On Removing Rejection Conditions in Practical Lattice-Based Signatures

Rouzbeh Behnia[1], Yilei Chen[2] and Daniel Masny[3]

[1] University of South Florida
[2] Tsinghua University
[3] VISA Research

Digital Signatures

Two main approaches to achieve signature schemes

- **Hash-and-Sign:** Based on trapdoor one-way functions (e.g., RSA)

- **Fiat-Shamir (FS) Transform:** Resulted from applying transformation on identification schemes.

State-of-the-art

Lattice-Based PQC Candidates
Round III

D I L I T H I U M

- Based on the Fiat Shamir with Aborts paradigm
- Faster signing
- Larger signature+key size
- Relies on Rejection Sampling

FALCON

Fast-Fourier Lattice-based Compact Signatures over NTRU

- Based on the Trapdoor approach (GPV)
- Smaller signature+key size
- Slower signing

Why is Rejection Sampling a Limitation?

- Rejection sampling causes repetition of the sign algorithm
- Not having a constant-time signing algorithm could introduce attacks
- In case of Dilithium, the repetition can be high (e.g., around 10 times)

LWE Problem

Matrix form of LWE: Given parameters n, k, q, m and two distributions D_s and D_e:

- Sample $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$
- Sample $\mathbf{S} \leftarrow D_S^{k\times n}$, $\mathbf{E} \leftarrow D_e^{k\times m}$

Problem?

Given $(\mathbf{A}, \mathbf{Y} = \mathbf{S}\mathbf{A} + \mathbf{E})$, find $\mathbf{S}$ or $\mathbf{E}$

A Naïve Approach: Lattice-Based Signatures from FS

$(SK, PK) \leftarrow KeyGen(1^\kappa)$	
1:	$\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}$
2:	$\boldsymbol{S} \leftarrow D_1^{k\times n}, \boldsymbol{E} \leftarrow D_1^{k\times m}$
3:	$\boldsymbol{Y} := \boldsymbol{SA} + \boldsymbol{E} \ mod\ q$
4:	$PK := (\boldsymbol{A}, \boldsymbol{Y})$, $SK := (\boldsymbol{S}, \boldsymbol{E})$

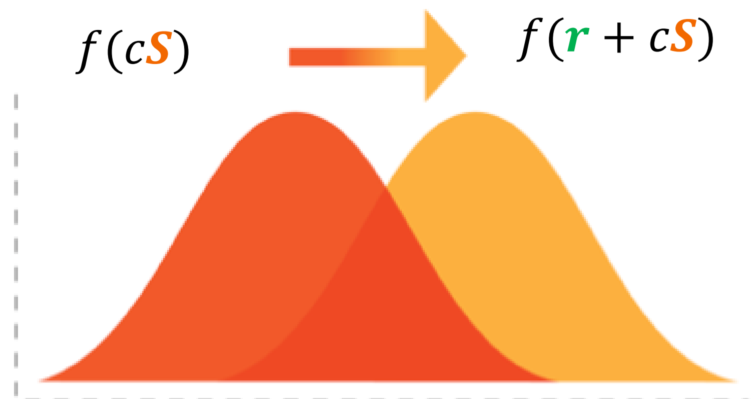
$\{0,1\} \leftarrow Verify(\langle \mathbf{z}, c\rangle, \mu, PK)$	
1:	$\boldsymbol{t} := \mathbf{z}\boldsymbol{A} - \boldsymbol{cY} = \boldsymbol{rA} - \boldsymbol{cE}$
2:	**IF:** $\boldsymbol{c} == H(\mu, \boldsymbol{t})$, output $valid$ ⚠

Signatures will never verify!

$\boldsymbol{r}$ needs to be from a smaller distribution for the underlying problem to hold

$\mathbf{z}, \boldsymbol{c} \leftarrow Sign(SK, \mu)$	
1:	$\boldsymbol{r} \leftarrow D_2^{1\times n}$
2:	$\boldsymbol{c} := H(\mu, \boldsymbol{rA}\ mod\ q)$
3:	$\mathbf{z} := \boldsymbol{r} + \boldsymbol{cS}$ ⚠

$\mathbf{z}$ leaks the distribution of $\boldsymbol{S}$



What do we need now???

Rejection Sampling

An ancient concept!!!

Applications to lattices due to [Lyu09]

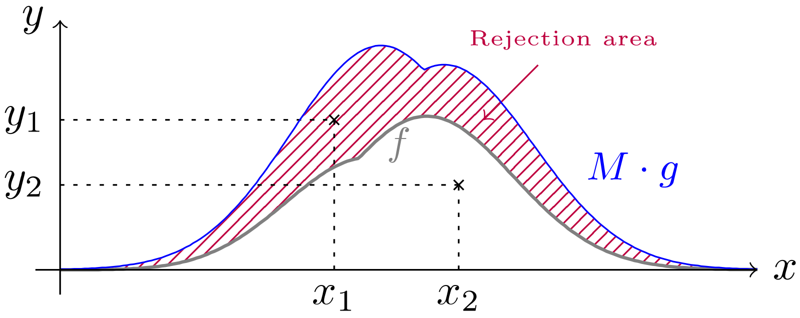
- Sample from an arbitrary target probability distribution f, given a source bound to a different probability distribution g.

Have access to $g(x)$

Want the output to be in $f(x)$

$$\Pr[x] = \frac{f(x)}{M \times g(x)}$$

M is some positive real



Lattice-Based Signatures from FS

$(SK, PK) \leftarrow KeyGen(1^\kappa)$	
1:	$\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}$
2:	$\boldsymbol{S} \leftarrow D_1^{k\times n}, \boldsymbol{E} \leftarrow D_1^{k\times m}$
3:	$\boldsymbol{Y} := \boldsymbol{SA} + \boldsymbol{E} \bmod q$
4:	$PK := (\boldsymbol{A}, \boldsymbol{Y})$, SK $:= (\boldsymbol{S}, \boldsymbol{E})$

$\{0,1\} \leftarrow Verify(\langle \mathbf{z}, c\rangle, \mu, PK)$	
1:	$\boldsymbol{v}' := \mathbf{z}\boldsymbol{A} - \boldsymbol{cY}$ $\boldsymbol{w} := \lfloor \boldsymbol{v}' \rceil$
2:	**IF:** $c == H(\mu, \boldsymbol{w})$ AND $\lVert \mathbf{z} \rVert_\infty \notin BAD_2$ $Valid$

$\mathbf{z}, c \leftarrow Sign(SK, \mu)$	
1:	$\boldsymbol{r} \leftarrow D_2^{1\times n}$
2:	$\boldsymbol{c} := H(\mu, \lfloor \boldsymbol{rA} \rceil \bmod q)$
3:	$\boldsymbol{v} = \boldsymbol{rA} - \boldsymbol{cE}$ **IF:** $\lVert [\boldsymbol{v}]_{2^\lambda} \rVert_\infty \in BAD_1$ $Restart$
4:	$\mathbf{z} := \boldsymbol{r} + \boldsymbol{cS}$
5:	**IF:** $\lVert \mathbf{z} \rVert_\infty \in BAD_2$ $Restart$

The new Scheme

	$(SK, PK) \leftarrow KeyGen(1^\kappa)$
1:	$\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}$
2:	$\boldsymbol{S} \leftarrow \mathbb{Z}_q^{h\times n}, \boldsymbol{E} \leftarrow D_1^{h\times m}$
3:	$\boldsymbol{Y} := \boldsymbol{SA} + \boldsymbol{E} \ mod\ q$
4:	$PK := (\boldsymbol{A}, \boldsymbol{Y}), SK := (\boldsymbol{S}, \boldsymbol{E})$

	$\mathbf{z}, c \leftarrow Sign(SK, \mu)$
1:	$\boldsymbol{r} \leftarrow \mathbb{Z}_q^{1\times n}$
2:	$\boldsymbol{c} := H(\mu, \lfloor \boldsymbol{rA} \rceil \ mod\ q)$
3:	$\boldsymbol{v} = \boldsymbol{rA} - \boldsymbol{cE}$ **IF:** $\left\Vert [\boldsymbol{v}]_{2^\lambda} \right\Vert_\infty \in BAD_1$ $Restart$
3:	$\mathbf{z} := \boldsymbol{r} + \boldsymbol{cS}$

	$\{0,1\} \leftarrow Verify(\langle \mathbf{z}, c\rangle, \mu, PK)$
1:	$\boldsymbol{v}' = \boldsymbol{zA} - \boldsymbol{cY}$ $\boldsymbol{w} := \lfloor \boldsymbol{v}' \rceil$
2:	**IF:** $c == H(\mu, \boldsymbol{w})$ $Valid$

Our underlying assumptions

- *Bounded Distance Decoding (BDD) Problem:* Given uniform $\boldsymbol{A} \leftarrow R_q^{l\times k}$ and $\boldsymbol{y} \leftarrow R_q^k$, the problem asks to find a $\boldsymbol{z}$ such that $\boldsymbol{z}^t\boldsymbol{A}$ is (*very*) close to $\boldsymbol{y}$ or $\boldsymbol{y}^{\boldsymbol{t}} - \boldsymbol{z}^t\boldsymbol{A}$ is small.

- Depending on the parameters/dimensions of $\boldsymbol{y}, \boldsymbol{A}$, this can be statistically or computationally hard

- Computational hardness results in more efficient parameters

The Proof

- Based on TWO hybrids:

Hybrid 1	Hybrid 2
$\boldsymbol{s} \leftarrow R_q^l, \boldsymbol{e} \leftarrow \mathrm{X}^k, \boldsymbol{A} \leftarrow R_q^{l\times k}$ $\boldsymbol{y}^t \coloneqq \boldsymbol{s}^t\boldsymbol{A} + \boldsymbol{e}^t$ $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_H;Sign(\cdot)}$ • Secret key is not known to the reduction • Queries are answered by using RO	$\boldsymbol{A} \leftarrow R_q^{l\times k}$ $\boldsymbol{y} \leftarrow R_q^k$ $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_H;Sign(\cdot)}$ • Public keys are uniform random • There is no secret key • Infeasible (based on BDD) for $\mathcal{A}$ to forge without RO

$\sigma \leftarrow Sign(m)$
Repeat till $\mathbf{z}^{\mathrm{t}}\mathbf{A} - \mathrm{c}\mathbf{y}^{\mathrm{t}} \in Good$ $\boldsymbol{z} \leftarrow R_q^l, c \leftarrow C, \boldsymbol{w} \coloneqq \lfloor \boldsymbol{z}^t\boldsymbol{A} - c\boldsymbol{y}^t \rceil_p$ $H(\boldsymbol{w}, (\boldsymbol{A}, \boldsymbol{y}), m) \coloneqq c$ Return $\sigma \coloneqq (\mathbf{z}, \mathrm{c})$

Results

- Two set of parameters are provided:
  - Statistical hardness of BDD, i.e., security in QROM
  - Computational hardness of BDD
- We do not use the public-key size optimization method in Dilithium

Table 2: Comparison with Dilithium-QROM and qTESLA-provable.

Parameters	Classical security	PK size	Sign size	Exp. repetitions
Dilithium-QROM standard	140	7712	5696	4.3
qTESLA-p standard	140	14880	2592	3.45*
Ours standard-I	138.1	13856	3588.5	5.41
Ours standard-II	140.2	14368	3716.5	4.08
Ours standard-III	139.4	19232	3972.5	1.55
Dilithium-QROM high	175	9632	7098	2.2
qTESLA-p high	279	38432	5664	3.84*
Ours high	170.0	17888	6021.8	1.83

Table 4: Comparison with Dilithium.

Parameters	Classical security	PK size	Sign size	Exp. repetitions
Dilithium standard	138	1472	2701	6.6
Ours standard-I	138.4	7200	3716.5	2.02
Ours standard-II	138.1	6944	3972.5	2.33
Dilithium high	174	1760	3366	4.3
Ours high	170.0	9952	6021.8	1.96

Removing the Remaining Rejection Condition

- One rejection condition is left to check $\lfloor \boldsymbol{z}^t \boldsymbol{A} - c\boldsymbol{y} \rceil_p = \lfloor \boldsymbol{r}^t \boldsymbol{A} \rceil_p$ holds

Can we remove the remaining rejection condition???

Looking at two potential approaches:

1. Extracting consistent values from commitments with errors
    - Two functions $g(\cdot)$ and $f(\cdot)$ that map $\boldsymbol{r}^t\boldsymbol{A}$ and $\boldsymbol{r}^t\boldsymbol{A} + \hat{\boldsymbol{e}}^t$, for unbounded error term $\hat{\boldsymbol{e}}$
    - $g(\boldsymbol{r}^t\boldsymbol{A})$ should serve as commitment OR preserve high min-entropy
    - Guo et al. [23]: For a poly q, no *balanced* functions $g(\cdot)$ and $f(\cdot)$ can guarantee $g(\boldsymbol{r}^t\boldsymbol{A}) = f(\boldsymbol{r}^t\boldsymbol{A} + \hat{\boldsymbol{e}}^t)$
2. Adapting the Reconciliation Mechanism used in lattice-based key exchange

Lattice-Based Key Exchange

$$\boldsymbol{M}_{Alice} = \boldsymbol{S}_{Alice}\boldsymbol{A} + \boldsymbol{E}_{Alice}$$

$$\boldsymbol{M}_{Bob} = \boldsymbol{A}\boldsymbol{S}_{Bob} + \boldsymbol{E}_{Bob}$$

$$k_{Alice} = \lfloor \boldsymbol{S}_{Alice}\boldsymbol{M}_{Bob} \rceil$$

$$k_{Bob} = \lfloor \boldsymbol{M}_{Alice}\boldsymbol{S}_{Bob} \rceil$$

h=Hint(k_{Alice})

$$\boxed{k_{Alice} = \text{Reconcile}(h, k_{Bob})}$$

Reconciled Scheme

$(SK, PK) \leftarrow KeyGen(1^\kappa)$	
1:	$\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}$
2:	$\boldsymbol{S} \leftarrow \mathbb{Z}_q^{n\times m}, \boldsymbol{E} \leftarrow D_1^{h\times m}$
3:	$\boldsymbol{Y} := \boldsymbol{SA} + \boldsymbol{E} \bmod q$
4:	$PK := (\boldsymbol{A}, \boldsymbol{Y})$, $SK := (\boldsymbol{S}, \boldsymbol{E})$

$z, c, h \leftarrow Sign(SK, \mu)$	
1:	$\boldsymbol{r} \leftarrow \mathbb{Z}_q^{n\times 1}$
2:	$\boldsymbol{c} := H(\mu, \lfloor \boldsymbol{rA} \rceil \bmod q)$
3:	$h := \mathrm{Hint}(\lfloor \boldsymbol{rA} \rceil)$
~~3:~~	$\cancel{\boldsymbol{v} = \boldsymbol{rA} - \boldsymbol{cE}}$ ~~**IF:**~~ $\cancel{\lfloor \boldsymbol{v} \rceil \neq \lfloor \boldsymbol{rA} \rceil}$ $\cancel{Restart}$
4:	$\boldsymbol{z} := \boldsymbol{r} + \boldsymbol{cS}$

$\{0,1\} \leftarrow Verify(z, c, h, \mu, PK)$	
1:	$\boldsymbol{v}' = \boldsymbol{zA} - \boldsymbol{cY}$ $\lfloor \boldsymbol{rA} \rceil := \mathrm{Reconcile}(h, \lfloor \boldsymbol{v}' \rceil)$
2:	**IF:** $c = H(\mu, \lfloor \boldsymbol{rA} \rceil)$ $Valid$

Problem?

Reconciled Scheme

$(SK, PK) \leftarrow KeyGen(1^\kappa)$	
1:	$\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}$
2:	$\boldsymbol{S} \leftarrow \mathbb{Z}_q^{n\times m}, \boldsymbol{E} \leftarrow D_1^{h\times m}$
3:	$\boldsymbol{Y} \coloneqq \boldsymbol{S}\boldsymbol{A} + \boldsymbol{E} \bmod q$
4:	$PK \coloneqq (\boldsymbol{A}, \boldsymbol{Y})$, $SK \coloneqq (\boldsymbol{S}, \boldsymbol{E})$

$\{0,1\} \leftarrow Verify(z, c, h, \mu, PK)$	
1:	$\boldsymbol{v}' = \boldsymbol{z}\boldsymbol{A} - \boldsymbol{c}\boldsymbol{Y}$ $\lfloor \boldsymbol{r}\boldsymbol{A} \rceil \coloneqq \mathrm{Reconcile}(h, \lfloor \boldsymbol{v}' \rceil)$
2:	**IF:** $c = H(\mu, \lfloor \boldsymbol{r}\boldsymbol{A} \rceil)$ $Valid$

Problem!

$$\boldsymbol{v}' - \lfloor \boldsymbol{r}\boldsymbol{A} \rceil = cE - E_{\boldsymbol{r},\boldsymbol{A}}$$

Example, let c be in {0,1}

- Get n samples of $E_{\boldsymbol{r},\boldsymbol{A}}$
- Get n samples of $E - E_{\boldsymbol{r},\boldsymbol{A}}$
- Compute $\frac{1}{n}\sum(E_{\boldsymbol{r},\boldsymbol{A}}) - \frac{1}{n}\sum(E - E_{\boldsymbol{r},\boldsymbol{A}})$ to get a good estimate of $\mathrm{E} = \mu(E - E_{\boldsymbol{r},\boldsymbol{A}}) + \mu(E_{\boldsymbol{r},\boldsymbol{A}})$

Thank you