Lattice Cryptography in the NIST Standardization Process

Vadim Lyubashevsky

IBM Research – Zurich

Hard Problem Intuition

$$\begin{pmatrix} \mathbf{A} \end{pmatrix} \begin{bmatrix} \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{z} \end{bmatrix}$$

Given (**A**,**z**), find **y**

Easy! Just invert **A** and multiply by **z**

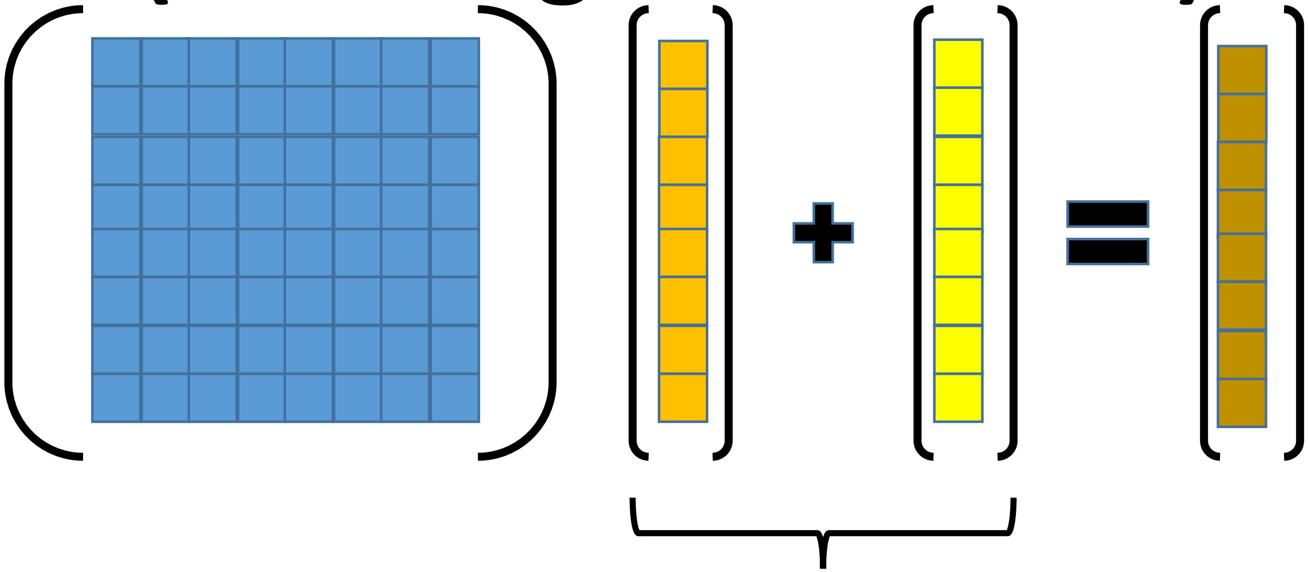
Hard Problem Intuition

$$\begin{pmatrix} \quad \mathbf{A} \quad \end{pmatrix} \underbrace{\begin{pmatrix} \mathbf{y} \end{pmatrix} + \begin{pmatrix} \mathbf{e} \end{pmatrix}}_{\text{Small coefficients}} = \begin{pmatrix} \mathbf{z} \end{pmatrix}$$

Given (**A**,**z**), find (**y**,**e**)

Seems hard.

Hard Problem Intuition (Learning With Errors)



Small coefficients to enforce uniqueness

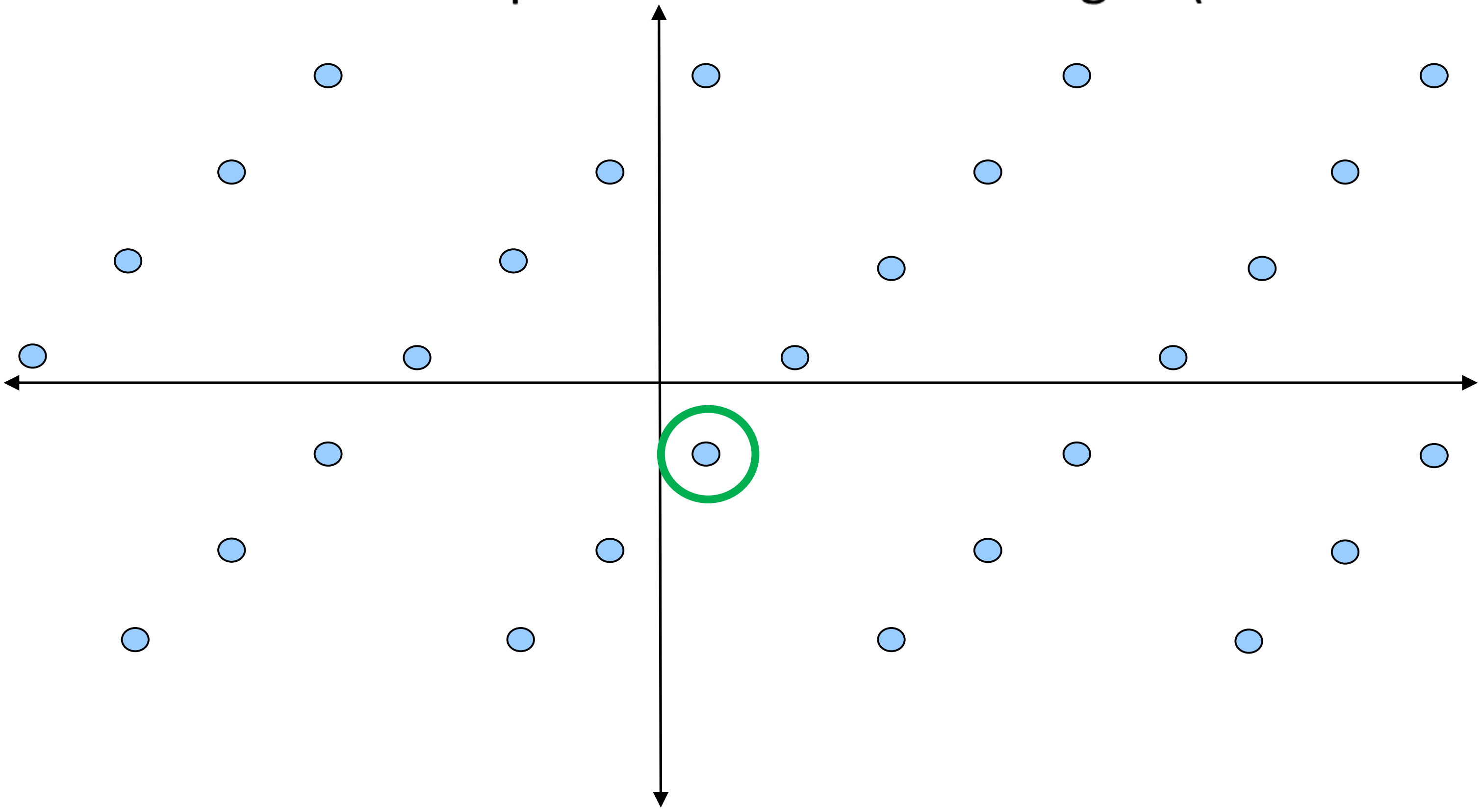
Given (**A**,**z**), find (**y**,**e**)

Seems hard.

Why is this “Lattice” Crypto?

All solutions $\binom{y}{e}$ to **A**y+e=**z** mod p form a “shifted” lattice.

We want to find the point closest to the origin (BDD Problem).

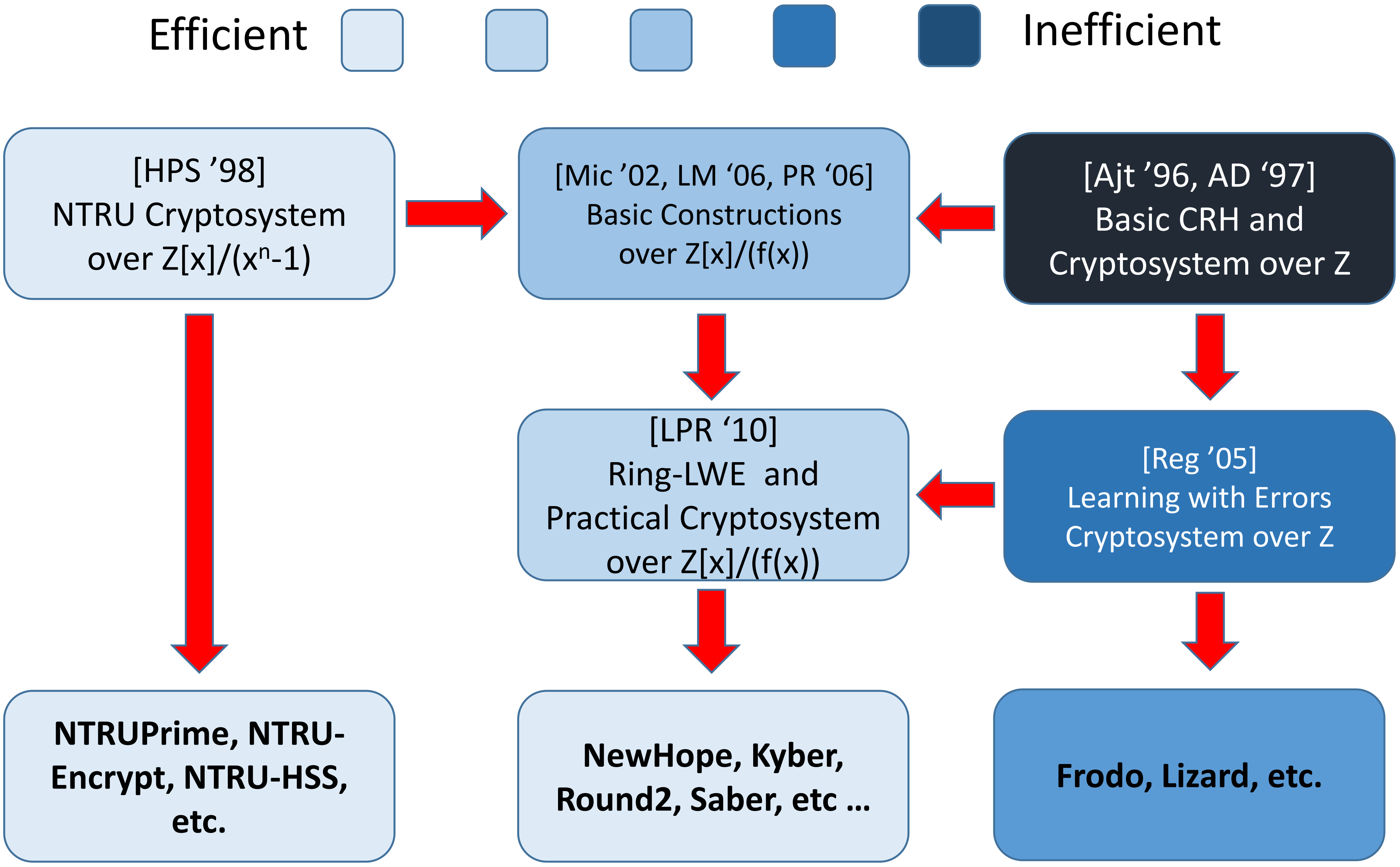


Connection to Lattices

- Solving a Lattice-Problem → Breaking Cryptosystems
- Breaking Cryptosystems → Solving a Lattice Problem in **all** lattices
  - Worst-Case to Average-Case Reduction [Ajt ‘96, Reg ‘05, etc.]
  - **Asymptotically,** the design of lattice-based schemes is sound

Lattice-Based Encryption

Encryption Scheme Overview



Efficient ▢ ▢ ▢ ■ ■ Inefficient

[HPS ’98]
NTRU Cryptosystem
over Z[x]/(x^n-1)

[Mic ’02, LM ‘06, PR ‘06]
Basic Constructions
over Z[x]/(f(x))

[Ajt ’96, AD ‘97]
Basic CRH and
Cryptosystem over Z

[LPR ‘10]
Ring-LWE and
Practical Cryptosystem
over Z[x]/(f(x))

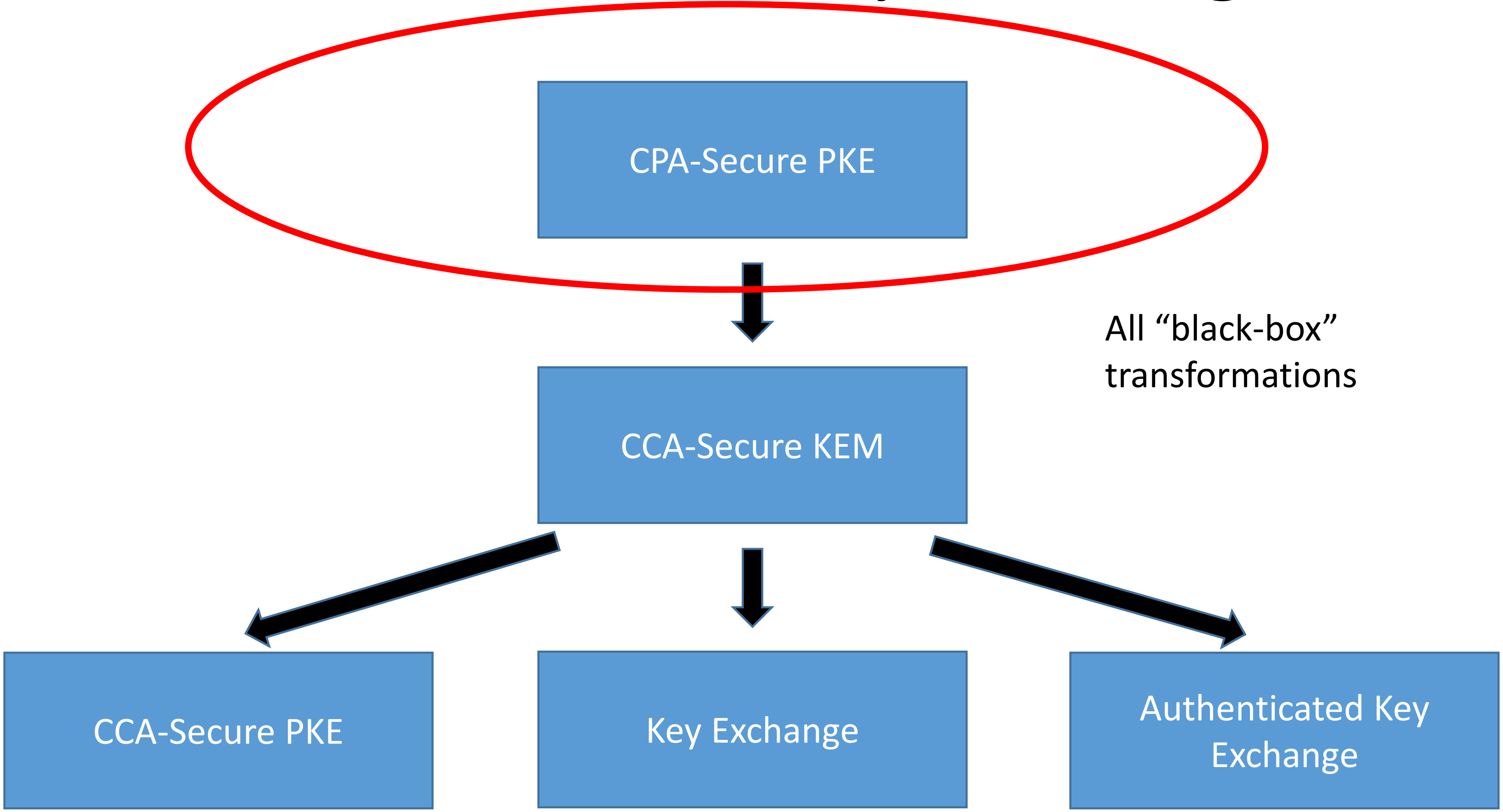
[Reg ’05]
Learning with Errors
Cryptosystem over Z

NTRUPrime, NTRU-Encrypt, NTRU-HSS, etc.

NewHope, Kyber, Round2, Saber, etc …

Frodo, Lizard, etc.

Key Exchange / CCA – Encryption/ Authenticated Key Exchange



CPA-Secure PKE

CCA-Secure KEM

All “black-box” transformations

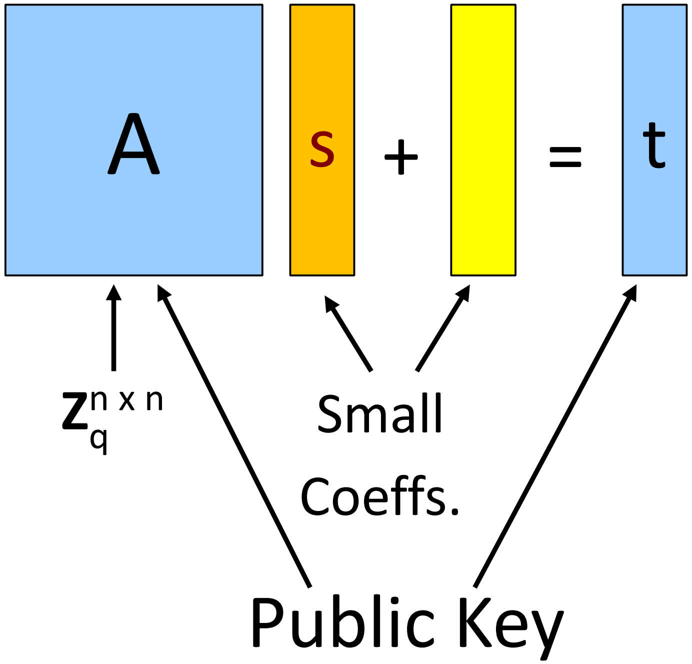
CCA-Secure PKE

Key Exchange

Authenticated Key Exchange

Encryption from LWE

Encryption Scheme



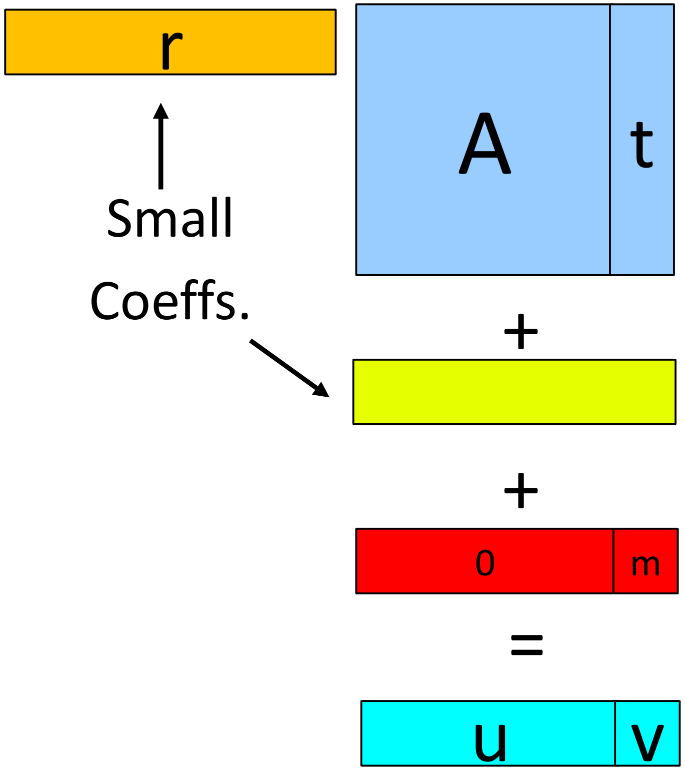
A s + = t

$\mathbf{Z}_q^{n \times n}$

Small

Coeffs.

Public Key



r

Small

Coeffs.

A t

+

+

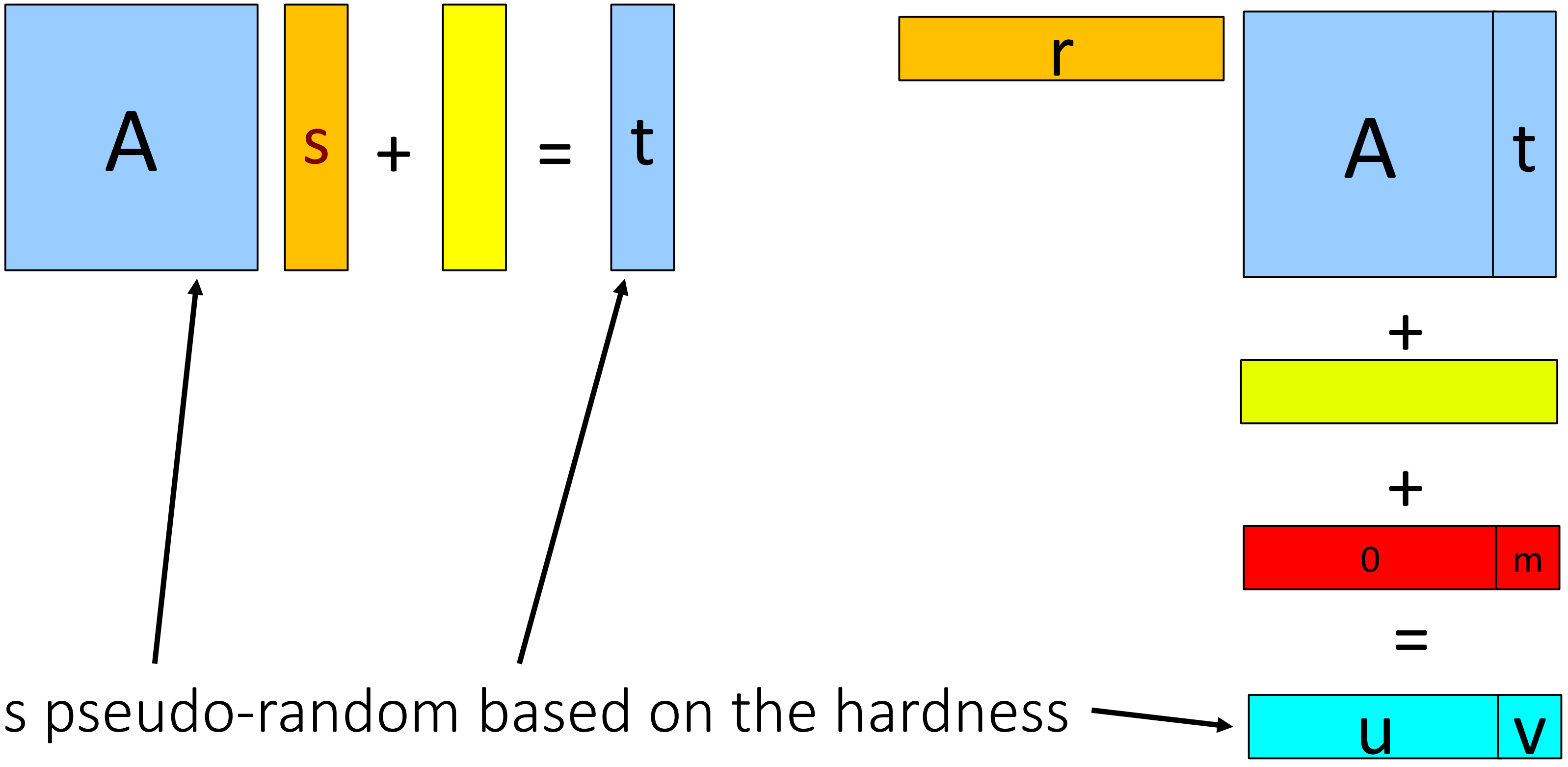
0 m

=

u v

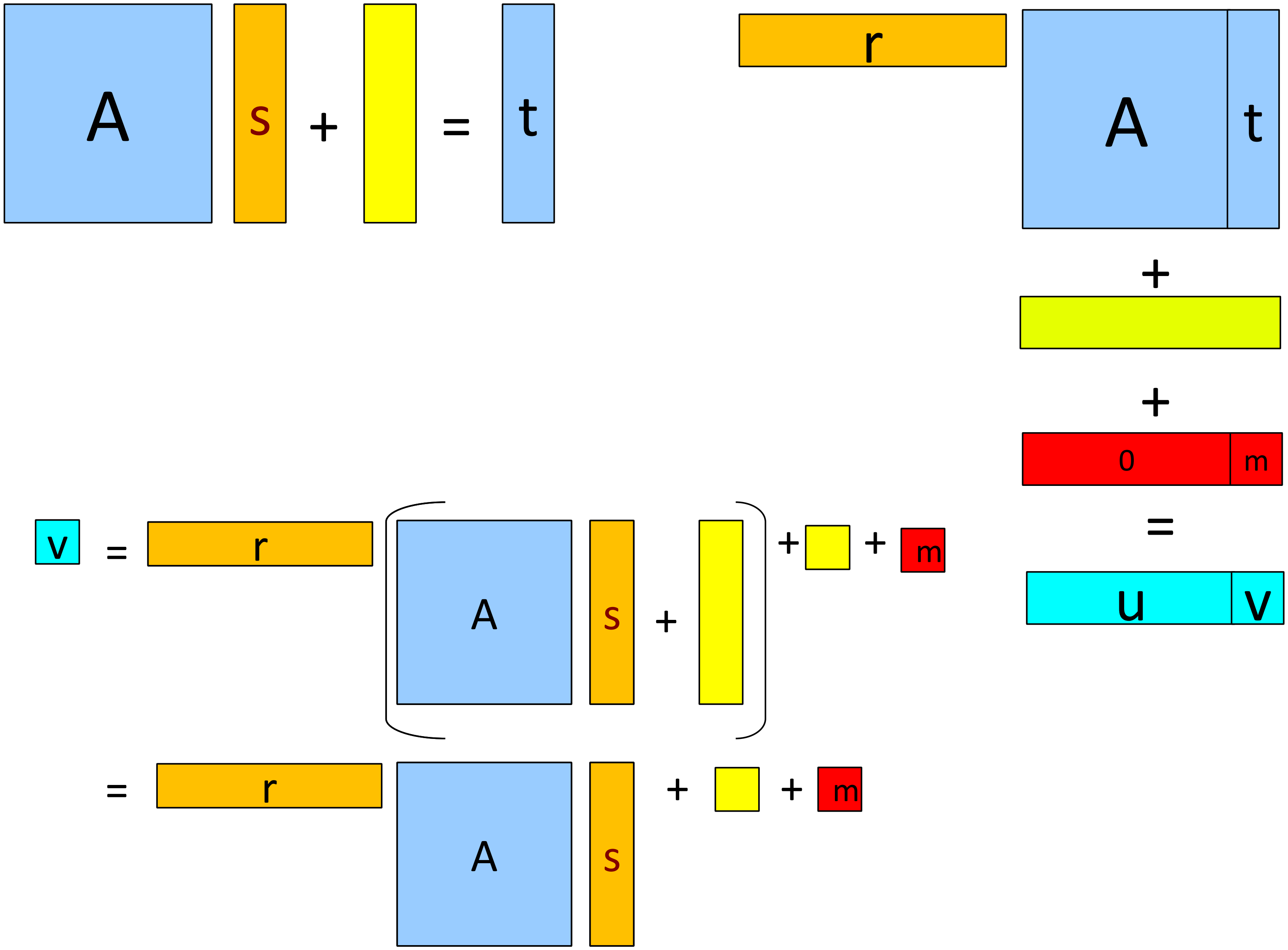
A is random – can be created as H(seed)

Encryption Scheme

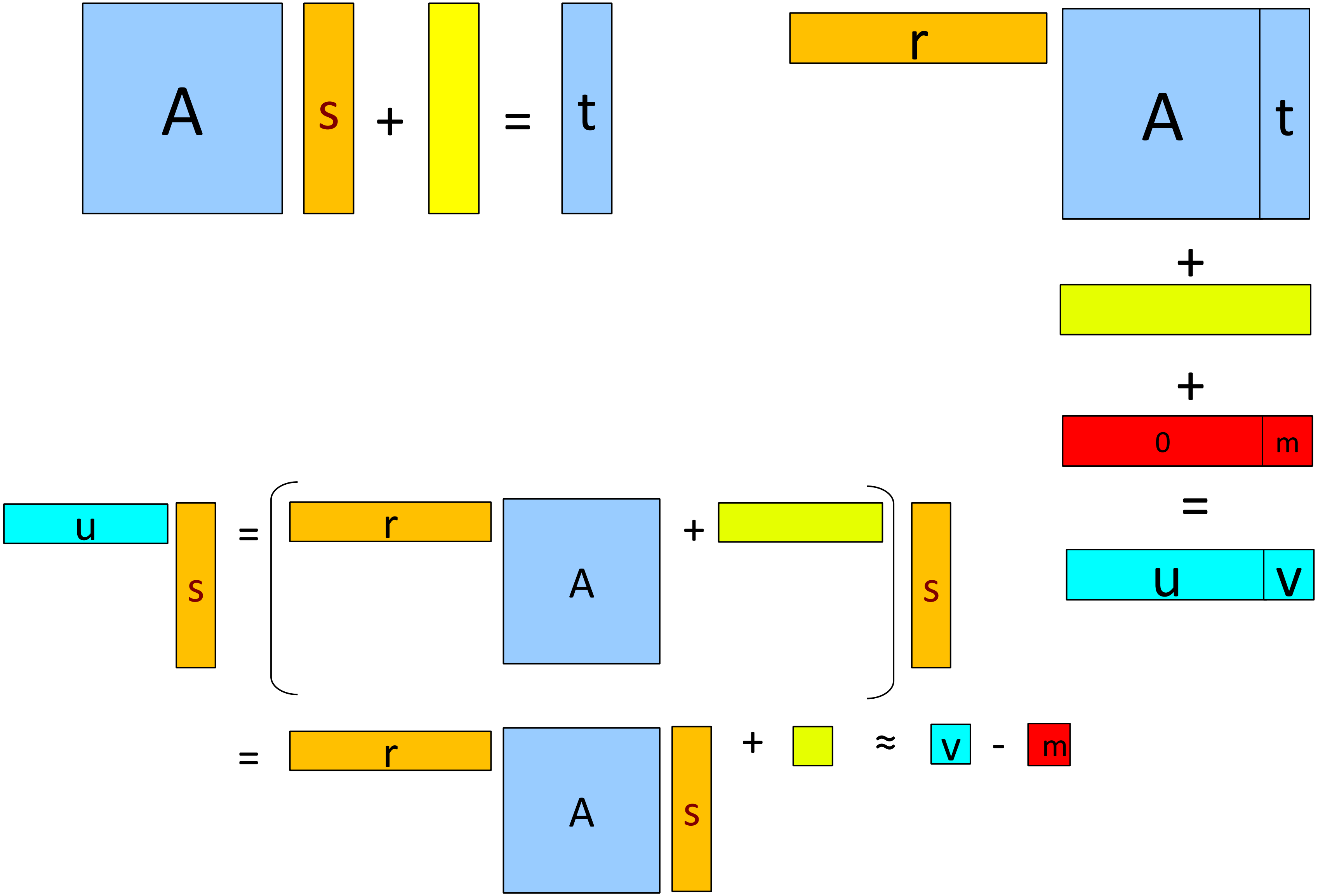


Is pseudo-random based on the hardness of the Learning with Errors Problem

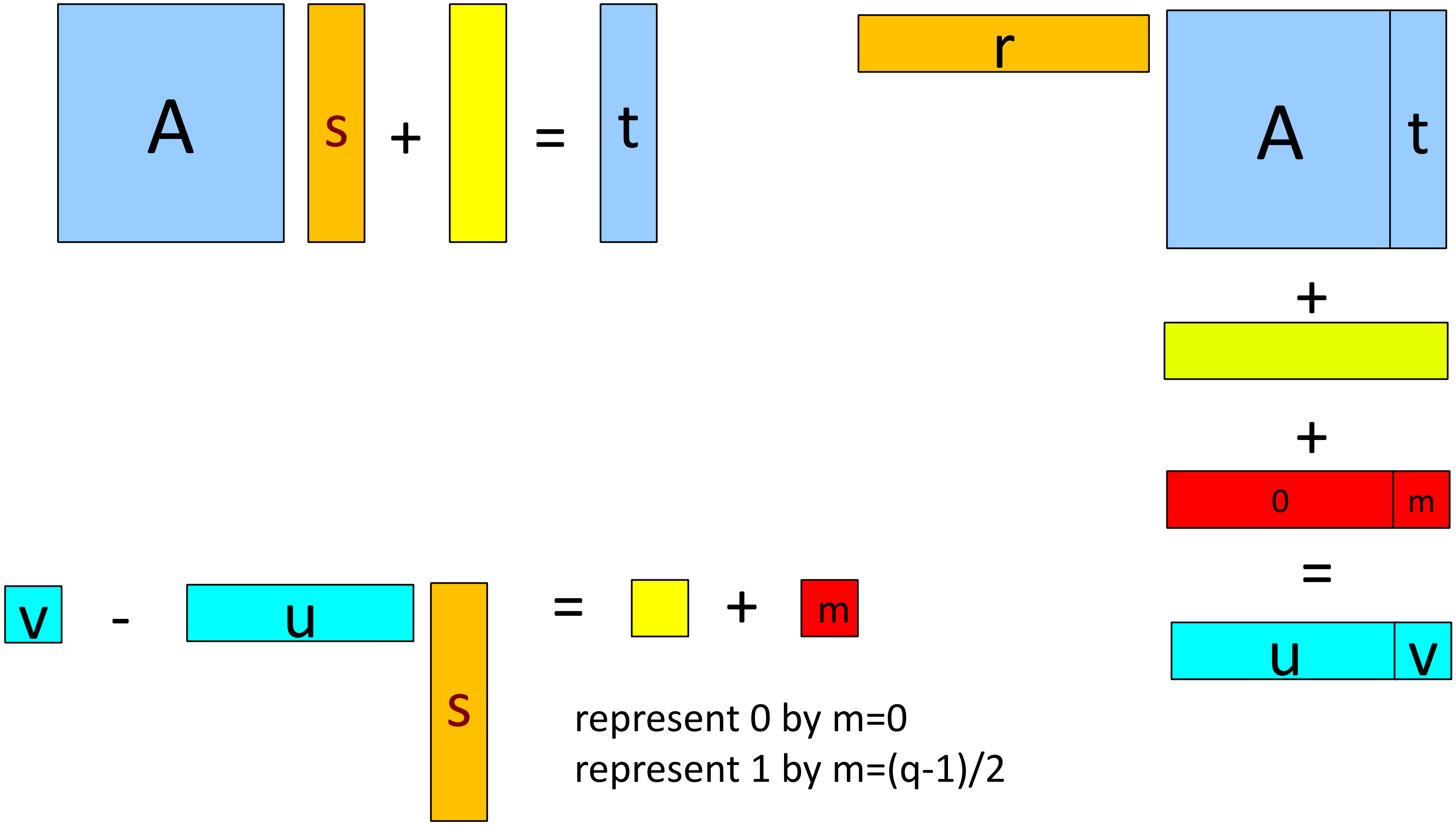
Encryption Scheme



Encryption Scheme



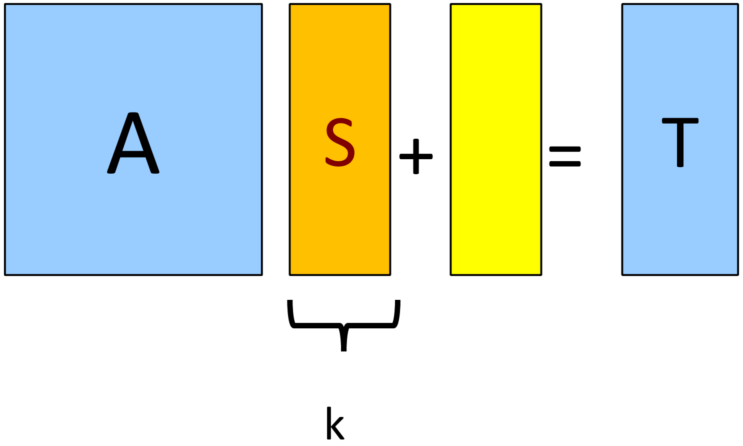
Encryption Scheme

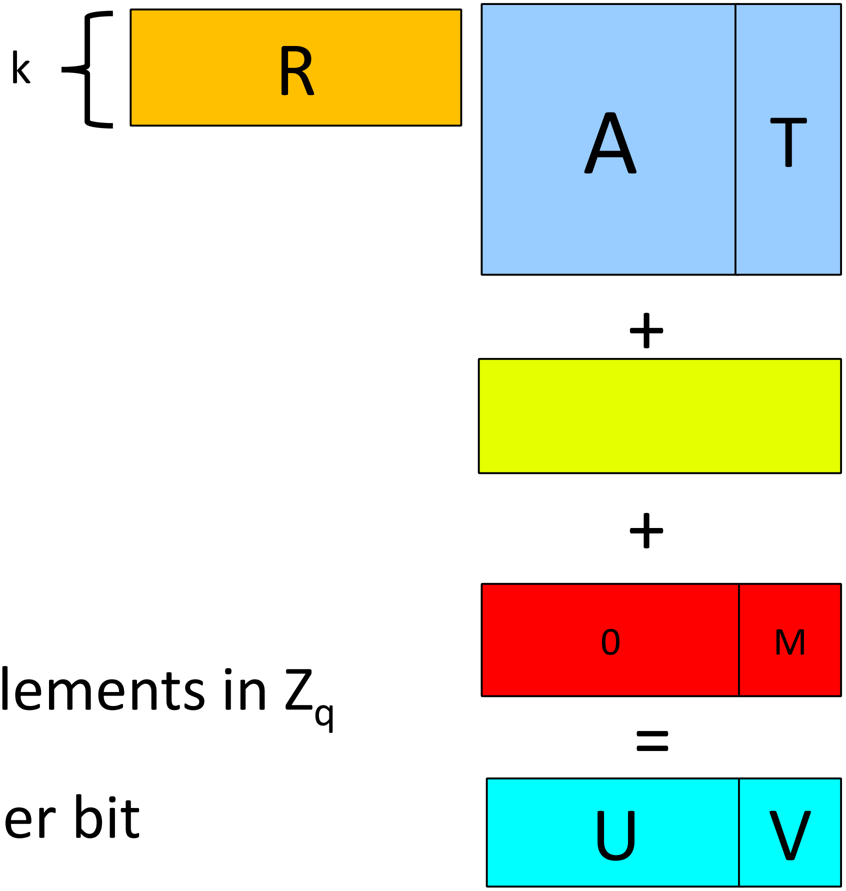


represent 0 by m=0
represent 1 by m=(q-1)/2

Encrypts only 1 bit – large ciphertext expansion
1 bit requires n elements in Z_q

Encrypting More Bits

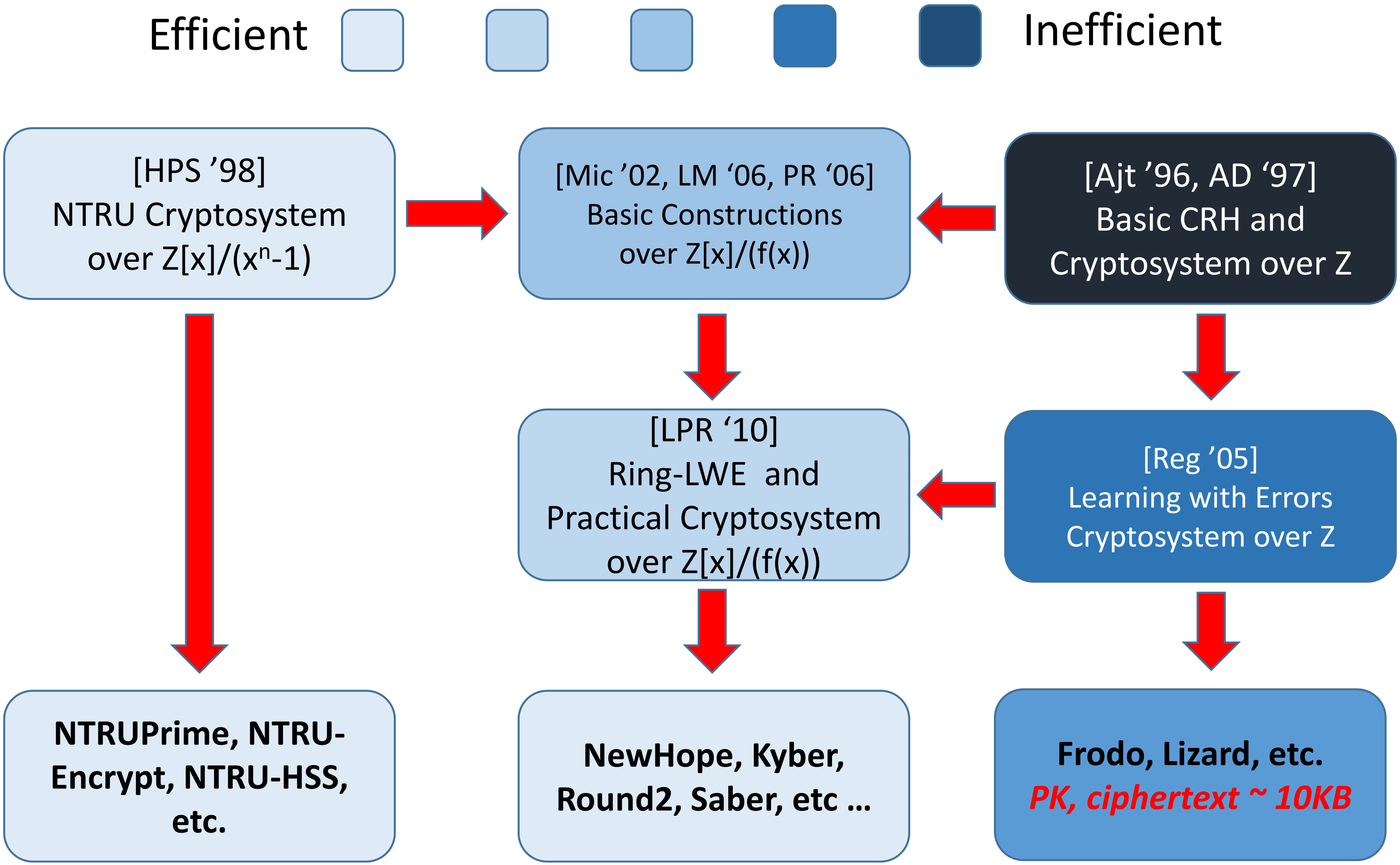




Encrypting k^2 bits requires nk elements in Z_q

i.e. n/k elements in Z_q per bit

Encryption Scheme Overview



Efficient ☐ ☐ ☐ ■ ■ Inefficient

[HPS ’98]
NTRU Cryptosystem
over Z[x]/(x^n-1)

[Mic ’02, LM ‘06, PR ‘06]
Basic Constructions
over Z[x]/(f(x))

[Ajt ’96, AD ‘97]
Basic CRH and
Cryptosystem over Z

[LPR ‘10]
Ring-LWE and
Practical Cryptosystem
over Z[x]/(f(x))

[Reg ’05]
Learning with Errors
Cryptosystem over Z

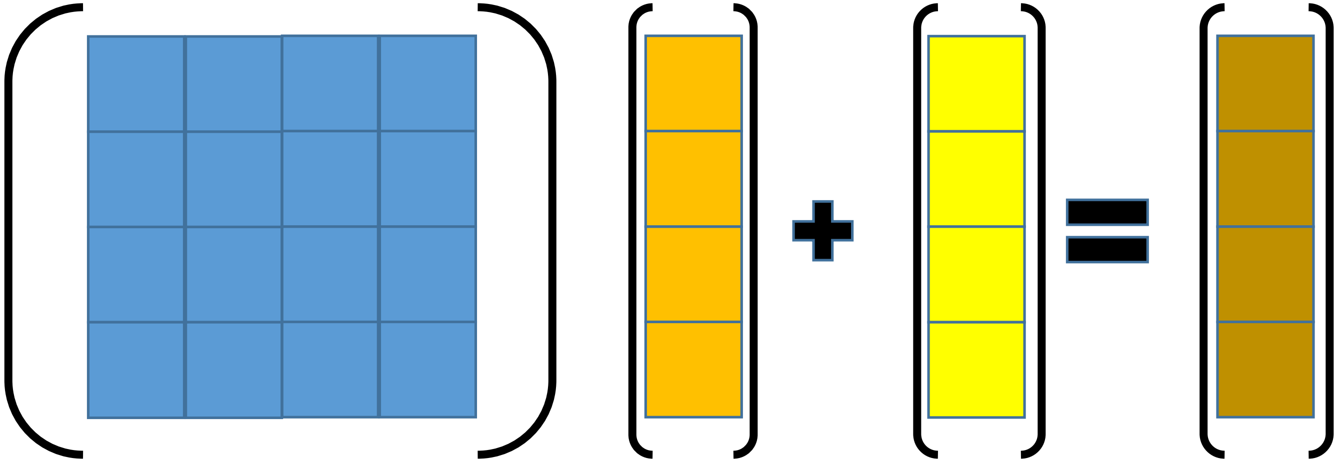
NTRUPrime, NTRU-Encrypt, NTRU-HSS, etc.

NewHope, Kyber, Round2, Saber, etc …

Frodo, Lizard, etc.
PK, ciphertext ~ 10KB

Encryption from (polynomial / ring / generalized / module)-LWE

Hard Problem Intuition (Generalized / Ring / Module-LWE)



Use Polynomial Rings Instead of Integers

Example Ring $Z_{17}[x]/(x^4+1)$

Elements are $z(x)=z_3x^3+z_2x^2+z_1x+z_0$

where z_i are integers mod 17

Addition is the usual coordinate-wise addition

Multiplication is the usual polynomial multiplication followed by reduction modulo x^4+1

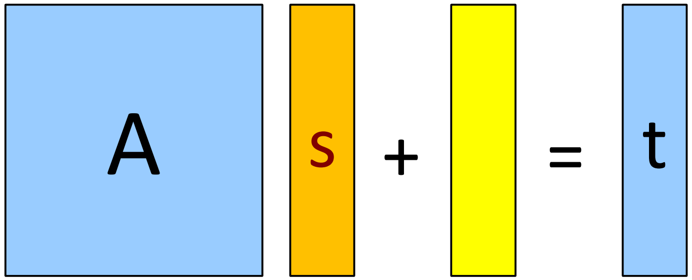
Example Ring $Z_{17}[x]/(x^4+1)$

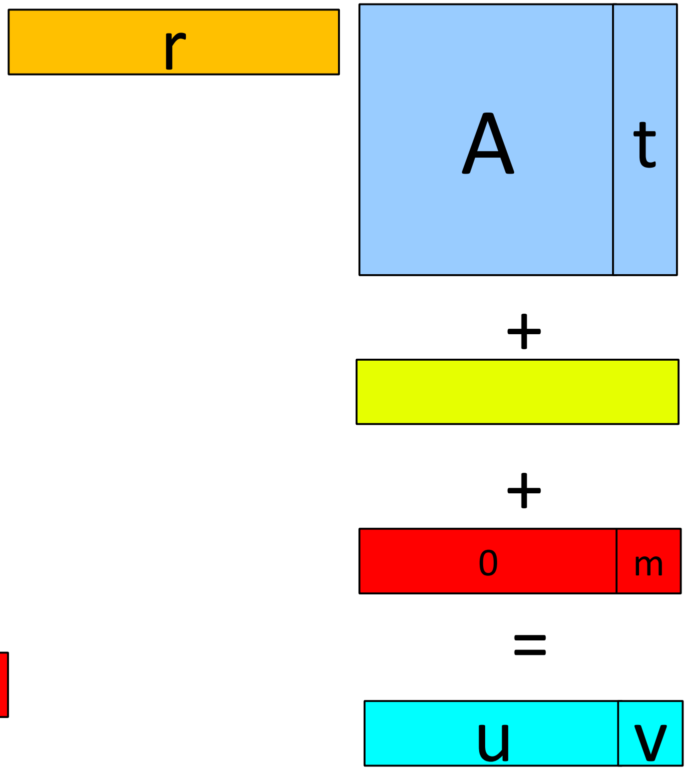
$$(X^3 - 2X - 1)(-3X^2 + 6) = (-3X^5 + 12X^3 + 3X^2 - 12X - 6)$$
$$= (3X + 12X^3 + 3X^2 - 12X - 6)$$
$$= (-5X^3 + 3X^2 + 8X - 6)$$

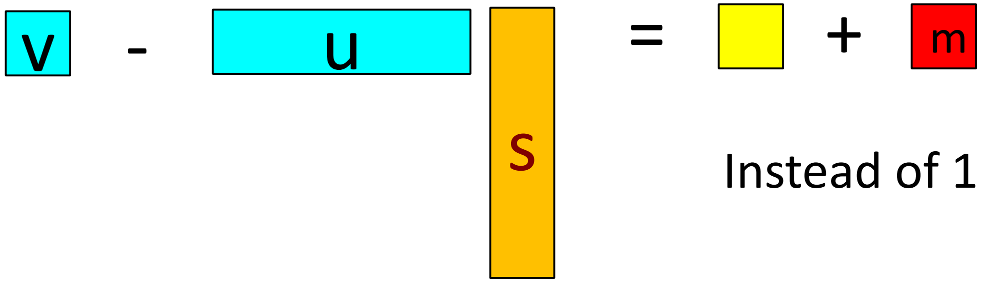
Important: Reductions modulo X^4+1 do not increase the coefficients!

(For some moduli, there could be an exponential increase – these are not useful for crypto).

Encrypting More Bits





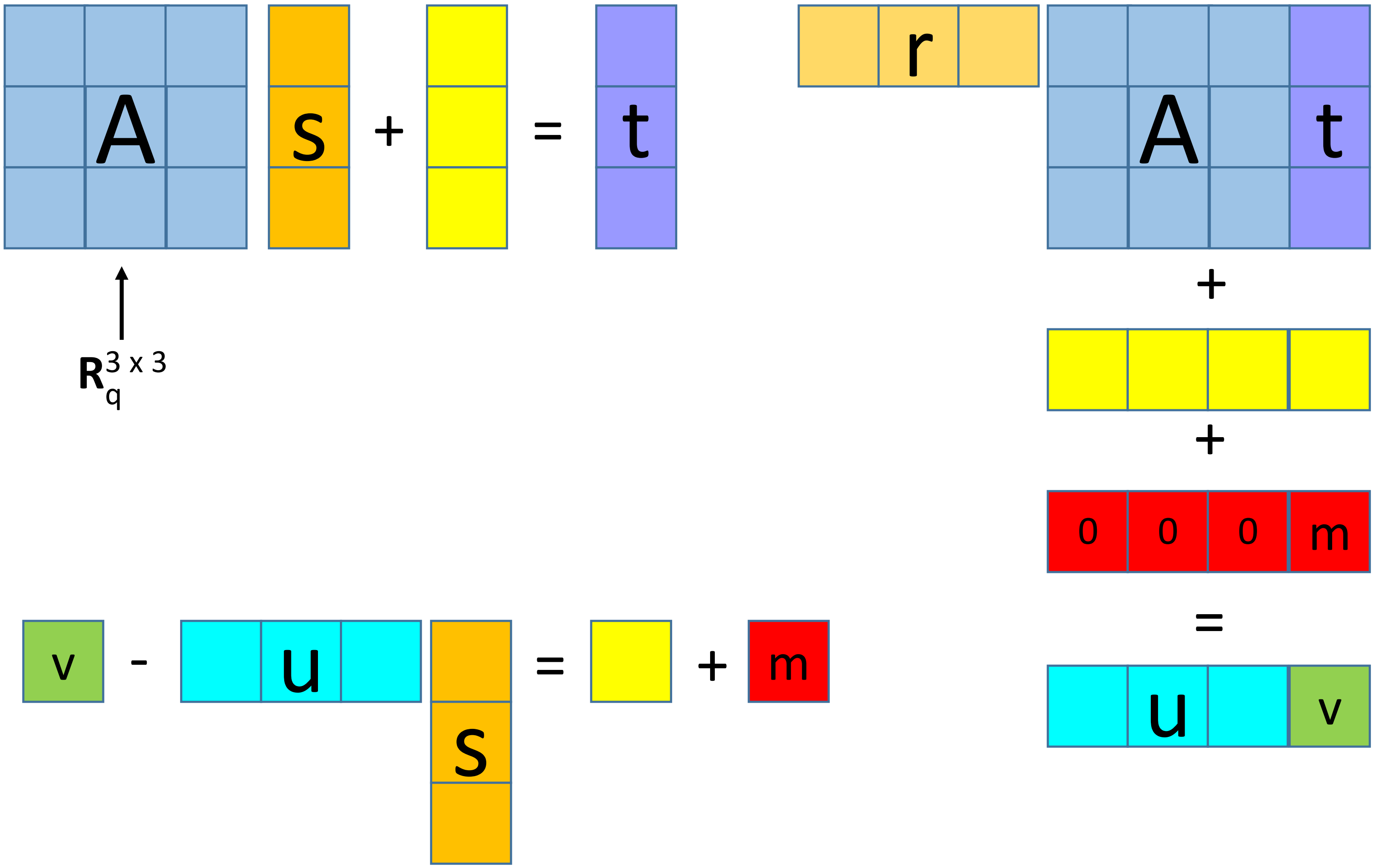


Instead of 1 element in Z, make it 1 element in Z[X]/(X^d+1)

i.e. work over **R**=Z_q[X]/(X^d+1) instead of Z_q

An encryption of d integers.

Encryption Scheme Over Polynomial Rings

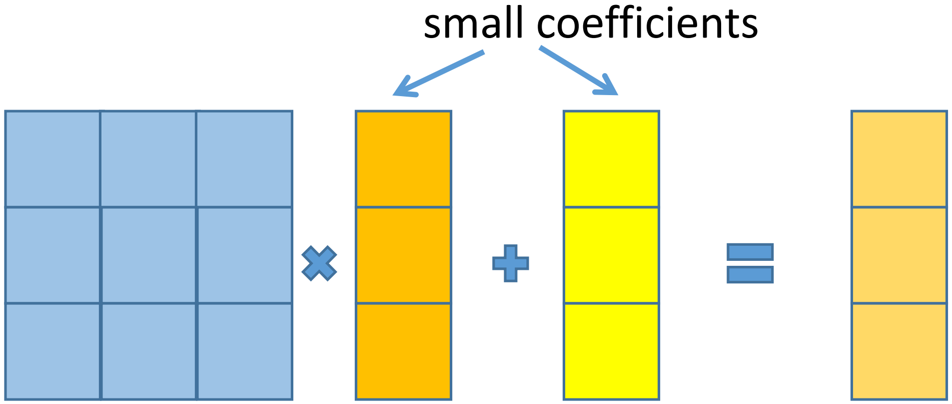


Operations in CRYSTALS
(our lattice suite submission to NIST)

Basic Computational Domain:

Polynomial ring $Z_p[x]/(x^{256}+1)$ 🟦

Operations used in the schemes: + and × in the ring:



Operations in CRYSTALS

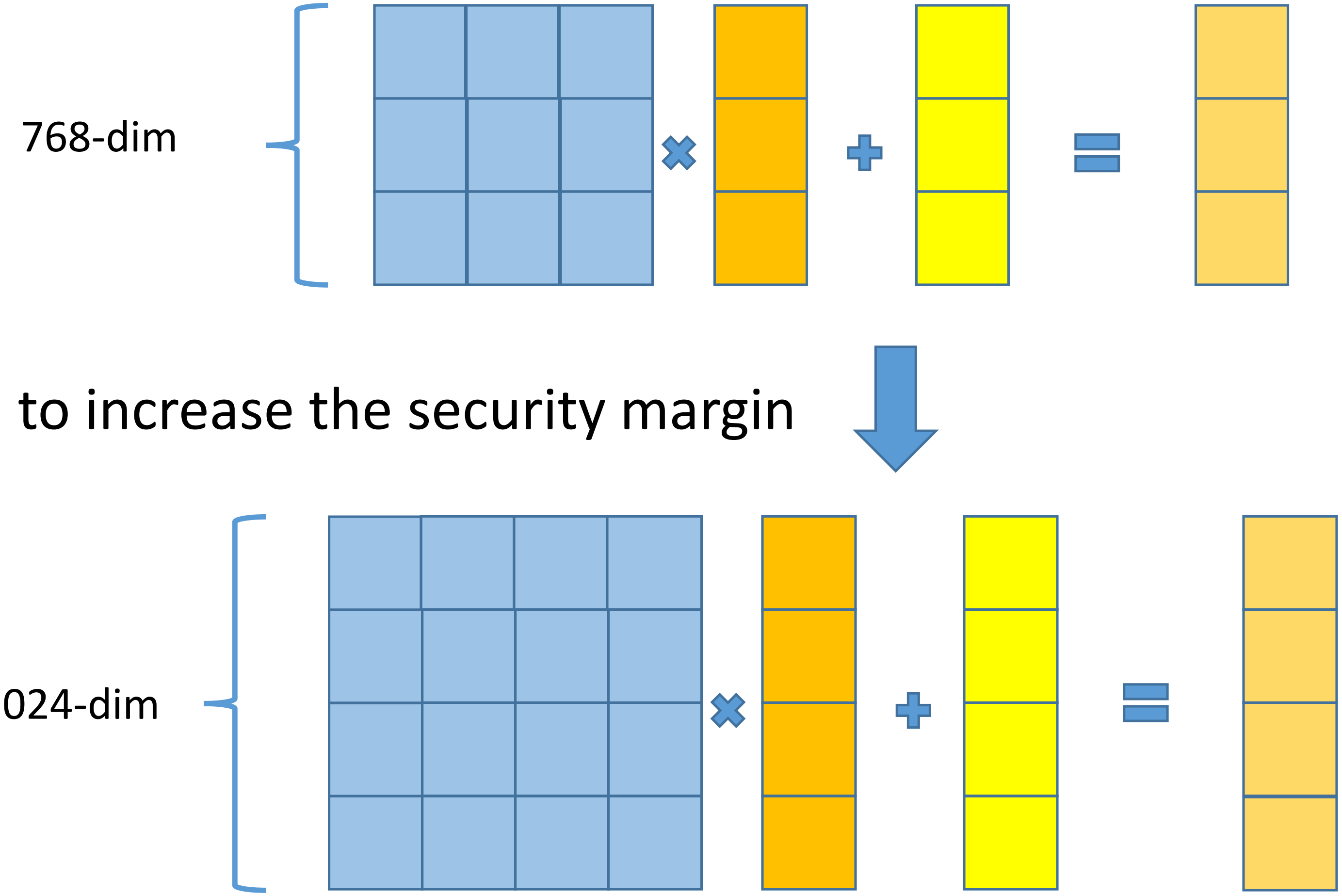
Only two main operations needed (and both are very fast):

1. Evaluations of SHAKE (can use another XOF too)
2. Add / multiply in the polynomial ring $Z_p[X]/(X^{256}+1)$
    - $p = 2^{13} – 2^9 + 1$ (for KEM / Encryption Kyber)
    - $p = 2^{23} – 2^{13} + 1$ (for Signature Dilithium)

To increase security, just do more of the same operations

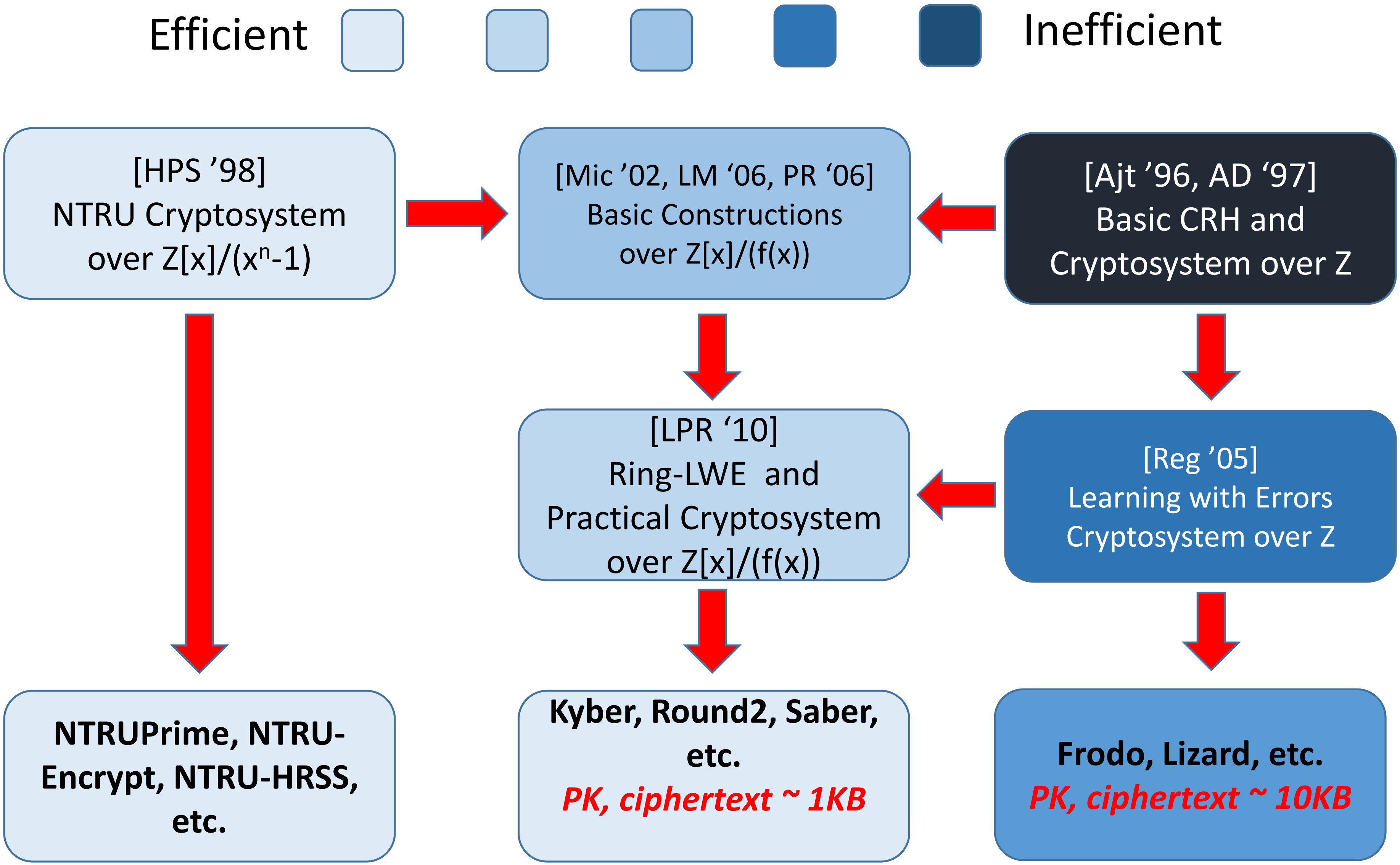
The exact same hardware/software can be reused

Modular security



to increase the security margin

Encryption Scheme Overview



Efficient □ □ □ ■ ■ Inefficient

[HPS ’98]
NTRU Cryptosystem
over Z[x]/(x^n-1)

[Mic ’02, LM ‘06, PR ‘06]
Basic Constructions
over Z[x]/(f(x))

[Ajt ’96, AD ‘97]
Basic CRH and
Cryptosystem over Z

[LPR ‘10]
Ring-LWE and
Practical Cryptosystem
over Z[x]/(f(x))

[Reg ’05]
Learning with Errors
Cryptosystem over Z

NTRUPrime, NTRU-Encrypt, NTRU-HRSS, etc.

Kyber, Round2, Saber, etc.
PK, ciphertext ~ 1KB

Frodo, Lizard, etc.
PK, ciphertext ~ 10KB

NTRU Encryption

NTRU Cryptosystem

$\boxed{f}$ $\boxed{g}$ Small secret keys

$$\frac{\boxed{f}}{\boxed{g}} = \boxed{a} \bmod p$$

“looks” random

$$\boxed{u} = 2\left[\boxed{a}\,\boxed{r} + \boxed{}\right] + \boxed{m} \bmod p$$

If a is random, then pseudorandom based on Ring-LWE

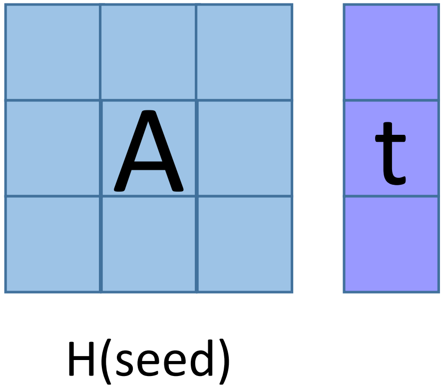
$$\boxed{u}\,\boxed{g} = 2\left[\boxed{f}\,\boxed{r} + \boxed{}\,\boxed{g}\right] + \boxed{g}\,\boxed{m}$$

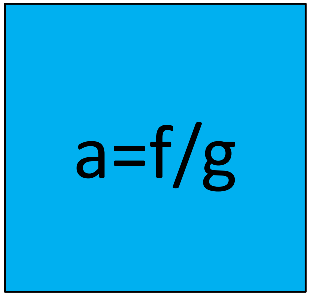
$$\boxed{u}\,\boxed{g} \bmod 2 = \boxed{g}\,\boxed{m}$$

$$\frac{\boxed{u}\,\boxed{g} \bmod 2}{\boxed{g}} = \boxed{m}$$

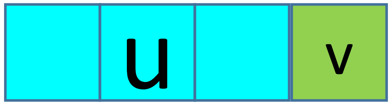
Comparison

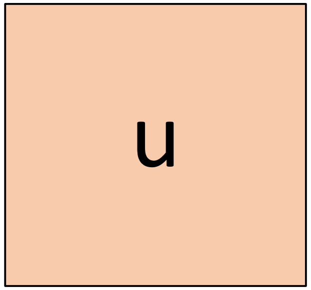
(they're virtually the same)





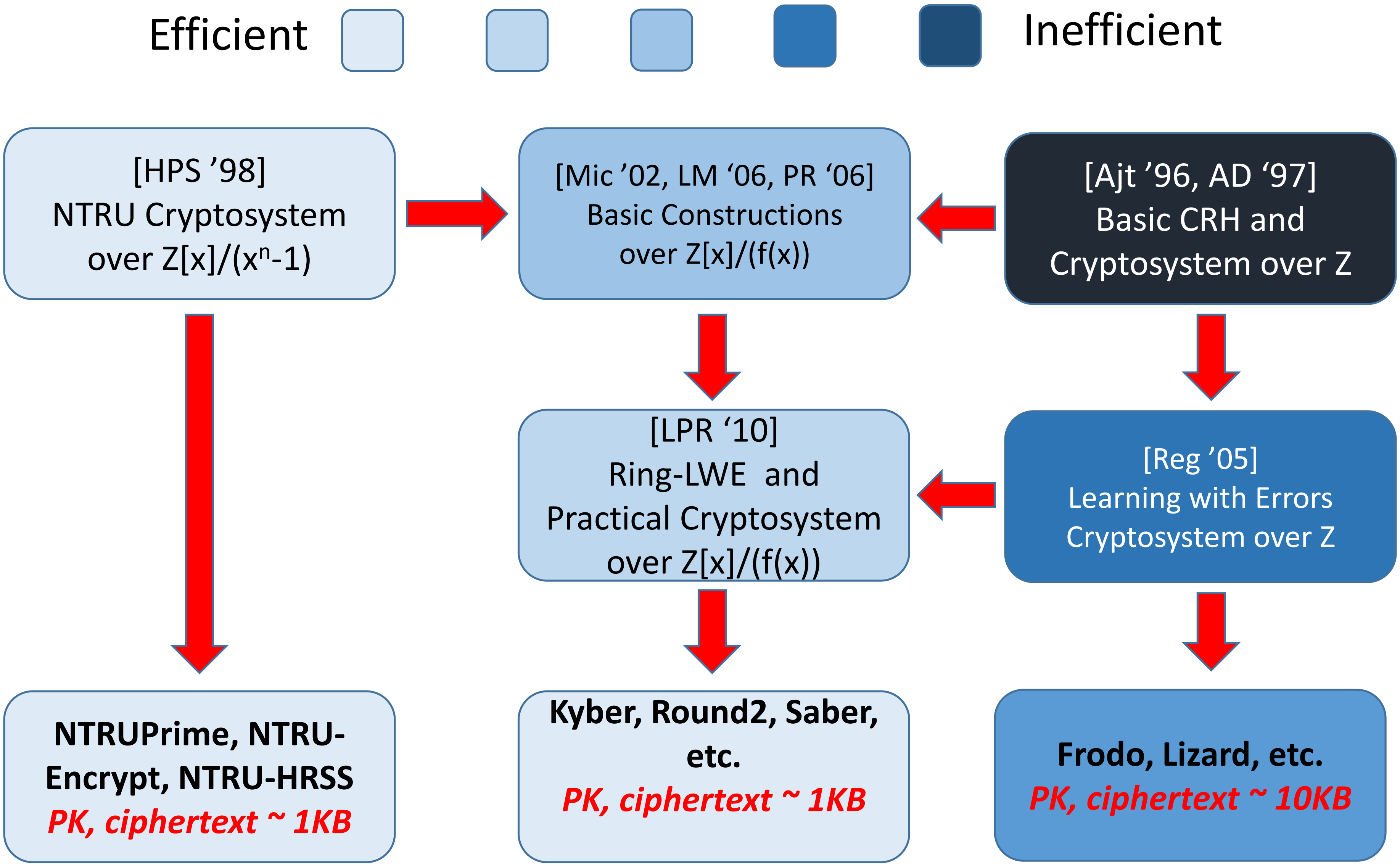
- t and a have the same size. But A=H(seed) needs to be re-generated
- Cannot efficiently make the NTRU public key consist of more polynomials
- f/g may be costlier to compute – makes a difference in ephemeral key exchange





The u have the same size. Only the high-order bits of v need to be transmitted.

Encryption Scheme Overview



Efficient ▢ ▢ ▢ ▢ ■ Inefficient

[HPS '98]
NTRU Cryptosystem
over Z[x]/(x^n-1)

[Mic '02, LM '06, PR '06]
Basic Constructions
over Z[x]/(f(x))

[Ajt '96, AD '97]
Basic CRH and
Cryptosystem over Z

[LPR '10]
Ring-LWE and
Practical Cryptosystem
over Z[x]/(f(x))

[Reg '05]
Learning with Errors
Cryptosystem over Z

NTRUPrime, NTRU-Encrypt, NTRU-HRSS
PK, ciphertext ~ 1KB

Kyber, Round2, Saber, etc.
PK, ciphertext ~ 1KB

Frodo, Lizard, etc.
PK, ciphertext ~ 10KB

Small Variations

Schemes made slightly more efficient by more "aggressive" constructions

- e.g. using secret / noise coefficients in a smaller range
- Instead of adding noise, doing rounding (chopping off bits)

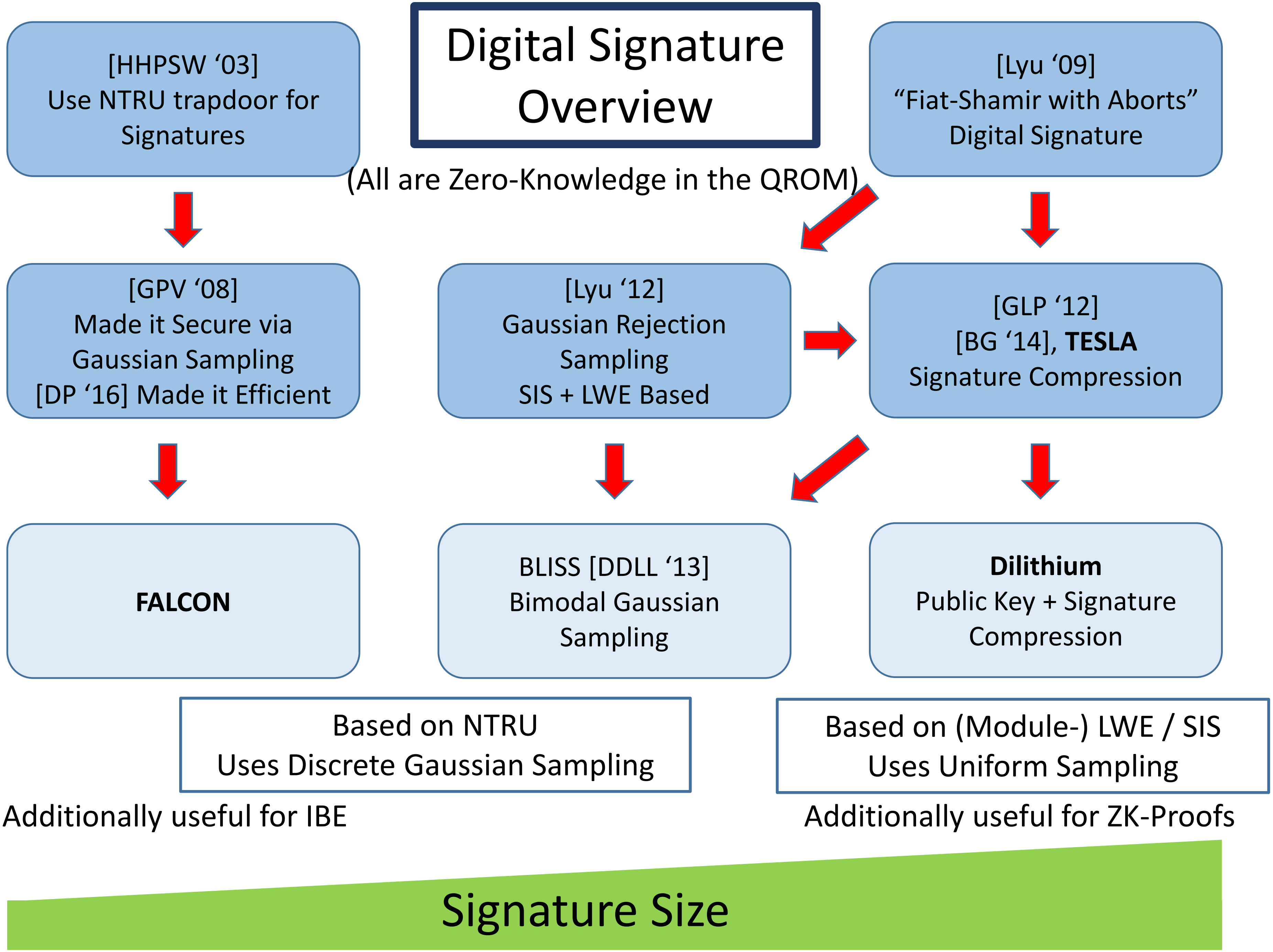
Unclear if there is any security penalty

Analogous to saying:

> "I made SHA-3 more efficient by changing the compression function from 24 to 20 rounds"

Digital Signatures

Digital Signature Overview



(All are Zero-Knowledge in the QROM)

[HHPSW '03]
Use NTRU trapdoor for Signatures

[Lyu '09]
"Fiat-Shamir with Aborts"
Digital Signature

[GPV '08]
Made it Secure via Gaussian Sampling
[DP '16] Made it Efficient

[Lyu '12]
Gaussian Rejection Sampling
SIS + LWE Based

[GLP '12]
[BG '14], **TESLA**
Signature Compression

FALCON

BLISS [DDLL '13]
Bimodal Gaussian Sampling

Dilithium
Public Key + Signature Compression

Based on NTRU
Uses Discrete Gaussian Sampling

Based on (Module-) LWE / SIS
Uses Uniform Sampling

Additionally useful for IBE

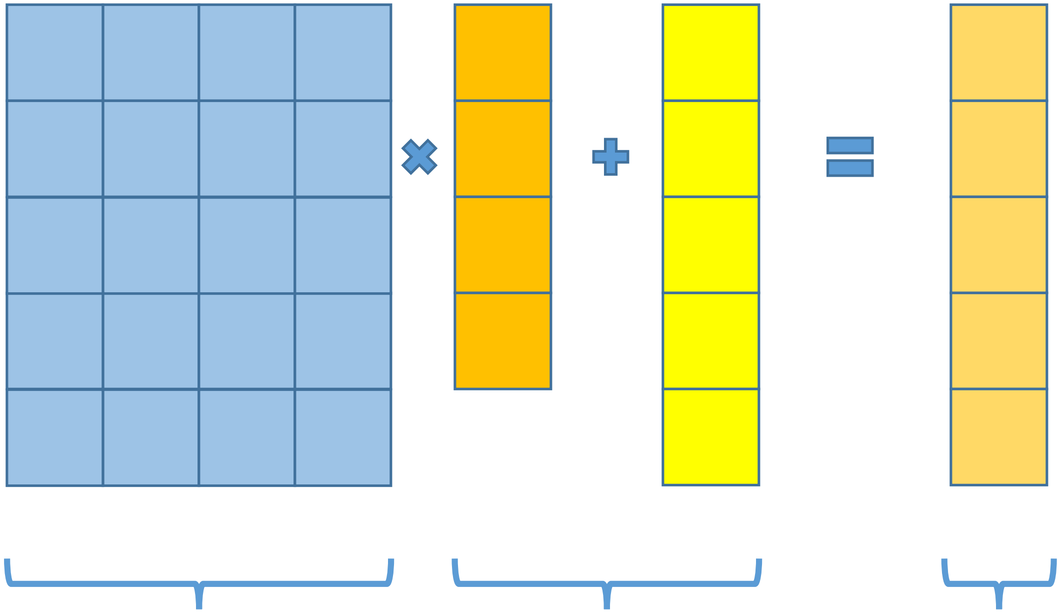
Additionally useful for ZK-Proofs

Signature Size

“Fiat-Shamir with Aborts”

[Lyu ‘09] → … → [BG ‘14]

Public / Secret Keys



Public key
A:=XOF(seed)
uniform mod p

Secret key
$\mathbf{s}_1$, $\mathbf{s}_2$ with
small coefficients

Public key
$\mathbf{t}:=\mathbf{A}\mathbf{s}_1+\mathbf{s}_2$

Signing and Verification

$$\mathbf{A}\mathbf{s}_1+\mathbf{s}_2=\mathbf{t}$$

Sign(μ)

$y \leftarrow$ Coefficients in $[-\gamma, \gamma]$

$c := H(\text{high}(\mathbf{A}y), \mu)$

$z := y + c\mathbf{s}_1$

Needed for security

if $|z| > \gamma - \beta$ or $|\text{low}(\mathbf{A}y - c\mathbf{s}_2)| > \gamma - \beta$

restart

Signature = (z, c)

Verify(z, c, μ)

Check that $|z| \leq \gamma - \beta$

and

$c=H(\text{high}(\mathbf{A}z - c\mathbf{t}), \mu)$

Correct because $\text{high}(\mathbf{A}y) = \text{high}(\mathbf{A}z - c\mathbf{t})$

Removing Low-Order PK bits

$$\mathbf{A}\mathbf{s}_1+\mathbf{s}_2=\mathbf{t}_0+b\mathbf{t}_1$$

Sign(μ)

$\mathbf{y} \leftarrow$ Coefficients in $[-\gamma, \gamma]$

$c := H(\text{high}(\mathbf{A}\mathbf{y}), \mu)$

$\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$

If $|\mathbf{z}| > \gamma - \beta$ or $|\text{low}(\mathbf{A}\mathbf{y} - c\mathbf{s}_2)| > \gamma - \beta$

restart

Signature = $(\mathbf{z}, c)$

Verify($\mathbf{z}$, c, μ)

Check that $|\mathbf{z}| \le \gamma - \beta$

and

$$c=H(\text{high}(\underbrace{\mathbf{A}\mathbf{z} - cb\mathbf{t}_1)}_{\mathbf{A}\mathbf{z} - c\mathbf{t} + c\mathbf{t}_0}, \mu)$$

Want $\text{high}(\mathbf{A}\mathbf{y}) = \text{high}(\mathbf{A}\mathbf{z} - c\mathbf{t}) = \text{high}(\mathbf{A}\mathbf{z} - c\mathbf{t} + c\mathbf{t}_0)$

Give out "carries" caused by $c\mathbf{t}_0$ as hints

Dilithium

(high-level overview)

$$\mathbf{A}s_1+s_2=t_0+bt_1$$

Sign(μ)

$y \leftarrow$ Coefficients in $[-\gamma, \gamma]$

$c := H(\text{high}(\mathbf{A}y), \mu)$

$z := y + cs_1$

If $|z| > \gamma - \beta$ or $|\text{low}(\mathbf{A}y - cs_2)| > \gamma - \beta$

restart

Create carry bit hint vector h

Signature = (z, h, c)

Verify(z, c, μ)

Check that $|z| \leq \gamma - \beta$

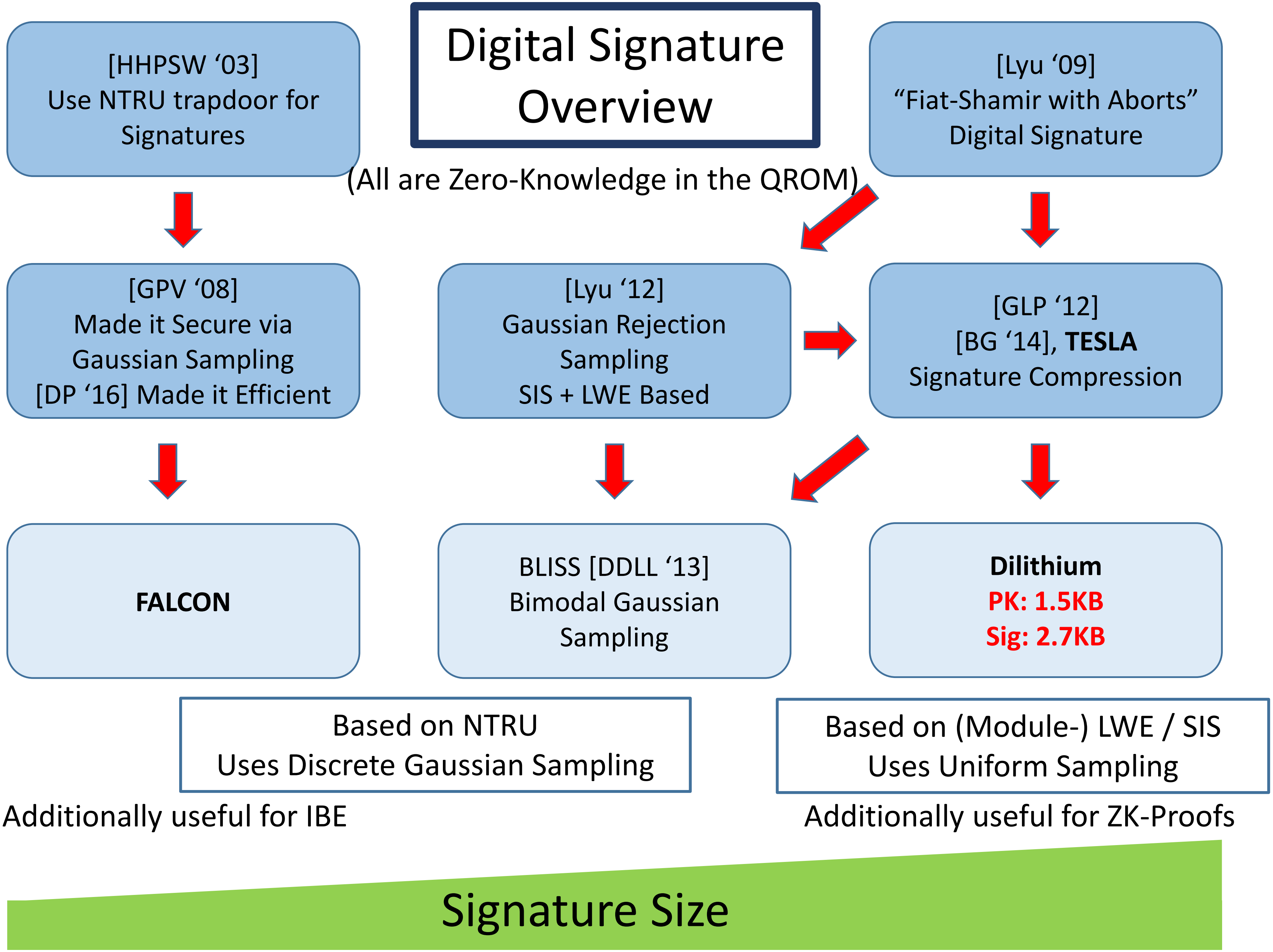
and

$$c=H(\text{high}(\underbrace{h \text{ “+” } \mathbf{A}z - cbt_1)}_{\text{high}(\mathbf{A}y)}, \mu)$$

Hint h

- adds 100 – 200 bytes to the signature
- Saves ≈ 2KB in the public key

Digital Signature Overview



(All are Zero-Knowledge in the QROM)

[HHPSW '03]
Use NTRU trapdoor for Signatures

[Lyu '09]
"Fiat-Shamir with Aborts" Digital Signature

[GPV '08]
Made it Secure via Gaussian Sampling
[DP '16] Made it Efficient

[Lyu '12]
Gaussian Rejection Sampling
SIS + LWE Based

[GLP '12]
[BG '14], **TESLA**
Signature Compression

FALCON

BLISS [DDLL '13]
Bimodal Gaussian Sampling

Dilithium
PK: 1.5KB
Sig: 2.7KB

Based on NTRU
Uses Discrete Gaussian Sampling

Based on (Module-) LWE / SIS
Uses Uniform Sampling

Additionally useful for IBE

Additionally useful for ZK-Proofs

Signature Size

Hash-and-Sign

[HHPSW] → [GVP]→… → FALCON

FALCON

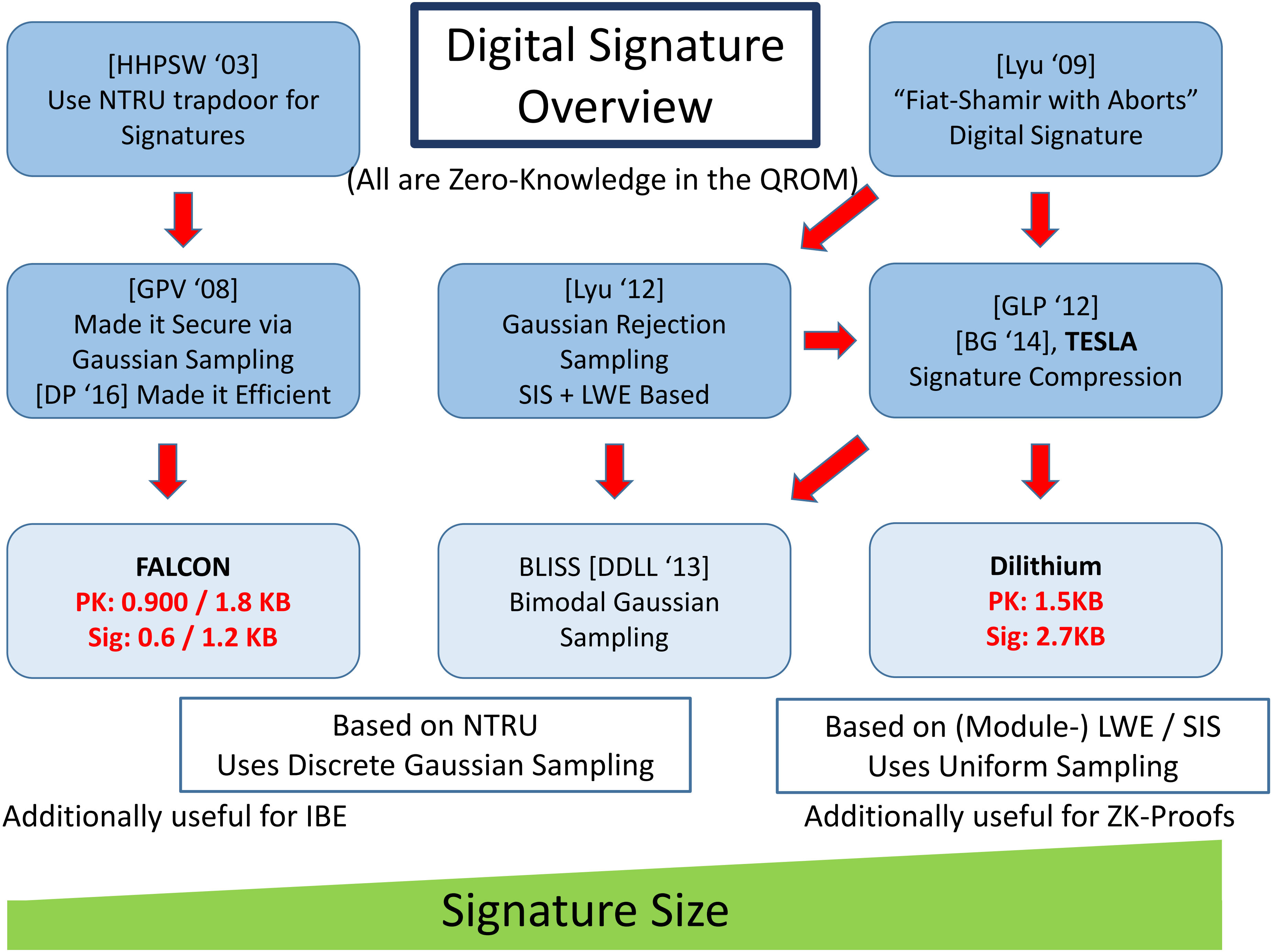
$$\begin{pmatrix} \quad a \quad \end{pmatrix}\underbrace{\begin{bmatrix} y \end{bmatrix} + \begin{bmatrix} e \end{bmatrix}}_{\text{Small coefficients}} = \begin{bmatrix} z \end{bmatrix}$$

z=H(message)

Signer has a “trapdoor” that allows him to find short y,e for any z

Signing does not leak anything about the trapdoor

Digital Signature Overview



(All are Zero-Knowledge in the QROM)

[HHPSW ‘03]
Use NTRU trapdoor for Signatures

[Lyu ‘09]
“Fiat-Shamir with Aborts” Digital Signature

[GPV ‘08]
Made it Secure via Gaussian Sampling
[DP ‘16] Made it Efficient

[Lyu ‘12]
Gaussian Rejection Sampling
SIS + LWE Based

[GLP ‘12]
[BG ‘14], **TESLA**
Signature Compression

FALCON
PK: 0.900 / 1.8 KB
Sig: 0.6 / 1.2 KB

BLISS [DDLL ‘13]
Bimodal Gaussian Sampling

Dilithium
PK: 1.5KB
Sig: 2.7KB

Based on NTRU
Uses Discrete Gaussian Sampling

Based on (Module-) LWE / SIS
Uses Uniform Sampling

Additionally useful for IBE

Additionally useful for ZK-Proofs

Signature Size

Personal PQ-Recommendations

- If you want minimal assumptions:
  - Encryption / Key Exchange: **Frodo** (or something like it based on LWE)
  - Signature: **SPHINCS** (or something like it using Merkle trees)

- If you care about efficiency:
  - Encryption / Key Exchange: **Kyber** (or some other 1KB equivalent)
  - Signature: **Dilithium**

Lattice Problems

Leads to the smallest:

- pk + ciphertext for encryption (except for isogeny-based crypto, but lattices are much faster right now)
- pk + signature for digital signatures

Lattice Problems

The most analyzed post-quantum assumption (against classical and quantum algorithms)

- Lovasz, Lenstra H., Lenstra A., Babai, Schnorr, Coppersmith, Shamir, Regev, Shor, etc. all worked on lattice algorithms or attacks against some lattice cryptoscheme
- No breakthrough novel techniques since LLL
- Cryptanalysis using known techniques is believed to be approaching a lower bound

Performance Comparisons

	PK Size	Cipher Size	KeyGen Cycles	Enc. Cycles	Dec. Cycles
Frodo	11 KB	11 KB	1200 K	1800 K	1800 K
Kyber	1.1 KB	1.2 KB	85 K	110 K	110 K

	PK Size	Sig. Size	Sign Cycles	Verify Cycles
SPHINCS	1 KB	40 KB	50,000 K	1,500 K
Dilithium	1.5 KB	2.7 KB	500 K	175 K

Action Recommendations

- If you need post-quantum crypto now, don’t wait for NIST standards
- Many proposals are just small variants of well-studied problems (no breakthrough ideas in lattice crypto)
- Pick something and use it in tandem with current crypto
- Europe can create its own set of standards in under a year

  “The enemy of a good plan is the dream of a perfect plan”

Thank You.