Cryptanalysis of RLWE-Based One-Pass Authenticated Key Exchange

Boru Gong, Yunlei Zhao

Fudan University, China

June 28, 2017

Outline

§1 Introduction

Lattice-based HMQV

A lattice-based analogue of HMQV was proposed at Eurocrypt 2015 [ZZDSD15].

- Similar to that of (DL-based) HMQV.
- It consists of a two-pass variant Π_2, and a one-pass variant Π_1.
- Both variants are proven secure under the (cyclotomic) ring-LWE assumption in the random oracle model (ROM).
  - A specific ring-LWE: the underlying number field K is the m-th cyclotomic number field $\mathbb{Q}(\zeta_m)$, where m is a power-of-two.

Our Contributions

In this work, we concentrate our analysis on the one-pass variant Π_1 in [ZZDSD15].

- We propose a special type of efficient attack against Π_1.
- Our attack is called **small field attack** (SFA), since it fully utilizes the algebraic properties of the ring $\mathcal{R}_q$ in ring-LWE.
- An SFA attacker can recover the private key of the victim party in Π_1 with overwhelming probability (*w.o.p.*)

The SFA attack may be applicable to other ring-LWE based one-pass AKE schemes.

Small Field Attack Against Π_1

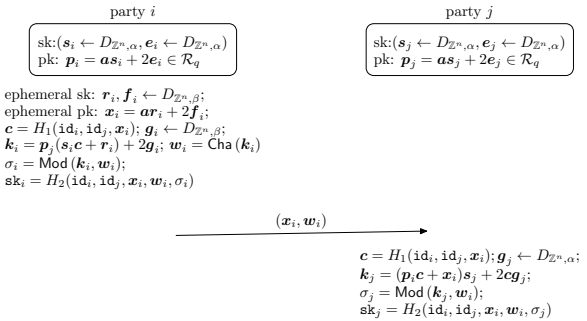
To be precise, two SFA attackers against Π_1 are proposed in this work.

- The **basic SFA attacker** is designed to demonstrate the notion of SFA.
- Furthermore, we can design an **advanced SFA attacker** that is "undetectable",
  - It is *hard in practice* for the victim party in Π_1 to identify both the static public key of SFA attacker, as well as those malicious query it makes.
  - Hence, our attack is *practical*.

We *stress* that the success of our attack relies on the **assumption** that the adversary can register a malicious public/private key pair on his own, which is beyond the security model of Π_1.

- Thus, although our attack is practical in essence, the existence of our attack does not violate the security of Π_1.

Introduction to Π_1



In Π_1,

- Party i and party j are involved.
- For party i:
  - Static sk: $(\mathbf{s}_i \leftarrow D_{\mathbb{Z}^n,\alpha}, \mathbf{e}_i \leftarrow D_{\mathbb{Z}^n,\alpha})$;
  - Static pk: $\mathbf{p}_i := \mathbf{a} \cdot \mathbf{s}_i + 2\mathbf{e}_i$ ($\mathbf{a}$ is a global parameter).
- Similar notations carry over to party j.
- To recover the (static) private key $(\mathbf{s}_j, \mathbf{e}_j)$ of party j, it suffices to recover $\mathbf{s}_j \in \mathcal{R}_q$.

Figure: A simplified depiction of Π_1

Party $j \Longrightarrow$ Oracle $\mathcal{M}_0$: 1/3

- In each session, party i sends $(\mathbf{x}_i, \mathbf{w}_i)$ to party j;
- For party j, the resultant session key is $\mathtt{sk}_j \leftarrow H_2(\mathtt{id}_i, \mathtt{id}_j, \mathbf{x}_i, \mathbf{w}_i, \sigma_j)$.
- **Observation**: for the hash input $(\mathtt{id}_i, \mathtt{id}_j, \mathbf{x}_i, \mathbf{w}_i, \sigma_j)$, all the values *except* σ_j are known to party i.
- When H_2 is modeled as an RO, if party i is able to figure out the session key $\mathtt{sk}_j$ of party j correctly *before* it issues the associated session-key query to party j, then party i must be able to figure out the associated σ_j *beforehand*, *and vice versa*.
- This observation enables us to simplify the description about SFA significantly.

Party $j \Longrightarrow$ Oracle $\mathcal{M}_0$: 2/3

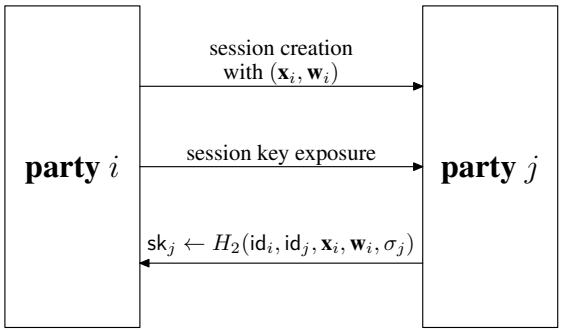


Figure: Some valid functionalities of party j

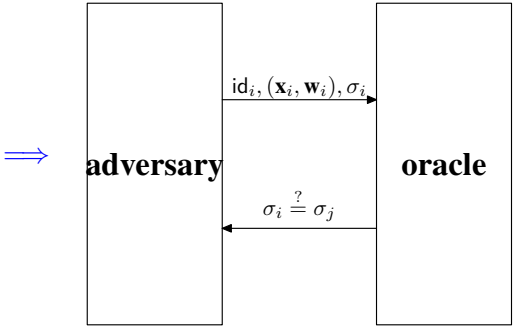


Figure: Oracle $\mathcal{M}_0$: an abstraction of party j

Party $j \Longrightarrow$ Oracle $\mathcal{M}_0$: 3/3

Claim

To recover the private key of party j in Π_1 efficiently,
it suffices to construct an efficient attacker against $\mathcal{M}_0$ (to be defined).

Formal definition of $\mathcal{M}_0$

The foregoing analysis motivates us to define an **oracle** $\mathcal{M}_0$ as follows:

- sk: $(\mathbf{s} \leftarrow D_{\mathbb{Z}^n,\alpha}, \mathbf{e} \leftarrow D_{\mathbb{Z}^n,\alpha})$;
  pk: $\mathbf{p} \triangleq \mathbf{a} \cdot \mathbf{s} + 2\mathbf{e} \in \mathcal{R}_q$;
  Identifier: $\mathtt{id}$.
- Given $(\mathtt{id}^*, \mathbf{p}^*, \mathbf{x}, \mathbf{w}, \mathbf{z})$ where $\mathtt{id}^*$ denotes the identifier of the adversary, $\mathbf{p}^*$ denotes the static public key of the adversary, $\mathbf{x} \in \mathcal{R}_q, \mathbf{w} \in \mathbb{B}^n, \mathbf{z} \in \mathbb{B}^n$,
  $\mathcal{M}_0$ does the following:

$$
\begin{aligned}
\mathbf{g} &\leftarrow D_{\mathbb{Z}^n,\alpha}, \\
\mathbf{c} &\leftarrow H_1(\mathtt{id}^*, \mathtt{id}, \mathbf{x}) \quad (\in \mathcal{R}_q), \\
\mathbf{k} &:= (\mathbf{p}^*\mathbf{c} + \mathbf{x})\mathbf{s} + q_0\mathbf{w} + 2\mathbf{c}\mathbf{g} \quad (\in \mathcal{R}_q), \quad (q_0 \triangleq \frac{q-1}{2}) \\
\sigma &:= \mathsf{Parity}(\mathbf{k}) \quad (\in \mathbb{B}^n);
\end{aligned}
$$

  Finally, $\mathcal{M}_0$ returns 1 if and only if

$$\sigma = \mathbf{z}.$$

Oracle $\mathcal{M}_0 \Longrightarrow$ Oracle $\mathcal{M}_1$

- Notice that in the definition of $\mathcal{M}_0$,

$$\mathbf{k} = (\mathbf{p}^*\mathbf{c} + \mathbf{x})\mathbf{s} + q_0\mathbf{w} + 2\mathbf{cg}.$$

- For an attacker against $\mathcal{M}_0$, if his static public key is $\mathbf{p}^* = \mathbf{0} \in \mathcal{R}_q$, the computation of $\mathbf{k}$ would be simplified dramatically.
- This motivates us to define the **oracle** $\mathcal{M}_1$ with secret $\mathbf{s} \leftarrow D_{\mathbb{Z}^n,\alpha}$ as follows:
  Given $(\mathbf{x}, \mathbf{w}, \mathbf{z}) \in \mathcal{R}_q \times \mathbb{B}^n \times \mathbb{B}^n$,
  it does the following:

$$\begin{aligned} \varepsilon &\leftarrow \mathbb{Z}^n_{1+2\theta}, \\ \mathbf{v} &:= \mathbf{xs} + q_0\mathbf{w} + 2\varepsilon \quad (\in \mathcal{R}_q), \\ \sigma &:= \mathsf{Parity}(\mathbf{v}) \quad (\in \mathbb{B}^n); \end{aligned}$$

  Finally, $\mathcal{M}_1$ returns 1 if and only if

$$\sigma = \mathbf{z}.$$

Intermediate Summary

Oracle $\mathcal{M}_0$: an abstraction of party j in Π_1

- To construct an efficient adversary against party j, it suffices to construct an efficient adversary against $\mathcal{M}_0$.

Oracle $\mathcal{M}_1$: a simplified variant of $\mathcal{M}_0$

- An efficient adversary against $\mathcal{M}_1$ corresponds to an efficient adversary against $\mathcal{M}_0$ with static public key $\mathbf{p}^* = \mathbf{0} \in \mathcal{R}_q$.

§2 The Basic SFA Attack (Against $\mathcal{M}_1$)

Difficulty in Attacking Π_1

- For the present, we aim to construct a (basic) attack against $\mathcal{M}_1$.
- Recall that $\mathcal{M}_1$ with secret $\mathbf{s} \leftarrow D_{\mathbb{Z}^n,\alpha}$ works as follows:
  - Each query is of the form $(\mathbf{x}, \mathbf{w}, \mathbf{z}) \in \mathcal{R}_q \times \mathbb{B}^n \times \mathbb{B}^n$;
  - On each query, it computes $\sigma \leftarrow \mathrm{Parity}\left(\mathbf{xs} + q_0\mathbf{w} + 2\varepsilon\right) (\in \mathbb{B}^n)$, and returns

$$\sigma \stackrel{?}{=} \mathbf{z}.$$

- Each time $\mathcal{M}_1$ returns only 1-bit information (with small noise) regarding its secret $\mathbf{s} \in \mathcal{R}_q$, which makes it difficult for the adversary to recover $\mathbf{s}$ *efficiently*.
  - Now, the CRT basis for $\mathcal{R}_q$ comes into play.

The CRT Basis for $\mathcal{R}_q$

- The notion of CRT basis in the ring-LWE setting was first proposed in [LPR10a].
- In the ring-LWE setting, $q \equiv 1 \pmod{m}$ is a positive rational prime.
  - Therefore, q **splits completely** in $K = \mathbb{Q}(\zeta_m)$, making $q\mathcal{R} = \prod_{i\in[n]} \mathfrak{q}_i$ *i.e.*, $q\mathcal{R}$ is the product of n *distinct* nonzero prime ideals in $\mathcal{R}$, each of norm q.
- It follows from Chinese Remainder Theorem that

$$\mathcal{R}_q \triangleq \mathcal{R}/q\mathcal{R} \cong \prod_{i\in[n]} \mathcal{R}/\mathfrak{q}_i.$$

  - Each $\mathcal{R}/\mathfrak{q}_i$ could be seen as a finite field of order q.
  - This isomorphism explains how our small field attack bears its name.
- Thus, there exist $\mathbf{c}_1, \cdots, \mathbf{c}_n \in \mathcal{R}_q$ such that

$$\mathbf{c}_i \equiv \delta_{i,j} \pmod{\mathfrak{q}_j}, \quad \forall i, j \in [n].$$

  - Such basis $\{\mathbf{c}_1, \cdots, \mathbf{c}_n\}$ is unique, and hence is called **the CRT basis for** $\mathcal{R}_q$.

Basic Properties of the CRT Basis for $\mathcal{R}_q$

$\{\mathbf{c}_1, \cdots, \mathbf{c}_n\}$ could be seen as an *integral* basis for $\mathcal{R}$. Moreover,

- Given n, q (in *unary* form), the CRT basis for $\mathcal{R}_q$ could be found *efficiently*.
- Every $\mathbf{u} \in \mathcal{R}_q$ can be written *uniquely* as $\mathbf{u} = \sum_{i \in [n]} u_i \cdot \mathbf{c}_i$, $u_i \in \mathbb{F}_q$.
  - Every $u_i \in \mathbb{F}_q$ is called a **CRT coefficient** of $\mathbf{u} \in \mathcal{R}_q$.
  - The set $\{i \in [n] \mid u_i \neq 0\}$ is called the **CRT-dimensionality** of $\mathbf{u}$.
- The map

$$\mathbf{u} \in \mathcal{R}_q \longmapsto (u_1, \cdots, u_n) \in \mathbb{F}_q^n$$

  is a ring homomorphism, *i.e.*, for every $\mathbf{u}, \mathbf{v} \in \mathcal{R}_q$, we have

$$\mathbf{u} + \mathbf{v} = \sum (u_i + v_i) \cdot \mathbf{c}_i, \quad \mathbf{u} \cdot \mathbf{v} = \sum (u_i v_i) \cdot \mathbf{c}_i.$$

An Algebraic Property of $\{\mathbf{c}_1, \cdots, \mathbf{c}_n\}$

Jumping ahead, the following lemma about $\{\mathbf{c}_1, \cdots, \mathbf{c}_n\}$ is useful for us to increase the efficiency of small field attack.

Lemma

Let

$$\mathbf{c}_i = \sum_{j \in [n]} c_{i,j} \cdot \zeta^{j-1} \in \mathcal{R}_q, \quad c_{i,j} \in \mathbb{F}_q.$$

When m is a power-of-two, the coefficients $c_{i,1}, \cdots, c_{i,n} \in \mathbb{F}_q$ form a *geometric* sequence in $\mathbb{F}_q$ for every $i \in [n]$.

General Strategy of $\mathcal{A}_1$: 1/2

We shall construct an efficient attacker $\mathcal{A}_1$ against $\mathcal{M}_1$ as follows:

- To recover the secret $\mathbf{s} \in \mathcal{R}_q$ of $\mathcal{M}_1$, it suffices to recover every CRT coefficient $s_i \in \mathbb{F}_q$ of $\mathbf{s}$.
- Application of the CRT basis:
  For the query $(\mathbf{x}, \mathbf{w}, \mathbf{z})$ made to $\mathcal{M}_1$, when $\mathbf{x} = k \cdot \mathbf{c}_i, k \in \mathbb{F}_q$, the product $\mathbf{xs}$ falls into the set $\{t \cdot \mathbf{c}_i \mid t \in \mathbb{F}_q\}$, which is of size $q = \mathrm{poly}(\lambda)$, making it *possible* for us to recover s_i efficiently.
- Search-to-decisional reduction:
  To recover s_i, we first pick $\tilde{s}_i \in \mathbb{F}_q$, and then verify whether $s_i = \tilde{s}_i$ or not via a sequence $\mathcal{Q}_i(\tilde{s}_i)$ of queries made to $\mathcal{M}_1$ such that
  - If $s_i = \tilde{s}_i$, then $\mathcal{M}_1$ returns 1 on *every* query in $\mathcal{Q}_i(\tilde{s}_i)$ *w.o.p.*;
  - If $s_i \neq \tilde{s}_i$, then $\mathcal{M}_1$ returns 0 on *at least one* query in $\mathcal{Q}_i(\tilde{s}_i)$ *w.o.p.*
- It remains to design $\mathcal{Q}_i(\tilde{s}_i)$.

General Strategy of $\mathcal{A}_1$: 2/2

- For every query $(\mathbf{x}_k = k \cdot \mathbf{c}_i, \mathbf{w}_k \in \mathbb{B}^n, \mathbf{z}_k \in \mathbb{B}^n)$, $\mathcal{M}_1$ computes

$$\begin{aligned} \mathbf{v}_k &= \mathbf{s} \cdot k\mathbf{c}_i + q_0\mathbf{w}_k + 2\varepsilon_k \\ &= \underbrace{k\tilde{s}_i \cdot \mathbf{c}_i + q_0\mathbf{w}_k}_{\text{known part}} + \underbrace{k \cdot \Delta s_i \cdot \mathbf{c}_i + 2\varepsilon_k}_{\text{unkown part}}; \end{aligned}$$

  where $\Delta s_i \triangleq s_i - \tilde{s}_i$.

- The difficulty of designing $\mathcal{Q}_i(\tilde{s}_i)$ lies in how to handle $k \cdot \Delta s_i \cdot \mathbf{c}_i + 2\varepsilon_k$.
  - The function $\mathrm{Cha}(\cdot)$ comes into play.

Properties of Cha(·)

In Π_1, a function $\mathrm{Cha} : \mathbb{F}_q \to \{0, 1\}$ is defined as follows:

$$\mathrm{Cha}(u) \triangleq 0 \Longleftrightarrow u \in \{-(q-1)/4, \cdots, (q-1)/4\}.$$

The following properties about Cha(·) are essential for Π_1, as well as for our SFA attack:

Properties of Cha(·)

For every $u \in \mathbb{F}_q$,

1. We always have $v \triangleq u + \mathrm{Cha}(u) \cdot q_0 \in \{-q_0/2, \cdots, +q_0/2\} \subseteq \mathbb{F}_q$.
2. The value $\mathrm{Parity}(v) \in \mathbb{B}$ is **immune to a small even noise** in the sense that

$$\mathrm{Parity}(v) = \mathrm{Parity}(v + 2e), \quad -q_0/4 < e < q_0/4.$$

3. The value $\mathrm{Parity}(v) \in \mathbb{B}$ is **sensitive to a small odd noise** in the sense that

$$\mathrm{Parity}(v + 2e - 1) \neq \mathrm{Parity}(v) \neq \mathrm{Parity}(v + 2e + 1), \quad -q_0/4 < e < q_0/4.$$

Design of $\mathcal{Q}_i(\tilde{s}_i)$: 1/2

- Recall that for every query $(\mathbf{x}_k = k \cdot \mathbf{c}_i, \mathbf{w}_k \in \mathbb{B}^n, \mathbf{z}_k \in \mathbb{B}^n)$, $\mathcal{M}_1$ computes

$$\begin{aligned} \mathbf{v}_k &= \mathbf{s} \cdot k\mathbf{c}_i + q_0\mathbf{w}_k + 2\varepsilon_k \\ &= k\tilde{s}_i \cdot \mathbf{c}_i + q_0\mathbf{w}_k + k \cdot \Delta s_i \cdot \mathbf{c}_i + 2\varepsilon_k; \end{aligned}$$

- Clearly, when $\Delta s_i = 0$, the unknown part $(k \cdot \Delta s_i \cdot \mathbf{c}_i + 2\varepsilon_k)$ is nothing but a small even noise; Thus, we can set

$$\mathbf{w}_k := \mathrm{Cha}\left(k\tilde{s}_i \cdot \mathbf{c}_i\right), \quad \mathbf{z}_k := \mathrm{Mod}\left(k\tilde{s}_i \cdot \mathbf{c}_i, \mathrm{Cha}\left(k\tilde{s}_i \cdot \mathbf{c}_i\right)\right),$$

  and $\mathcal{M}_1$ would always returns 1 *w.o.p.*
  - *In such setting*, if $\Delta s_i \neq 0$, then there exists a nonzero $k^* \in \{1, 2, \cdots, q_0\}$ such that

$$k^* \cdot \Delta s_i \cdot c_{i,1} = \pm 1.$$

    This forces $\mathcal{M}_1$ returns 0 on the k^*-th query.
  - Moreover, the lemma regarding $c_{i,1}, \cdots, c_{i,n}$ could be applied to make the query set $\mathcal{Q}_i(\tilde{s}_i)$ as *small* as possible.

Design of $\mathcal{Q}_i(\tilde{s}_i)$: 2/2

Correctness Analysis on $\mathcal{Q}_i(\tilde{s}_i)$

Let $g \in \mathbb{F}_q^{\times}$ denote a primitive element of $\mathbb{F}_q$, $S_g \triangleq \{g^r \mid r \in [d]\}$ and $d \triangleq q_0/n$.
Let

$$\mathcal{Q}_i(\tilde{s}_i) \triangleq \left\{ \left(k\mathbf{c}_i, [w_{k,j}]_{j\in[n]}, [z_{k,j}]_{j\in[n]}\right) \;\middle|\; \begin{array}{c} k \in S_g, \quad j \in [n], \quad u_{k,j} = \tilde{s}_i \cdot kc_{i,j}, \\ w_{k,j} = \mathrm{Cha}\left(u_{k,j}\right), z_{k,j} = \mathrm{Mod}\left(u_{k,j}, w_{k,j}\right) \end{array} \right\}.$$

If $q > 1 + \max\left\{8\theta,\ 2\alpha\sqrt{n}\right\}$, then *except with negligible probability*, $s_i = \tilde{s}_i$ if and only if $\mathcal{M}_1$ returns 1 on every query in $\mathcal{Q}_i(\tilde{s}_i)$.

- This finishes the construction of the desired efficient attacker $\mathcal{A}_1$ against $\mathcal{M}_1$.
- It is routine to check the correctness and efficiency of $\mathcal{A}_1$.

The SFA Attack Against $\mathcal{M}_1$

Until now,

- we have finished the construction of an efficient adversary $\mathcal{A}_1$ that can recover the secret of $\mathcal{M}_1$ *w.o.p.*
- The attacker is called a **small field attacker**, since it fully utilizes the algebraic properties of ring-LWE setting.
- The existence of $\mathcal{A}_1$ implies that we can construct an efficient attacker with static public key $\mathbf{p}^* := \mathbf{0} \in \mathcal{R}_q$ that can recover the private key of party j in Π_1 *w.o.p.*

However,

- It is easy to identify the $\mathbf{x}$-entry of each malicious queries made by $\mathcal{A}_1$.
- It is easy to identify both the static public key of the foregoing attacker against party j.
- So improvement is necessary to make the attack "undetectable".

§3 The Advanced SFA Attack (Against $\mathcal{M}_1$)

$$\mathcal{A}_1 \Longrightarrow \mathcal{A}_1'$$

- It is easy for $\mathcal{M}_1$ to identify queries made by $\mathcal{A}_1$, since the algebraic/numeric structure of the $\mathbf{x}$-entry is too simple, *i.e.*,

$$\mathbf{x} \in \left\{ k \cdot \mathbf{c}_i \,\middle|\, k \in \mathbb{F}_q^{\times}, i \in [n] \right\}.$$

- **Observation #1**: the attacker still works if a small and *even* noise is added into the $\mathbf{x}$-entry;
- **Observation #2**: when we work on recovering the $s_i \in \mathbb{F}_q$, the $s_1, \cdots, s_{i-1}$ have already been known; These could be used to "complicate" the structure of $\mathbf{x}$-entry.
- In sum, the attack still succeeds if

$$\mathbf{x} = k \cdot \mathbf{c}_i + \sum_{r \in [i-1]} h_r \cdot \mathbf{c}_r + 2\mathbf{e},$$

where $h_r \leftarrow \mathbb{F}_q^{\times}$ and $\mathbf{e} \leftarrow \mathbb{Z}_{1+2\alpha'}^n$.

The SFA Attacker $\mathcal{A}_1'$ Against $\mathcal{M}_1$

Thus, we can construct an improved variant of $\mathcal{A}_1$, *i.e.*, $\mathcal{A}_1'$, as follows:

- It consists of n-round loop, and the i-th round is devoted to the recovery of $s_i \in \mathbb{F}_q$;
- In the i-th round, $\mathcal{A}_1'$ first chooses $\tilde{s}_i \leftarrow \mathbb{F}_q$ randomly, and then verify whether $s_i = \tilde{s}_i$ or not via a sequence of random queries made to $\mathcal{M}_1$.
- It is routine to see that *except with negligible probability*, $s_i = \tilde{s}_i$ if and only if $\mathcal{M}_1$ returns 1 on every query in

$$\mathcal{Q}_i(\tilde{s}_i) = \left\{ \left(k\mathbf{c}_i + \mathbf{h}_k + 2\mathbf{e}_k, [w_{k,j}]_{j\in[n]}, [z_{k,j}]_{j\in[n]}\right) \;\middle|\; \begin{array}{c} k \in S_g,\ j \in [n],\ h_{k,1}, \cdots, h_{k,i-1} \leftarrow \mathbb{F}_q^{\times}, \\ \mathbf{h}_k = \sum_{r\in[i-1]} h_{k,r}\mathbf{c}_r, \mathbf{e}_k \leftarrow \mathbb{Z}^n_{1+2\beta\sqrt{n}}, \\ u_{k,j} = \tilde{s}_i \cdot kc_{i,j} + \sum_{r\in[i-1]} s_r h_{k,r} c_{r,j}, \\ w_{k,j} = \mathsf{Cha}\left(u_{k,j}\right), z_{k,j} = \mathsf{Mod}\left(u_{k,j}, w_{k,j}\right) \end{array} \right\}.$$

Limitation of $\mathcal{A}_1'$

- Compared with $\mathcal{A}_1$, the $\mathbf{x}$-entry of every query made by $\mathcal{A}_1'$ is much more "complex", making it much difficult for $\mathcal{M}_1$ to identify.
  - The more CRT-coefficients $\mathcal{A}_1'$ gets, the more difficult for $\mathcal{M}_1$ to identify.
- However, there is still one **limitation issue** regarding $\mathcal{A}_1'$.
  - When $\mathcal{A}_1'$ tries to recover *the first* CRT-coefficient of $\mathbf{s} \in \mathcal{R}_q$, the $\mathbf{x}$-entry of every query always falls into
    $$\{k \cdot \mathbf{c}_i + 2\mathbf{e} \mid k \in \mathbb{F}_q, i \in [n], \|\mathbf{e}\|_\infty \ll q\}.$$
  - Hence, it is not hard for $\mathcal{M}_1$ to identify, and thus reject, the *first* sequence of queries made by $\mathcal{A}_1'$.
    Then, $\mathcal{A}_1'$ cannot recover the first, and thus the remaining, CRT-coefficients of $\mathbf{s}$.
- Therefore, if similar **restrictions** are imposed by $\mathcal{M}_1$, more should be done to make our small field attacker "undetectable".

The Problems $\mathcal{P}_1$ and $\mathcal{P}_2$

- **Problem** $\mathcal{P}_1$: given oracle access to $\mathcal{M}_1$, recover the secret $\mathbf{s} \in \mathcal{R}_q$ of $\mathcal{M}_1$;
- **Problem** $\mathcal{P}_2$: given oracle access to $\mathcal{M}_1$, an arbitrary index set $I \subseteq [n]$, and $[\tilde{s}_i]_{i \in I} \in \mathbb{F}_q^{|I|}$, decide whether $[s_i]_{i \in I} = [\tilde{s}_i]_{i \in I}$ or not.
- Clearly, these two problems are firmly related to each other.
  - In particular, an efficient solver for $\mathcal{P}_2$ could be adapted to solve $\mathcal{P}_1$.
- Fortunately, we could construct such an efficient solver $\mathcal{V}$ for $\mathcal{P}_2$.
  - It should be stressed, of every query made by $\mathcal{V}$, the $\mathbf{x}$-entry is always of the form $\mathbf{x}_0 + 2\mathbf{e}$, where the **CRT-dimensionality** of $\mathbf{x}_0$ is I, and $\|\mathbf{e}\|_\infty \ll q$.
- Moreover, with the aid of $\mathcal{V}$, we could resolve the foregoing limitation issue regarding $\mathcal{A}_1'$.

The Hybrid Attacker $(\mathcal{V}/\mathcal{A}_1')_\delta$: 1/2

We can construct an "undetectable" attacker $(\mathcal{V}/\mathcal{A}_1')_\delta$ against $\mathcal{M}_1$ as follows.

- The $(\mathcal{V}/\mathcal{A}_1')_\delta$ consists of two phases.
- In **Phase 1**, we first pick an index set $I \subseteq [n]$ of size $\delta = |I|$ *randomly*; Then, feed $\mathcal{V}$ with q^δ instances, each of the form $(I, [\tilde{s}_i]_{i\in I}), \tilde{s}_i \in \mathbb{F}_q$.
  Thus, when $[\tilde{s}_i]_{i\in I}$ runs over the set $\mathbb{F}_q^\delta$, the CRT-coefficients $s_i, i \in I$, would be recovered *w.o.p.*
- **Phase 2** consists of $n - \delta$ rounds, each devoted to recovering one of the remaining $n - \delta$ CRT-coefficients of $\mathbf{s}$, as is done in $\mathcal{A}_1'$.
- The notation $(\mathcal{V}/\mathcal{A}_1')_\delta$ is applied to emphasize the structure of this *hybrid* attacker.

The Hybrid Attacker $(\mathcal{V}/\mathcal{A}'_1)_\delta$: 2/2

- To our knowledge, in order for $\mathcal{M}_1$ to identify those malicious queries made by $(\mathcal{V}/\mathcal{A}'_1)_\delta$, the most practical way is to check the algebraic/numeric structure of $\mathbf{x}$-entry, and the best algorithm to do the check on $\mathbf{x}$-entry is asymptotically close to the brute-force search.
- It should be stressed that, of every query made by $(\mathcal{V}/\mathcal{A}'_1)_\delta$, the $\mathbf{x}$-entry is always of the form $\mathbf{x}_0 + 2\mathbf{e}$, where the CRT-dimensionality of $\mathbf{x}_0 \in \mathcal{R}_q$ is of size at least δ, and $\|\mathbf{e}\|_\infty \ll q$.
- Given the randomness of the index set I (chosen in Phase 1), the hybrid attacker $(\mathcal{V}/\mathcal{A}'_1)_\delta$ is "undetectable" in practice.

§4 Small Field Attack (Against $\mathcal{M}_0$)

Recall

- To build an "undetectable" attacker against party j in Π_1, it suffices to construct an "undetectable" attacker against $\mathcal{M}_0$;
- The oracle $\mathcal{M}_0$ is defined as follows:
  Given $(\mathtt{id}^*, \mathbf{p}^*, \mathbf{x}, \mathbf{w}, \mathbf{z})$ where $\mathbf{x} \in \mathcal{R}_q, \mathbf{w} \in \mathbb{B}^n, \mathbf{z} \in \mathbb{B}^n$,
  it computes

$$\begin{aligned} \mathbf{g} &\leftarrow D_{\mathbb{Z}^n,\alpha}, \\ \mathbf{c} &\leftarrow H_1(\mathtt{id}^*, \mathtt{id}, \mathbf{x}) \quad (\in \mathcal{R}_q), \\ \mathbf{k} &:= (\mathbf{p}^*\mathbf{c} + \mathbf{x})\mathbf{s} + q_0\mathbf{w} + 2\mathbf{cg} \quad (\in \mathcal{R}_q), \\ \sigma &:= \mathsf{Parity}(\mathbf{k}) \quad (\in \mathbb{B}^n); \end{aligned}$$

  Finally, $\mathcal{M}_0$ returns 1 if and only if

$$\sigma = \mathbf{z}.$$

Small Field Attack Against $\mathcal{M}_0$: 1/3

- The foregoing $(\mathcal{V}/\mathcal{A}_1')_\delta$ against $\mathcal{M}_1$ corresponds to an efficient attacker against $\mathcal{M}_0$, whose static public key is $\mathbf{p}^* = \mathbf{0} \in \mathcal{R}_q$.
- To design an "undetectable" attacker against $\mathcal{M}_0$, the static public key $\mathbf{p}^*$ must be set so that it is as "random-looking" as possible.

Small Field Attack Against $\mathcal{M}_0$: 2/3

We can define an "undetectable" SFA attacker $\mathcal{A}_0$ against $\mathcal{M}_0$ as follows:

- $\mathcal{A}_0$ is very close to $(\mathcal{V}/\mathcal{A}'_1)_\delta$, and it consists of three phase;
- In **Phase 0**,
  - It first picks an index set $I \subseteq [n]$ of size δ randomly, and then chooses an element $\mathbf{t} \in \mathcal{R}_q$ randomly such that the CRT-dimensionality of $\mathbf{t}$ is I;
  - Let $\mathbf{e} \leftarrow \mathbb{Z}^n_{1+2\theta}$, and $\mathbf{p}^* := \mathbf{t} + 2\mathbf{e}$;
  - It is easy to find $(\mathbf{s}^*, \mathbf{e}^*) \in \mathcal{R}_q \times \mathcal{R}_q$ such that $\mathbf{a} \cdot \mathbf{s}^* + 2\mathbf{e}^* = \mathbf{p}^*$;
  - Let $\mathbf{p}^*$ be the static public key of $\mathcal{A}_0$, $(\mathbf{s}^*, \mathbf{e}^*)$ its static private key.
- The **Phase 1** of $\mathcal{A}_0$ is similar to that of $(\mathcal{V}/\mathcal{A}'_1)_\delta$, as it aims to recover the CRT-coefficients s_i of $\mathbf{s}$, $i \in I$.
- The **Phase 2** of $\mathcal{A}_0$ is similar to that of $(\mathcal{V}/\mathcal{A}'_1)_\delta$, as it aims to recover the remaining CRT-coefficients of $\mathbf{s}$.

Small Field Attack Against $\mathcal{M}_0$: 3/3

- The static public key $\mathbf{p}^*$ of $\mathcal{A}_0$ is of the form $\mathbf{t} + 2\mathbf{e}$, where the CRT-dimensionality of $\mathbf{t}$ is of size δ, and $\|\mathbf{e}\|_\infty \ll q$.
- Similar to $(\mathcal{V}/\mathcal{A}_1')_\delta$, for each query made by $\mathcal{A}_0$, the $\mathbf{x}$-entry is always of the form $\mathbf{x}_0 + 2\mathbf{e}$, where the CRT-dimensionality of $\mathbf{x}_0 \in \mathcal{R}_q$ is of size at least δ, and $\|\mathbf{e}\|_\infty \ll q$.
- In sum, both the static public key of $\mathcal{A}_0$ and its malicious queries are "undetectable" in practice.

§5 Conclusion

Conclusion

In this work,

- We propose a special type of efficient attack against Π_1 in [ZZDSD15], which may be applicable to other ring-LWE-based one-pass AKE schemes.
- This attack is called *small field attack*, since it fully utilizes the algebraic structure of $\mathcal{R}_q$ in ring-LWE.
- An SFA attacker can recover the private key of honest party j in Π_1 *w.o.p.*
  - The attack is beyond the security model of Π_1, and thus does not jeopardise the security of Π_1.
- Moreover, the SFA attacker against party j can be made "undetectable" in practice.

Thanks!