# Lattice-based Accumulators and Applications

## AP671316 - Master's Thesis in Computer Science

Chuanwei Lin

June 23, 2022

# Table of Contents

# Accumulators

A cryptographic accumulator provides

- Short representation for large sets
- Membership witnesses

Most constructions are

- Compact construction
- Merkle-tree style

# Laconic Cryptography

Laconic cryptography enables realizing cryptographic tasks with asymptotically-optimal communication in just two messages.

**Laconic Private Set Intersection Paradigm**

| **Sender** S | | **Receiver** R |
|---|---|---|
| Input: a small set $S_S$ | | Input: a large set $S_R$ |
| | $\text{psi}_1 \leftarrow \text{R}_1(\text{pp}, S_R)$ $\longleftarrow$ | |
| | $\text{psi}_2 \leftarrow \text{S}(\text{pp}, S_S, \text{psi}_1)$ $\longrightarrow$ | |
| | | $S_S \cap S_R \leftarrow \text{R}_2(\text{pp}, \text{psi}_2)$ |

# Current Status

- The most popular accumulators are not quantum secure.
- There are only two lattice-based accumulators, one with trapdoor and the other in Merkle-tree.
- There are no laconic PSI protocol from quantum-secure accumulators.

# Our Questions

- Is it possible to construct a compact accumulator from lattices without trapdoor?
- Can we construct quantum-secure PSI from lattice-based accumulators?

# Table of Contents

# Accumulators

An *one-shot* accumulator scheme parameterized by a domain $\mathcal{D}$ consists of four PPT algorithms

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$
- $\mathsf{Acc}(\mathsf{pp}, S) \to v_S$
- $\mathsf{Wit}(\mathsf{pp}, S, x) \to w$
- $\mathsf{Ver}(\mathsf{pp}, v_S, x, w) \to \texttt{accept/reject}$

# Accumulator for Laconic PSI

An ideal accumulator for laconic PSI protocol should satisfy the properties:

- The accumulation function $A = \mathsf{Acc}(X)$ allows to compress a set $X$ into a small representation $A$.
- The witness $w$ for an element $y$ is not a function of $y$, instead, it is generated from the set of $X$ except the element $y$, i.e., $w = \mathsf{Wit}(X_{-y})$.
- If $\mathsf{Ver}(\mathsf{Acc}(X), x, w) = 1$, there exist two functions $\psi, \phi$ such that $\phi\left(\psi(x), X_{-y}\right) = \mathsf{Acc}(X)$ and the functions output elements in a group where it holds that $\phi\left(\psi(y)^{\beta}, X\right) = \phi(\psi(y), X)^{\beta}$ for all $y, X$ and scalar $\beta$.

# Laconic PSI from Accumulators

**Private Membership Test from Accumulator**

| **Sender** S | | **Receiver** R |
|---|---|---|
| Input: a string $y$ | | Input: a set $X$ |
| sample random $\beta$ | | sample random $\alpha$ |
| | $R = \mathsf{Acc}(X)^{\alpha}$ $\longleftarrow$ | |
| | $T = R^{\beta}, U = \psi(y)^{\beta}$ $\longrightarrow$ | |
| | | $\forall x_i \in X$ : |
| | | If $\phi(U, X \setminus \{x_i\})^{\alpha} = T$ |
| | | output $y = x_i$ |

To extend this paradigm to a full PSI protocol, we need to have the sender send a different $(T, U)$ tuple for each $y$ in their input set.

# RSA-based Accumulators

- $\mathsf{Setup}(1^\lambda) \to \{N, g, \{p_i\}_{i\in[\ell]}\}$:
  1. Sample $N \leftarrow\$ \mathsf{RSA}(\lambda)$,
  2. Pick a random generator $g \in \mathbb{Z}_N^*$,
  3. Sample distinct primes $p_1, \ldots, p_\ell$ respectively for universe $\mathcal{U} = \{1, \ldots, \ell\}$.
- $\mathsf{Acc}(\mathsf{pp}, X) \to v_X$:
  Return $v_X = g^{\prod_{i\in X} p_i} \mod N$.
- $\mathsf{Wit}(\mathsf{pp}, X, x_i) \to w$:
  Return $w = g^{\prod_{j\in X_{-i}} p_j} \mod N$.
- $\mathsf{Ver}(\mathsf{pp}, v_X, x, w) \to \texttt{accept}/\texttt{reject}$:
  If $v_X = w^p \bmod N$ where $p$ is the prime corresponding to $x$, then return `accept`, otherwise, return `reject`.

# Laconic PSI from RSA-based Accumulators

## Laconic Private Membership Test from RSA

| **Sender** | | **Receiver** |
|---|---|---|
| Input: $Y = \{y\}$ | | Input: $X = \{x_1, \ldots, x_m\}$ |
| | **Trusted Party**<br>$N \leftarrow\$ \mathsf{RSA}(\lambda)$<br>$g \in \mathbb{Z}_N^*$ | |
| | $\longleftarrow$ | $r \leftarrow\$ [N]$ |
| | $(N, g)$ | |
| | | $R = g^{r \prod_{i \in [m]} H(x_i)} \bmod N$ |
| $t \leftarrow\$ [N]$ | $R$ $\longleftarrow$ | |
| $T = F\left(R^t \bmod N\right)$<br>$U = g^{t \cdot H(y)} \bmod N$ | | |
| | $\{T, U\}$ $\longrightarrow$ | $\forall k \in [m]:$ |
| | | $R_{-k} = U^{r \cdot \prod_{i \in [m] \setminus k} H(x_i)} \bmod N,$<br>if $F(R_{-k}) = T$, output 1 |

# Security of RSA-based Accumulators and $\ell$PSI Protocol

- The accumulator construction from RSA is collision-resistant under the strong RSA assumption.
- The laconic private set intersection protocol from RSA is correct and secure in the semi-honest model due to the usage of randomizers and the hardness of $\phi$-hiding assumption.

# Pairing-based Accumulators

- Setup$(1^\lambda) \rightarrow$ pp:
  1. Select a pairing instance pi $= (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$,
  2. Pick a random $s \in \mathbb{Z}_q^*$, and let the accumulator's domain $\mathcal{D} = \mathbb{Z}_q - \{s\}$.
  3. Compute a $(B+3)$-tuple where $B$ is the capacity of the accumulator $\left(g_1, g_1^s, g_1^{s^2}, \ldots, g_1^{s^B}, g_2, g_2^s\right)$.
  4. Return pp $= \left(\text{pi}, \left(g_1, g_1^s, g_1^{s^2}, \ldots, g_1^{s^B}, g_2, g_2^s\right)\right)$.
- Acc(pp, $S$) $\rightarrow v_S$:
  Return $v_S = g_1^{P(X,s)} = \prod_{i=1}^{|X|} \left(g_1^{s^i}\right)^{p(X,i)}$.
- Wit(pp, $X, x_i$) $\rightarrow w$:
  Return $w = g_1^{P(X_{-i},s)} = \prod_{i=1}^{|X|-1} \left(g_1^{s^i}\right)^{p(X_{-i},i)}$.
- Ver(pp, $v_S, x, w$) $\rightarrow$ `accept`/`reject`:
  If $e\left(v_X, g_2\right) = e\left(w, g_2^s \cdot g_2^x\right)$, then return `accept`, otherwise, return `reject`.

# Laconic PSI from RSA-based Accumulators

## Laconic Private Membership Test from Pairing

**Sender**

Input: $Y = \{y\}$

**Trusted Party**

$s \leftarrow\$ \mathbb{Z}_q^*, S' = g_1^s$

$S_i = g_2^{s^i}, i \in [0, m]$

$S'$ $\qquad$ $\{S_i\}_{i\in[m]}$

$t \leftarrow\$ \mathbb{Z}_q$

$T = H\left(e\left(g_1^t, R\right)\right)$

$U = \left(S' \cdot g_1^{-y}\right)^t$

$\{T, U\}$

**Receiver**

Input: $X = \{x_1, \ldots, x_m\}$

$r \leftarrow\$ \mathbb{Z}_q$

$$R = \left(\prod_{i=0}^{m} S_i^{p(X,i)}\right)^r$$

$R$

$$\forall k \in [m] : R_{-k} = \left(\prod_{i=0}^{m-1} S_i^{p(X_{-k},i)}\right)^r$$

If $H\left(e\left(U, R_{-k}\right)\right) = T$, output 1

# Security of RSA-based Accumulators and $\ell$PSI Protocol

- The accumulator construction from pairing is collision-resistant under the strong Diffie-Hellman assumption.
- The laconic private set intersection protocol from pairing is correct and secure in the semi-honest model due to the usage of randomizers and the hardness of Strong Bilinear Decisional Diffie-Hellman assumption.

# Table of Contents

# Essential Algorithms from Lattices

- There exists a PPT algorithm that extend a lattice to an arbitrary higher-dimensional extension, without any loss of quality, denoted by $\mathsf{BasisDel}(\cdot)$.
- Given a basis $\boldsymbol{B}$ for a lattice $\Lambda$, one can efficiently sample points in $\Lambda$ with discrete Gaussian distribution for some large $s$ and a center vector $\boldsymbol{c}$, denoted by $\mathsf{SampleD}(\cdot)$.

# Compact Construction with Trapdoor

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$:
  1. Generate a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ and a basis $\boldsymbol{T_A}$ for $\Lambda^\perp(\boldsymbol{A})$;
  2. Set the public key $\mathsf{pk} = \boldsymbol{A}$ and the secret key $\mathsf{sk} = \boldsymbol{T_A}$.
- $\mathsf{Acc}(S) \to v_S$: suppose $S = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{Q'}\}$ for $S \subset \mathcal{U}$.
  Return $v_S = \left[\sum_{i=1}^{Q'} \boldsymbol{B}_i\right] \in \mathbb{Z}_q^{n \times m'}$.
- $\mathsf{Wit}(\mathsf{pk}, \mathsf{sk}, S, \boldsymbol{B}_i) \to w$:
  Return
  $w \leftarrow \mathsf{SampleD}\left(\mathsf{BasisDel}\left(\boldsymbol{T_A}, \boldsymbol{A}, \sum_{1 \le j (\neq i) \le Q'} \boldsymbol{B}_j\right), s, \boldsymbol{0}\right)$.
- $\mathsf{Ver}(\mathsf{pk}, v_S, \boldsymbol{B}, w) \to \texttt{accept/reject}$:
  1. Compute $\boldsymbol{F_B} = [\boldsymbol{A} || (v_S - \boldsymbol{B})] \in \mathbb{Z}_q^{n \times (m+m')}$,
  2. Check if $\boldsymbol{F_B} \cdot w = \boldsymbol{0} \mod q$ and $0 < ||w|| \le s\sqrt{m+m'}$.
  3. If both checks pass, return `accept`, otherwise, return `reject`.

# Merkle-tree Construction

The hash function $h_{\boldsymbol{A}} : \{0,1\}^{nk} \times \{0,1\}^{nk} \mapsto \{0,1\}^{nk}$, described by a matrix $\boldsymbol{A} = [\boldsymbol{A}_0 \mid \boldsymbol{A}_1]$ with $\boldsymbol{A}_0, \boldsymbol{A}_1 \in \mathbb{Z}_q^{n \times nk}$, is defined as

$$h_{\boldsymbol{A}}(\boldsymbol{u}_0, \boldsymbol{u}_1) = \mathsf{bin}\left(\boldsymbol{A}_0 \cdot \boldsymbol{u}_0 + \boldsymbol{A}_1 \cdot \boldsymbol{u}_1 \bmod q\right) \in \{0,1\}^{nk}.$$

The hash function is collision-resistant under the hardness of SIS problem.

# Accumulator Construction

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$:
  Return $\mathsf{pp} = \boldsymbol{A} \leftarrow\$ \mathbb{Z}_q^{n \times m}$.
- $\mathsf{Acc}(\mathsf{pp}, S = \{\boldsymbol{d}_0, \ldots, \boldsymbol{d}_{N-1}\}) \to v_S$:
  1. Let $\boldsymbol{d}_j = \boldsymbol{u}_{[j_1 \ldots j_\ell]}$ be the leaf nodes,
  2. At depth $i \in [\ell - 1]$, the nodes $\boldsymbol{u}_{b_1 \ldots b_i}$ is defined as $h_{\boldsymbol{A}}(\boldsymbol{u}_{[b_1 \ldots b_i 0]}, \boldsymbol{u}_{[b_1 \ldots b_i 1]})$,
  3. At depth 0, the root $\boldsymbol{u} \in \{0,1\}^{nk}$ is defined as $h_{\boldsymbol{A}}(\boldsymbol{u}_{[0]}, \boldsymbol{u}_{[1]})$.

  Return the accumulated value as root value $\boldsymbol{u}$.
- $\mathsf{Wit}(\mathsf{pp}, S, \boldsymbol{d}_i) \to w$:
  Return the witness for $\boldsymbol{d}_i \in S$ as

$$w = \left( [j_1 \ldots j_\ell], \left( \boldsymbol{u}_{[j_1 \ldots j_{\ell-1} \bar{j_\ell}]}, \ldots, \boldsymbol{u}_{[j_1 \bar{j_2}]}, \boldsymbol{u}_{[\bar{j_1}]} \right) \right).$$

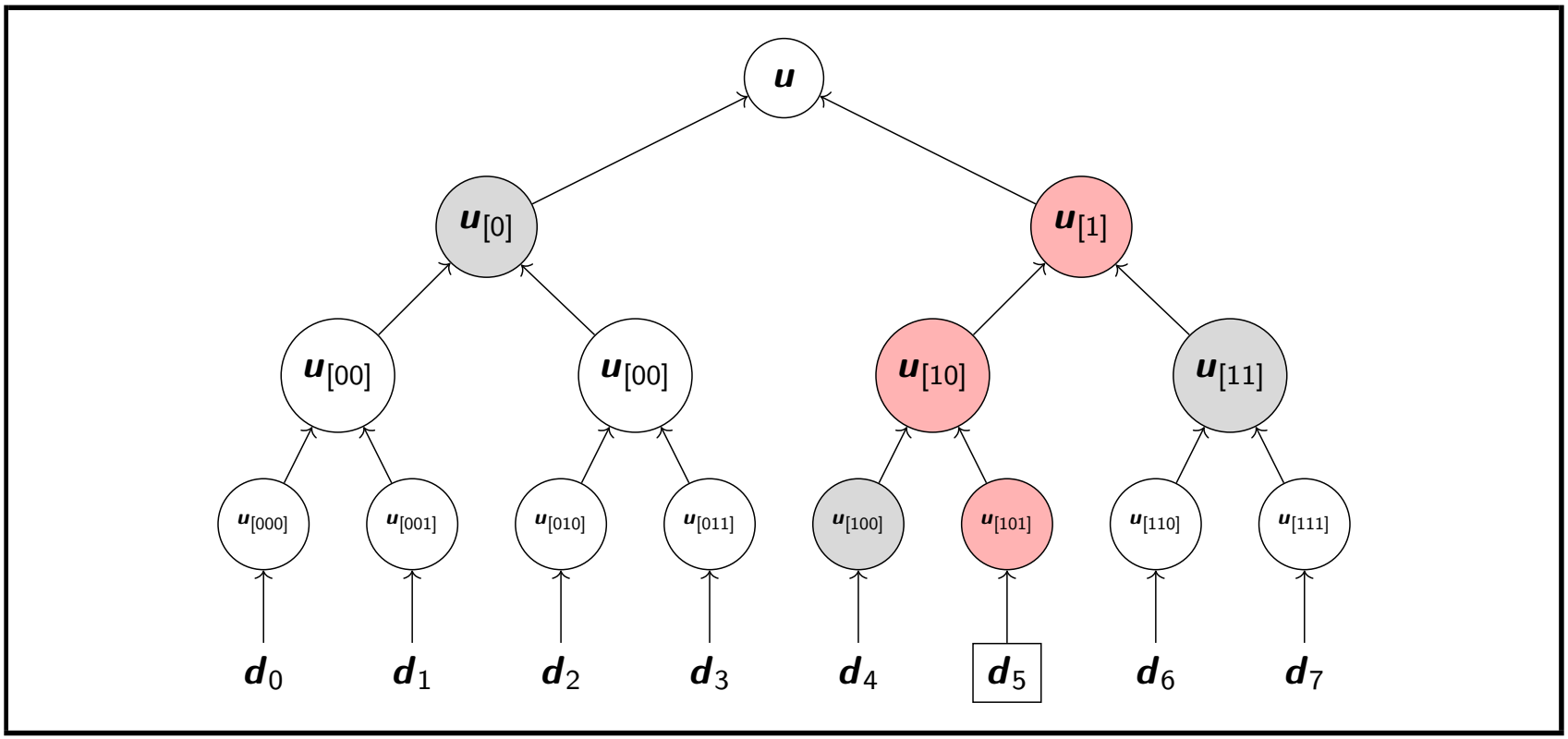
# Accumulator Construction

- $\mathsf{Ver}(\mathsf{pp}, v_S, \boldsymbol{d}, w) \rightarrow \texttt{accept/reject}$:
  Let the witness be of the form $w = ([j_1 \dots j_\ell], (\boldsymbol{w}_\ell, \dots, \boldsymbol{w}_1))$
  The path $\boldsymbol{v}_\ell, \dots, \boldsymbol{v}_1, \boldsymbol{v}_0$ can be computed as follows: $\boldsymbol{v}_\ell = \boldsymbol{d}$ and

$$\forall i \in \{\ell - 1, \dots, 1, 0\} : \boldsymbol{v}_i = \begin{cases} h_{\boldsymbol{A}}(\boldsymbol{v}_{i+1}, \boldsymbol{w}_{i+1}), & \text{if } j_{i+1} = 0 \\ h_{\boldsymbol{A}}(\boldsymbol{w}_{i+1}, \boldsymbol{v}_{i+1}), & \text{if } j_{i+1} = 1 \end{cases}$$

  If $\boldsymbol{v}_0 = \boldsymbol{u}$, then return `accept`, otherwise, return `reject`.

# Accumulator Example




In this accumulator, $S = \{\boldsymbol{d}_0, \ldots, \boldsymbol{d}_7\}$ is accumulated, represented by the root node $\boldsymbol{u}$. The witness of the element $\boldsymbol{d}_5$ is the element index [101] along with the gray nodes $(\boldsymbol{u}_{[100]}, \boldsymbol{u}_{[11]}, \boldsymbol{u}_{[0]})$.

# Stern's Protocol

P, V: The common inputs are $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ and $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \bmod q$. The prover P's auxiliary input is $\boldsymbol{x} \in \mathrm{B}(m, m/2)$.

- **Round 1**: P chooses a random permutation $\pi$ over $[m]$ and a random vector $\boldsymbol{r} \in \mathbb{Z}_q^m$ and send commitments
  - $\mathrm{com}_1 = \mathrm{Com}(\pi, \boldsymbol{A}\boldsymbol{r})$,
  - $\mathrm{com}_2 = \mathrm{Com}(\pi(\boldsymbol{r}))$,
  - $\mathrm{com}_3 = \mathrm{Com}(\pi(\boldsymbol{x} + \boldsymbol{r}))$.
- **Round 2**: V sends a random $\mathrm{ch} \in \{1, 2, 3\}$ to P.
- **Round 3**: The prover P responses upon the challenge ch received:
  - If $\mathrm{ch} = 1$, reveal $\mathrm{com}_2$ and $\mathrm{com}_3$. P sends $\boldsymbol{s} = \pi(\boldsymbol{x})$ and $\boldsymbol{t} = \pi(\boldsymbol{r})$.
    V checks that $\mathrm{com}_2 = \mathrm{Com}(\boldsymbol{t})$, $\mathrm{com}_3 = \mathrm{Com}(\boldsymbol{s} + \boldsymbol{t})$ and $\boldsymbol{s} \in \mathrm{B}(m, m/2)$.

# Stern's Protocol (cont.)

- **Round 3** (cont.): The prover P responses upon the challenge ch received:
  - If $\mathsf{ch} = 2$, reveal $\mathsf{com}_1$ and $\mathsf{com}_3$. P sends $\phi = \pi$ and $\boldsymbol{u} = \boldsymbol{x} + \boldsymbol{r}$.
    V checks that $\mathsf{com}_1 = \mathsf{Com}(\phi, \boldsymbol{Au} - \boldsymbol{y})$ and $\mathsf{com}_3 = \mathsf{Com}(\phi(\boldsymbol{u}))$.
  - If $\mathsf{ch} = 3$, reveal $\mathsf{com}_1$ and $\mathsf{com}_2$. P sends $\psi = \pi$ and $\boldsymbol{v} = \boldsymbol{r}$.
    V check that $\mathsf{com}_1 = \mathsf{Com}(\psi, \boldsymbol{Av})$ and $\mathsf{com}_2 = \mathsf{Com}(\psi(\boldsymbol{v}))$.

  The verifier V outputs 1 if all the checks are passed, otherwise outputs 0.

# Stern's Protocol (cont.)

- The protocol achieves a soundness error of $2/3$. By repeating the protocol $\tau$ times, the soundness error can be reduced to $(2/3)^{\tau}$.
- The protocol is 3-special sound, and thus it is a proof of knowledge for a relation

$$R = \{(\boldsymbol{A}, \boldsymbol{y}); \boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \bmod q \wedge \boldsymbol{x} \in \mathrm{B}(m, m/2)\}$$

# Zero-knowledge Proof for Accumulated Values

On input $(\boldsymbol{A}, \boldsymbol{u})$, the prover P proves to the verifier V that P owns $(\boldsymbol{d}, w)$ which is accepted by the verification algorithm Ver.

$$R = \left\{ \left( (\boldsymbol{A}, \boldsymbol{u}) \in \mathbb{Z}_q^{n \times m} \times \{0,1\}^{nk}; \boldsymbol{d} \in \{0,1\}^{nk}, w \in \{0,1\}^{\ell} \times \left(\{0,1\}^{nk}\right)^{\ell} \right) : \mathsf{Ver}(\boldsymbol{A}, \boldsymbol{u}, \boldsymbol{d}, w) = 1 \right\}$$

$\mathsf{Ver} = 1$ implies $\boldsymbol{v}_\ell = \boldsymbol{d}$, $\boldsymbol{v}_0 = \boldsymbol{u}$ and $\forall i \in \{\ell - 1, \ldots, 1, 0\}$ :

$$\boldsymbol{A} \cdot \begin{pmatrix} \bar{j}_{i+1} \cdot \boldsymbol{v}_{i+1} \\ j_{i+1} \cdot \boldsymbol{v}_{i+1} \end{pmatrix} + \boldsymbol{A} \cdot \begin{pmatrix} j_{i+1} \cdot \boldsymbol{w}_{i+1} \\ \bar{j}_{i+1} \cdot \boldsymbol{w}_{i+1} \end{pmatrix} = \boldsymbol{G} \cdot \boldsymbol{v}_i \bmod q$$

Techniques for Stern's type protocol:

extension technique, permutations and $\ell$ statements in parallel

# Table of Contents

# Incremental Hash Functions

Let $N, d$ be positive integers, $h : \{0,1\}^* \mapsto \{0,1\}^{Nd}$. For an input $x \in \{0,1\}^*$, the output is written as $\boldsymbol{h}(x) = ([h(x)]_1, \ldots, [h(x)]_N)$. Using $q = 2^d$, the hash function $\mathsf{LtHash}_{N,d} : \mathcal{P}(\{0,1\}^*) \mapsto \mathbb{Z}_q^N$ is defined as

$$\mathsf{LtHash}_{N,d}(\{x_1, \ldots, x_n\}) = \sum_{i=1}^{n} \boldsymbol{h}(x_i) \bmod q$$

where the sum operation is component-wise addition modulo $q$. The set incremental hash function is collision-resistant from the hardness of SIS problem.

# An Accumulator from Lattices

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$:
  Generate the description for $\mathsf{LtHash}_{N,d} : \mathcal{P}(\{0,1\}^*) \mapsto \mathbb{Z}_q^N$.
  Return $\mathsf{pp} = (q, h)$.
- $\mathsf{Acc}(\mathsf{pp}, X) \to v_X$:
  Return $v_X = \sum_{x_i \in X} \boldsymbol{h}(x_i) \bmod q$ where the sum operation is component-wise addition modulo $q$.
- $\mathsf{Wit}(\mathsf{pp}, X, x_i) \to w$:
  Return $w = \sum_{x_j \in X \setminus \{x_i\}} \boldsymbol{h}(x_j) \bmod q$.
- $\mathsf{Ver}(\mathsf{pp}, v_X, x, w) \to \texttt{accept}/\texttt{reject}$:
  If $v_X = \boldsymbol{h}(x) + w \bmod q$, then return `accept`, otherwise, return `reject`.

# Insecurity

- Given a value for the accumulator, say $v_X$ for the set $X$, the adversary can forge a witness for an element $y \notin Y$ as $w = v_X - \boldsymbol{h}(y) \bmod q$, which is not hard to compute.
- Then given $(y, w)$ as the value-witness pair, the verification algorithm Ver is more than happy to accept since it always holds that $w + \boldsymbol{h}(y) = v_X \bmod q$.

# Laconic PSI from Incremental Hashing

Laconic Private Membership Test from Lattices

| **Sender** | | **Receiver** |
|---|---|---|
| Input: $Y = \{y\}$ | | Input: $X = \{x_1, \ldots, x_m\}$ |
| | **Trusted Party** $h, g \leftarrow\$ \mathcal{H}(\lambda)$ | |
| | $\longleftrightarrow$ $(h, g)$ | $r \leftarrow\$ \mathbb{Z}_q^N$ |
| | | $R = r + \sum_{i \in [m]} \boldsymbol{h}(x_i) \bmod q$ |
| $s \leftarrow\$ \mathbb{Z}_q^N$ | $\longleftarrow R$ | |
| $T = s + R \bmod q$ | | |
| $U = s + \boldsymbol{h}\left((y_{\pi(j)})\right) \bmod q$ | | |
| | $\longrightarrow \{T, U\}_{j \in [n]}$ | $\forall k \in [m]$ : |
| | | $R_{-k} = U + r + \sum_{i \in [m] \setminus \{k\}} \boldsymbol{h}(x_i) \bmod q,$ |
| | | if $R_{-k} = T$, output 1 |

# Insecurity

- The usage of randomizers does not help to prevent an adversary fabricating a witness for some elements not in the receiver's set.

# Other Attempts

- There seems no solution for a zero-knowledge proof for accumulated values in the lattice-based accumulator with trapdoor.
- The current two lattice-based accumulators cannot fit into the laconic PSI construction.

# Further Work

To find a compact, lattice-based, trapdoor-less accumulator is not only to complete the types of accumulators, but to help to construct efficient protocols in the large data scale against quantum adversary as well.