THALES

LIP6

Breaking HuFu with 0 Leakage

Julien Devevey, Morgane Guerreau, **Thomas Legavre**,
Ange Martinelli, Thomas Ricosset

WRACH

Agence Nationale de la Sécurité des Systèmes d'information, Paris
Sorbonne Université, CNRS, LIP6, Paris
Thales, Gennevilliers

April 22, 2025

Introduction

What is HuFu?

- Signature scheme based on unstructured lattices
- Based on the Hash-and-Sign paradigm [GPV08] (like Falcon)
- Round 1 candidate to NIST additional post-quantum signature competition

Introduction

What is HuFu?

- Signature scheme based on unstructured lattices
- Based on the Hash-and-Sign paradigm [GPV08] (like Falcon)
- Round 1 candidate to NIST additional post-quantum signature competition

Why attack it?

- Absence of structure counters previous SCA done on Falcon
- Trapdoor sampling a la [MP12] is used in other contexts (IBEs...)

ISIS and LWE

(Forgery) ISIS_B

For t and B, find $\mathbf{z}$ with $\|\mathbf{z}\| < B$ such that

$$\mathbf{A} \cdot \mathbf{z} = \mathbf{t} \mod Q$$

(Key recovery) LWE with short secret

For b and B, find s, e with $\|(\mathbf{e}, \mathbf{s})\| < B$ such that

$$\mathbf{A} \cdot \mathbf{s} + \mathbf{e} = (\mathbf{I} \mid \mathbf{A}) \cdot (\mathbf{e}, \mathbf{s}) = \mathbf{b} \mod Q$$

Primal Attack on LWE

Problem: We want to find small $\mathbf{s}$ and $\mathbf{e}$ such that $\mathbf{A} \cdot \mathbf{s} + \mathbf{e} = \mathbf{b} \mod Q$

Primal Attack on LWE

Problem: We want to find small $\mathbf{s}$ and $\mathbf{e}$ such that $\mathbf{A} \cdot \mathbf{s} + \mathbf{e} = \mathbf{b} \mod Q$

Lattice construction:

- Consider the lattice

$$\Lambda = \{\mathbf{v} \in \mathbb{Z}^{n+m+1} \mid (\mathbf{A} \mid \mathbf{I}_m \mid \mathbf{b}) \cdot \mathbf{v} \equiv 0 \pmod{q}\}$$

- It contains an unusually short vector $(\mathbf{s} \mid \mathbf{e} \mid -1)$ since

$$(\mathbf{A} \mid \mathbf{I}_m \mid \mathbf{b}) \cdot (\mathbf{s} \mid \mathbf{e} \mid -1) = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} - \mathbf{b} \equiv 0 \pmod{q}$$

Primal Attack on LWE

Problem: We want to find small $\mathbf{s}$ and $\mathbf{e}$ such that $\mathbf{A} \cdot \mathbf{s} + \mathbf{e} = \mathbf{b} \mod Q$

Lattice construction:

- Consider the lattice

$$\Lambda = \{\mathbf{v} \in \mathbb{Z}^{n+m+1} \mid (\mathbf{A} \mid \mathbf{I}_m \mid \mathbf{b}) \cdot \mathbf{v} \equiv 0 \pmod{q}\}$$

- It contains an unusually short vector $(\mathbf{s} \mid \mathbf{e} \mid -1)$ since

$$(\mathbf{A} \mid \mathbf{I}_m \mid \mathbf{b}) \cdot (\mathbf{s} \mid \mathbf{e} \mid -1) = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} - \mathbf{b} \equiv 0 \pmod{q}$$

Hardness: For $d > \beta > 50$, BKZ finds a vector $\mathbf{v} \in \Lambda$ such that:

$$\|\mathbf{v}\| \leq \delta_\beta^d \cdot \mathrm{Vol}(\Lambda)^{1/d} \qquad \text{and} \qquad \delta_\beta \approx \left(\frac{(\pi\beta)^{1/\beta}}{2\pi e}\right)^{\frac{1}{2(\beta-1)}}$$

Hash-and-sign for Lattices and HuFu

Generic framework for lattice-based signatures [GPV08] such as Falcon. Instanciated as follows for HuFu:

- Verification key: a matrix $\mathbf{A} = (\mathbf{I}_m | \hat{\mathbf{A}} | \mathbf{B})$ with $\mathbf{B} = p\mathbf{I}_m - \hat{\mathbf{A}}\mathbf{S} - \mathbf{E} \bmod pq$,
- Signing key: $\mathbf{sk}^\top = q(\mathbf{I}_m | \mathbf{S} | \mathbf{E})$, a short basis of $\Lambda = \{\mathbf{Ax} = 0 \bmod pq, \mathbf{x} \in \mathbb{Z}^k\}$,
- Given a message μ, sign by giving a short preimage $\mathbf{x}$ of $\mathbf{u} = H(\mu)$ by $\mathbf{A}$,
- How is $\mathbf{x}$ sampled?

Gadget

Goal: compute short $\mathbf{x}$ such $\mathbf{A} \cdot \mathbf{x} = \mathcal{H}(\mathsf{m}) \mod Q$

Gadget

Goal: compute short $\mathbf{x}$ such $\mathbf{A} \cdot \mathbf{x} = \mathcal{H}(\mathrm{m}) \mod Q$

Gadget from [MP12]

Family of $\mathbf{A}$, $\mathbf{T}$ and $\mathbf{G}$ such that:

$$\mathbf{AT} = \mathbf{G} \mod Q$$

Public Key: $\mathbf{A}$ Private Key: $\mathbf{T}$ Gadget: $\mathbf{G}$

Compute $\mathbf{z}$ so that $\mathbf{G} \cdot \mathbf{z} = \mathcal{H}(\mathrm{m}) \mod Q$, and return $\mathbf{x} = \mathbf{T}\mathbf{z}$ as preimage of $\mathcal{H}(\mathrm{m})$

Gadget

Goal: compute short $\mathbf{x}$ such $\mathbf{A} \cdot \mathbf{x} = \mathcal{H}(\mathrm{m}) \mod Q$

Gadget from [MP12]

Family of $\mathbf{A}$, $\mathbf{T}$ and $\mathbf{G}$ such that:

$$\mathbf{AT} = \mathbf{G} \mod Q$$

Public Key: $\mathbf{A}$ Private Key: $\mathbf{T}$ Gadget: $\mathbf{G}$

Compute $\mathbf{z}$ so that $\mathbf{G} \cdot \mathbf{z} = \mathcal{H}(\mathrm{m}) \mod Q$, and return $\mathbf{x} = \mathbf{T}\mathbf{z}$ as preimage of $\mathcal{H}(\mathrm{m})$

× Collecting many preimages will leak $\mathbf{T}$...

✓ Add mask $\mathbf{p}$: preimages $\mathbf{x} = \mathbf{p} + \mathbf{T}\mathbf{z}$
and the target become $\mathbf{u} = \mathcal{H}(\mathrm{m}) - \mathbf{A}\mathbf{p}$ instead of $\mathcal{H}(\mathrm{m})$

HuFu Gadget

Compact gadget:

$$p\mathbf{I},\ q\mathbf{I} \in \mathbb{Z}^{n\times n} \quad \text{such that} \quad p\mathbf{I} \cdot q\mathbf{I} = Q \cdot \mathbf{I} \quad \text{with } Q = p \cdot q.$$

Trapdoor:

$$\mathbf{AT} = p\mathbf{I} \mod Q$$

LWE-based construction:

$$\mathbf{A} = \left[\mathbf{I} \,\middle|\, \hat{\mathbf{A}} \,\middle|\, p\mathbf{I} - \hat{\mathbf{A}}\mathbf{S} - \mathbf{E}\right], \quad \mathbf{T} = \begin{pmatrix} \mathbf{E} \\ \mathbf{S} \\ \mathbf{I} \end{pmatrix}$$

HuFu Gadget

Compact gadget:

$$p\mathbf{I},\ q\mathbf{I} \in \mathbb{Z}^{n\times n} \quad \text{such that} \quad p\mathbf{I} \cdot q\mathbf{I} = Q \cdot \mathbf{I} \quad \text{with } Q = p \cdot q.$$

Trapdoor:

$$\mathbf{AT} = p\mathbf{I} \mod Q$$

LWE-based construction:

$$\mathbf{A} = \left[\mathbf{I} \,\middle|\, \hat{\mathbf{A}} \,\middle|\, p\mathbf{I} - \hat{\mathbf{A}}\mathbf{S} - \mathbf{E}\right], \quad \mathbf{T} = \begin{pmatrix} \mathbf{E} \\ \mathbf{S} \\ \mathbf{I} \end{pmatrix}$$

Objective: Invert $f_{p\mathbf{I}} : \mathbf{x} \mapsto p\mathbf{I}\mathbf{x} \mod Q$, i.e.,

$$p\mathbf{I}\mathbf{x} = \mathbf{u} - e \mod Q$$

HuFu Gadget

Compact gadget:

$$p\mathbf{I},\ q\mathbf{I} \in \mathbb{Z}^{n\times n} \quad \text{such that} \quad p\mathbf{I} \cdot q\mathbf{I} = Q \cdot \mathbf{I} \quad \text{with } Q = p \cdot q.$$

Trapdoor:

$$\mathbf{AT} = p\mathbf{I} \mod Q$$

LWE-based construction:

$$\mathbf{A} = \left[\mathbf{I} \,\middle|\, \hat{\mathbf{A}} \,\middle|\, p\mathbf{I} - \hat{\mathbf{A}}\mathbf{S} - \mathbf{E}\right], \quad \mathbf{T} = \begin{pmatrix} \mathbf{E} \\ \mathbf{S} \\ \mathbf{I} \end{pmatrix}$$

Objective: Invert $f_{p\mathbf{I}} : \mathbf{x} \mapsto p\mathbf{I}\mathbf{x} \mod Q$, i.e.,

$$p\mathbf{I}\mathbf{x} = \mathbf{u} - e \mod Q$$

Deterministic error decoding: Compute e such that $\mathbf{u} - e = p\mathbf{I}\mathbf{v} \in p \cdot \mathbb{Z}^n$.

HuFu Gadget

Compact gadget:

$$p\mathbf{I},\ q\mathbf{I} \in \mathbb{Z}^{n\times n} \quad \text{such that} \quad p\mathbf{I} \cdot q\mathbf{I} = Q \cdot \mathbf{I} \quad \text{with } Q = p \cdot q.$$

Trapdoor:

$$\mathbf{AT} = p\mathbf{I} \mod Q$$

LWE-based construction:

$$\mathbf{A} = \left[\mathbf{I} \,\middle|\, \hat{\mathbf{A}} \,\middle|\, p\mathbf{I} - \hat{\mathbf{A}}\mathbf{S} - \mathbf{E}\right], \quad \mathbf{T} = \begin{pmatrix} \mathbf{E} \\ \mathbf{S} \\ \mathbf{I} \end{pmatrix}$$

Objective: Invert $f_{p\mathbf{I}} : \mathbf{x} \mapsto p\mathbf{I}\mathbf{x} \mod Q$, i.e.,

$$p\mathbf{I}\mathbf{x} = \mathbf{u} - e \mod Q$$

Deterministic error decoding: Compute e such that $\mathbf{u} - e = p\mathbf{I}\mathbf{v} \in p \cdot \mathbb{Z}^n$.
Random preimage sampling: Sample short $\mathbf{z} \in q \cdot \mathbb{Z}^n + \mathbf{v}$ using Gaussian sampling.

HuFu Gadget

Compact gadget:

$$p\mathbf{I},\ q\mathbf{I} \in \mathbb{Z}^{n\times n} \quad \text{such that} \quad p\mathbf{I} \cdot q\mathbf{I} = Q \cdot \mathbf{I} \quad \text{with } Q = p \cdot q.$$

Trapdoor:

$$\mathbf{AT} = p\mathbf{I} \mod Q$$

LWE-based construction:

$$\mathbf{A} = \left[\mathbf{I} \,\middle|\, \hat{\mathbf{A}} \,\middle|\, p\mathbf{I} - \hat{\mathbf{A}}\mathbf{S} - \mathbf{E}\right], \quad \mathbf{T} = \begin{pmatrix} \mathbf{E} \\ \mathbf{S} \\ \mathbf{I} \end{pmatrix}$$

Objective: Invert $f_{p\mathbf{I}} : \mathbf{x} \mapsto p\mathbf{I}\mathbf{x} \mod Q$, i.e.,

$$p\mathbf{I}\mathbf{x} = \mathbf{u} - e \mod Q$$

Deterministic error decoding: Compute e such that $\mathbf{u} - e = p\mathbf{I}\mathbf{v} \in p \cdot \mathbb{Z}^n$.
Random preimage sampling: Sample short $\mathbf{z} \in q \cdot \mathbb{Z}^n + \mathbf{v}$ using Gaussian sampling.
Correctness:

$$p\mathbf{I}\mathbf{z} = p\mathbf{I}(q\mathbf{I}\mathbf{y} + \mathbf{v}) = Q\mathbf{y} + \mathbf{u} - e = \mathbf{u} - e \mod Q$$

HuFu Algorithms

Key Generation

1. Secret key: $\mathbf{T} = \begin{pmatrix} \mathbf{E} \\ \mathbf{S} \\ \mathbf{I} \end{pmatrix}$
2. Public key: $\mathbf{A} = [\mathbf{I}, \widehat{\mathbf{A}}, \mathbf{B}]$ and $\mathbf{B} = \mathbf{P} - (\widehat{\mathbf{A}}\mathbf{S} + \mathbf{E})$
   - $\mathbf{A} \cdot \mathbf{T} = \mathbf{P}$

Verification

1. $\mathbf{x_0'} = \mathcal{H}(\mathrm{m}) - \widehat{\mathbf{A}}\mathbf{x_1} - \mathbf{B}\mathbf{x_2}$
2. Accept if $\|\mathbf{x}_0', \mathbf{x}_1, \mathbf{x}_2\| < B$

Sign

1. Sample $\mathbf{p}$ from a short Gaussian $\mathcal{D}_{\mathbf{T}}$.
2. $\mathbf{u} = \mathcal{H}(\mathrm{m}) - \mathbf{A}\mathbf{p} \mod Q$
3. $\mathbf{v} = \lfloor \mathbf{u}/p \rceil \mod Q$
4. Sample $\mathbf{z} \hookleftarrow D_{q \cdot \mathbb{Z}^k + \mathbf{v}, \bar{r}^2}$.
5. $(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2) = \begin{pmatrix} \mathbf{E} \\ \mathbf{S} \\ \mathbf{I} \end{pmatrix} \mathbf{z} + \mathbf{p} \mod Q$
6. if $\|\mathbf{x}_0 + e, \mathbf{x}_1, \mathbf{x}_2\| < B$
7. return $(\mathbf{x}_1, \mathbf{x}_2)$

HuFu Algorithms

Key Generation

1. Secret key: $\mathbf{T} = \begin{pmatrix} \mathbf{E} \\ \mathbf{S} \\ \mathbf{I} \end{pmatrix}$
2. Public key: $\mathbf{A} = [\mathbf{I}, \widehat{\mathbf{A}}, \mathbf{B}]$ and $\mathbf{B} = \mathbf{P} - (\widehat{\mathbf{A}}\mathbf{S} + \mathbf{E})$
   - $\mathbf{A} \cdot \mathbf{T} = \mathbf{P}$

Verification

1. $\mathbf{x_0'} = \mathcal{H}(\mathrm{m}) - \widehat{\mathbf{A}}\mathbf{x_1} - \mathbf{B}\mathbf{x_2}$
2. Accept if $\|\mathbf{x}_0', \mathbf{x}_1, \mathbf{x}_2\| < B$

Sign

1. Sample $\mathbf{p}$ from a short Gaussian $\mathcal{D}_{\mathbf{T}}$.
2. $\mathbf{u} = \mathcal{H}(\mathrm{m}) - \mathbf{A}\mathbf{p} \mod Q$
3. $\mathbf{v} = \lfloor \mathbf{u}/p \rceil \mod Q$
4. Sample $\mathbf{z} \hookleftarrow D_{q \cdot \mathbb{Z}^k + \mathbf{v}, \bar{r}^2}$.
5. $(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2) = \begin{pmatrix} \mathbf{E} \\ \mathbf{S} \\ \mathbf{I} \end{pmatrix} \mathbf{z} + \mathbf{p} \mod Q$
6. if $\|\mathbf{x}_0 + e, \mathbf{x}_1, \mathbf{x}_2\| < B$
7. return $(\mathbf{x}_1, \mathbf{x}_2)$

Leakage in matrix-vector multiplication

Targeted operations: $\mathbf{S}_{i,j} \cdot \mathbf{z}_i$ (resp. $\mathbf{E}_{i,j} \cdot \mathbf{z}_i$)

Leakage in matrix-vector multiplication

Targeted operations: $\mathbf{S}_{i,j} \cdot \mathbf{z}_i$ (resp. $\mathbf{E}_{i,j} \cdot \mathbf{z}_i$)

Coefficients of $\mathbf{S}$ (resp. $\mathbf{E}$) are ternary and follow a binomial distribution.
$\rightarrow$ only three possible outputs for $\mathbf{S}_{i,j} \cdot \mathbf{z}_i$:

Leakage in matrix-vector multiplication

Targeted operations: $\mathbf{S}_{i,j} \cdot \mathbf{z}_i$ (resp. $\mathbf{E}_{i,j} \cdot \mathbf{z}_i$)

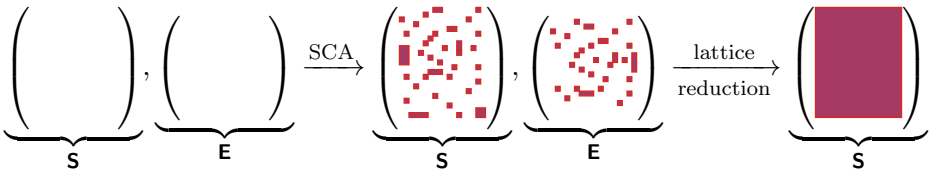
Coefficients of $\mathbf{S}$ (resp. $\mathbf{E}$) are ternary and follow a binomial distribution.
$\rightarrow$ only three possible outputs for $\mathbf{S}_{i,j} \cdot \mathbf{z}_i$:

1. 0, setting the Hamming weight to 0.
2. $\mathbf{z}_i$, keeping the Hamming weight identical.
3. $-\mathbf{z}_i$, greatly changing the Hamming weight.

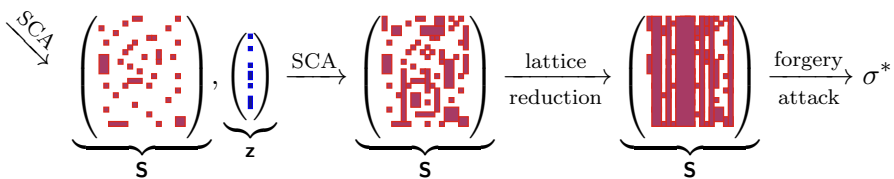
Overview

Two Attack scenarios:

Key Recovery with 0's knowledge



Forgery with more knowledge



How to use partial information

Given an LWE sample $\mathbf{As} + \mathbf{e}$ and some $\mathbf{0}$s of $\mathbf{s}$ and $\mathbf{e}$, how do we exploit them?

How to use partial information

Given an LWE sample $\mathbf{As} + \mathbf{e}$ and some $\mathbf{0}$s of $\mathbf{s}$ and $\mathbf{e}$, how do we exploit them?

- Remove the i-th column of $\mathbf{A}$ if $s_i = 0$: dimension reduced by one.

How to use partial information

Given an LWE sample $\mathbf{As} + \mathbf{e}$ and some $\mathbf{0}$s of $\mathbf{s}$ and $\mathbf{e}$, how do we exploit them?

- Remove the i-th column of $\mathbf{A}$ if $s_i = 0$: dimension reduced by one.
- Write $b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle$ if $e_i = 0$. Dimension reduced by one. Some rewriting involved to find a new LWE instance with one less dimension.

What is the cost of BKZ on the new LWE instance once every hint has been incorporated?

Key recovery with 0 knowledge

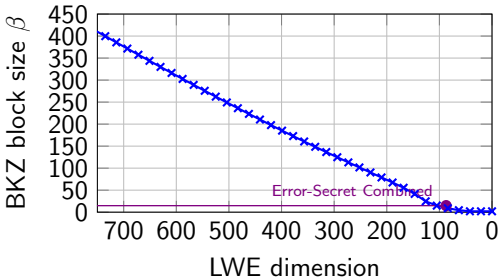
Goal: Find all zeros of **S** and **E**

Key recovery with 0 knowledge

Goal: Find all zeros of **S** and **E**
Due to the distribution : half of their coordinates are 0.

Key recovery with 0 knowledge

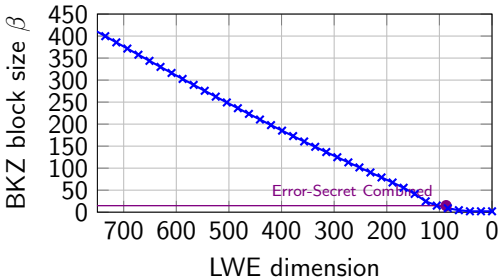
Goal: Find all zeros of **S** and **E**
Due to the distribution : half of their coordinates are 0.



# Traces	Recovered	False Positives
200	93.4%	0.12%
600	97.9%	0%
1500	98.5%	0%

Key recovery with 0 knowledge

Goal: Find all zeros of **S** and **E**
Due to the distribution : half of their coordinates are 0.



# Traces	Recovered	False Positives
200	93.4%	0.12%
600	97.9%	0%
1500	98.5%	0%

Countermeasure

$$\mathbf{x}_0 = \mathbf{E}\mathbf{z} + \mathbf{p} \mod Q$$

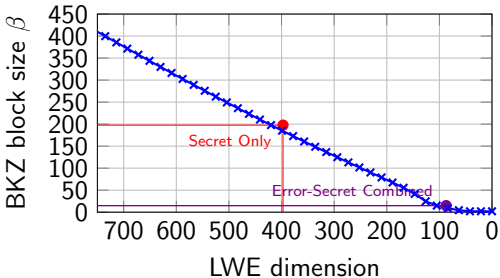
can be replaced by

$$\mathbf{x}_0' = \mathcal{H}(\mathrm{m}) - \widehat{\mathbf{A}}\mathbf{x}_1 - \mathbf{B}\mathbf{x}_2 \mod Q$$

which involves only public values.

Key recovery with 0 knowledge

Goal: Find all zeros of **S** and **E**
Due to the distribution : half of their coordinates are 0.



# Traces	Recovered	False Positives
200	93.4%	0.12%
600	97.9%	0%
1500	98.5%	0%

Countermeasure

$$\mathbf{x}_0 = \mathbf{E}\mathbf{z} + \mathbf{p} \mod Q$$

can be replaced by

$$\mathbf{x}'_0 = \mathcal{H}(\mathrm{m}) - \widehat{\mathbf{A}}\mathbf{x}_1 - \mathbf{B}\mathbf{x}_2 \mod Q$$

which involves only public values.

Forgery with more knowledge

New SCA

We attack the Gaussian sampler to recover information of the sign of $\mathbf{z}_i$.
But we can only do it for half of the values of $\mathbf{z_i}$.
$\rightarrow$ only 75% of $\mathbf{S}$ can be recovered (without false positives)

Forgery with more knowledge

New SCA

We attack the Gaussian sampler to recover information of the sign of $\mathbf{z}_i$.
But we can only do it for half of the values of $\mathbf{z_i}$.
$\rightarrow$ only 75% of $\mathbf{S}$ can be recovered (without false positives)

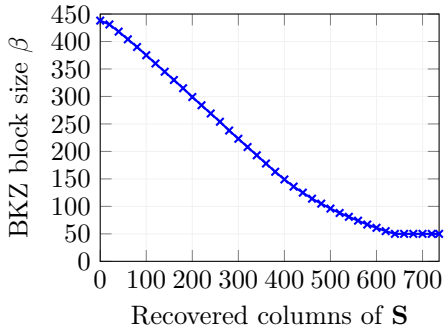
Key recovery can be compromised..

Forgery with more knowledge

New SCA

We attack the Gaussian sampler to recover information of the sign of $\mathbf{z}_i$.
But we can only do it for half of the values of $\mathbf{z_i}$.
$\rightarrow$ only 75% of $\mathbf{S}$ can be recovered (without false positives)

Key recovery can be compromised..



But we can forge !

Forging for specific vectors

Assuming the first k columns $\mathbf{S}_k$ of $\mathbf{S}$ are known via the previous attack, what can we do with them?

Forging for specific vectors

Assuming the first k columns $\mathbf{S}_k$ of $\mathbf{S}$ are known via the previous attack, what can we do with them?

- If the target is $\mathbf{u} = \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{0} \end{pmatrix}$, then we set $\mathbf{p} = \mathbf{0}$, $\mathbf{v} = \lfloor \mathbf{u}/p \rceil$ and $\mathbf{z} = \mathbf{v}$. A signature would then be:

$$\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{S} \\ \mathbf{I}_m \end{pmatrix} \cdot \mathbf{z} = \begin{pmatrix} \mathbf{S}_k & \mathbf{0} \\ \mathbf{I}_k & \mathbf{0} \end{pmatrix} \cdot \mathbf{z}.$$

Forging for specific vectors

Assuming the first k columns $\mathbf{S}_k$ of $\mathbf{S}$ are known via the previous attack, what can we do with them?

- If the target is $\mathbf{u} = \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{0} \end{pmatrix}$, then we set $\mathbf{p} = \mathbf{0}$, $\mathbf{v} = \lfloor \mathbf{u}/p \rceil$ and $\mathbf{z} = \mathbf{v}$. A signature would then be:

$$\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{S} \\ \mathbf{I}_m \end{pmatrix} \cdot \mathbf{z} = \begin{pmatrix} \mathbf{S}_k & \mathbf{0} \\ \mathbf{I}_k & \mathbf{0} \end{pmatrix} \cdot \mathbf{z}.$$

This vector is short, but which message did we sign?

Finding specific vectors

- Choose any μ and compute $\mathbf{u} = H(\mu) = \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix}$.
- Write $\mathbf{A} = \begin{pmatrix} \mathbf{A}_h \\ \mathbf{A}_l \end{pmatrix}$
- Find short $\mathbf{x}'$ such that $\mathbf{A}_l\mathbf{x}' = \mathbf{u}_2$ with lattice reduction
- Set $\mathbf{u}' = \mathbf{u} - \mathbf{A}\mathbf{x}' = \begin{pmatrix} \mathbf{u}'_1 \\ \mathbf{0} \end{pmatrix}$
- We are back to the previous case!

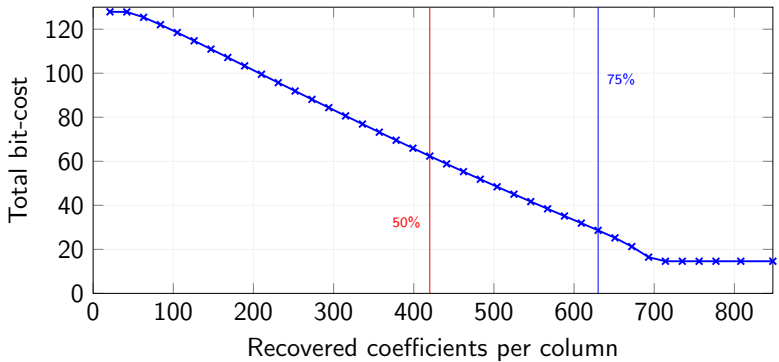
How much costs a forgery?

We start by gathering d coefficients per column.

- First step: complete k columns via lattice reduction: k times LWE with dimension reduced by d
- Second step: one more lattice reduction to find $\mathbf{x}'$: dimension reduced by k.
- Third step: forgery for specific vectors (essentially free)

All that remains is to optimize over k.

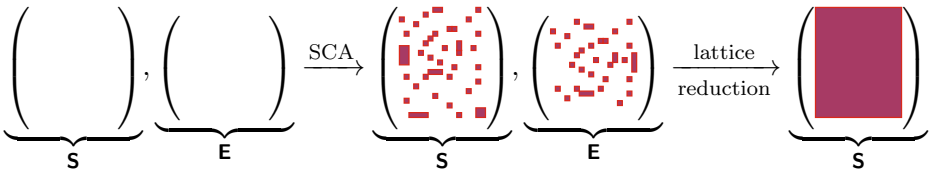
Final Cost


Total bit-cost
120
100
80
60
40
20
0
0
100
200
300
400
500
600
700
800
Recovered coefficients per column
75%
50%

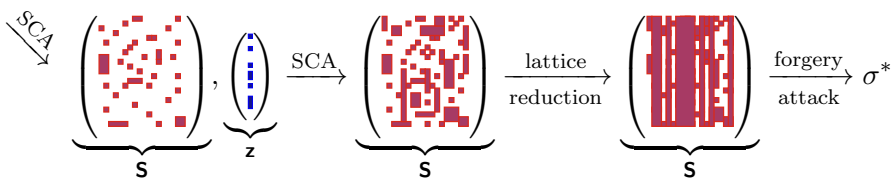
Conclusion

Two Attack scenarios:

Key Recovery with 0's knowledge



Forgery with more knowledge



Future work

Hint-(M)LWE

Recover from an LWE instance $\mathbf{s}$ with additional knowledge: $\mathbf{z}_i = \mathbf{y}_i + \mathbf{c}_i \cdot \mathbf{s}$

Can be used to construct primitive, and estimate residual security with reduction from Hint-MLWE to MLWE.

Future work

Hint-(M)LWE

Recover from an LWE instance $\mathbf{s}$ with additional knowledge: $\mathbf{z}_i = \mathbf{y}_i + \mathbf{c}_i \cdot \mathbf{s}$

Can be used to construct primitive, and estimate residual security with reduction from Hint-MLWE to MLWE.

For exemple : Raccoon, Katana, Plover.
Problem: The problem is well understood only in gaussian settings.
Open Question: Cryptanalysis in other settings / try to have a reduction Hint-MLWE $\rightarrow$ MLWE.