

Lattice Signatures Without Trapdoors

Vadim Lyubashevsky

INRIA / ENS, Paris

Signature Schemes

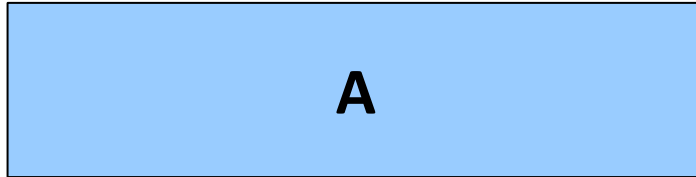
- Hash-and-Sign
 - Requires a trap-door function
- Fiat-Shamir transformation
 - Conversion from an identification scheme
 - No trap-door function needed

Lattice Signature Schemes

- Hash-and-Sign
 - [GPV '08] + [A '99]
 - [GPV '08] + [AP '09]
 - [P '10]
 - [MP '12]
- Fiat-Shamir
 - [L '08, '09]

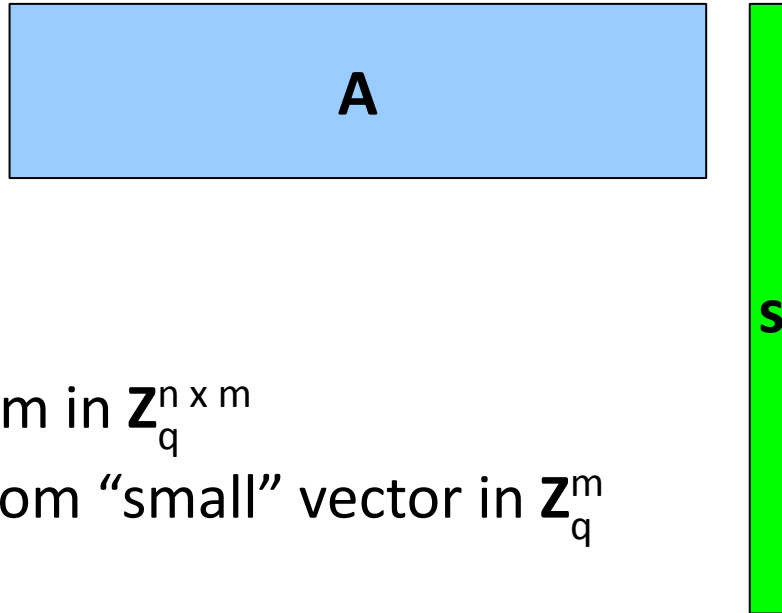
The Knapsack Problem

The Knapsack Problem



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

The Knapsack Problem



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

s is a random “small” vector in \mathbf{Z}_q^m

The Knapsack Problem

The diagram illustrates the knapsack problem equation. On the left, a light blue rectangle labeled **A** represents a matrix. To its right is a vertical green bar labeled **s**, representing a vector. An equals sign follows, then a light blue vertical bar labeled **t**, representing a vector, and finally the text "mod q".

$$\mathbf{A} \mathbf{s} = \mathbf{t} \pmod{q}$$

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

s is a random “small” vector in \mathbf{Z}_q^m

t = **As** mod q

The Knapsack Problem

The diagram illustrates the equation $As \equiv t \pmod{q}$. On the left, a light blue rectangle labeled 'A' represents the matrix. To its right is a tall, thin green vertical rectangle labeled 's', representing the vector. An equals sign follows, then a light blue vertical rectangle labeled 't', and finally 'mod q'.

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

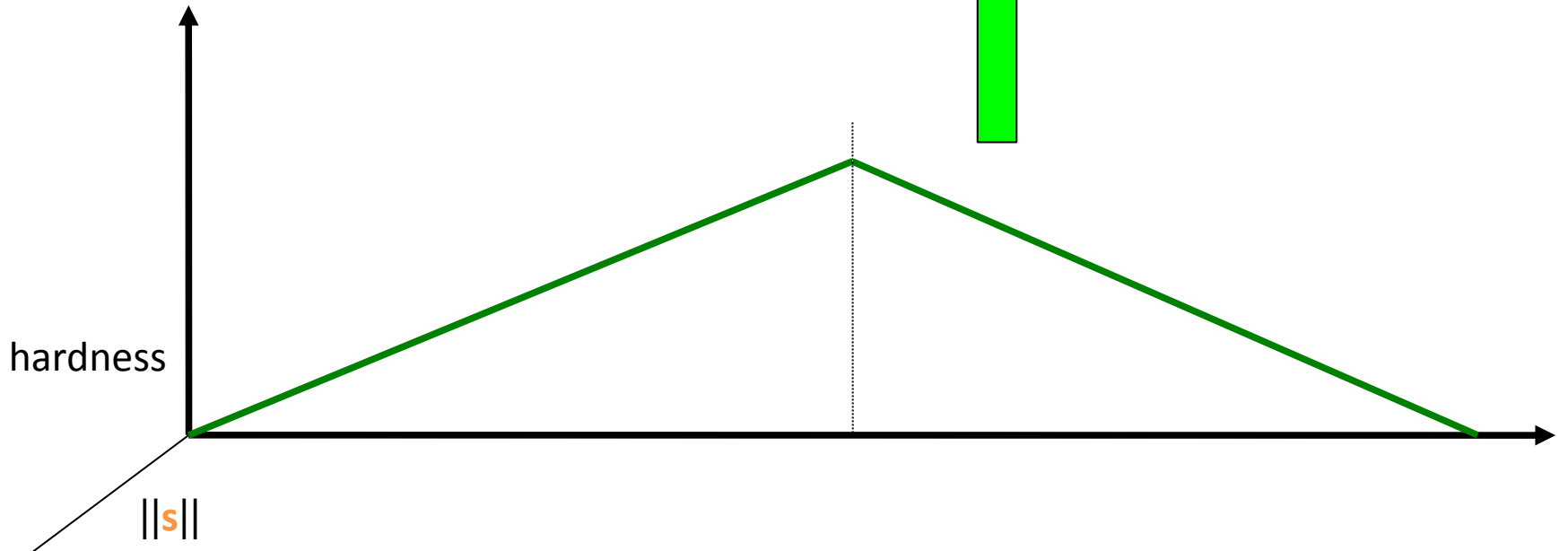
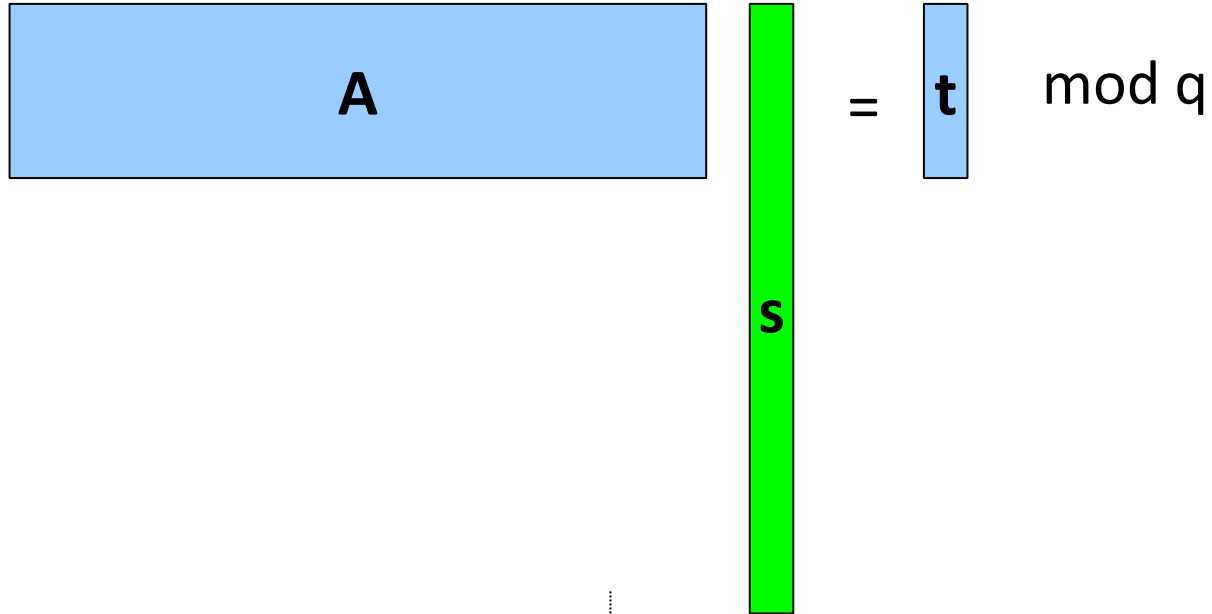
s is a random “small” vector in \mathbf{Z}_q^m

t = **As** mod q

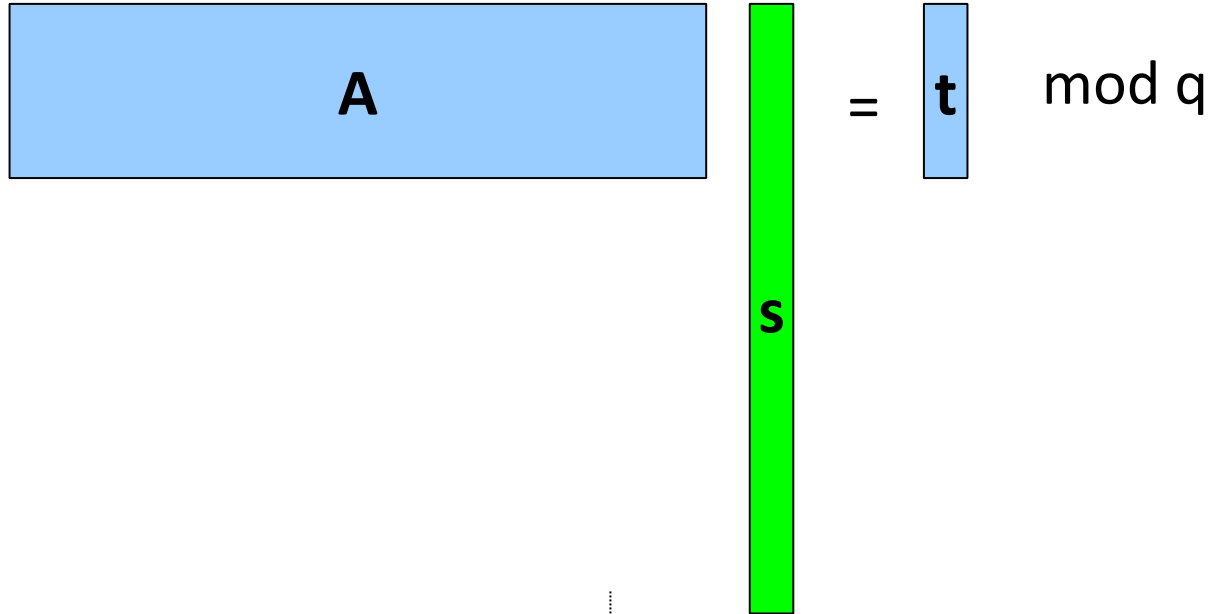
Given **(A,t)**, find small **s'** such that

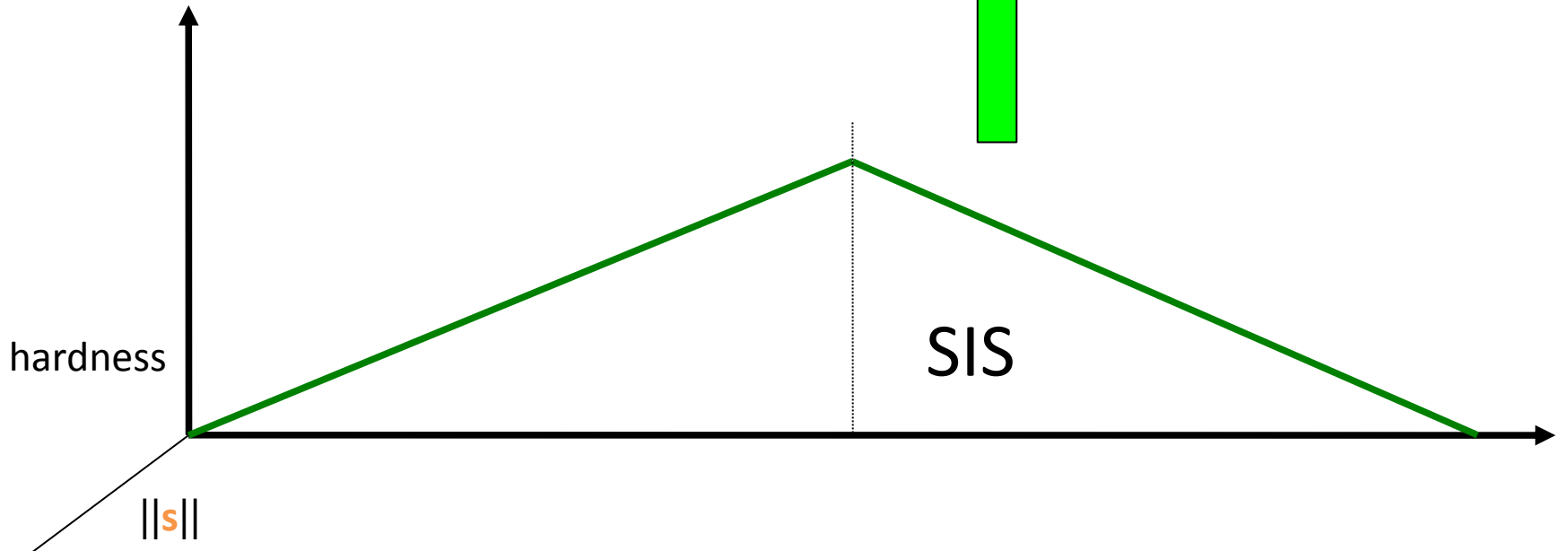
As' = **t** mod q

Hardness of the Knapsack Problem



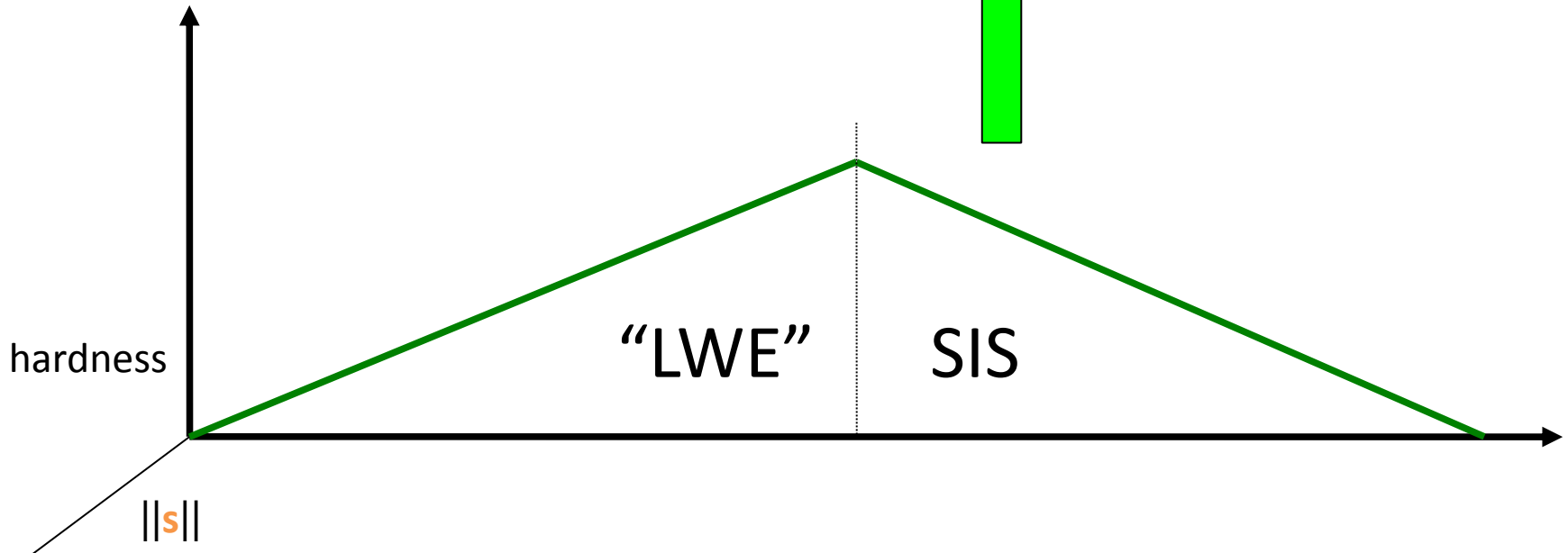
Hardness of the Knapsack Problem

$$A \cdot s = t \pmod{q}$$




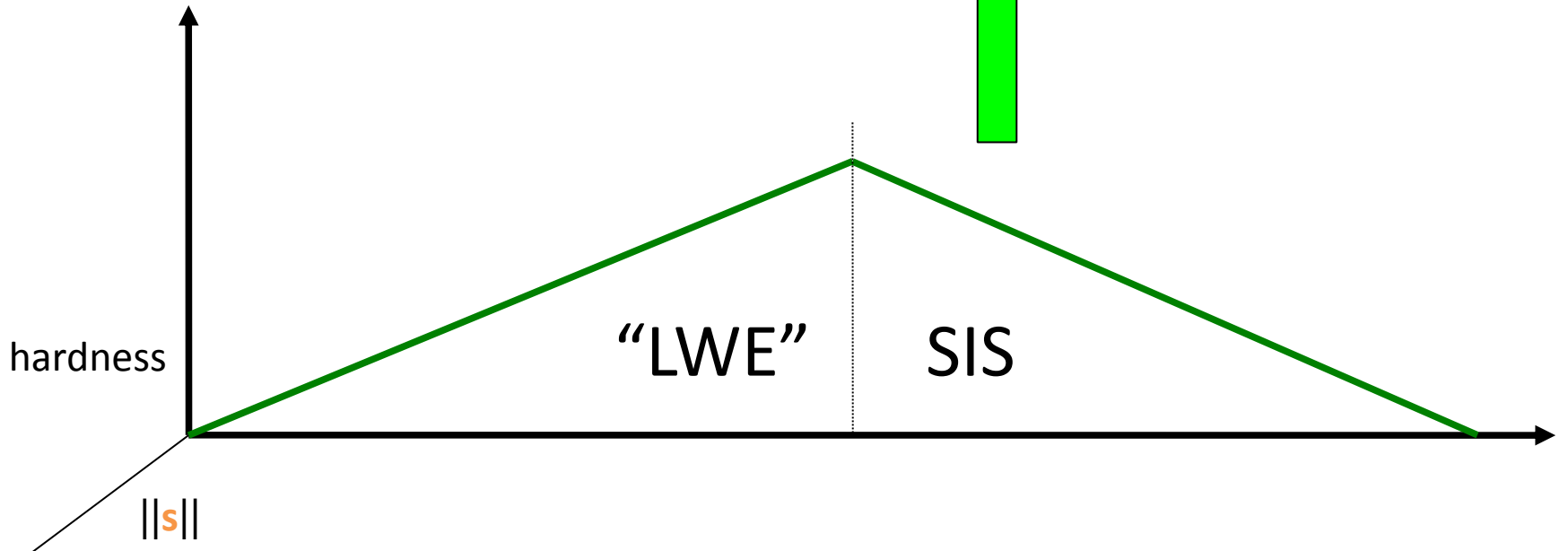
Hardness of the Knapsack Problem

$$A \cdot s = t \pmod{q}$$



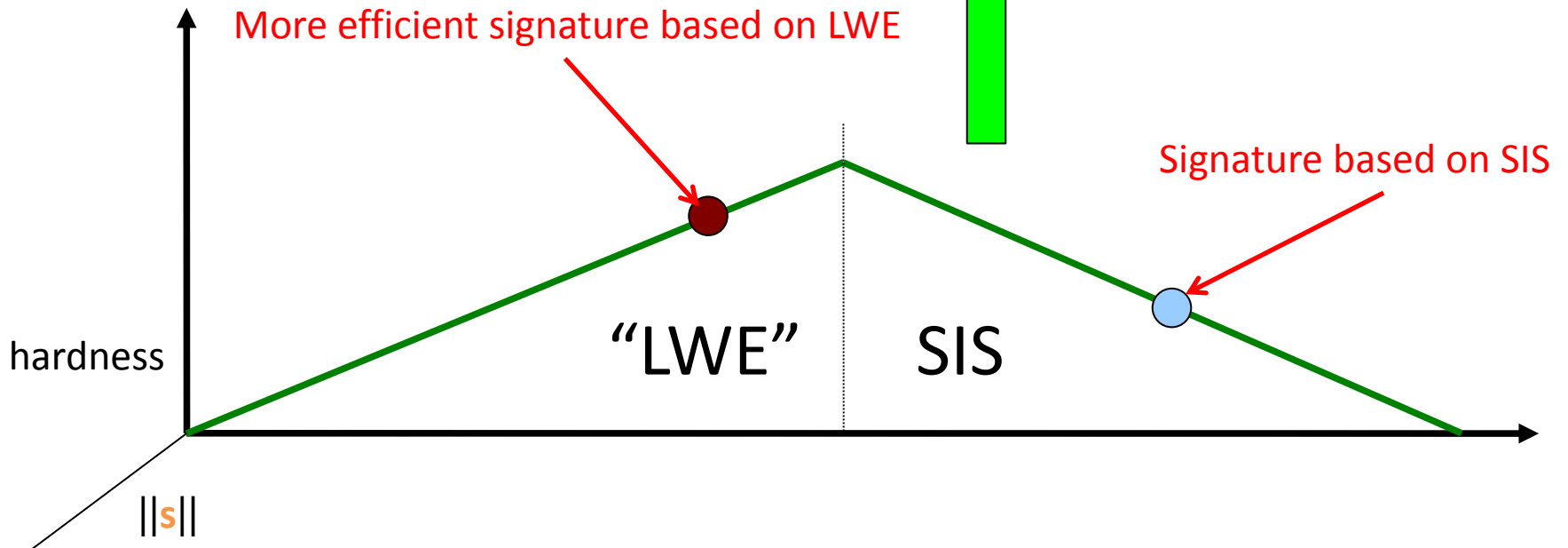
Our Results

$$A \begin{matrix} \color{red} s \end{matrix} = \begin{matrix} \color{blue} t \end{matrix} \pmod{q}$$



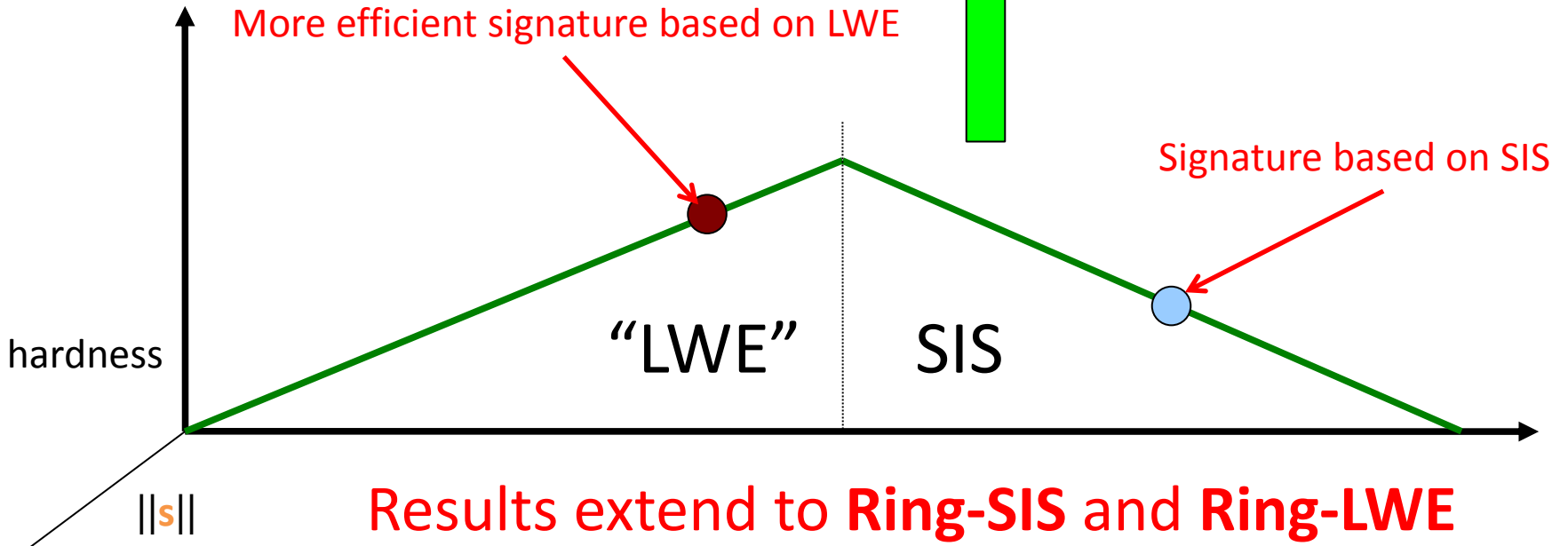
Our Results

$$\begin{matrix} \boxed{A} & \begin{matrix} \color{green} \boxed{s} \\ \color{green} s \end{matrix} & = & \boxed{t} \pmod{q} \end{matrix}$$



Our Results

$$\begin{matrix} \boxed{A} \\ \boxed{s} \end{matrix} = \begin{matrix} \boxed{t} \\ \text{mod } q \end{matrix}$$



Results extend to **Ring-SIS** and **Ring-LWE**

Signature Based on **SIS**

Signature Scheme (Main Idea)

Signature Scheme (Main Idea)

Secret Key: **S**

Public Key: **A**, $T=AS \pmod q$

Signature Scheme (Main Idea)

Secret Key: **S**

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

Sign(μ)

Pick a random **y**

Compute **c**=H(**Ay** mod q , μ)

Signature Scheme (Main Idea)

Secret Key: S

Public Key: A , $T=AS \pmod q$

Sign(μ)

Pick a random y

Compute $c=H(Ay \pmod q, \mu)$

$z=Sc+y$

Signature Scheme (Main Idea)

Secret Key: S

Public Key: A , $T=AS \pmod q$

Sign(μ)

Pick a random y

Compute $c=H(Ay \pmod q, \mu)$

$z=Sc+y$

Output(z, c)

Signature Scheme (Main Idea)

Secret Key: S

Public Key: A , $T=AS \pmod q$

Sign(μ)

Pick a random y

Compute $c=H(Ay \pmod q, \mu)$

$z=Sc+y$

Output(z, c)

Verify(z, c)

Signature Scheme (Main Idea)

Secret Key: S

Public Key: A , $T=AS \pmod q$

Sign(μ)

Pick a random y

Compute $c=H(Ay \pmod q, \mu)$

$z=Sc+y$

Output(z, c)

Verify(z, c)

Check that z is “small”
and

$c = H(Az - Tc \pmod q, \mu)$

Security Reduction Requirements

Secret Key: S

Given the public key, the secret key is not unique

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

Sign(μ)

Pick a random y

Compute $c=H(Ay \bmod q, \mu)$

$z=Sc+y$

Output (z, c)

Signature is independent of the secret key

Verify(z, c)

Check that z is “small”
and

$c = H(Az - Tc \bmod q, \mu)$

Security Reduction

Simulator

Adversary

Security Reduction

Simulator

Adversary



Security Reduction

Simulator

Adversary



Pick random **S**

Security Reduction

Simulator

Adversary



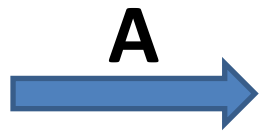
Pick random **S**



Security Reduction

Simulator

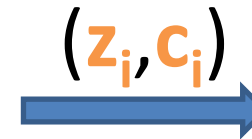
Adversary



Pick random **S**



$(z_i, c_i) = \text{Sign}(\mu_i)$



...

Security Reduction

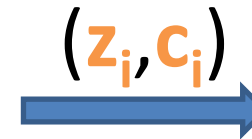
Simulator

Adversary

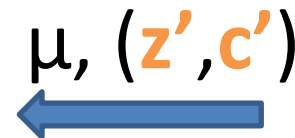
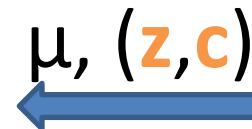


Pick random **S**

$(z_i, c_i) = \text{Sign}(\mu_i)$



...



Security Reduction

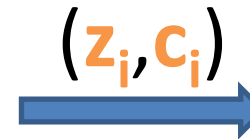
Simulator

Adversary



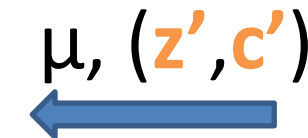
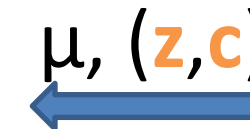
Pick random **S**

$(z_i, c_i) = \text{Sign}(\mu_i)$



...

$$A(z - z' + Sc' - Sc) = 0$$



Security Reduction

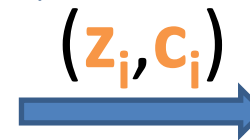
Simulator

Adversary



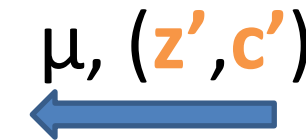
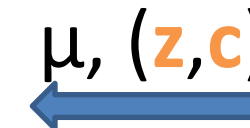
Pick random **S**

$$(z_i, c_i) = \text{Sign}(\mu_i)$$



...

$$A(z - z' + Sc' - Sc) = 0$$



If this is not 0, then **SIS** is solved.
Important for adversary to not know **S**.

Security Reduction

$$A(\underbrace{z-z'+Sc'-Sc}_{\text{Solution to SIS}})=0$$

Solution to SIS

Security Reduction

$$A(\underbrace{z-z'+Sc'-Sc}_{\text{Solution to SIS}})=0$$

Solution to SIS

We Want:

Security Reduction

$$A(\underbrace{z-z'+Sc'-Sc}_{\text{Solution to SIS}})=0$$

Solution to SIS

We Want:

1. Signature (z,c) to be independent of S so that $z-z'+Sc'-Sc$ is not 0

Security Reduction

$$A(\underbrace{z-z'+Sc'-Sc})=0$$

Solution to **SIS**

We Want:

1. Signature (z,c) to be independent of S so that $z-z'+Sc'-Sc$ is not 0
2. $z-z'+Sc'-Sc$ to be small so that **SIS** is hard

Signature Scheme

Secret Key: S

Public Key: A , $T=AS \pmod q$

Sign(μ)

Pick a random y

Compute $c=H(Ay \pmod q, \mu)$

$z=Sc+y$

Output(z, c)

Signature Scheme

Secret Key: S

Public Key: A , $T=AS \pmod q$

Sign(μ)

Pick a random y make y uniformly random mod q ?

Compute $c=H(Ay \pmod q, \mu)$

$z=Sc+y$

Output(z, c)

Signature Scheme

Secret Key: S

Public Key: A , $T=AS \pmod q$

Sign(μ)

Pick a random y make y uniformly random mod q ?

Compute $c=H(Ay \pmod q, \mu)$

$$z=Sc+y$$

Output(z, c) then z is too big and **SIS** (and forging) is easy ☹️

Signature Scheme

Secret Key: S

Public Key: A , $T=AS \pmod q$

Sign(μ)

Pick a random y make y small?

Compute $c=H(Ay \pmod q, \mu)$

$z=Sc+y$

Output(z, c)

Signature Scheme

Secret Key: S

Public Key: A , $T=AS \pmod q$

Sign(μ)

Pick a random y make y small?

Compute $c=H(Ay \pmod q, \mu)$

$z=Sc+y$

Output(z, c) then z will not be independent of S ☹️

Rejection Sampling

Secret Key: S

Public Key: A , $T=AS \pmod q$

Sign(μ)

Pick a random y

Compute $c=H(Ay \pmod q, \mu)$

$z=Sc+y$

Rejection Sampling

Secret Key: **S**

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

Sign(μ)

Pick a random **y** make **y** small

Compute **c**=H(**Ay** mod q , μ)

z=Sc+y

Rejection Sampling

Secret Key: **S**

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

Sign(μ)

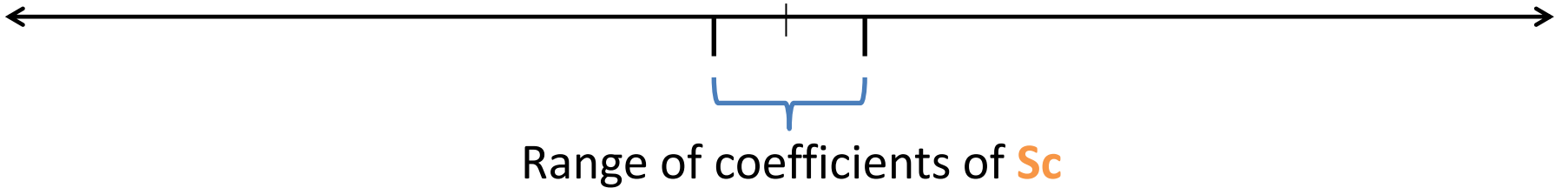
Pick a random **y** *make y small*

Compute **c**=H(**Ay** mod q , μ)

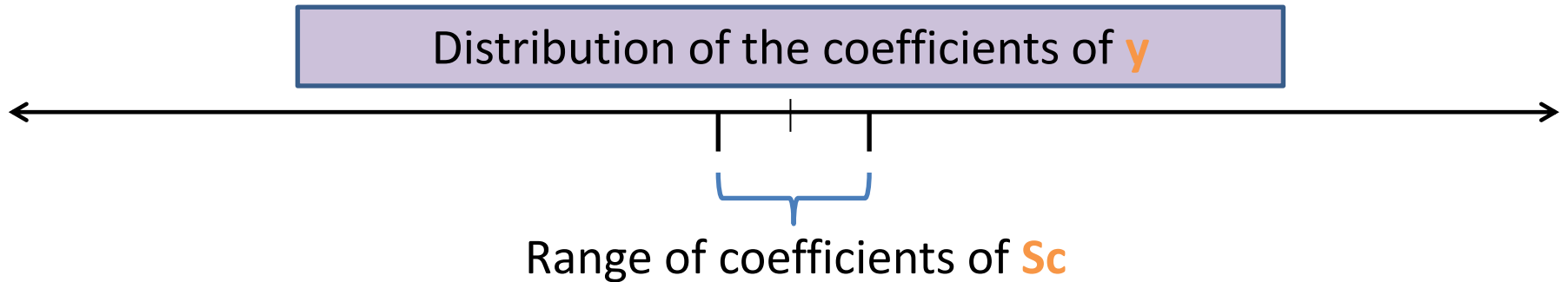
z=Sc+y

Output(**z,c**) *if z meets certain criteria, else repeat*

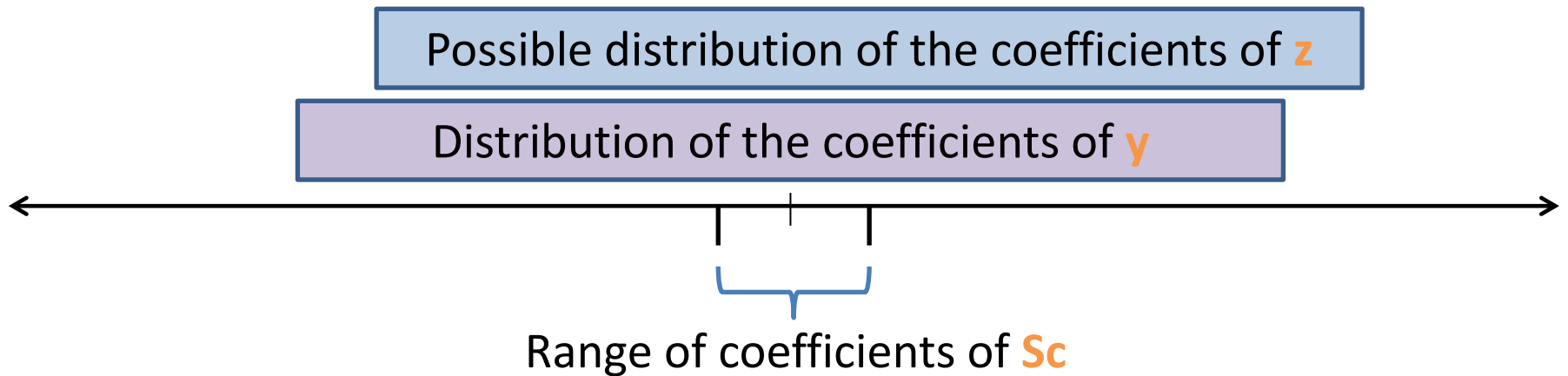
Rejection Sampling (L '09)



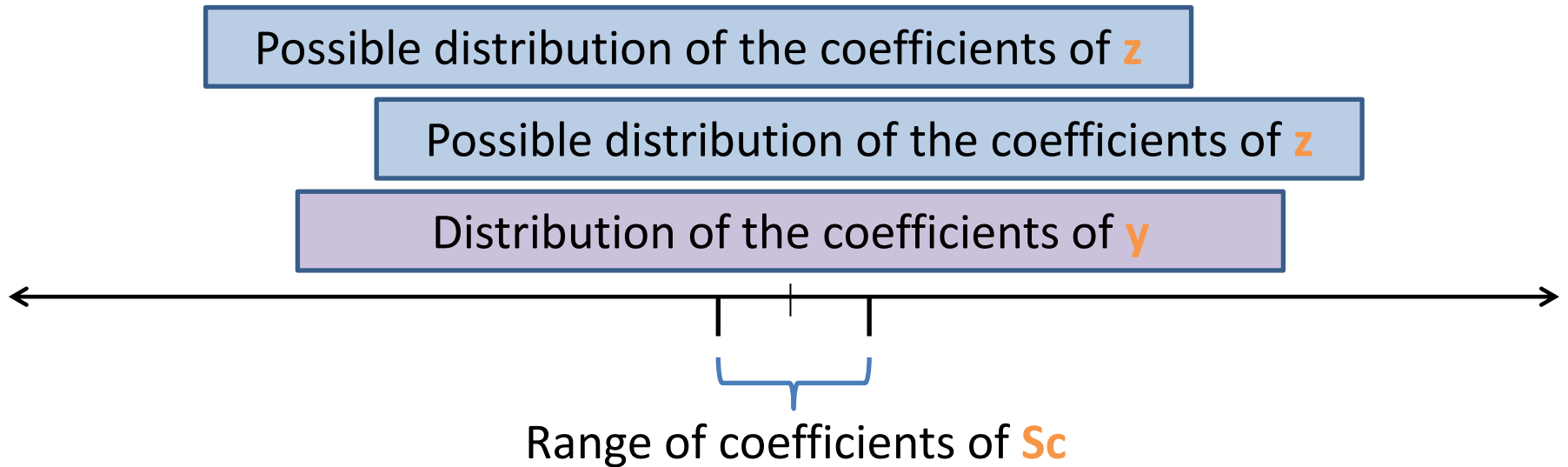
Rejection Sampling (L '09)



Rejection Sampling (L '09)



Rejection Sampling (L '09)



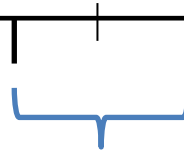
Rejection Sampling (L '09)

Target distribution of the coefficients of z

Possible distribution of the coefficients of z

Possible distribution of the coefficients of z

Distribution of the coefficients of y



Range of coefficients of Sc

Rejection Sampling (L '09)

Target distribution of the coefficients of z

Possible distribution of the coefficients of z

Possible distribution of the coefficients of z

Distribution of the coefficients of y



Range of coefficients of Sc

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

Want $p^m \approx \text{constant}$

Rejection Sampling (L '09)

Target distribution of the coefficients of z

Possible distribution of the coefficients of z

Possible distribution of the coefficients of z

Distribution of the coefficients of y



Range of coefficients of Sc

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

Want $p^m \approx \text{constant}$

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** = $O(1)$

Size of the **SIS** solution

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

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

Size of the **SIS** solution

Coefficients of **S** $\mathbf{c} = O(1)$

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

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

Size of the **SIS** solution

Coefficients of **S** $\mathbf{c} = O(1)$

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

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

Can we do better??

Size of the **SIS** solution

Coefficients of **S****c** = $O(1)$

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

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

Can we do better??

This work: Can get $\|\mathbf{z}\| = O(m)$

Different Rejection Sampling

- Previous rejection sampling constructed a **uniform** distribution in a box (or a ball)
- New rejection sampling constructs a discrete **Normal** distribution

m-dimensional Normal distribution:

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

m-dimensional *discrete* normal distribution

$$D_{\sigma, \mathbf{v}}^m(\mathbf{x}) = \rho_{\sigma, \mathbf{v}}^m(\mathbf{x}) / \rho_{\sigma, \mathbf{0}}^m(\mathbf{Z}^m)$$

Different Rejection Sampling

Secret Key: S

Public Key: A , $T=AS \pmod q$

Sign(μ)

Pick a random y

Compute $c=H(Ay \pmod q, \mu)$

$z=Sc+y$

Output(z, c)

Different Rejection Sampling

Secret Key: \mathbf{S}

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

Sign(μ)

Pick a random $\mathbf{y} \sim D_{\sigma,0}^m$

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

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

Output(\mathbf{z}, \mathbf{c})

Different Rejection Sampling

Secret Key: **S**

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

Sign(μ)

Pick a random **y** $\sim D_{\sigma,0}^m$

Compute **c**=H(**Ay** mod q , μ)

z=Sc+y (has distribution $D_{\sigma,Sc}^m(\mathbf{z})$)

Output(**z,c**)

Different Rejection Sampling

Secret Key: **S**

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

Sign(μ)

Pick a random **y** $\sim D_{\sigma,0}^m$

Compute **c**=H(**Ay** mod q , μ)

z=Sc+y (has distribution $D_{\sigma,Sc}^m(\mathbf{z})$)

Output(**z,c**) *with probability* $D_{\sigma,0}^m(\mathbf{z})/kD_{\sigma,Sc}^m(\mathbf{z})$

Different Rejection Sampling

Secret Key: **S**

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

Sign(μ)

Pick a random **y** $\sim D_{\sigma,0}^m$

Compute **c**=H(**Ay** mod q , μ)

z=Sc+y (has distribution $D_{\sigma,Sc}^m(\mathbf{z})$)

Output(**z,c**) with probability $D_{\sigma,0}^m(\mathbf{z})/kD_{\sigma,Sc}^m(\mathbf{z})$

Pick $\sigma = O(\sqrt{m})$, $k = O(1) \rightarrow \|\mathbf{z}\| = O(m)$

Signature Based on **LWE**

Security Reduction Requirements

Secret Key: S

Given the public key, the secret key is not unique

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

Sign(μ)

Pick a random y

Compute $c = H(Ay \bmod q, \mu)$

$z = Sc + y$

Output (z, c) (or reject)

Verify(z, c)

Check that z is “small”

and

$c = H(Az - Tc \bmod q, \mu)$

Signature is independent of the secret key

Security Reduction Requirements

Secret Key: S

Given the public key, the secret key is not unique

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

Given the public key, it's computationally indistinguishable whether the secret key is unique

Sign(μ)

Pick a random y

Compute $c = H(Ay \bmod q, \mu)$

$z = Sc + y$

Output (z, c) (or reject)

Verify(z, c)

Check that z is “small”

and

$c = H(Az - Tc \bmod q, \mu)$

Signature is independent of the secret key

Security Intuition

Secret Key: **S**

Public Key: **A**, $T=AS \pmod q$

Sign(μ)

Pick a random **y**

Compute $c=H(Ay \pmod q, \mu)$

$z=Sc+y$

Output(**z**,**c**) (or reject)

Signature is independent of the secret key

Security Intuition

Secret Key: **S**

Public Key: **A**, $T=AS \pmod q$

Sign(μ)

Pick a random **y**

Compute $c=H(Ay \pmod q, \mu)$

$z=Sc+y$

Output(**z**,**c**) (or reject)

Signature is independent of the secret key

Secret Key: **S**

Public Key: **A**, $T=AS \pmod q$

Security Intuition

Secret Key: **S**

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

Sign(μ)

Pick a random **y**

Compute **c**=H(**Ay** mod q , μ)

z=Sc+y

Output(**z,c**) (or reject)

Signature is independent of the secret key

Secret Key: **S**

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

The secret key is not unique

Security Intuition

Secret Key: **S**

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

Sign(μ)

Pick a random **y**

Compute **c**=H(**Ay** mod q , μ)

z=Sc+y

Output(**z,c**) (or reject)

Signature is independent of the secret key

Secret Key: **S**

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

The secret key is not unique

Sign(μ)

Pick a random **c**

Pick a random **z**

Security Intuition

Secret Key: **S**

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

Sign(μ)

Pick a random **y**

Compute **c**=H(**Ay** mod q , μ)

z=Sc+y

Output(**z,c**) (or reject)

Signature is independent of the secret key

Secret Key: **S**

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

The secret key is not unique

Sign(μ)

Pick a random **c**

Pick a random **z**

Same Distribution

Security Intuition

Secret Key: **S**

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

Sign(μ)

Pick a random **y**

Compute **c**=H(**Ay** mod q , μ)

z=Sc+y

Output(**z,c**) (or reject)

Signature is independent of the secret key

Secret Key: **S**

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

The secret key is not unique

Sign(μ)

Pick a random **c**

Pick a random **z**

Same Distribution

With some probability

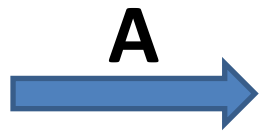
Program H(**Az-Tc**, μ)=**c**

Output(**z,c**)

Security Reduction

Simulator

Adversary

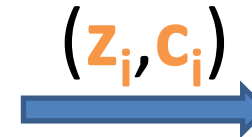


Pick random **S**



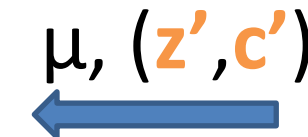
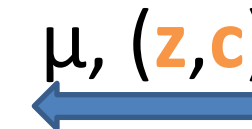
Generate (z, c)

Program $H(Az - Tc, \mu) = c$



...

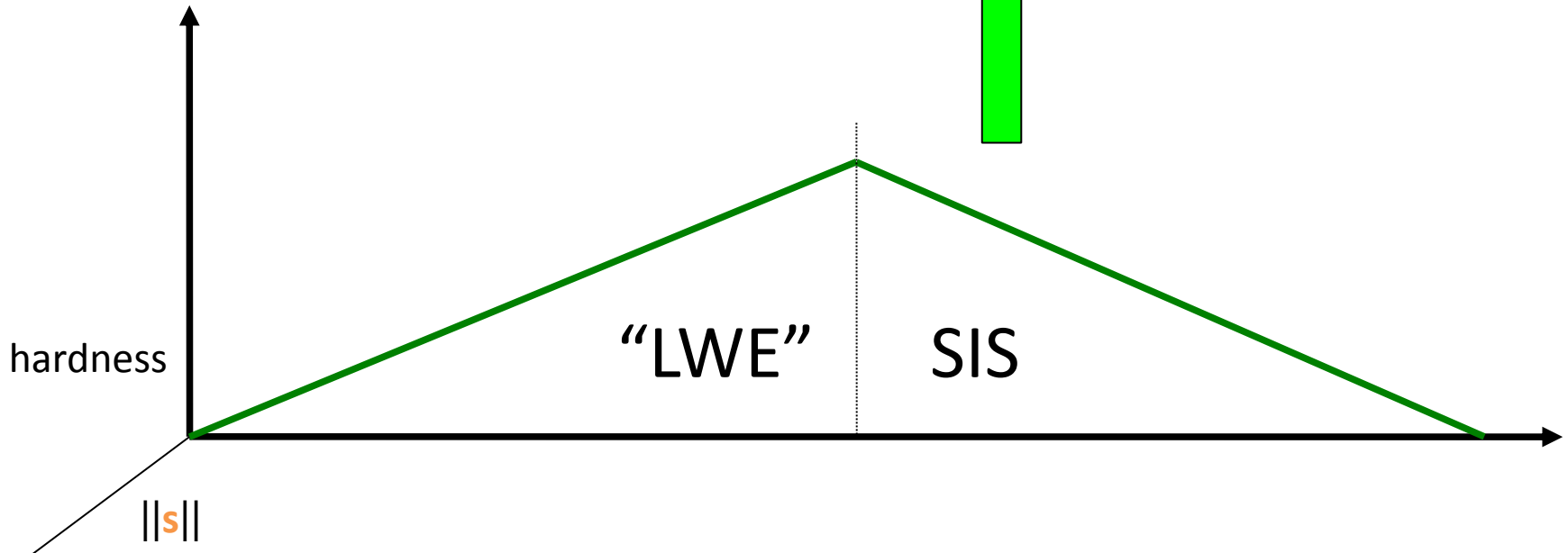
$$A(\underbrace{z - z' + Sc' - Sc}_{=0}) = 0$$



If this is not 0, then **SIS** is solved.
Important for adversary to not know **S**.

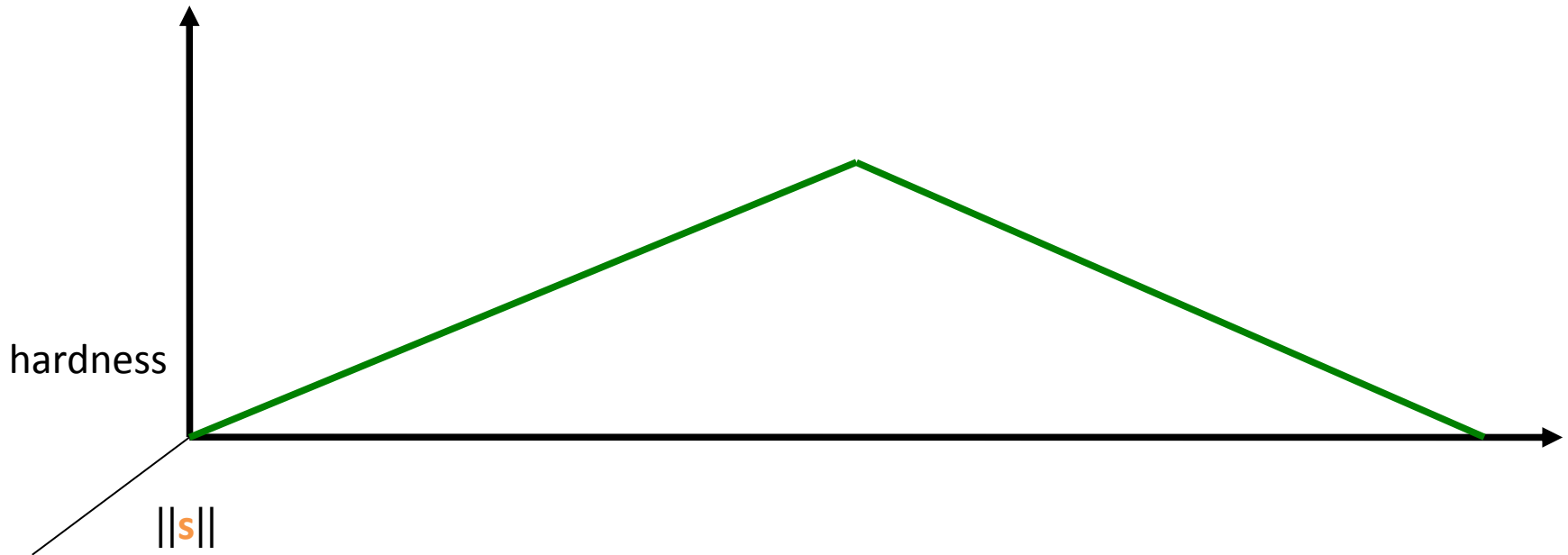
Hardness of the Knapsack Problem

$$A \cdot s = t \pmod{q}$$



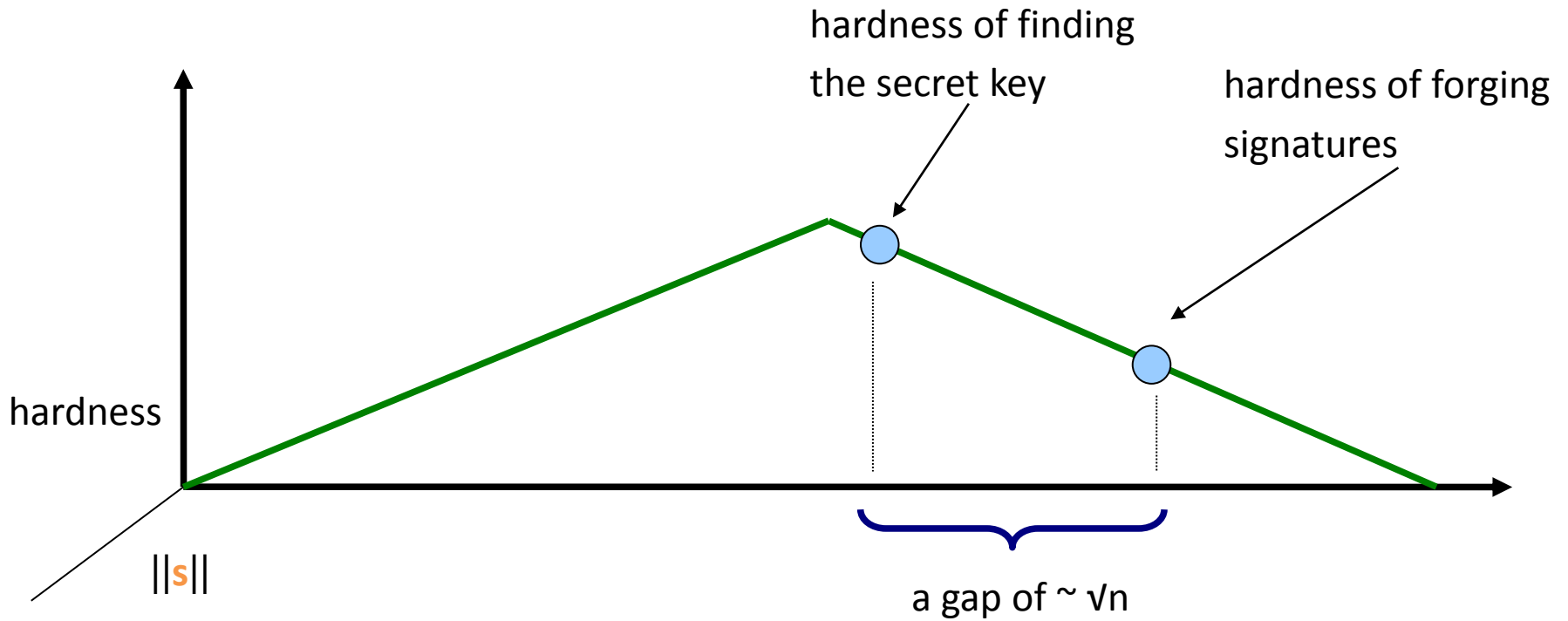
Signature Hardness

- Construction based on **SIS**
- Construction based on **LWE**



Signature Hardness

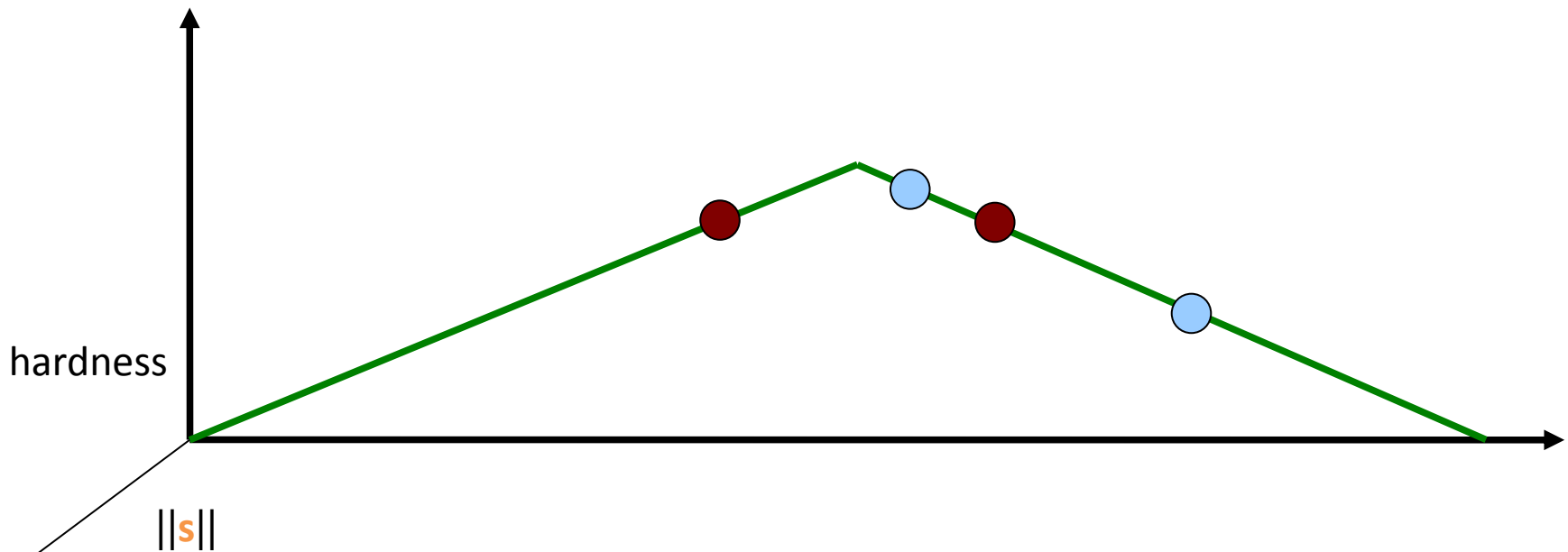
- Construction based on **SIS**
- Construction based on **LWE**



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

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



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

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

