Lattice-based cryptography II
Constructions and implementation issues

Leon Groot Bruinderink

July 1st, 2019

Lattice-based cryptography II

In this talk:

- Introduction to (ring-)LWE
- Lattice-based key-exchange and encryption schemes
- Reaction attacks and countermeasures
- Lattice-based signature schemes
- Side-channel attacks and countermeasures

Lattice-based cryptography

Some features of lattice-based cryptography:

- Key-exchange, encryption, digital signatures
- But also more exotic stuff, e.g. homomorphic encryption

Lattice-based cryptography

Some features of lattice-based cryptography:

- Key-exchange, encryption, digital signatures
- But also more exotic stuff, e.g. homomorphic encryption
- Pro's:
  - The algorithms are quite fast
  - The keys, cipher-texts, signatures are *quite small*

Lattice-based cryptography

Some features of lattice-based cryptography:

- Key-exchange, encryption, digital signatures
- But also more exotic stuff, e.g. homomorphic encryption
- Pro's:
  - The algorithms are quite fast
  - The keys, cipher-texts, signatures are *quite small*
- Con's:
  - Many design parameters to choose (and attacks to avoid)
  - Asymptotic hardness results vs concrete security/cryptanalysis

Lattice-based cryptography

Some features of lattice-based cryptography:

- Key-exchange, encryption, digital signatures
- But also more exotic stuff, e.g. homomorphic encryption
- Pro's:
  - The algorithms are quite fast
  - The keys, cipher-texts, signatures are *quite small*
- Con's:
  - Many design parameters to choose (and attacks to avoid)
  - Asymptotic hardness results vs concrete security/cryptanalysis
- Largest category of NIST post-quantum submissions
- Some real-life experiments (e.g. Google)

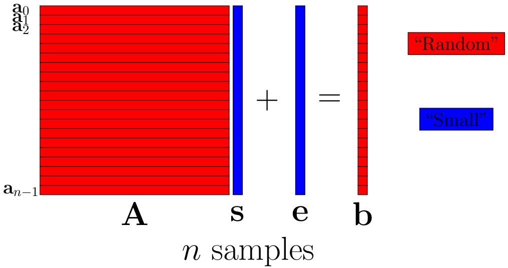
Learning With Errors

Learning with Errors (LWE) - Noisy inner product

- Let q be a prime, $n > 0$ (usually a power of 2), χ some *narrow* error distribution in $\mathbb{Z}_q$, $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i y_i \bmod q$ usual inner-product
- Let $\mathbf{s} \leftarrow \chi^n$ be a secret
- Given pairs of $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + e)$ with
  - $\mathbf{a} \in \mathbb{Z}_q^n$ sampled uniform at random
  - e sampled from χ
- (plain-) LWE: find $\mathbf{s}$

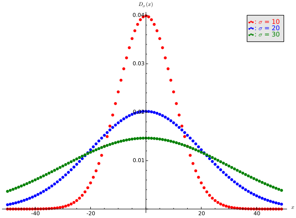
Learning with Errors (LWE) - Noisy inner product

- Let q be a prime, $n > 0$ (usually a power of 2), χ some *narrow* error distribution in $\mathbb{Z}_q$, $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i y_i \bmod q$ usual inner-product
- Let $\mathbf{s} \leftarrow \chi^n$ be a secret
- Given pairs of $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + e)$ with
  - $\mathbf{a} \in \mathbb{Z}_q^n$ sampled uniform at random
  - e sampled from χ
- (plain-) LWE: find $\mathbf{s}$



Learning with Errors (LWE) - Noisy inner product

- Let q be a prime, $n > 0$ (usually a power of 2), χ some *narrow* error distribution in $\mathbb{Z}_q$, $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i y_i \bmod q$ usual inner-product
- Let $\mathbf{s} \leftarrow \chi^n$ be a secret
- Given pairs of $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + e)$ with
  - $\mathbf{a} \in \mathbb{Z}_q^n$ sampled uniform at random
  - e sampled from χ
- (plain-) LWE: find $\mathbf{s}$
- Common choice for χ: the discrete Gaussian distribution D_σ
- Regev showed that a hard lattice problem can be reduced to LWE



Learning with Errors (LWE) - Noisy inner product

- Let q be a prime, $n > 0$ (usually a power of 2), χ some *narrow* error distribution in $\mathbb{Z}_q$, $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i y_i \bmod q$ usual inner-product
- Let $\mathbf{s} \leftarrow \chi^n$ be a secret
- Given pairs of $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + e)$ with
  - $\mathbf{a} \in \mathbb{Z}_q^n$ sampled uniform at random
  - e sampled from χ
- (plain-) LWE: find $\mathbf{s}$
- Common choice for χ: the discrete Gaussian distribution D_σ
- Regev showed that a hard lattice problem can be reduced to LWE
- First proposals for cryptosystems were quite big...

Ring-LWE: noisy polynomials

- Let q be a prime, $n > 0$ (usually a power of 2),
- Now define $\mathcal{R} = \mathbb{Z}_q[x]/(x^n \pm 1)$. Can add/subtract and multiply

$$\mathbf{f} = f_0 + f_1 x + \ldots + f_{n-1} x^{n-1} \in \mathcal{R}$$

$$f_i \in [0, q)$$

$$\mathbf{f} + \mathbf{g} \in \mathcal{R}$$

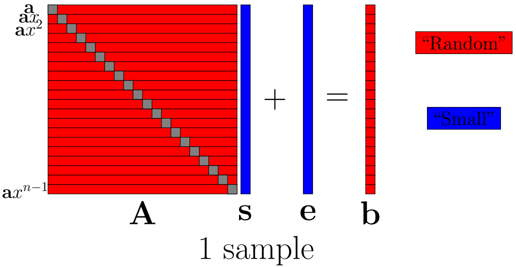
$$\mathbf{fg} \in \mathcal{R}$$

Ring-LWE: noisy polynomials

- Let q be a prime, $n > 0$ (usually a power of 2),
- Now define $\mathcal{R} = \mathbb{Z}_q[x]/(x^n \pm 1)$. Can add/subtract and multiply
- χ some *narrow* error distribution in $\mathcal{R}$
- Let $\mathbf{s} \leftarrow \chi$ be a secret
- Given pairs of $(\mathbf{a}, \mathbf{b} = \mathbf{as} + \mathbf{e})$ with
  - $\mathbf{a} \in \mathcal{R}$ sampled uniform at random
  - $\mathbf{e}$ sampled from χ
- ring-LWE: find $\mathbf{s}$

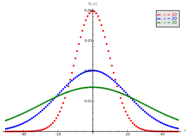
Ring-LWE: noisy polynomials

- Let q be a prime, $n > 0$ (usually a power of 2),
- Now define $\mathcal{R} = \mathbb{Z}_q[x]/(x^n \pm 1)$. Can add/subtract and multiply
- χ some *narrow* error distribution in $\mathcal{R}$
- Let $\mathbf{s} \leftarrow \chi$ be a secret
- Given pairs of $(\mathbf{a}, \mathbf{b} = \mathbf{as} + \mathbf{e})$ with
  - $\mathbf{a} \in \mathcal{R}$ sampled uniform at random
  - $\mathbf{e}$ sampled from χ
- ring-LWE: find $\mathbf{s}$



Ring-LWE: noisy polynomials

- Let q be a prime, $n > 0$ (usually a power of 2),
- Now define $\mathcal{R} = \mathbb{Z}_q[x]/(x^n \pm 1)$. Can add/subtract and multiply
- χ some *narrow* error distribution in $\mathcal{R}$
- Let $\mathbf{s} \leftarrow \chi$ be a secret
- Given pairs of $(\mathbf{a}, \mathbf{b} = \mathbf{as} + \mathbf{e})$ with
  - $\mathbf{a} \in \mathcal{R}$ sampled uniform at random
  - $\mathbf{e}$ sampled from χ
- ring-LWE: find $\mathbf{s}$
- Common choice for χ: the discrete Gaussian distribution D_σ^n
- Related to problems in *ideal* (or “cyclic”) lattices



Ring-LWE: noisy polynomials

- Let q be a prime, $n > 0$ (usually a power of 2),
- Now define $\mathcal{R} = \mathbb{Z}_q[x]/(x^n \pm 1)$. Can add/subtract and multiply
- χ some *narrow* error distribution in $\mathcal{R}$
- Let $\mathbf{s} \leftarrow \chi$ be a secret
- Given pairs of $(\mathbf{a}, \mathbf{b} = \mathbf{as} + \mathbf{e})$ with
  - $\mathbf{a} \in \mathcal{R}$ sampled uniform at random
  - $\mathbf{e}$ sampled from χ
- ring-LWE: find $\mathbf{s}$
- Common choice for χ: the discrete Gaussian distribution D_σ^n
- Related to problems in *ideal* (or "cyclic") lattices
- Many design choices (e.g. NTRU: $q = 2^\ell$; n prime; χ sparse)

Lattice-based Key-Exchange

Mimic Diffie-Hellman key-exchange

- Recall Diffie-Hellman key-exchange

Public: $G = \langle g \rangle$, $|G| = n$

Alice

$a \leftarrow_{\$} [1, n)$

$pub_A = g^a$

$K_A = (pub_B)^a = g^{ab}$

pub_A → (Alice to Bob)

pub_B ← (Bob to Alice)

Bob

$b \leftarrow_{\$} [1, n)$

$pub_B = g^b$

$K_B = (pub_A)^b = g^{ab}$

Mimic Diffie-Hellman key-exchange

- Recall Diffie-Hellman key-exchange

Public: $G = \langle g \rangle$, $|G| = n$

Alice

$a \leftarrow_{\$} [1, n)$

$pub_A = g^a$

$K_A = (pub_B)^a = g^{ab}$

Alice → Bob: pub_A

Bob → Alice: pub_B

Bob

$b \leftarrow_{\$} [1, n)$

$pub_B = g^b$

$K_B = (pub_A)^b = g^{ab}$

- Both parties end up with shared key $K = g^{ab}$

LWE key-exchange: noisy Diffie-Hellman

- ring-LWE key-exchange

Public: $\mathbf{g} \in \mathcal{R}$, distribution $\chi = D_\sigma^n$

Alice

$\mathbf{a}, \mathbf{e} \leftarrow_\$ \chi$

$pub_A = \mathbf{ga} + \mathbf{e}$

$S_A = (pub_B)\mathbf{a}$

$\quad = \mathbf{gab} + \mathbf{e}'\mathbf{a}$

$pub_A \longrightarrow$

$\longleftarrow pub_B$

Bob

$\mathbf{b}, \mathbf{e}' \leftarrow_\$ \chi$

$pub_B = \mathbf{gb} + \mathbf{e}'$

$S_B = (pub_A)\mathbf{b}$

$\quad = \mathbf{gab} + \mathbf{eb}$

LWE key-exchange: noisy Diffie-Hellman

- ring-LWE key-exchange

Public: $\mathbf{g} \in \mathcal{R}$, distribution $\chi = D_\sigma^n$

Alice

$\mathbf{a}, \mathbf{e} \leftarrow_\$ \chi$

$pub_A = \mathbf{ga} + \mathbf{e}$

$S_A = (pub_B)\mathbf{a}$

$= \mathbf{gab} + \mathbf{e}'\mathbf{a}$

Alice → Bob: pub_A

Bob → Alice: pub_B

Bob

$\mathbf{b}, \mathbf{e}' \leftarrow_\$ \chi$

$pub_B = \mathbf{gb} + \mathbf{e}'$

$S_B = (pub_A)\mathbf{b}$

$= \mathbf{gab} + \mathbf{eb}$

- $\mathbf{a}, \mathbf{b}, \mathbf{e}, \mathbf{e}' \leftarrow D_\sigma^n$, so small!
- Keys are approximately equal: $\mathbf{gab} + \mathbf{e}'\mathbf{a} \approx \mathbf{gab} + \mathbf{eb}$

LWE key-exchange: noisy Diffie-Hellman

- ring-LWE key-exchange

Public: $\mathbf{g} \in \mathcal{R}$, distribution $\chi = D_\sigma^n$

Alice

$\mathbf{a}, \mathbf{e} \leftarrow_\$ \chi$

$pub_A = \mathbf{ga} + \mathbf{e}$

$S_A = (pub_B)\mathbf{a}$

$\quad = \mathbf{gab} + \mathbf{e}'\mathbf{a}$

$pub_A \longrightarrow$

$\longleftarrow pub_B$

Bob

$\mathbf{b}, \mathbf{e}' \leftarrow_\$ \chi$

$pub_B = \mathbf{gb} + \mathbf{e}'$

$S_B = (pub_A)\mathbf{b}$

$\quad = \mathbf{gab} + \mathbf{eb}$

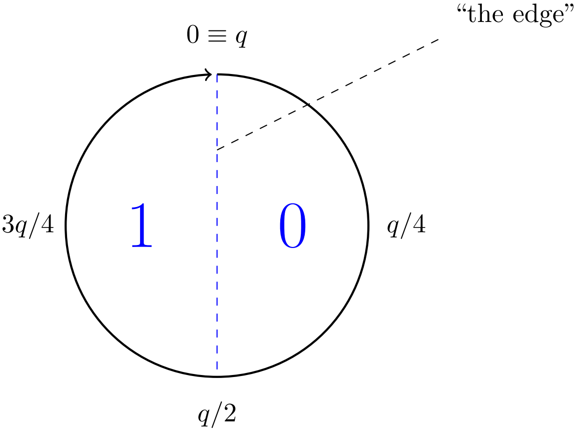
- $\mathbf{a}, \mathbf{b}, \mathbf{e}, \mathbf{e}' \leftarrow D_\sigma^n$, so small!
- Keys are approximately equal: $\mathbf{gab} + \mathbf{e}'\mathbf{a} \approx \mathbf{gab} + \mathbf{eb}$
- Need a way to get shared secret bits

LWE key-exchange: mapping coefficients

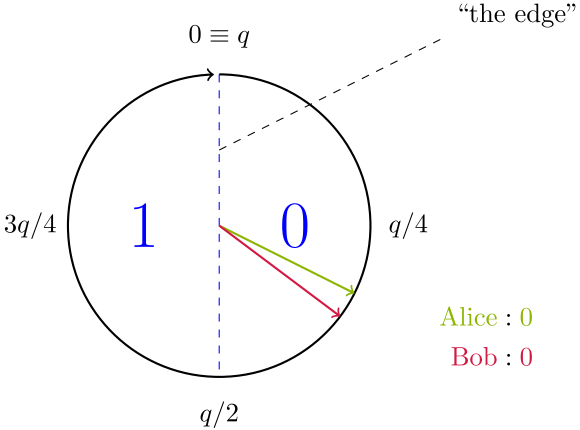
- How to map coefficients to bits
- Alice and Bob obtained close vectors $\mathbf{S_A}, \mathbf{S_B} \in \mathbb{Z}_q^n$

LWE key-exchange: mapping coefficients

- How to map coefficients to bits
- Alice and Bob obtained close vectors $\mathbf{S_A}, \mathbf{S_B} \in \mathbb{Z}_q^n$

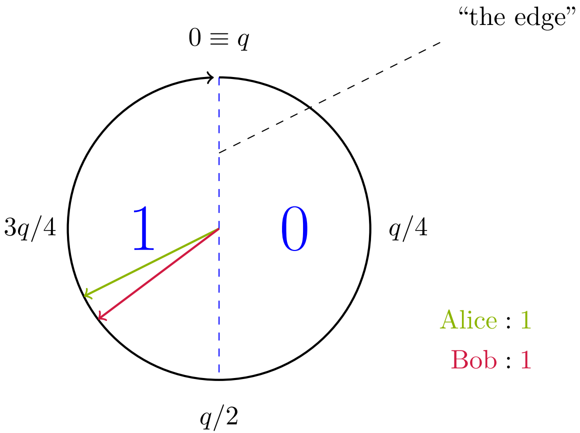


- How to map coefficients to bits
- Alice and Bob obtained close vectors $\mathbf{S_A}, \mathbf{S_B} \in \mathbb{Z}_q^n$



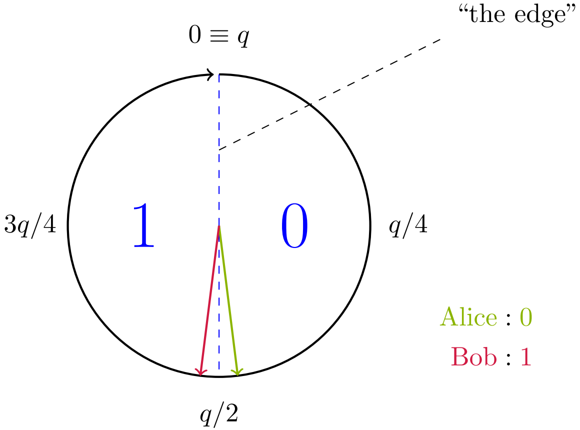
LWE key-exchange: mapping coefficients

- How to map coefficients to bits
- Alice and Bob obtained close vectors $\mathbf{S_A}, \mathbf{S_B} \in \mathbb{Z}_q^n$



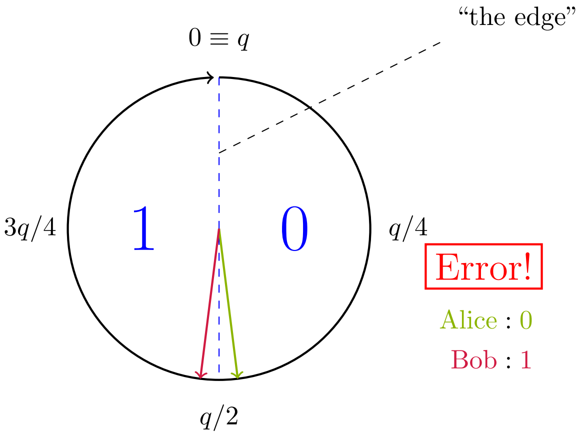
LWE key-exchange: mapping coefficients

- How to map coefficients to bits
- Alice and Bob obtained close vectors $\mathbf{S_A}, \mathbf{S_B} \in \mathbb{Z}_q^n$



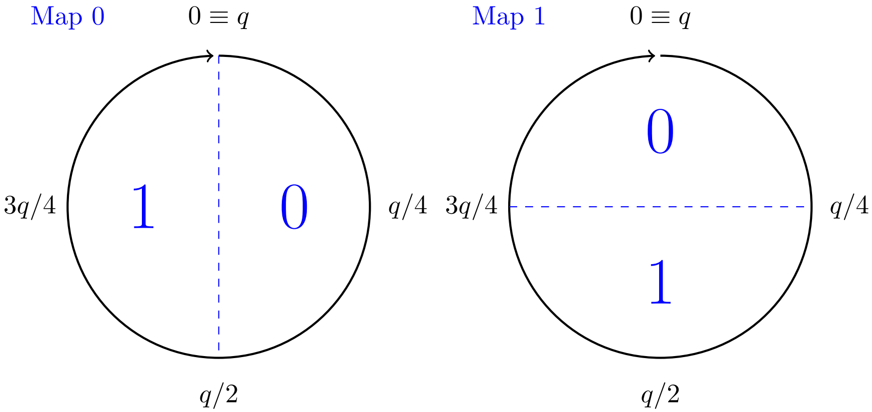
LWE key-exchange: mapping coefficients

- How to map coefficients to bits
- Alice and Bob obtained close vectors $\mathbf{S_A}, \mathbf{S_B} \in \mathbb{Z}_q^n$



LWE key-exchange: reconciliation

- Mapping coefficients by fixed map induces many errors
- Better idea: use two mappings and let Bob decide on which map
- Choose map where $\mathbf{S_B}$ is far from edge



LWE key-exchange: putting it together

- LWE key-exchange with reconciliation

Public: $\mathbf{g} \in \mathcal{R}$, distribution $\chi = D_\sigma^n$

Alice		Bob
$\mathbf{a}, \mathbf{e} \leftarrow_\$ \chi$		
$pub_A = \mathbf{g}\mathbf{a} + \mathbf{e}$	$\xrightarrow{\quad pub_A \quad}$	
		$\mathbf{b}, \mathbf{e}' \leftarrow_\$ \chi$
		$pub_B = \mathbf{g}\mathbf{b} + \mathbf{e}'$
	$\xleftarrow{\quad (pub_B, \mathbf{u}) \quad}$	$\mathbf{u} = \mathrm{reconc}(pub_A \mathbf{b})$
$K = \mathrm{map}(pub_B \mathbf{a}, \mathbf{u})$		$K = \mathrm{map}(pub_A \mathbf{b}, \mathbf{u})$

- Can show that probability of errors is small for q, n, σ well-chosen

LWE key-exchange: putting it together

- LWE key-exchange with reconciliation

Public: ring $\mathcal{R}$, distribution $\chi = D_\sigma^n$

Alice

$\mathbf{g} \leftarrow_\$ \mathcal{R}$

$\mathbf{a}, \mathbf{e} \leftarrow_\$ \chi$

$pub_A = \mathbf{g}\mathbf{a} + \mathbf{e}$

$(\mathbf{g}, pub_A) \longrightarrow$

Bob

$\mathbf{b}, \mathbf{e}' \leftarrow_\$ \chi$

$pub_B = \mathbf{g}\mathbf{b} + \mathbf{e}'$

$\mathbf{u} = \mathrm{reconc}(pub_A \mathbf{b})$

$K = \mathrm{map}(pub_A \mathbf{b}, \mathbf{u})$

$\longleftarrow (pub_B, \mathbf{u})$

Alice

$K = \mathrm{map}(pub_B \mathbf{a}, \mathbf{u})$

- Can show that probability of errors is small for q, n, σ well-chosen
- Several tweaks; e.g. let Alice choose $\mathbf{g}$ (New-Hope)

What about LWE encryption?

- Can do LWE encryption by masking the message into LWE sample:

Public: $\mathbf{g} \in \mathcal{R}$, distribution $\chi = D_\sigma^n$

Alice

$\mathbf{a}, \mathbf{e} \leftarrow_\$ \chi$

$pub_A = \mathbf{g}\mathbf{a} + \mathbf{e}$

$\mathbf{c} - pub_B\mathbf{a}$

$\approx \text{encode}(\mathbf{m})$

Alice → Bob: pub_A

Bob → Alice: $(pub_B, \mathbf{c})$

Bob

$\mathbf{m} \in \{0,1\}^n$

$\mathbf{b}, \mathbf{e}', \mathbf{e}'' \leftarrow \chi$

$pub_B = \mathbf{g}\mathbf{b} + \mathbf{e}'$

$\mathbf{v} = pub_A\mathbf{b} + \mathbf{e}''$

$\mathbf{c} = \mathbf{v} + \text{encode}(\mathbf{m})$

What about LWE encryption?

- Can do LWE encryption by masking the message into LWE sample:

Public: $\mathbf{g} \in \mathcal{R}$, distribution $\chi = D_\sigma^n$

Alice		Bob
$\mathbf{a}, \mathbf{e} \leftarrow_\$ \chi$		$\mathbf{m} \in \{0,1\}^n$
$pub_A = \mathbf{g}\mathbf{a} + \mathbf{e}$	$\xrightarrow{pub_A}$	$\mathbf{b}, \mathbf{e}', \mathbf{e}'' \leftarrow \chi$
		$pub_B = \mathbf{g}\mathbf{b} + \mathbf{e}'$
$\mathbf{c} - pub_B\mathbf{a} \approx \text{encode}(\mathbf{m})$	$\xleftarrow{(pub_B, \mathbf{c})}$	$\mathbf{v} = pub_A\mathbf{b} + \mathbf{e}''$
		$\mathbf{c} = \mathbf{v} + \text{encode}(\mathbf{m})$

- $\mathbf{c} - pub_B\mathbf{a} = \text{encode}(\mathbf{m}) + \mathbf{e}'' + \mathbf{e}\mathbf{b} + \mathbf{e}'\mathbf{a}$

What about LWE encryption?

- Can do LWE encryption by masking the message into LWE sample:

Public: $\mathbf{g} \in \mathcal{R}$, distribution $\chi = D_\sigma^n$

Alice

$\mathbf{a}, \mathbf{e} \leftarrow_\$ \chi$

$pub_A = \mathbf{ga} + \mathbf{e}$

$\mathbf{c} - pub_B\mathbf{a}$

$\approx \text{encode}(\mathbf{m})$

Alice → Bob: pub_A

Bob → Alice: $(pub_B, \mathbf{c})$

Bob

$\mathbf{m} \in \{0,1\}^n$

$\mathbf{b}, \mathbf{e}', \mathbf{e}'' \leftarrow \chi$

$pub_B = \mathbf{gb} + \mathbf{e}'$

$\mathbf{v} = pub_A\mathbf{b} + \mathbf{e}''$

$\mathbf{c} = \mathbf{v} + \text{encode}(\mathbf{m})$

- $\mathbf{c} - pub_B\mathbf{a} = \text{encode}(\mathbf{m}) + \mathbf{e}'' + \mathbf{eb} + \mathbf{e}'\mathbf{a}$
- $\text{encode}(\mathbf{m}) = (q/2)\mathbf{m}$
- Recover $\mathbf{m}$ by some mapping operation (reconciliation)

LWE key-exchange: reaction attacks!

- Can we now replace (EC)DH with LWE?

LWE key-exchange: reaction attacks!

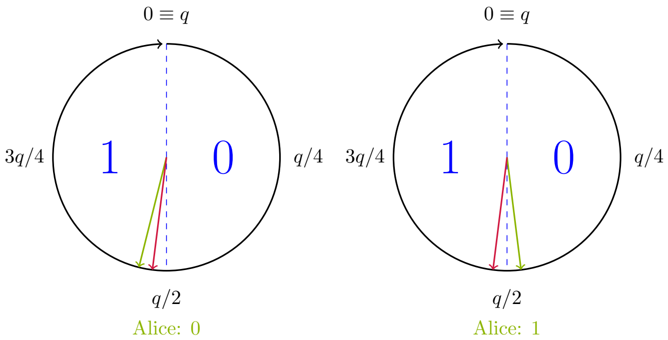
- Can we now replace (EC)DH with LWE? NO!
- Watch out for **reaction attacks**!

LWE key-exchange: reaction attacks!

- Can we now replace (EC)DH with LWE? NO!
- Watch out for **reaction attacks**! or "Evil Bob"

LWE key-exchange: reaction attacks!

- Can we now replace (EC)DH with LWE? NO!
- Watch out for **reaction attacks**! or "Evil Bob"
- Bob can deliberately choose "bad" elements $\mathbf{b}, \mathbf{e}', \mathbf{u}$
- Watches if errors occur during key-exchange/protocol



LWE key-exchange: ephemeral versus cached keys

- The shown LWE key-exchange/encryption must be used **ephemeral**

LWE key-exchange: ephemeral versus cached keys

- The shown LWE key-exchange/encryption must be used **ephemeral**
- To cache keys, most of the LWE schemes use the FO-transform
- There are two possibilities: IND-CPA or IND-CCA

LWE key-exchange: ephemeral versus cached keys

- The shown LWE key-exchange/encryption must be used **ephemeral**
- To cache keys, most of the LWE schemes use the FO-transform
- There are two possibilities: IND-CPA or IND-CCA

IND-C EPHEMERAL A

IND-C CACHE A

LWE key-exchange: ephemeral versus cached keys

- The shown LWE key-exchange/encryption must be used **ephemeral**
- To cache keys, most of the LWE schemes use the FO-transform
- There are two possibilities: IND-CPA or IND-CCA
- Claims of IND-CCA without FO are fishy ("Hilaas Pindakaas")

Lattice-based Signatures

Lattice-based Signatures

- Thijs covered GGH Signatures
- Hash-and-sign signature: requires a trapdoor (e.g. RSA, CVP)
- What about ring-LWE signatures?

Lattice-based Signatures

- Thijs covered GGH Signatures
- Hash-and-sign signature: requires a trapdoor (e.g. RSA, CVP)
- What about ring-LWE signatures?
- Need to slightly adapt the problem
- The Ring-Short-Integer-Solution (ring-SIS), is the problem of:
  - Given $\mathbf{a} \in \mathcal{R}$
  - Target polynomial $\mathbf{t} \in \mathcal{R}$ (can be $\mathbf{0}$)
- Find non-zero $\mathbf{s} \in \mathcal{R}$ s.t. $\mathbf{as} \equiv \mathbf{t} \bmod q$ and $\mathbf{s}$ small
- Also plain versions (plain-SIS)

Hash-and-Sign by SIS

- Public key: $\mathbf{a} \in R$
- Secret key: $\mathbf{s}$: "some way" to solve ring-SIS for any target $\mathbf{b}$

Hash-and-Sign by SIS

- Public key: $\mathbf{a} \in R$
- Secret key: $\mathbf{s}$: "some way" to solve ring-SIS for any target $\mathbf{b}$
- Sign$(\mathbf{s}, \mathbf{m})$: return small $\mathbf{z}$ with $\mathbf{az} \equiv \mathbf{H}(\mathbf{m}) \bmod q$

Hash-and-Sign by SIS

- Public key: $\mathbf{a} \in R$
- Secret key: $\mathbf{s}$: “some way” to solve ring-SIS for any target $\mathbf{b}$
- Sign$(\mathbf{s}, \mathbf{m})$: return small $\mathbf{z}$ with $\mathbf{az} \equiv \mathbf{H}(\mathbf{m}) \bmod q$
- Verify$(\mathbf{z}, \mathbf{m})$: check wether $\mathbf{az} \stackrel{?}{\equiv} \mathbf{H}(\mathbf{m}) \bmod q$ and $\mathbf{z}$ small

Hash-and-Sign by SIS

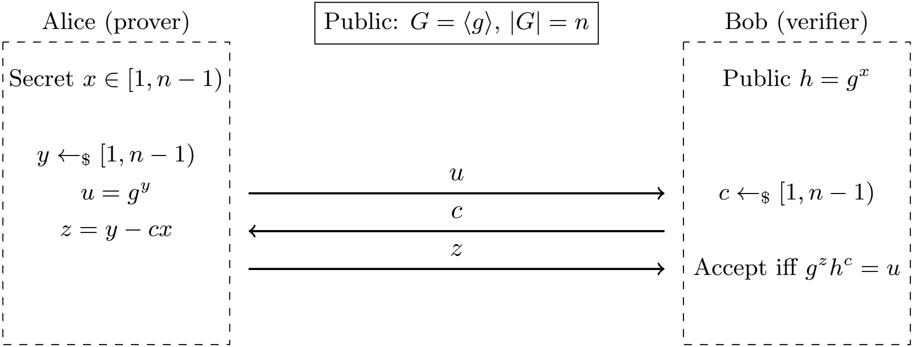
- Public key: $\mathbf{a} \in R$
- Secret key: $\mathbf{s}$: "some way" to solve ring-SIS for any target $\mathbf{b}$
- Sign$(\mathbf{s}, \mathbf{m})$: return small $\mathbf{z}$ with $\mathbf{az} \equiv \mathbf{H}(\mathbf{m}) \bmod q$
- Verify$(\mathbf{z}, \mathbf{m})$: check wether $\mathbf{az} \stackrel{?}{\equiv} \mathbf{H}(\mathbf{m}) \bmod q$ and $\mathbf{z}$ small
- Every signature leaks "some" way of solving SIS
- Long history of "parallelepiped learning attacks"!
- Also applies to GGH, NTRUSign, DRS(submitted to NIST)

LWE/SIS Signatures: the other way

- Hash-and-sign “problematic”, so what else?
- DSA (i.e. DH signatures) is not hash-and-sign...
- So instead, try Fiat-Shamir!

Diffie-Hellman identification protocol

Proof-of-knowledge



Diffie-Hellman identification protocol

Signature scheme (Fiat-Shamir)

Public: $G = \langle g \rangle$, $|G| = n$

Alice (prover)

Secret $x \in [1, n-1)$

Message m

$y \leftarrow_{\$} [1, n-1)$

$u = g^y$

$c = H(u, m)$

$z = y - cx$

Alice → Bob: (c, z)

Bob (verifier)

Public $h = g^x$

$H(g^z h^c, m) \stackrel{?}{=} c$

Diffie-Hellman identification protocol

Signature scheme (Fiat-Shamir)

Public: $G = \langle g \rangle$, $|G| = n$

Alice (prover)

Secret $x \in [1, n-1)$

Message m

$y \leftarrow_{\$} [1, n-1)$

$u = g^y$

$c = H(u, m)$

$z = y - cx$

$\xrightarrow{(c, z)}$

Bob (verifier)

Public $h = g^x$

$H(g^z h^c, m) \stackrel{?}{=} c$

- Let's replace g, x, g^x by $\mathbf{a}$, short $\mathbf{s}$, $\mathbf{t} = \mathbf{as} \bmod q$
- And y, u by $\mathbf{y}$, $\mathbf{u} = \mathbf{ay}$

Fiat-Shamir lattice-based signatures

Mimic DSA with ring-SIS:

Public: $\mathbf{a} \in \mathcal{R}$

Alice (prover)

Secret short $\mathbf{s}$

Message m

$\mathbf{y} \leftarrow_{\$} \mathbb{Z}_q^n$

$\mathbf{u} = \mathbf{a}\mathbf{y} \bmod q$

$\mathbf{c} = H(\mathbf{u}, m)$

$\mathbf{z} = \mathbf{y} + \mathbf{s}\mathbf{c}$

$(\mathbf{c}, \mathbf{z}) \longrightarrow$

Bob (verifier)

Public $\mathbf{t} \equiv \mathbf{a}\mathbf{s} \bmod q$

$H(\mathbf{a}\mathbf{z} - \mathbf{t}\mathbf{c}, m) \stackrel{?}{=} \mathbf{c}$

- $\mathbf{y}$ "hides" the secret part
- H outputs sparse binary polynomials

Fiat-Shamir lattice-based signatures

Mimic DSA with ring-SIS:

Public: $\mathbf{a} \in \mathcal{R}$

Alice (prover)

Secret short $\mathbf{s}$

Message m

$\mathbf{y} \leftarrow_{\$} \mathbb{Z}_q^n$

$\mathbf{u} = \mathbf{a}\mathbf{y} \bmod q$

$\mathbf{c} = H(\mathbf{u}, m)$

$\mathbf{z} = \mathbf{y} + \mathbf{s}\mathbf{c}$

$(\mathbf{c}, \mathbf{z}) \longrightarrow$

Bob (verifier)

Public $\mathbf{t} \equiv \mathbf{a}\mathbf{s} \bmod q$

$H(\mathbf{a}\mathbf{z} - \mathbf{t}\mathbf{c}, m) \stackrel{?}{=} \mathbf{c}$

- $\mathbf{y}$ "hides" the secret part
- H outputs sparse binary polynomials
- But now $\mathbf{u} = \mathbf{a}\mathbf{y}$ not SIS as $\mathbf{y}$ not small $\rightarrow$ use $\mathbf{y} \leftarrow_{\$} D_{\sigma}^{n}$

Fiat-Shamir lattice-based signatures

Mimic DSA with discrete Gaussians:

Public: $\mathbf{a} \in \mathcal{R}$

Alice (prover)

Secret short $\mathbf{s}$

Message m

$\mathbf{y} \leftarrow_{\$} D_\sigma^n$

$\mathbf{u} = \mathbf{a}\mathbf{y} \bmod q$

$\mathbf{c} = H(\mathbf{u}, m)$

$\mathbf{z} = \mathbf{y} + \mathbf{s}\mathbf{c}$

$(\mathbf{c}, \mathbf{z}) \longrightarrow$

Bob (verifier)

Public $\mathbf{t} \equiv \mathbf{a}\mathbf{s} \bmod q$

$H(\mathbf{a}\mathbf{z} - \mathbf{t}\mathbf{c}, m) \stackrel{?}{=} \mathbf{c}$

$\mathbf{z}$ small?

Fiat-Shamir lattice-based signatures

Mimic DSA with discrete Gaussians:

Public: $\mathbf{a} \in \mathcal{R}$

Alice (prover)

Secret short $\mathbf{s}$

Message m

$\mathbf{y} \leftarrow_{\$} D_{\sigma}^{n}$

$\mathbf{u} = \mathbf{a}\mathbf{y} \bmod q$

$\mathbf{c} = H(\mathbf{u}, m)$

$\mathbf{z} = \mathbf{y} + \mathbf{s}\mathbf{c}$

$(\mathbf{c}, \mathbf{z}) \longrightarrow$

Bob (verifier)

Public $\mathbf{t} \equiv \mathbf{a}\mathbf{s} \bmod q$

$H(\mathbf{a}\mathbf{z} - \mathbf{t}\mathbf{c}, m) \stackrel{?}{=} \mathbf{c}$

$\mathbf{z}$ small?

- But now still leaking noisy information on $\mathbf{s}$
- Use Fiat-Shamir with Aborts!

Fiat-Shamir lattice-based signatures

Fiat-Shamir with discrete Gaussians and aborts:

Public: $\mathbf{a} \in \mathcal{R}$

Alice (prover)

Secret short $\mathbf{s}$

Message m

$\mathbf{y} \leftarrow_{\$} D_{\sigma}^{n}$

$\mathbf{u} = \mathbf{ay} \bmod q$

$\mathbf{c} = H(\mathbf{u}, m)$

$\mathbf{z} = \mathbf{y} + \mathbf{sc}$

Abort w.p. $\rho(\mathbf{z})$

$(\mathbf{c}, \mathbf{z}) \longrightarrow$

Bob (verifier)

Public $\mathbf{t} \equiv \mathbf{as} \bmod q$

$H(\mathbf{az} - \mathbf{tc}, m) \stackrel{?}{=} \mathbf{c}$

$\mathbf{z}$ small?

- Signatures statistically independent of $\mathbf{s}$, i.e. $\mathbf{z} \sim D_{\sigma}^{n}$

Fiat-Shamir lattice-based signatures

Fiat-Shamir with discrete Gaussians and aborts:

Public: $\mathbf{a} \in \mathcal{R}$

Alice (prover)

Secret short $\mathbf{s}$

Message m

$\mathbf{y} \leftarrow_{\$} D_\sigma^n$

$\mathbf{u} = \mathbf{ay} \bmod q$

$\mathbf{c} = H(\mathbf{u}, m)$

$\mathbf{z} = \mathbf{y} + \mathbf{sc}$

Abort w.p. $\rho(\mathbf{z})$

$(\mathbf{c}, \mathbf{z}) \longrightarrow$

Bob (verifier)

Public $\mathbf{t} \equiv \mathbf{as} \bmod q$

$H(\mathbf{az} - \mathbf{tc}, m) \stackrel{?}{=} \mathbf{c}$

$\mathbf{z}$ small?

- Signatures statistically independent of $\mathbf{s}$, i.e. $\mathbf{z} \sim D_\sigma^n$
- Several optimizations (i.e. BLISS)

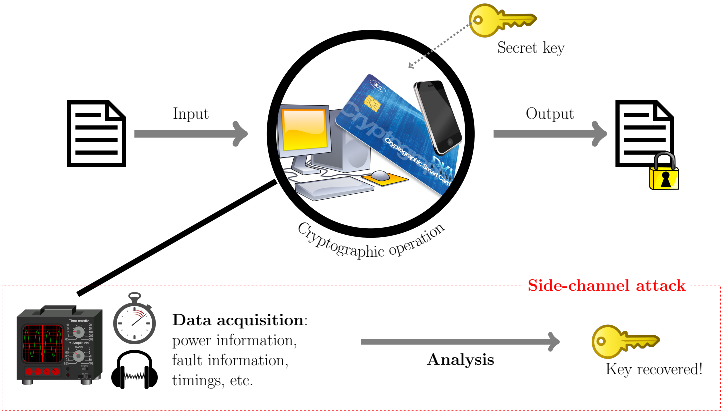
Implementation Issues

Lattice-based signatures: side-channel attacks!

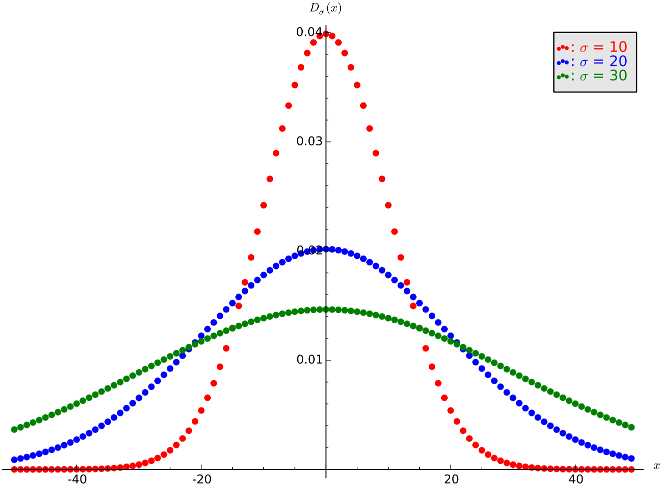
- Can we now replace (EC)DSA/RSA with e.g. BLISS?

Lattice-based signatures: side-channel attacks!

- Can we now replace (EC)DSA/RSA with e.g. BLISS? *Kinda, it depends...*
- Watch out for **side-channel attacks**!



- Signature $\mathbf{z} = \mathbf{y} + \mathbf{sc}, \mathbf{c}$
- $\mathbf{y} \leftarrow_{\$} D_\sigma^n$ looks nice and short on paper...



Side-channel attacks on lattice-based signatures

- Signature $\mathbf{z} = \mathbf{y} + \mathbf{sc}, \mathbf{c}$
- $\mathbf{y} \leftarrow_{\$} D_\sigma$ looks nice and short on paper...
- ...but very nasty in code: about 30% of the running time!
- Good target for a side-channel attack

Side-channel attacks on lattice-based signatures

- Signature $\mathbf{z} = \mathbf{y} + \mathbf{sc}, \mathbf{c}$
- $\mathbf{y} \leftarrow_\$ D_\sigma$ looks nice and short on paper...
- ...but very nasty in code: about 30% of the running time!
- Good target for a side-channel attack
- In 2016, we showed how to break BLISS with cache-attacks
- From noisy information on $\mathbf{y}$, construct an "easy lattice problem"
- All discrete Gaussian samplers have vulnerabilities

Side-channel attacks on lattice-based signatures

- Signature $\mathbf{z} = \mathbf{y} + \mathbf{s}\mathbf{c}, \mathbf{c}$
- $\mathbf{y} \leftarrow_{\$} D_\sigma$ looks nice and short on paper...
- ...but very nasty in code: about 30% of the running time!
- Good target for a side-channel attack
- In 2016, we showed how to break BLISS with cache-attacks
- From noisy information on $\mathbf{y}$, construct an "easy lattice problem"
- All discrete Gaussian samplers have vulnerabilities
- Possibly the reason why BLISS was not submitted to NIST

Lattice-based signatures 2.0

- Discrete Gaussian sampling problematic
- Use small uniform noise instead?

Lattice-based signatures 2.0

- Discrete Gaussian sampling problematic
- Use small uniform noise instead?
- Possible, but signatures become larger
- Dilithium and TESLA still reasonable size

Lattice-based signatures 2.0

- Discrete Gaussian sampling problematic
- Use small uniform noise instead?
- Possible, but signatures become larger
- Dilithium and TESLA still reasonable size
- Additionally remove sampling all-together, i.e. deterministic schemes

Lattice-based signatures 2.0

- Discrete Gaussian sampling problematic
- Use small uniform noise instead?
- Possible, but signatures become larger
- Dilithium and TESLA still reasonable size
- Additionally remove sampling all-together, i.e. deterministic schemes
- In 2018, we showed several differential fault attacks
- TESLA is now randomized again

Lattice-based cryptography: the takeaways

- For key-exchange/encryption, several good options
- Many design choices! (ring-)LWE, NTRU, LWR; IND-CPA/CCA.

Lattice-based cryptography: the takeaways

- For key-exchange/encryption, several good options
- Many design choices! (ring-)LWE, NTRU, LWR; IND-CPA/CCA.
- For digital signatures, sampling randomness can be problematic.
- Watch out for side-channel attacks, i.e. **write constant-time code**!
- Many ongoing improvements to signature schemes and samplers

Lattice-based cryptography: the takeaways

- For key-exchange/encryption, several good options
- Many design choices! (ring-)LWE, NTRU, LWR; IND-CPA/CCA.
- For digital signatures, sampling randomness can be problematic.
- Watch out for side-channel attacks, i.e. **write constant-time code**!
- Many ongoing improvements to signature schemes and samplers

Questions?

Further reading

LWE and Ring-LWE

- Goldreich, Goldwasser, and Halevi, "Public-Key Cryptosystems from Lattice Reduction Problems", 1997
- Regev, "On lattices, learning with errors, random linear codes, and cryptography", 2009
- Lyubashevsky, Peikert, and Regev, "On Ideal Lattices and Learning with Errors over Rings", 2010
- Silverman, "Lattices, cryptography, and the NTRU public key cryptosystem", 2000
- Lyubashevsky, Peikert, and Regev, "A Toolkit for Ring-LWE Cryptography", 2013

Further reading

Lattice-based key-exchange/encryption

- Ding, "A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem", 2012
- Bos, Costello, Naehrig, and Stebila, "Post-quantum key exchange for the TLS protocol from the ring learning with errors problem", 2014
- Alkim, Ducas, Pöppelmann, and Schwabe, "Post-quantum Key Exchange - A New Hope", 2016
- Bos, Costello, Ducas, Mironov, Naehrig, Nikolaenko, Raghunathan, and Stebila, "Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE", 2016

Further reading

More Lattice-based key-exchange/encryption

- Bernstein, Chuengsatiansup, Lange, and Vredendaal, "NTRU Prime: Reducing Attack Surface at Low Cost", 2017
- Bos, Ducas, Kiltz, Lepoint, Lyubashevsky, Schanck, Schwabe, Seiler, and Stehlé, "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM", 2018
- Baan, Bhattacharya, Fluhrer, García-Morchón, Laarhoven, Rietman, Saarinen, Tolhuizen, and Zhang, "Round5: Compact and Fast Post-Quantum Public-Key Encryption", 2019
- Bos, Costello, Ducas, Mironov, Naehrig, Nikolaenko, Raghunathan, and Stebila, "Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE", 2016

Further reading

Reaction attacks and attacks on lattice cryptography designs

- Fluhrer, "Cryptanalysis of ring-LWE based key exchange with key share reuse", 2016
- Bernstein, Groot Bruinderink, Lange, and Panny, "HILA5 Pindakaas: On the CCA Security of Lattice-Based Encryption with Error Correction", 2018
- Cramer, Ducas, Peikert, and Regev, "Recovering Short Generators of Principal Ideals in Cyclotomic Rings", 2016
- Bauch, Bernstein, Valence, Lange, and Vredendaal, "Short Generators Without Quantum Computers: The Case of Multiquadratics", 2017

Further reading

Lattice-based signatures

- Lyubashevsky, "Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures", 2009
- Ducas, Durmus, Lepoint, and Lyubashevsky, "Lattice Signatures and Bimodal Gaussians", 2013
- Ducas, Lepoint, Lyubashevsky, Schwabe, Seiler, and Stehlé, "CRYSTALS - Dilithium: Digital Signatures from Module Lattices", 2017
- Alkim, Bindel, Buchmann, and Dagdelen, "TESLA: Tightly-Secure Efficient Signatures from Standard Lattices", 2015

Further reading

Learning attacks on lattice-based signatures

- Nguyen and Regev, “Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures”, 2006
- Ducas and Nguyen, “Learning a Zonotope and More: Cryptanalysis of NTRUSign Countermeasures”, 2012
- Yu and Ducas, “Learning Strikes Again: The Case of the DRS Signature Scheme”, 2018

Further reading

Side-channel attacks on lattice-based signatures

- Groot Bruinderink, Hülsing, Lange, and Yarom, "Flush, Gauss, and Reload - A Cache Attack on the BLISS Lattice-Based Signature Scheme", 2016
- Pessl, Groot Bruinderink, and Yarom, "To BLISS-B or not to be: Attacking strongSwan's Implementation of Post-Quantum Signatures", 2017
- Espitau, Fouque, Gérard, and Tibouchi, "Side-Channel Attacks on BLISS Lattice-Based Signatures: Exploiting Branch Tracing against strongSwan and Electromagnetic Emanations in Microcontrollers", 2017
- Groot Bruinderink and Pessl, "Differential Fault Attacks on Deterministic Lattice Signatures", 2018