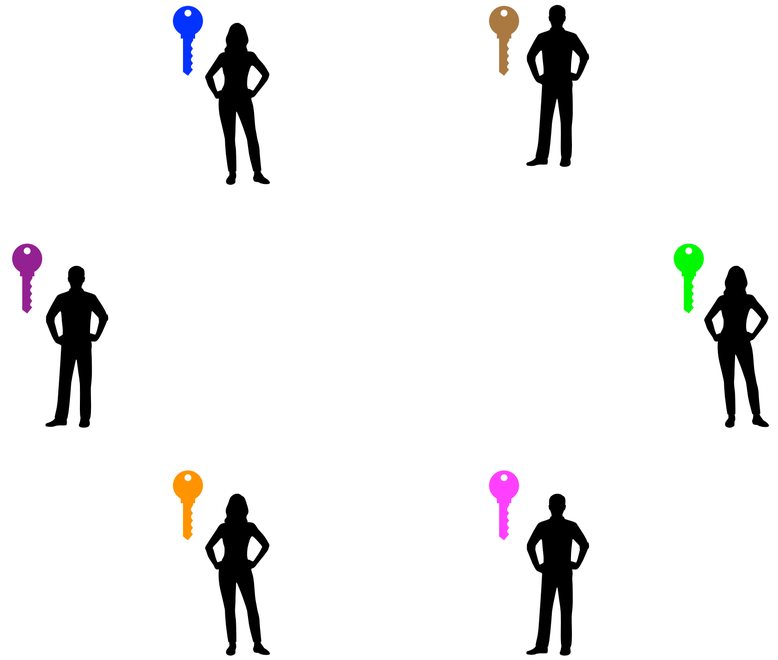
How to share lattice trapdoors

Literally

Based on the ongoing work with Martin Albrecht, Russell Lai and Ivy Woo.

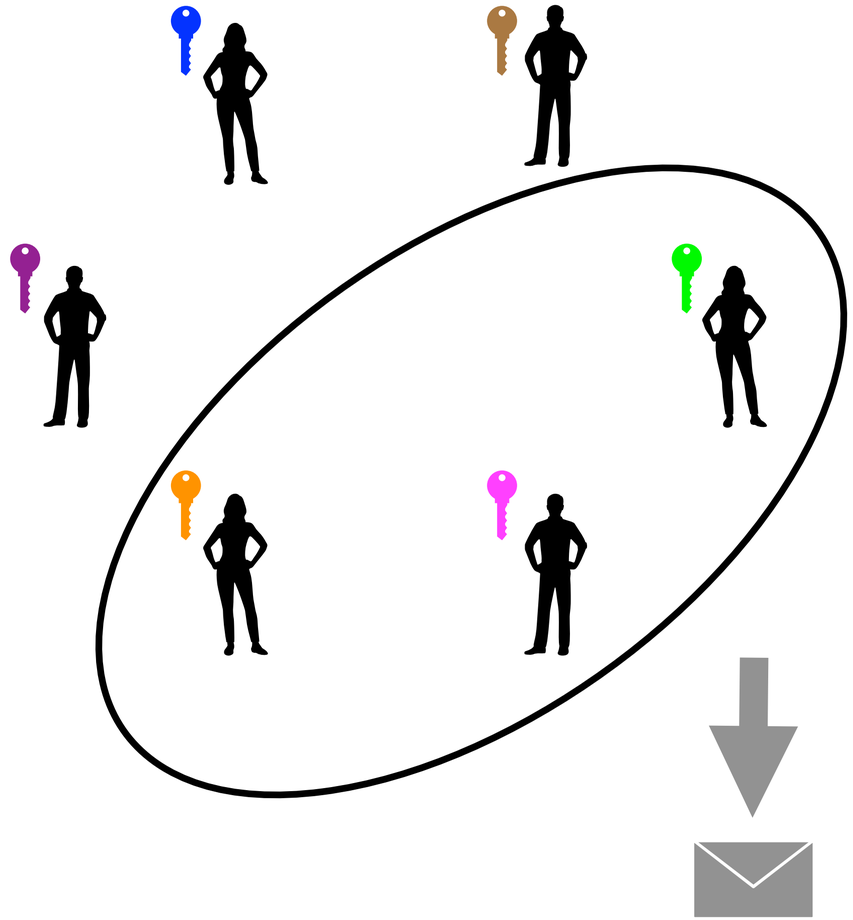
Threshold Cryptography



Goal: Distribute Trust
E.g. Threshold Signatures

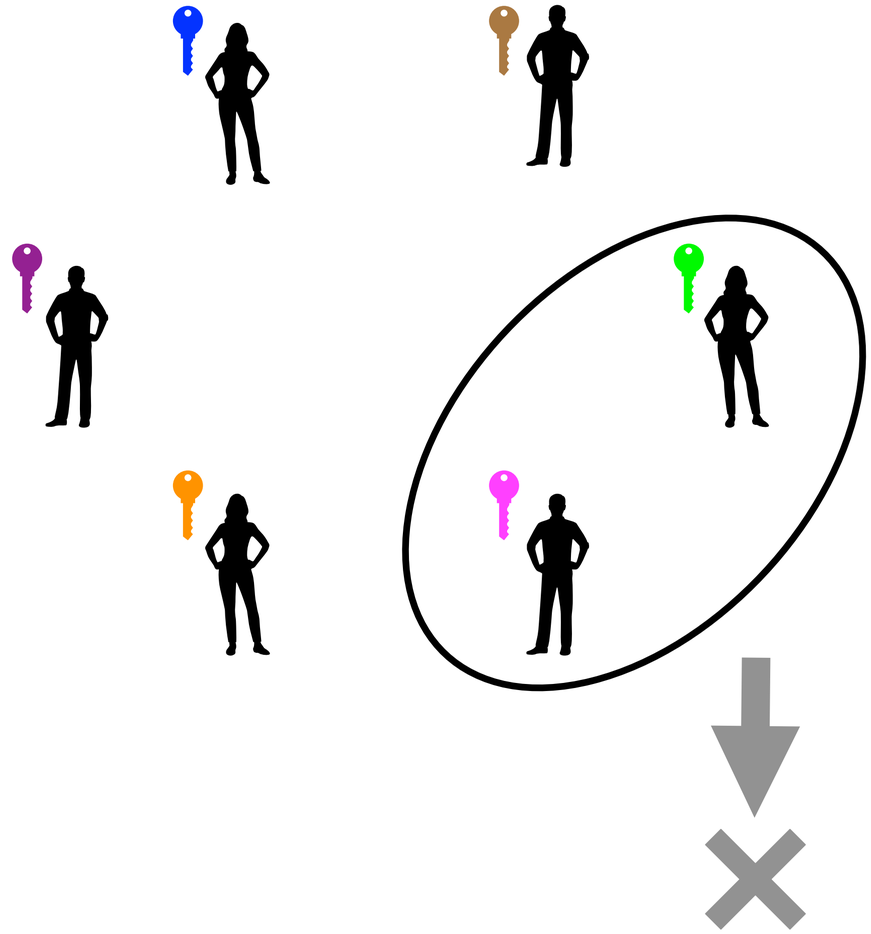
Threshold Signatures

Total number of *k* users
Any group *t* can sign

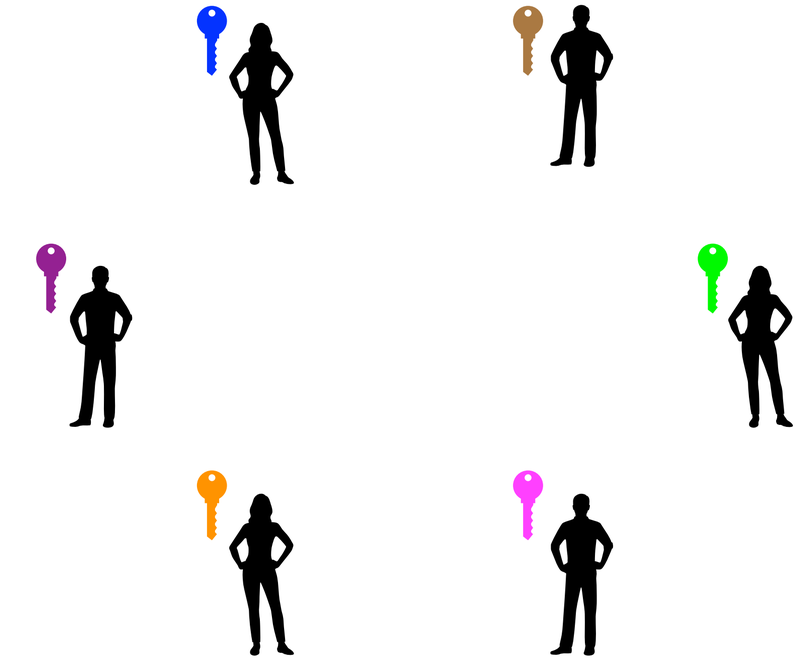


Threshold Signatures

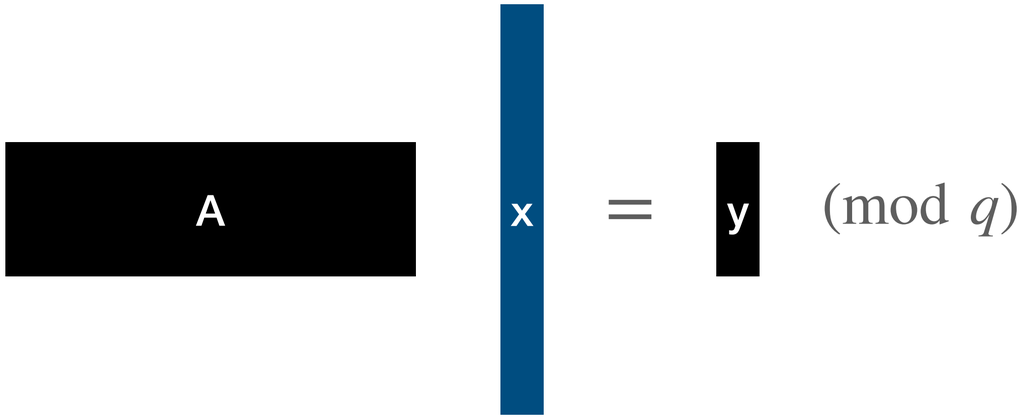
Total number of *k* users
Any group *t* can sign
No smaller group can produce a signature



Lattice Trapdoors

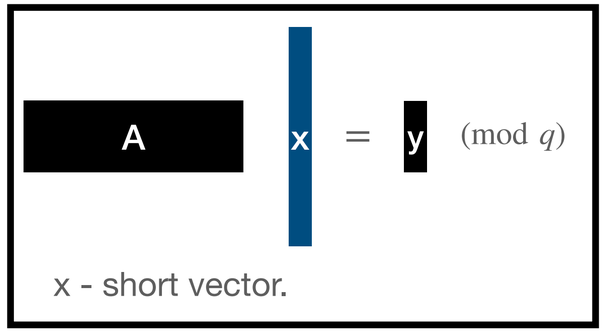


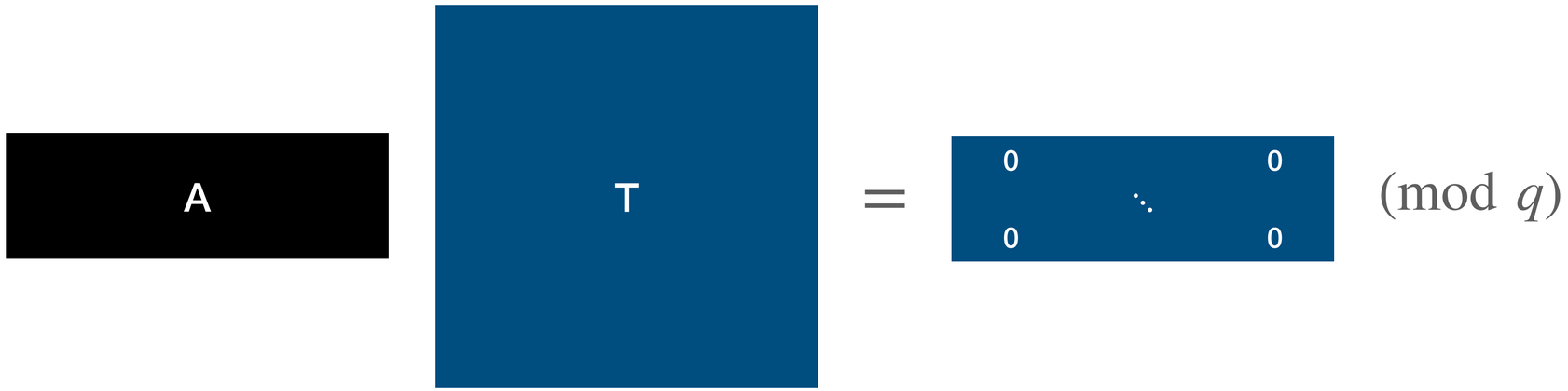
Lattice Trapdoors



x - sufficiently short vector.

Lattice Trapdoors

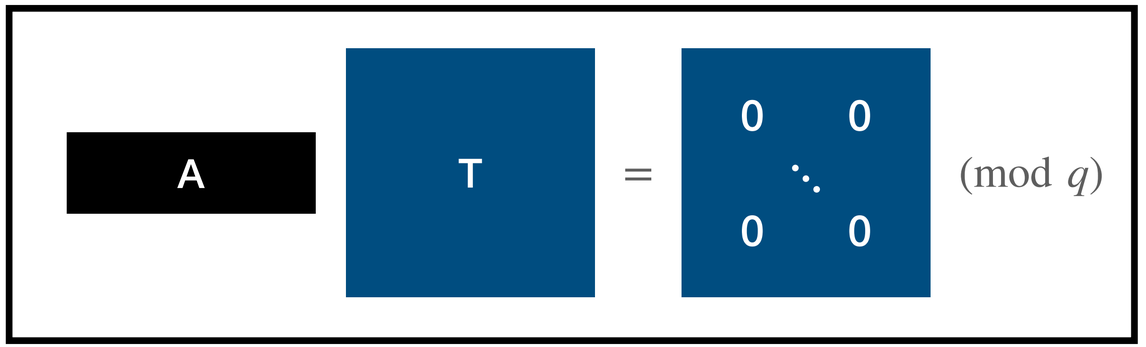




T - matrix of short vectors

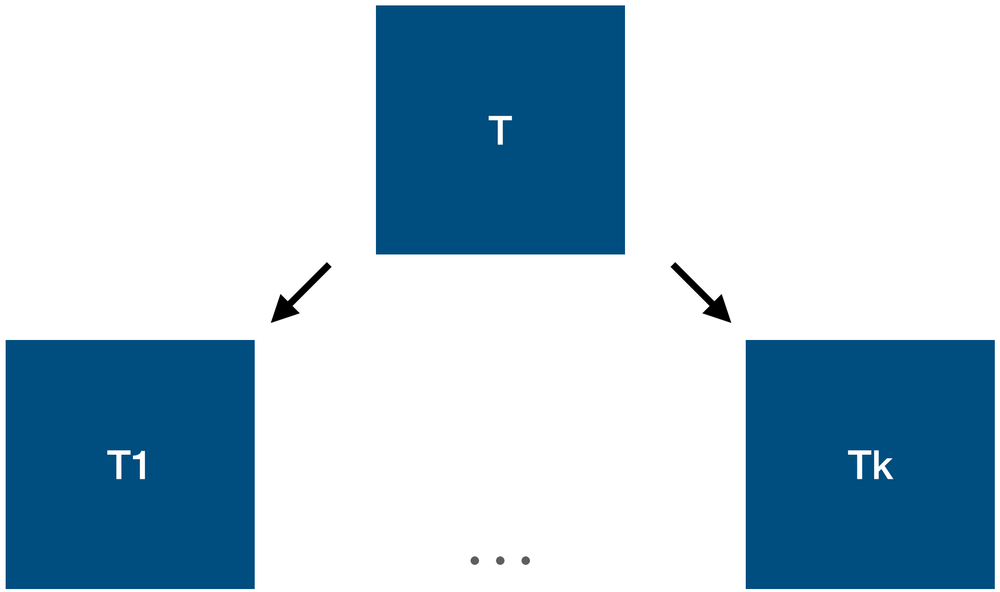
How to share T?

- Fancy Crypto
  - Homomophic encryption
  - Multi-party computation
- Natively
  - ??????

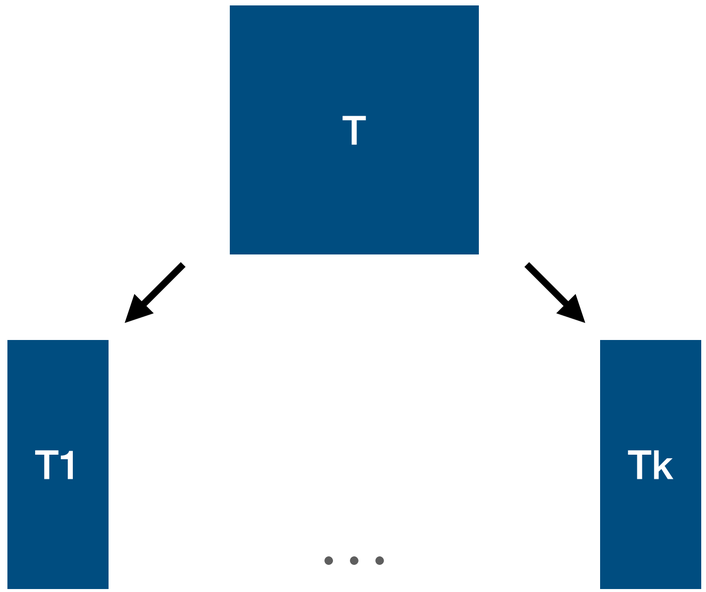


Bad Ideas

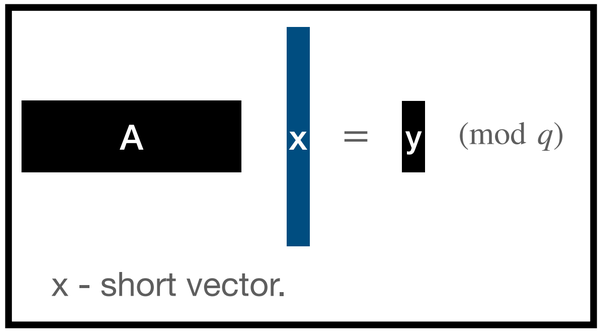
Give T - insecure

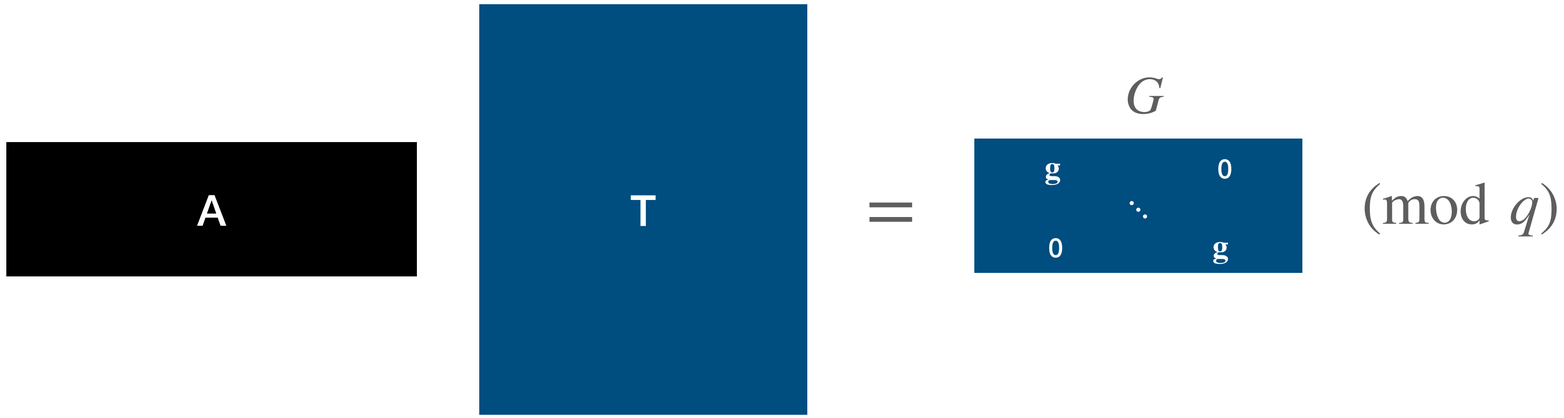


Split T - incorrect



Gadget Trapdoors

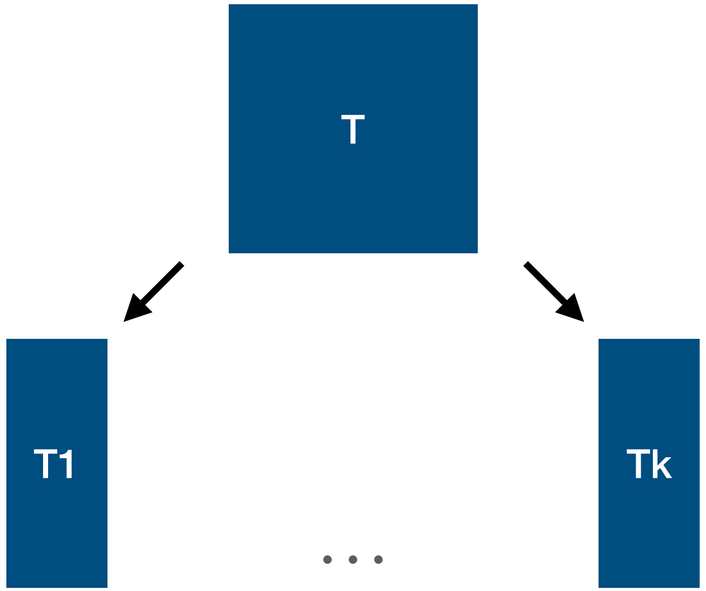




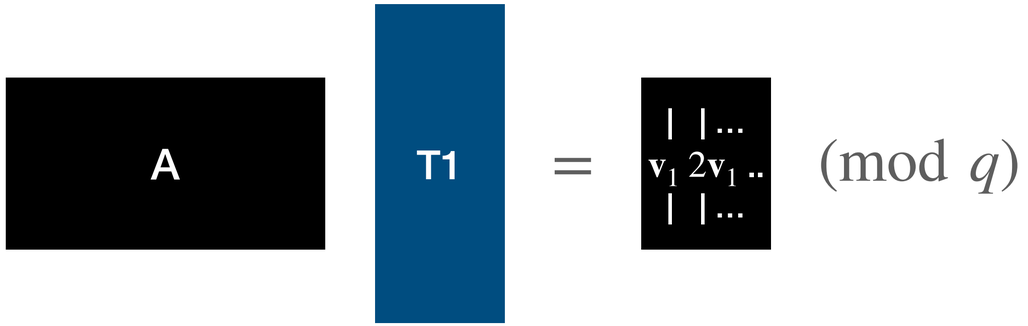
T - matrix of short vectors

Our Idea

Split T + Split the image space.

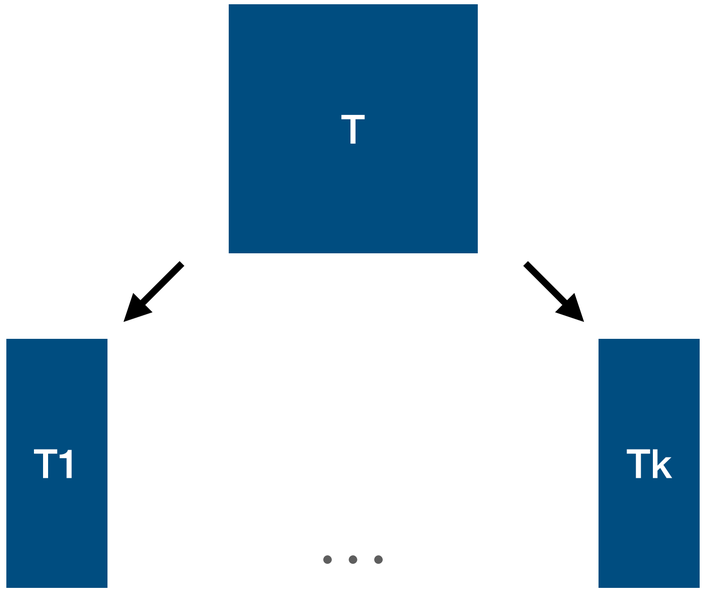


Given vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ such that any combination of t of them forms an invertible matrix.

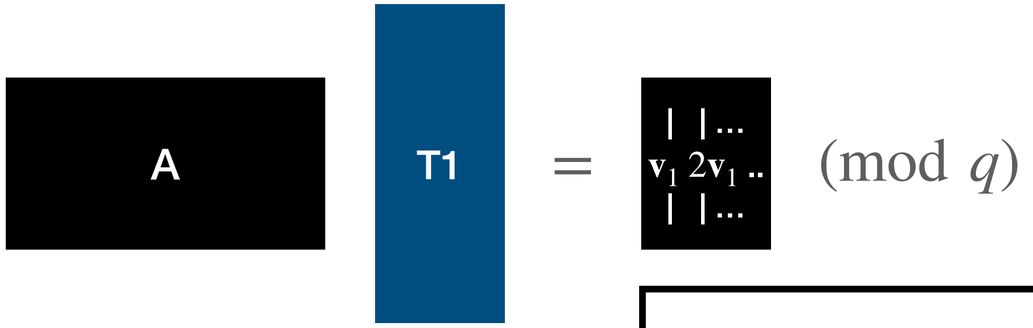


Our Idea

Split T + Split the image space.



Given vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ such that any combination of t of them forms an invertible matrix.



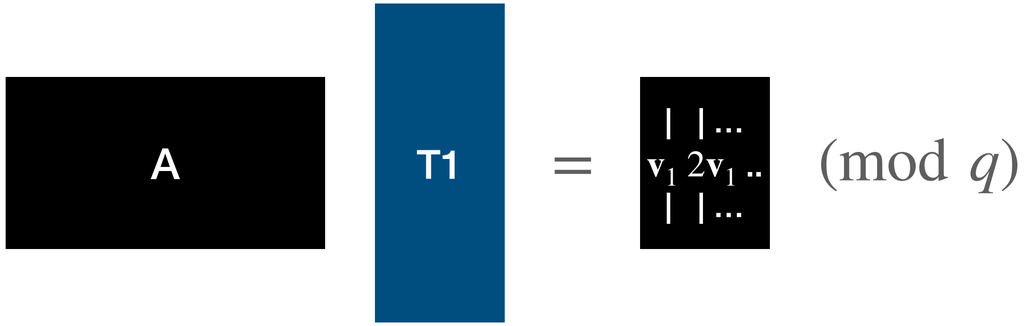
In the full version:

$$\mathbf{v}_1 \otimes \mathbf{g} \rightarrow \mathbf{v}_1 \otimes \mathbf{G}$$

Our Idea

To compute a preimage of $\mathbf{y}$ with parties $1, \ldots, t$

1. Set $\mathbf{V} = (\mathbf{v}_1 \mid \ldots \mid \mathbf{v}_t)$.

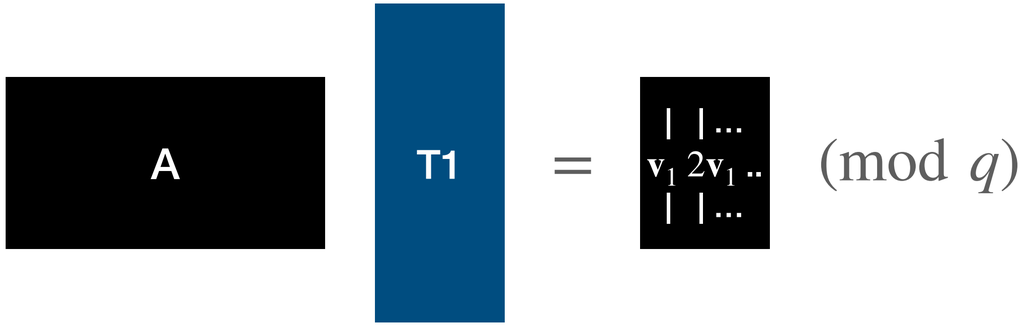


Our Idea

To compute a preimage of $\mathbf{y}$ with parties $1,\ldots,t$

1. Set $\mathbf{V} = (\mathbf{v}_1 \,|\, \ldots \,|\, \mathbf{v}_t)$.

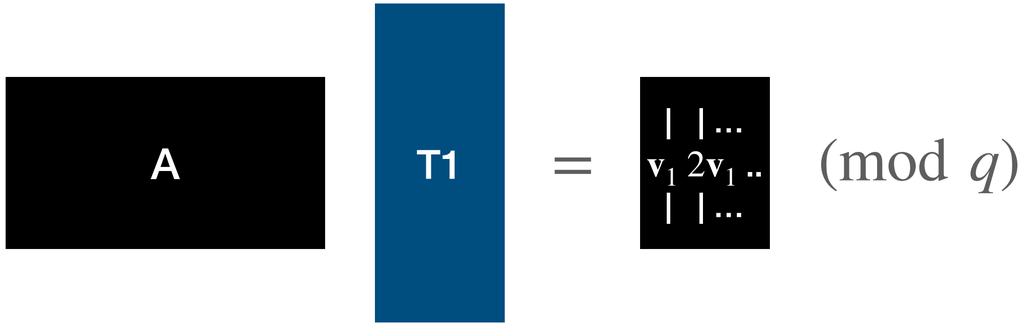
2. Compute $\mathbf{z} = \mathbf{V}^{-1}\mathbf{y}$ then we have: $\mathbf{V}\mathbf{z} = \mathbf{y}$.



Our Idea

To compute a preimage of $\mathbf{y}$ with parties $1, \ldots, t$

1. **Set $\mathbf{V} = (\mathbf{v}_1 \mid \ldots \mid \mathbf{v}_t)$.**
2. **Compute $\mathbf{z} = \mathbf{V}^{-1}\mathbf{y}$ then we have: $\mathbf{V}\mathbf{z} = \mathbf{y}$.**
3. **Party i computes $\mathbf{x}_i : \mathbf{A}\mathbf{x}_i = z_i\mathbf{v}_i$**

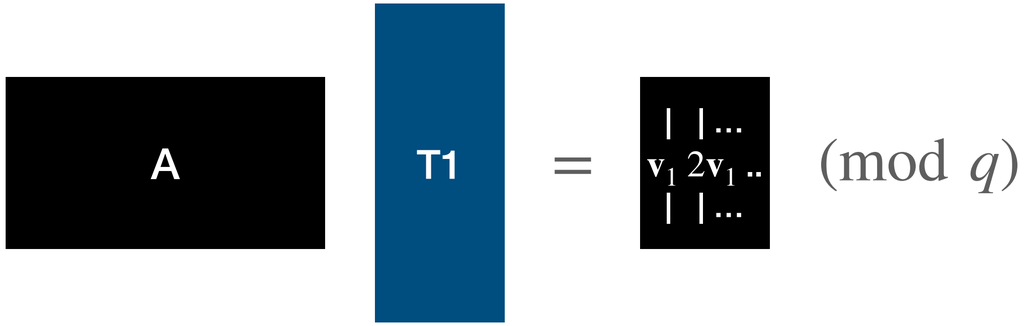


Our Idea

To compute a preimage of $\mathbf{y}$ with parties $1,\ldots,t$

1. **Set $\mathbf{V} = (\mathbf{v}_1 \,|\, \ldots \,|\, \mathbf{v}_t)$.**
2. **Compute $\mathbf{z} = \mathbf{V}^{-1}\mathbf{y}$ then we have: $\mathbf{Vz} = \mathbf{y}$.**
3. **Party i computes $\mathbf{x}_i : \mathbf{Ax}_i = z_i\mathbf{v}_i$**
4. **The signature is the sum:**

$$\mathbf{A}\Sigma\mathbf{x}_i = \Sigma z_i\mathbf{v}_i = \mathbf{Vz} = \mathbf{y}$$

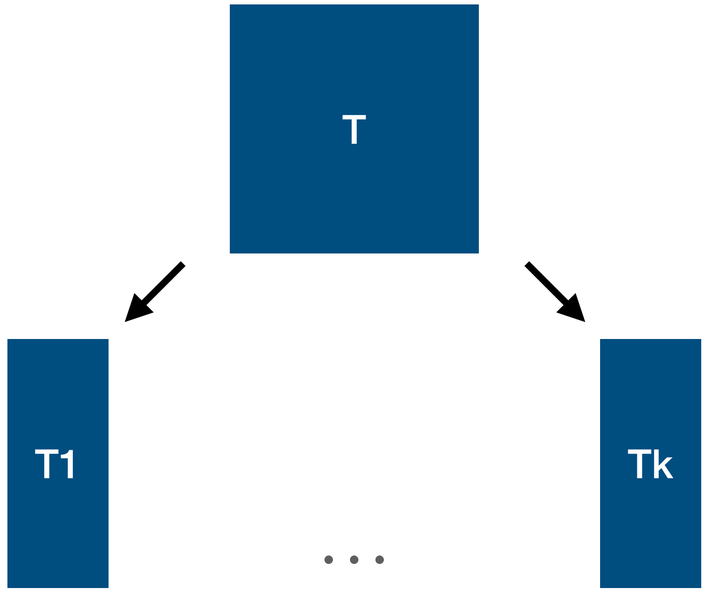


Challenges

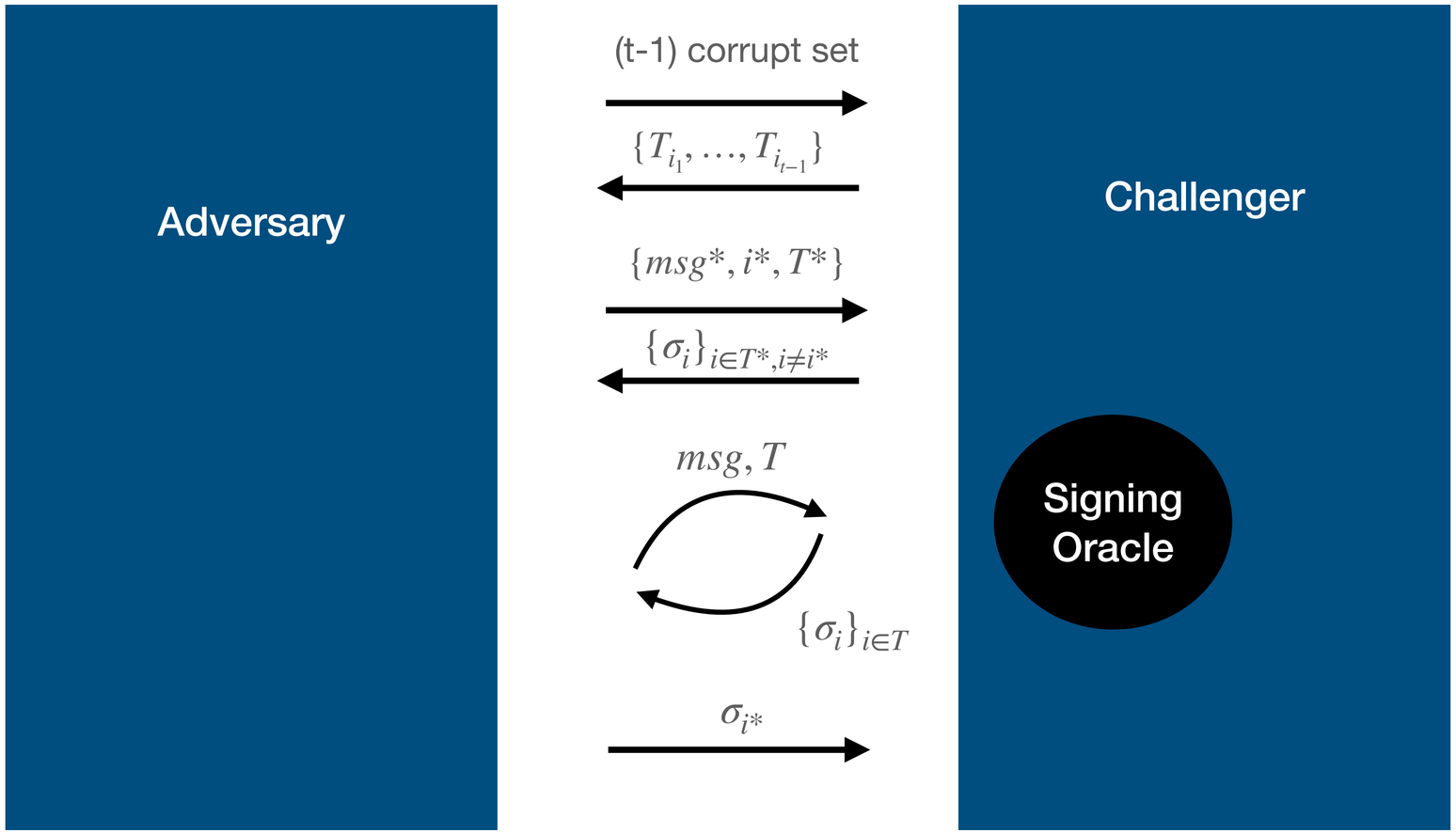
Partial Trapdoor Definitions

New Signature Distributions

Distributions of Lattices

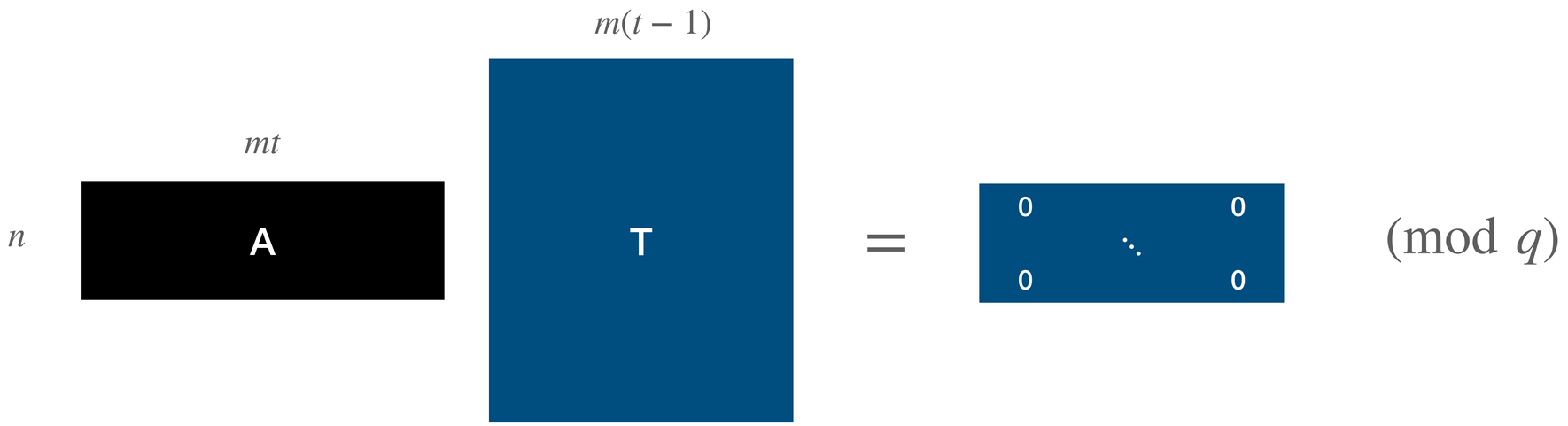


Selective Security



k-SIS Assumption

Given:



Find: short $\mathbf{x} \in \mathcal{R}^{mt}$ such that

- $\mathbf{A} \cdot \mathbf{x} = \mathbf{0} \pmod{q}$
- **$\mathbf{x}$ not in field span of T.**

Reduces to $(n \times m)$ MSIS.

GPV08

The key pair: $(pk, sk) = (\mathbf{A}, \mathbf{T}_A)$

To sign msg**:**

- **Compute** $\mathbf{y} := H(msg)$
- **Using** $\mathbf{T}_A$ **sample small** $\mathbf{x} : \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \pmod{q}$
- **Return** $\mathbf{x}$**.**

To verify: Check $\mathbf{A} \cdot \mathbf{x} = H(msg) \pmod{q}$

GPV08

Signature distribution:

$$\mathbf{x} \leftarrow \mathscr{D}_{\Lambda_q^{\mathbf{y}}(\mathbf{A}),\sigma}$$

The key pair: $(pk, sk) = (\mathbf{A}, \mathbf{T}_A)$

To sign msg**:**

- **Compute** $\mathbf{y} := H(msg)$
- **Using** $\mathbf{T}_A$ **sample small** $\mathbf{x} : \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \pmod{q}$
- **Return** $\mathbf{x}$**.**

To verify: Check $\mathbf{A} \cdot \mathbf{x} = H(msg) \pmod{q}$

Simulating oracles

On hash query msg :

- **Sample random small element** $\mathbf{x} \leftarrow \mathcal{D}_{\mathcal{R}^m,\sigma}$
- **Compute** $\mathbf{y} := \mathbf{A} \cdot \mathbf{x} \pmod{q}$
- **Program RO** $H(msg) := \mathbf{y}$

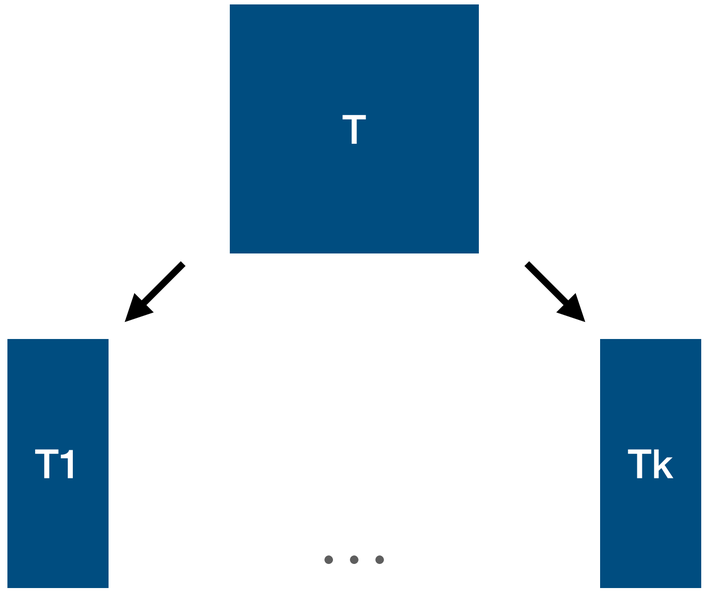
On signature query msg :

- **Return** $\mathbf{x}$

Partial Trapdoor Distributions

$$\mathbf{T}_i \leftarrow PTrapGen(\mathbf{A}, \mathbf{T}_A, \sigma)$$

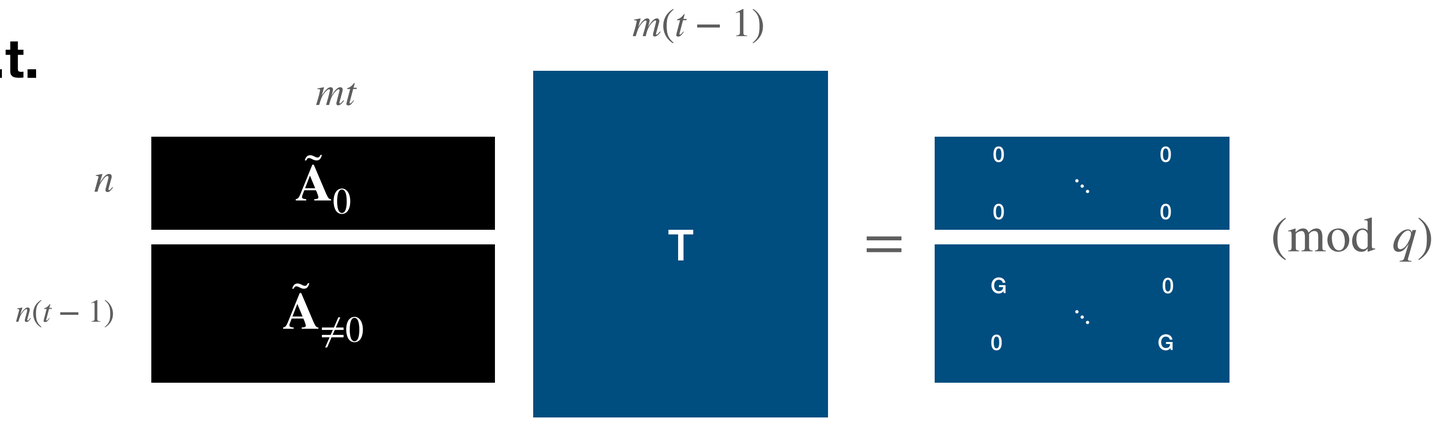
$$\mathbf{x}_i \leftarrow \mathscr{D}_{\Lambda(\mathbf{T}_i) \cap \Lambda_q^{\mathbf{v}_i \otimes \mathbf{y}}(\mathbf{A}), \sigma'}$$



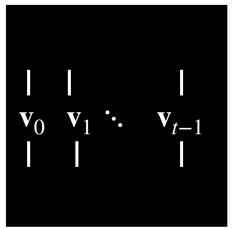
The Corrupt Trapdoors

Given a k-SIS instance: $\tilde{\mathbf{A}}_0 \cdot \mathbf{T} = \mathbf{0} \pmod{q}$

Compute $\tilde{\mathbf{A}}_{\neq 0}$ s.t.

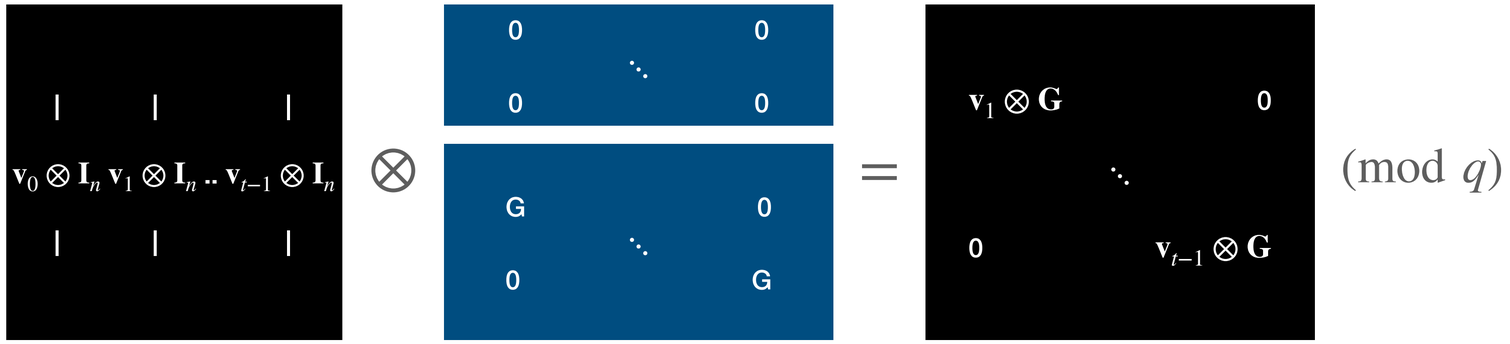


The Corrupt Trapdoors

Then set $\mathbf{A} = (\mathbf{V}' \otimes \mathbf{I}_n) \cdot \tilde{\mathbf{A}}$ **for** $\mathbf{V}' =$ 

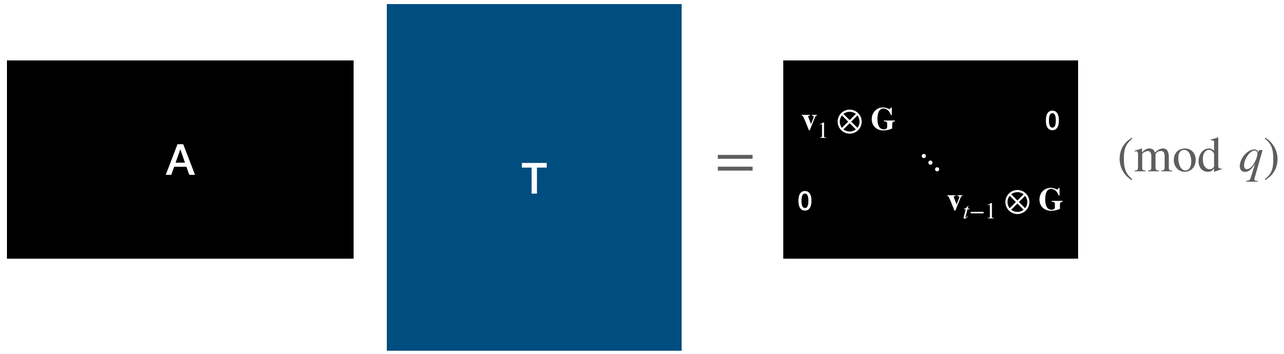
a $(t \times t)$ **matrix.**

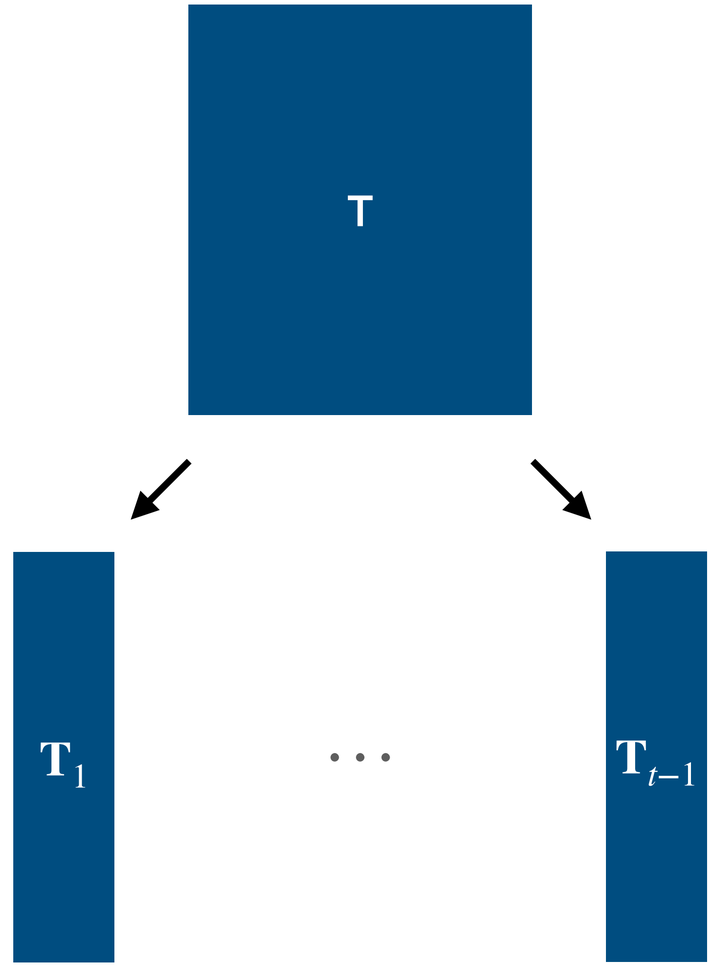
Since



$(\mathrm{mod}\ q)$

The Corrupt Trapdoors





And $\mathbf{A} \cdot \mathbf{T}_i = \mathbf{v}_i \otimes \mathbf{G} \pmod{q}$

Signing Oracle

Idea: Use $\mathbf{T}$ again to get preimages of other $\mathbf{v}_i \otimes \mathbf{y}$.

$$\mathbf{B}_i = \left[\begin{array}{c|c} v_{i,0}^{-1}\mathbf{A}_0 & -\mathbf{G} \\ v_{i,1}^{-1}\mathbf{A}_1 & -\mathbf{G} \\ \vdots & \vdots \\ v_{i,t-1}^{-1}\mathbf{A}_{t-1} & -\mathbf{G} \end{array}\right]$$

Signing Oracle

Idea: Use $\mathbf{T}$ again to get preimages of other $\mathbf{v}_i \otimes \mathbf{y}$.

$$\mathbf{B}_i = \left[\begin{array}{c|c} v_{i,0}^{-1}\mathbf{A}_0 & -\mathbf{G} \\ v_{i,1}^{-1}\mathbf{A}_1 & -\mathbf{G} \\ \vdots & \vdots \\ v_{i,t-1}^{-1}\mathbf{A}_{t-1} & -\mathbf{G} \end{array}\right]$$

Remember:

$$\left(\begin{array}{cc} v_{i,0}^{-1} & 0 \\ & \ddots \\ 0 & v_{i,t-1}^{-1} \end{array} \otimes \mathbf{I}_n\right) \cdot (\mathbf{V}' \otimes \mathbf{I}_n) \cdot \tilde{\mathbf{A}}$$

Signing Oracle

Idea: Use $\mathbf{T}$ again to get preimages of other $\mathbf{v}_i \otimes \mathbf{y}$.

$$\mathbf{B}_i = \left[\begin{array}{c|c} v_{i,0}^{-1}\mathbf{A}_0 & -\mathbf{G} \\ v_{i,1}^{-1}\mathbf{A}_1 & -\mathbf{G} \\ \vdots & \vdots \\ v_{i,t-1}^{-1}\mathbf{A}_{t-1} & -\mathbf{G} \end{array}\right]$$

Remember:

$$\left(\begin{array}{cc} v_{i,0}^{-1} & 0 \\ & \ddots \\ 0 & v_{i,t-1}^{-1} \end{array} \otimes \mathbf{I}_n\right) \cdot (\mathbf{V}' \otimes \mathbf{I}_n) \cdot \tilde{\mathbf{A}} := \mathbf{H}_i \cdot \tilde{\mathbf{A}}$$

Signing Oracle

Idea: Use $\mathbf{T}$ again to get preimages of other $\mathbf{v}_i \otimes \mathbf{y}$.

$$\mathbf{B}_i = \left[\begin{array}{c|c} v_{i,0}^{-1}\mathbf{A}_0 & -\mathbf{G} \\ v_{i,1}^{-1}\mathbf{A}_1 & -\mathbf{G} \\ \vdots & \vdots \\ v_{i,t-1}^{-1}\mathbf{A}_{t-1} & -\mathbf{G} \end{array}\right]$$

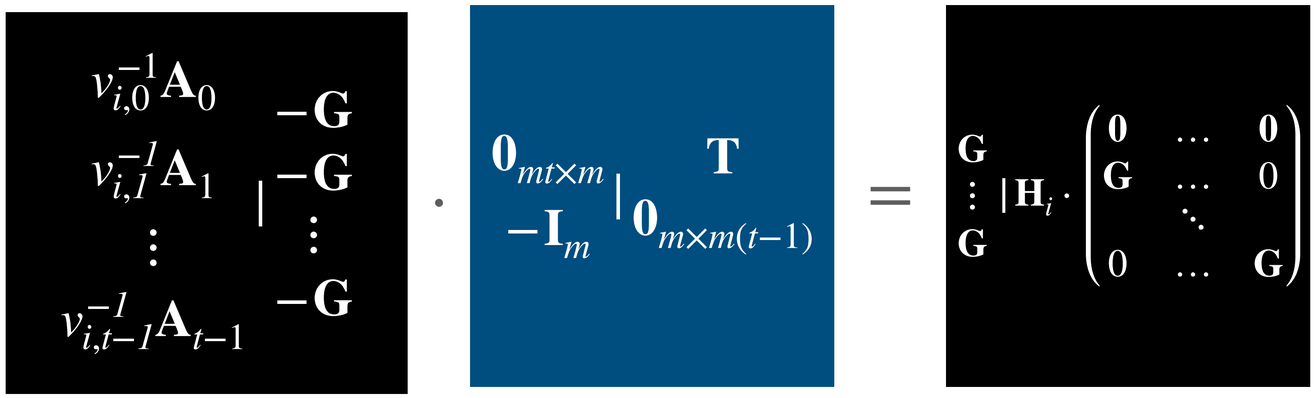
Remember:

$$\left(\begin{array}{cc} v_{i,0}^{-1} & 0 \\ & \ddots \\ 0 & v_{i,t-1}^{-1} \end{array} \otimes \mathbf{I}_n\right) \cdot (\mathbf{V}' \otimes \mathbf{I}_n) \cdot \tilde{\mathbf{A}} := \mathbf{H}_i \cdot \tilde{\mathbf{A}}$$

and $\tilde{\mathbf{A}} \cdot \mathbf{T} = \begin{pmatrix} \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{G} & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & \mathbf{G} \end{pmatrix} \pmod{q}$

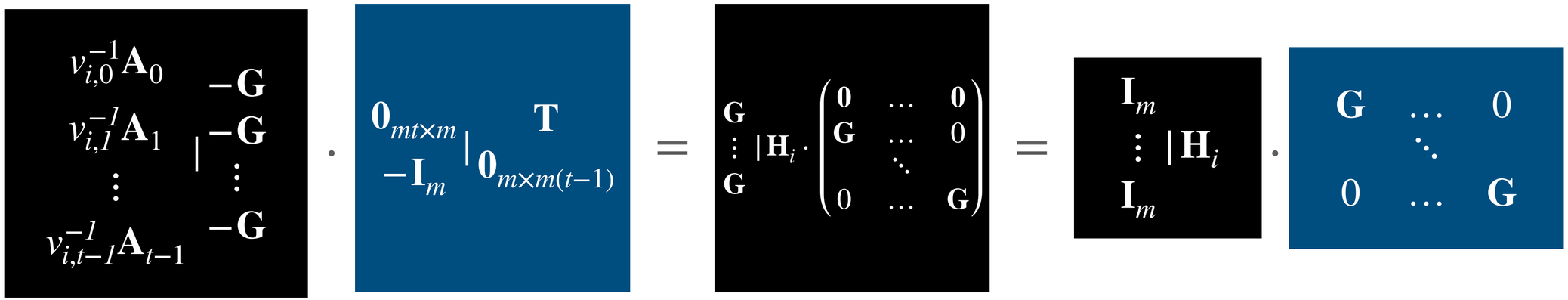
Signing Oracle

Then



Signing Oracle

Then



Signing Oracle

Sample $\mathbf{U}_i, \mathbf{W}_i$ such that

$$\begin{bmatrix} v_{i,0}^{-1}\mathbf{A}_0 & -\mathbf{G} \\ v_{i,1}^{-1}\mathbf{A}_1 & -\mathbf{G} \\ \vdots & \vdots \\ v_{i,t-1}^{-1}\mathbf{A}_{t-1} & -\mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{U}_i \\ \mathbf{W}_i \end{bmatrix} = \mathbf{0} \pmod{q}$$

Then for each $j = 0, \ldots, t-1$ we have $\mathbf{A}_j \cdot \mathbf{U}_i = v_{i,j} \cdot \mathbf{G} \cdot \mathbf{W}_i \pmod{q}$

Signing Oracle

Sample $\mathbf{U}_i, \mathbf{W}_i$ such that

$$\begin{bmatrix} v_{i,0}^{-1}\mathbf{A}_0 & -\mathbf{G} \\ v_{i,1}^{-1}\mathbf{A}_1 & -\mathbf{G} \\ \vdots & \vdots \\ v_{i,t-1}^{-1}\mathbf{A}_{t-1} & -\mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{U}_i \\ \mathbf{W}_i \end{bmatrix} = \mathbf{0} \pmod{q}$$

Then for each $j = 0, \ldots, t-1$ we have $\mathbf{A}_j \cdot \mathbf{U}_i = v_{i,j} \cdot \mathbf{G} \cdot \mathbf{W}_i \pmod{q}$

So $\mathbf{A} \cdot \mathbf{U}_i = \mathbf{v}_i \otimes \mathbf{G} \cdot \mathbf{W}_i \pmod{q}$

Signing Oracle

Sample $\mathbf{U}_i, \mathbf{W}_i$ such that

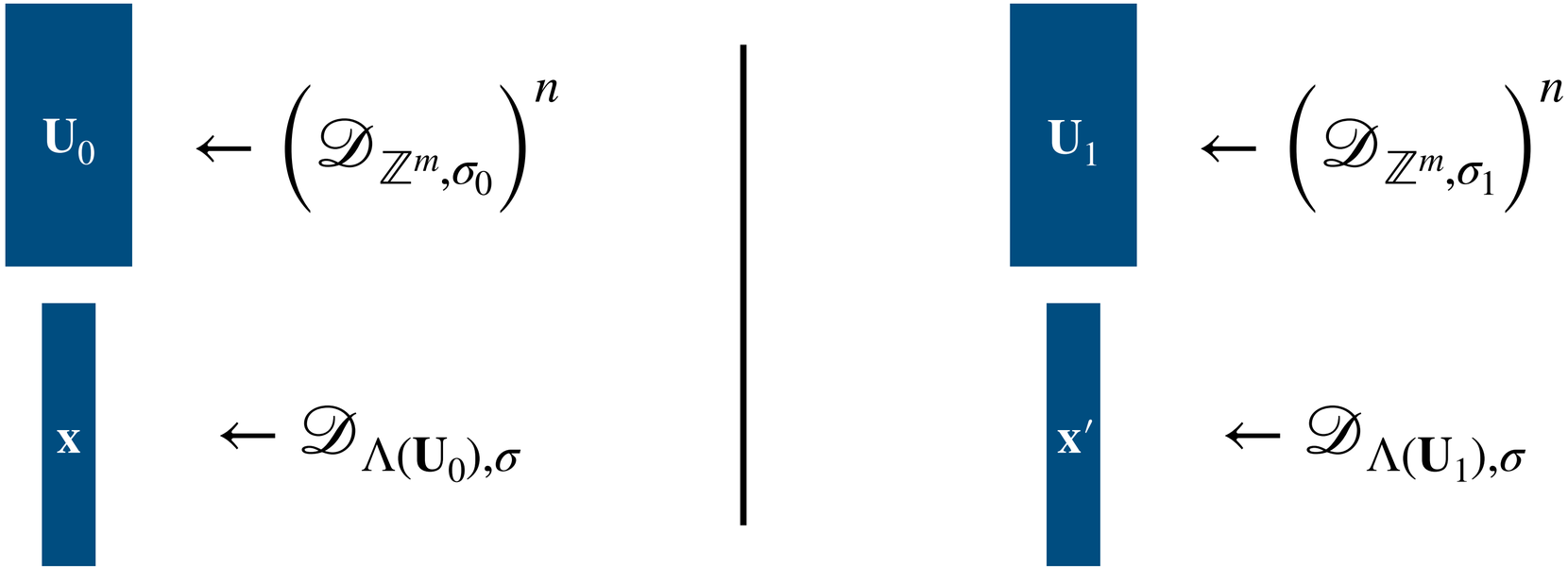
$$\begin{bmatrix} v_{i,0}^{-1}\mathbf{A}_0 & -\mathbf{G} \\ v_{i,1}^{-1}\mathbf{A}_1 & -\mathbf{G} \\ \vdots & \vdots \\ v_{i,t-1}^{-1}\mathbf{A}_{t-1} & -\mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{U}_i \\ \mathbf{W}_i \end{bmatrix} = \mathbf{0} \pmod{q}$$

Then for each $j = 0, \ldots, t-1$ we have $\mathbf{A}_j \cdot \mathbf{U}_i = v_{i,j} \cdot \mathbf{G} \cdot \mathbf{W}_i \pmod{q}$

So $\mathbf{A} \cdot \mathbf{U}_i = \mathbf{v}_i \otimes \mathbf{G} \cdot \mathbf{W}_i = \mathbf{v}_i \otimes \mathbf{R}_i \pmod{q}$

The Distribution Conjecture

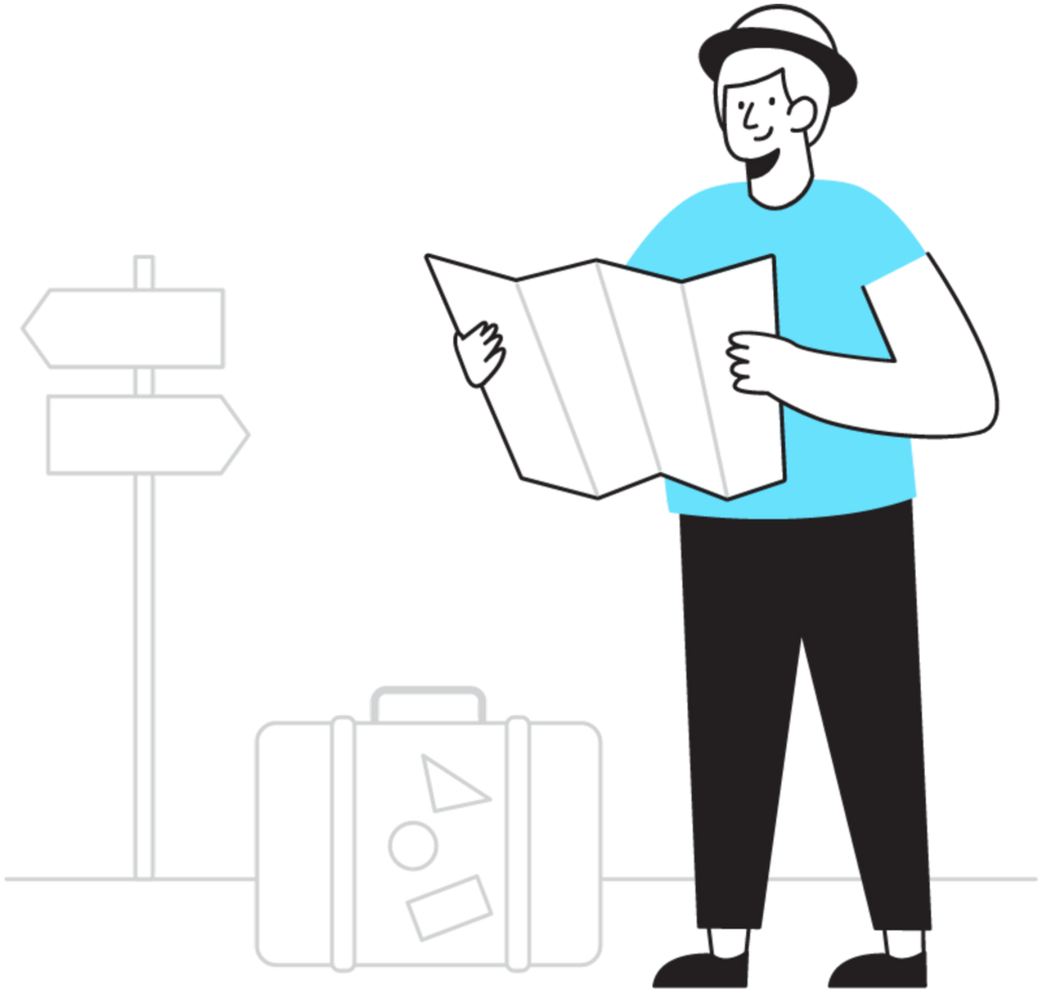
Let $m > n$, and $\sigma/m > \sigma_1 > \sigma_0$.



Then $\mathbf{x} \approx \mathbf{x}'$.

Open directions

Conjecture
Stronger Security
Set of Collaborators
Anonymity



Other projects

Image by rawpixel.com on Freepik

Thank you!