Post-quantum cryptography

Mădălina Bolboceanu
mbolboceanu@bitdefender.com

Bitdefender®

<TechDay/2020>

D0 NØT FΞAR CΘMPUTΣRS

• PILOT EDITION •

Research in cryptography

- studying security foundations and building advanced post-quantum cryptographic solutions, mainly **lattice-based**

Miruna Roșca

Mădălina Bolboceanu

Radu Țițiu

{mrosca, mbolboceanu, rtitiu}@bitdefender.com

Collaborations

- ENS de Lyon
- Weizmann Institute of Science
- Monash University
- Indian Institute of Technology Madras
- Florida Atlantic University
- Mitsubishi Electric
- NIT Durgapur
- Algorand
- NTT

Overview

- Why post-quantum cryptography?
- Lattice-based cryptography
- Our contributions

Crypto Today

Crypto Primitives used Today

symmetric:

- AES
- SHA-1, SHA-2, SHA-3

public-key:

- RSA
- ECDH
- DSA
- ECDSA

Crypto Primitives used Today

symmetric:

- AES
- SHA-1, SHA-2, SHA-3

public-key:

- **RSA**
- **ECDH**
- **DSA**
- **ECDSA**

General recipe to build public-key crypto

break cryptographic system

$\Downarrow$

solve hard problem

General recipe to build public-key crypto

break RSA

$\Downarrow$

solve factorization

$$35 = 5 \cdot 7$$
$$217 = 7 \cdot 31$$
$$113879 = 263 \cdot 433$$
$$646769 = 641 \cdot 1009$$

General recipe to build public-key crypto

break Diffie-Hellman

$\Downarrow$

find discrete logarithm

$\mathbb{Z}_7 = \{0, 1, 2, 3, 4, 5, 6\}$

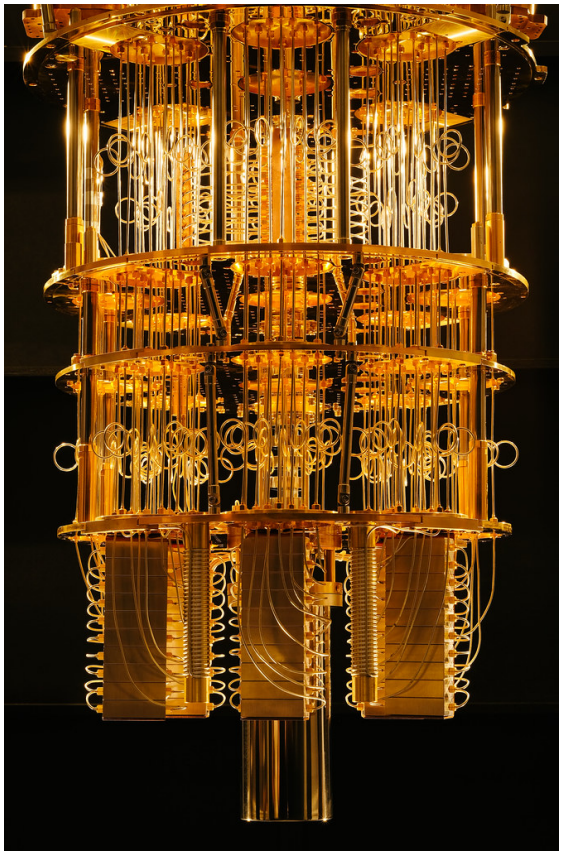
$3^2 = 7 \cdot 1 + 2$

$3^2 \equiv 2 \mod 7$

Quantum computers & more

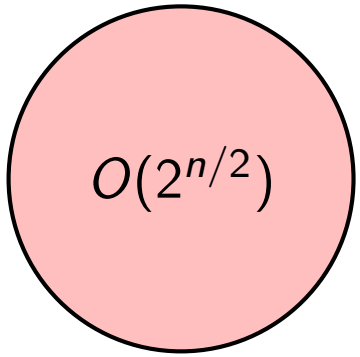
What is a quantum computer?

- a device relying on the laws of quantum physics
- can perform faster on particular tasks



What kind of tasks?

Solves



Database Searching
(Grover's algorithm)

What kind of tasks?

Solves

$O(2^{n/2})$

Database Searching
(Grover's algorithm)

$N = pq$

Factoring
(Shor's algorithm)

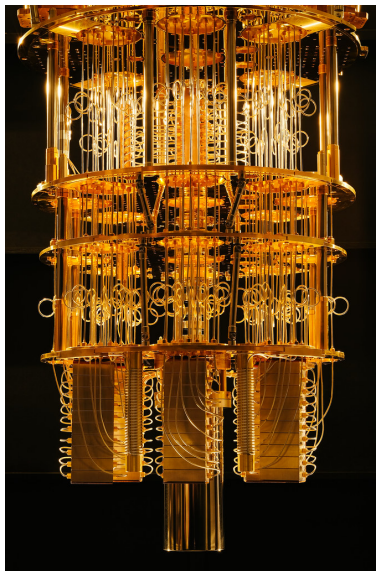
$g^x \equiv y \mod q$

Discrete logarithm
(Shor's algorithm)

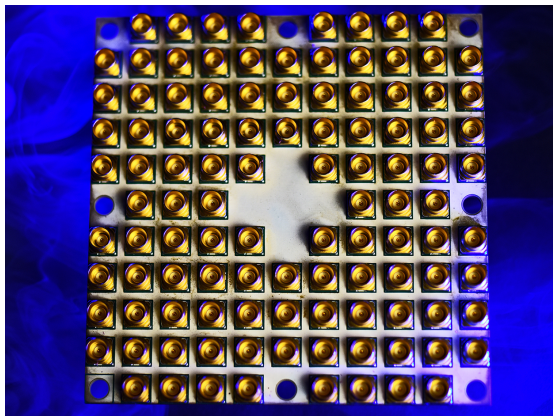
Crypto Primitives used Today

symmetric:	public-key:
	✗RSA
	✗ECDH
✓AES	✗DSA
✓SHA-1, SHA-2, SHA-3	✗ECDSA

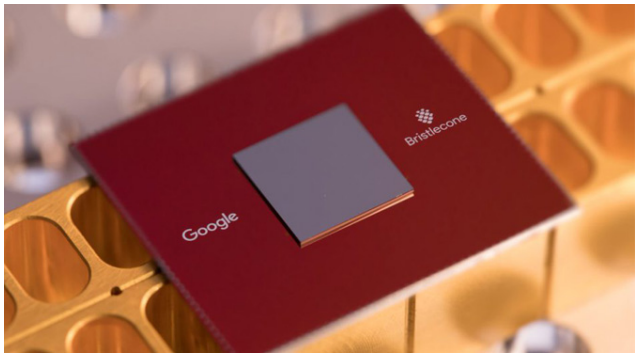
How close are we to quantum computing?



IBM
(2017)
50 qubits



Intel (2018)
49 qubits



Google (2018)
72 qubits

...still laboratory experiments

Reaction of the crypto community

ARE WE READY FOR A 'QUANTUM SURPRISE' FROM CHINA? | TECH & SCIENCE

manipulating bits, quantum computers take advantage of a peculiar quality of subatomic particles to exist in more than one "state" at a time. The physicist Edwin Schrodinger famously likened this "superposition of states" to a cat being both dead and alive at the same time. A particle of light (called a photon) can be made to represent 0, 1 and other values all at once. A quantum computer can manipulate these particles to perform many calculations simultaneously, vastly increasing the speed at which it can solve complex problems, such as cracking encryption.

China has made quantum computing a strategic imperative. Although China has been accused of stealing technology in the past, its quantum computing effort is home-grown and substantial. It reportedly spent $400 million on new research labs in Anhui province. China is not the only country developing quantum technology--the U.S., Europe and Japan also have projects in the works. An $80 million NSA project to build a quantum computer, called Penetrating Hard Targets, was revealed among the documents leaked by Edward Snowden.

Duke | PRATT SCHOOL of ENGINEERING Undergraduate

Home » About » News »

Duke to Lead $15 Million Program to Create First Practical Quantum Computer

AUGUST 7, 2018

Seven-university, five-year interdisciplinary collaboration is funded by the National Science Foundation's largest quantum computing effort to date

By Ken Kingery

NIST is calling!

Announcing Request for Nominations for Public-Key Post-Quantum Cryptographic Algorithms

December 20, 2016

NIST has initiated a process to develop and standardize one or more additional public-key cryptographic algorithms to augment FIPS 186-4, Digital Signature Standard (DSS), as well as special publications SP 800-56A Revision 2, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*, and SP 800-56B, Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography.

It is intended that the new public-key cryptography standards will specify one or more additional unclassified, publicly disclosed digital signature, public-key encryption, and key-establishment algorithms that are capable of protecting sensitive government information well into the foreseeable future, including after the advent of quantum computers.

FEDERAL REGISTER NOTICE

Document Number: 2016-30615

PARENT PROJECT

See: Post-Quantum Cryptography
See: Key Management
See: Hash-Based Signatures

General recipe in post-quantum setting

break cryptographic system

$\Downarrow$

solve hard problem

quantum secure

$\Uparrow$

quantum resistant

General recipe in post-quantum setting

break cryptographic system

$\Downarrow$

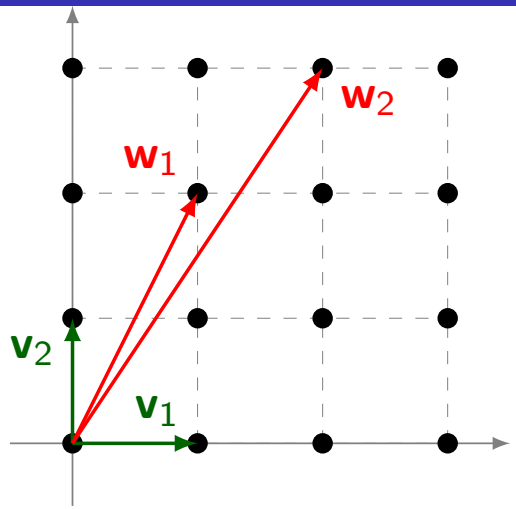
solve hard problem

quantum secure

$\Uparrow$

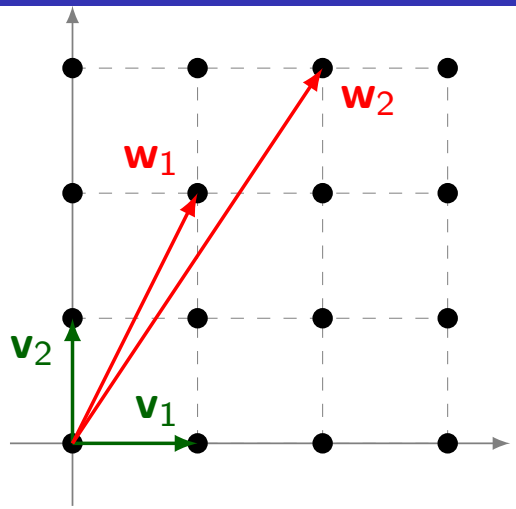
quantum resistant

Where to find hard problems?

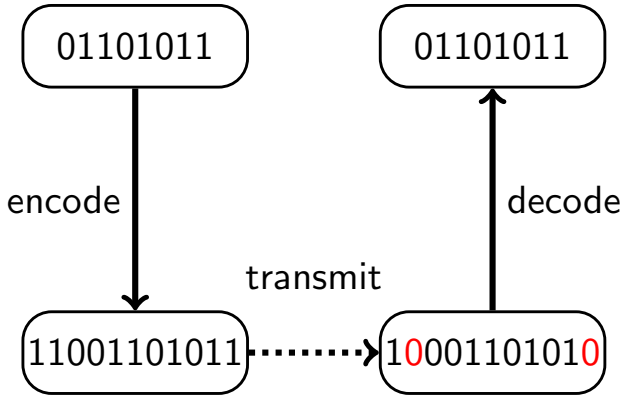


lattices

Where to find hard problems?

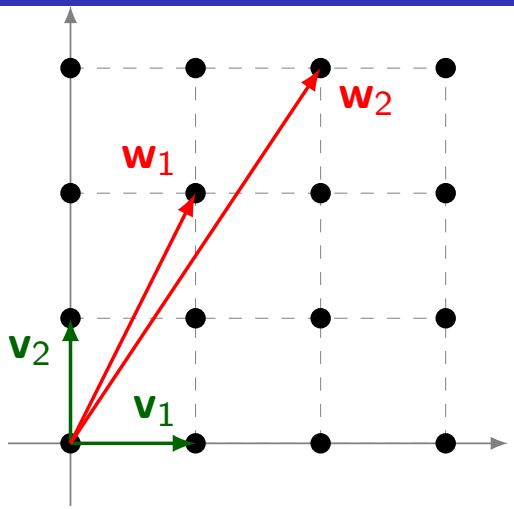


lattices

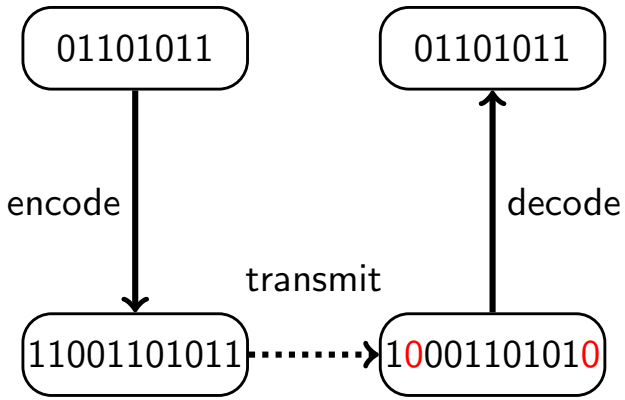


codes

Where to find hard problems?



lattices

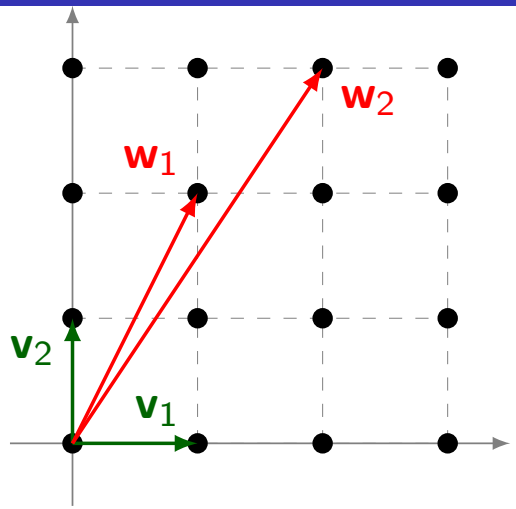


codes

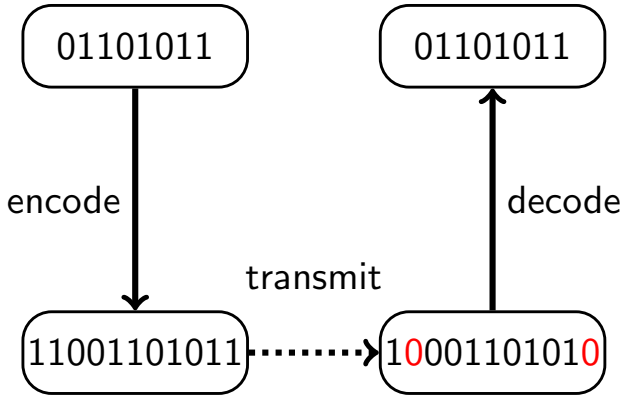
$$x_0x_1 + x_1x_3 = 2$$
$$x_1x_2 + x_0x_3 = 0$$
$$x_1^2 + x_3^2 = 1$$
$$x_1x_3 + x_2^2 = 0$$
$$\text{in } \mathbb{F}_3$$

multivariate equations

Where to find hard problems?



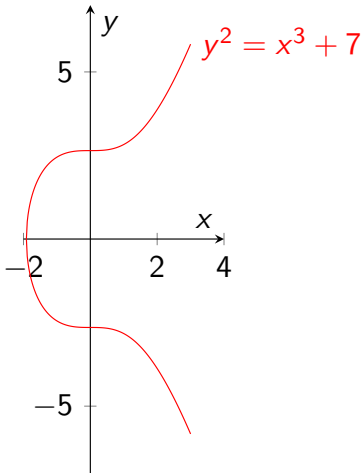
lattices



codes

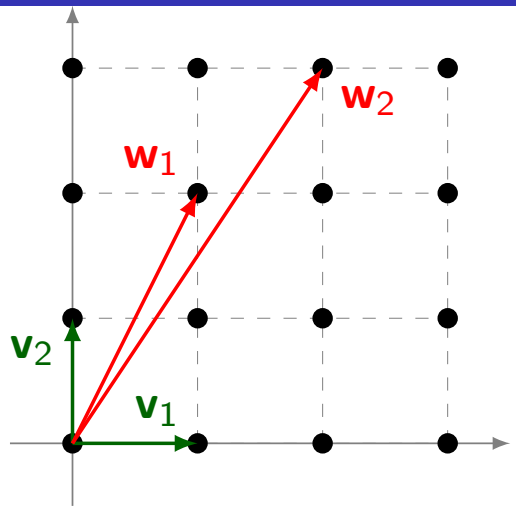
$$x_0x_1 + x_1x_3 = 2$$
$$x_1x_2 + x_0x_3 = 0$$
$$x_1^2 + x_3^2 = 1$$
$$x_1x_3 + x_2^2 = 0$$
$$\text{in } \mathbb{F}_3$$

multivariate equations

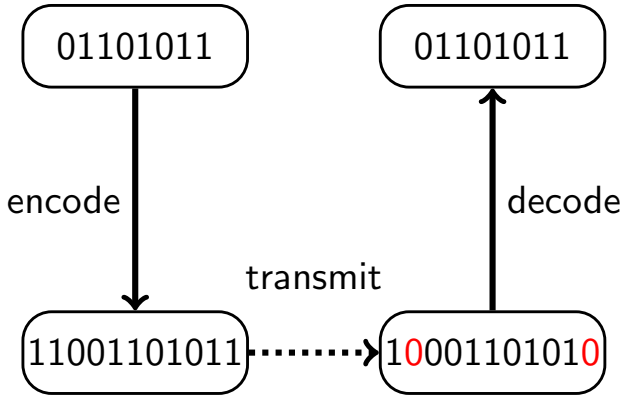


elliptic curves

Where to find hard problems?



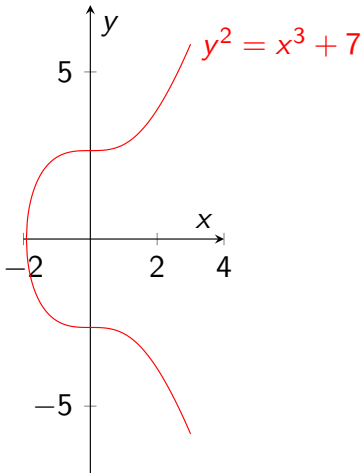
lattices



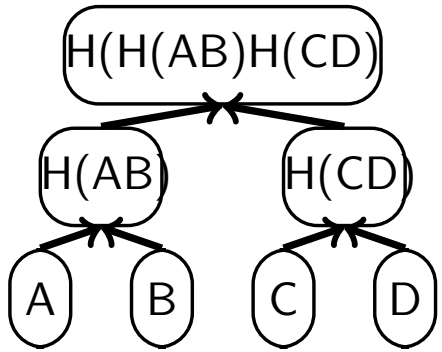
codes

$$x_0x_1 + x_1x_3 = 2$$
$$x_1x_2 + x_0x_3 = 0$$
$$x_1^2 + x_3^2 = 1$$
$$x_1x_3 + x_2^2 = 0$$
$$\text{in } \mathbb{F}_3$$

multivariate equations



elliptic curves



hash functions

More about NIST competition

- 82 submissions until Nov 2017
- 69 first-round candidates on Dec 2017 (5 withdrawn)
- 26 second-round candidates on Jan 26, 2019:

KEM	
lattices	9
codes	7
isogenies	1

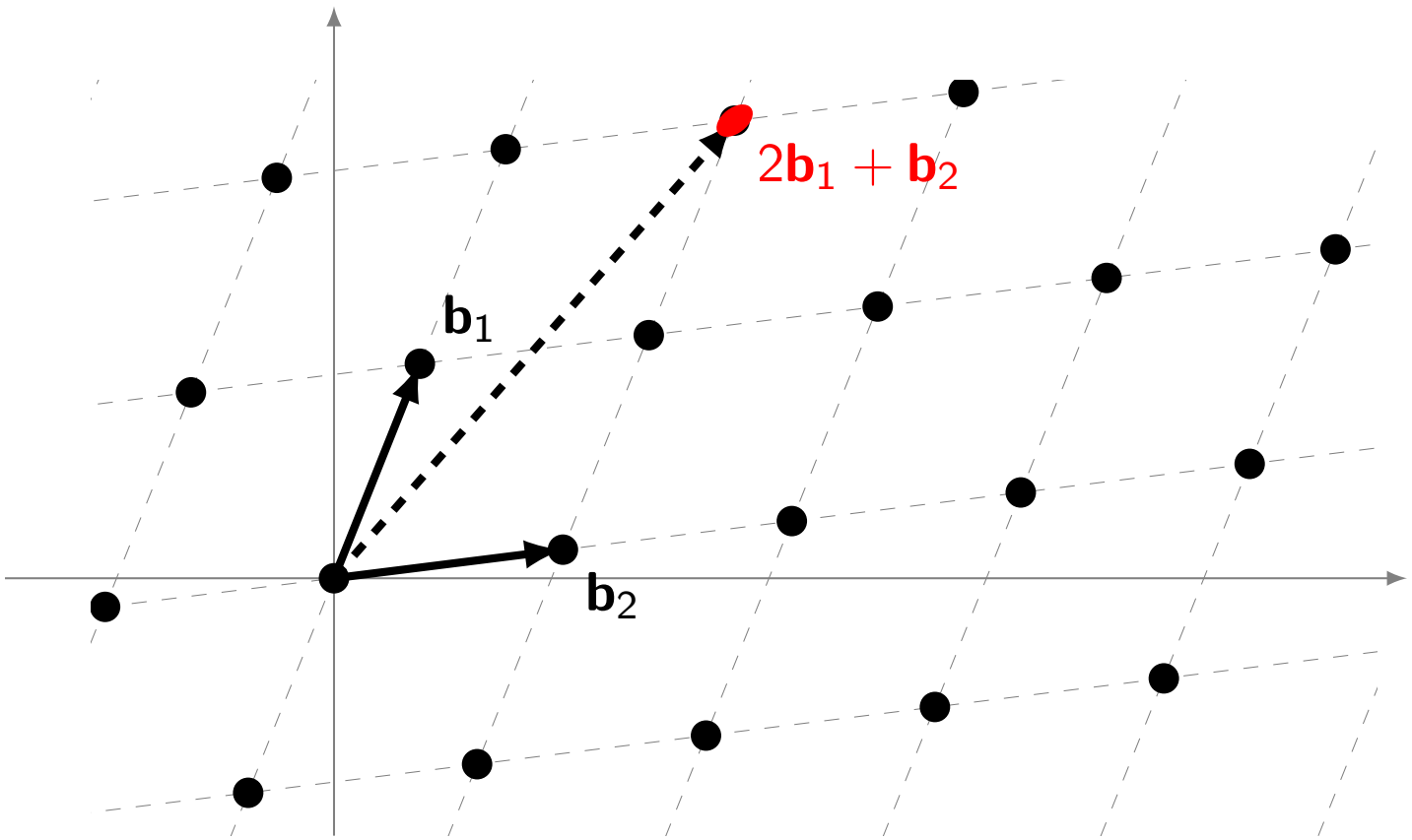
Digital Signatures	
multivariate	4
lattices	3
hash-based	2

- a possible third-round/select algorithms in 2020/2021
- more info: https://csrc.nist.gov/Projects/post-quantum-cryptography/

Lattice-based cryptography

What is a lattice?

$n = 2$:

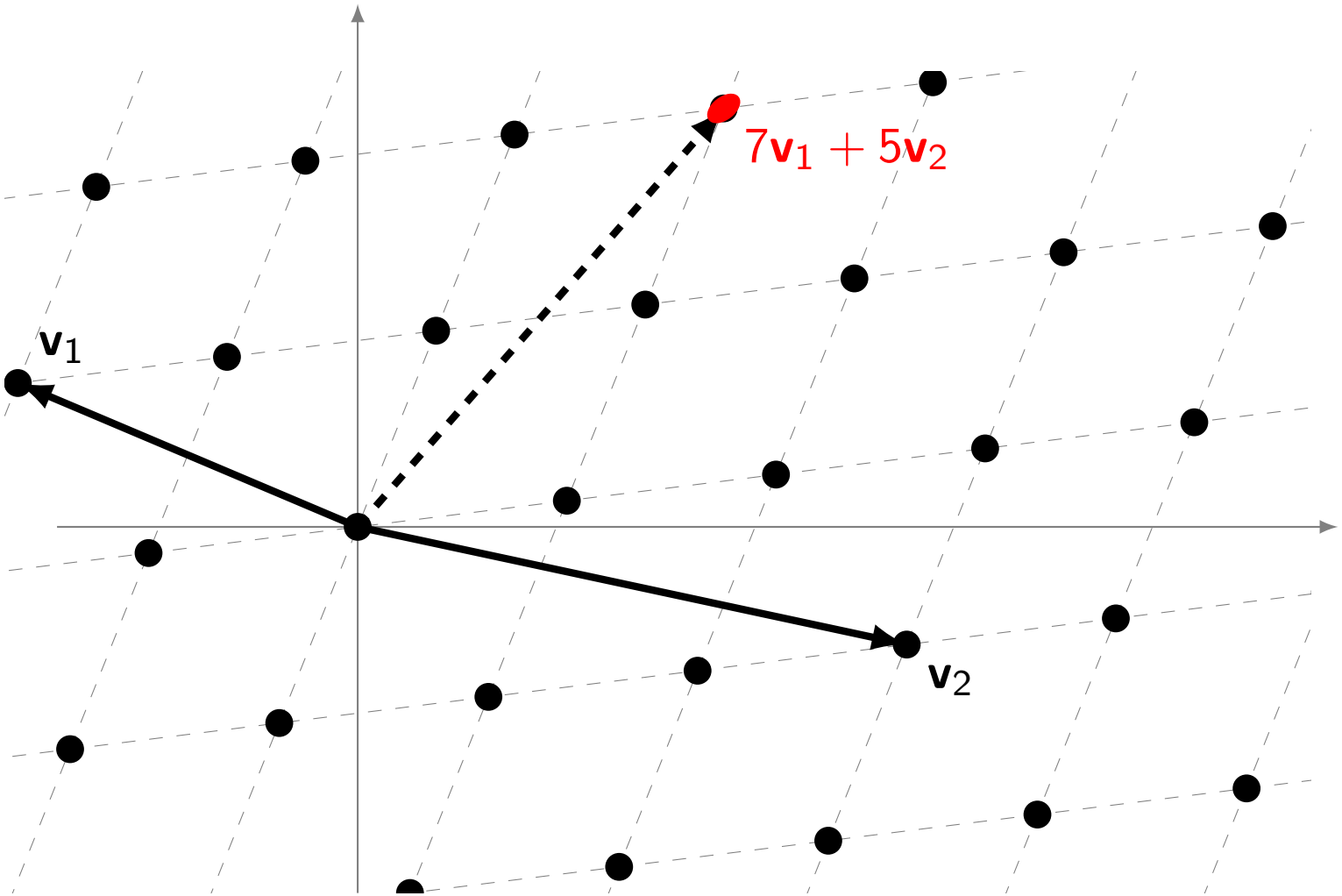


Lattice

Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ be linearly independent vectors from $\mathbb{R}^n$. Then $L = L(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^{n} a_i \mathbf{b}_i | a_i \in \mathbb{Z}\}$ is the lattice generated by them.

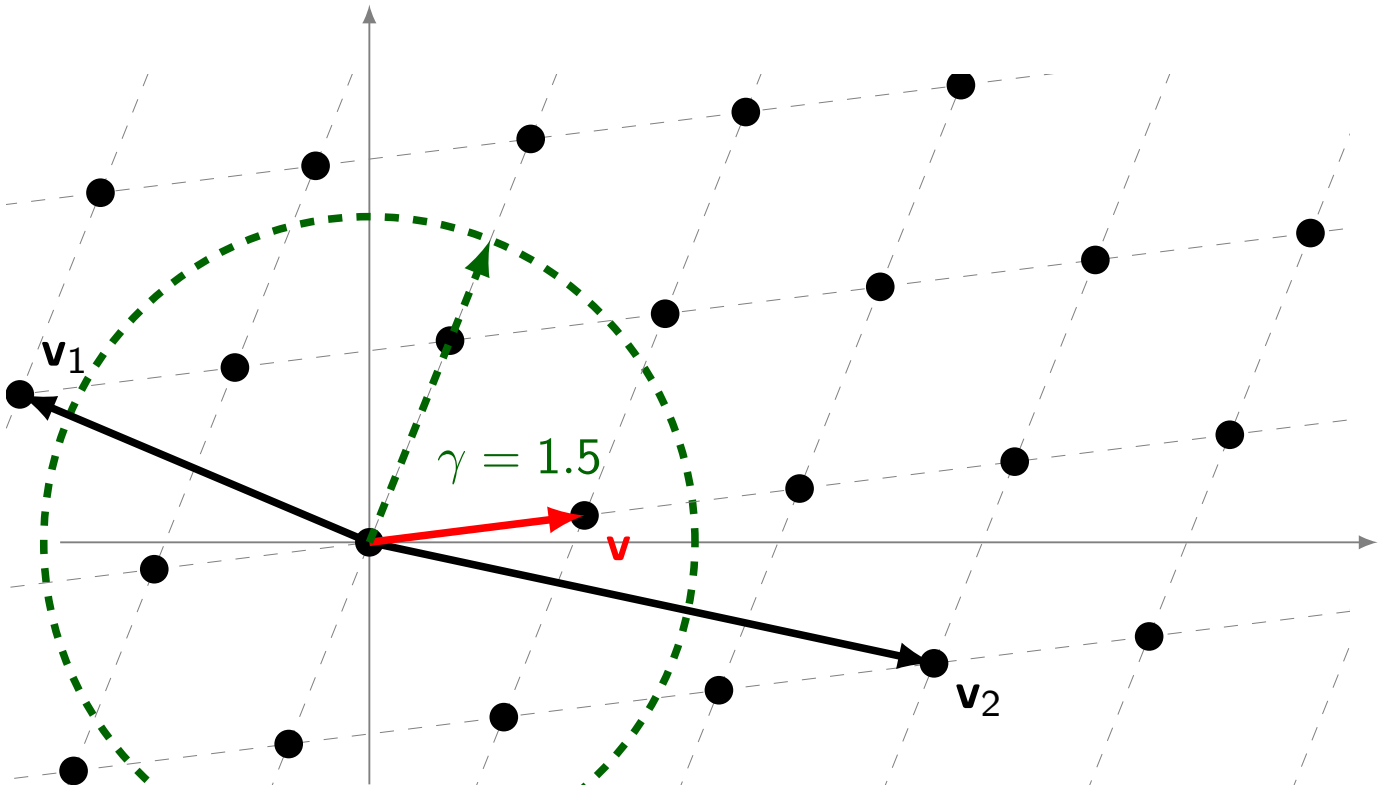
Same lattice... different bases!

$n = 2$:



Shortest Vector Problem (SVP)

$n = 2$:



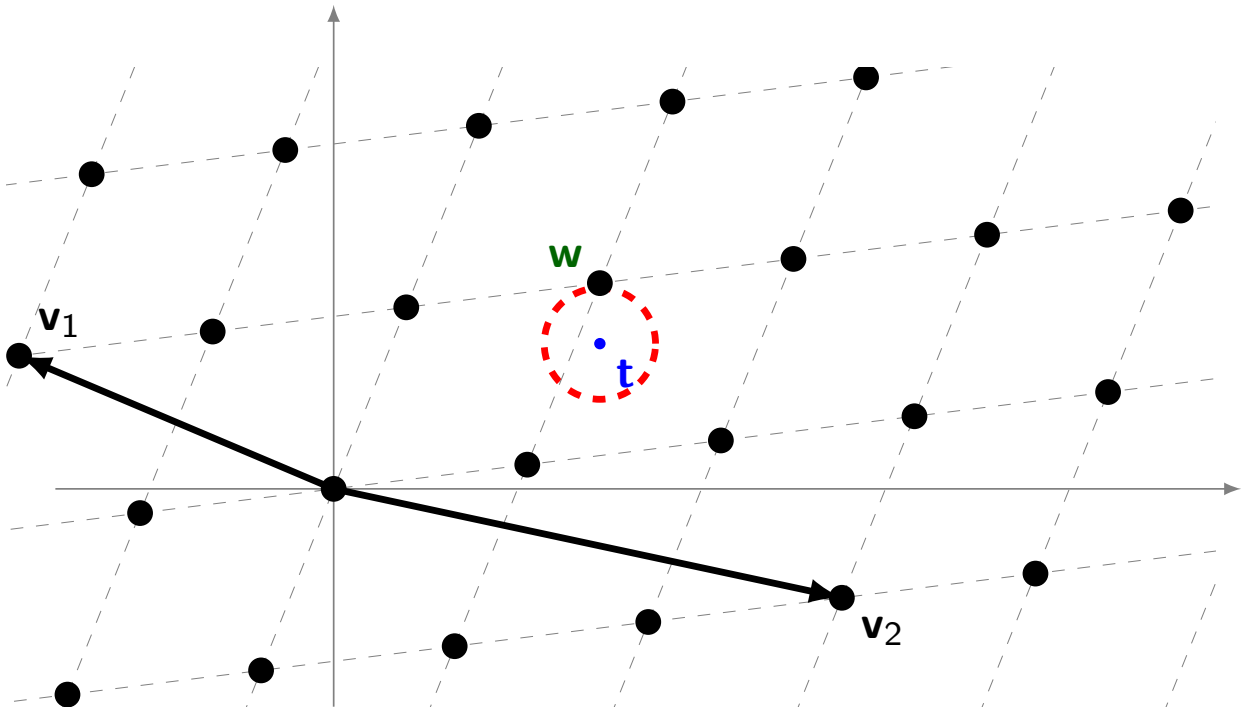
ApproxSVP$_\gamma$

Given $L \subset \mathbb{R}^n$, find a short nonzero $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq \gamma \lambda_1(L)$.

$\lambda_1(L)$:= the length of a shortest nonzero vector from L.

Closest Vector Problem (CVP)

$n = 2$:



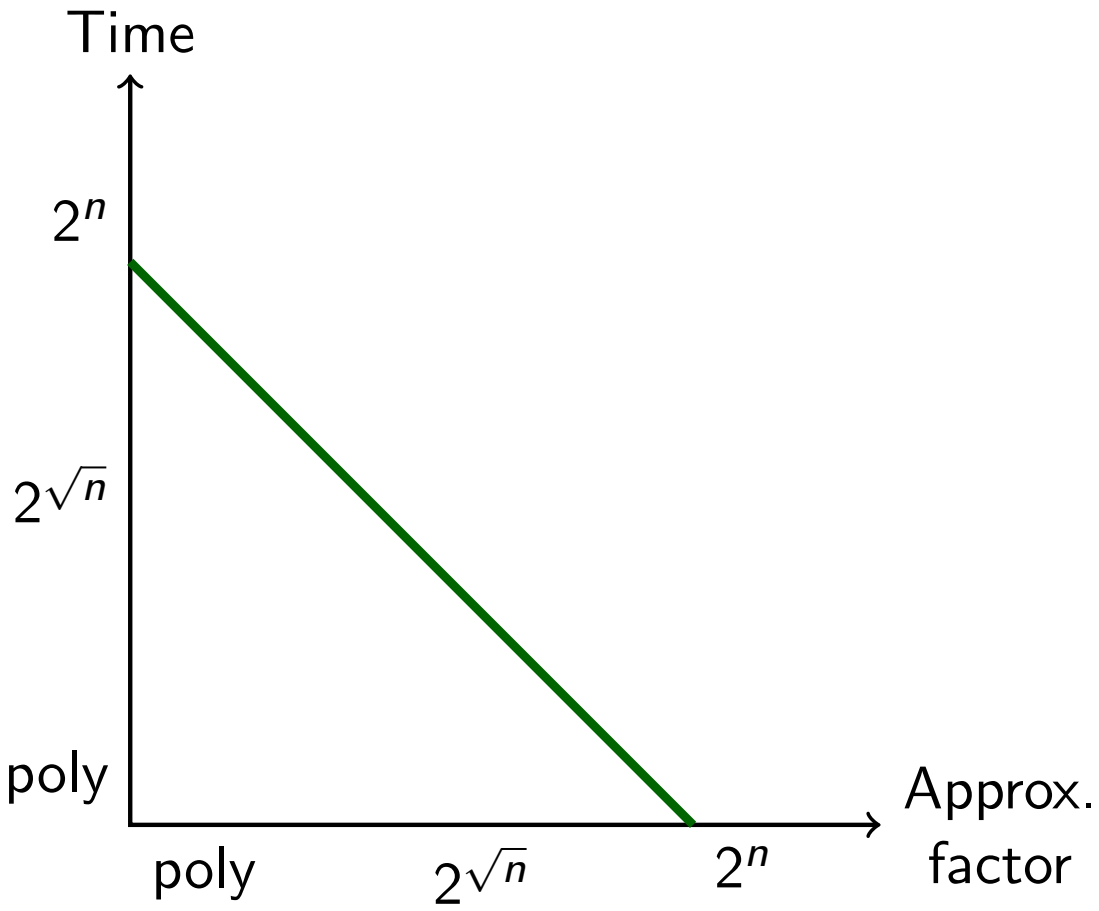
ApproxCVP$_\gamma$

Given $L \subset \mathbb{R}^n$ and a target $\mathbf{t} \in \mathbb{R}^n$, find $\mathbf{w} \in L$ s.t. $\|\mathbf{t} - \mathbf{w}\| \leq \gamma \cdot \text{dist}(\mathbf{t}, L)$.

$\text{dist}(\mathbf{t}, L) := \min_{\mathbf{w}' \in L} \|\mathbf{t} - \mathbf{w}'\|$

How hard are ApproxSVP$_\gamma$/ApproxCVP$_\gamma$?

$n :=$ dimension of the lattice



poly factors $\rightsquigarrow$ hard problems $\rightsquigarrow$ **use them in crypto!**

General recipe in lattice-based setting

break cryptographic system

$\Downarrow$

solve ApproxSVP_γ

...is it done?

break cryptographic system

$\Downarrow$

Learning with Errors

$\Downarrow$

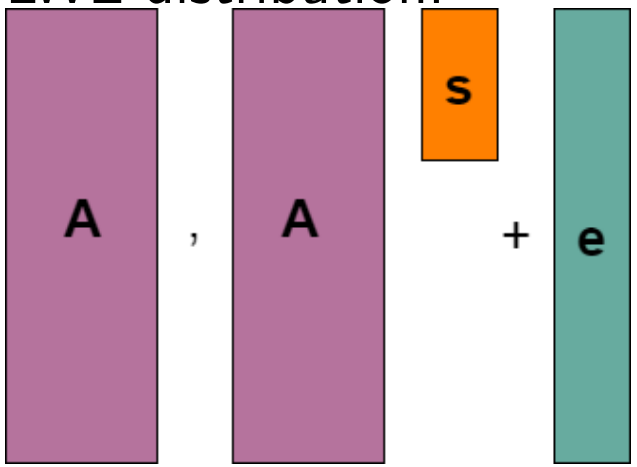
solve ApproxSVP_γ

Learning with Errors [Regev05]

Let $\mathbf{s} \in \mathbb{Z}_q^n, m \geq n$

$$\begin{cases} \mathbf{A} \xleftarrow{u} \mathbb{Z}_q^{m \times n} \\ \mathbf{e} \text{ is small} \end{cases}$$

LWE distribution:



Search: Given LWE samples, find $\mathbf{s}$.
Decision: Distinguish LWE samples from uniform samples.

Playing with linear algebra

Given: $\mathbf{A} \in \mathbb{Z}^{m \times n}$, $\mathbf{b} \in \mathbb{Z}^m$

Find: $\mathbf{s} \in \mathbb{Z}^n$ s.t. $\mathbf{A} \cdot \mathbf{s} = \mathbf{b}$, i.e. from:

$$\begin{aligned} a_{11}s_1 + a_{12}s_2 + \ldots + a_{1n}s_n &= b_1 \\ a_{21}s_1 + a_{22}s_2 + \ldots + a_{2n}s_n &= b_2 \\ &\vdots \\ a_{m1}s_1 + a_{m2}s_2 + \ldots + a_{mn}s_n &= b_m \end{aligned}$$

$$s_1 \underbrace{\begin{pmatrix} a_{11} \\ \vdots \\ a_{m1} \end{pmatrix}}_{\mathbf{v}_1} + \ldots + s_n \underbrace{\begin{pmatrix} a_{1n} \\ \vdots \\ a_{mn} \end{pmatrix}}_{\mathbf{v}_m} = \underbrace{\begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}}_{\mathbf{b}}$$

The modular case

Let q be prime.

Given: $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{b} \in \mathbb{Z}_q^m$

Find: $\mathbf{s} \in \mathbb{Z}_q^n$ s.t. $\mathbf{A} \cdot \mathbf{s} = \mathbf{b} \bmod q$, i.e. from:

$$\begin{aligned} a_{11}s_1 + a_{12}s_2 + \ldots + a_{1n}s_n &= b_1 \bmod q \\ a_{21}s_1 + a_{22}s_2 + \ldots + a_{2n}s_n &= b_2 \bmod q \\ &\vdots \\ a_{m1}s_1 + a_{m2}s_2 + \ldots + a_{mn}s_n &= b_m \bmod q \end{aligned}$$

The modular case

Let q be prime.

Given: $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{b} \in \mathbb{Z}_q^m$

Find: $\mathbf{s} \in \mathbb{Z}_q^n$ s.t. $\mathbf{A} \cdot \mathbf{s} = \mathbf{b} \bmod q$, i.e. from:

$$a_{11}s_1 + a_{12}s_2 + \ldots + a_{1n}s_n + qs_{n+1} = b_1$$
$$a_{21}s_1 + a_{22}s_2 + \ldots + a_{2n}s_n + qs_{n+2} = b_2$$
$$\vdots$$
$$a_{m1}s_1 + a_{m2}s_2 + \ldots + a_{mn}s_n + qs_{n+m} = b_m$$

$$s_1 \underbrace{\begin{pmatrix} a_{11} \\ \vdots \\ a_{m1} \end{pmatrix}}_{\mathbf{v}_1} + \ldots + s_n \underbrace{\begin{pmatrix} a_{1n} \\ \vdots \\ a_{mn} \end{pmatrix}}_{\mathbf{v}_n} + s_{n+1} \underbrace{\begin{pmatrix} q \\ \vdots \\ 0 \end{pmatrix}}_{\mathbf{v}_{n+1}} + \ldots + s_{n+m} \underbrace{\begin{pmatrix} 0 \\ \vdots \\ q \end{pmatrix}}_{\mathbf{v}_{n+m}} = \underbrace{\begin{pmatrix} b_1 \\ \vdots \\ b_1 \end{pmatrix}}_{\mathbf{b}}$$

The LWE case

Let q be prime and $\mathbf{e}$ small and unknown.

Given: $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{b} \in \mathbb{Z}_q^m$.

Find: $\mathbf{s} \in \mathbb{Z}_q^n$ s.t. $\mathbf{A} \cdot \mathbf{s} + \mathbf{e} = \mathbf{b} \bmod q$, i.e. from

$$\begin{aligned}
a_{11}s_1 + a_{12}s_2 + \ldots + a_{1n}s_n &\quad + e_1 = b_1 \bmod q \\
a_{21}s_1 + a_{22}s_2 + \ldots + a_{2n}s_n &\quad + e_2 = b_2 \bmod q \\
&\vdots \\
a_{m1}s_1 + a_{m2}s_2 + \ldots + a_{mn}s_n &\quad + e_m = b_m \bmod q
\end{aligned}$$

The LWE case

Let q be prime and $\mathbf{e}$ small and unknown.

Given: $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{b} \in \mathbb{Z}_q^m$.

Find: $\mathbf{s} \in \mathbb{Z}_q^n$ s.t. $\mathbf{A} \cdot \mathbf{s} + \mathbf{e} = \mathbf{b} \bmod q$, i.e. from

$$a_{11}s_1 + a_{12}s_2 + \ldots + a_{1n}s_n + qs_{n+1} + e_1 = b_1$$
$$a_{21}s_1 + a_{22}s_2 + \ldots + a_{2n}s_n + qs_{n+2} + e_2 = b_2$$
$$\vdots$$
$$a_{m1}s_1 + a_{m2}s_2 + \ldots + a_{mn}s_n + qs_{n+m} + e_m = b_m$$

$$s_1 \underbrace{\begin{pmatrix} a_{11} \\ \vdots \\ a_{m1} \end{pmatrix}}_{\mathbf{v}_1} + \ldots + s_n \underbrace{\begin{pmatrix} a_{1n} \\ \vdots \\ a_{mn} \end{pmatrix}}_{\mathbf{v}_n} + s_{n+1} \underbrace{\begin{pmatrix} q \\ \vdots \\ 0 \end{pmatrix}}_{\mathbf{v}_{n+1}} + \ldots + s_{n+m} \underbrace{\begin{pmatrix} 0 \\ \vdots \\ q \end{pmatrix}}_{\mathbf{v}_{n+m}} + \underbrace{\begin{pmatrix} e_1 \\ \vdots \\ e_m \end{pmatrix}}_{\mathbf{e}} = \underbrace{\begin{pmatrix} b_1 \\ \vdots \\ b_1 \end{pmatrix}}_{\mathbf{b}}$$

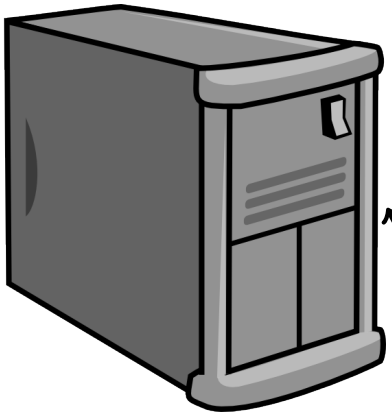
Our contributions

Our domains of interest

1. build advanced cryptographic primitives from LWE
   - multi-client functional encryption [LT19], [ALMT20]
   - distributed pseudorandom functions [LST18]
2. algebraic variants of LWE
   - hardness foundations [RSW18], [BoI18], [BBPS19]
   - applications in crypto
     - public-key crypto [RSSS17]
     - digital signatures [BDHR+20]

Advanced crypto primitives from LWE

Long-term encryption

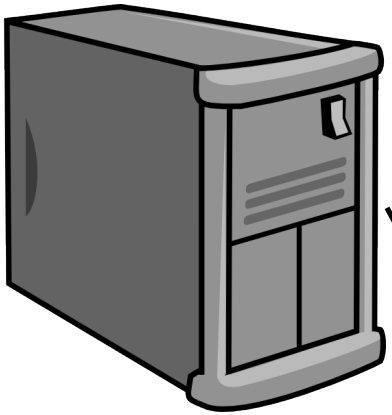


'Alice'

I wanna store my key.

application of [LST18]

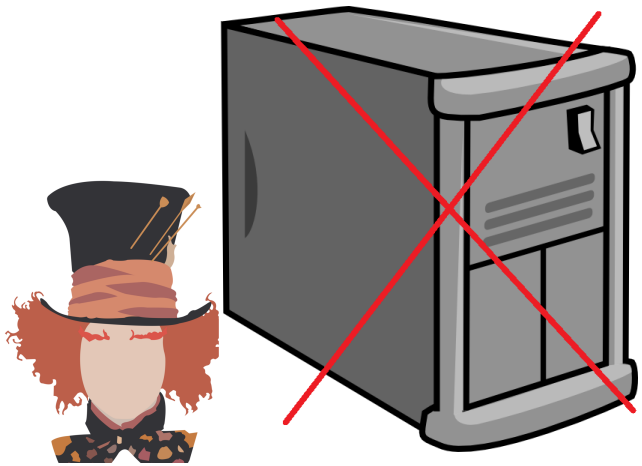
Long-term encryption



I can store your key!

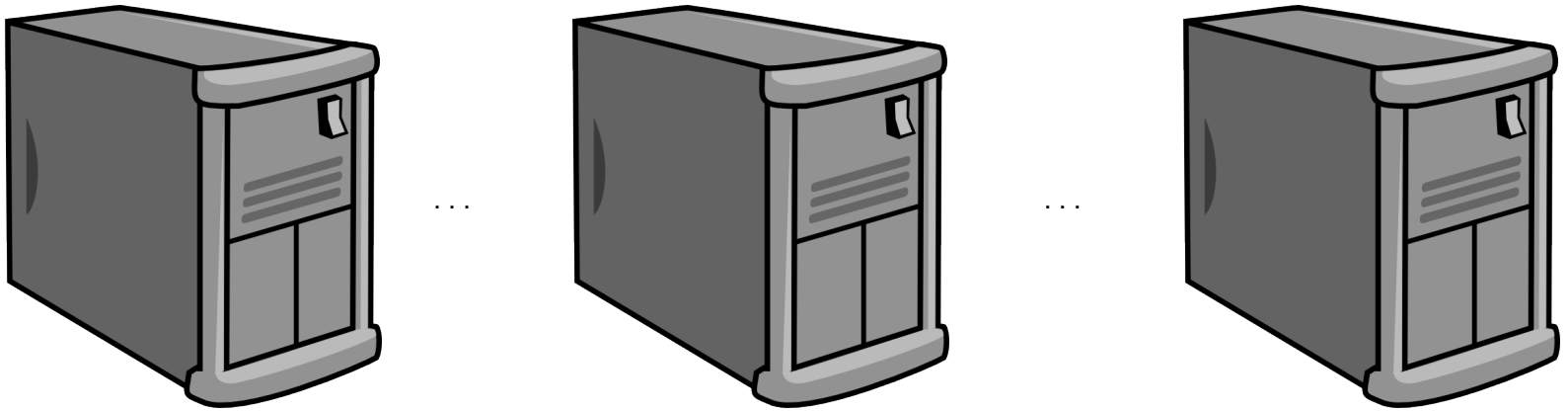
key

Long-term encryption

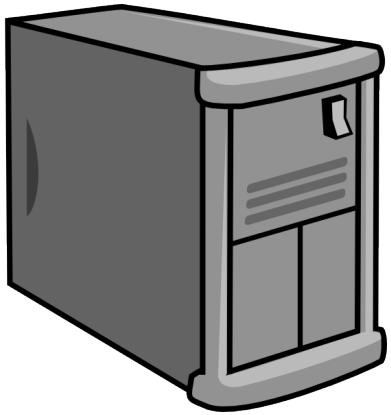


Oh no...

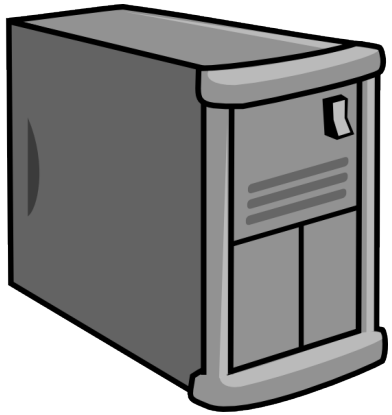
Long-term encryption



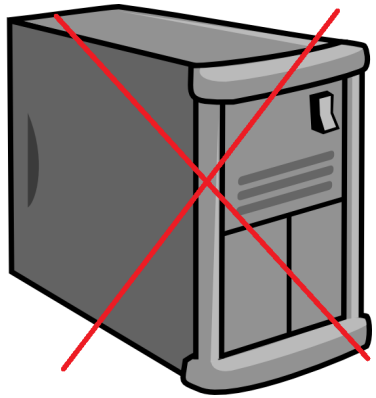
Long-term encryption



...

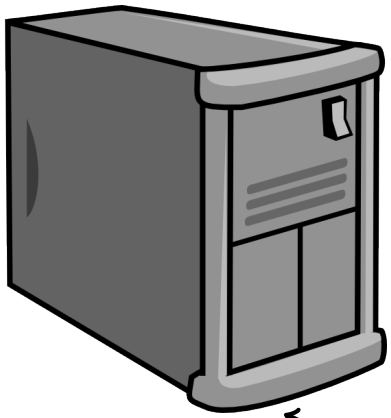


...

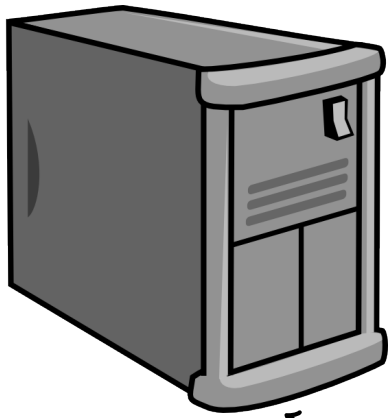


Long-term encryption

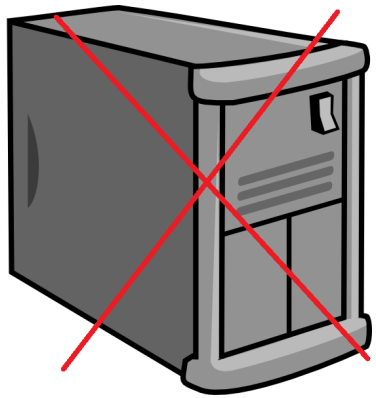
We can help you store the key!



...



...

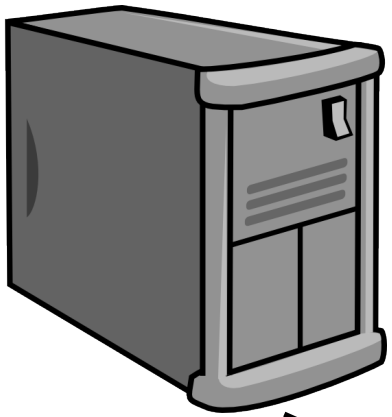


'Alice'

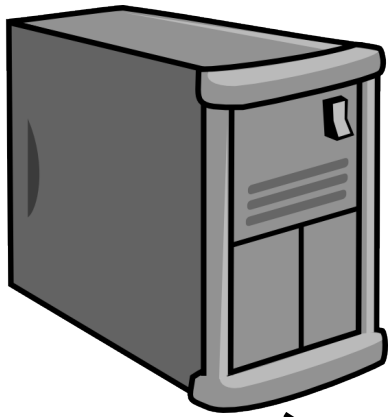
'Alice'

Long-term encryption

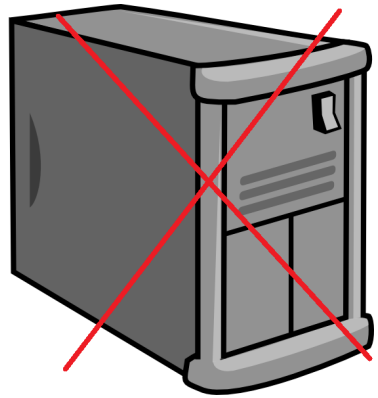
We can help you store the key!



...



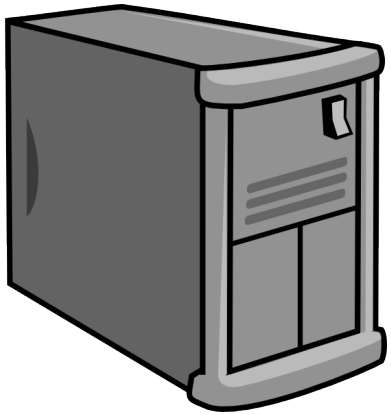
...



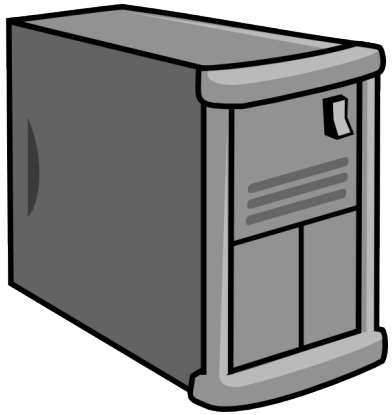
key_1

key_t

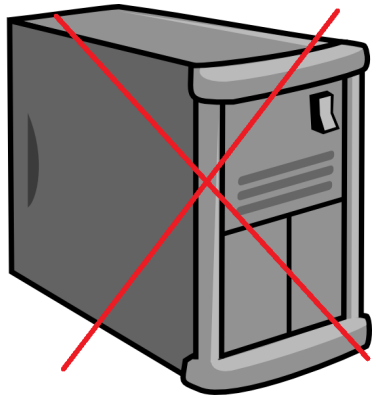
Long-term encryption



...



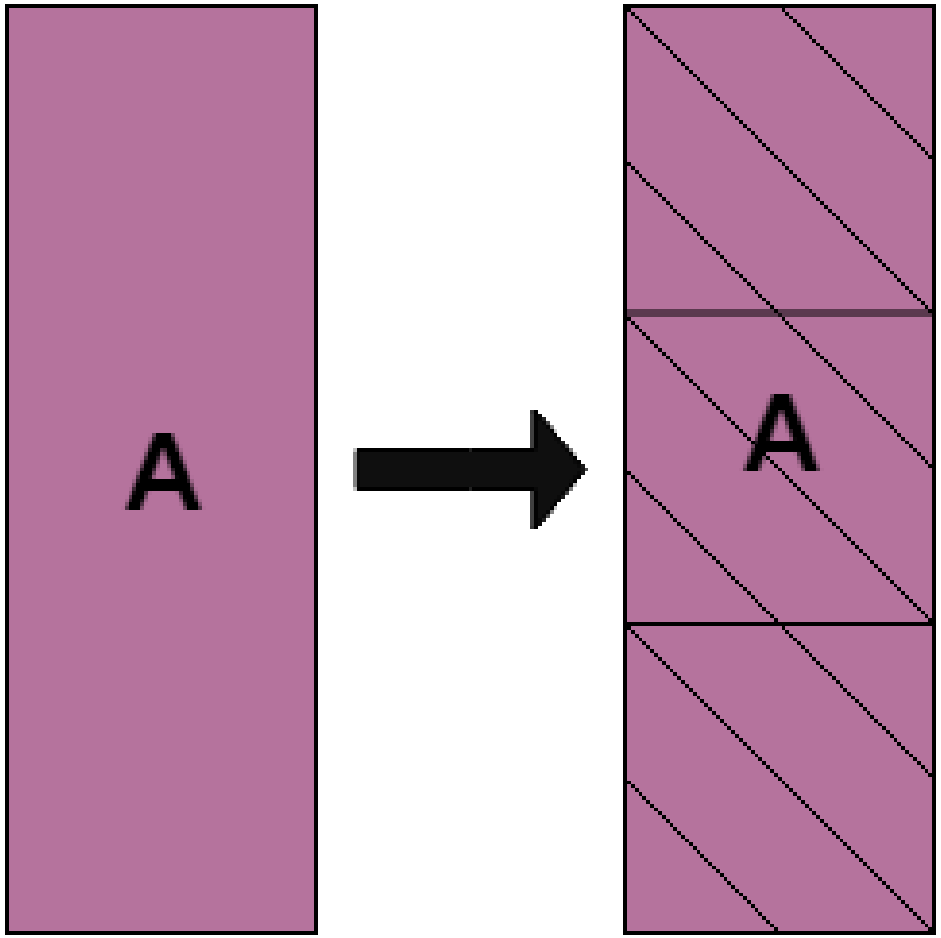
...



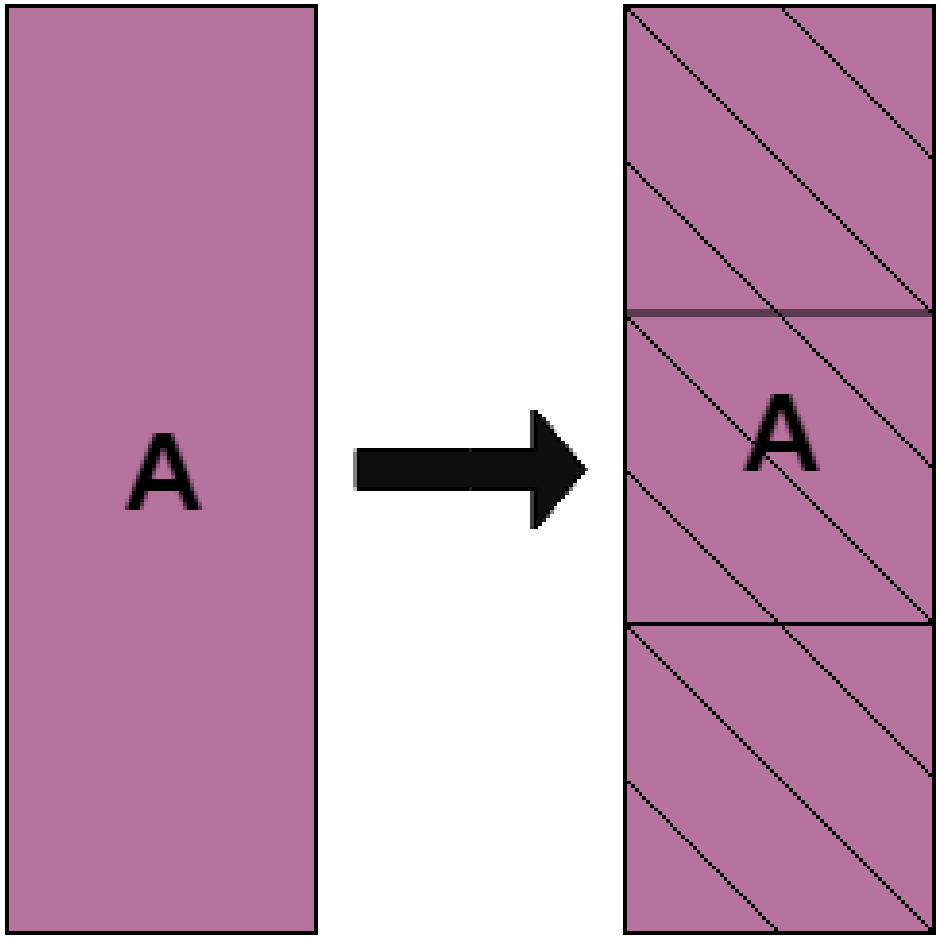
Yay, I can safely store my key!

Algebraic variants of LWE

Why algebraic variants of LWE?

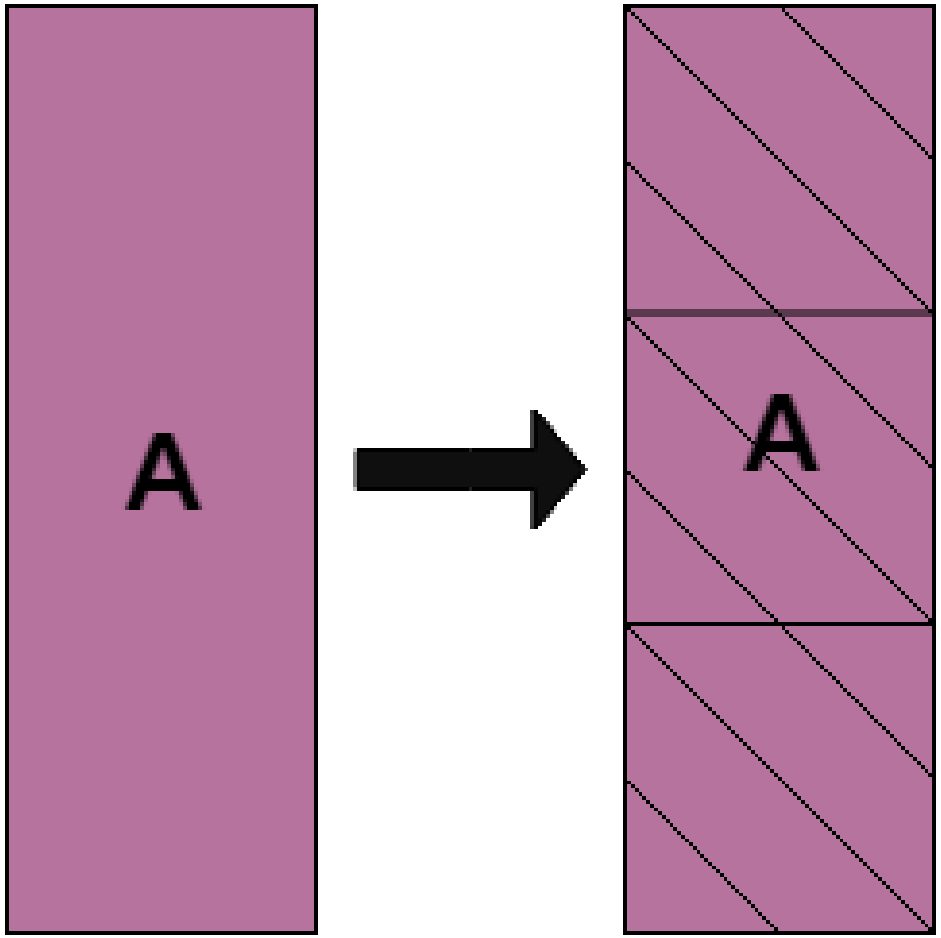


Why algebraic variants of LWE?



✓ less memory

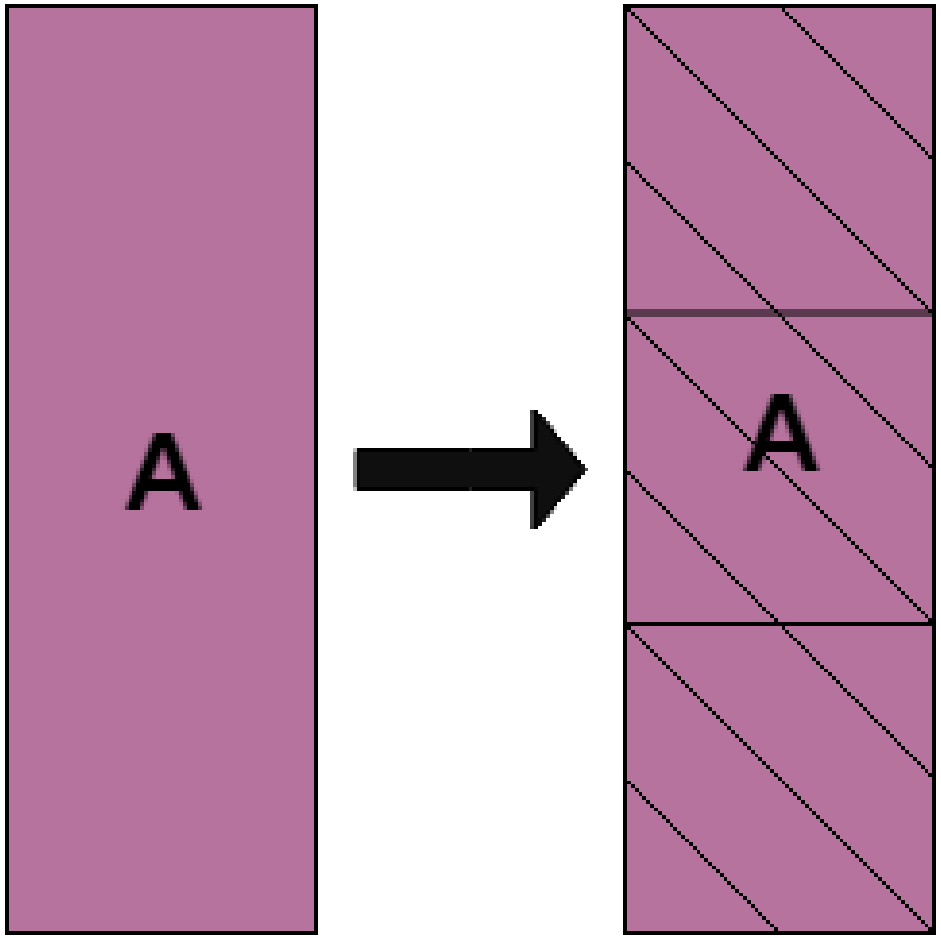
Why algebraic variants of LWE?



✓ less memory

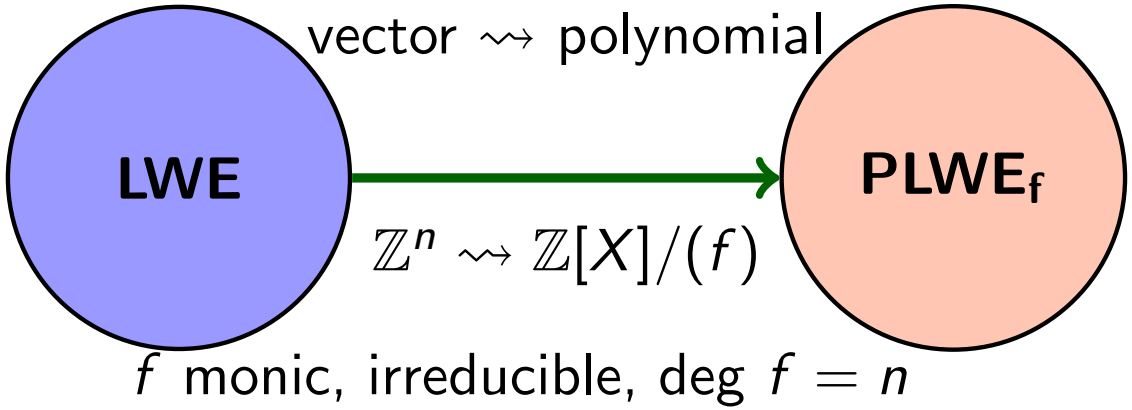
✓ faster operations

Why algebraic variants of LWE?



✓ less memory ✓ faster operations ✓ still hard

Polynomial Learning With Errors (PLWE)



example: $f = X^4 + 1$

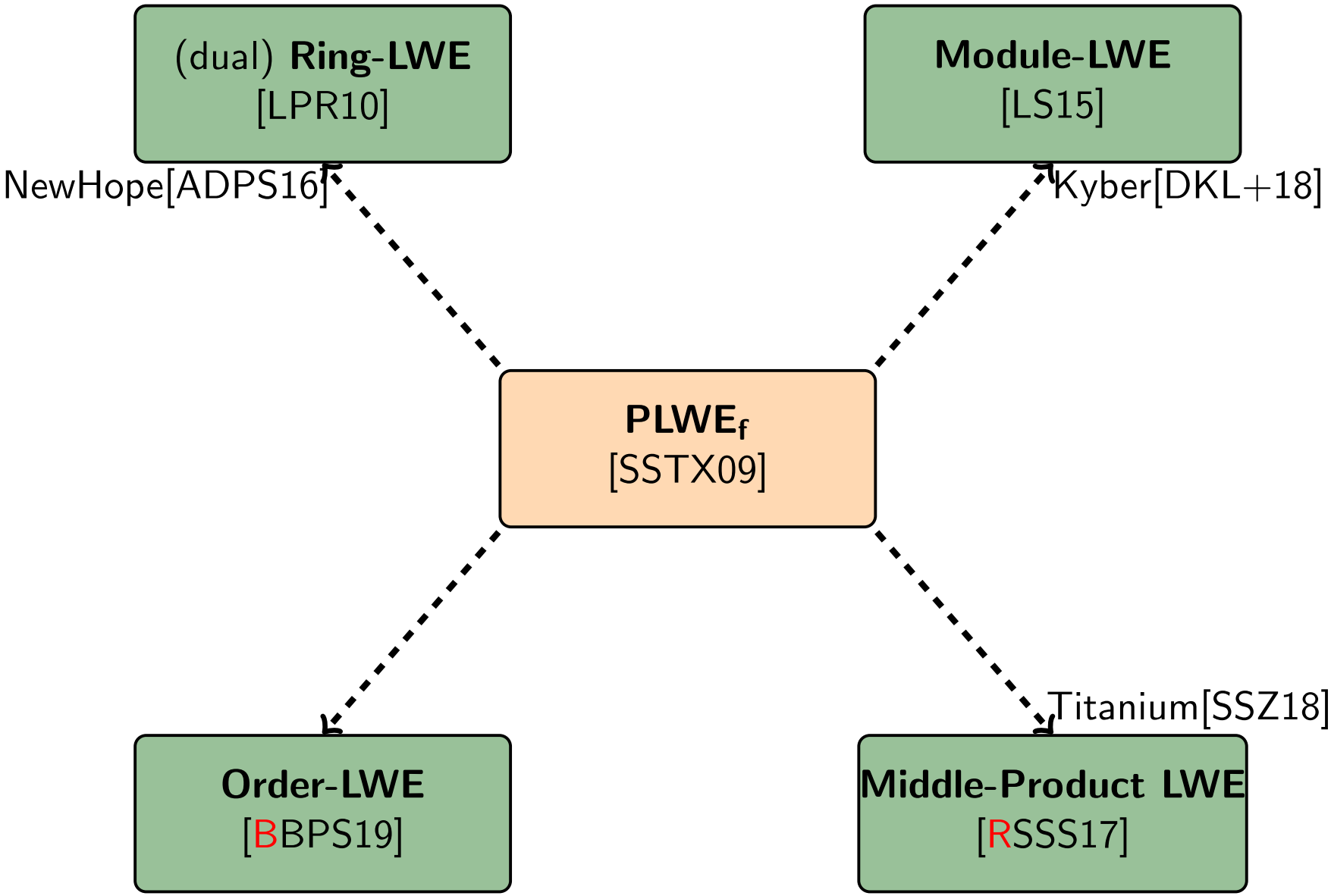
vectors/matrices

$$\begin{pmatrix} a_0 & -a_3 & -a_2 & -a_1 \\ a_1 & a_0 & -a_3 & -a_2 \\ a_2 & a_1 & a_0 & -a_3 \\ a_3 & a_2 & a_1 & a_0 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} + \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix}$$

polynomials

$$\mathbf{a} \cdot \mathbf{s} + \mathbf{e} \bmod f$$

More and more algebraic variants of LWE



The algebraic zoo

Thank you.

References

- [ADPS16]: E. Alkim, L. Ducas, T. Pöppelmann, P. Schwabe, *Post-quantum Key Exchange - A New Hope*. In *Proc. of* **USENIX**, **2016**
- [BDK+18]: J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, *CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM*. In **Euro S&P**, **2018**
- [LPR10]: V. Lyubashevsky, C. Peikert, O. Regev, *On Ideal Lattices and Learning with Errors over Rings*. In **JACM**, **2013**
- [LS15]: A. Langlois, D. Stehlé, *Worst-case to average-case reductions for module lattices*. In **Des. Codes Cryptography**, **2015**
- [Regev05]: O. Regev, *On lattices, learning with errors, random linear codes and cryptography*. In *Proc. of* **STOC**, **2005**
- [SSTX09]: D. Stehlé, R. Steinfeld, K. Tanaka, K. Xagawa, *Efficient Public Key Encryption Based on Ideal Lattices*. In *Proc. of* **ASIACRYPT**, **2009**
- [SSZ18]: R. Steinfeld, A. Sakzad, R. K. Zhao, *Proposal for a NIST Post-Quantum Public-key Encryption and KEM Standard*. Available at http://users.monash.edu.au/∼rste/Titanium_NISTSub.pdf, **2018**

- [ALMṬ20]: S. Agrawal, B. Libert, M. Maitra, R. Ṭiṭiu, *Adaptive Simulation Security for Inner Product Functional Encryption*. Accepted at **PKC**, **2020**
- [BDHR+20]: S. Bai, D. Das, R. Hiromasa, M. Roșca, A. Sakzad, D. Stehlé, R. Steinfeld, Z. Zhang, *MPSign: A Signature from Small-Secret Middle-Product Learning with Errors*. Accepted at **PKC**, **2020**
- [Bol18]: M. Bolboceanu, *Relating different Polynomial-LWE problems*. In *Proc. of* **SecITC**, **2018**
- [BBPS19]: M. Bolboceanu, Z. Brakerski, R. Perlman, D. Sharma, *Order-LWE and the Hardness of Ring-LWE with Entropic Secrets*. In *Proc. of* **ASIACRYPT**, **2019**
- [LSṬ18]: B. Libert, D. Stehlé, R. Ṭiṭiu, *Adaptively Secure Distributed PRFs from LWE*. In *Proc. of* **TCC**, **2018**
- [LṬ19]: B. Libert, R. Ṭiṭiu, *Multi-Client Functional Encryption for Linear Functions in the Standard Model from LWE*. In *Proc. of* **ASIACRYPT**, **2019**
- [RSSS17]: M. Roșca, A. Sakzad, D. Stehlé, R. Steinfeld, *Middle-Product Learning With Errors*. In *Proc. of* **CRYPTO**, **2017**
- [RSW18]: M. Roșca, D. Stehlé, A. Wallet, *On the Ring-LWE and Polynomial-LWE problems*. In *Proc. of* **EUROCRYPT**, **2018**