Lattice-based signature schemes

Adeline Roux-Langlois

Univ Rennes, CNRS, IRISA

June 13, 2019

Lattice-based cryptography

Post-quantum cryptography

- Quantum computers?
  1. NIST competition: basic primitives such as encryption scheme, signature scheme and key exchange mechanism,
  2. Advanced/New functionalities: for many applications.

Lattice-based cryptography

Post-quantum cryptography

- Quantum computers?
  1. NIST competition: basic primitives such as encryption scheme, signature scheme and key exchange mechanism,
  2. Advanced/New functionalities: for many applications.

Lattice-based cryptography

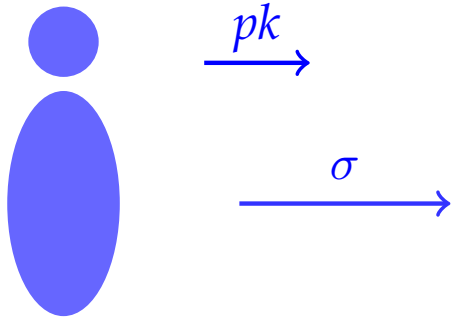
- Efficient (asymptotically),
- **Proven security** based on hard problems on lattices,
- Likely to resist attacks from quantum computers,
- From basic to very advanced primitives:
  - Public key encryption and signature scheme (practical) ...
  - Advanced signature / encryption scheme (IBE, ABE, ..),
  - Fully homomorphic encryption.

Topic today: public-key signature scheme

Generates pair of keys pk, sk

keeps sk

$\sigma = \text{Sign}(sk, M)$



Given pk anyone can verify

$\text{Verify}(pk, M, \sigma)$

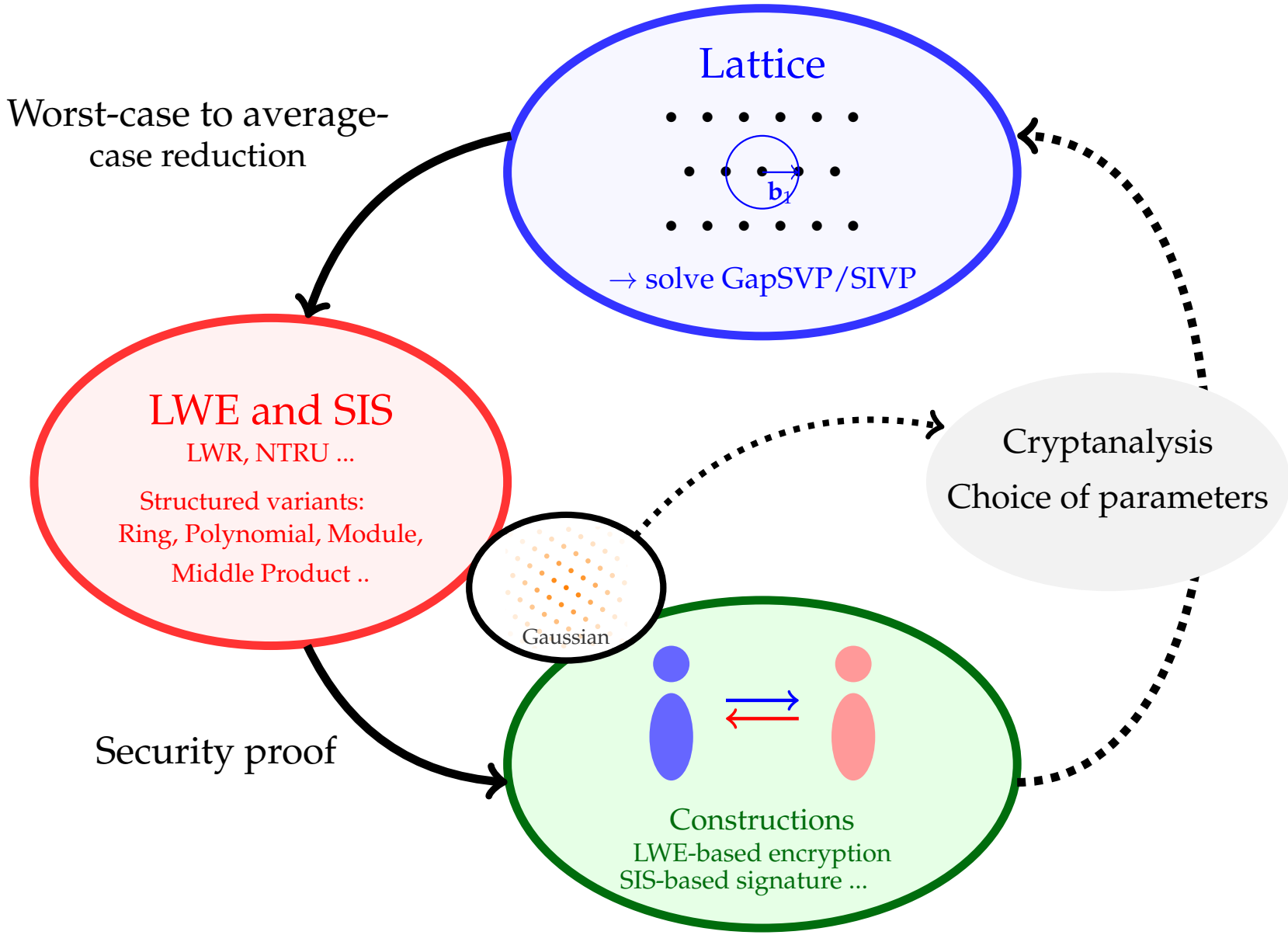
Two requirements: Correctness and Security

- Verification of valid signatures returns 1
- Unforgeability: not possible to forge a valid signature without the secret key

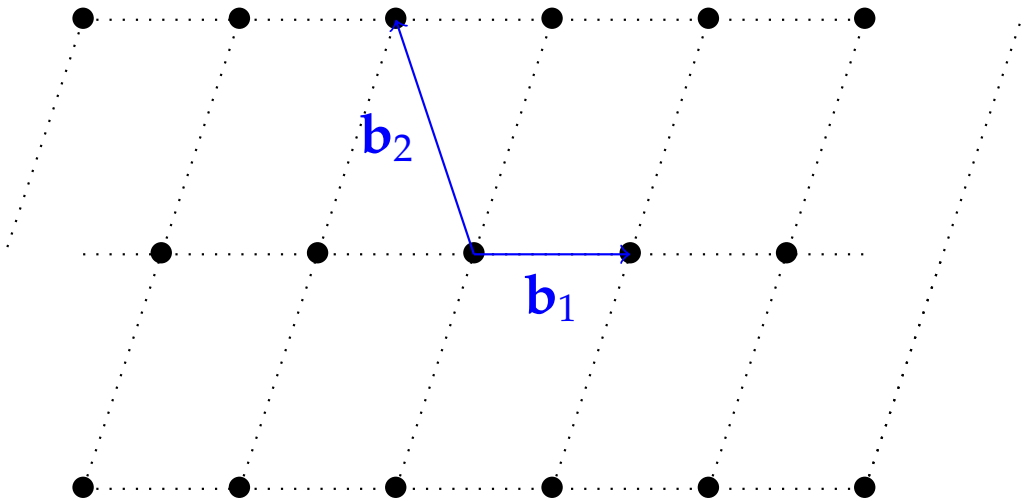
Lattice-based cryptography
- Hardness of lattice problems
- SIS and its trapdoor
- Signature scheme on lattices

NIST competition

Some recent results
- Implementing an efficient and modular trapdoor
- Blind signature scheme


Lattice
b_1
→ solve GapSVP/SIVP
Worst-case to average-case reduction
LWE and SIS
LWR, NTRU ...
Structured variants:
Ring, Polynomial, Module,
Middle Product ..
Gaussian
Cryptanalysis
Choice of parameters
Security proof
Constructions
LWE-based encryption
SIS-based signature ...

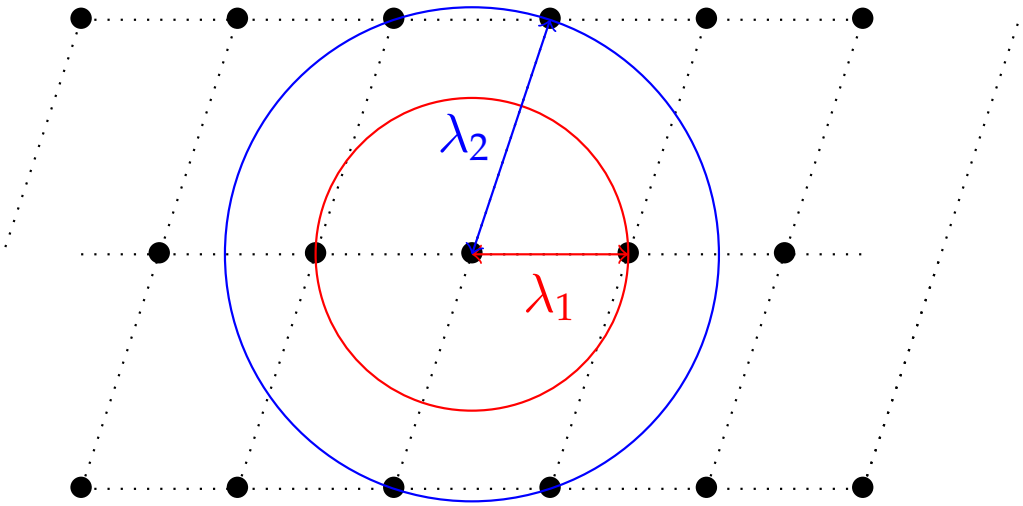
Lattices and problems



Lattice

$\mathcal{L}(\mathbf{B}) = \{\sum_{1=i}^{n} a_i \mathbf{b}_i, a_i \in \mathbb{Z}\}$, where the $(\mathbf{b}_i)_{1 \leq i \leq n}$'s, linearly independent vectors, are a basis of $\mathcal{L}(\mathbf{B})$.

Lattices and problems



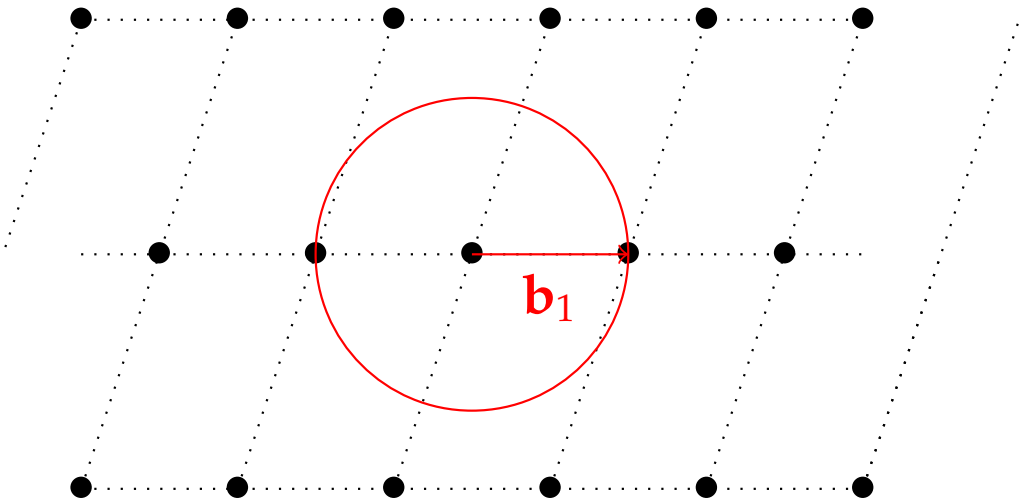
Definitions:

- 1st minimum;
- 2nd minimum.

Lattice

$\mathcal{L}(\mathbf{B}) = \{\sum_{1=i}^{n} a_i \mathbf{b}_i, a_i \in \mathbb{Z}\}$, where the $(\mathbf{b}_i)_{1 \leq i \leq n}$'s, linearly independent vectors, are a basis of $\mathcal{L}(\mathbf{B})$.

Lattices and problems



Definitions:

- 1st minimum;
- 2nd minimum.

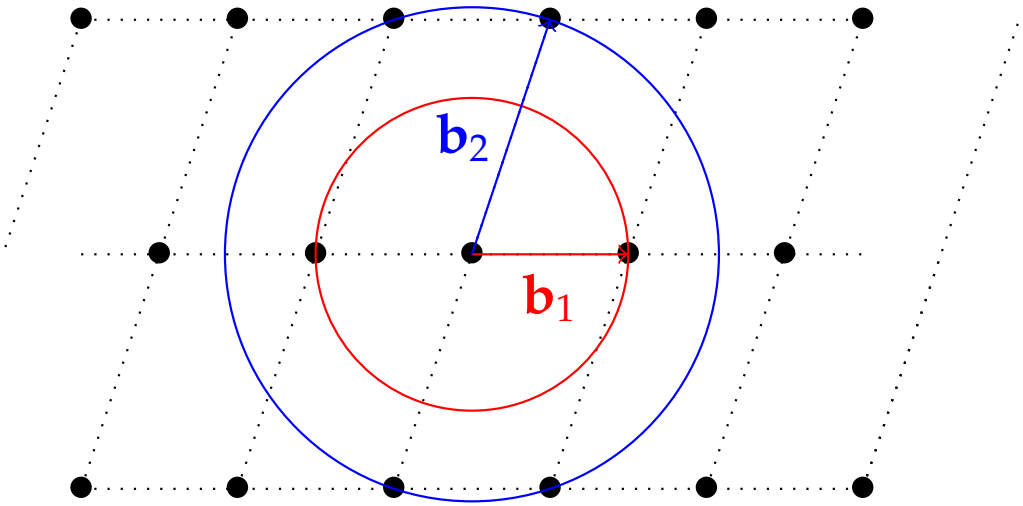
Problems :

- Shortest Vector Pbm. (computational or decisional version)

Lattice

$\mathcal{L}(\mathbf{B}) = \{\sum_{1=i}^{n} a_i \mathbf{b}_i, a_i \in \mathbb{Z}\}$, where the $(\mathbf{b}_i)_{1 \leq i \leq n}$'s, linearly independent vectors, are a basis of $\mathcal{L}(\mathbf{B})$.

Lattices and problems



Definitions:

- 1st minimum;
- 2nd minimum.

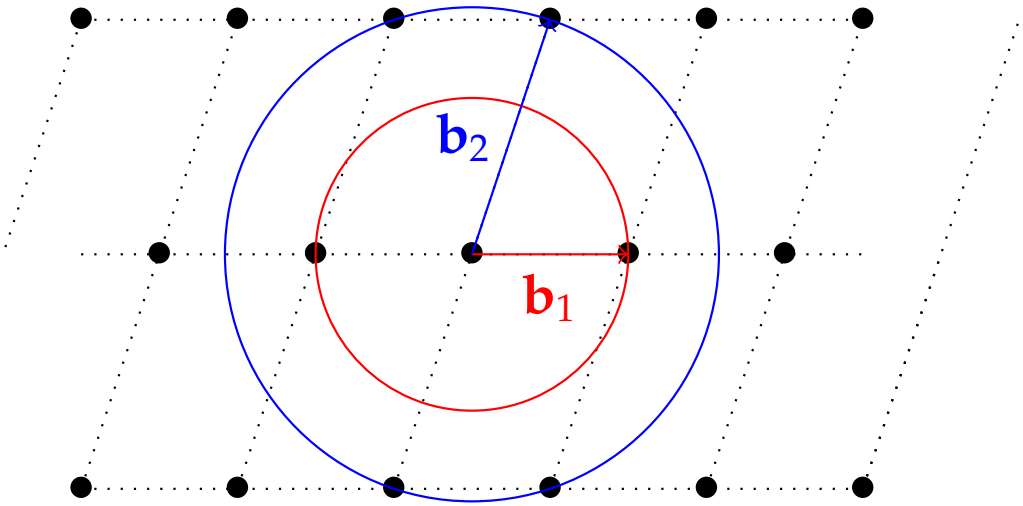
Problems :

- Shortest Vector Pbm. (computational or decisional version)
- Shortest Independent Vectors Pbm.

Lattice

$\mathcal{L}(\mathbf{B}) = \{\sum_{1=i}^{n} a_i \mathbf{b}_i, a_i \in \mathbb{Z}\}$, where the $(\mathbf{b}_i)_{1 \leq i \leq n}$'s, linearly independent vectors, are a basis of $\mathcal{L}(\mathbf{B})$.

Lattices and problems



Definitions:

- 1st minimum;
- 2nd minimum.

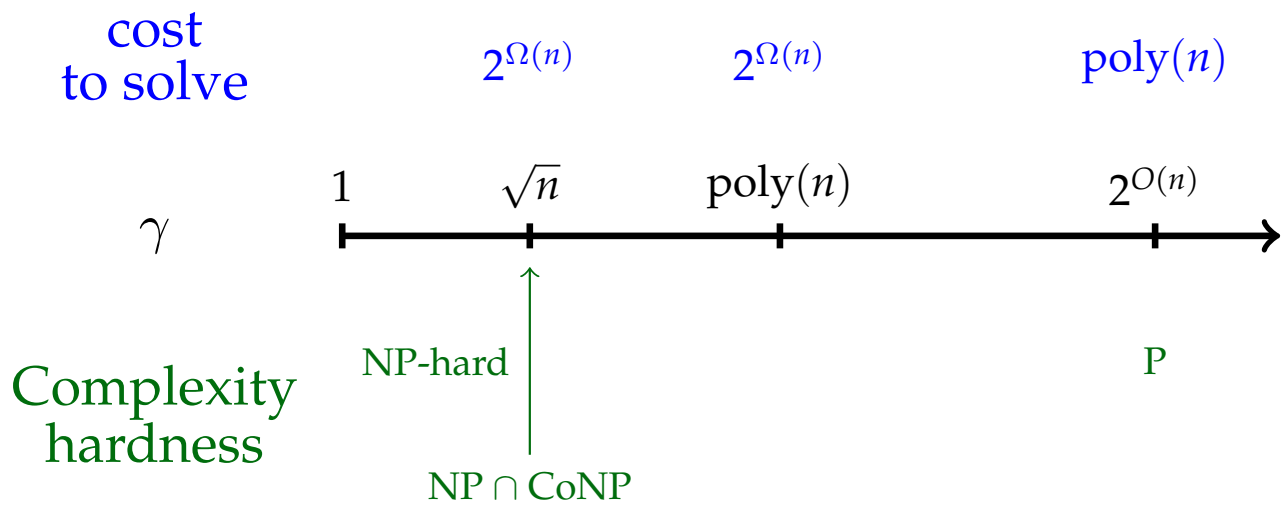
Problems :

- Shortest Vector Pbm. (computational or decisional version)
- Shortest Independent Vectors Pbm.
- Approximation factor: γ.

Lattice

$\mathcal{L}(\mathbf{B}) = \{\sum_{1=i}^{n} a_i \mathbf{b}_i, a_i \in \mathbb{Z}\}$, where the $(\mathbf{b}_i)_{1 \leq i \leq n}$'s, linearly independent vectors, are a basis of $\mathcal{L}(\mathbf{B})$.

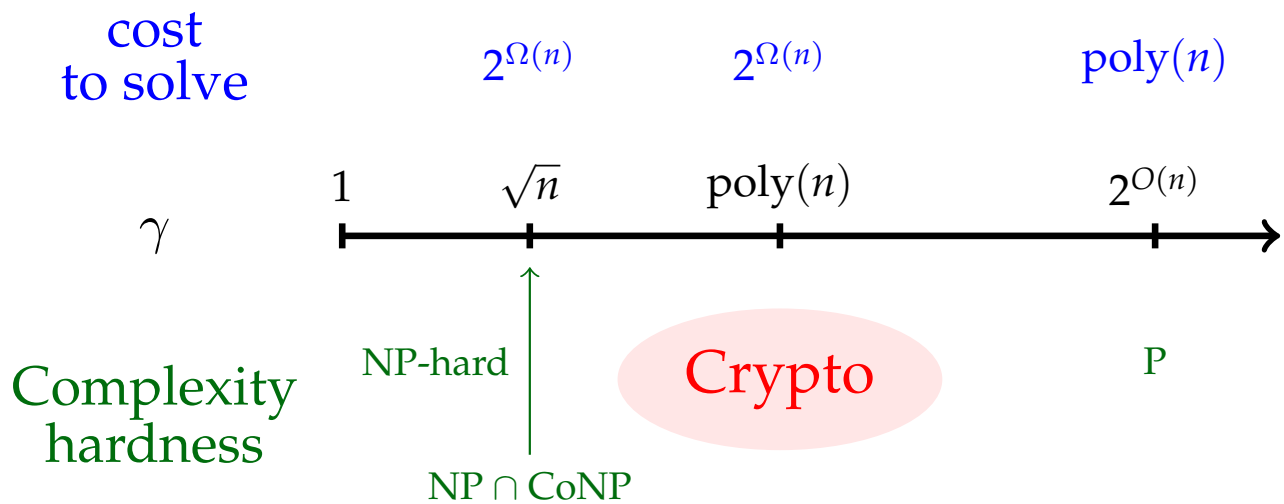
Hardness of SVP_γ



Conjecture

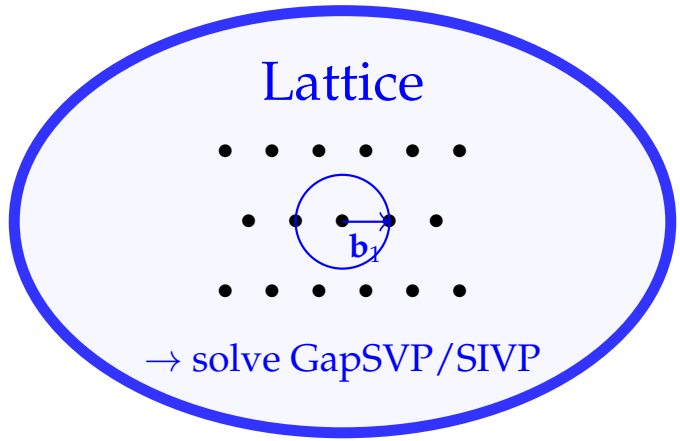
There is no polynomial time algorithm that approximates those lattice problems to within polynomial factors.

Hardness of SVP_γ



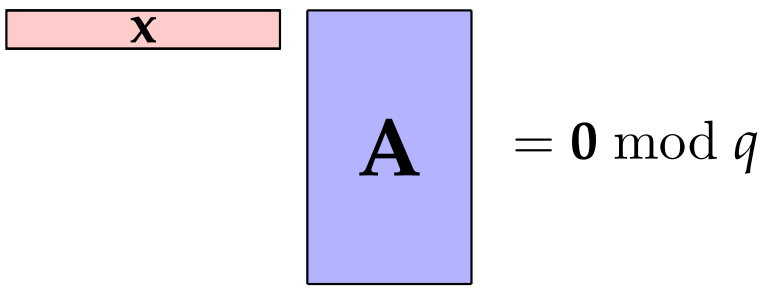
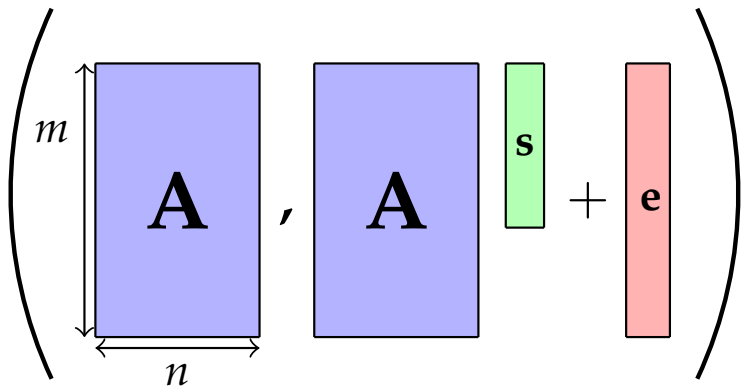
Conjecture

There is no polynomial time algorithm that approximates those lattice problems to within polynomial factors.


Lattice
$\mathbf{b}_1$
$\rightarrow$ solve GapSVP/SIVP

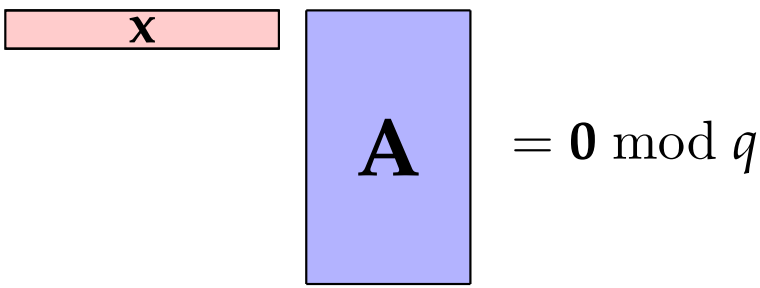
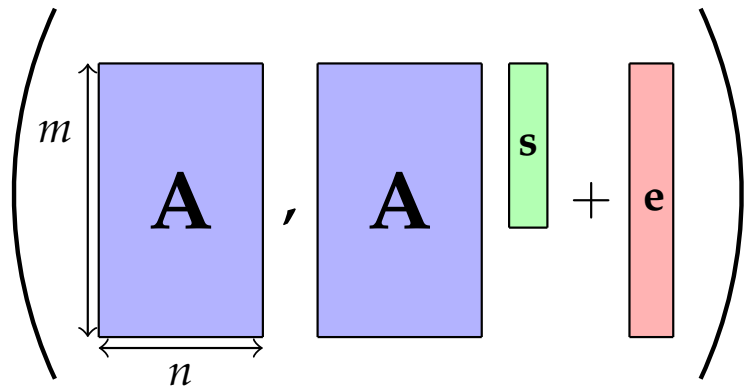
Foundamental problems to build cryptography

Parameters: dimension n, $m \geq n$, moduli q.
For $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m \times n})$:

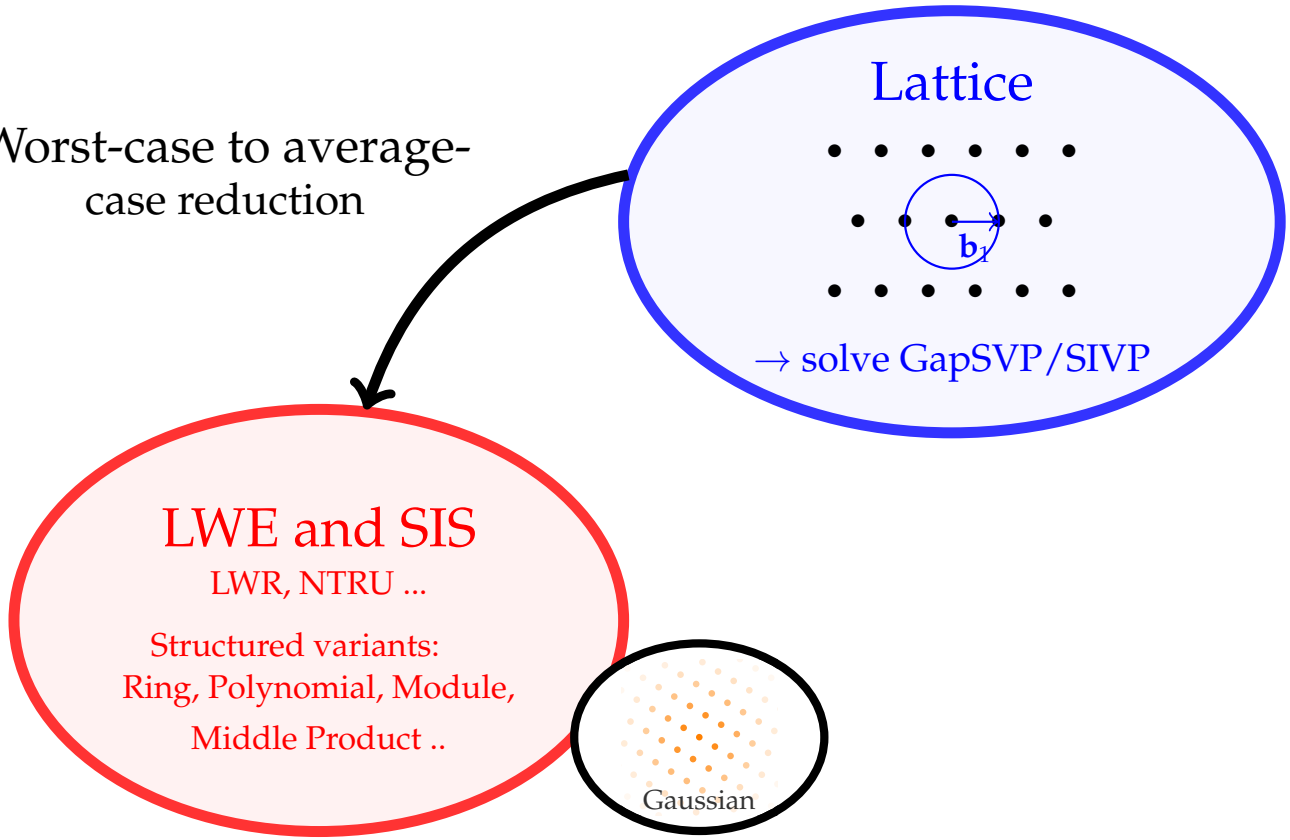
$\mathbf{SIS}_\beta$	$\mathbf{LWE}_\alpha$
	
	$\mathbf{s} \leftarrow U(\mathbb{Z}_q^n)$, $\mathbf{e}$ a small error $\approx \alpha q$.
Goal: Given $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m \times n})$, find $\mathbf{x}$ s.t. $0 < \|\mathbf{x}\| \leq \beta$.	Goal: Given $(\mathbf{A}, \mathbf{A}\,\mathbf{s} + \mathbf{e})$, find $\mathbf{s}$.
[Ajtai 96, GPV 08]	[Regev 05]

Foundamental problems to build cryptography

Parameters: dimension n, $m \geq n$, moduli q.
For $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m \times n})$:

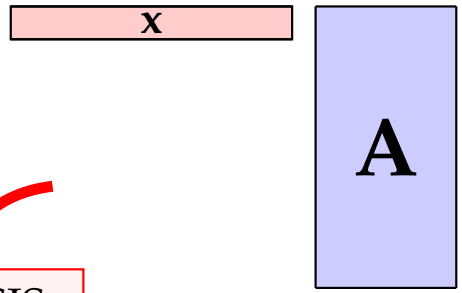
$\mathbf{SIS}_\beta$	$\mathbf{LWE}_\alpha$
	 $\mathbf{s} \leftarrow U(\mathbb{Z}_q^n)$, $\mathbf{e}$ a small error $\approx \alpha q$.
Find a small vector in $\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{x}^T \mathbf{A} = 0 \bmod q\}$	Solve BDD in $\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y} = \mathbf{A}\mathbf{s} \bmod q$ for some $\mathbf{s} \in \mathbb{Z}^n\}$
[Ajtai 96, GPV 08]	[Regev 05]

Worst-case to average-case reduction



Trapdoor for SIS

- TrapGen $\rightsquigarrow (\mathbf{A}, \mathbf{T_A})$ such that $\mathbf{T_A}$ allows to find short $\mathbf{x}$ ('s)



$= \mathbf{0} \bmod q$

With $\mathbf{T_A}$, we can solve (I)SIS.

Computing $\mathbf{T_A}$ *given* $\mathbf{A}$ is hard,
Constructing $\mathbf{A}$ *and* $\mathbf{T_A}$ is easy.

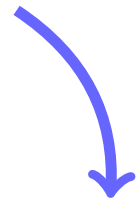
- $\mathbf{T_A}$ is a short basis of $\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m | \mathbf{x}^T \mathbf{A} = 0 \bmod q\}$
- With $\mathbf{T_A}$, we can sample short vectors in $\Lambda_q^\perp(\mathbf{A})$.

- In a public key scheme:
  - public key: $\mathbf{A}$
  - secret key: $\mathbf{T_A}$

Signature schemes

Two (main) frameworks to build lattice-based signature schemes:

- From identification scheme using Fiat Shamir transformation,
  For example: Dilithium and qTesla.
- Using trapdoors,
  For example: Falcon.



- Exists in Standard model,
- Trapdoor are costly!

- Very efficient,
- always in the Random Oracle Model.

Trapdoor-based signature scheme [GPV 2008]

GPV signature scheme

- Key generation:
  - $pk = \mathbf{A}$
  - $sk = \mathbf{T}$
- To sign a message M:
  - use $\mathbf{T}$ to solve ISIS: find small $\mathbf{x}$ such that $\mathbf{x}^T \mathbf{A} = H(M) \bmod q$.
- To verify a signature $\mathbf{x}$ given M:
  - check $\mathbf{x}^T \mathbf{A} = H(M) \bmod q$ and $\mathbf{x}$ small.

- Proven secure in the Random Oracle Model.

In the standard model

- add $(\mathbf{A}_i)_i$ to the public key, and build $\mathbf{A}_M$ from $\mathbf{A}$, $(\mathbf{A}_i)_i$ and M,
- use $\mathbf{T}_\mathbf{A}$ to solve SIS: find small $\mathbf{x}$ such that $\mathbf{x}^T \mathbf{A}_M = 0 \bmod q$,
- knowing a trapdoor for $\mathbf{A} \Rightarrow$ knowing a trapdoor for $\mathbf{A}_M$.

Fiat-Shamir constructions

Prover $_{sk}$		Verifier $_{pk}$
	$\xrightarrow{CMT}$	
	$\xleftarrow{CH}$	$CH \leftarrow U(\{0,1\}^\lambda)$
	$\xrightarrow{RSP}$	Given $CMT\|\|CH\|\|RSP$
		Accept or not

Fiat-Shamir transform

Given a hash function H

For a message M, $CH \leftarrow H(CMT, M)$, and $\sigma = (CMT, RSP)$.

Fiat-Shamir constructions [Lyu 2012]

$\mathbf{S} \in \{-1,0,1\}^{k\times m}$ uniform, $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ uniform,
$\mathbf{T} = \mathbf{S}\,\mathbf{A} \bmod q$.

Prover $_{sk=\mathbf{S}}$		Verifier $_{pk=(\mathbf{A},\mathbf{T})}$
$\mathbf{y} \in \mathbb{Z}^m$ gaussian $\mathbf{u} = \mathbf{y}^T\mathbf{A} \bmod q$	$\overset{CMT=\mathbf{u}}{\rightarrow}$	
	$\overset{CH=\mathbf{c}^T}{\leftarrow}$	$CH \leftarrow U(\{0,1\}^\lambda)$
$\mathbf{z}^T = \mathbf{c}^T\mathbf{S} + \mathbf{y}^T$ Accept $\mathbf{z}$ with proba $P(z)$	$\overset{RSP=\mathbf{z}}{\rightarrow}$	Given $\mathbf{u}, \mathbf{c}, \mathbf{z}$ Accept if $\mathbf{z}$ has small norm and $\mathbf{u} = \mathbf{z}^T\mathbf{A} - \mathbf{c}^T\mathbf{T} \bmod q$

Fiat-Shamir transform

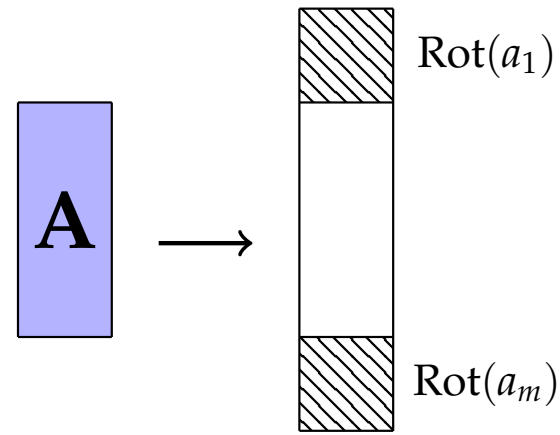
Given a hash function $H : \{0,1\}^* \to \{-1,0,1\}^k$
For a message M, $CH = H(\mathbf{y}^T\mathbf{A} \bmod q, M)$, and $\sigma = (\mathbf{z}, \mathbf{c})$.

From SIS/LWE to structured variants

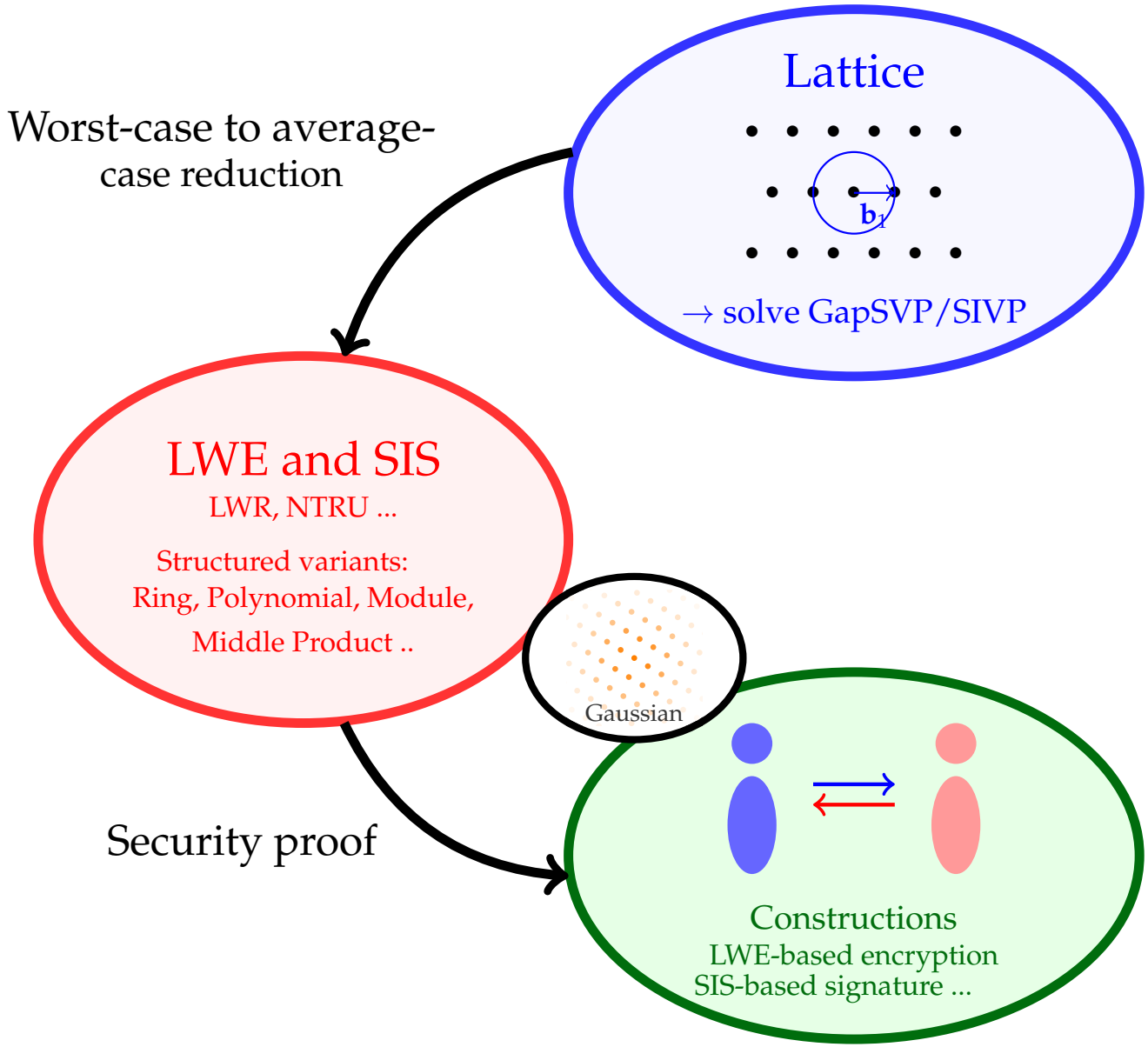
- **Problem:** constructions based on SIS/LWE enjoy a nice guaranty of security but are too costly in practice.

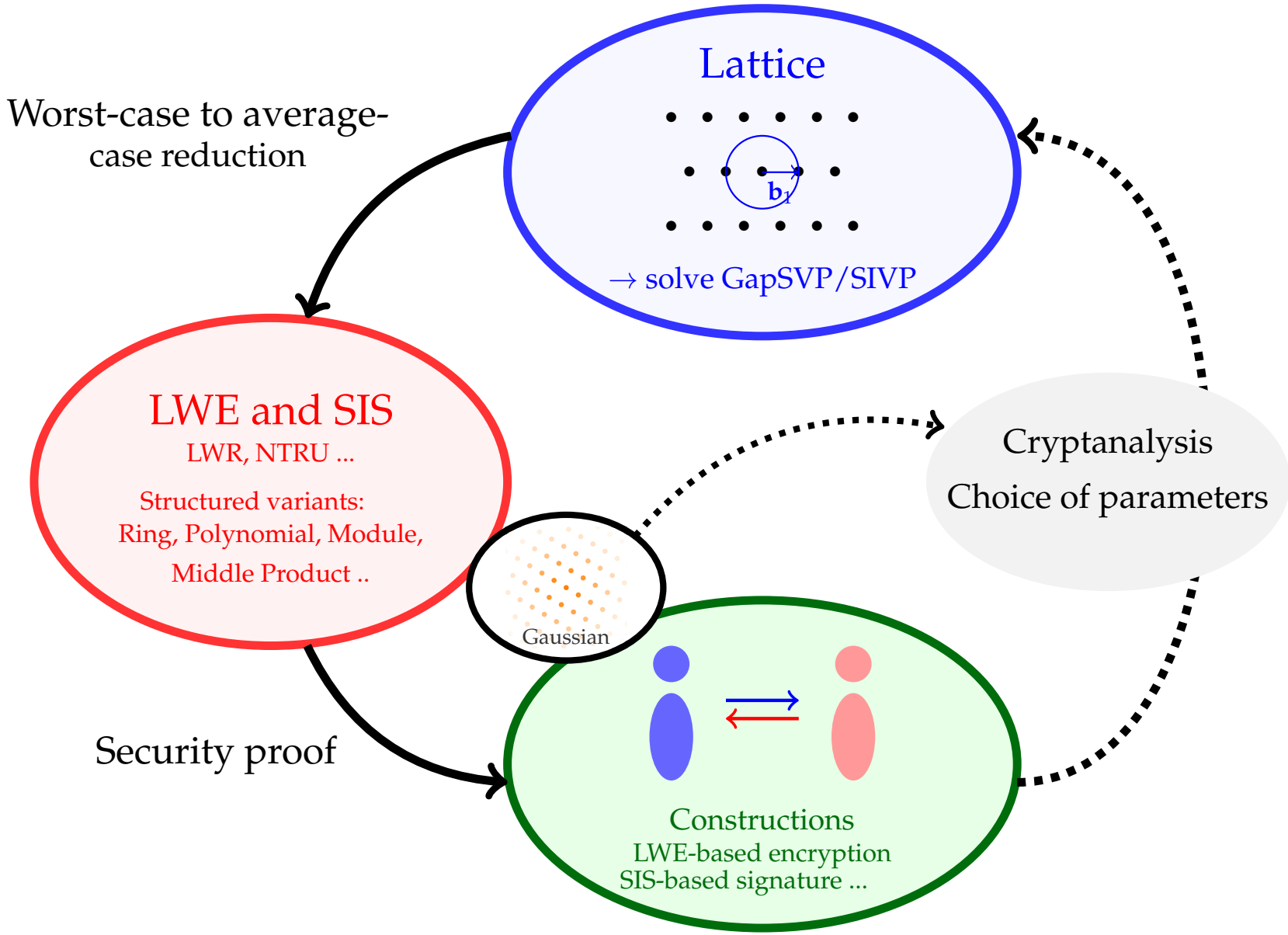
$\rightarrow$ replace $\mathbb{Z}^n$ by a Ring, for example $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ ($n = 2^k$).

- Ring variants since 2006:



- Structured $\mathbf{A} \in \mathbb{Z}_q^{m \cdot n \times n}$ represented by $m \cdot n$ elements,
- Product with matrix/vector more efficient,
- Hardness of Ring-SIS, [Lyubashevsky and Micciancio 06] and [Peikert and Rosen 06]
- Hardness of Ring-LWE [Lyubashevsky, Peikert and Regev 10].


Worst-case to average-case reduction
Lattice
$\mathbf{b}_1$
$\rightarrow$ solve GapSVP/SIVP
LWE and SIS
LWR, NTRU ...
Structured variants:
Ring, Polynomial, Module,
Middle Product ..
Gaussian
Security proof
Constructions
LWE-based encryption
SIS-based signature ...


Lattice
$\mathbf{b}_1$
$\rightarrow$ solve GapSVP/SIVP
Worst-case to average-case reduction
LWE and SIS
LWR, NTRU ...
Structured variants:
Ring, Polynomial, Module,
Middle Product ..
Gaussian
Cryptanalysis
Choice of parameters
Security proof
Constructions
LWE-based encryption
SIS-based signature ...

NIST Competition

Goal: *"NIST has initiated a process to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms."*

- November 2017: candidates submissions,
- December 2017: Round 1 announced (69 submissions),
- April 2018: First PQC Standardization Conference,
- January 2019: Second Round Candidates announced (26 algorithms),
- August, 2019: Second PQC Standardization Conference,
- 2020/2021: Round 3 begins or select algorithms,
- 2022/2024: Draft Standards Available.

NIST Competition

All constructions:

	1st round	2nd round
Lattices	26	12
Code-based	19	7
Multivariate	9	4
Symmetric/Hash-based	3	1
Other	7	2
Total	64	26

Signature schemes:

	1st round	2nd round
Lattices	5	3
Code-based	2	
Multivariate	7	4
Symmetric/Hash-based	3	1
Other	2	1
Total	19	9

What about lattice-based signatures?

- 1st Round: Dilithium, qTesla, Falcon, DRS, NTRUsign.

- 2nd Round:
  - Dilithium: Fiat-Shamir construction, based on Module-LWE,
  - qTesla: Fiat-Shamir construction, based on Ring-LWE,
  - Falcon: Trapdoor-based (GPV like), based on Ring-SIS and NTRU.

https://www.safecrypto.eu/pqclounge/round-1-candidates/software-analysis-signatures/

Implementing an efficient and modular trapdoor

with Pauline Bert, Gautier Eberhart and Mohamed Sabt - thanks Gautier for the slides!

- Gaussian preimage sampling techniques in the module setting
- Two proven signature schemes based on trapdoors
  - ROM and standard model
- Implementations in C

- Modularity: possible to use on rings or modules, and with a different arithmetic over $R_q = \mathbb{Z}_q/\langle x^n + 1\rangle$,
- Trapdoors have many applications: used in signature schemes but also in more advanced constructions (IBE, ABE ...).

Ring-SIS based signature scheme

with Pauline Bert, Mohamed Sabt, PA Fouque

Underlying to [ABB10]

- $\texttt{KeyGen}(\lambda) \to$ (vk,sk)
  - choose uniform $\mathbf{a}' \in R_q^{m-2}$
  - sk$= \mathbf{T} \in R^{(m-2)\times 2}$ gaussian
  - pk$= \mathbf{a} = \left(\mathbf{a}'^T | -\mathbf{a}'^T\mathbf{T}\right)^T$

For M: $\mathbf{a}_M = \left(\mathbf{a}'^T | H(M)\mathbf{g} - \mathbf{a}'^T\mathbf{T}\right)^T$

- $\texttt{Sign}(\mathbf{a}, \mathbf{T}, M) \to \mathbf{x}$
  - Using $\mathbf{T}$, find small $\mathbf{x} \in R_q^m$ with $\mathbf{x}^T\mathbf{a}_M = 0$,

- $\texttt{Verify}(\mathbf{a}, \mathbf{x}, M) \to \{0,1\}$
  - Accept iff $\mathbf{x}^T\mathbf{a}_M = 0 \bmod qR$ and $\|\mathbf{x}\|$ small.

Discrete Gaussian $\Rightarrow$ short elements in R

MP12 Trapdoors:
– $\mathbf{a}$ looks uniform,
– $\mathbf{T}$ trapdoor (allows to solve Ring-SIS)

$\mathbf{g}$ gadget vector
$H : \{0,1\}^n \to R_q$

Implementing such a scheme

Lot of conditions on parameters: hardness of Ring-SIS, correctness ...
How to be efficient ?

- Preimage sampling [MP 12, GM 18],
- Fast multiplication of ring elements in $R_q = \mathbb{Z}_q/\langle x^n + 1\rangle$

For example: use the NFLlib library [Aguilar et al. 16]

- Two important conditions: $n = 2^k$ and $q = 1 \bmod 2n$

$x^n + 1$ splits completely into linear factors

$\Rightarrow$ 3 main constraints on $q = \prod q_i$ described to use the NTT

Example of parameters

Table: Parameters set for the signature scheme

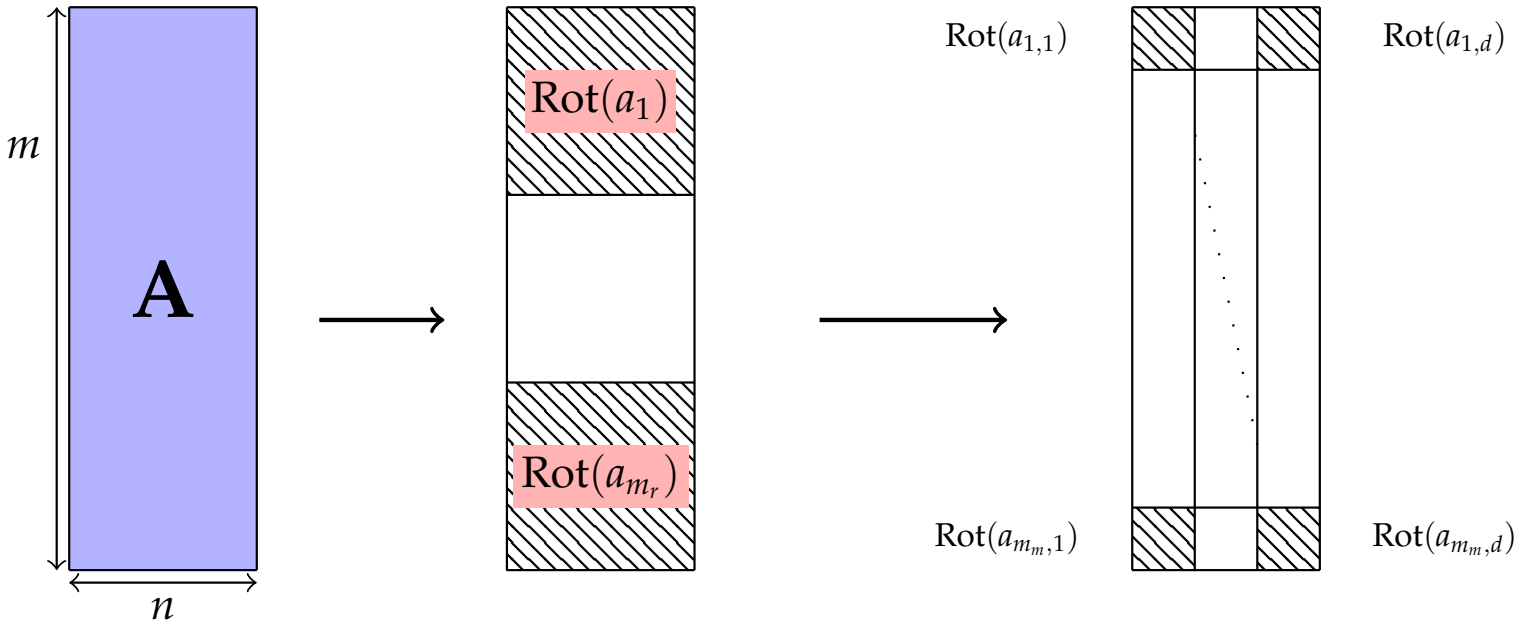
n	$\log q$	σ	R-LWE$_\sigma$	δ	R-SIS	λ
512	30	4.2	2^{64}	1.011380	2^{74}	60
1024	24	5.8	2^{378}	1.008012	2^{156}	140
1024	30	6.3	2^{246}	1.007348	2^{184}	170

$\rightarrow$ Gap in security because of the constraints on the parameter.

Module variants $\Rightarrow$ tradeoff between security and efficiency

- Hardness of Module SIS and LWE [LS15,AD17]
- Dilithium & Kyber - Crystals NIST submissions [Avanzi et al.]

Module variants



$m_r = m/n$ blocks of size n

$m_m \times d$ blocks of size $n_d = n/d$

$R = \mathbb{Z}[x]/\langle x^n + 1 \rangle$

$R = \mathbb{Z}[x]/\langle x^{n_d} + 1 \rangle$

$\mathbf{a}_i \in \mathbb{Z}_q^n$ $\quad a_i \in R_q$ $\quad \mathbf{a}_i \in (R_q)^d$

Gaussian preimage sampling

Sample $\boldsymbol{\nu}$ from a spherical discrete Gaussian distribution over

$$\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A}) = \{\, \boldsymbol{x} \in \mathcal{R}^m \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{u} \bmod q \,\}$$

G-sampling

- Message-dependent
- Sample on a coset of the very structured lattice $\Lambda_q^{\perp}(\boldsymbol{G}) \subset \mathcal{R}^{dk}$
- Direct adaptation from [MP12]

Perturbation sampling

- Trapdoor-dependent
- Sample on $\mathcal{R}^m$ with covariance $\boldsymbol{\Sigma}_p = \zeta^2 \boldsymbol{I} - \alpha^2 \begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{T}^T & \boldsymbol{I} \end{bmatrix}$
- Generalization of [GM18]

Performances and comparison

G-sampling	Perturbation sampling	Arithmetic
7.48 ms (57%)	4.13 ms (36%)	0.82 ms (7%)

Table: Cost of the operations in preimage sampling

Performances and comparison

G-sampling	Perturbation sampling	Arithmetic
7.48 ms (57%)	4.13 ms (36%)	0.82 ms (7%)

Table: Cost of the operations in preimage sampling

Scheme	Keygen	Sign	Verify
Dilithium	0.07 ms	0.17 ms	0.07 ms
qTESLA	0.91 ms	0.11 ms	0.04 ms
Falcon	6.26 ms	0.13 ms	0.03 ms
our scheme (ROM)	63.62 ms	17.48 ms	0.84 ms
our scheme (non ROM)	71.55 ms	21.94 ms	2.44 ms

Table: Running times of the signature schemes (128-bit security)

Blind Signature scheme

with Guillaume Kaim, Sébastien Canard and Jacques Traoré

- User wants a valid signature of M,
- Signer knows the secret key s_k,
- Blindness: Signer cannot link the signature to the generation transcript.

$\mathcal{S}(s_k)$		$\mathcal{U}(p_k, M)$
choose x	$\xrightarrow{x}$	$e = \text{challenge}(M, p_k, x)$
	$\xleftarrow{e^*}$	$e^* = \text{blind}(e)$
$z^* = \text{sign}(e^*, s_k)$	$\xrightarrow{z^*}$	$z = \text{unblind}(z^*)$ $\text{Output}(M, (z, e))$

Blind signature scheme

One known lattice-based scheme [Rückert 2010]

$\mathcal{S}(s_k)$		$\mathcal{U}(p_k,M)$
	$\rightarrow$	Generate the challenge *local restart*
Signature generation	$\leftarrow$	
Restart from scratch (1)	$\rightarrow$	Unblind the signature *Restart from scratch (2)*

Our objective

- Avoid restart to build a more efficient scheme,
- Tool used: Trapdoors and rejection sampling.

Our scheme (ring setting)

- secret key $\mathbf{s} \in \mathrm{R}_3^m$
- vector of polynomial $\mathbf{a} \in \mathrm{R}_q^m$ with trapdoor $\mathbf{T_a}$,
- hash function $h_{\mathbf{a}}(\mathbf{x}) = \sum_i a_i x_i$
- public key $p = h_{\mathbf{a}}(\mathbf{s})$

$\mathcal{S}(\mathbf{s}, \mathbf{T_a})$		$\mathcal{U}(\mathbf{a}, p, M)$
$\mathbf{y} \in \mathbb{Z}^m$ gaussian $x = h_{\mathbf{a}}(\mathbf{y})$	$\xrightarrow{x}$	$t_1, \mathbf{t}_2$ gaussian $e = H(x - p \cdot t_1 - h_{\mathbf{a}}(\mathbf{t}_2), M)$ $e^* = e - t_1$, accept with proba $P(e)$
$\mathbf{v}$ such that $h_{\mathbf{a}}(\mathbf{v}) = 0$ $\mathbf{z}^* = e^* \cdot \mathbf{s} + \mathbf{y} + \mathbf{v}$ Accept $\mathbf{z}^*$ with proba $P(s)$	$\xleftarrow{e^*}$	otherwise restart with fresh t_1
otherwise restart with fresh $\mathbf{v}$	$\xrightarrow{\mathbf{z}^*}$	$\mathbf{z} = \mathbf{z}^* - \mathbf{t}_2$ Output $(M, (z, e))$

Conclusion

- Lattice-based signature scheme
  - Efficient constructions,
  - Importance of Gaussian sampling, cryptanalysis, choice of parameters ...

- Trapdoor-based construction
  - Allows signature in the standard model, and more advanced constructions (lot of applications!).
  - Still costly.

Conclusion

- Lattice-based signature scheme
  - Efficient constructions,
  - Importance of Gaussian sampling, cryptanalysis, choice of parameters ...

- Trapdoor-based construction
  - Allows signature in the standard model, and more advanced constructions (lot of applications!).
  - Still costly.

Thank You