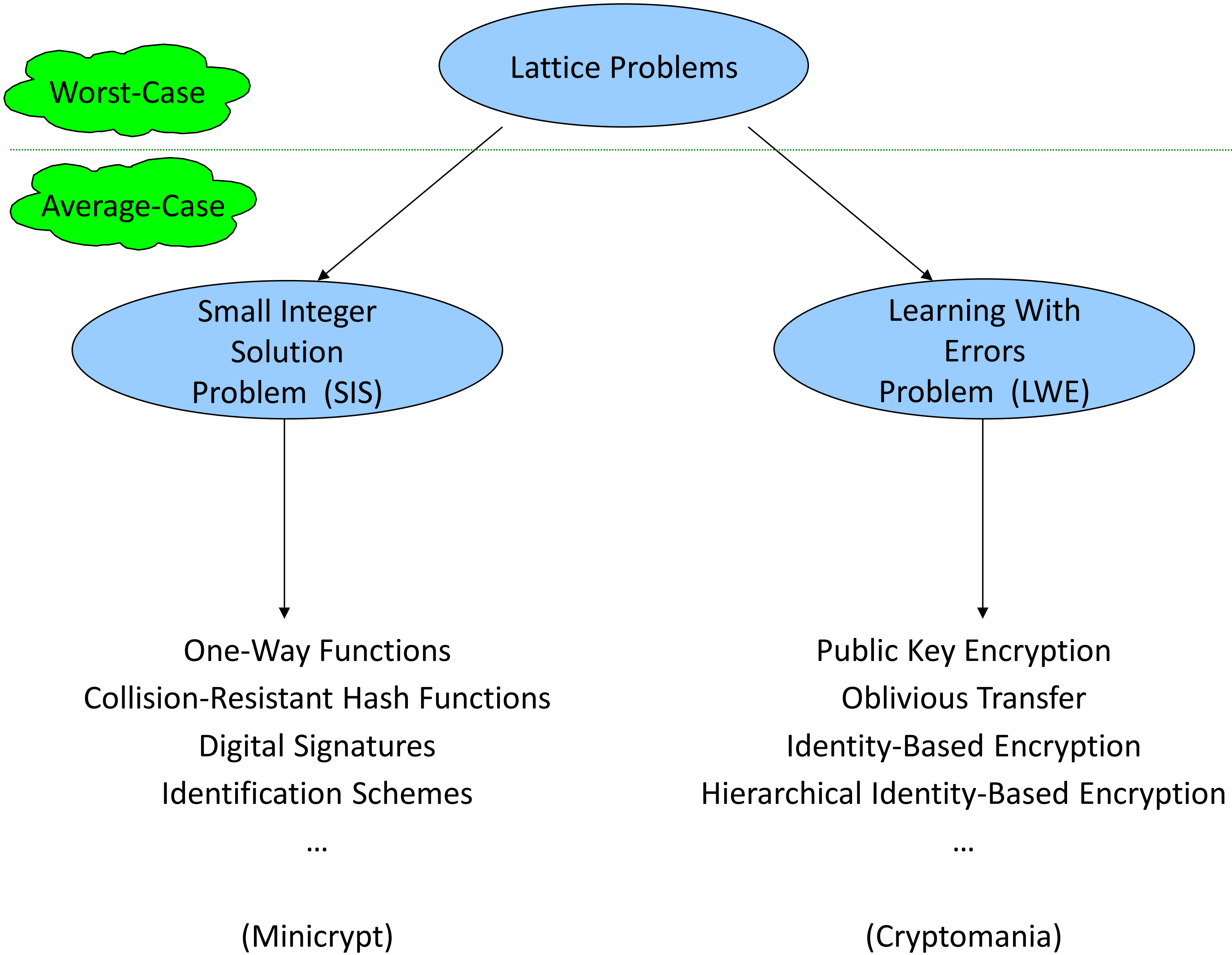
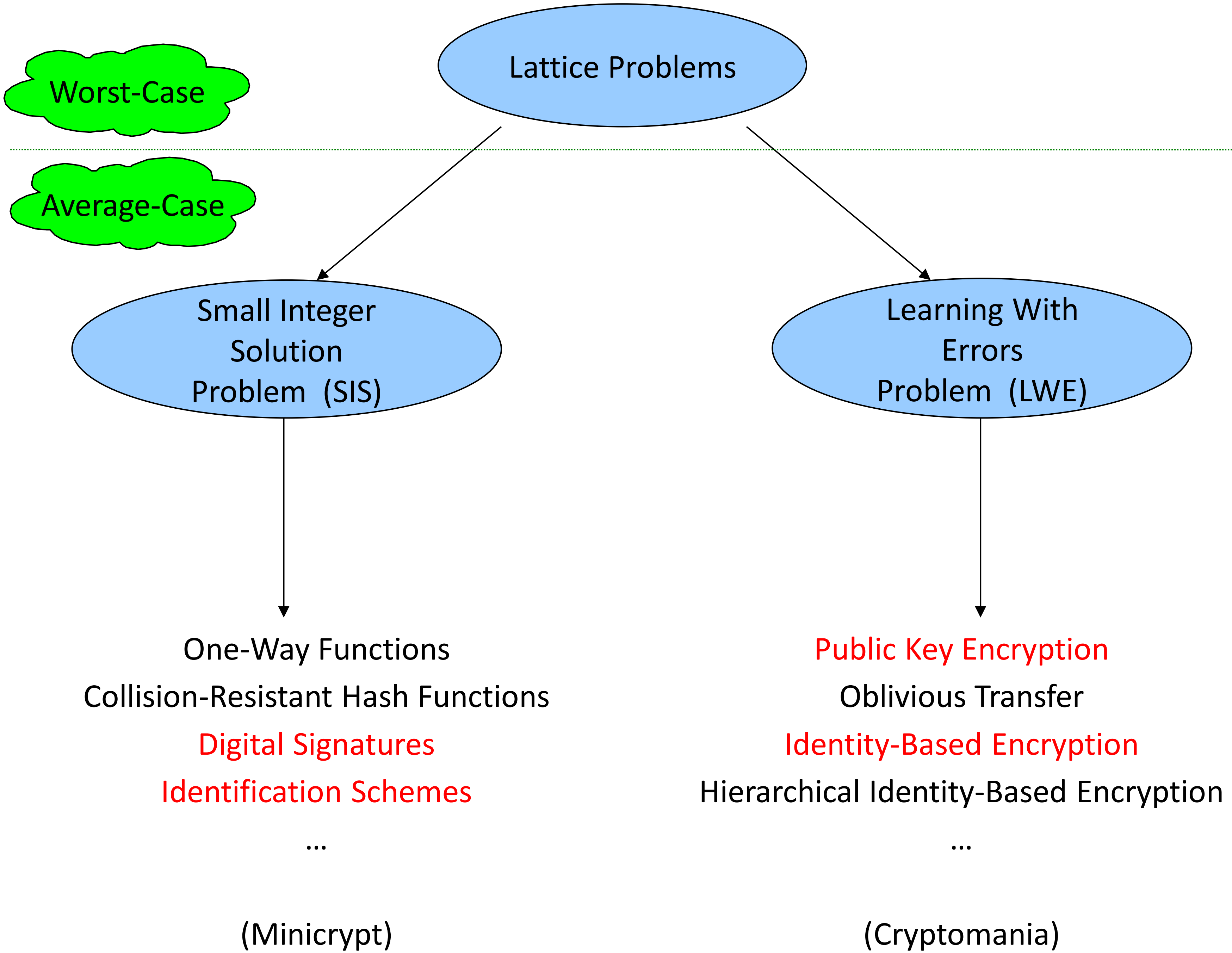
Lattice-Based Cryptography

Vadim Lyubashevsky
INRIA / ENS, Paris

March 20, 2013


Lattice Problems
Worst-Case
Average-Case
Small Integer Solution Problem (SIS)
Learning With Errors Problem (LWE)
One-Way Functions
Collision-Resistant Hash Functions
Digital Signatures
Identification Schemes
…
(Minicrypt)
Public Key Encryption
Oblivious Transfer
Identity-Based Encryption
Hierarchical Identity-Based Encryption
…
(Cryptomania)


Worst-Case
Average-Case
Lattice Problems
Small Integer Solution Problem (SIS)
Learning With Errors Problem (LWE)
One-Way Functions
Collision-Resistant Hash Functions
Digital Signatures
Identification Schemes
…
(Minicrypt)
Public Key Encryption
Oblivious Transfer
Identity-Based Encryption
Hierarchical Identity-Based Encryption
…
(Cryptomania)

LEARNING WITH ERRORS PROBLEM

Learning With Errors (LWE) Problem

There is a secret vector s in Z_p^n (we'll use Z_{17}^4 as a running example)

An oracle (who knows s) generates a random vector a in Z_p^n

and

“small” noise element e in Z

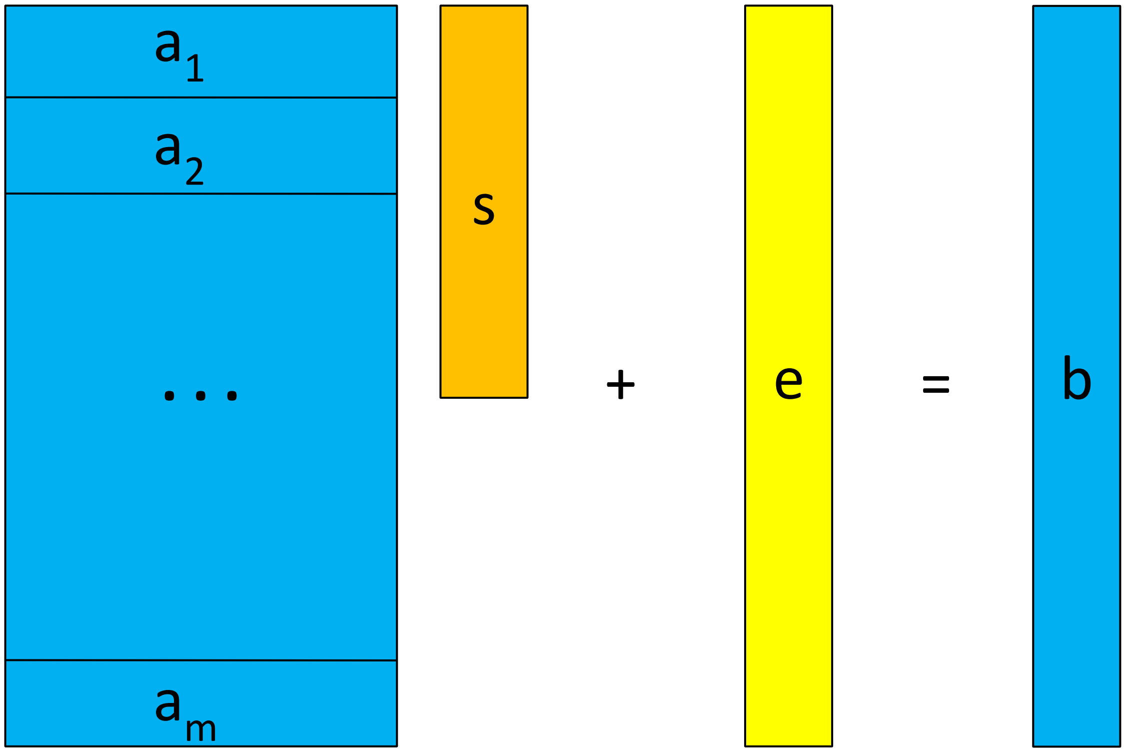
The oracle outputs (a,b=<a,s>+e mod 17)

$$\begin{bmatrix} 2 & 13 & 7 & 3 \\ 4 & 7 & 9 & 1 \\ 6 & 14 & 5 & 11 \end{bmatrix} * \begin{bmatrix} 8 \\ 3 \\ 12 \\ 5 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 13 \\ 12 \\ 3 \end{bmatrix}$$

This procedure is repeated with the same s and fresh a and e

Our task is to find s

Learning With Errors (LWE) Problem

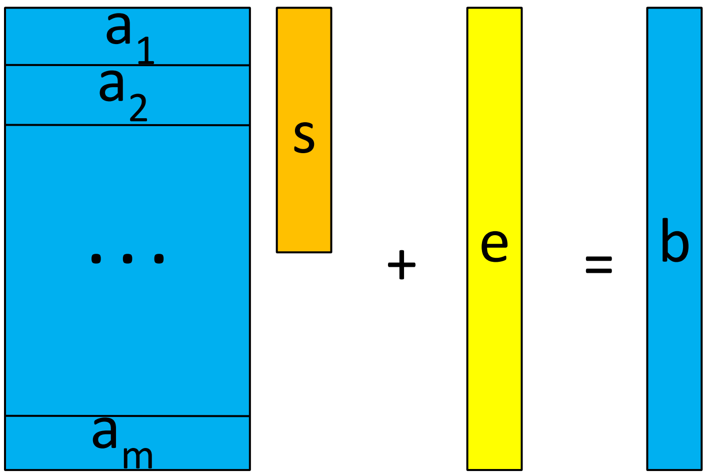


Once there are enough a_i , the s is uniquely determined

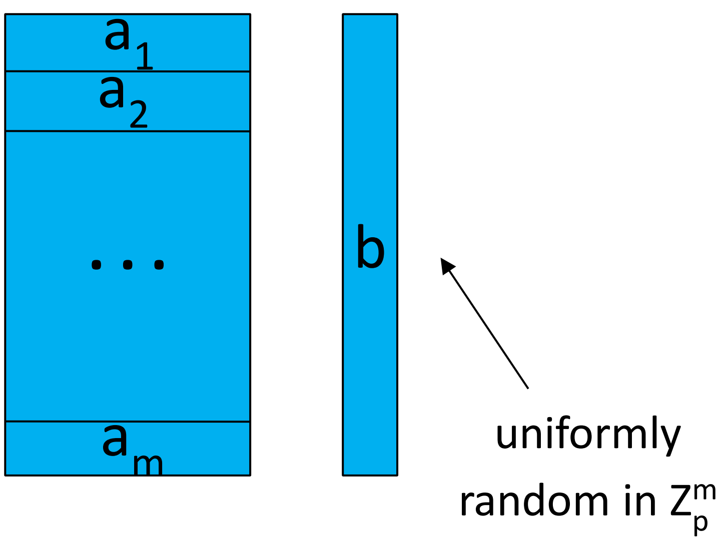
Theorem [Regev '05] : There is a polynomial-time quantum reduction from solving certain lattice problems in the worst-case to solving LWE.

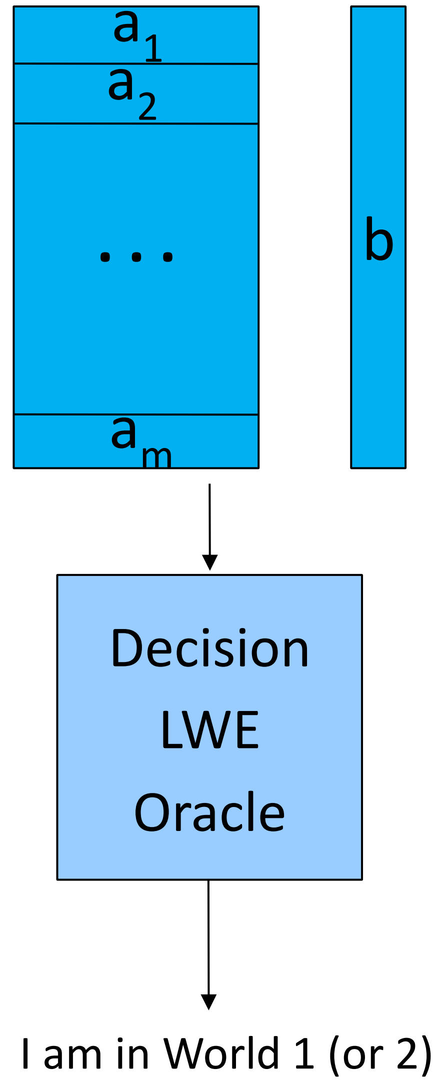
Decision LWE Problem

World 1



World 2





Search LWE < Decision LWE

Use the Decision oracle to figure out the coefficients of s one at a time

Let g be our guess for the first coefficient of s

Repeat the following:

Receive LWE pair (a,b)

$$\underbrace{\begin{bmatrix} 2 & 13 & 7 & 3 \end{bmatrix}}_{a} * \begin{bmatrix} 8 \\ 3 \\ 12 \\ 5 \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 13 \end{bmatrix}}_{b}$$

Pick random r in Z_{17}

Send sample below to the Decision Oracle

$$\begin{bmatrix} 2+r & 13 & 7 & 3 \end{bmatrix} \qquad \begin{bmatrix} 13+rg \end{bmatrix}$$

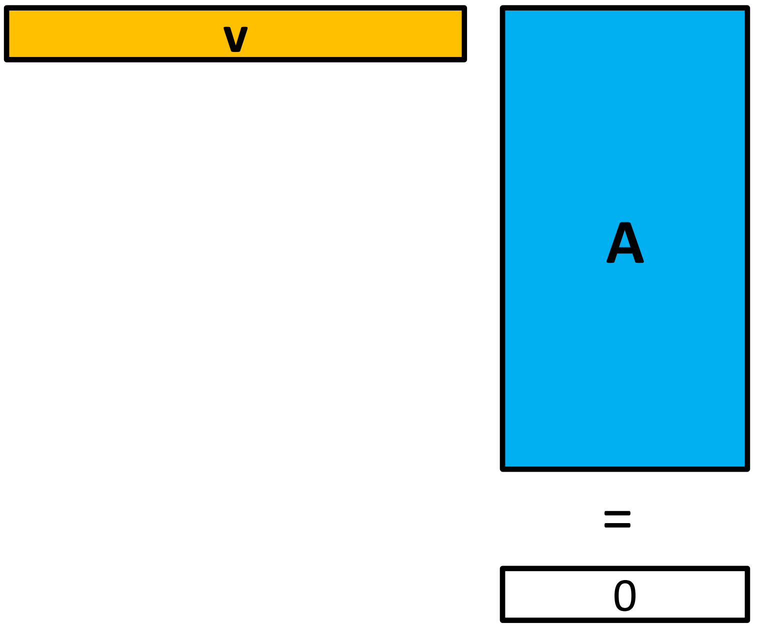
If g is right, then we are sending a distribution from World 1

If g is wrong, then we are sending a distribution from World 2

We will find the right g in O(p) time

Use the same idea to recover all coefficients of s one at a time

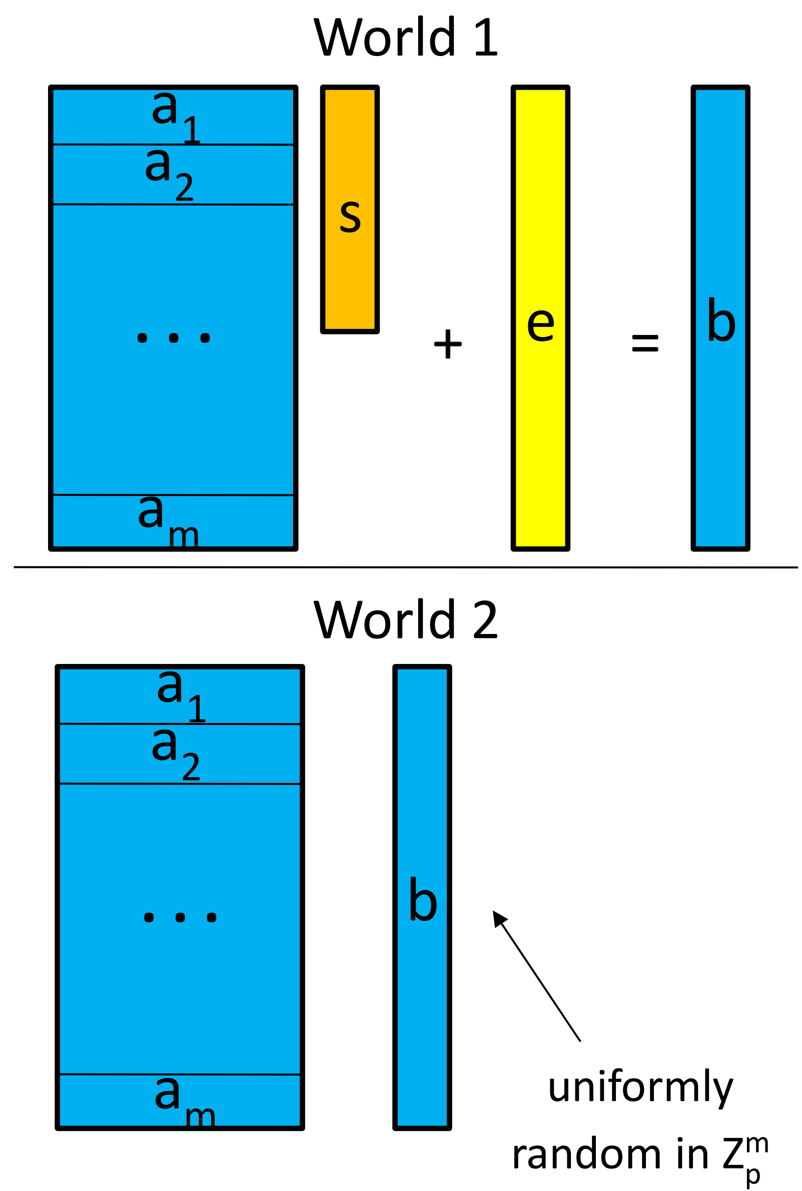
LWE and Lattices



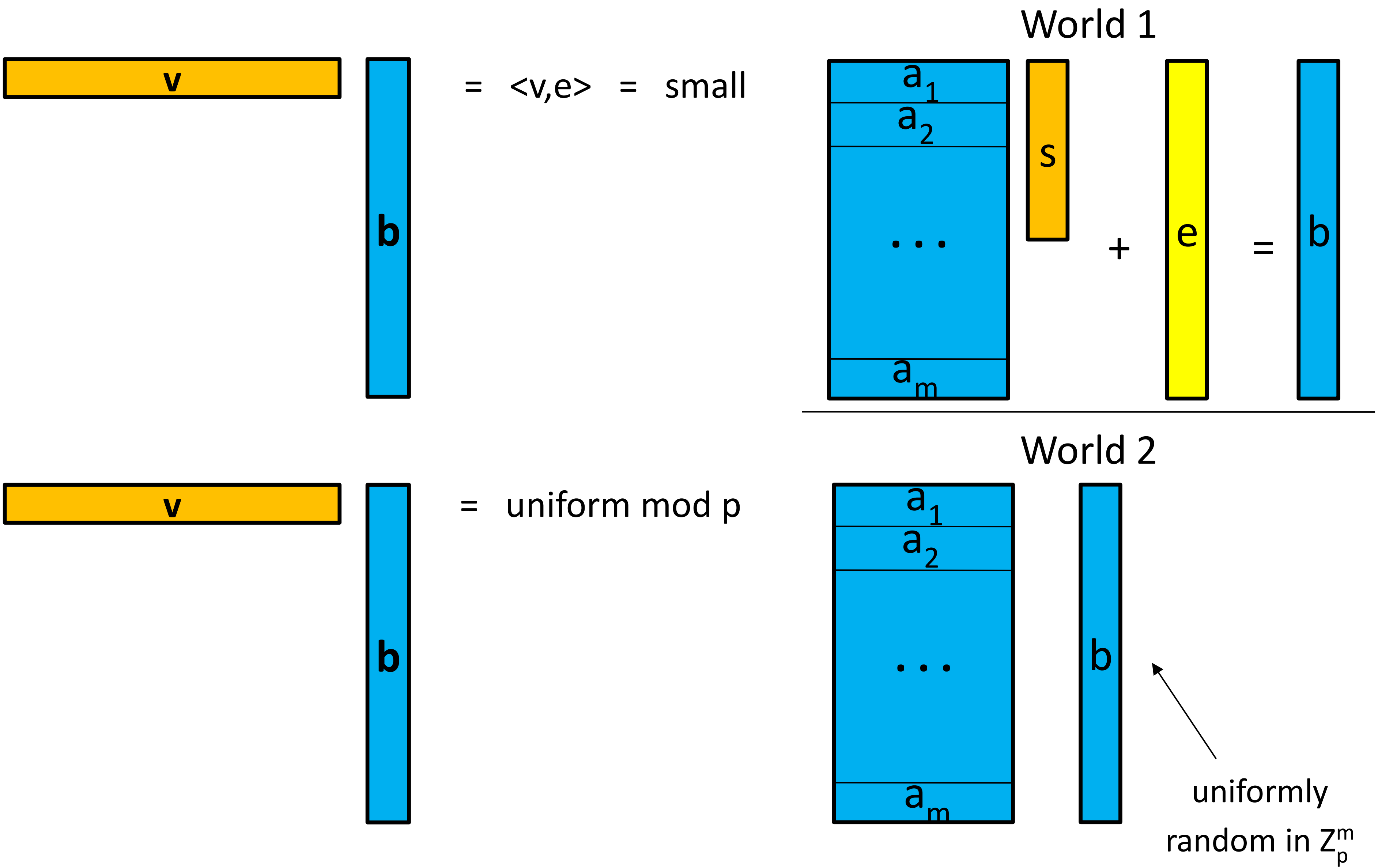
Lattice $L_p^{\perp}(\mathbf{A}) = \{ \mathbf{y} : \mathbf{yA} = \mathbf{0} \bmod p\}$

Find a short vector **v** in $L_p^{\perp}(\mathbf{A})$.

Called the **Small Independent Solution** (SIS) problem

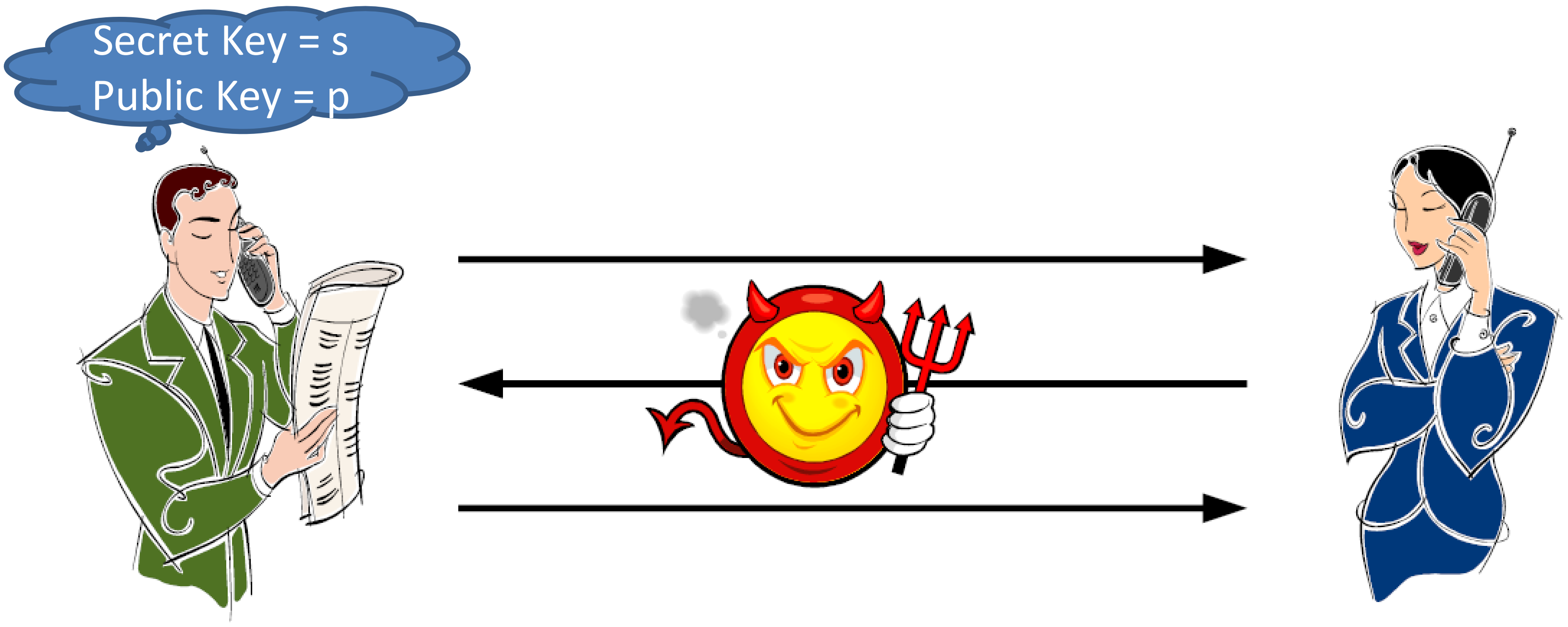


Decision LWE < SIS

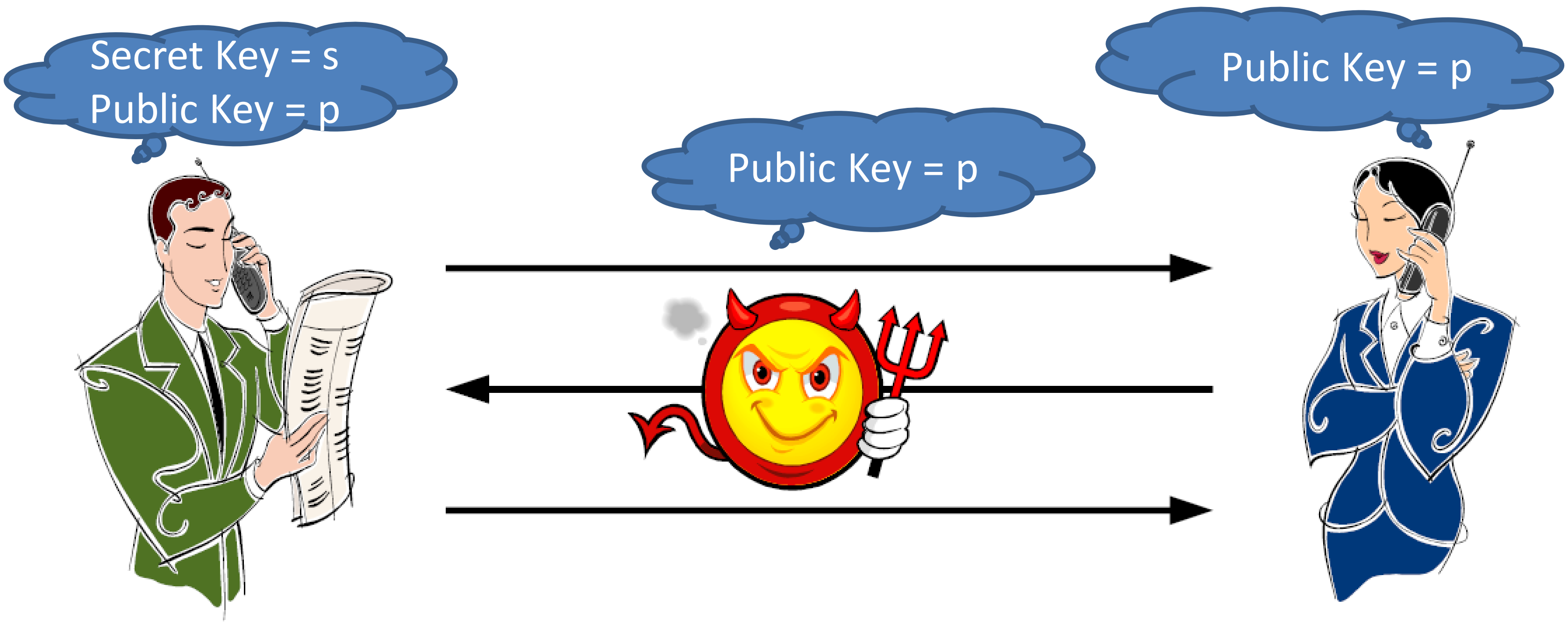


PUBLIC KEY ENCRYPTION FROM LWE

Public-Key Cryptography



Public-Key Cryptography



Public Key Encryption

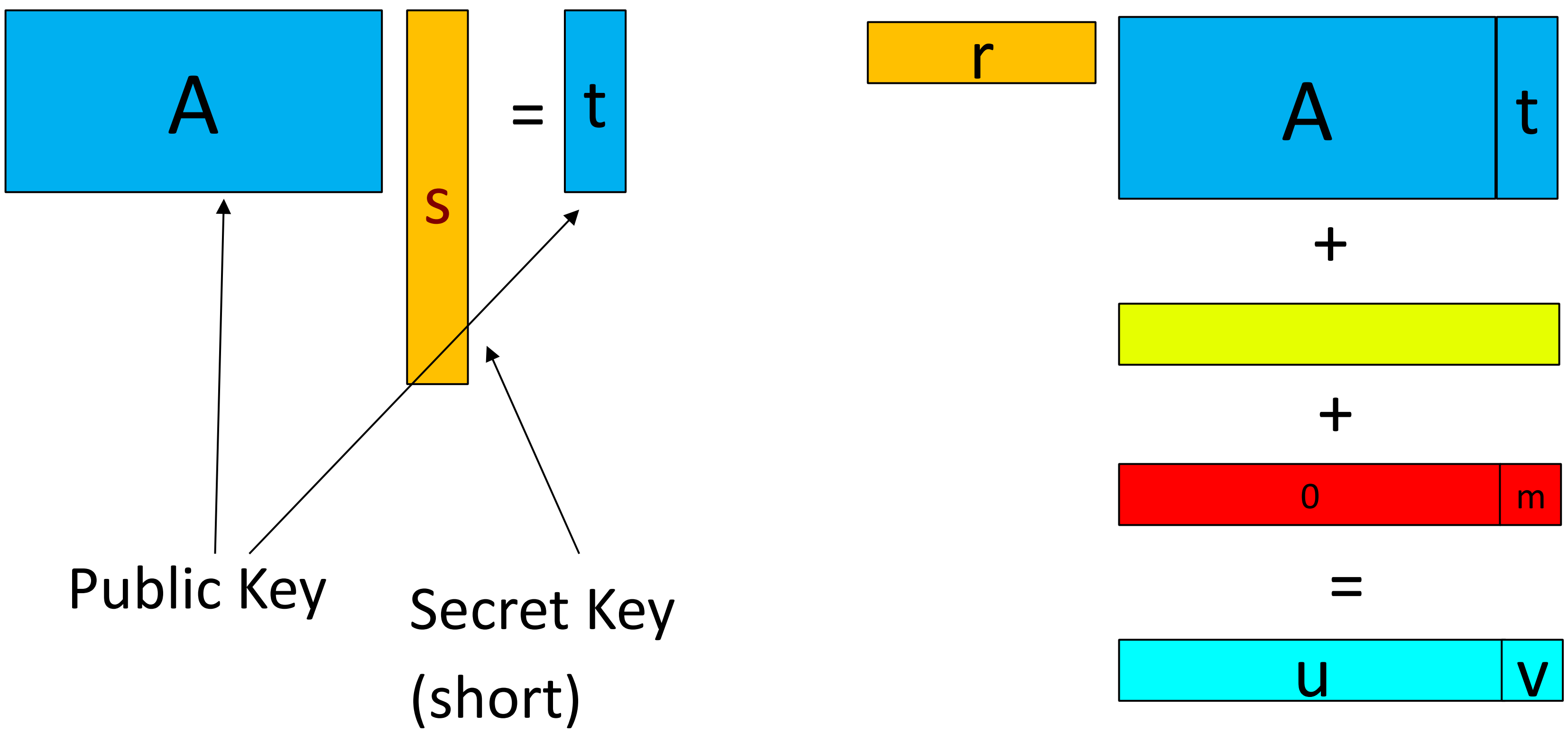
- (sk,pk) $\leftarrow$ KeyGen(1^n)
- c = Enc (pk,m)
- m = Dec(sk,c)

- Correctness: Dec(sk,Enc(pk,m))=m
- CPA-Security: Enc(pk,m_i) are **computationally indistinguishable** from each other

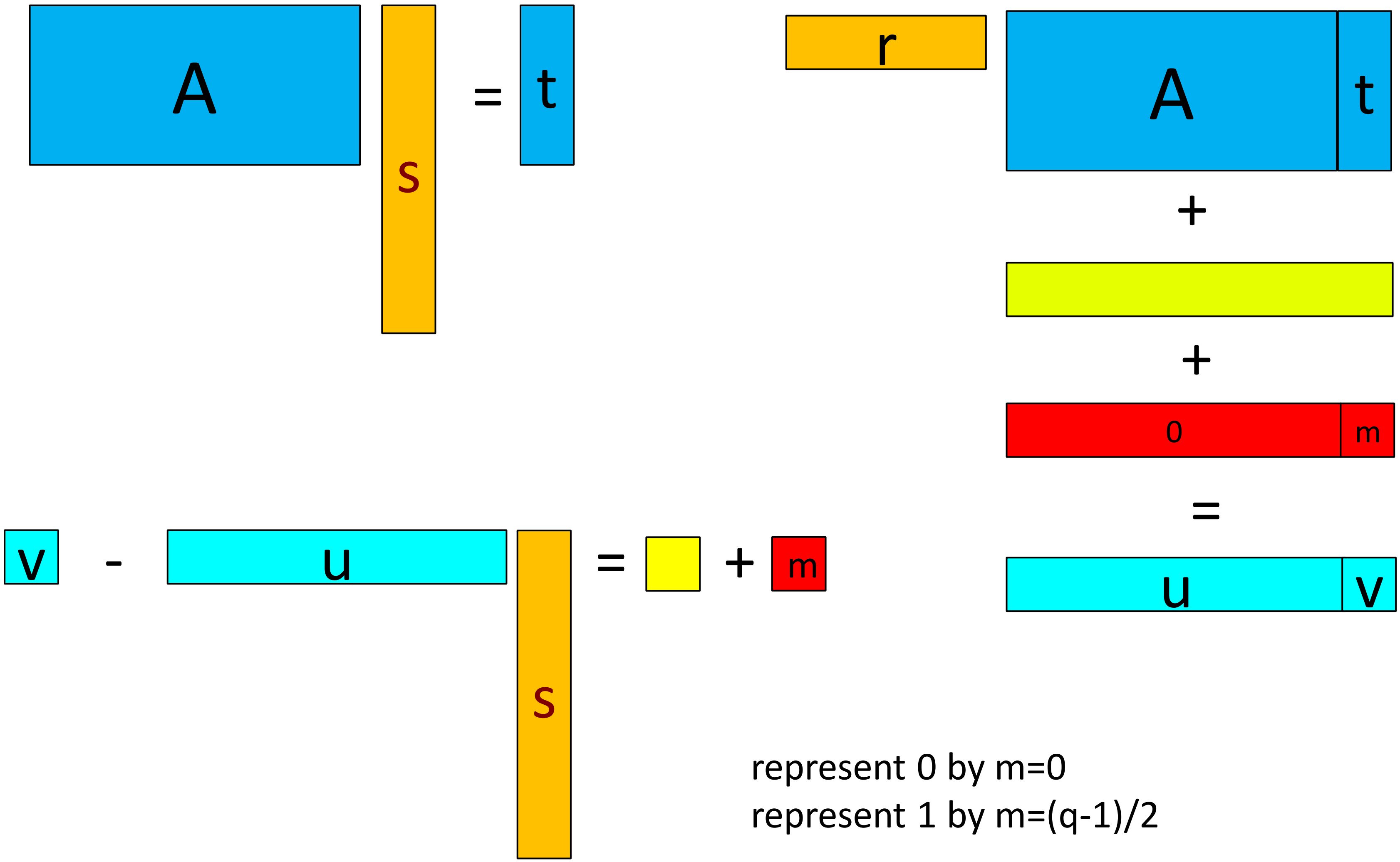
"Computationally Indistinguishable"

<u>DX</u>		<u>D?</u>		<u>DY</u>
X_1		Z_1		Y_1
X_2	$\stackrel{?}{=}$	Z_2	$\stackrel{?}{=}$	Y_2
…		…		…
X_k		Z_k		Y_k
…		…		…

“Dual” Cryptosystem

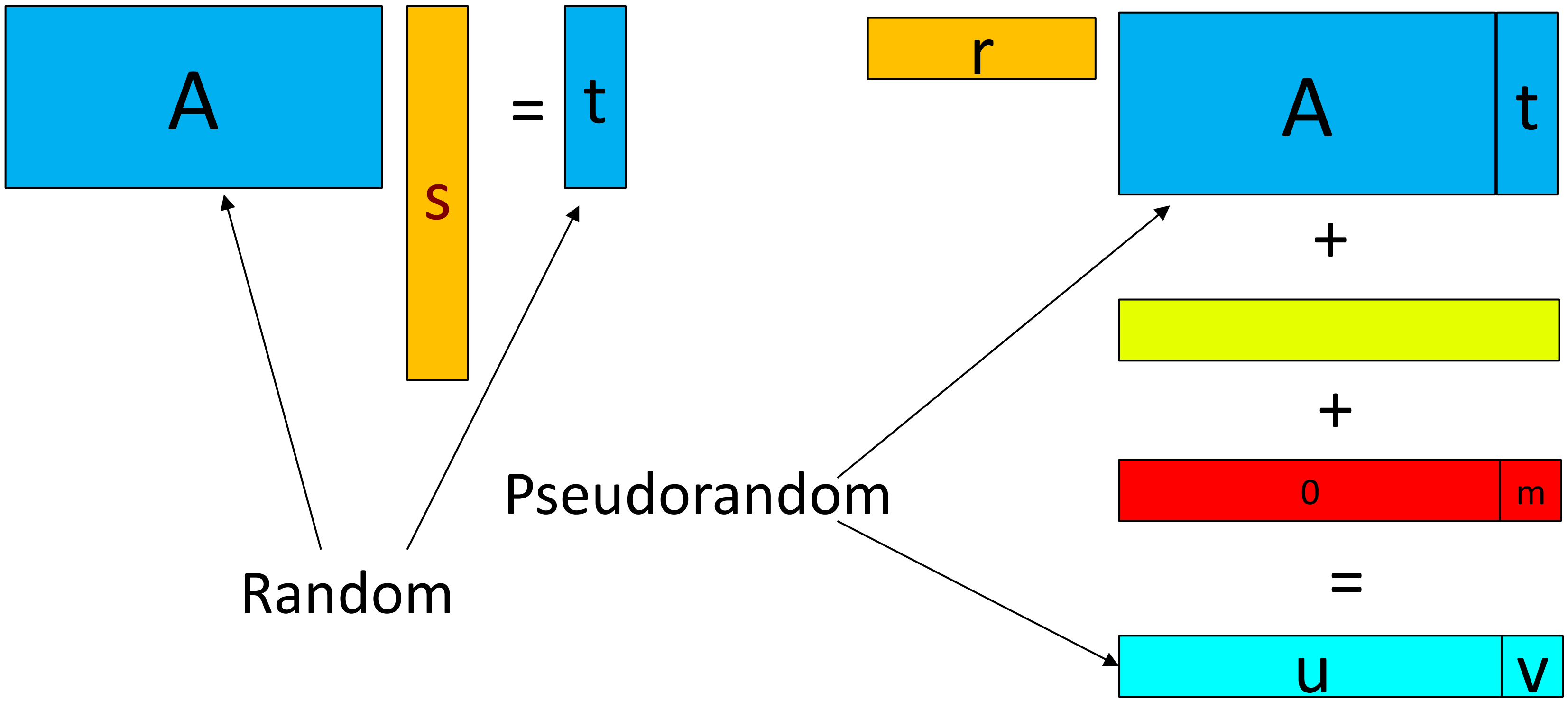


"Dual" Cryptosystem



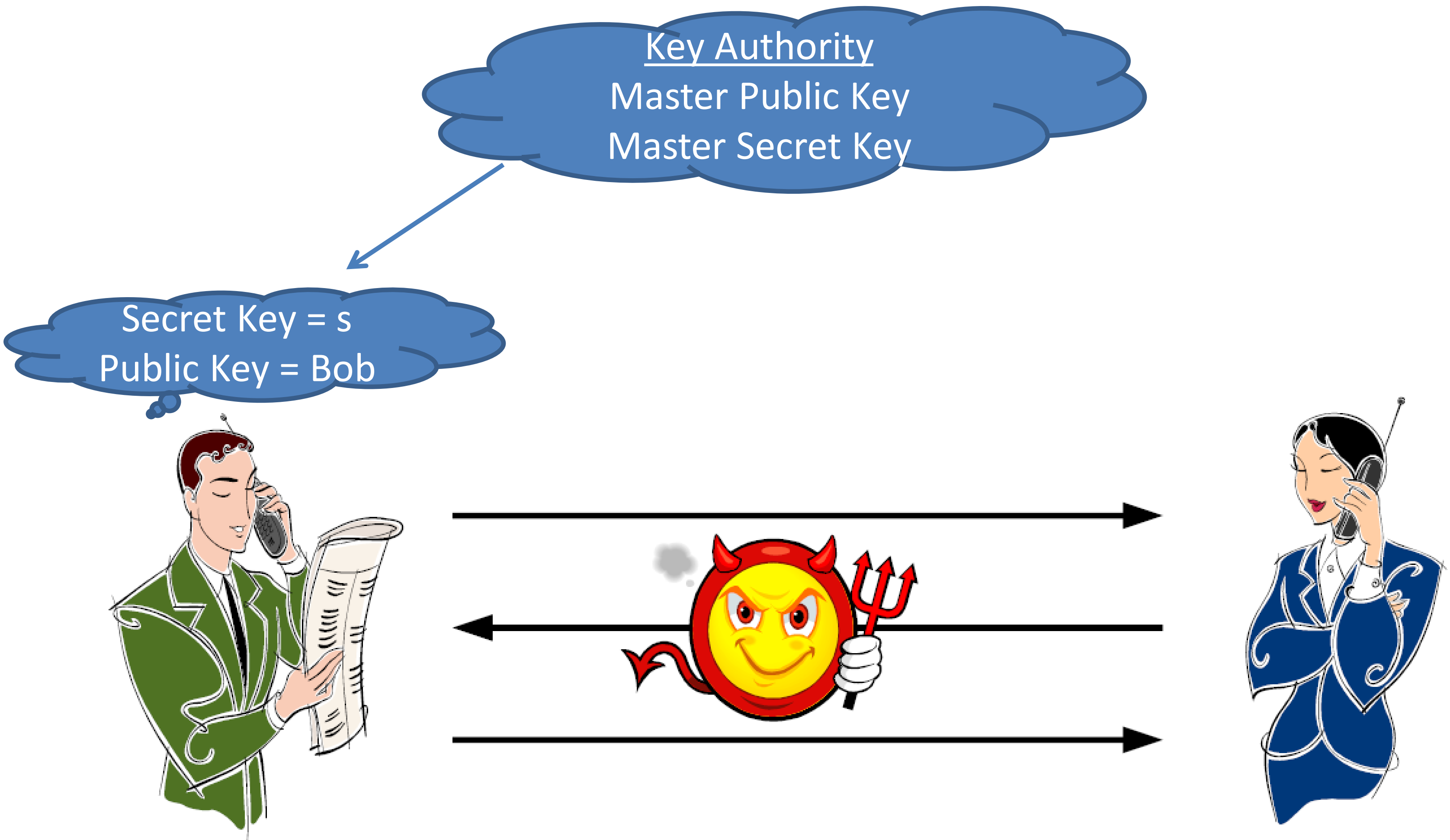
represent 0 by m=0
represent 1 by m=(q-1)/2

“Dual” Cryptosystem Security

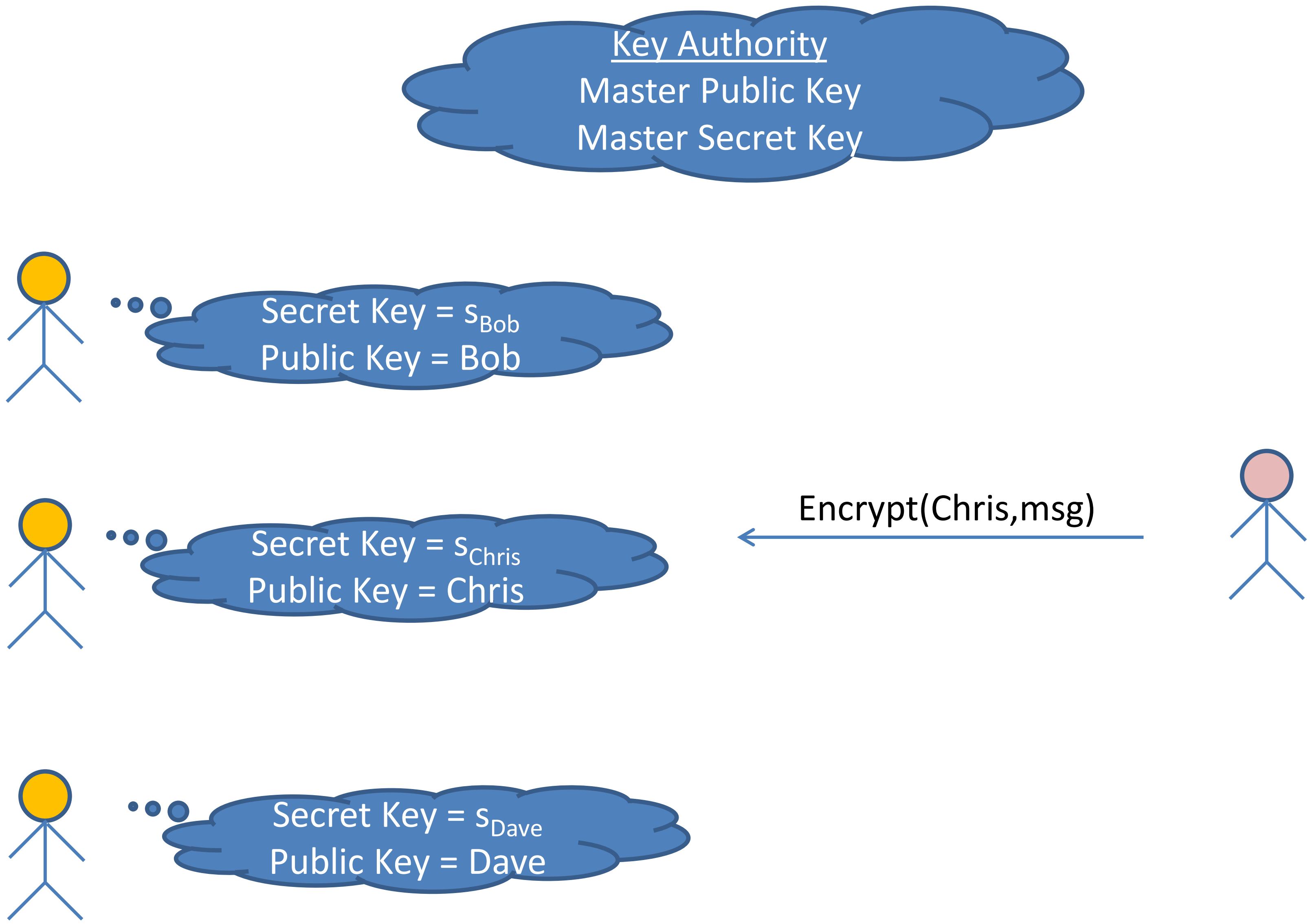


IDENTITY-BASED ENCRYPTION

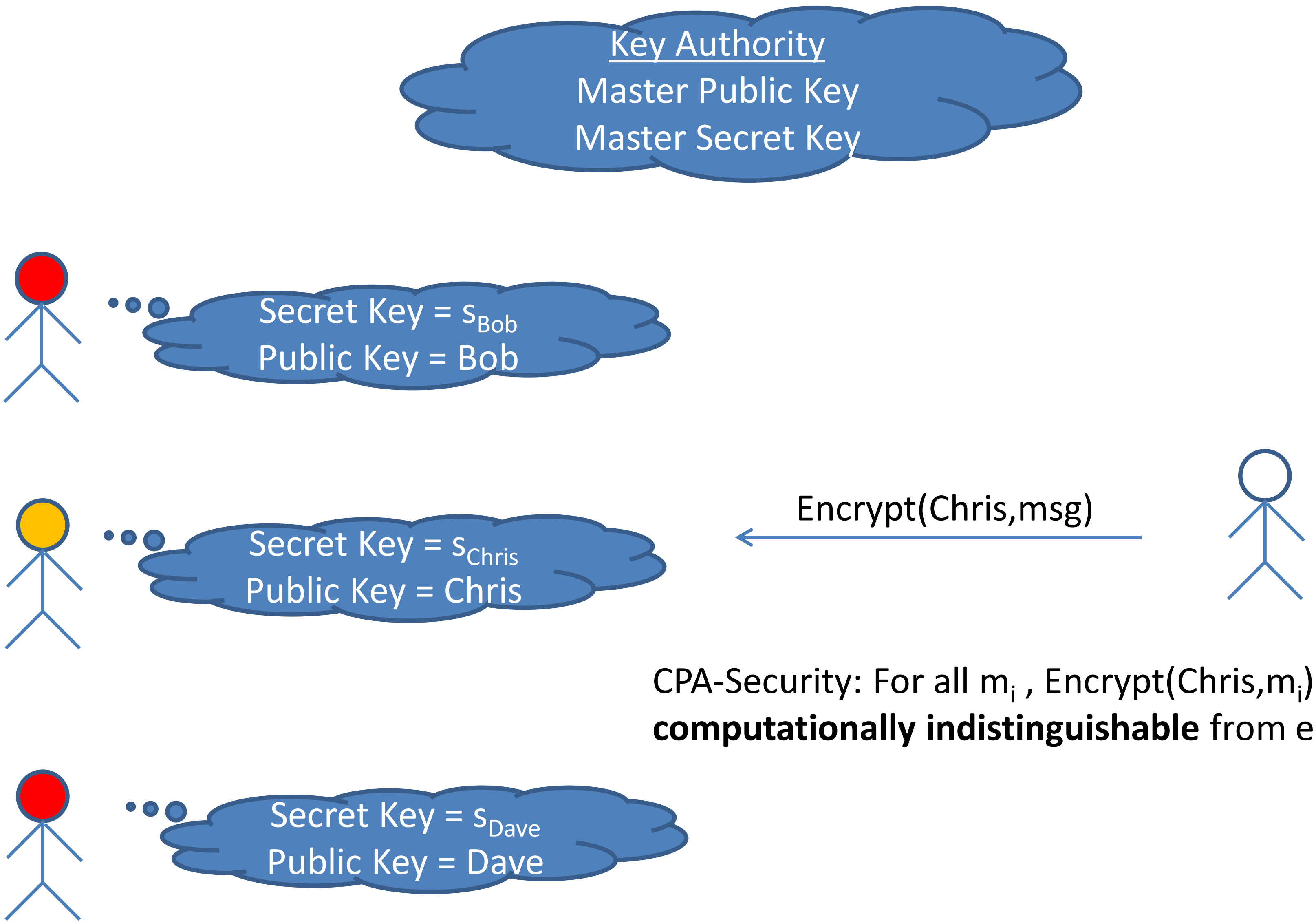
Identity-Based Encryption



Identity-Based Encryption

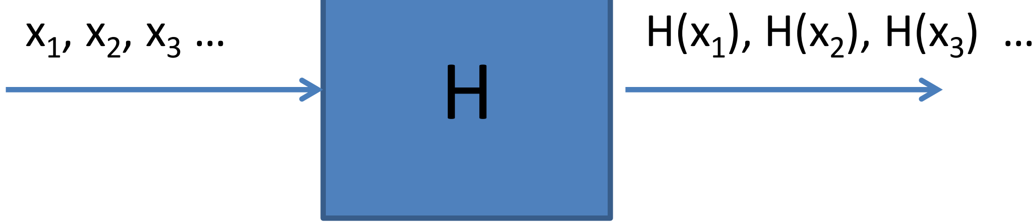


Security for IBE



CPA-Security: For all m_i , Encrypt(Chris,m_i) are **computationally indistinguishable** from each other

IBE based on LWE
(in the Random Oracle Model)



$(x_1, H(x_1)), (x_2, H(x_2)), (x_3, H(x_3)), \ldots$
is computationally indistinguishable from $(x_1,u_1), (x_2,u_2), (x_3,u_3), \ldots$

Security in the Random Oracle Model:

There is a "pseudorandom function" H
Prove security assuming that everyone only has "black box access" to H
In reality, H is replaced by a "cryptographic hash function" (e.g. SHA-256)

If the real scheme is insecure, then there is something wrong
with the hash function

Security Proofs Using a Random Oracle

Adversary cannot access H directly
He must ask us (i.e. the reduction) for H(z)
We pick a random y and output y=H(z)

What's the point?
Suppose f is a 1-way function and f(x) is uniform for random x
For a random y, it's hard to find an x such that f(x)=y

But, for any z, we can *simulate* knowing an x such that f(x)=H(z)
Given z, we pick a random x, compute y=f(x), and *program* y=H(z)
So f(x)=H(z) and all the distributions are as they should be

IBE Based on LWE

$$\mathbf{A}\,\mathbf{s} = \mathbf{b}$$

Lattice $L_p^{\perp}(\mathbf{A}) = \{\, \mathbf{y} : \mathbf{A}\mathbf{y} = \mathbf{0} \bmod p \,\}$

$$\mathbf{T}$$

T is a basis for $L_p^{\perp}(\mathbf{A})$ and has "short" vectors

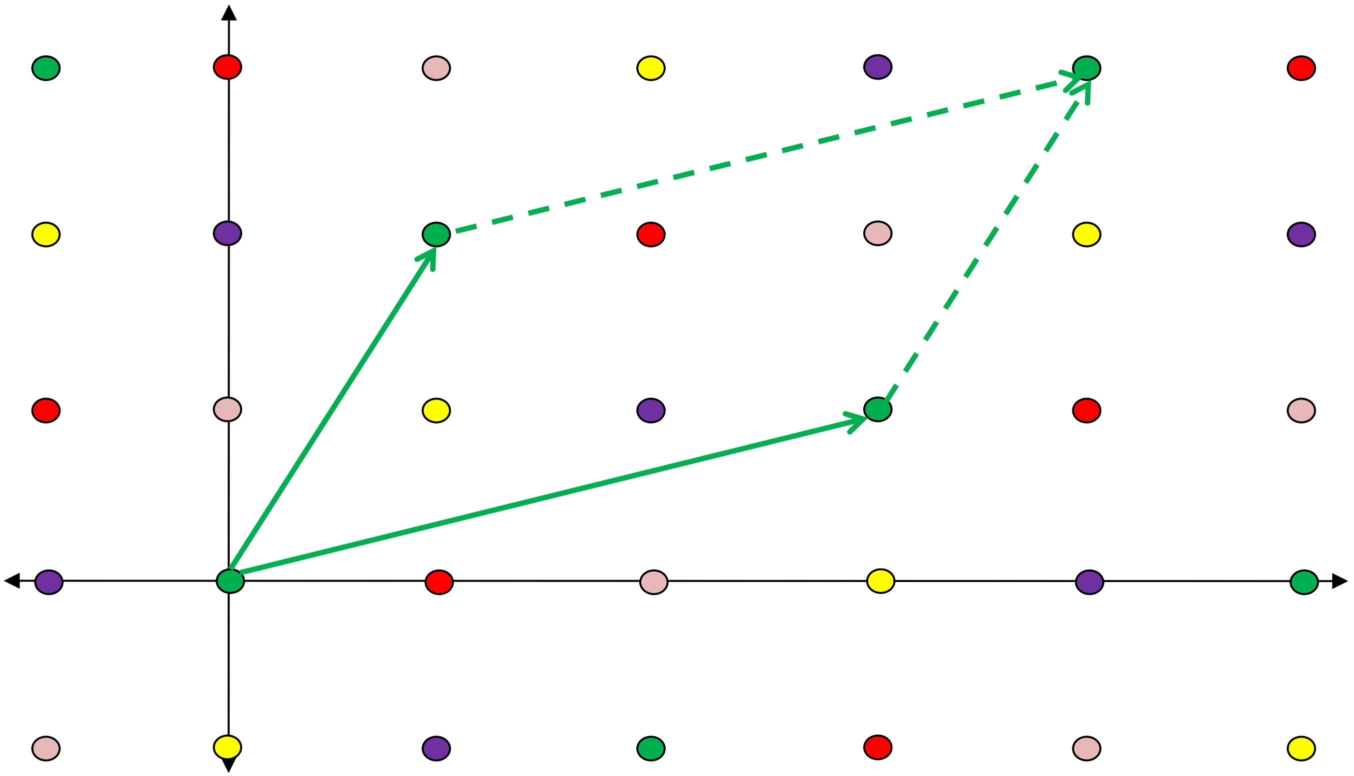
Master Public Key: **A**
Master Secret Key: **T**

Identity = "Bob"
b = H(Bob)

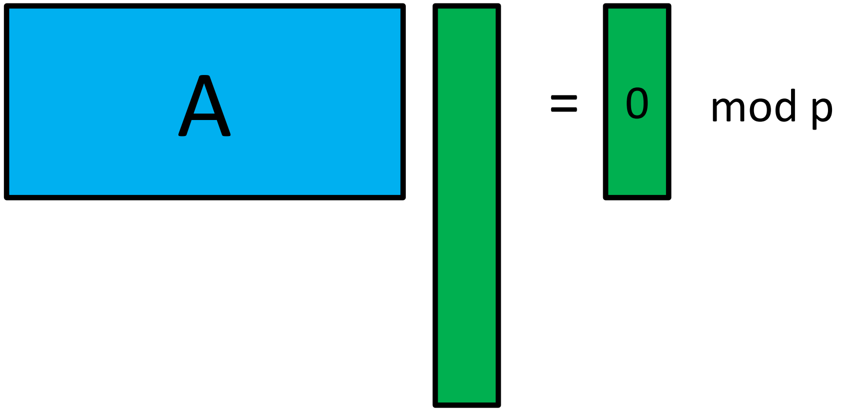
Use the GPV algorithm to find a short **s** such that **As** = **b** mod p

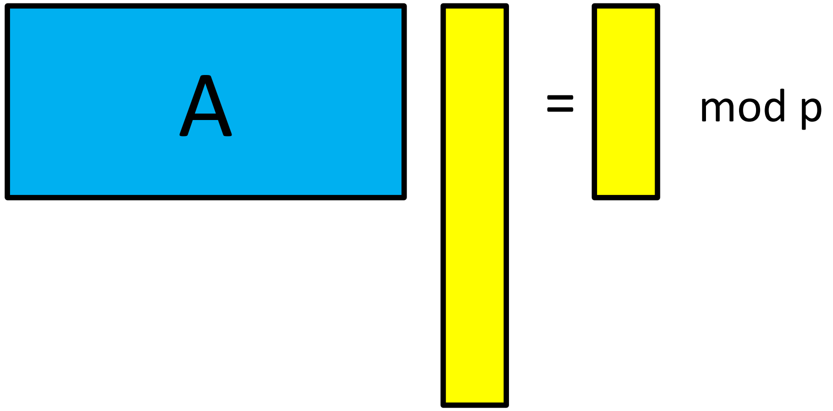
Use "Dual" LWE encryption to Encrypt to Bob

Lattice $L_p^{\perp}(\mathbf{A}) = \{\, \mathbf{y} : \mathbf{A}\mathbf{y} = \mathbf{0} \bmod p \,\}$

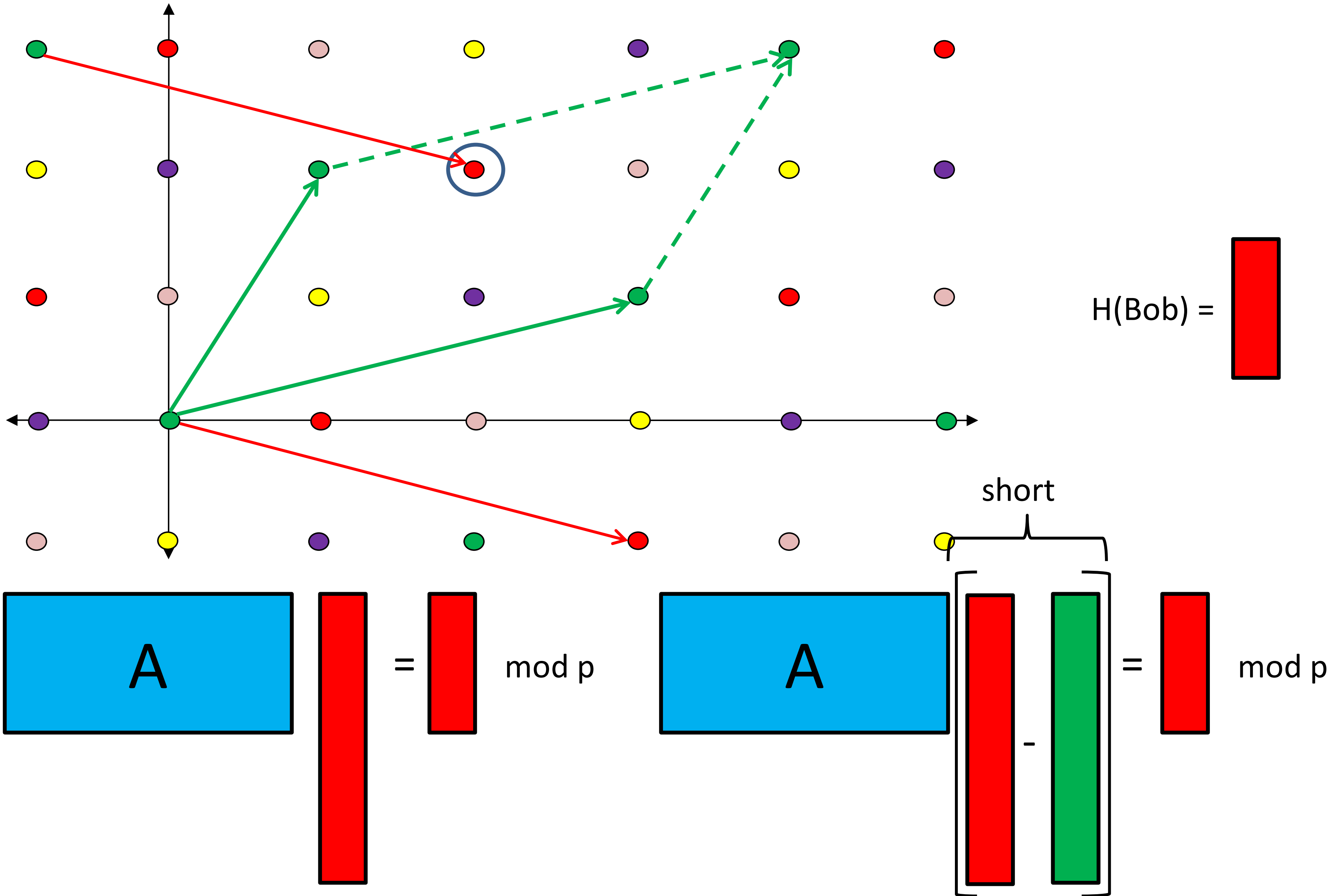


Cosets of $\mathbf{Z}^m / L_p^{\perp}(\mathbf{A})$

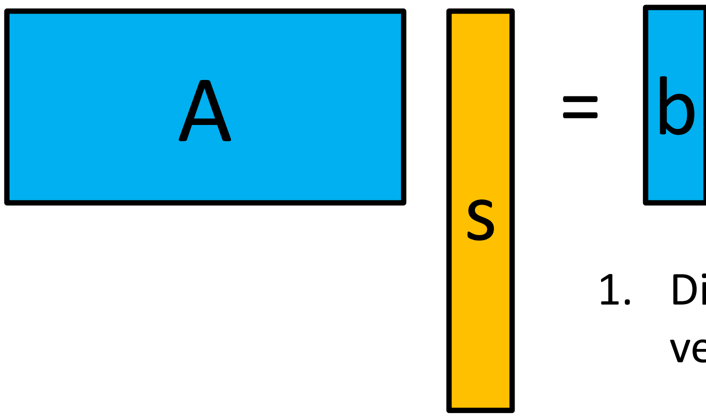




Lattice $L_p^{\perp}(\mathbf{A}) = \{\, \mathbf{y} : \mathbf{A}\mathbf{y} = \mathbf{0} \bmod p \,\}$



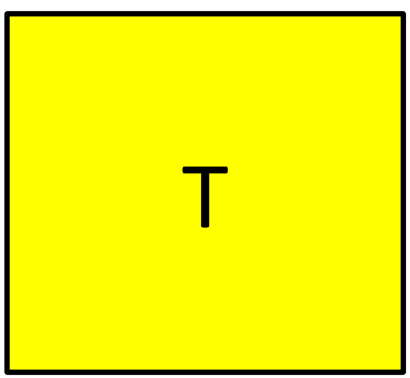
Properties Needed



1. Distribution D of **s** only depends on the length of the vectors comprising **T**

2. The following produce the same distribution of (**s**,**b**)

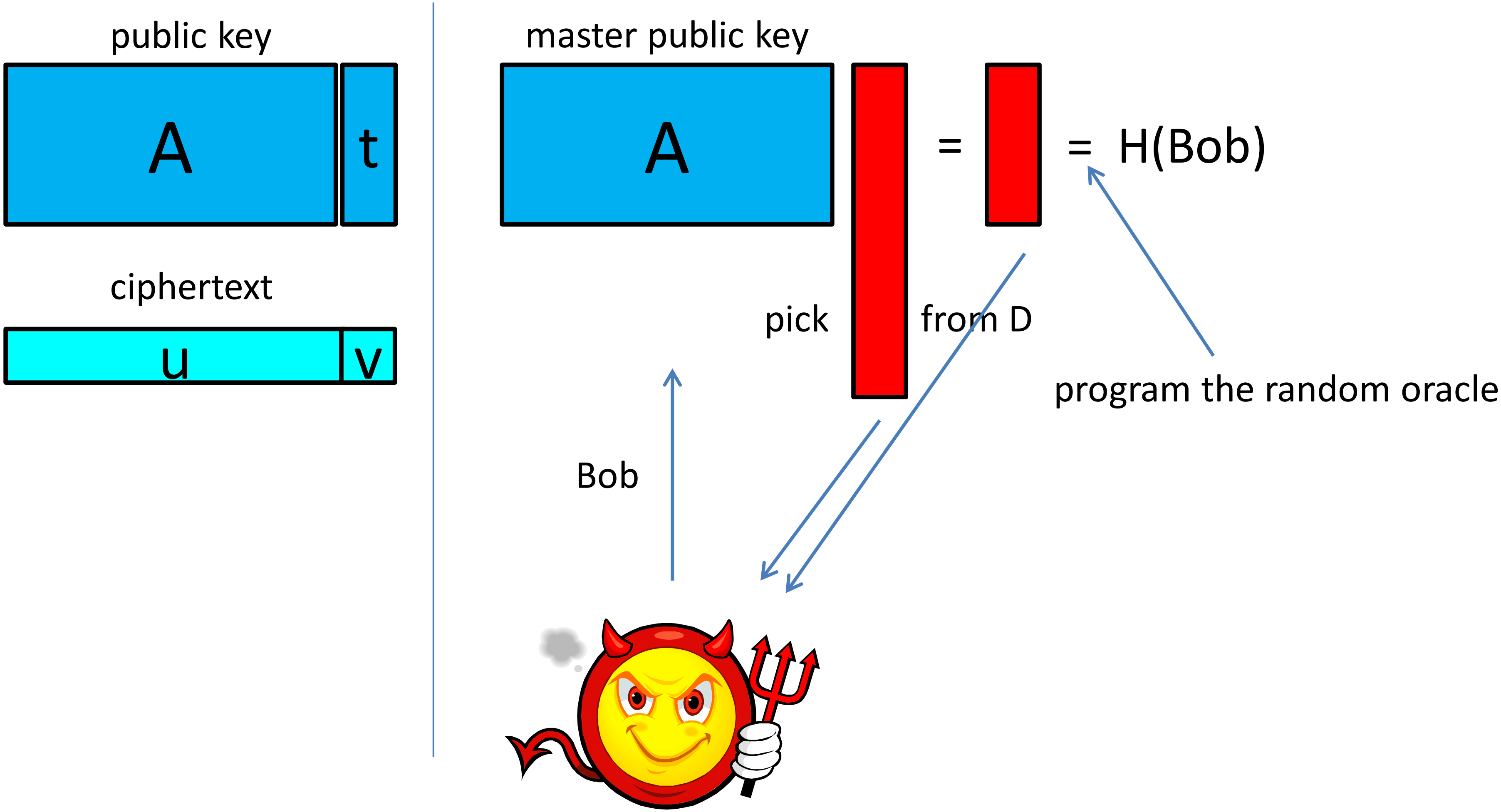
   (a) Choose **s** ~ D. Set **b**=**As**
   (b) Choose random **b**. Use **T** to find an **s** such that **As**=**b**.



(1) is guaranteed by the GPV algorithm
(2) is true if **s** has enough entropy (to make A**s**=**b** uniform mod p)

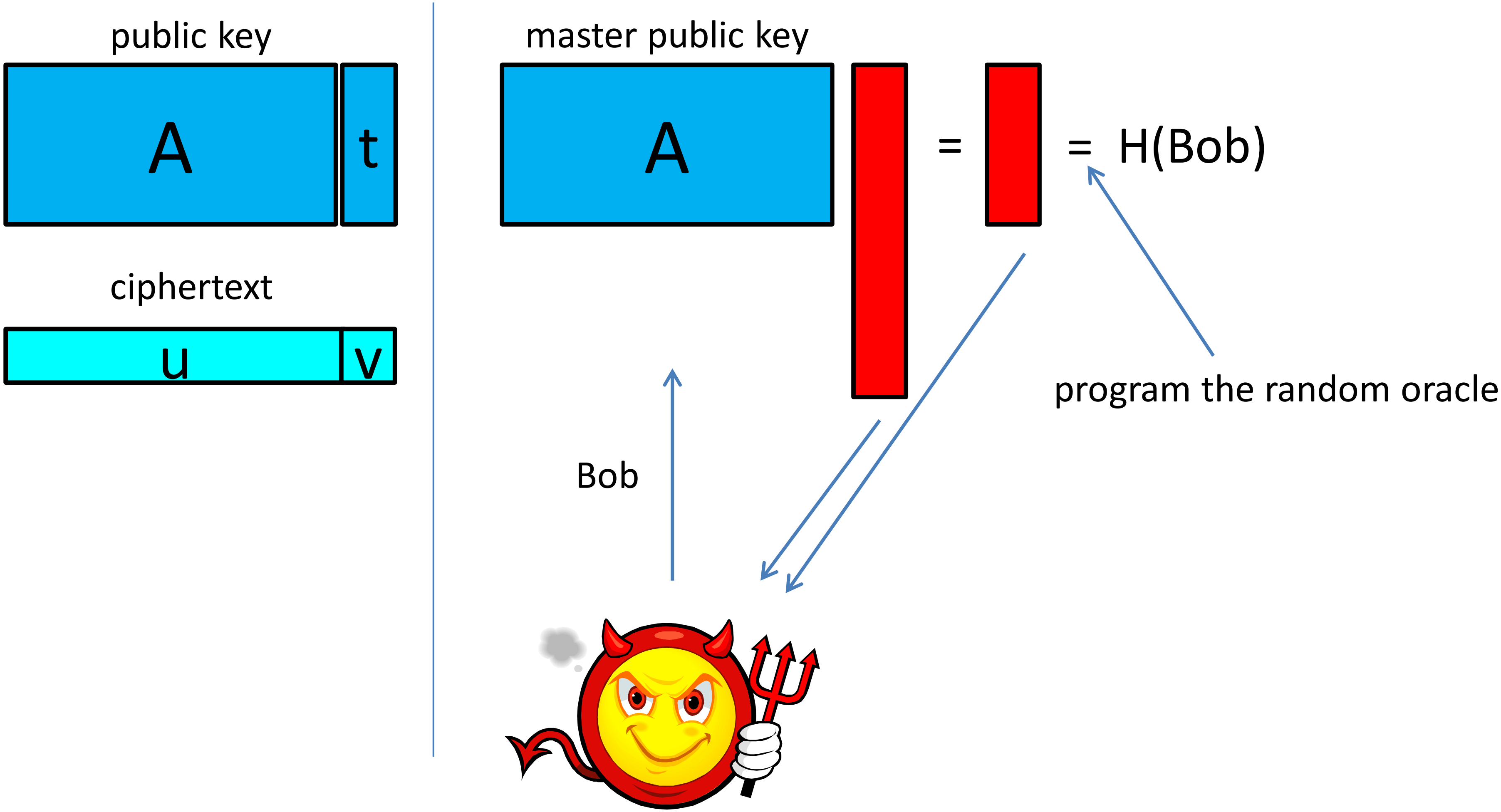
Security Proof Sketch

Show that breaking IBE implies breaking the “Dual” cryptosystem



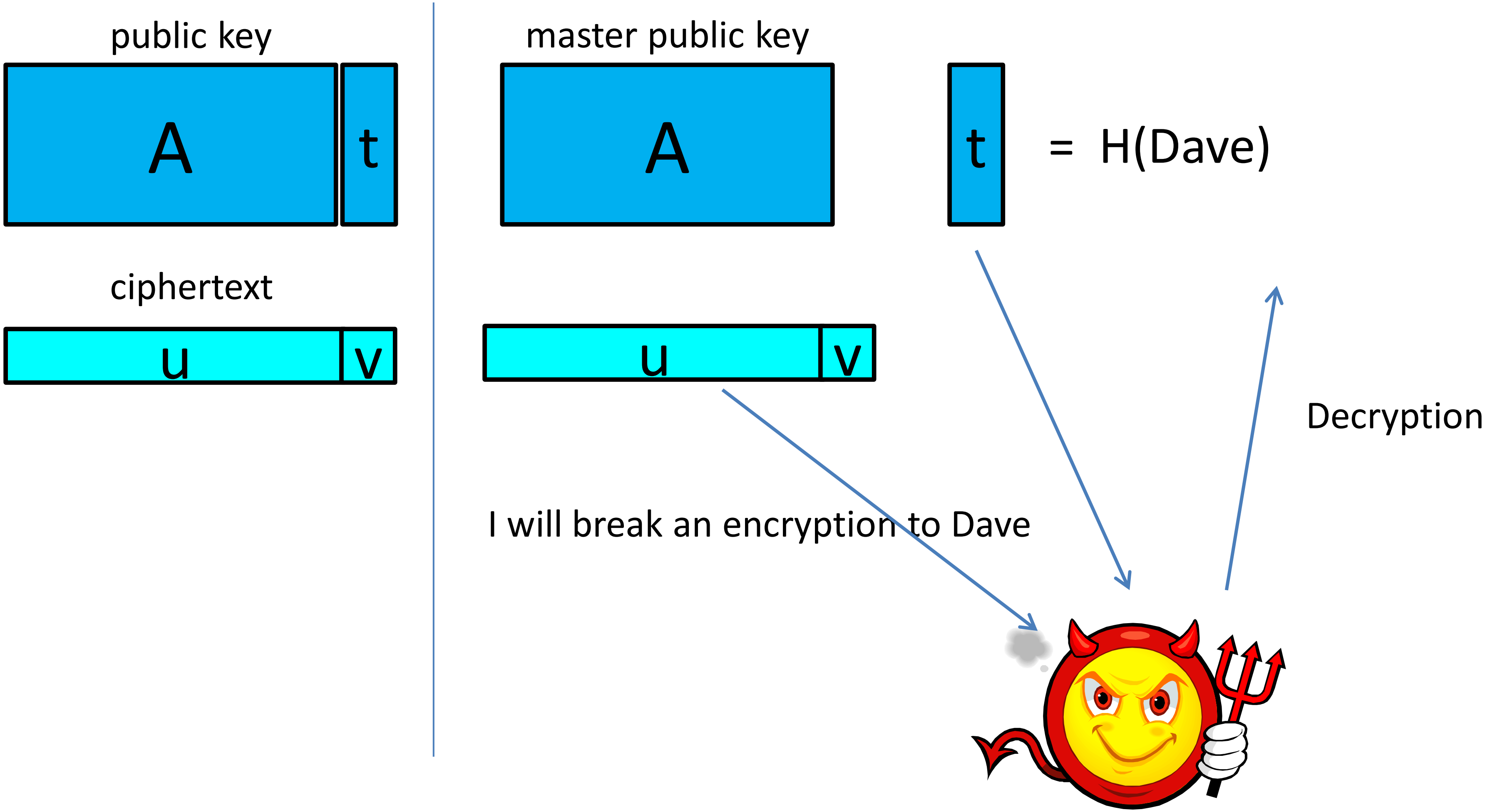
Security Proof Sketch

Show that breaking IBE implies breaking the "Dual" cryptosystem



Security Proof Sketch

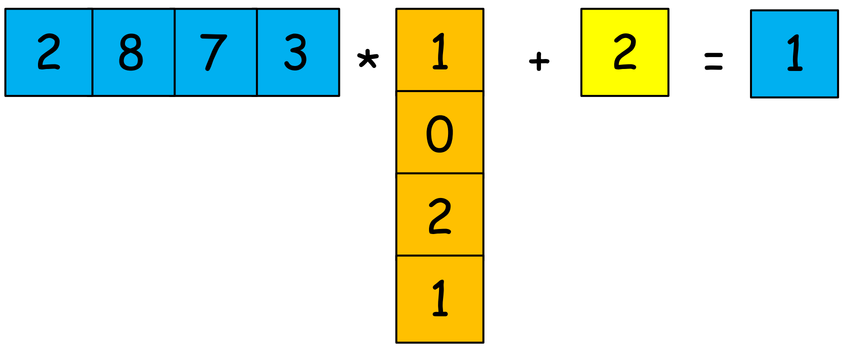
Show that breaking IBE implies breaking the “Dual” cryptosystem



LWE Encryption

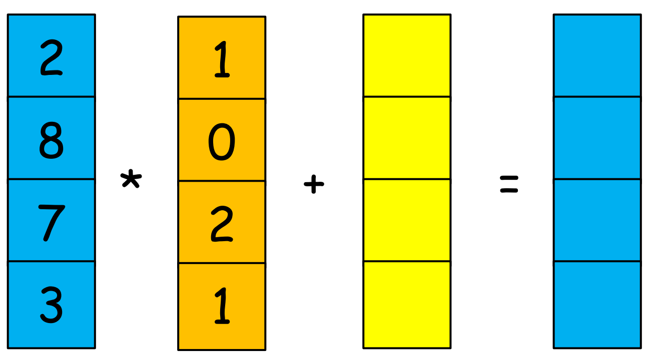
n-bit Encryption	Have	Want
Public Key Size	$\tilde{O}(n)$ / $\tilde{O}(n^2)$	O(n)
Secret Key Size	$\tilde{O}(n)$ / $\tilde{O}$ (n^2)	O(n)
Ciphertext Expansion	$\tilde{O}(n)$ / $\tilde{O}$ (1)	O(1)
Encryption Time	$\tilde{O}(n^3)$ / $\tilde{O}$ (n^2)	O(n)
Decryption Time	$\tilde{O}(n^2)$	O(n)

Source of Inefficiency of LWE



Getting just **one** extra random-looking number requires **n** random numbers and a small error element.

Wishful thinking: get **n** random numbers and produce **n** pseudo-random numbers in "one shot"



IDEAL LATTICES

Cyclic Lattices

A set L in $\mathbf{Z}^n$ is a *cyclic lattice* if:

1.) For all v,w in L, v+w is also in L

-1	2	3	-4

+

-7	-2	3	6

=

-8	0	6	2

2.) For all v in L, -v is also in L

-1	2	3	-4

1	-2	-3	4

3.) For all v in L, a cyclic shift of v is also in L

-1	2	3	-4
-4	-1	2	3
3	-4	-1	2
2	3	-4	-1

Cyclic Lattices = Ideals in $\mathbf{Z}[x]/(x^n-1)$

A set L in $\mathbf{Z}^n$ is a *cyclic lattice* if L is an *ideal* in $\mathbf{Z}[x]/(x^n-1)$

1.) For all v,w in L, v+w is also in L

-1	2	3	-4	+	-7	-2	3	6	=	-8	0	6	2

$(-1+2x+3x^2-4x^3) + (-7-2x+3x^2+6x^3) = (-8+0x+6x^2+2x^3)$

2.) For all v in L, -v is also in L

-1	2	3	-4		1	-2	-3	4

$(-1+2x+3x^2-4x^3)$ $(1-2x-3x^2+4x^3)$

3.) For all v in L, ~~a cyclic shift of v is also in L~~ vx is also in L

-1	2	3	-4	$-1+2x+3x^2-4x^3$
-4	-1	2	3	$(-1+2x+3x^2-4x^3)x=-4-x+2x^2+3x^3$
3	-4	-1	2	$(-1+2x+3x^2-4x^3)x^2 =3-4x-x^2+2x^3$
2	3	-4	-1	$(-1+2x+3x^2-4x^3)x^3 =2+3x-4x^2-x^3$

Why Cyclic Lattices?

- Succinct representations
  - Can represent an n-dimensional lattice with 1 vector
- Algebraic structure
  - Allows for fast arithmetic (using FFT)
  - Makes proofs possible
- One-way functions based on worst-case hardness of SVP in cyclic lattices [Mic02]

Shortest Vector Problem (SVP)

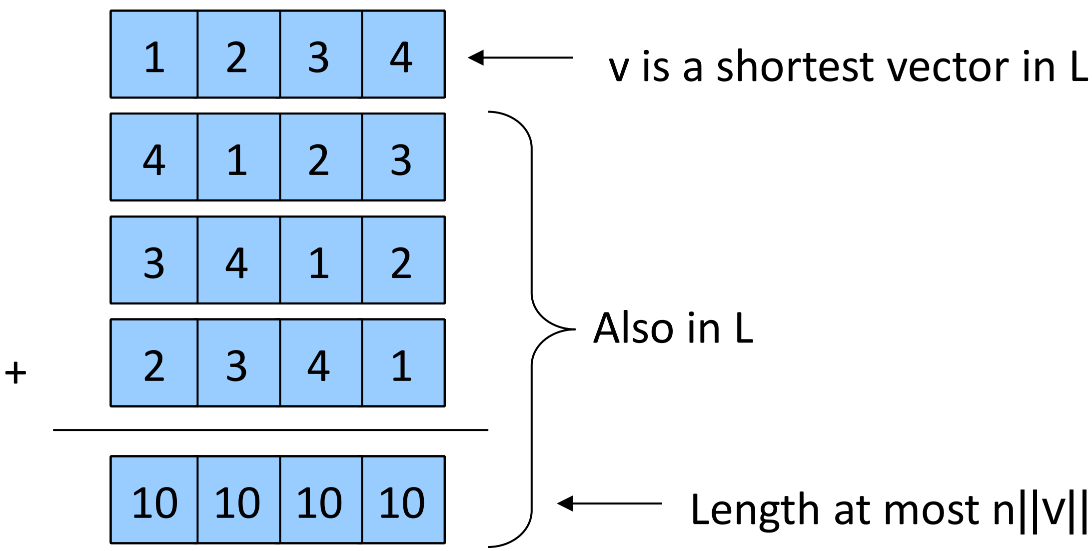
- SVP: Given a lattice L, find the (non-zero) vector with the smallest norm in L

- SVP_γ : Given a lattice L, find a non-zero vector whose length is within a factor γ of the shortest vector

Is $\text{SVP}_{\text{poly}(n)}$ Hard for Cyclic Lattices?

Short answer: we don't know but conjecture it is.

What's wrong with the following argument that SVP_n is easy?



Algorithm for solving $\text{SVP}_n(L)$ for a cyclic lattice L:

1. Construct 1-dimensional lattice $L' = L \cap \{1^n\}$
2. Find and output the shortest vector in L'

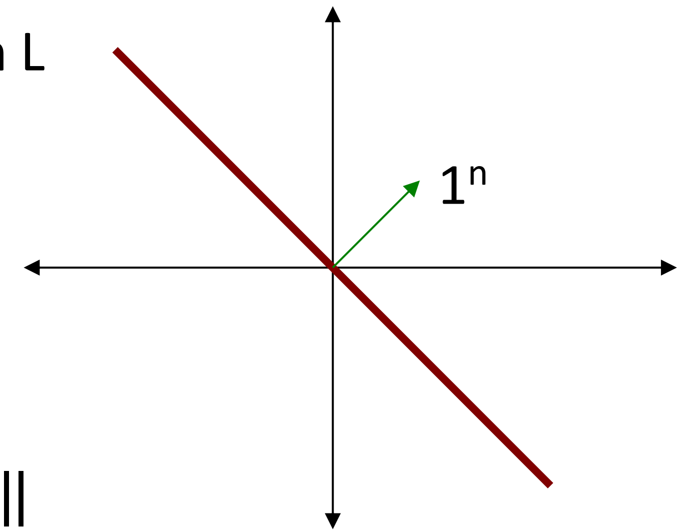
The Hard Cyclic Lattice Instances

	-1	2	3	-4
	-4	-1	2	3
	3	-4	-1	2
+	2	3	-4	-1
	0	0	0	0

v is a shortest vector in L

Also in L

Length at most n||v||



The "hard" instances of cyclic lattices lie on plane P perpendicular to the 1^n vector

In algebra language:

If $R=\mathbf{Z}[x]/(x^n-1)$, then

$1^n = (x^{n-1}+x^{n-2}+\ldots+1) \approx \mathbf{Z}[x]/(x-1)$

$P = (x-1) \approx \mathbf{Z}[x]/(x^{n-1}+x^{n-2}+\ldots+1)$

f-Ideal Lattices = Ideals in **Z**[x]/(f)

Want f to have 3 properties:

1) Monic (i.e. coefficient of largest exponent is 1)
2) Irreducible over **Z**
3) For all polynomials g,h $\|gh \bmod f\| < \mathrm{poly}(n)\|g\|\cdot\|h\|$

Conjecture: For all f that satisfy the above 3 properties, solving $SVP_{poly(n)}$ for ideals in **Z**[x]/(f) takes time $2^{\Omega(n)}$.

Some "good" f to use:

$f=x^{n-1}+x^{n-2}+...+1$ where n is prime

$f=x^n+1$ where n is a power of 2

(x^n+1)-Ideal Lattices = Ideals in $\mathbf{Z}[x]/(x^n+1)$

A set L in $\mathbf{Z}^n$ is a *(x^n+1)-ideal lattice* if L is an *ideal* in $\mathbf{Z}[x]/(x^n+1)$

1.) For all v,w in L, v+w is also in L

-1	2	3	-4

\+

-7	-2	3	6

=

-8	0	6	2

$(-1+2x+3x^2-4x^3) + (-7-2x+3x^2+6x^3) = (-8+0x+6x^2+2x^3)$

2.) For all v in L, -v is also in L

-1	2	3	-4

1	-2	-3	4

$(-1+2x+3x^2-4x^3)$ $(1-2x-3x^2+4x^3)$

3.) For all v in L, vx is also in L

-1	2	3	-4	$-1+2x+3x^2-4x^3$
4	-1	2	3	$(-1+2x+3x^2-4x^3)x=4-x+2x^2+3x^3$
-3	4	-1	2	$(-1+2x+3x^2-4x^3)x^2 =-3+4x-x^2+2x^3$
-2	-3	4	-1	$(-1+2x+3x^2-4x^3)x^3 =-2-3x+4x^2-x^3$

RING-LWE

Ring-LWE

Ring $R=Z_q[x]/(x^n+1)$

Given:

a_1, a_1s+e_1

a_2, a_2s+e_2

…

a_k, a_ks+e_k

Find: s

a_i are random in R

s and e_i have "small" coefficients (distribution symmetric around 0)

Decision Ring-LWE

Ring $R=Z_q[x]/(x^n+1)$

Given:

a_1, b_1

a_2, b_2

…

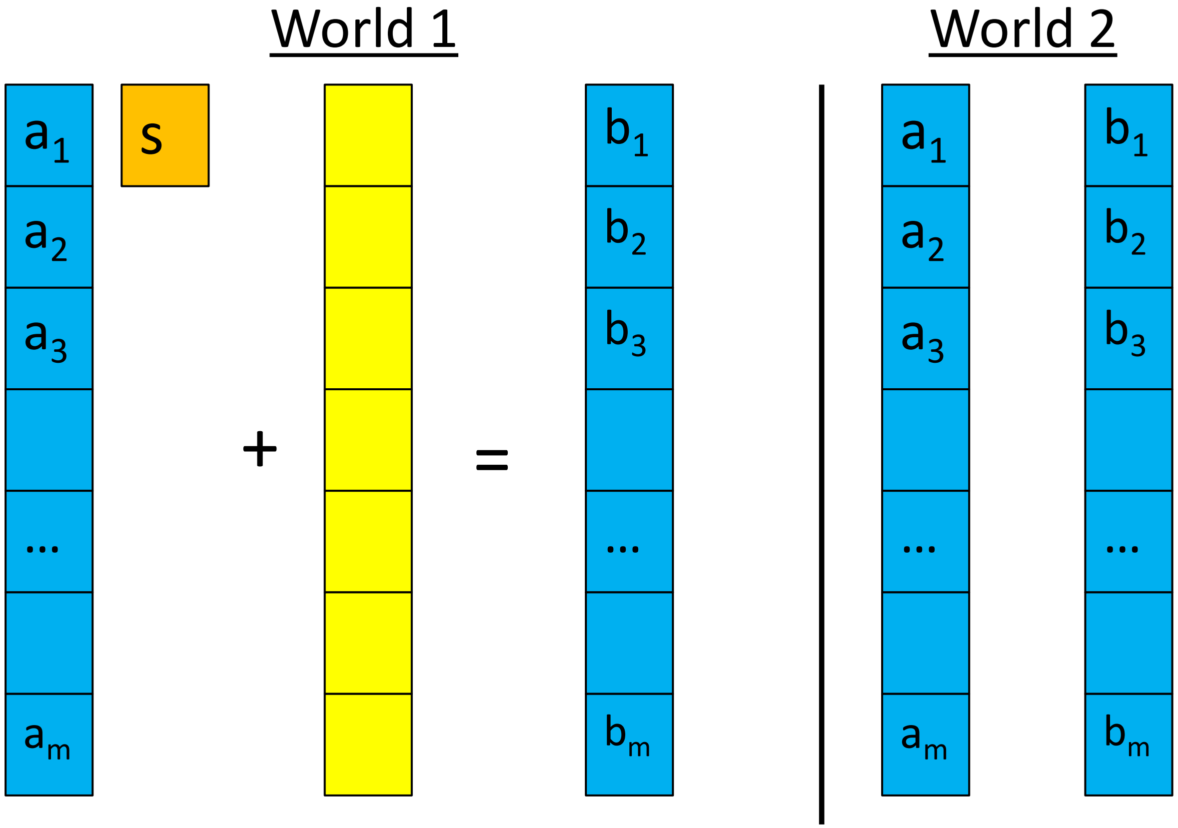
a_k, b_k

Question: Does there exist "small" s and

$e_1, \ldots, e_k$ such that $b_i=a_is+e_i$

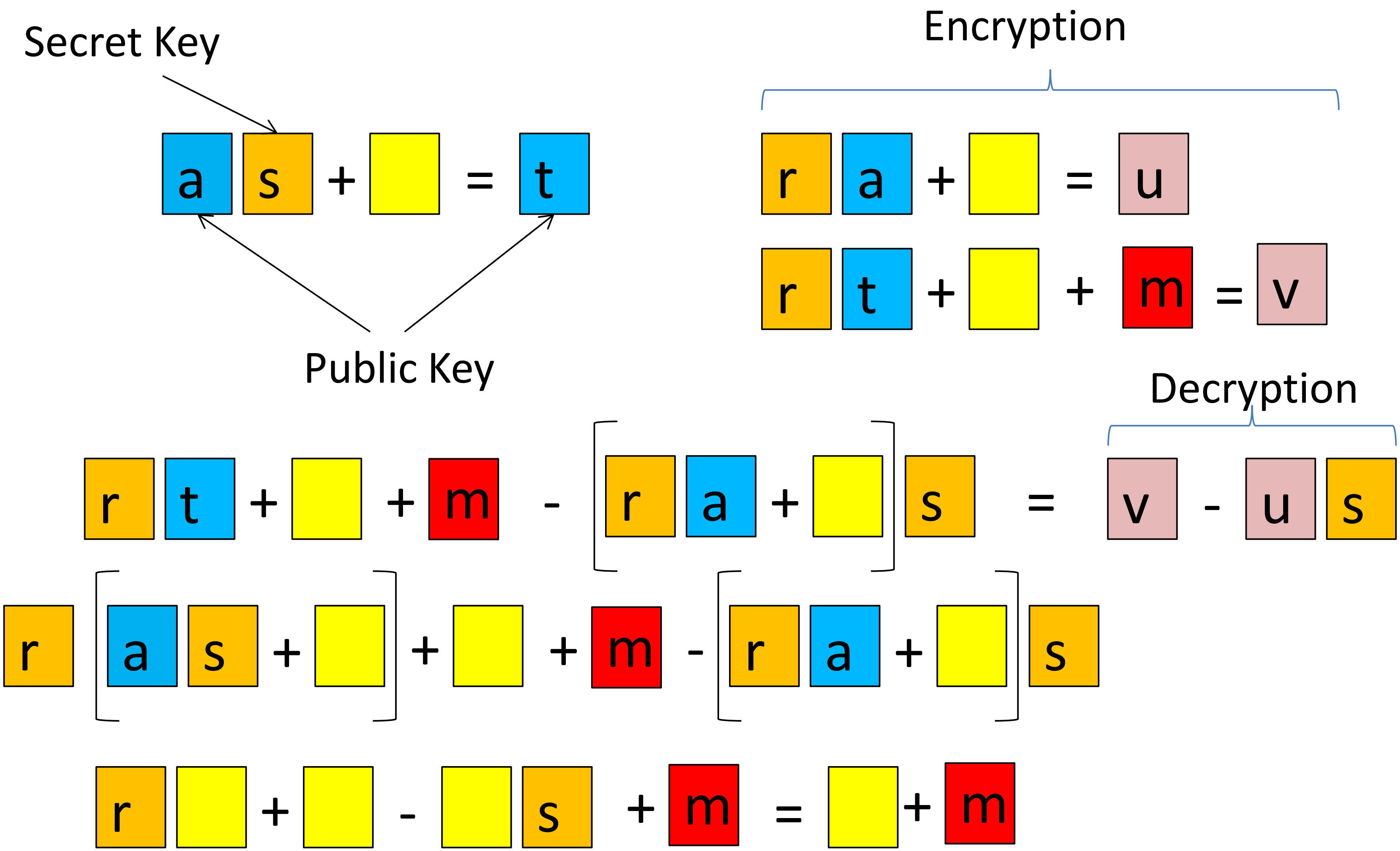
or are all b_i uniformly random in R?

Decision
Learning With Errors over Rings

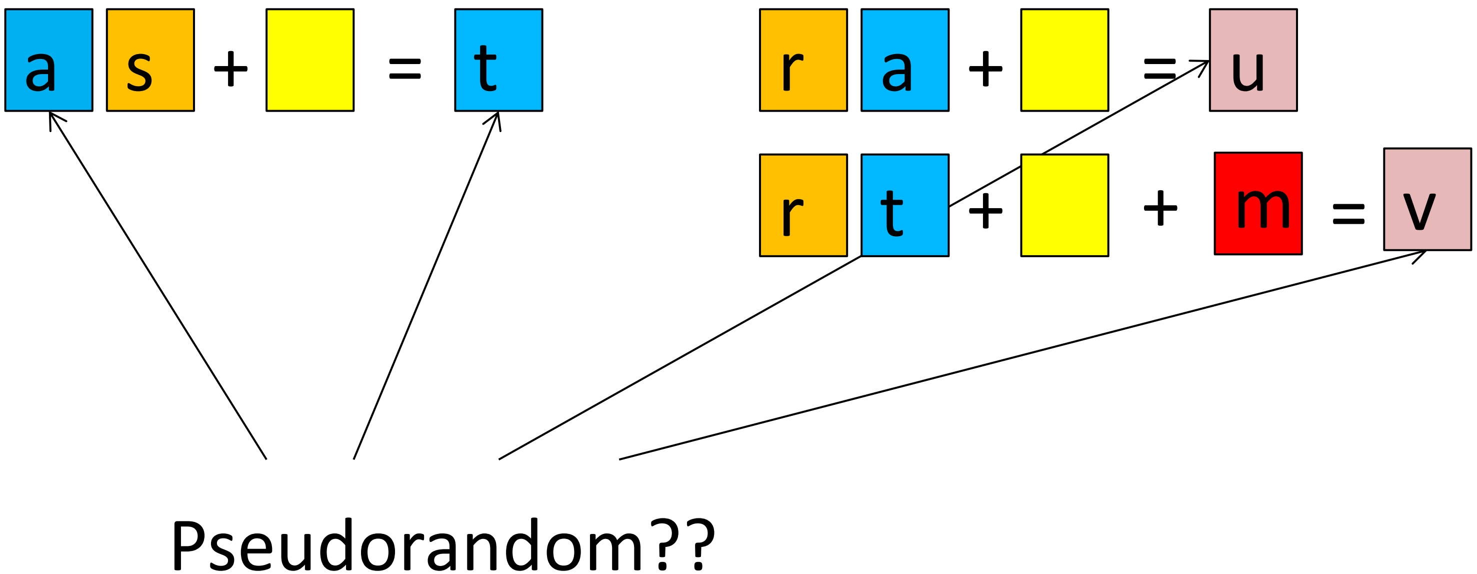


Theorem [LPR '10]: In *cyclotomic* rings, there is a quantum reduction from solving worst-case problems in ideal lattices to solving Decision-RLWE

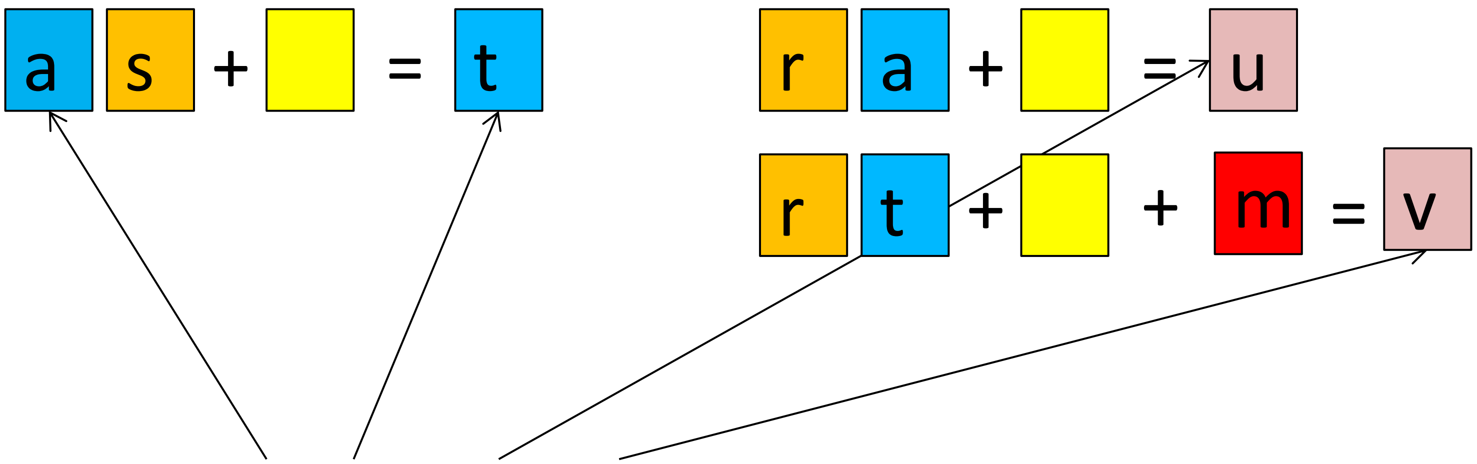
Ring-LWE cryptosystem



Security



Security



Pseudorandom based on
Decision Ring-LWE!!

Efficiency

$$a \, s + \square = t$$

$$r \, a + \square = u$$

$$r \, t + \square + m = v$$

n-bit Encryption	From LWE	From Ring-LWE
Public Key Size	$\tilde{O}(n)$ / $\tilde{O}(n^2)$	$\tilde{O}(n)$
Secret Key Size	$\tilde{O}(n)$ / $\tilde{O}(n^2)$	$\tilde{O}(n)$
Ciphertext Expansion	$\tilde{O}(n)$ / $\tilde{O}(1)$	$\tilde{O}(1)$
Encryption Time	$\tilde{O}(n^3)$ / $\tilde{O}(n^2)$	$\tilde{O}(n)$
Decryption Time	$\tilde{O}(n^2)$	$\tilde{O}(n)$

DIGITAL SIGNATURE SCHEMES

Digital Signatures

$(sk,pk) \leftarrow$ KeyGen
$\text{Sign}(sk,m_i) = s_i$
$\text{Verify}(pk,m_i,s_i) = \text{YES / NO}$

Correctness: $\text{Verify}(pk, m_i, \text{Sign}(sk,m_i)) = \text{YES}$
Security: Unforgeability

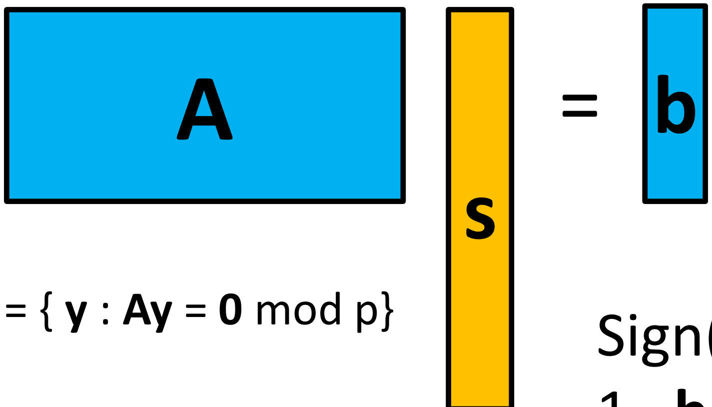
1. Adversary gets pk
2. Adversary asks for signatures of m_1, m_2, …
3. Adversary outputs (m,s) where $m \neq m_i$ and wins if $\text{Verify}(pk,m,s) = \text{YES}$

Signature Schemes

- Hash-and-Sign
  - Requires a trap-door function (like the GPV one)
- Fiat-Shamir transformation
  - Conversion from an identification scheme
  - No trap-door function needed

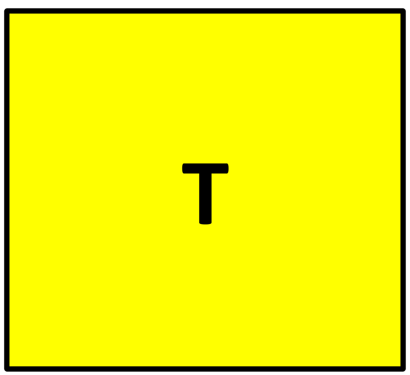
HASH-AND-SIGN SIGNATURE SCHEMES BASED ON SIS

Hash-and-Sign Lattice Signature



Public Key: **A**
Secret Key: **T**

Lattice $L_p^{\perp}(\mathbf{A}) = \{\, \mathbf{y} : \mathbf{A}\mathbf{y} = \mathbf{0} \bmod p \,\}$

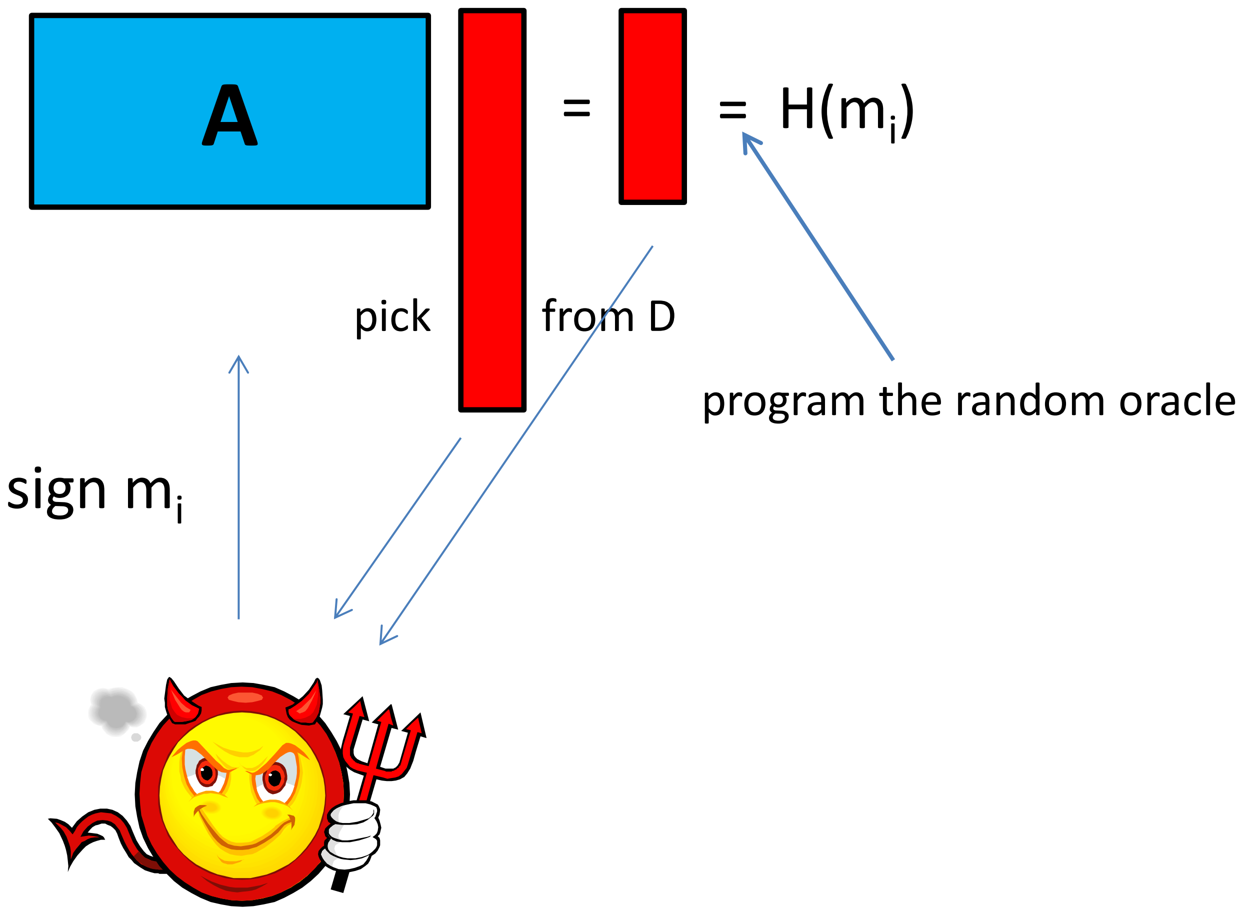


T is a basis for $L_p^{\perp}(\mathbf{A})$ and has “short” vectors

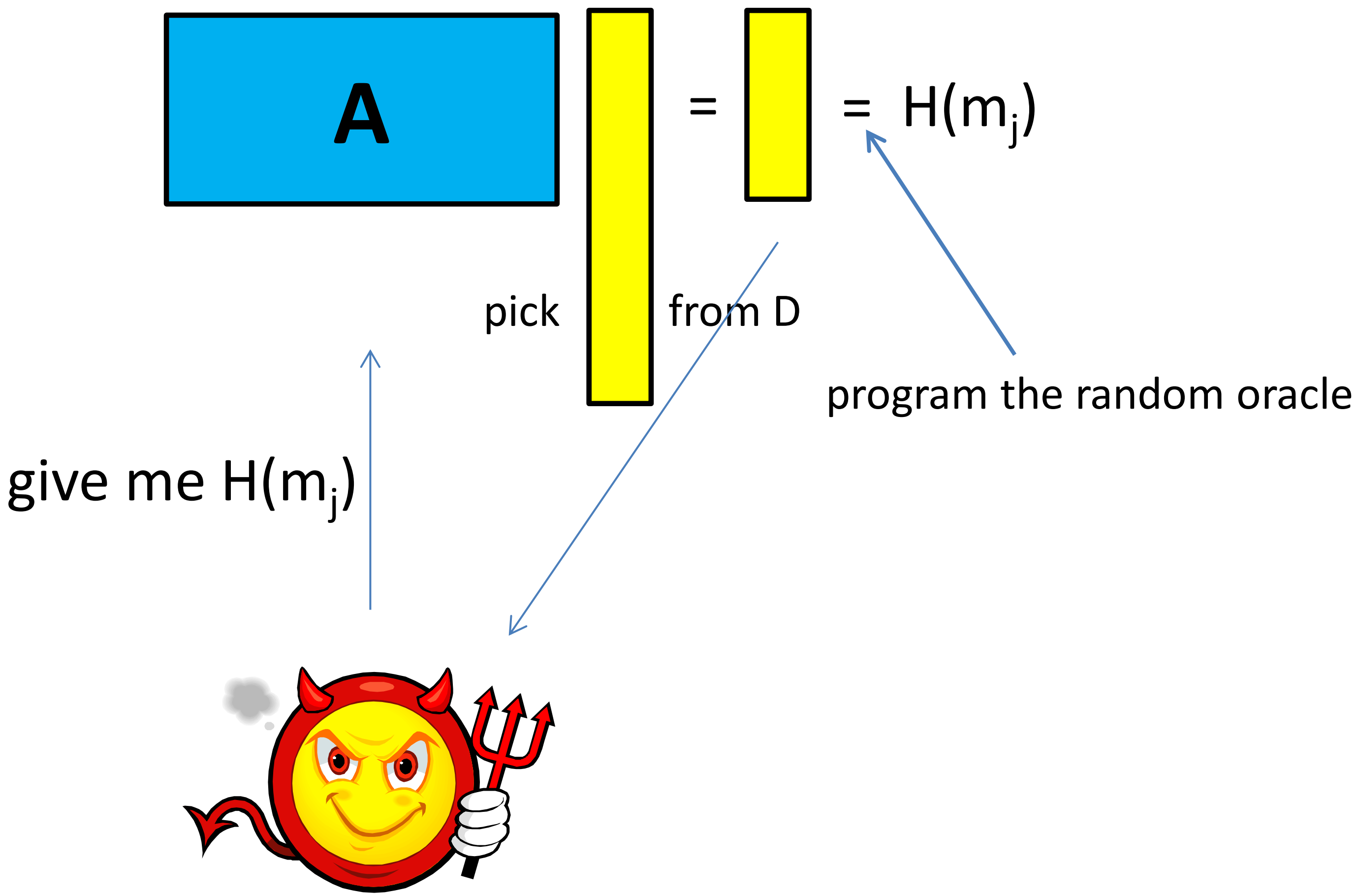
Sign(**T**,m)
1. **b** = H(m)
2. Use the GPV algorithm to find a short **s** such that **As** = **b** mod p
3. **s** is the signature of m

Verify(**A**,m,**s**)
1. check that **s** is “short” and
   As = H(m) mod p

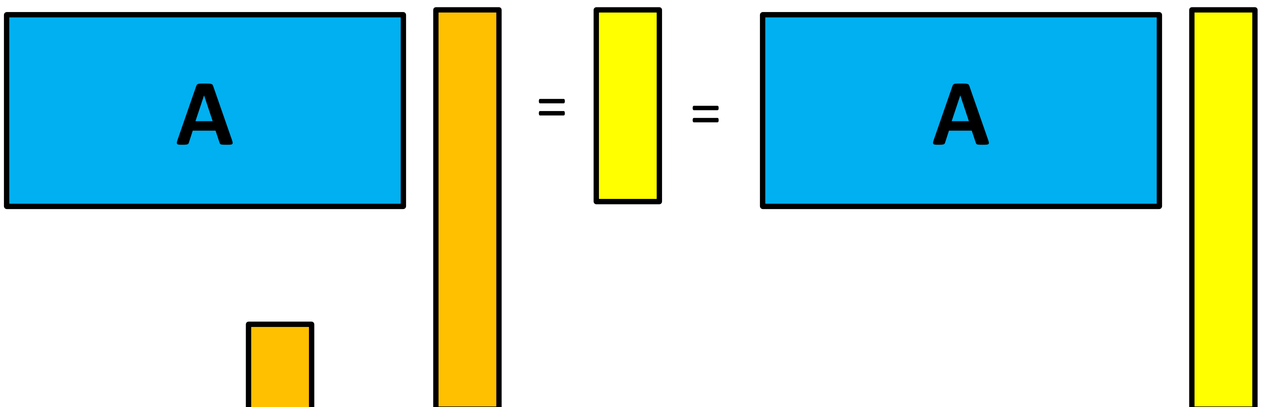
Security Proof Sketch

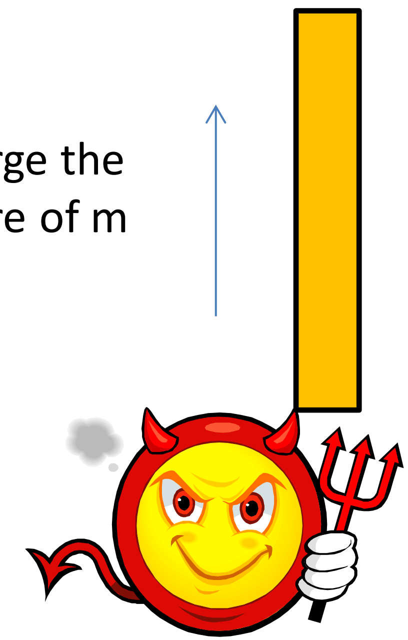


Security Proof Sketch



Security Proof Sketch





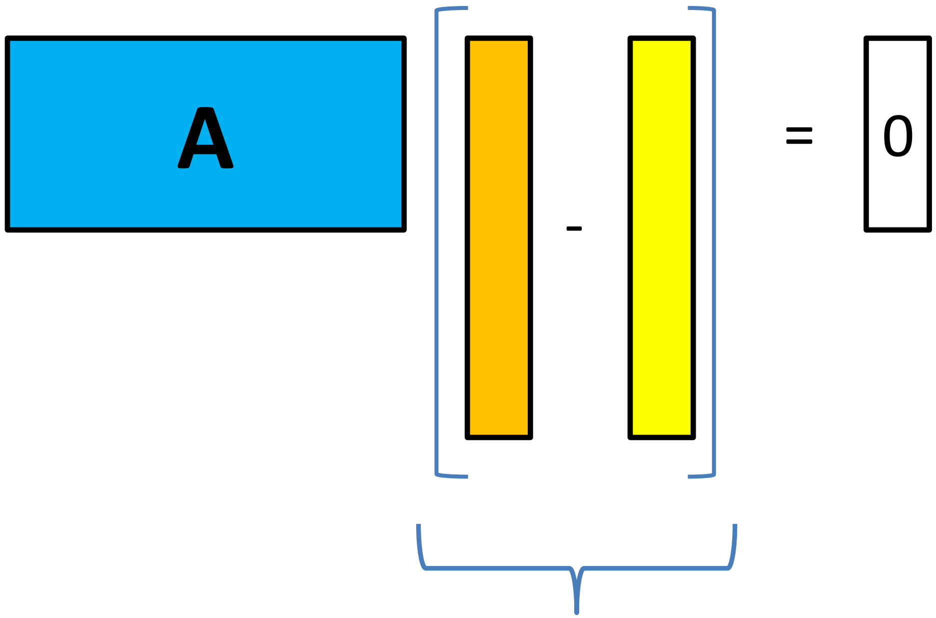
I will forge the signature of m

To forge on m, the Adversary needs H(m)

So m is one of the m_j he asked for $H(m_j)$

Thus we know an s_j such that $As_j=H(m_j)$

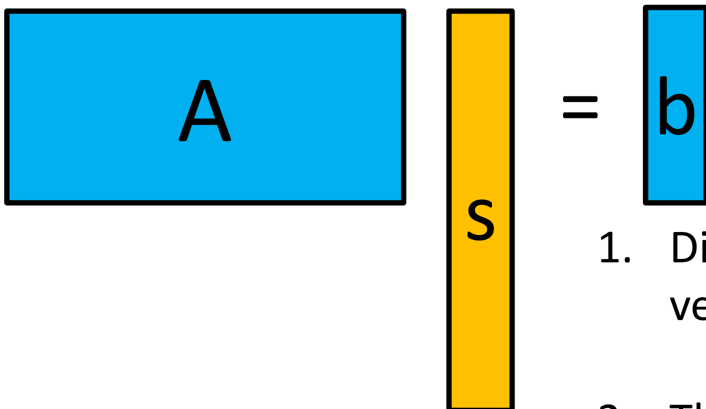
Security Proof Sketch

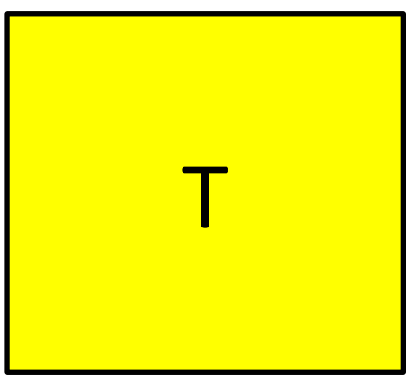


short and hopefully non-zero

if it’s non-zero, then we have a solution to SIS

Properties Needed





1. Distribution D of **s** only depends on the length of the vectors comprising **T**

2. The following produce the same distribution of (**s**,**b**)

   (a) Choose **s** ~ D. Set **b**=**As**

   (b) Choose random **b**. Use **T** to find an **s** such that **As**=**b**.

3. For a random **b**, there is more than one likely possible output **s** such that **b**=**As**.

(1) is guaranteed by the GPV algorithm

(2) is true if **s** has enough entropy (to make A**s**=**b** uniform mod p)

(3) is true because the standard deviation of GPV is big

IDENTIFICATION AND “FIAT-SHAMIR” SIGNATURE SCHEMES BASED ON SIS

Canonical 3-move Identification Scheme

Prover (sk) Verifier (pk)

$$\xrightarrow{\text{commit}}$$

$$\xleftarrow{\text{challenge}}$$

$$\xrightarrow{\text{response}}$$

Verify(pk, commit, challenge,response)=1?

Security of ID Schemes

Passive Adversary

1. Receive public key
2. Receive interaction transcripts
3. Try to impersonate the valid prover

Active Adversary

1. Receive public key
2. Interact with the valid prover
3. Try to impersonate the valid prover

Fiat-Shamir Transform

Passively-Secure 3-round scheme →

Signature scheme in the random oracle model

Sign(μ)
    commit
    challenge = H(μ, commit)
    response
    (commit,challenge,response)

VerifySig(μ, pk, commit, challenge, response)
    challenge=H(μ, commit)?
    VerifyID(pk, commit, challenge, response)=1?

Identification Scheme Based on SIS

Secret Key: **S**

Public Key: **A**, **T**=**AS** mod q

Pick a random **y**

$\xrightarrow{\mathbf{w}=\mathbf{A}\mathbf{y} \bmod q}$

pick a random **c**

$\xleftarrow{\mathbf{c}}$

$\xrightarrow{\mathbf{z}=\mathbf{S}\mathbf{c}+\mathbf{y}}$

check that

1. $||\mathbf{z}||$ is small
2. $\mathbf{A}\mathbf{z}=\mathbf{T}\mathbf{c}+\mathbf{w} \bmod q$

($\mathbf{A}\mathbf{z}=\mathbf{A}(\mathbf{S}\mathbf{c}+\mathbf{y})=\mathbf{T}\mathbf{c}+\mathbf{w}$)

Active Security Reduction (Stage 1)

A

Adversary

Secret Key: **S**

Public Key: **A**, **T**=**AS** mod q

Pick a random **y**

w=**Ay** mod q $\rightarrow$

c $\leftarrow$

z=Sc+y $\rightarrow$

Active Security Reduction (Stage 2)

Adversary

Public Key: $\mathbf{A}, \mathbf{T}=\mathbf{A}\mathbf{S} \bmod q$

$\xrightarrow{\mathbf{w}}$

$\xleftarrow{\mathbf{c}}$ $\xleftarrow{\mathbf{c}'}$

$\xrightarrow{\mathbf{z}}$ $\xrightarrow{\mathbf{z}'}$

$\mathbf{A}\mathbf{z}=\mathbf{T}\mathbf{c}+\mathbf{w} \bmod q$

$\mathbf{A}\mathbf{z}'=\mathbf{T}\mathbf{c}'+\mathbf{w} \bmod q$

$\mathbf{A}(\mathbf{z}-\mathbf{z}')=\mathbf{T}(\mathbf{c}-\mathbf{c}') \bmod q$

$\mathbf{A}(\mathbf{z}-\mathbf{z}')=\mathbf{A}\mathbf{S}(\mathbf{c}-\mathbf{c}') \bmod q$

Observation: If the adversary knows $\mathbf{S}$, then he can always give us $\mathbf{z}-\mathbf{z}' = \mathbf{S}(\mathbf{c}-\mathbf{c}')$

Solution: Make sure adversary does not learn $\mathbf{S}$

Hope: $\mathbf{z}-\mathbf{z}' \neq \mathbf{S}(\mathbf{c}-\mathbf{c}')$

Identification Scheme Based on SIS

Secret Key: $\mathbf{S}$

Public Key: $\mathbf{A}$, $\mathbf{T}=\mathbf{AS} \bmod q$

Pick a random $\mathbf{y}$

$\xrightarrow{\mathbf{w}=\mathbf{Ay} \bmod q}$

pick a random $\mathbf{c}$

$\xleftarrow{\mathbf{c}}$

Make $\mathbf{y}$ uniform??
NO! Then $\mathbf{z}$ is too big
and SIS is not hard.

$\xrightarrow{\mathbf{z}=\mathbf{Sc}+\mathbf{y}}$

check that

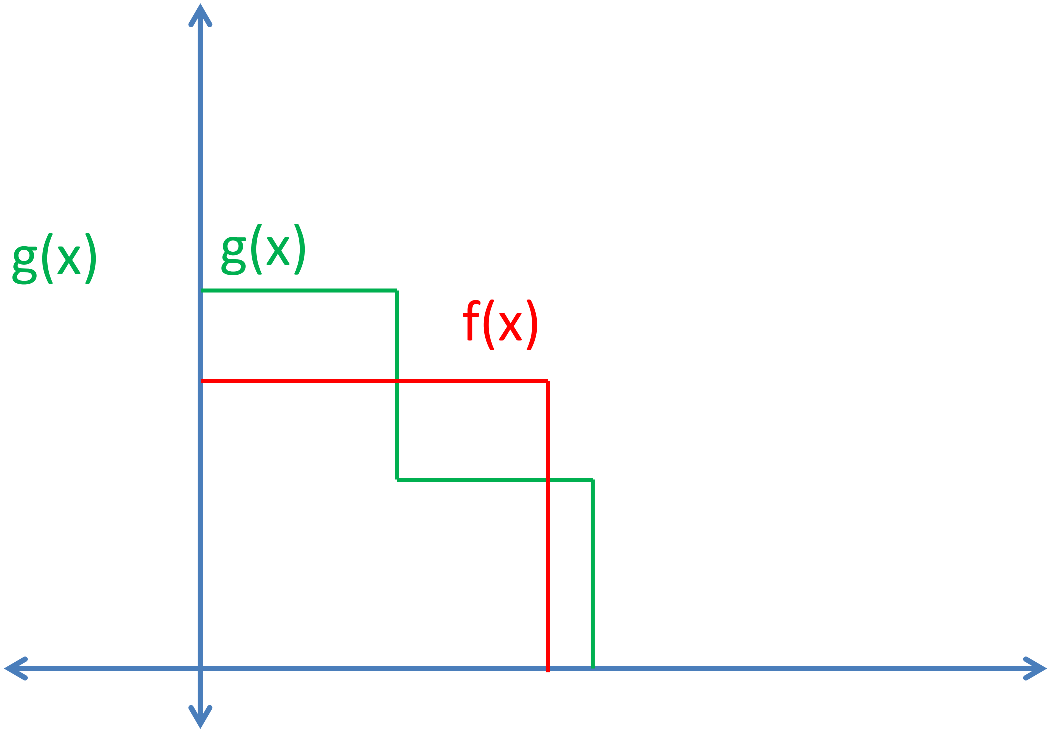
1. $\|\mathbf{z}\|$ is small
2. $\mathbf{Az}=\mathbf{Tc}+\mathbf{w} \bmod q$

($\mathbf{Az}=\mathbf{A}(\mathbf{Sc}+\mathbf{y})=\mathbf{Tc}+\mathbf{w}$)

Rejection Sampling

Have access to samples from g(x)

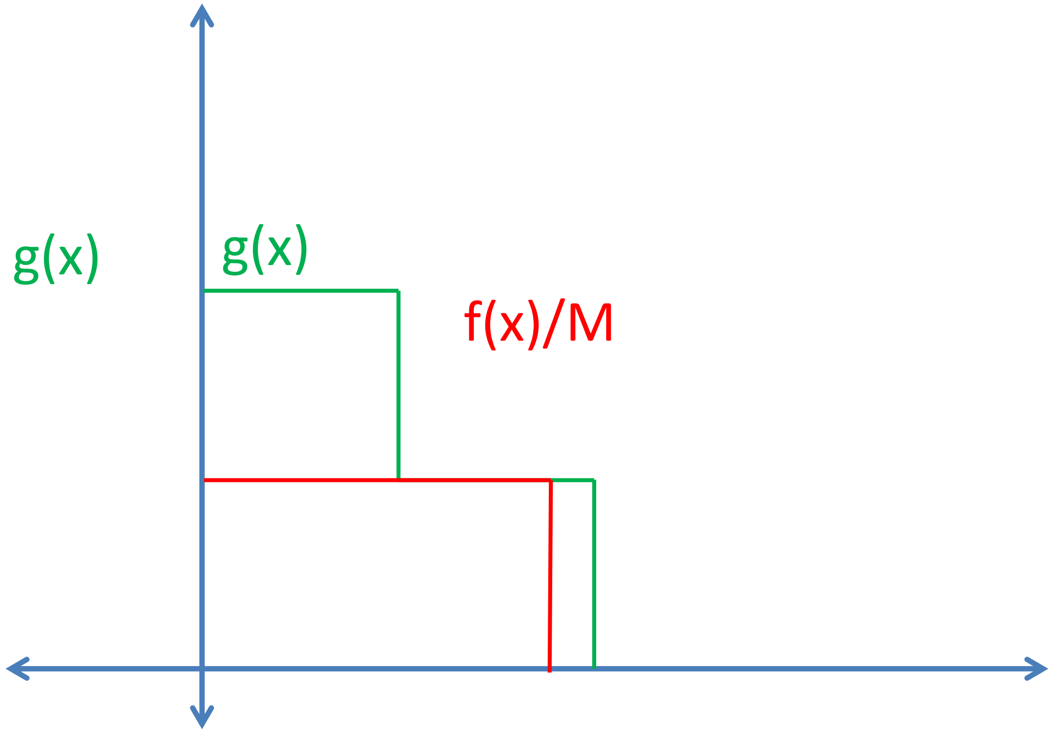
Want f(x)



Rejection Sampling

Have access to samples from g(x)

Want f(x)



Sample from g(x), accept x with probability $f(x)/Mg(x) \leq 1$

$\Pr[x] = g(x)\cdot(f(x)/Mg(x)) = f(x)/M$

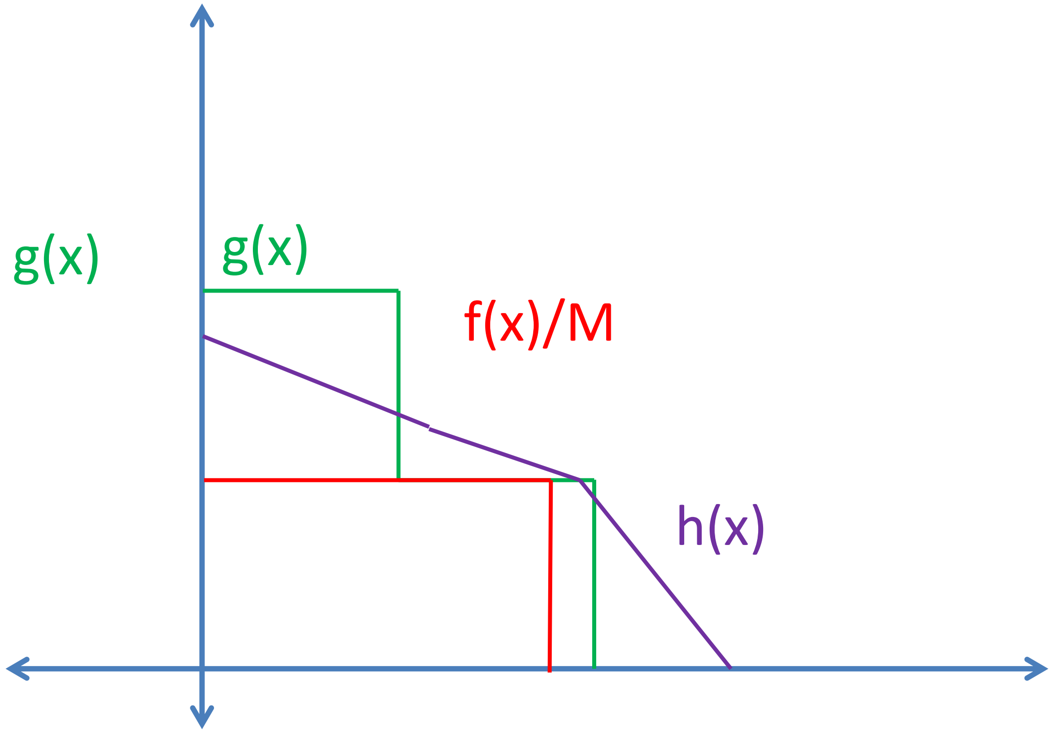
Something is output with probability 1/M

Rejection Sampling

Impossible to tell whether g(x) or h(x) was the original distribution

Have access to samples from g(x)

Want f(x)



Sample from g(x), accept x with probability f(x)/Mg(x) ≤ 1
or … Sample from h(x), accept x with probability f(x)/Mh(x) ≤ 1
Pr[x] = g(x)·(f(x)/Mg(x)) = f(x)/M = h(x)·(f(x)/Mh(x))
Something is output with probability 1/M

Rejection Sampling

Secret Key: **S**

y ~ f

w=**Ay** mod q

c

z=**Sc**+**y**

Rejection Sampling

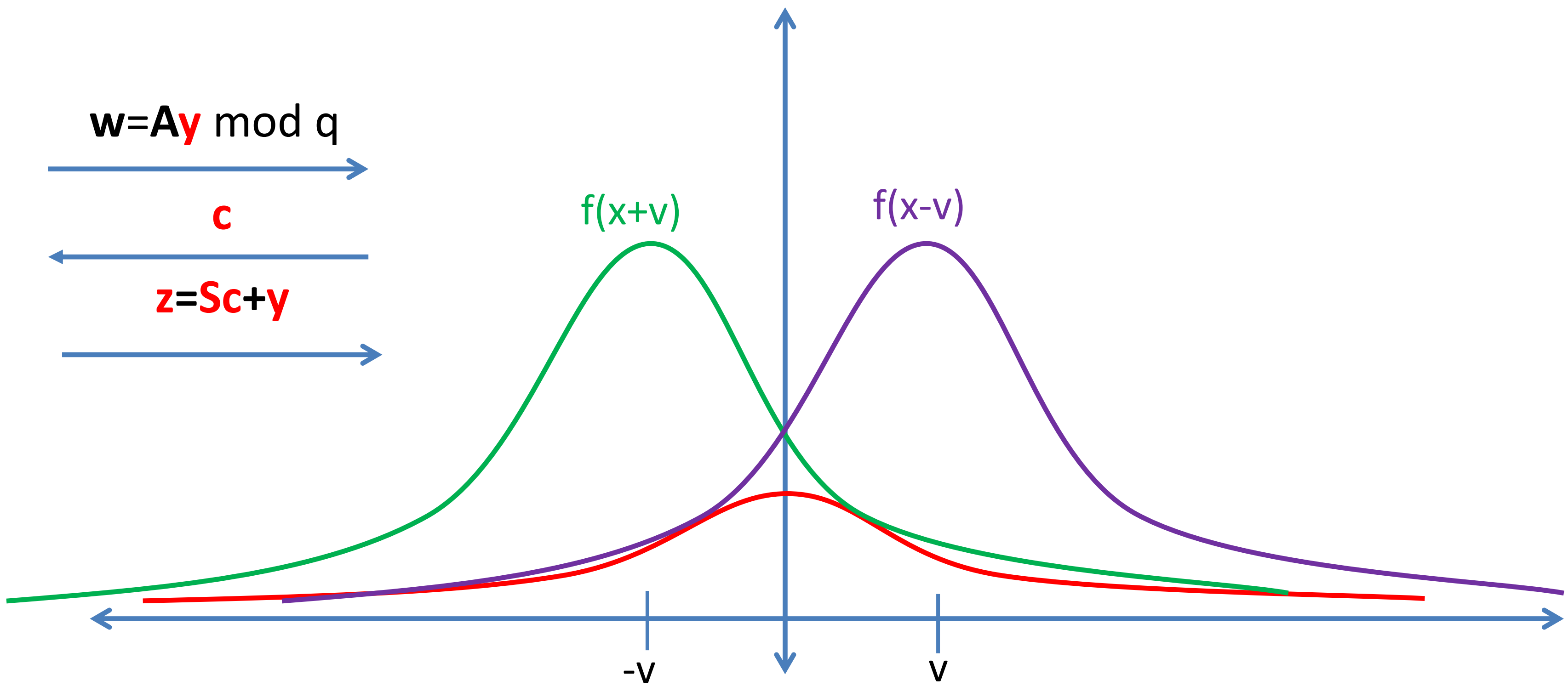
Secret Key: **S**

y ~ f

w=**Ay** mod q

c

z=Sc+y



v=max ||**Sc**||

What We Want

Choose a target distribution f for **z**
Choose a distribution g for **y**
Distribution of **z** will be g(**y**-**Sc**)
Rejection sample to make the distribution of **z** be f

Need:
For all *likely* **x** and **Sc**, f(**x**)/M ≤ g(**x**-**Sc**)

Want:
1. M to be as small as possible (1/M is acceptance rate)
2. E[||**x**|| ; **x** ~ f] to be as small as possible (determines size of signature and hardness of SIS problem)

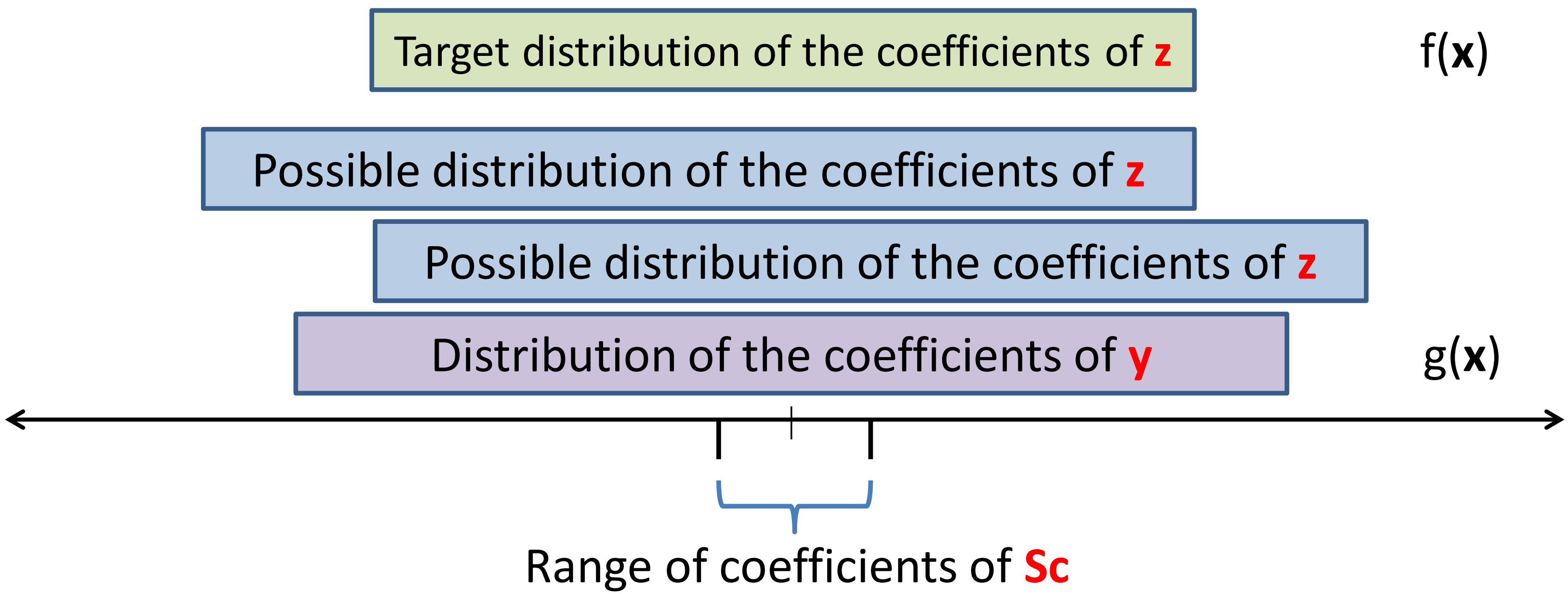
Secret Key: **S**

w=**Ay** mod q $\longrightarrow$

c $\longleftarrow$

z=**Sc**+**y** $\longrightarrow$

Rejection Sampling (L '09)



Probability each coefficient of **z** is in the target range = p

Want $p^m \approx$ constant

So $p \approx 1-1/m$

So coefficients of **Sc** must be m times smaller than coefficients of **y**

Size of the **SIS** solution

Coefficients of **Sc** = O(1)

Coefficients of **y** = O(m)

$||\mathbf{z}|| \approx ||\mathbf{y}|| = O(m^{1.5})$

Can we do better??

Use Normal distribution to get $||\mathbf{z}|| = O(m)$

Normal Distribution

1-dimensional Normal distribution:

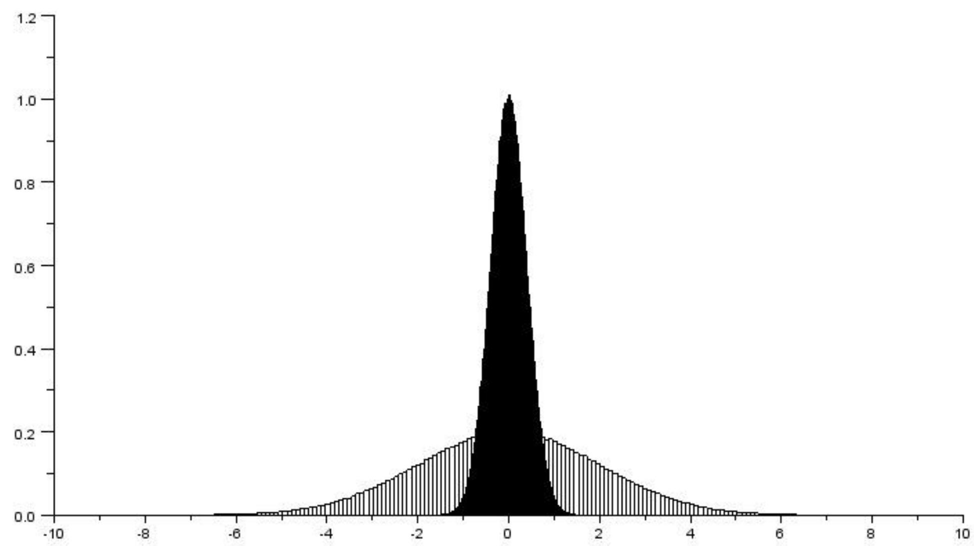
$$\rho_\sigma(x) = 1/(\sqrt{2\pi}\sigma)e^{-x^2/2\sigma^2}$$

It is:

Centered at 0

Standard deviation: σ

Examples



Shifted Normal Distribution

1-dimensional shifted Normal distribution:

$$\rho_{\sigma,v}(x) = 1/(\sqrt{2\pi}\sigma)e^{-(x-v)^2/2\sigma^2}$$

It is:

Centered at v

Standard deviation: σ

n-Dimensional Normal Distribution

n-dimensional shifted Normal distribution:

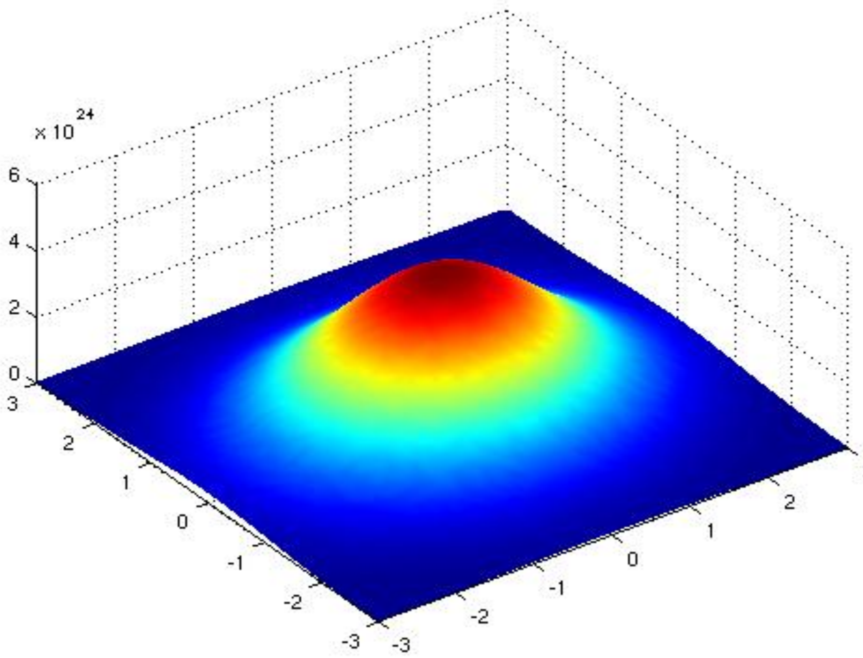
$$\rho_{\sigma,\mathbf{v}}(\mathbf{x}) = 1/(\sqrt{2\pi}\sigma)^n e^{-||\mathbf{x}-\mathbf{v}||^2/2\sigma^2}$$

It is:

Centered at **v**

Standard deviation: σ

2-Dimensional Example



n-Dimensional Normal Distribution

n-dimensional shifted Normal distribution:

$$\rho_{\sigma,\mathbf{v}}(\mathbf{x}) = 1/(\sqrt{2\pi}\sigma)^n e^{-||\mathbf{x}-\mathbf{v}||^2/2\sigma^2}$$

It is:

Centered at **v**

Standard deviation: σ

Discrete Normal: for **x** in $\mathbf{Z}^n$,

$$D_{\sigma,\mathbf{v}}(\mathbf{x}) = \rho_{\sigma,\mathbf{v}}(\mathbf{x}) / \rho_{\sigma,\mathbf{v}}(\mathbf{Z}^n)$$

New Rejection Sampling

$g(x)=f(x) = D_{\sigma,\mathbf{0}}(\mathbf{x})$

Lemma: If $\sigma = k||v||$, then with very high probability, for all likely $\mathbf{x} \sim f$,

$$D_{\sigma,\mathbf{0}}(\mathbf{x}) \,/\, D_{\sigma,\mathbf{v}}(\mathbf{x}) < e^{12/k}$$

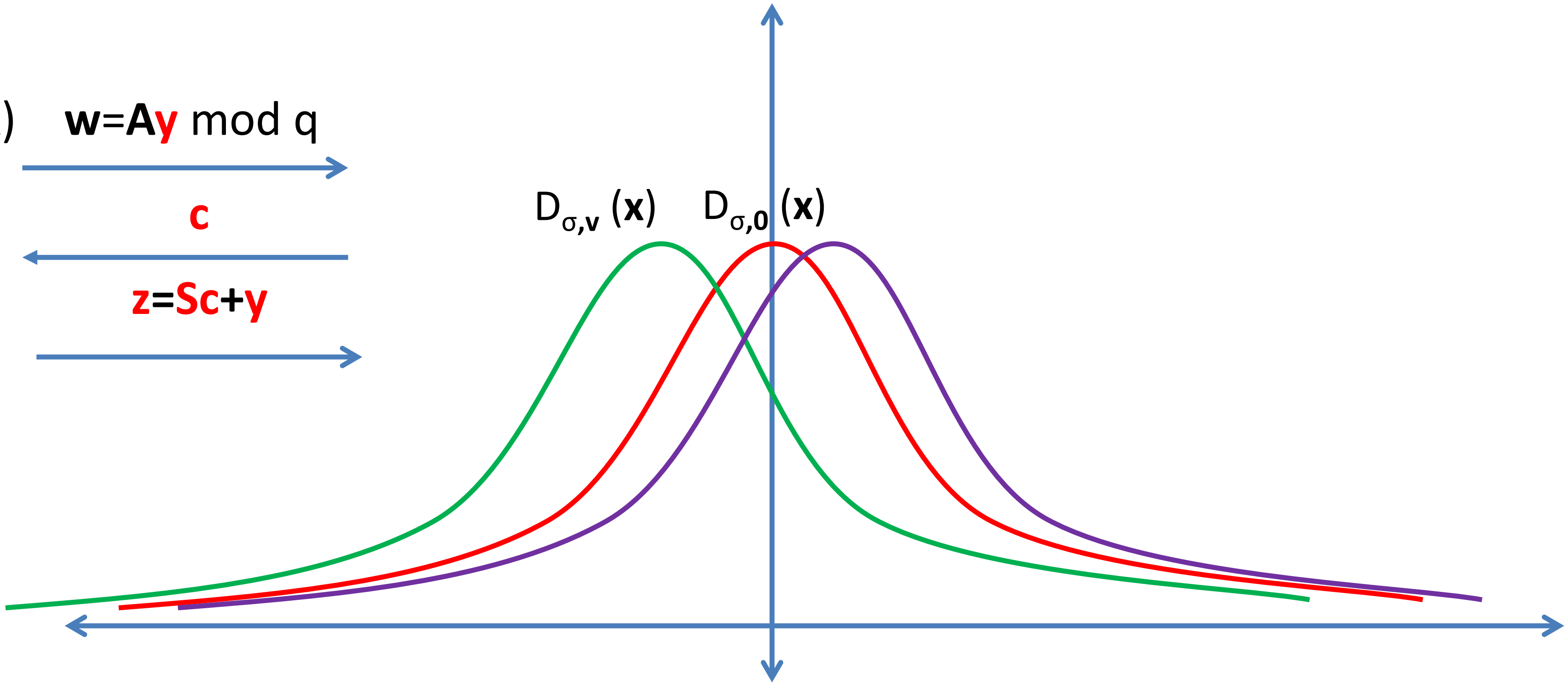
Rejection Sampling

Secret Key: **S**

y ~ $D_{\sigma,\mathbf{0}}(\mathbf{x})$ **w**=**Ay** mod q

c

z=**Sc**+**y**



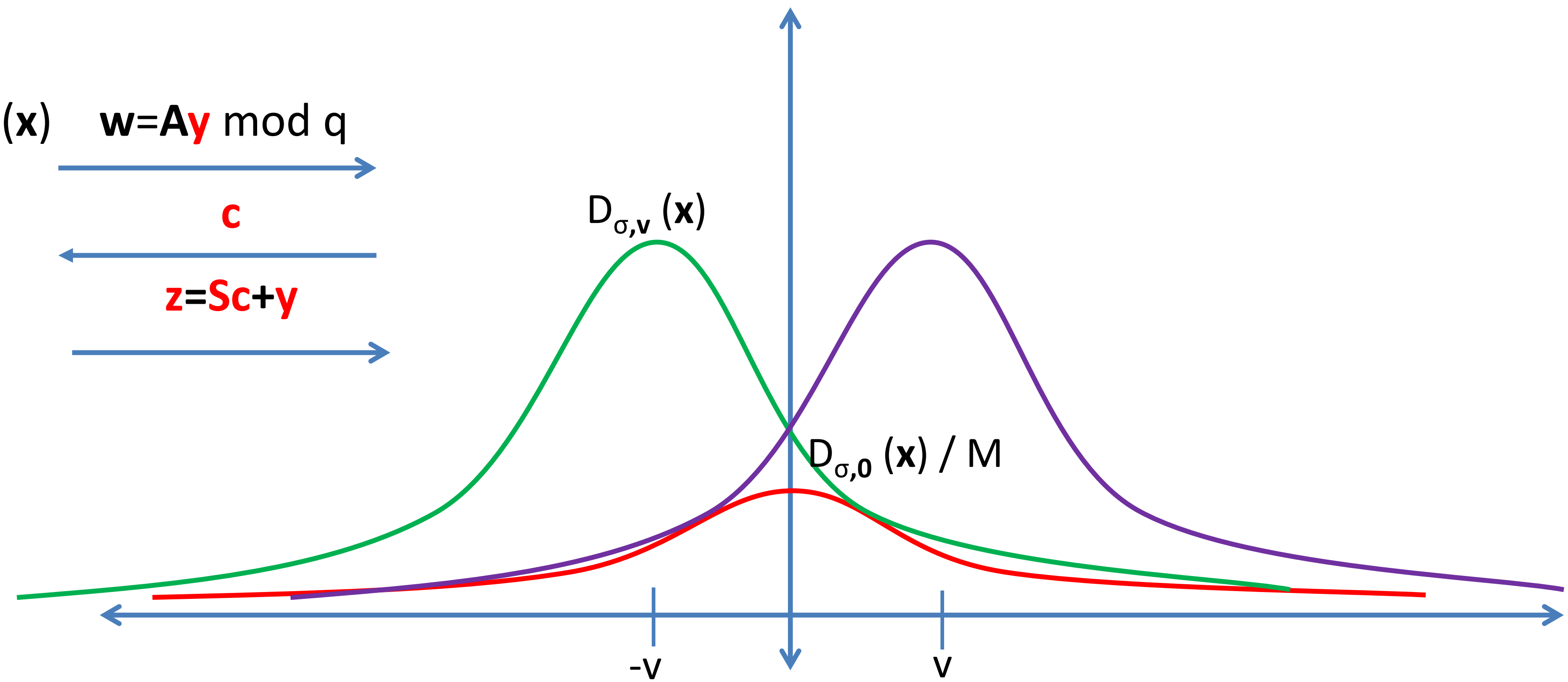
Rejection Sampling

Secret Key: **S**

$\mathbf{y} \sim D_{\sigma,\mathbf{0}}(\mathbf{x})$ $\mathbf{w}=\mathbf{A}\mathbf{y} \bmod q$

c

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



$v=\max ||\mathbf{Sc}||$

$M < e^{12/k}$

New Rejection Sampling

$g(\mathbf{x})=f(\mathbf{x}) = D_{\sigma,\mathbf{0}}(\mathbf{x})$

Lemma: If $\sigma = k||v||$, then with very high probability, for all likely $\mathbf{x} \sim f$,

$$D_{\sigma,\mathbf{0}}(\mathbf{x}) / D_{\sigma,\mathbf{v}}(\mathbf{x}) < e^{12/k}$$

Set $k=12$ (asymptotically $\sqrt{\log m}$) $\rightarrow$ $M < e$

New Rejection Sampling

$g(\mathbf{x})=f(\mathbf{x}) = D_{\sigma,\mathbf{0}}(\mathbf{x})$

Lemma: If $\sigma = k||v||$, then with very high probability, for all likely $\mathbf{x} \sim f$,

$$D_{\sigma,\mathbf{0}}(\mathbf{x}) / D_{\sigma,\mathbf{v}}(\mathbf{x}) < e^{12/k}$$

Set k=12 (asymptotically $\sqrt{\log m}$) $\rightarrow$ $M < e$

$$||x|| \approx \sqrt{m}\,||v|| \approx O(m)$$

Identification Scheme Based on SIS

Secret Key: $\mathbf{S}$

Public Key: $\mathbf{A}$, $\mathbf{T}=\mathbf{AS} \bmod q$

Pick $\mathbf{y} \sim D_{\sigma,\mathbf{0}}$

$\xrightarrow{\mathbf{w}=\mathbf{Ay} \bmod q}$

pick a random $\mathbf{c}$

$\xleftarrow{\mathbf{c}}$

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

$\mathbf{z} = \square$ with probability

$1 - D_{\sigma,\mathbf{0}}(\mathbf{z}) / (M D_{\sigma,\mathbf{Sc}}(\mathbf{z}))$

$\xrightarrow{\mathbf{z}}$

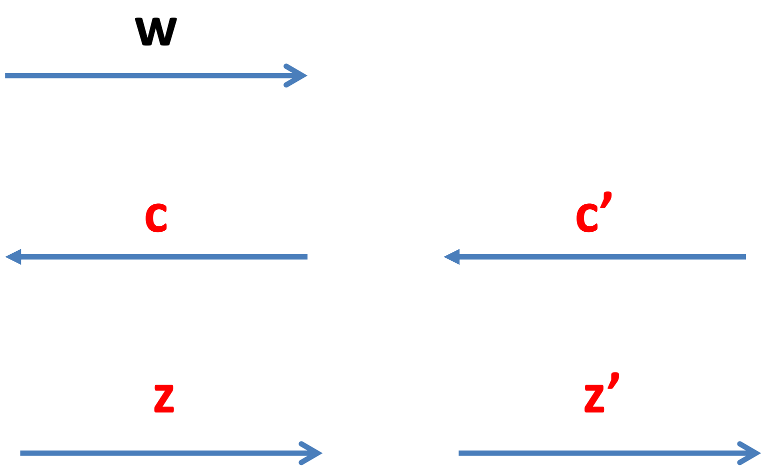
If $\mathbf{AS}=\mathbf{AS'} \bmod q$, then $(\mathbf{c},\mathbf{z})$ has the same distribution whether $\mathbf{S}$ or $\mathbf{S'}$ is used

$(\mathbf{w},\mathbf{c},\mathbf{z})$ as well ...

Security Reduction (Stage 2)

Adversary

Public Key: **A**, **T**=**AS** mod q



$\mathbf{A}\mathbf{z}=\mathbf{T}\mathbf{c}+\mathbf{w} \bmod q$

$\mathbf{A}\mathbf{z}'=\mathbf{T}\mathbf{c}'+\mathbf{w} \bmod q$

$\mathbf{A}(\mathbf{z}-\mathbf{z}')=\mathbf{T}(\mathbf{c}-\mathbf{c}') \bmod q$

$\mathbf{A}(\mathbf{z}-\mathbf{z}')=\mathbf{A}\mathbf{S}(\mathbf{c}-\mathbf{c}') \bmod q$

Observation: If the adversary knows **S**, then he can always give us $\mathbf{z}-\mathbf{z}' \neq \mathbf{S}(\mathbf{c}-\mathbf{c}')$

Hope: $\mathbf{z}-\mathbf{z}' \neq \mathbf{S}(\mathbf{c}-\mathbf{c}')$

Solution: Make sure adversary does not learn **S**

With probability at least ½, we solve SIS.

Signature Scheme

Secret Key: $\mathbf{S}$

Public Key: $\mathbf{A}$, $\mathbf{T}=\mathbf{AS}$ mod q

Sign(μ)

Pick $\mathbf{y} \sim D_{\sigma,\mathbf{0}}$

Compute $\mathbf{c}=H(\mathbf{Ay} \text{ mod } q,\mu)$

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

Output($\mathbf{z}$,$\mathbf{c}$) with probability

$$D_{\sigma,\mathbf{0}}(\mathbf{z}) / (M D_{\sigma,\mathbf{Sc}}(\mathbf{z}))$$

(If nothing was output, repeat)

Verify($\mathbf{z}$,$\mathbf{c}$,μ)

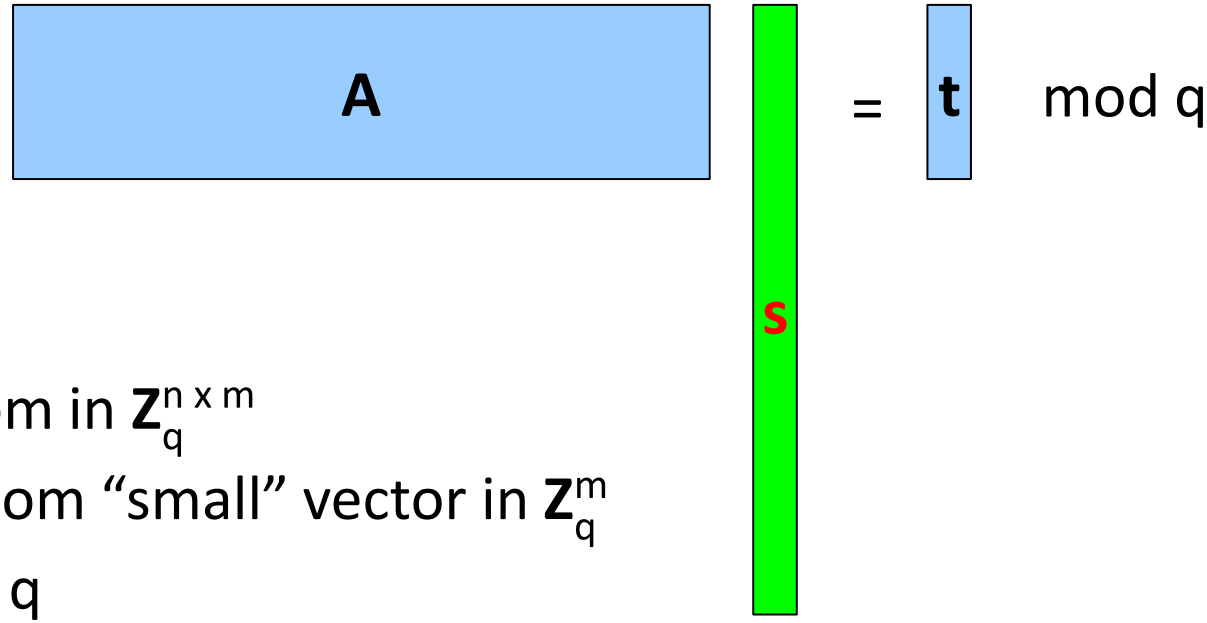
Check that $\mathbf{z}$ is "small"

and

$\mathbf{c} = H(\mathbf{Az} - \mathbf{Tc} \text{ mod } q, \mu)$

PRACTICAL CONSIDERATIONS

The Knapsack Problem



A is random in $\mathbf{Z}_q^{n \times m}$

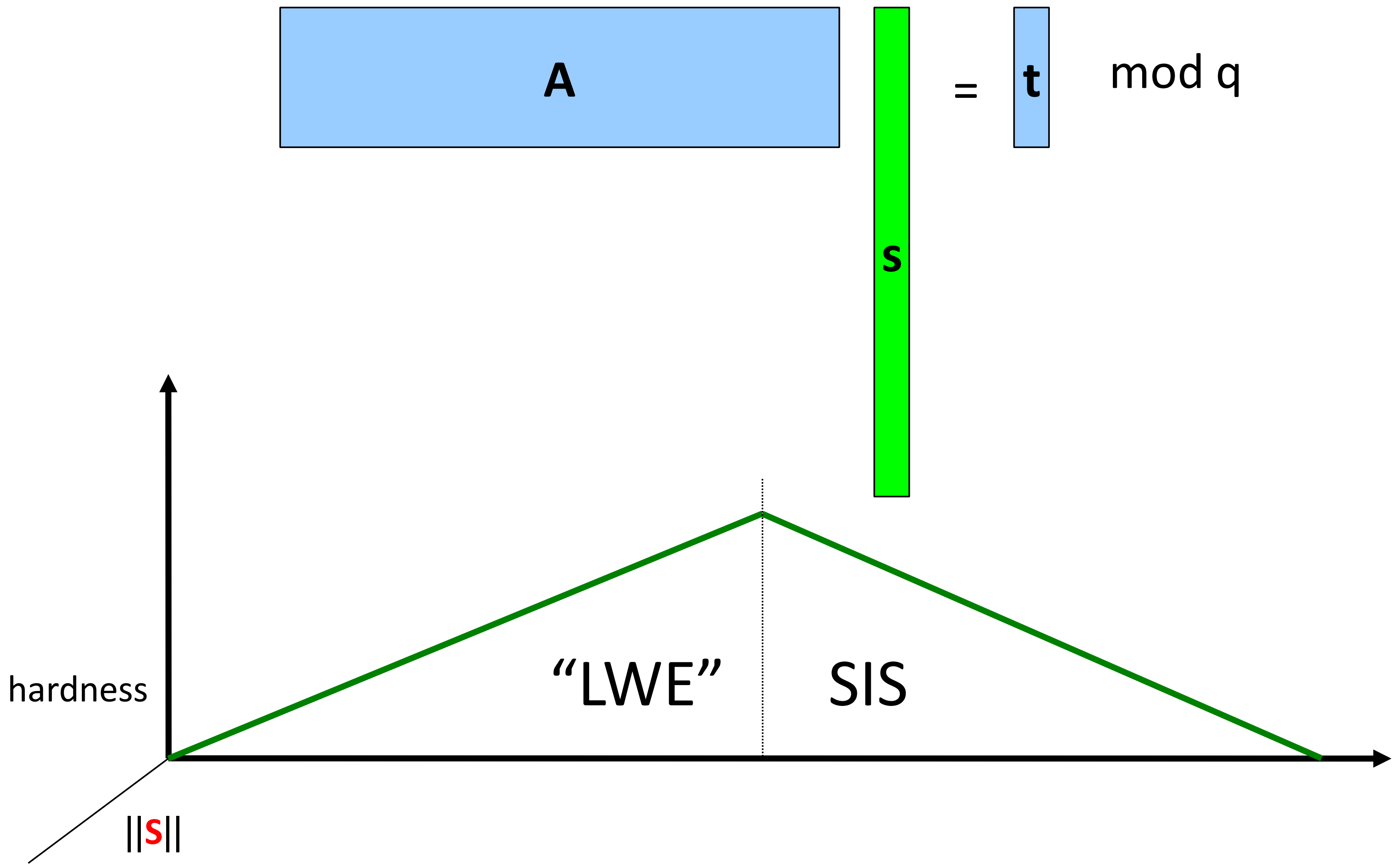
s is a random "small" vector in $\mathbf{Z}_q^{m}$

t=**A**s mod q

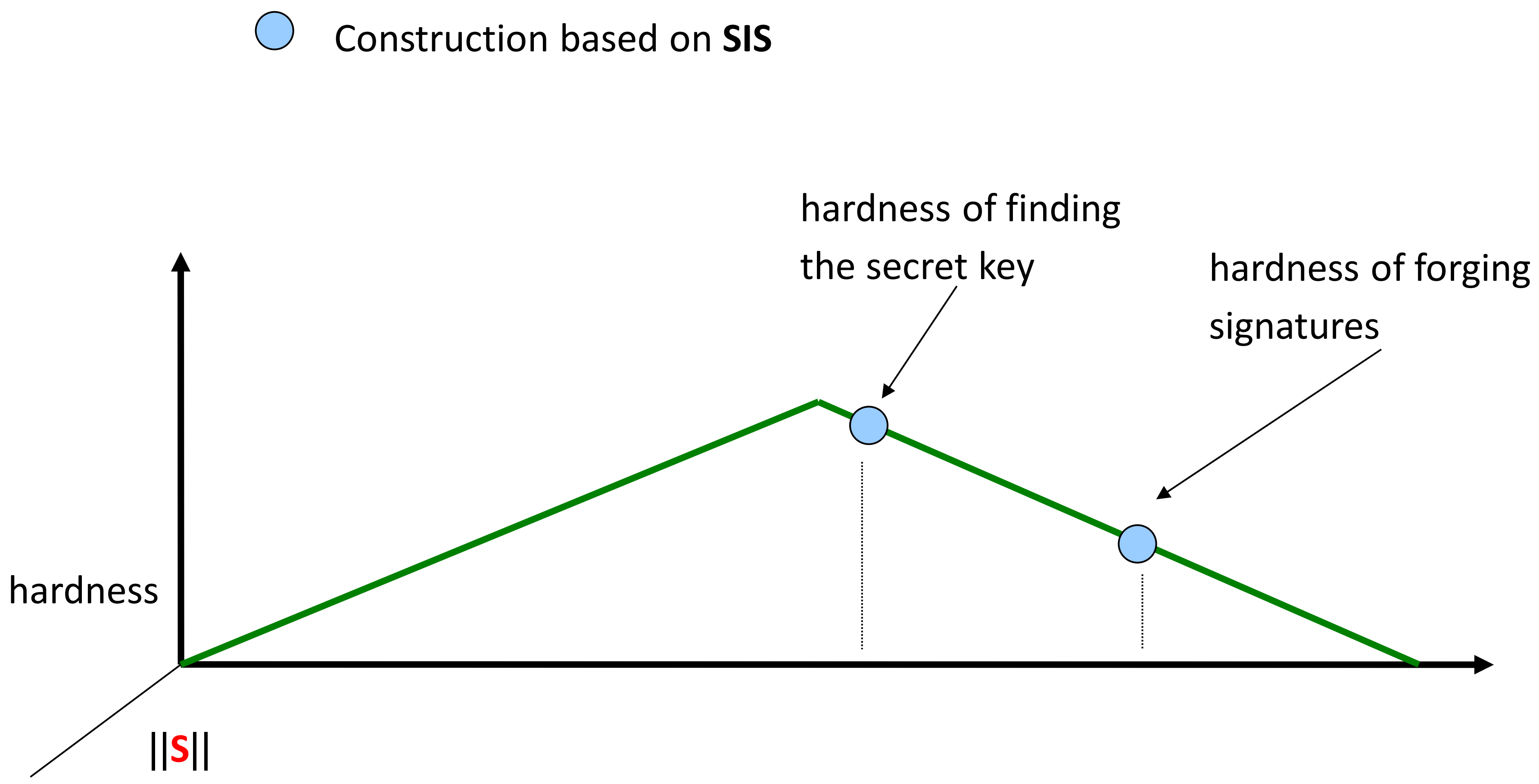
Given (**A**,**t**), find small **s'** such that

As'=**t** mod q

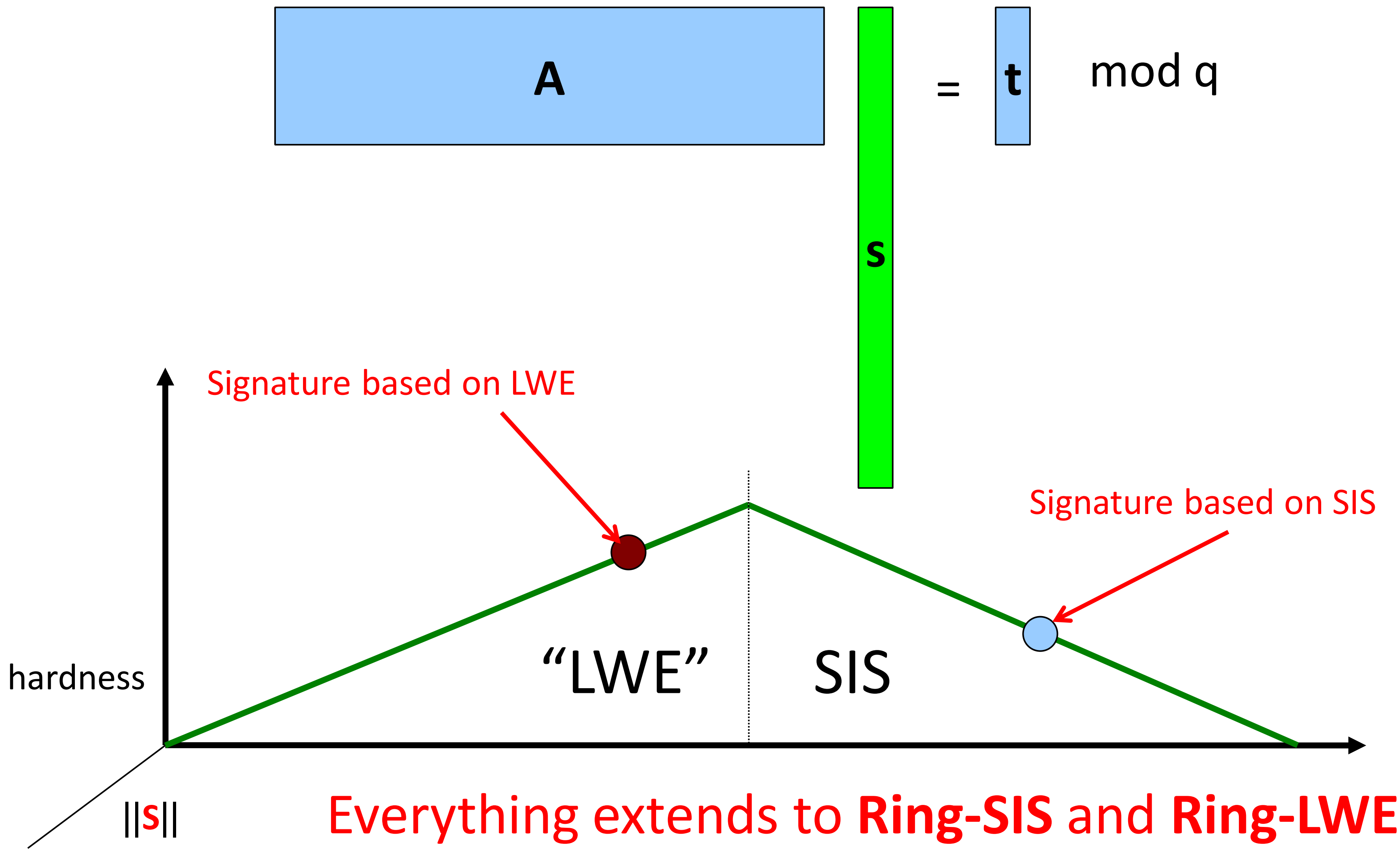
Hardness of the Knapsack Problem



Signature Hardness



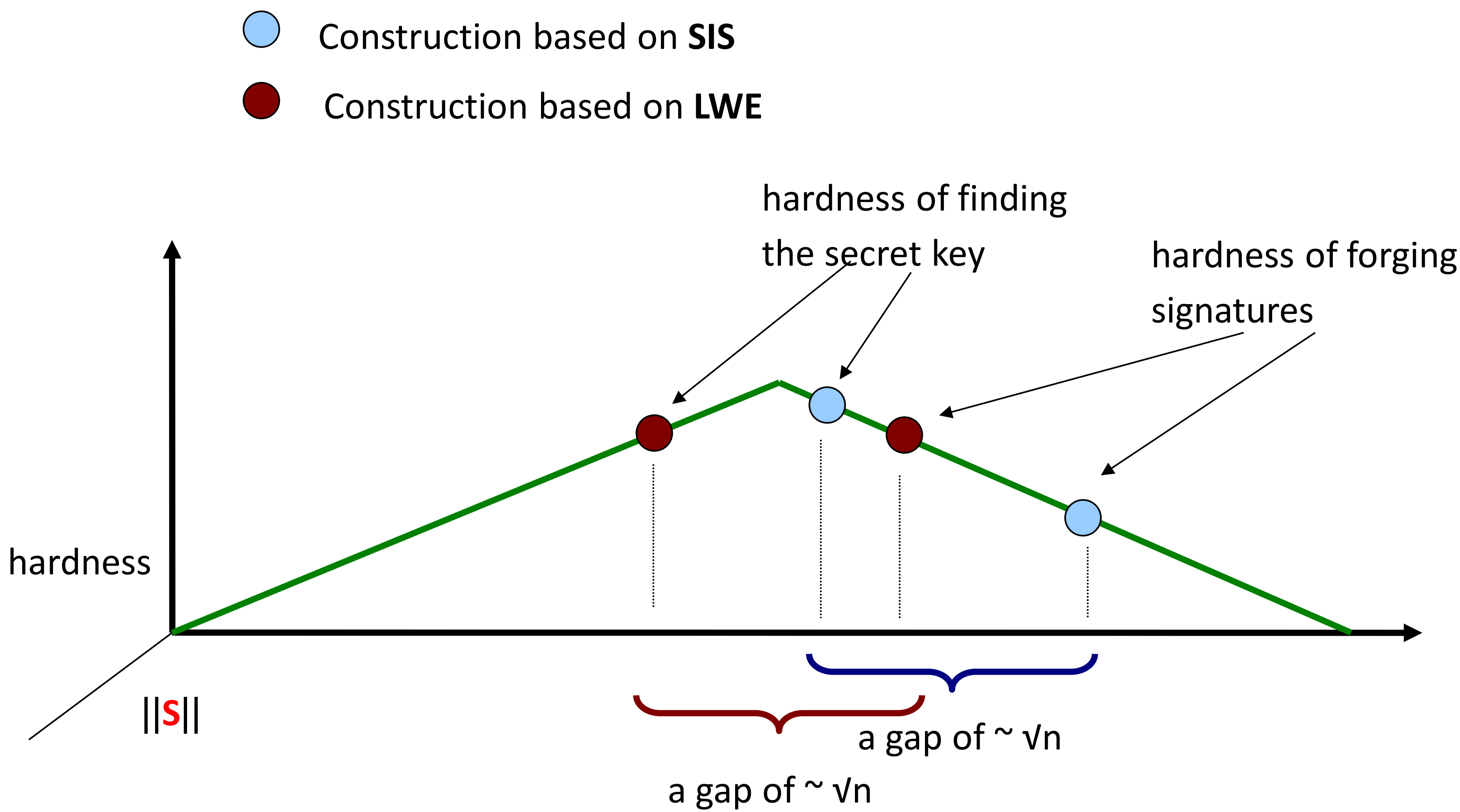
Lattice Signatures



Everything extends to **Ring-SIS** and **Ring-LWE**

IDENTIFICATION AND “FIAT-SHAMIR” SIGNATURE SCHEMES BASED ON LWE

Signature Based on LWE



Identification Scheme Based on SIS

Adversary

Secret Key: **S**

Public Key: **A**, **T**=**AS** mod q

Pick **y** ~ $D_{\sigma,0}$

w=**Ay** mod q

pick a random **c**

c

z=**Sc**+**y**

z = □ with probability

1 - $D_{\sigma,0}$ (**z**) / (M$D_{\sigma,Sc}$ (**z**))

z

If **AS**=**AS'** mod q, then (**c**,**z**) has the same distribution whether **S** or **S'** is used

(**w**,**c**,**z**) as well ...

There is only one **S**, so reduction does not work in the second step!!

Passive Adversary

A,T (Decide whether **T** is LWE or Random)

<u>Adversary</u>

Public Key: **A, T**

$\mathbf{w}=\mathbf{A}z-\mathbf{T}c \bmod q$ →

pick a random **c**

← **c**

z ~ $D_{\sigma,\mathbf{0}}$

z →

Important: cannot do simulation if z = □, (how do you generate **w**??)
But this is not needed because z = □ never appears in the signature scheme.

Security Reduction (Stage 2)

<u>Adversary</u>

Public Key: **A**, **T** mod q

$\xrightarrow{\mathbf{w}}$

$\xleftarrow{\mathbf{c}}$ $\xleftarrow{\mathbf{c'}}$

$\xrightarrow{\mathbf{z}}$ $\xrightarrow{\mathbf{z'}}$

$\mathbf{Az}=\mathbf{Tc}+\mathbf{w} \bmod q$

$\mathbf{Az'}=\mathbf{Tc'}+\mathbf{w} \bmod q$

$\mathbf{A}(\mathbf{z}-\mathbf{z'})=\mathbf{T}(\mathbf{c}-\mathbf{c'}) \bmod q$

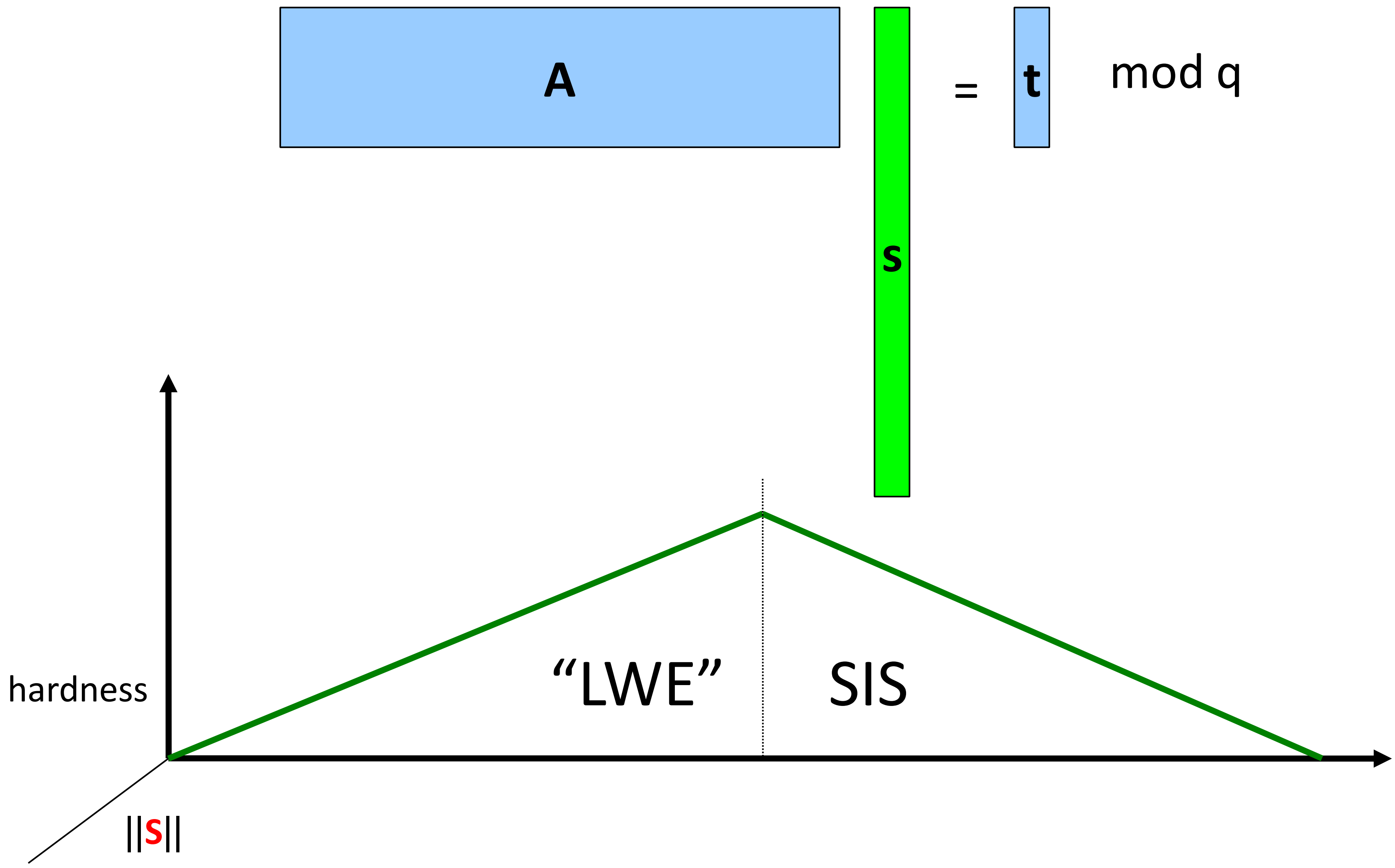
If there is an **S** such that **AS**=**T**, then the Adversary must succeed.

If **T** is random:

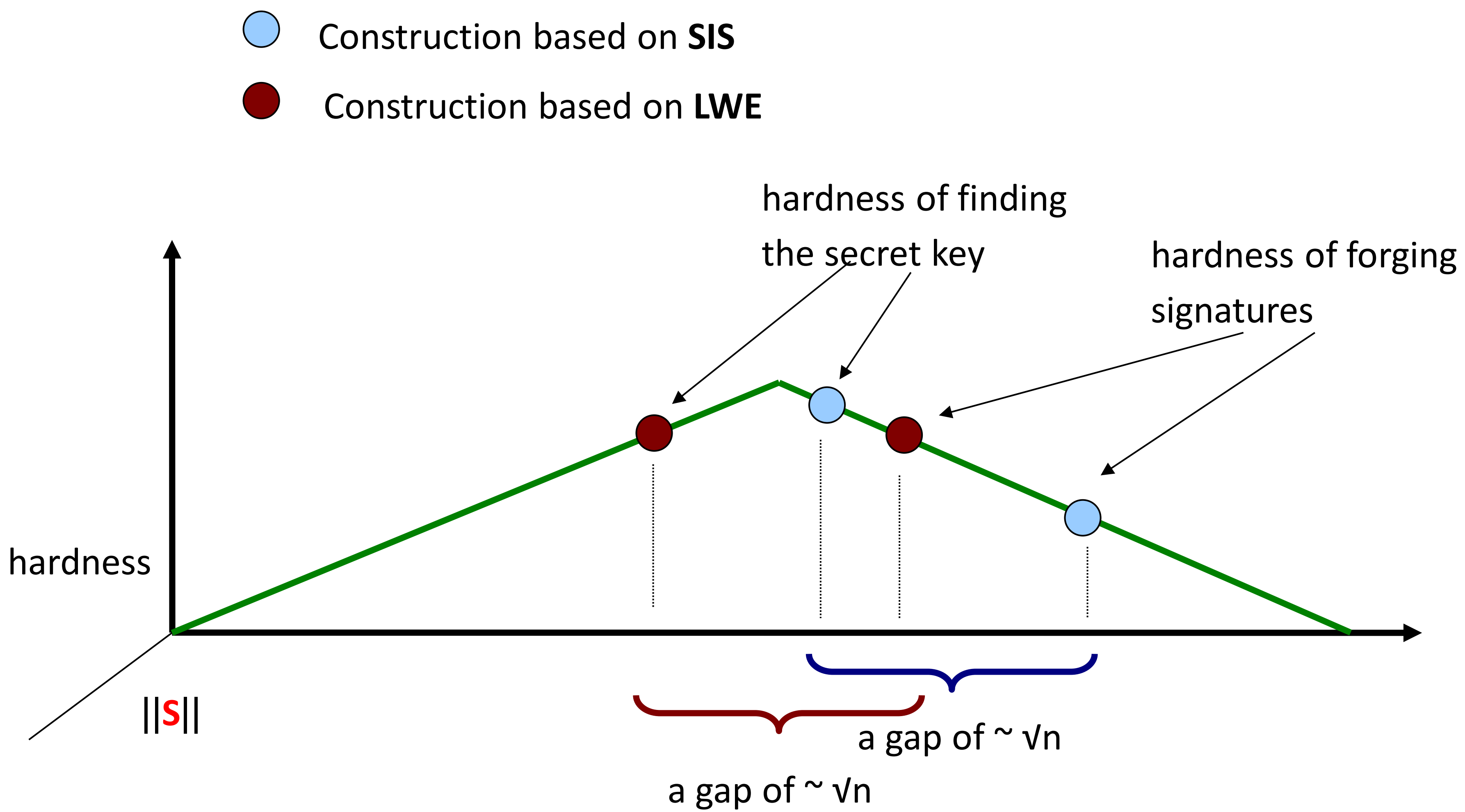
If adversary does not succeed, then we can solve LWE

If adversary still succeeds, then we solve SIS for (**A**|**T**) → can solve LWE

Hardness of the Knapsack Problem



Signature Hardness



Parameters (Using Rings)

			[GLP '12]
sk size (bits)	12,000	2000	2000
pk size (bits)	12,000	12,000	12,000
sig size (bits)	140,000	17,000	9000

≈ 80-100 bit security level [GN '08, CN '11]