Basic Lattice Constructions II: Signatures

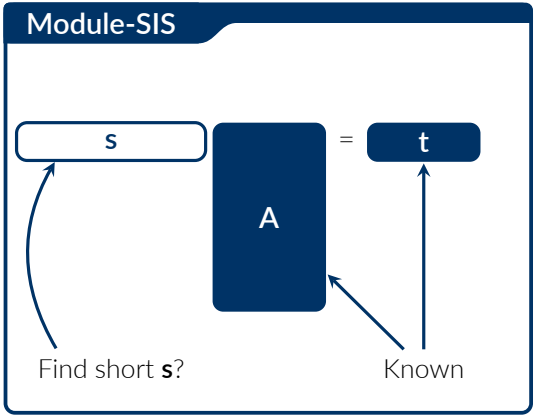
Thomas Prest

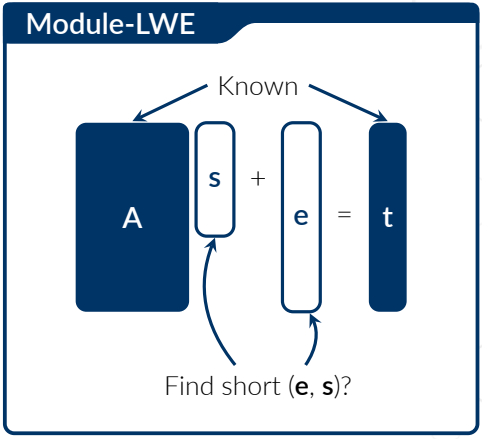
PQShield

ASCRYPTO 2021

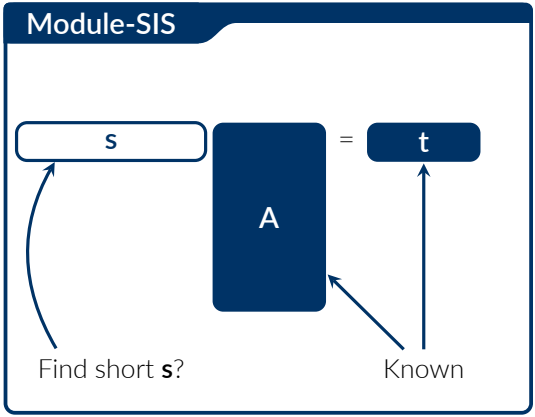
(More) Lattice Problems

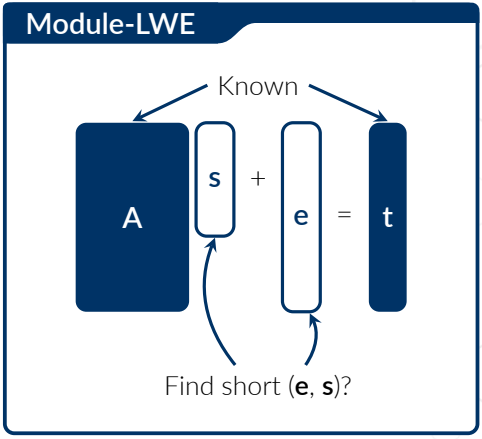
Selected problems





Selected problems

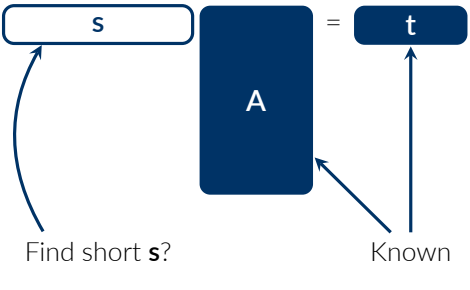




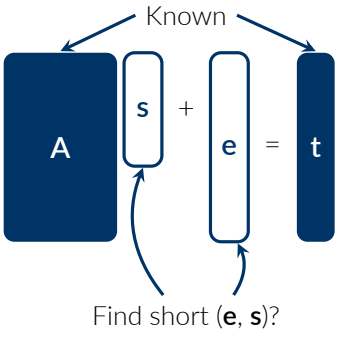
- In cryptography, SIS and LWE are typically used in very different regimes:
  - *SIS in a dense regime:* each $\mathbf{t}$ has several preimages, $(\mathbf{A}, \mathbf{t})$ is statistically uniform
  - *LWE in a sparse regime:* each $\mathbf{t}$ has at ≤ 1 preimage, $(\mathbf{A}, \mathbf{t})$ is computationally uniform

Selected problems

Module-SIS



Module-LWE

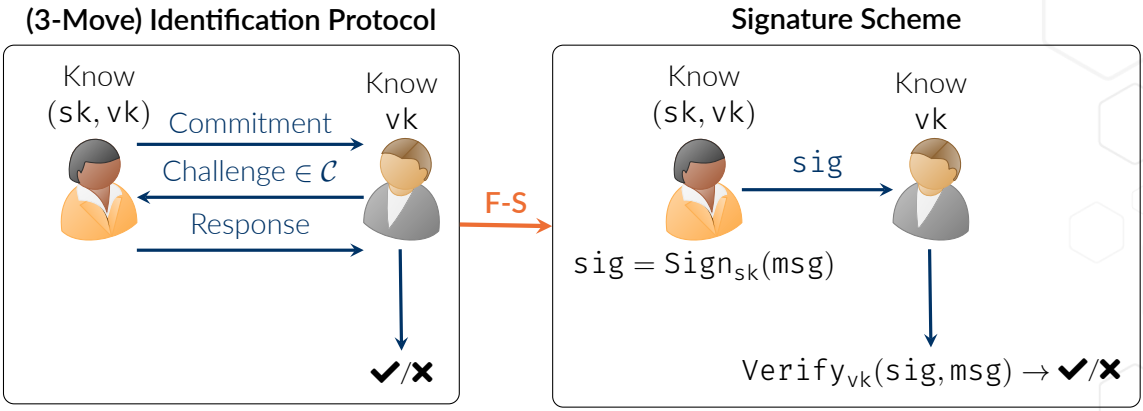


NTRU

Find small f, g such that $g \cdot f^{-1} = h$ in $\mathcal{R}_q = \mathbb{Z}_q[x]/(\varphi)$

Fiat-Shamir Signatures

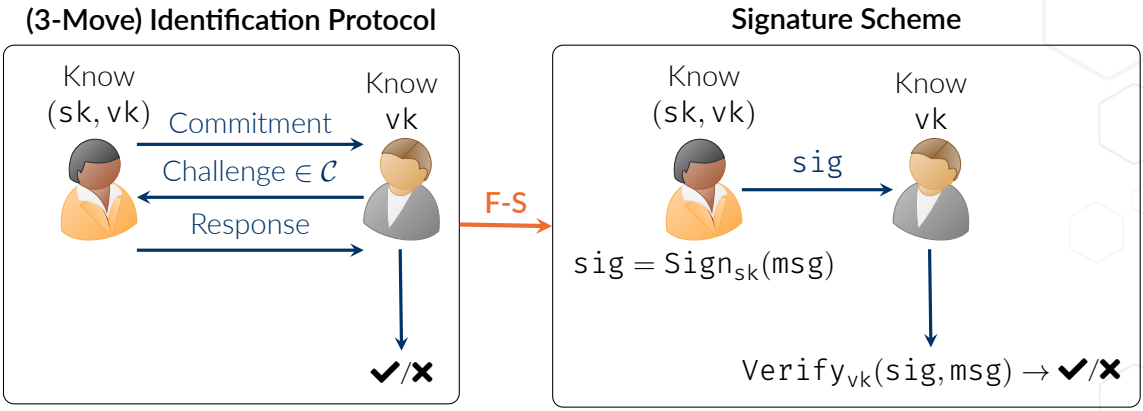
Fiat-Shamir Signatures



F-S refers to the Fiat-Shamir transform:

- The challenge is now defined as $H(\text{Commitment}\|\texttt{msg})$.
- The signature is (Commitment,Response).

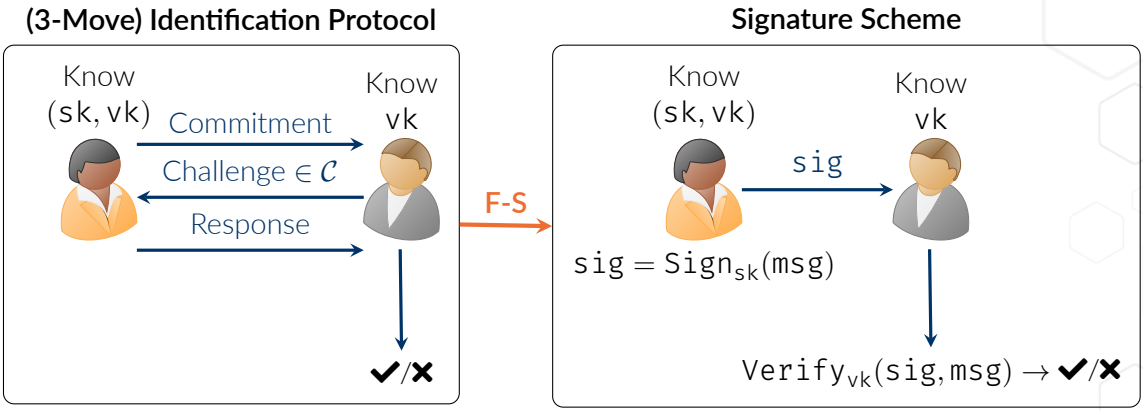
Fiat-Shamir Signatures



We obtain an existentially unforgeable signature scheme in the ROM if the ID protocol is:

1. **Correct:** An honest prover can convince a verifier he knows $\mathtt{sk}$
2. **Honest verifier zero-knowledge:** A valid transcript leaks no information about $\mathtt{sk}$
3. **Sound:** A dishonest prover cannot convince a verifier he knows $\mathtt{sk}$ (except with probability $\leq \varepsilon$ (the *soundness error*))

Fiat-Shamir Signatures



It is often more convenient to work with *(2-)special-soundness* instead of soundness:

4. **2-Special-Soundness**: Given two valid transcripts $(com, chal, resp)$ and $(com, chal', resp')$, we can extract the secret.

2-special soundness implies soundness: If a protocol is 2-special-sound with a challenge space $\mathcal{C}$, then it is sound with soundness error $1/|\mathcal{C}|$.

Schnorr signatures (Fiat-Shamir w/ discrete log)

PQ SHIELD

Keygen$(g \in \mathbb{G})$

1. $x \leftarrow \mathbb{Z}_q^\times$ $(q = |\mathbb{G}|)$
2. $h \leftarrow g^x$
3. $\mathtt{sk} := x, \mathtt{vk} := h$

Sign$(\mathtt{msg}, \mathtt{sk})$

1. $r \leftarrow \mathbb{Z}_q^\times$
2. $u \leftarrow g^r$ (Commitment)
3. $c \leftarrow H(u \| \mathtt{msg})$ (Challenge)
4. $z \leftarrow r - cx$ (Response)
5. $\mathtt{sig} := (u, z)$

Verify$(\mathtt{msg}, \mathtt{vk})$

1. Accept if and only if $(g^z \cdot h^c = u)$

⚠ This slide shows the signature scheme but we will discuss 3 (sufficient) properties of the ID protocol:

1. **Correctness**
2. **HVZK**
4. **Special soundness**

Schnorr signatures (Fiat-Shamir w/ discrete log)

Keygen$(g \in \mathbb{G})$

1. $x \leftarrow \mathbb{Z}_q^\times$ $(q = |\mathbb{G}|)$
2. $h \leftarrow g^x$
3. $\mathtt{sk} := x, \mathtt{vk} := h$

Sign$(\mathtt{msg}, \mathtt{sk})$

1. $r \leftarrow \mathbb{Z}_q^\times$
2. $u \leftarrow g^r$ (Commitment)
3. $c \leftarrow H(u \| \mathtt{msg})$ (Challenge)
4. $z \leftarrow r - cx$ (Response)
5. $\mathtt{sig} := (u, z)$

Verify$(\mathtt{msg}, \mathtt{vk})$

1. Accept if and only if $(g^z \cdot h^c = u)$

Correctness (of ID protocol):

$$g^z \cdot h^c = g^{r-cx} \cdot g^{xc} = g^r = u \qquad (1)$$

Schnorr signatures (Fiat-Shamir w/ discrete log)

PQ SHIELD

Keygen$(g \in \mathbb{G})$

1. $x \leftarrow \mathbb{Z}_q^\times$ $(q = |\mathbb{G}|)$
2. $h \leftarrow g^x$
3. $\mathtt{sk} := x, \mathtt{vk} := h$

Sign$(\mathtt{msg}, \mathtt{sk})$

1. $r \leftarrow \mathbb{Z}_q^\times$
2. $u \leftarrow g^r$ (Commitment)
3. $c \leftarrow H(u \| \mathtt{msg})$ (Challenge)
4. $z \leftarrow r - cx$ (Response)
5. $\mathtt{sig} := (u, z)$

Verify$(\mathtt{msg}, \mathtt{vk})$

1. Accept if and only if $(g^z \cdot h^c = u)$

HVZK (of ID protocol): We can simulate perfectly a valid transcript (u, c, z) as follows:

1. Sample $c \in \mathcal{C}$
2. Sample $z \in \mathbb{Z}_q^\times$
3. Compute $u := g^z \cdot h^c$

Schnorr signatures (Fiat-Shamir w/ discrete log)

PQ SHIELD

Keygen$(g \in \mathbb{G})$

1. $x \leftarrow \mathbb{Z}_q^\times$ $\quad (q = |\mathbb{G}|)$
2. $h \leftarrow g^x$
3. $\mathsf{sk} := x, \mathsf{vk} := h$

Sign$(\mathbf{msg}, \mathbf{sk})$

1. $r \leftarrow \mathbb{Z}_q^\times$
2. $u \leftarrow g^r$ $\quad$ (Commitment)
3. $c \leftarrow H(u \| \mathtt{msg})$ $\quad$ (Challenge)
4. $z \leftarrow r - cx$ $\quad$ (Response)
5. $\mathtt{sig} := (u, z)$

Verify$(\mathbf{msg}, \mathbf{vk})$

1. Accept if and only if $(g^z \cdot h^c = u)$

Special-Soundness: Suppose we have two slightly different transcripts:

- Interaction 1: $(u, c_1, z_1 \mid g^{z_1} \cdot h^{c_1} = u)$
- Interaction 2: $(u, c_2, z_2 \mid g^{z_2} \cdot h^{c_2} = u)$

If $(c_2 - c_1)$ is invertible, then:

$$h = g^{(z_1 - z_2)/(c_2 - c_1)}$$

Schnorr signatures (Fiat-Shamir w/ discrete log)

PQ SHIELD

Keygen$(g \in \mathbb{G})$

1. $x \leftarrow \mathbb{Z}_q^\times$ $\quad (q = |\mathbb{G}|)$
2. $h \leftarrow g^x$
3. $\mathtt{sk} := x, \mathtt{vk} := h$

Sign$(\mathtt{msg}, \mathtt{sk})$

1. $r \leftarrow \mathbb{Z}_q^\times$
2. $u \leftarrow g^r$ $\quad$ (Commitment)
3. $c \leftarrow H(u \| \mathtt{msg})$ $\quad$ (Challenge)
4. $z \leftarrow r - cx$ $\quad$ (Response)
5. $\mathtt{sig} := (c, z)$

Verify$(\mathtt{msg}, \mathtt{vk})$

1. Accept if and only if $H(g^z \cdot h^c \| \mathtt{msg}) = c$

Optimisation: send the challenge c instead of the commitment u.

Now with SIS and LWE

Fiat-Shamir with SIS

Keygen$(\mathbf{A} \in \mathcal{R}_q^{k \times \ell})$

1. $\mathbf{s} \leftarrow \chi$ (short)
2. $\mathbf{t} \leftarrow \mathbf{A}\mathbf{s}$
3. $\mathtt{sk} := \mathbf{s}, \mathtt{vk} := \mathbf{t}$

Sign$(\mathtt{msg}, \mathbf{sk})$

1. $\mathbf{r} \leftarrow \chi$ (short)
2. $\mathbf{u} \leftarrow \mathbf{A}\mathbf{r}$
3. $\mathbf{c} \leftarrow H(\mathbf{u} \| \mathtt{msg})$
4. $\mathbf{z} \leftarrow \mathbf{r} - \mathbf{c}\mathbf{s}$
5. $\mathtt{sig} := (\mathbf{u}, \mathbf{z})$

Verify$(\mathtt{msg}, \mathbf{vk}, \mathtt{sig})$

1. Accept iff ($\mathbf{z}$ is short) and ($\mathbf{A}\mathbf{z} - \mathbf{c}\mathbf{t} = \mathbf{u}$).

Fiat-Shamir with SIS

Keygen$(\mathbf{A} \in \mathcal{R}_q^{k \times \ell})$

1. $\mathbf{s} \leftarrow \chi$ (short)
2. $\mathbf{t} \leftarrow \mathbf{A}\mathbf{s}$
3. $\mathsf{sk} := \mathbf{s}, \mathsf{vk} := \mathbf{t}$

Sign$(\mathtt{msg}, \mathbf{sk})$

1. $\mathbf{r} \leftarrow \chi$ (short)
2. $\mathbf{u} \leftarrow \mathbf{A}\mathbf{r}$
3. $\mathbf{c} \leftarrow H(\mathbf{u} \| \mathtt{msg})$
4. $\mathbf{z} \leftarrow \mathbf{r} - \mathbf{c}\mathbf{s}$
5. $\mathtt{sig} := (\mathbf{u}, \mathbf{z})$

Verify$(\mathtt{msg}, \mathbf{vk}, \mathtt{sig})$

1. Accept iff ($\mathbf{z}$ is short) and ($\mathbf{A}\mathbf{z} - \mathbf{c}\mathbf{t} = \mathbf{u}$).

Correctness: Exercise.

Fiat-Shamir with SIS

Keygen$(\mathbf{A} \in \mathcal{R}_q^{k \times \ell})$

1. $\mathbf{s} \leftarrow \chi$ (short)
2. $\mathbf{t} \leftarrow \mathbf{A}\mathbf{s}$
3. $\mathtt{sk} := \mathbf{s}, \mathtt{vk} := \mathbf{t}$

Sign$(\mathtt{msg}, \mathtt{sk})$

1. $\mathbf{r} \leftarrow \chi$ (short)
2. $\mathbf{u} \leftarrow \mathbf{A}\mathbf{r}$
3. $\mathbf{c} \leftarrow H(\mathbf{u} \| \mathtt{msg})$
4. $\mathbf{z} \leftarrow \mathbf{r} - \mathbf{c}\mathbf{s}$
5. $\mathtt{sig} := (\mathbf{u}, \mathbf{z})$

Verify$(\mathtt{msg}, \mathtt{vk}, \mathtt{sig})$

1. Accept iff ($\mathbf{z}$ is short) and ($\mathbf{A}\mathbf{z} - \mathbf{c}\mathbf{t} = \mathbf{u}$).

Soundness: Using the previous technique:

- Interaction 1: $(\mathbf{u}, \mathbf{c}, \mathbf{z} \mid \mathbf{A}\mathbf{z} - \mathbf{c}\mathbf{t} = \mathbf{u})$
- Interaction 2: $(\mathbf{u}, \mathbf{c}', \mathbf{z}' \mid \mathbf{A}\mathbf{z}' - \mathbf{c}'\mathbf{t} = \mathbf{u})$

We have:

$$[\, \mathbf{A} \,\|\, \mathbf{t} \,] \cdot \begin{bmatrix} \mathbf{z} - \mathbf{z}' \\ \mathbf{c} - \mathbf{c}' \end{bmatrix} = \mathbf{0} \tag{1}$$

Question: Does the prover know a shorter SIS solution to $[\, \mathbf{A} \,\|\, \mathbf{t} \,]$ than (1)?

Fiat-Shamir with SIS

Keygen$(\mathbf{A} \in \mathcal{R}_q^{k \times \ell})$

1. $\mathbf{s} \leftarrow \chi$ (short)
2. $\mathbf{t} \leftarrow \mathbf{A}\mathbf{s}$
3. $\mathtt{sk} := \mathbf{s}, \mathtt{vk} := \mathbf{t}$

Sign$(\mathtt{msg}, \mathtt{sk})$

1. $\mathbf{r} \leftarrow \chi$ (short)
2. $\mathbf{u} \leftarrow \mathbf{A}\mathbf{r}$
3. $\mathbf{c} \leftarrow H(\mathbf{u} \| \mathtt{msg})$ (short)
4. $\mathbf{z} \leftarrow \mathbf{r} - \mathbf{c}\mathbf{s}$
5. $\mathtt{sig} := (\mathbf{u}, \mathbf{z})$

Verify$(\mathtt{msg}, \mathtt{vk}, \mathtt{sig})$

1. Accept iff ($\mathbf{z}$ is short) and ($\mathbf{A}\mathbf{z} - \mathbf{c}\mathbf{t} = \mathbf{u}$).

Solution: The prover knows

$$[\, \mathbf{A} \,\|\, \mathbf{t} \,] \cdot \begin{bmatrix} \mathbf{s} \\ -1 \end{bmatrix} = \mathbf{0} \qquad (1)$$

- The solution that the prover knows is shorter by a factor $\geq \|\mathbf{c}\|$ than the one he proves (*slack* in the proof).
- An adverse constraint is that $\mathbf{c}$ should have enough entropy.

Awkward situation, but we can live with it.

Fiat-Shamir with SIS

Keygen$(\mathbf{A} \in \mathcal{R}_q^{k \times \ell})$

1. $\mathbf{s} \leftarrow \chi$ (short)
2. $\mathbf{t} \leftarrow \mathbf{As}$
3. $\mathtt{sk} := \mathbf{s}, \mathtt{vk} := \mathbf{t}$

Sign$(\mathtt{msg}, \mathtt{sk})$

1. $\mathbf{r} \leftarrow \chi$ (short)
2. $\mathbf{u} \leftarrow \mathbf{Ar}$
3. $\mathbf{c} \leftarrow H(\mathbf{u} \| \mathtt{msg})$ (short)
4. $\mathbf{z} \leftarrow \mathbf{r} - \mathbf{cs}$
5. $\mathtt{sig} := (\mathbf{u}, \mathbf{z})$

Verify$(\mathtt{msg}, \mathtt{vk}, \mathtt{sig})$

1. Accept iff ($\mathbf{z}$ is short) and ($\mathbf{Az} - \mathbf{ct} = \mathbf{u}$).

HVZK: Attempt to simulate a valid transcript:

1. Sample $\mathbf{c} \in \mathcal{C}$
2. Sample $\mathbf{z}$ in the expected range
3. Compute $\mathbf{u} = \mathbf{Az} - \mathbf{ct}$

This is not a perfect simulation.

→ Bigger problem: actual transcripts leak information about the secret.

→ Exploited by key recovery attacks (e.g. [GJSS01] on NTRU-based [HPS01])

Fiat-Shamir with SIS

Keygen$(\mathbf{A} \in \mathcal{R}_q^{k \times \ell})$

1. $\mathbf{s} \leftarrow \chi$ (short)
2. $\mathbf{t} \leftarrow \mathbf{As}$
3. $\mathtt{sk} := \mathbf{s}, \mathtt{vk} := \mathbf{t}$

Sign$(\mathtt{msg}, \mathtt{sk})$

1. $\mathbf{r} \leftarrow \chi$ (short)
2. $\mathbf{u} \leftarrow \mathbf{Ar}$
3. $\mathbf{c} \leftarrow H(\mathbf{u} \| \mathtt{msg})$ (short)
4. $\mathbf{z} \leftarrow \mathbf{r} - \mathbf{cs}$
5. Rejection sampling step
6. $\mathtt{sig} := (\mathbf{u}, \mathbf{z})$

Verify$(\mathtt{msg}, \mathtt{vk}, \mathtt{sig})$

1. Accept iff ($\mathbf{z}$ is short) and ($\mathbf{Az} - \mathbf{ct} = \mathbf{u}$).

Solution: use rejection sampling [Lyu09].

- High-level idea: the signer may restart the signing procedure if the signature leaks information about $\mathtt{sk}$.
- Example: if $\mathbf{r}$ is uniform over a small set S, accept iff $\mathbf{z} \in S$.
- Rejection sampling may be:
  - deterministic (e.g. Dilithium [LDK+17])
  - probabilistic (e.g. BLISS [DDLL13])

Fiat-Shamir with SIS

Keygen$(\mathbf{A} \in \mathcal{R}_q^{k \times \ell})$

1. $\mathbf{s} \leftarrow \chi$ (short)
2. $\mathbf{t} \leftarrow \mathbf{As}$
3. $\mathtt{sk} := \mathbf{s}, \mathtt{vk} := \mathbf{t}$

Sign$(\mathtt{msg}, \mathtt{sk})$

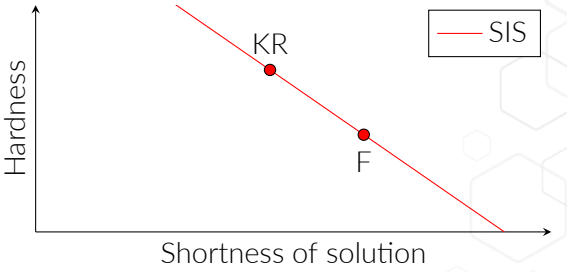
1. $\mathbf{r} \leftarrow \chi$ (short)
2. $\mathbf{u} \leftarrow \mathbf{Ar}$
3. $\mathbf{c} \leftarrow H(\mathbf{u} \| \mathtt{msg})$ (short)
4. $\mathbf{z} \leftarrow \mathbf{r} - \mathbf{cs}$
5. Rejection sampling step
6. $\mathtt{sig} := (\mathbf{u}, \mathbf{z})$

Verify$(\mathtt{msg}, \mathtt{vk}, \mathtt{sig})$

1. Accept iff ($\mathbf{z}$ is short) and ($\mathbf{Az} - \mathbf{ct} = \mathbf{u}$).

Concrete hardness:

- *Key-recovery*: SIS with a short $\mathbf{s}$
- *Forgery*: SIS with a short-ish $\mathbf{z}$



Exercise 1

Suppose:

- $\mathcal{R} = \mathbb{Z}[x]/(x^d + 1)$.
- Each coefficient of $\mathbf{s} \in \mathcal{R}^\ell$ is sampled uniformly in $\{-s, \ldots, s\}$.
- $\mathcal{C}$ is the subset $\mathcal{C} \subseteq \mathcal{R}$ of polynomials with w ones and $(d-1)$ zeroes.
- Each coefficient of $\mathbf{r} \in \mathcal{R}^\ell$ is sampled uniformly $\{-r, \ldots, r\}$, with $r > dws$.
- The response $\mathbf{z} := \mathbf{r} - \mathbf{cs}$ is accepted if and only if $\mathbf{z} \in \mathrm{Supp}(\mathbf{r})$.

Give a simple upper bound on the probability that a given response $\mathbf{z}$ is accepted.
(*hint:* compute $|\mathrm{Supp}(\mathbf{z})|$ and use the inequality $(\forall f, g \in \mathcal{R}, \|fg\|_\infty \leq d\|f\|_\infty\|g\|_\infty)$)

Exercise 1

Suppose:

- $\mathcal{R} = \mathbb{Z}[x]/(x^d + 1)$.
- Each coefficient of $\mathbf{s} \in \mathcal{R}^\ell$ is sampled uniformly in $\{-s, \ldots, s\}$.
- $\mathcal{C}$ is the subset $\mathcal{C} \subseteq \mathcal{R}$ of polynomials with w ones and $(d-1)$ zeroes.
- Each coefficient of $\mathbf{r} \in \mathcal{R}^\ell$ is sampled uniformly $\{-r, \ldots, r\}$, with $r > dws$.
- The response $\mathbf{z} := \mathbf{r} - \mathbf{cs}$ is accepted if and only if $\mathbf{z} \in \mathrm{Supp}(\mathbf{r})$.

Give a simple upper bound on the probability that a given response $\mathbf{z}$ is accepted. (*hint:* compute $|\mathrm{Supp}(\mathbf{z})|$ and use the inequality $(\forall f, g \in \mathcal{R}, \|fg\|_\infty \leq d\|f\|_\infty\|g\|_\infty)$)

Solution 1

For any $(\mathbf{z}_0, \mathbf{z}_1) \in \mathrm{Supp}(\mathbf{r}) \times \mathrm{Supp}(\mathbf{z}) \setminus \mathrm{Supp}(\mathbf{r}), \mathbb{P}[\mathbf{z} = \mathbf{z}_0] \geq \mathbb{P}[\mathbf{z} = \mathbf{z}_1]$. Therefore:

$$\mathbb{P}_{\mathbf{s},\mathbf{c},\mathbf{z}}[\mathbf{z} \in \mathrm{Supp}(\mathbf{r})] \leq \frac{|\mathrm{Supp}(\mathbf{r})|}{|\mathrm{Supp}(\mathbf{z})|} \leq \left(\frac{2r+1}{2(r+dws)+1}\right)^{\ell d}.$$

Note: For this probability to be $\Omega(1)$, it suffices that $r = \Omega(\ell d \cdot dws)$.

Fiat-Shamir w/ (LWE+SIS) [Lyu12]

Keygen$(\mathbf{A} \in \mathcal{R}_q^{k \times \ell})$

1. $\mathbf{s}_1, \mathbf{s}_2 \leftarrow \chi_1 \times \chi_2$ (short)
2. $\mathbf{t} \leftarrow \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$
3. $\mathsf{sk} := (\mathbf{s}_1, \mathbf{s}_2), \mathsf{vk} := \mathbf{t}$

Sign$(\mathtt{msg}, \mathtt{sk})$

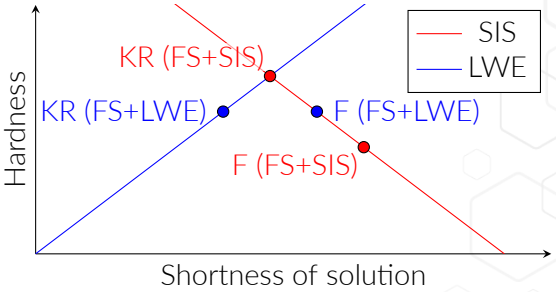
1. $\mathbf{r}_1, \mathbf{r}_2 \leftarrow \chi_3 \times \chi_4$ (short)
2. $\mathbf{u} \leftarrow \mathbf{A}\mathbf{r}_1 + \mathbf{r}_2$
3. $\mathbf{c} \leftarrow H(\mathbf{u} \| \mathtt{msg})$ (short)
4. $\mathbf{z}_1 \leftarrow \mathbf{r}_1 - \mathbf{c}\mathbf{s}_1$
5. $\mathbf{z}_2 \leftarrow \mathbf{r}_2 - \mathbf{c}\mathbf{s}_2$
6. Rejection sampling step
7. $\mathtt{sig} := (\mathbf{u}, \mathbf{z}_1, \mathbf{z}_2)$

Verify$(\mathtt{msg}, \mathtt{vk}, \mathtt{sig})$

1. Accept iff $(\mathbf{z}_1, \mathbf{z}_2)$ is short and $\mathbf{A}\mathbf{z}_1 + \mathbf{z}_2 - \mathbf{t}\mathbf{c} = \mathbf{u}$

Concrete hardness:

- *Key-recovery:* LWE with a short $\mathbf{s}$
- *Forgery:* SIS with a short-ish $\mathbf{z}$



Fiat-Shamir w/ (LWE+SIS) [Lyu12]

PQ SHIELD

Keygen$(\mathbf{A} \in \mathcal{R}_q^{k \times \ell})$

1. $\mathbf{s}_1, \mathbf{s}_2 \leftarrow \chi_1 \times \chi_2$ (short)
2. $\mathbf{t} \leftarrow \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$
3. $\mathtt{sk} := (\mathbf{s}_1, \mathbf{s}_2), \mathtt{vk} := \mathbf{t}$

Sign$(\mathtt{msg}, \mathtt{sk})$

1. $\mathbf{r}_1, \mathbf{r}_2 \leftarrow \chi_3 \times \chi_4$ (short)
2. $\mathbf{u} \leftarrow \mathbf{A}\mathbf{r}_1 + \mathbf{r}_2$
3. $\mathbf{c} \leftarrow H(\mathbf{u} \| \mathtt{msg})$ (short)
4. $\mathbf{z}_1 \leftarrow \mathbf{r}_1 - \mathbf{c}\mathbf{s}_1$
5. $\mathbf{z}_2 \leftarrow \mathbf{r}_2 - \mathbf{c}\mathbf{s}_2$
6. Rejection sampling step
7. $\mathtt{sig} := (\mathbf{u}, \mathbf{z}_1, \mathbf{z}_2)$

Verify$(\mathtt{msg}, \mathtt{vk}, \mathtt{sig})$

1. Accept iff $(\mathbf{z}_1, \mathbf{z}_2)$ is short and $\mathbf{A}\mathbf{z}_1 + \mathbf{z}_2 - \mathbf{t}\mathbf{c} = \mathbf{u}$

The LWE regime is more interesting than the SIS regime from an efficiency point of view.

LWE also allows two optimisations that can be summarised by:

> "*If you are solving LWE for* $(\mathbf{A}, \mathbf{t} + \mathbf{e})$, *you are also solving LWE for* $(\mathbf{A}, \mathbf{t})$."

These optimisations are:

- The Bai-Gaibraith trick
- The Dilithium trick

We will note MSB := "most significant bits" (the proportion may vary).

Fiat-Shamir w/ (LWE+SIS) - Bai-Galbraith trick [BG14a]

PQ SHIELD

Keygen$(\mathbf{A} \in \mathcal{R}_q^{k \times \ell})$

1. $\mathbf{s}_1, \mathbf{s}_2 \leftarrow \chi_1 \times \chi_2$ (short)
2. $\mathbf{t} \leftarrow \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$
3. $\mathsf{sk} := (\mathbf{s}_1, \mathbf{s}_2), \mathsf{vk} := \mathbf{t}$

Sign$(\mathbf{msg}, \mathbf{sk})$

1. $\mathbf{r} \leftarrow \chi_3$ (short)
2. $\mathbf{u} \leftarrow \mathrm{MSB}(\mathbf{Ar})$
3. $\mathbf{c} \leftarrow H(\mathbf{u} \| \mathsf{msg})$ (short)
4. $\mathbf{z} \leftarrow \mathbf{r} - \mathbf{c}\mathbf{s}_1$
5. Rejection sampling step
6. $\mathsf{sig} := (\mathbf{u}, \mathbf{z})$

Verify$(\mathbf{msg}, \mathbf{vk}, \mathbf{sig})$

1. Accept iff $\mathbf{z}$ is short and $\mathrm{MSB}(\mathbf{Az} - \mathbf{tc}) = \mathbf{u}$

High-level idea is that the response sends only $\mathbf{z} := \mathbf{z}_1$ instead of $(\mathbf{z}_1, \mathbf{z}_2)$.

- "*Our scheme proves knowledge of only* $\mathbf{s}$. *The proof of knowledge of* $\mathbf{e}$ *becomes implicit in the verification.*" [BG14b]
- To preserve correctness, only check that $(\mathbf{Az} - \mathbf{tc})$ and $\mathbf{u}$ match on their MSBs.
- If moderate, bit dropping only mildly affect the hardness of LWE.
- Also, no error term $\mathbf{r}_2$ necessary in $\mathbf{u}$.

Fiat-Shamir w/ (LWE+SIS) - Dilithium trick [LDK+17]

PQ SHIELD

Keygen$(\mathbf{A} \in \mathcal{R}_q^{k \times \ell})$

1. $\mathbf{s}_1, \mathbf{s}_2 \leftarrow \chi_1 \times \chi_2$ (short)
2. $\mathbf{t} \leftarrow \mathrm{MSB}(\mathbf{A}\mathbf{s}_1 + \mathbf{s}_2)$
3. $\mathtt{sk} := (\mathbf{s}_1, \mathbf{s}_2), \mathtt{vk} := \mathbf{t}$

Sign$(\mathtt{msg}, \mathtt{sk})$

1. $\mathbf{r} \leftarrow \chi_3$ (short)
2. $\mathbf{u} \leftarrow \mathrm{MSB}(\mathbf{A}\mathbf{r})$
3. $\mathbf{c} \leftarrow H(\mathbf{u} \| \mathtt{msg})$ (short)
4. $\mathbf{z} \leftarrow \mathbf{r} - \mathbf{c}\mathbf{s}_1$
5. Rejection sampling step
6. $\mathtt{sig} := (\mathbf{u}, \mathbf{z})$

Verify$(\mathtt{msg}, \mathtt{vk}, \mathtt{sig})$

1. Accept iff $\mathbf{z}$ is short and $\mathrm{MSB}(\mathbf{A}\mathbf{z} - \mathbf{t}\mathbf{c}) = \mathbf{u}$

Naive version: the signer drops the least significant bits of $\mathbf{t}$ during `Keygen`.

- → vk gets shorter.
- → Intuitively, this adds an error term $\mathbf{e}$ to $\mathbf{t}$
- → $\mathbf{A}\mathbf{z} - \mathbf{c}\mathbf{t} = \mathbf{u} - \mathbf{c}(\mathbf{s}_2 + \mathbf{e})$

With mild bit dropping, the signature is valid with good probability (if it isn't, restart).

Fiat-Shamir w/ (LWE+SIS) - Dilithium trick [LDK+17]

PQ SHIELD

Keygen$(\mathbf{A} \in \mathcal{R}_q^{k \times \ell})$

1. $\mathbf{s}_1, \mathbf{s}_2 \leftarrow \chi_1 \times \chi_2$ (short)
2. $\mathbf{t} \leftarrow \mathrm{MSB}(\mathbf{A}\mathbf{s}_1 + \mathbf{s}_2)$
3. $\mathtt{sk} := (\mathbf{s}_1, \mathbf{s}_2), \mathtt{vk} := \mathbf{t}$

Sign$(\mathtt{msg}, \mathtt{sk})$

1. $\mathbf{r} \leftarrow \chi_3$ (short)
2. $\mathbf{u} \leftarrow \mathrm{MSB}(\mathbf{A}\mathbf{r})$
3. $\mathbf{c} \leftarrow H(\mathbf{u} \| \mathtt{msg})$ (short)
4. $\mathbf{z} \leftarrow \mathbf{r} - \mathbf{c}\mathbf{s}_1$
5. $\mathbf{h} = \mathrm{MSB}(\mathbf{A}\mathbf{z} - \mathbf{t}\mathbf{c}) \oplus \mathbf{u}$
6. Rejection sampling step
7. $\mathtt{sig} := (\mathbf{u}, \mathbf{z}, \mathbf{h})$

Verify$(\mathtt{msg}, \mathtt{vk}, \mathtt{sig})$

1. Accept iff $\mathbf{z}$ is short and $\mathrm{MSB}(\mathbf{A}\mathbf{z} - \mathbf{t}\mathbf{c}) \oplus \mathbf{h} = \mathbf{u}$

Advanced version: more aggressive bit dropping during `Keygen`.

- → vk gets even shorter.
- → Due to the larger error term $\mathbf{c}(\mathbf{s}_2 + \mathbf{e})$, $\mathrm{MSB}(\mathbf{A}\mathbf{z} - \mathbf{c}\mathbf{t})$ and $\mathbf{u}$ no longer match.
- → Solution: send the difference in the signature!

Question: Is this always secure?

Fiat-Shamir w/ (LWE+SIS) - Dilithium trick [LDK+17]

PQ SHIELD

Keygen$(\mathbf{A} \in \mathcal{R}_q^{k \times \ell})$

1. $\mathbf{s}_1, \mathbf{s}_2 \leftarrow \chi_1 \times \chi_2$ (short)
2. $\mathbf{t} \leftarrow \mathrm{MSB}(\mathbf{A}\mathbf{s}_1 + \mathbf{s}_2)$
3. $\mathtt{sk} := (\mathbf{s}_1, \mathbf{s}_2), \mathtt{vk} := \mathbf{t}$

Sign$(\mathtt{msg}, \mathtt{sk})$

1. $\mathbf{r} \leftarrow \chi_3$ (short)
2. $\mathbf{u} \leftarrow \mathrm{MSB}(\mathbf{A}\mathbf{r})$
3. $\mathbf{c} \leftarrow H(\mathbf{u} \| \mathtt{msg})$ (short)
4. $\mathbf{z} \leftarrow \mathbf{r} - \mathbf{c}\mathbf{s}_1$
5. $\mathbf{h} = \mathrm{MSB}(\mathbf{A}\mathbf{z} - \mathbf{t}\mathbf{c}) \oplus \mathbf{u}$
6. Rejection sampling step
7. $\mathtt{sig} := (\mathbf{u}, \mathbf{z}, \mathbf{h})$

Verify$(\mathtt{msg}, \mathtt{vk}, \mathtt{sig})$

1. Accept iff $\mathbf{z}$ is short and $\mathrm{MSB}(\mathbf{A}\mathbf{z} - \mathbf{t}\mathbf{c}) \oplus \mathbf{h} = \mathbf{u}$

Answer: If there is no constraint on $\mathbf{h}$, then forgery is trivial.

1. Generate $\mathbf{u}$ and a short $\mathbf{z}$ at random.
2. Compute $\mathbf{c}$ honestly.
3. Set $\mathbf{h} = \mathrm{MSB}(\mathbf{A}\mathbf{z} - \mathbf{c}\mathbf{t}) \oplus \mathbf{u}$

Then $(\mathbf{u}, \mathbf{z}, \mathbf{h})$ is a valid signature.

For security, the format of $\mathbf{h}$ is therefore constrained in [LDK+17]:

- → Fixed number ω of non-zero entries
- → Non-zero entries can only modify one bit (the "smallest MSB").

Fiat-Shamir w/ (LWE+SIS) - Dilithium trick [LDK+17]

PQ SHIELD

Keygen$(\mathbf{A} \in \mathcal{R}_q^{k \times \ell})$

1. $\mathbf{s}_1, \mathbf{s}_2 \leftarrow \chi_1 \times \chi_2$ (short)
2. $\mathbf{t} \leftarrow \text{MSB}(\mathbf{A}\mathbf{s}_1 + \mathbf{s}_2)$
3. $\text{sk} := (\mathbf{s}_1, \mathbf{s}_2), \text{vk} := \mathbf{t}$

Sign$(\mathtt{msg}, \mathtt{sk})$

1. $\mathbf{r} \leftarrow \chi_3$ (short)
2. $\mathbf{u} \leftarrow \text{MSB}(\mathbf{A}\mathbf{r})$
3. $\mathbf{c} \leftarrow H(\mathbf{u} \| \mathtt{msg})$ (short)
4. $\mathbf{z} \leftarrow \mathbf{r} - \mathbf{c}\mathbf{s}_1$
5. Rejection sampling step
6. $\mathtt{sig} := (\mathbf{c}, \mathbf{z}, \mathbf{h})$

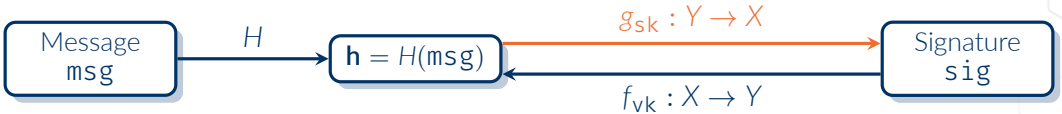
Verify$(\mathtt{msg}, \mathtt{vk}, \mathtt{sig})$

1. Accept iff $\mathbf{z}$ is short and $H(\text{MSB}(\mathbf{A}\mathbf{z} - \mathbf{t}\mathbf{c}) \oplus \mathbf{h}, \mathtt{msg}) = \mathbf{c}$

And don't forget the generic "*send challenge instead of commitment*" trick!

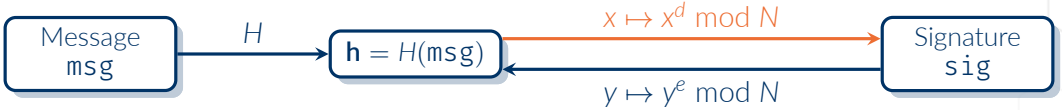
Hash-then-Sign

Signatures based on *Hash-then-Sign*



- **The signer** computes $\mathbf{h} = H(\mathtt{msg})$, then $\mathtt{sig} = g_{\mathtt{sk}}(\mathbf{h})$ using the signing key $\mathtt{sk}$.
- **The verifier** computes $\mathbf{h} = H(\mathtt{msg})$, then $\mathbf{h}' = f_{\mathtt{vk}}(\mathtt{sig})$ using the verification key $\mathtt{vk}$, and checks that the results match (i.e. $\mathbf{h}' = \mathbf{h}$).
- It is typical to salt the hash: $\mathbf{h} = H(\mathtt{msg}, \mathtt{salt})$, we omit the salt for concision.

The case of RSA signatures



Example with RSA signatures:

- $g_{\mathsf{sk}}(x) = x^d \bmod N$, and $f_{\mathsf{vk}}(y) = y^e \bmod N$.
- $(f_{\mathsf{vk}}, g_{\mathsf{sk}})$ are often abstracted as trapdoor permutations [BR96, Cor02]:
  1. Given only vk, f_{vk} is hard to invert for (almost) all inputs.
  2. $f_{\mathsf{vk}} \circ g_{\mathsf{sk}} = Id$ and $X = Y$ (hence f_{vk} and g_{sk} are permutations).

  We do not know how to realise this abstraction under PQ assumptions.

The case of lattices (first attempt)

Following [GGH97, HHP+03], let us try to instantiate this blueprint with lattices:

- *Verification key:* vk is a (pseudo)random matrix $\mathbf{A} \in \mathcal{R}_q^{n \times m}$.
- *Signing key:* sk is a short matrix $\mathbf{B} \in \mathcal{R}_q^{m \times m}$ such that $\mathbf{A} \cdot \mathbf{B} = \mathbf{0} \bmod q$.

For example, following [HHP+03, PFH+17] let $f, g, F, G \in \mathcal{R}$ such that:

$$fG - gF = q \tag{1}$$

$$h := g/f \bmod q \tag{2}$$

Exercise: Show that $\mathbf{A} = \begin{bmatrix} 1 & h \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} g & G \\ -f & -F \end{bmatrix}$ satisfy $\mathbf{A} \cdot \mathbf{B} = \mathbf{0} \bmod q$.

The case of lattices (first attempt)

Following [GGH97, HHP+03], let us try to instantiate this blueprint with lattices:

- *Verification key:* vk is a (pseudo)random matrix $\mathbf{A} \in \mathcal{R}_q^{n \times m}$.
- *Signing key:* sk is a short matrix $\mathbf{B} \in \mathcal{R}_q^{m \times m}$ such that $\mathbf{A} \cdot \mathbf{B} = \mathbf{0} \bmod q$.

For example, following [HHP+03, PFH+17] let $f, g, F, G \in \mathcal{R}$ such that:

$$fG - gF = q \tag{1}$$

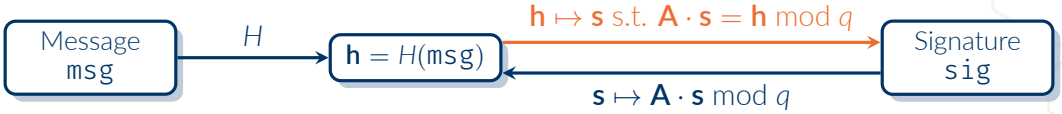
$$h := g/f \bmod q \tag{2}$$

Exercise: Show that $\mathbf{A} = \begin{bmatrix} 1 & h \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} g & G \\ -f & -F \end{bmatrix}$ satisfy $\mathbf{A} \cdot \mathbf{B} = \mathbf{0} \bmod q$.

Solution: By applying (2) and (1) on the left and right column:

$$\mathbf{A} \cdot \mathbf{B} \bmod q = \begin{bmatrix} g - hf & \frac{fG - gF}{f} \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix}.$$

The case of lattices (first attempt)



Recall ($\mathbf{A} \cdot \mathbf{B} = \mathbf{0} \bmod q$) and $\mathbf{B}$ is short (say, $\|\mathbf{B}\|_2 = \max_{\{\mathbf{x}\}} \frac{\|\mathbf{B}\cdot\mathbf{x}\|_2}{\|\mathbf{x}\|_2}$ is small).

→ *Signing:*

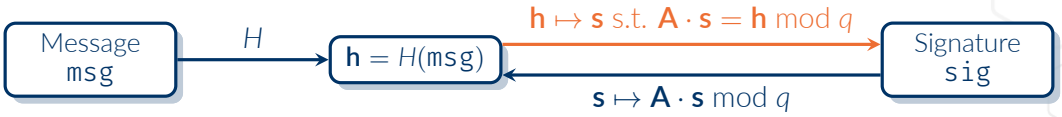
1. Hash `msg` to a point $\mathbf{h} \in \mathcal{R}_q^n$.
2. Compute $\mathbf{c} \in \mathcal{R}_q^m$ s.t. $\mathbf{A} \cdot \mathbf{c} = \mathbf{h}$.
3. Compute $\mathbf{v} := \mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c} \rceil$
4. The signature is $\mathbf{s} := \mathbf{c} - \mathbf{v}$

→ *Verification:*

1. Check that $\mathbf{A} \cdot \mathbf{s} = \mathbf{h}$.
2. Check that $\mathbf{s}$ is short (say, $\|\mathbf{s}\|_2$ is small).

Exercise: Show a honestly generated signature passes verification (assume $\mathcal{R} = \mathbb{Z}$).

The case of lattices (first attempt)



Recall ($\mathbf{A} \cdot \mathbf{B} = \mathbf{0} \bmod q$) and $\mathbf{B}$ is short (say, $\|\mathbf{B}\|_2 = \max_{\{\mathbf{x}\}} \frac{\|\mathbf{B}\cdot\mathbf{x}\|_2}{\|\mathbf{x}\|_2}$ is small).

→ *Signing:*

1. Hash `msg` to a point $\mathbf{h} \in \mathcal{R}_q^n$.
2. Compute $\mathbf{c} \in \mathcal{R}_q^m$ s.t. $\mathbf{A} \cdot \mathbf{c} = \mathbf{h}$.
3. Compute $\mathbf{v} := \mathbf{B} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c} \rceil$
4. The signature is $\mathbf{s} := \mathbf{c} - \mathbf{v}$

→ *Verification:*

1. Check that $\mathbf{A} \cdot \mathbf{s} = \mathbf{h}$.
2. Check that $\mathbf{s}$ is short (say, $\|\mathbf{s}\|_2$ is small).

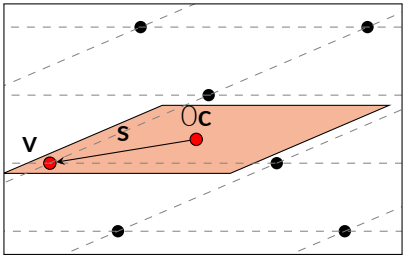
Exercise: Show a honestly generated signature passes verification (assume $\mathcal{R} = \mathbb{Z}$).

Solution:

1. $\mathbf{A} \cdot \mathbf{s} = \mathbf{A} \cdot (\mathbf{c} - \mathbf{v}) = \overbrace{\mathbf{A} \cdot \mathbf{c}}^{=\mathbf{h}} - \overbrace{\mathbf{A} \cdot \mathbf{B}}^{=\mathbf{0}} \cdot \lfloor \mathbf{B}^{-1} \cdot \mathbf{c} \rceil = \mathbf{h}$
2. If $\mathbf{c} = \mathbf{B} \cdot \mathbf{t}$,[1] then $\mathbf{s} = \mathbf{B} \cdot (\mathbf{t} - \lfloor \mathbf{t} \rceil) \in \mathbf{B} \cdot \left[-\frac{1}{2}, \frac{1}{2}\right]^m$. Therefore $\|\mathbf{s}\|_2 \leq \frac{\sqrt{m}}{2} \|\mathbf{B}\|_2$

[1] Note: the entries of $\mathbf{t}$ are real-valued (numbers in this exercise, or polynomials in general).

What about security?

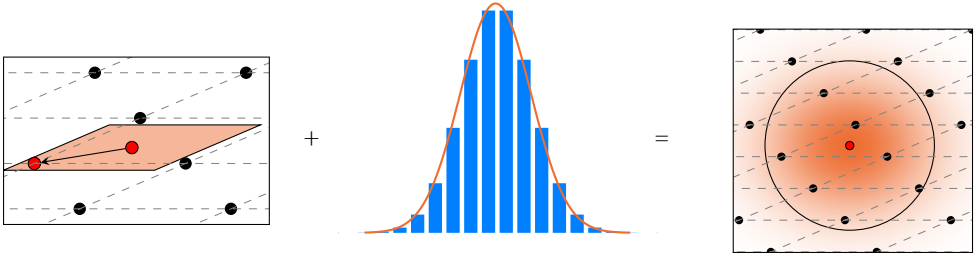


We have seen in the previous slide that $\mathtt{sig} \in \mathtt{sk} \cdot \left[-\frac{1}{2}, \frac{1}{2}\right]^m$.

- This means signatures will leak the shape of the signing key $\mathtt{sk} = \mathbf{B}$.
- Exploited in statistical key-recovery attacks [NR06, DN12]

Solution: trapdoor sampling [GPV08]

By carefully randomising the signing procedure, we can make the signature distribution independent of the secret basis $\mathbf{B}$.



Note $\lfloor \mathbf{x} \rceil_r$ the integral vector such that each coefficient of $\mathbf{x}$ is rounded according to a discrete Gaussian (thus rounding is probabilistic).

- If $\mathbf{v} \leftarrow \mathbf{B} \cdot \left\lfloor \mathbf{B}^{-1} \cdot \mathbf{c} \right\rceil_r$, then $\mathbf{v}$ follows an elliptic distribution skewed by $\mathbf{B}$ $\Rightarrow$ the signing key still leaks.
- If $\mathbf{v} \leftarrow \mathbf{B} \cdot \left\lfloor \mathbf{B}^{-1} \cdot \mathbf{c} + \mathbf{M} \cdot \lfloor \mathbf{0} \rceil_{r_1} \right\rceil_{r_2}$ for appropriately chosen $(r_1, r_2, \mathbf{M})$, then there is no leakage of the short basis (= the signing key) [Pei10].

More inversion algoritnms

There exist several algorithms for computing short preimages **s** to **h**:

- **Round-off algorithm** [Bab85, Bab86] and its randomised variant [Pei10]
- **Nearest-plane algorithm** [Bab85, Bab86] and its randomised variant [GPV08]
  - *Pros:* sample shorter vectors than round-off
  - *Cons:* a bit more complex, slower over structured lattices
- **Fast Fourier nearest plane** [DP16] and its randomised variant [PFH+17]
  - *Pros:* same quality as nearest-plane, as fast as round-off over structured lattices
  - *Cons:* complex
- See also the **Micciancio-Peikert framework** [MP12, GM18]
- Etc.

More inversion algoritnms

There exist several algorithms for computing short preimages **s** to **h**:

- **Round-off algorithm** [Bab85, Bab86] and its randomised variant [Pei10]
- **Nearest-plane algorithm** [Bab85, Bab86] and its randomised variant [GPV08]
  - *Pros:* sample shorter vectors than round-off
  - *Cons:* a bit more complex, slower over structured lattices
- **Fast Fourier nearest plane** [DP16] and its randomised variant [PFH+17]
  - *Pros:* same quality as nearest-plane, as fast as round-off over structured lattices
  - *Cons:* complex
- See also the **Micciancio-Peikert framework** [MP12, GM18]
- Etc.

We realise a weaker notion than trapdoor permutations: *trapdoor preimage sampleabe functions* (TPSFs).

- Trapdoor permutation ⇒ TPSF
- TPSF suffices for a FDH proof

See [HPA21] for a more detailed discussion.

Putting everything together

Keygen(1^λ)

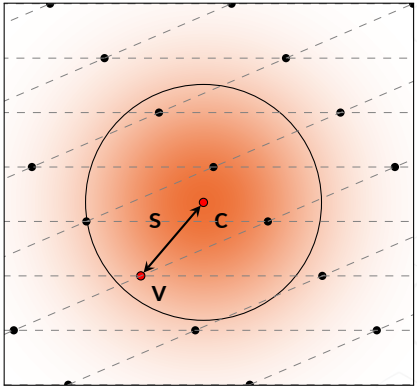
1. Gen. matrices $\mathbf{A}, \mathbf{B}$ s.t.:
   - $\mathbf{A} \cdot \mathbf{B} = 0$
   - $\mathbf{B}$ has small coefficients
2. $\mathtt{vk} := \mathbf{A}$, $\mathtt{sk} := \mathbf{B}$

Sign$(\mathtt{msg}, \mathtt{sk} = \mathbf{B})$

1. Compute $\mathbf{c}$ such that $\mathbf{A} \cdot \mathbf{c} = H(\mathtt{msg})$
2. $\mathbf{v} \leftarrow$ vector in $\Lambda(\mathbf{B})$, close to $\mathbf{c}$
3. $\mathtt{sig} := \mathbf{s} = (\mathbf{c} - \mathbf{v})$

Verify$(\mathtt{msg}, \mathtt{vk} = \mathbf{A}, \mathtt{sig} = \mathbf{s})$

Check ($\mathbf{s}$ short) & ($\mathbf{A} \cdot \mathbf{s} = H(\mathtt{msg})$)



Conclusion

Classical and lattice-based schemes share many similarities:

- *Same paradigms:* Schnorr-type identification protocols, El Gamal-type encryption, full-domain hash, etc.
- The analogy sometimes fail, but this is also informative
  - Highlights differences between the two families of assumptions
  - Some optimisations are unique to the lattice setting (e.g. bit dropping)

For a more comprehensive overview of lattice-based cryptography, see Simon's Institute 2020 programme "*Lattices: Algorithms, Complexity, and Cryptography*": https://simons.berkeley.edu/programs/lattices2020.

Questions?

L Babai.

On lovasz' lattice reduction and the nearest lattice point problem.

In *Proceedings on STACS 85 2Nd Annual Symposium on Theoretical Aspects of Computer Science*, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

László Babai.

On lovasz' lattice reduction and the nearest lattice point problem.

Combinatorica, 6(1), 1986.

Shi Bai and Steven D. Galbraith.

An improved compression technique for signatures based on learning with errors.

In Josh Benaloh, editor, *CT-RSA 2014*, volume 8366 of *LNCS*, pages 28–47. Springer, Heidelberg, February 2014.

Shi Bai and Steven D. Galbraith.

An improved compression technique for signatures based on learning with errors (presentation), 2014.

https://docs.huihoo.com/rsaconference/usa-2014/cryp-t07-non-integral-asymmetric-functions-copy1.pdf.

Mihir Bellare and Phillip Rogaway.

The exact security of digital signatures: How to sign with RSA and Rabin.

In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 399–416. Springer, Heidelberg, May 1996.

Jean-Sébastien Coron.
Optimal security proofs for PSS and other signature schemes.
In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 272–287. Springer, Heidelberg, April / May 2002.

Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky.
Lattice signatures and bimodal Gaussians.
In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 40–56. Springer, Heidelberg, August 2013.

Léo Ducas and Phong Q. Nguyen.
Learning a zonotope and more: Cryptanalysis of NTRUSign countermeasures.
In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 433–450. Springer, Heidelberg, December 2012.

Léo Ducas and Thomas Prest.
Fast fourier orthogonalization.
In Sergei A. Abramov, Eugene V. Zima, and Xiao-Shan Gao, editors, *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2016, Waterloo, ON, Canada, July 19-22, 2016*, pages 191–198. ACM, 2016.

Oded Goldreich, Shafi Goldwasser, and Shai Halevi.
Public-key cryptosystems from lattice reduction problems.
In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 112–131. Springer, Heidelberg, August 1997.


Craig Gentry, Jakob Jonsson, Jacques Stern, and Michael Szydlo.

Cryptanalysis of the NTRU signature scheme (NSS) from Eurocrypt 2001.

In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 1–20. Springer, Heidelberg, December 2001.


Nicholas Genise and Daniele Micciancio.

Faster Gaussian sampling for trapdoor lattices with arbitrary modulus.

In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 174–203. Springer, Heidelberg, April / May 2018.


Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan.

Trapdoors for hard lattices and new cryptographic constructions.

In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.


Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte.

NTRUSIGN: Digital signatures using the NTRU lattice.

In Marc Joye, editor, *CT-RSA 2003*, volume 2612 of *LNCS*, pages 122–140. Springer, Heidelberg, April 2003.


James Howe, Thomas Prest, and Daniel Apon.

SoK: How (not) to design and implement post-quantum cryptography.

In Kenneth G. Paterson, editor, *CT-RSA 2021*, volume 12704 of *LNCS*, pages 444–477. Springer, Heidelberg, May 2021.


Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman.

NSS: An NTRU lattice-based signature scheme.

In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 211–228. Springer, Heidelberg, May 2001.


Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, and Damien Stehlé.

CRYSTALS-DILITHIUM.

Technical report, National Institute of Standards and Technology, 2017.

available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.


Vadim Lyubashevsky.

Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures.

In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Heidelberg, December 2009.


Vadim Lyubashevsky.

Lattice signatures without trapdoors.

In Pointcheval and Johansson [PJ12], pages 738–755.


Daniele Micciancio and Chris Peikert.

Trapdoors for lattices: Simpler, tighter, faster, smaller.

In Pointcheval and Johansson [PJ12], pages 700–718.

Phong Q. Nguyen and Oded Regev.
Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures.
In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 271–288. Springer, Heidelberg, May / June 2006.

Chris Peikert.
An efficient and parallel Gaussian sampler for lattices.
In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 80–97. Springer, Heidelberg, August 2010.

Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang.
FALCON.
Technical report, National Institute of Standards and Technology, 2017.
available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.

David Pointcheval and Thomas Johansson, editors.
EUROCRYPT 2012, volume 7237 of *LNCS*. Springer, Heidelberg, April 2012.