateless scheme [BHH$^+$15]	Need: more quantum cryptanalysis considering internal design of SHA
“Homomorphic” hash hash $\to \mathsf{ID} \xrightarrow{*}$ Sign *: Fiat-Shamir	**Lattice**: safety check with aborting. SIS [Lyu08, KTX08], Ring-SIS [Lyu09], both [Lyu12, DDLL13] **Code**: [Ste96] **MQ**: N/A	[KTX08] inspired by [Ste96]. Adapt ideas from lattice ID to code-based? Quantum security unclear: quantum rewinding+FS in QRO
With trapdoors		
“Text-book” RSA $\sigma = f^{-1}(sk, m)$ e.g. $\sigma = m^d \pmod N$	**Lattice**: [GGH97], NTUSign [HPS01, HHGP$^+$03] (broken [NR09, DN12]) **Code**: early proposals broken **MQ**: majority (many broken)	Bad idea, **avoid**!
Hash-&-Sign in RO (Full-domain hash) $\sigma = f^{-1}(sk, \mathcal{O}(m))$ $\mathcal{O}$: random-oracle	**Lattice**: [GPV08, MP12] **Code**: [CFS01]. Formal proof of CFS01 in [Dal07], but one of the assumptions was disproved in [FGUO$^+$13].	adapt lattice ideas to code? Fix [CFS01] proof?
Direct constructions **without RO**	**Lattice**: [Boy10, CHKP12, DM14, AS15]	provably secure code & MQ-based unclear

References

[AS15] Jacob Alperin-Sheriff. Short signatures with short public keys from homomorphic trapdoor functions. In *Public-Key Cryptography–PKC 2015*, pages 236–255. Springer, 2015.

[BDH11] Johannes Buchmann, Erik Dahmen, and Andreas Hülsing. Xmss-a practical forward secure signature scheme based on minimal security assumptions. In *Post-Quantum Cryptography*, pages 117–129. Springer, 2011.

[BHH$^+$15] Daniel J Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. Sphincs: practical stateless hash-based signatures. In *Advances in Cryptology–EUROCRYPT 2015*, pages 368–397. Springer, 2015.

[Boy10] Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *Public Key Cryptography–PKC 2010*, pages 499–517. Springer, 2010.

[CFS01] Nicolas T Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology—ASIACRYPT 2001*, pages 157–174. Springer, 2001.

[CHKP12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *Journal of cryptology*, 25(4):601–639, 2012. Preliminary version in Eurocrypt 2010.

[Dal07] Léonard Dallot. Towards a concrete security proof of courtois, finiasz and sendrier signature scheme. In *Research in Cryptology*, pages 65–77. Springer, 2007.

[DDLL13] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In *Advances in Cryptology–CRYPTO 2013*, pages 40–56. Springer, 2013.

[DM14] Léo Ducas and Daniele Micciancio. Improved short lattice signatures in the standard model. In *Advances in Cryptology–CRYPTO 2014*, pages 335–352. Springer, 2014.

[DN12] Léo Ducas and Phong Q Nguyen. Learning a zonotope and more: Cryptanalysis of ntrusign countermeasures. In *Advances in Cryptology–ASIACRYPT 2012*, pages 433–450. Springer, 2012.

[FGUO$^+$13] Jean-Charles Faugere, Valérie Gauthier-Umana, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high-rate McEliece cryptosystems. *Information Theory, IEEE Transactions on*, 59(10):6830–6844, 2013.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology—CRYPTO'97*, pages 112–131. Springer, 1997.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206. ACM, 2008.

[HHGP$^+$03] Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H Silverman, and William Whyte. Ntrusign: Digital signatures using the ntru lattice. In *Topics in cryptology—CT-RSA 2003*, pages 122–140. Springer, 2003.

[HPS01] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. Nss: An ntru lattice-based signature scheme. In *Advances in Cryptology—Eurocrypt 2001*, pages 211–228. Springer, 2001.

[Kat10] Jonathan Katz. *Digital Signatures*. Springer, 2010.

[KTX08] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *Advances in Cryptology-ASIACRYPT 2008*, pages 372–389. Springer, 2008.

[Lam79] Leslie Lamport. Constructing digital signatures from a one-way function. *Tech. Report: SRI International Computer Science Laboratory*, 1979.

[Lyu08] Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *Public Key Cryptography–PKC 2008*, pages 162–179. Springer, 2008.

[Lyu09] Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *Advances in Cryptology–ASIACRYPT 2009*, pages 598–616. Springer, 2009.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In *Advances in Cryptology–EUROCRYPT 2012*, pages 738–755. Springer, 2012.

[Mer90] Ralph C Merkle. A certified digital signature. In *Advances in Cryptology–CRYPTO 1989*, pages 218–238. Springer, 1990.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Advances in Cryptology–EUROCRYPT 2012*, pages 700–718. Springer, 2012.

[NR09] Phong Q Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of ggh and ntru signatures. *Journal of Cryptology*, 22(2):139–160, 2009. Preliminary version in Eurocrypt 2006.

[Son14] Fang Song. A note on quantum security for post-quantum cryptography. In *Proceedings of the 6th International Workshop on Post-Quantum Cryptography*, volume 8772 of *Lecture Notes in Computer Science*, pages 246–265. Springer, 2014.

[Ste96] Jacques Stern. A new paradigm for public key identification. *Information Theory, IEEE Transactions on*, 42(6):1757–1768, 1996.