# 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

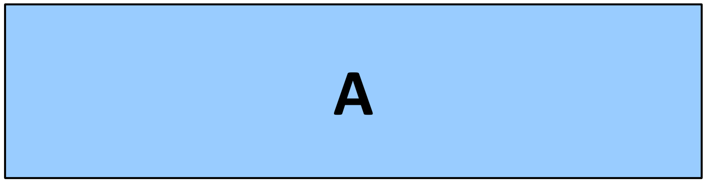
# Lattice Signature Schemes

- Hash-and-Sign
  - [GPV ‘08] + [A ’99]
  - [GPV ‘08] + [AP ‘09]
  - [P ‘10]
  - [MP ‘12]

- Fiat-Shamir
  - [L ‘08, ‘09]

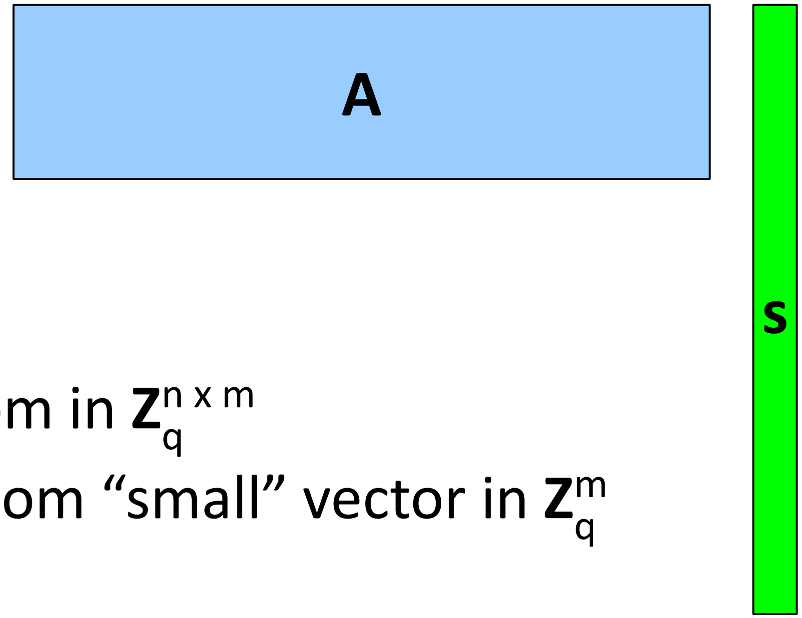
# The Knapsack Problem

# The Knapsack Problem




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

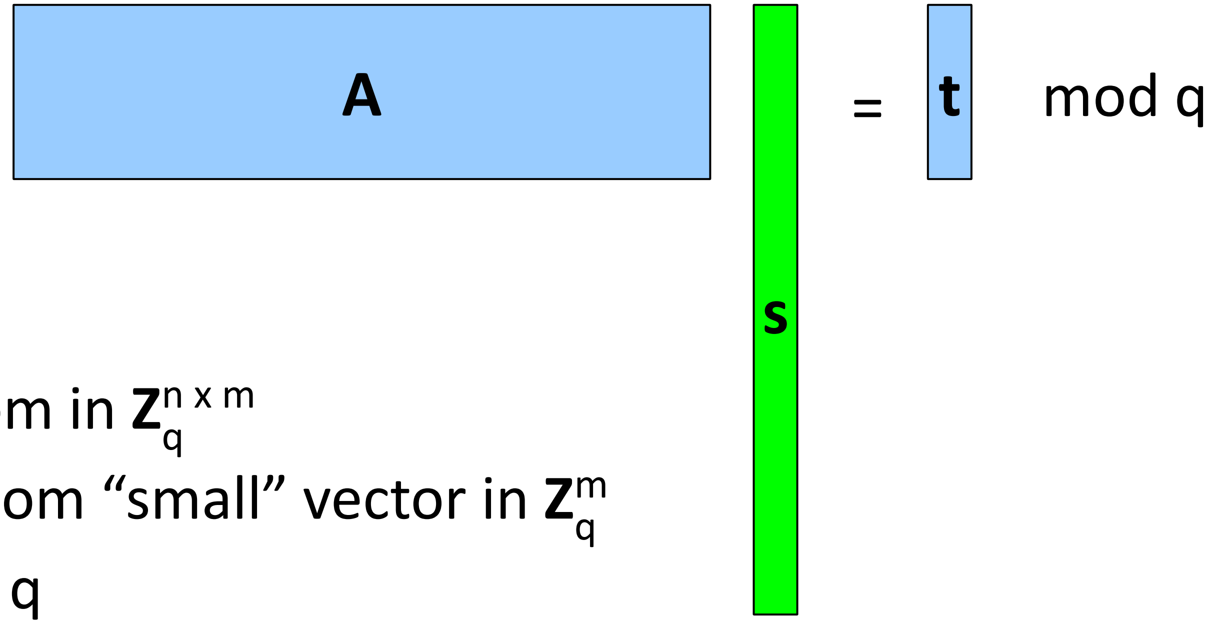
# The Knapsack Problem




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

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

# The Knapsack Problem

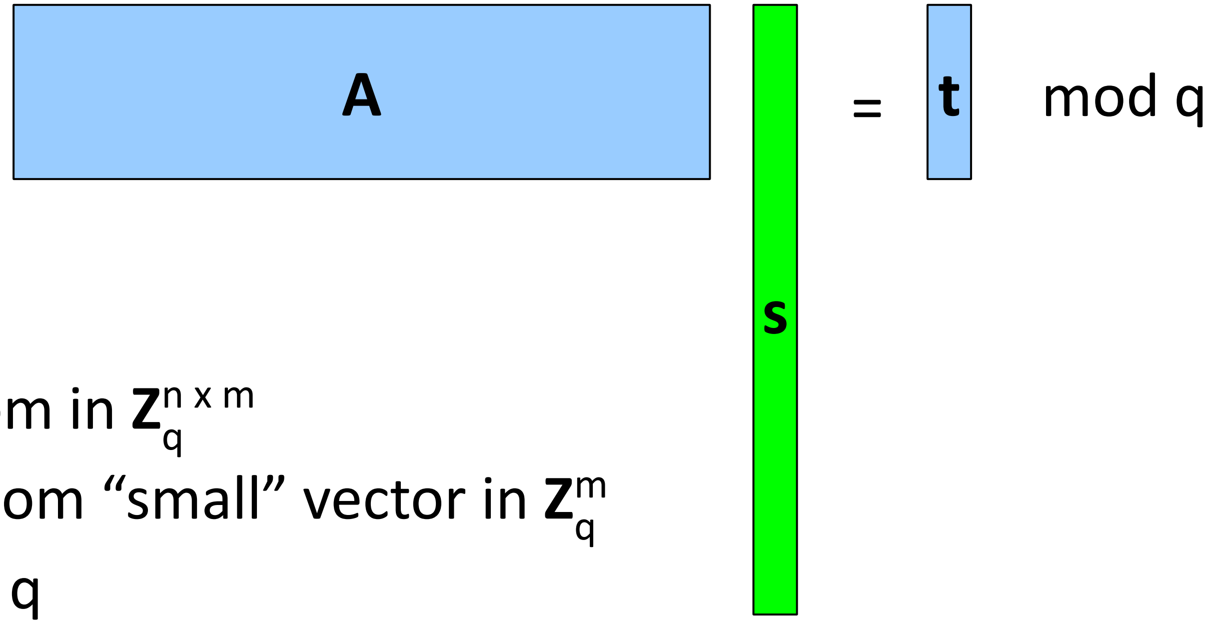



**A** is random in $\mathbf{Z}_q^{n \times m}$

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

**t**=**A**s mod q

# The Knapsack Problem




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

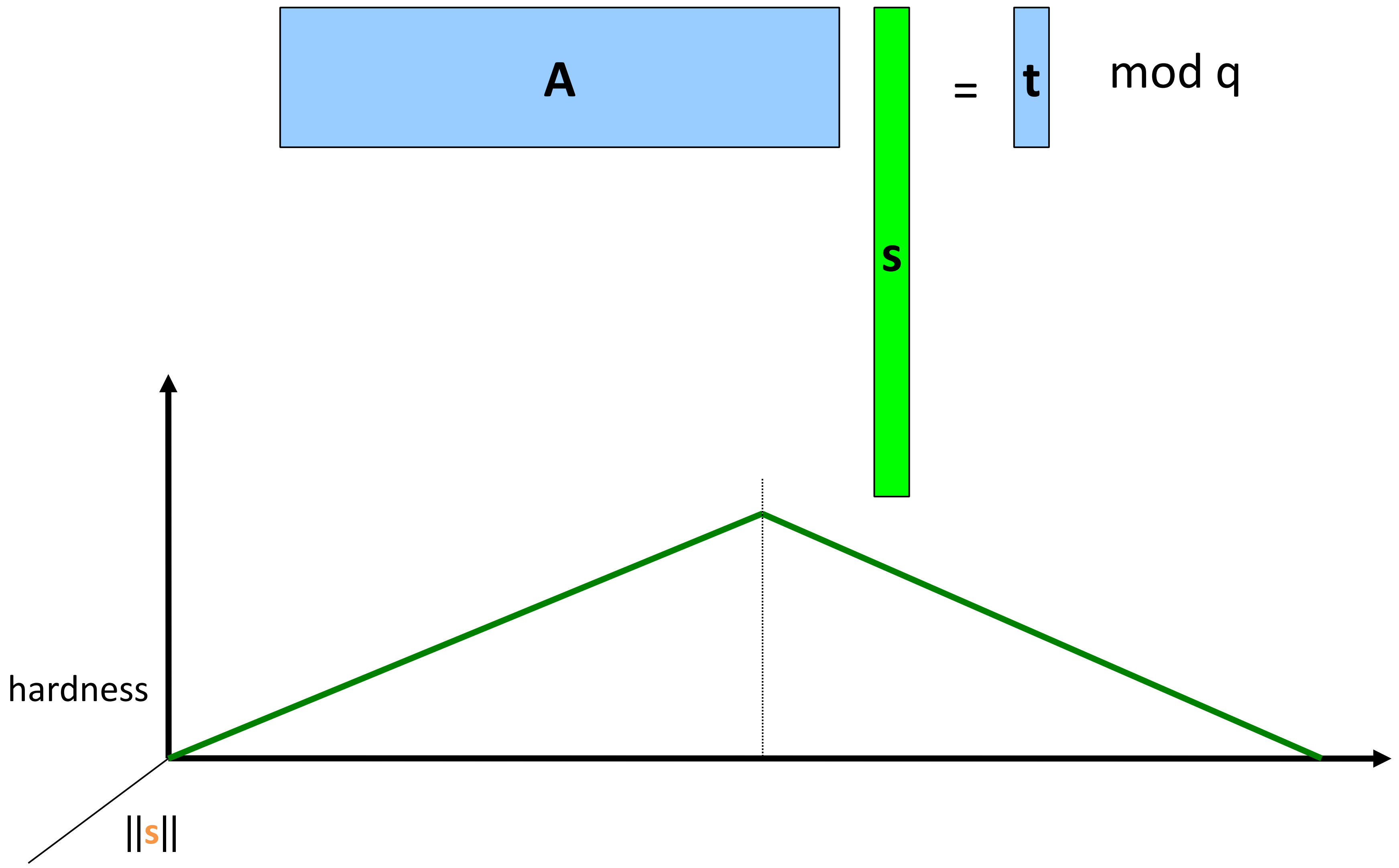
s is a random “small” vector in $\mathbf{Z}_q^{m}$

**t**=**A**s mod q

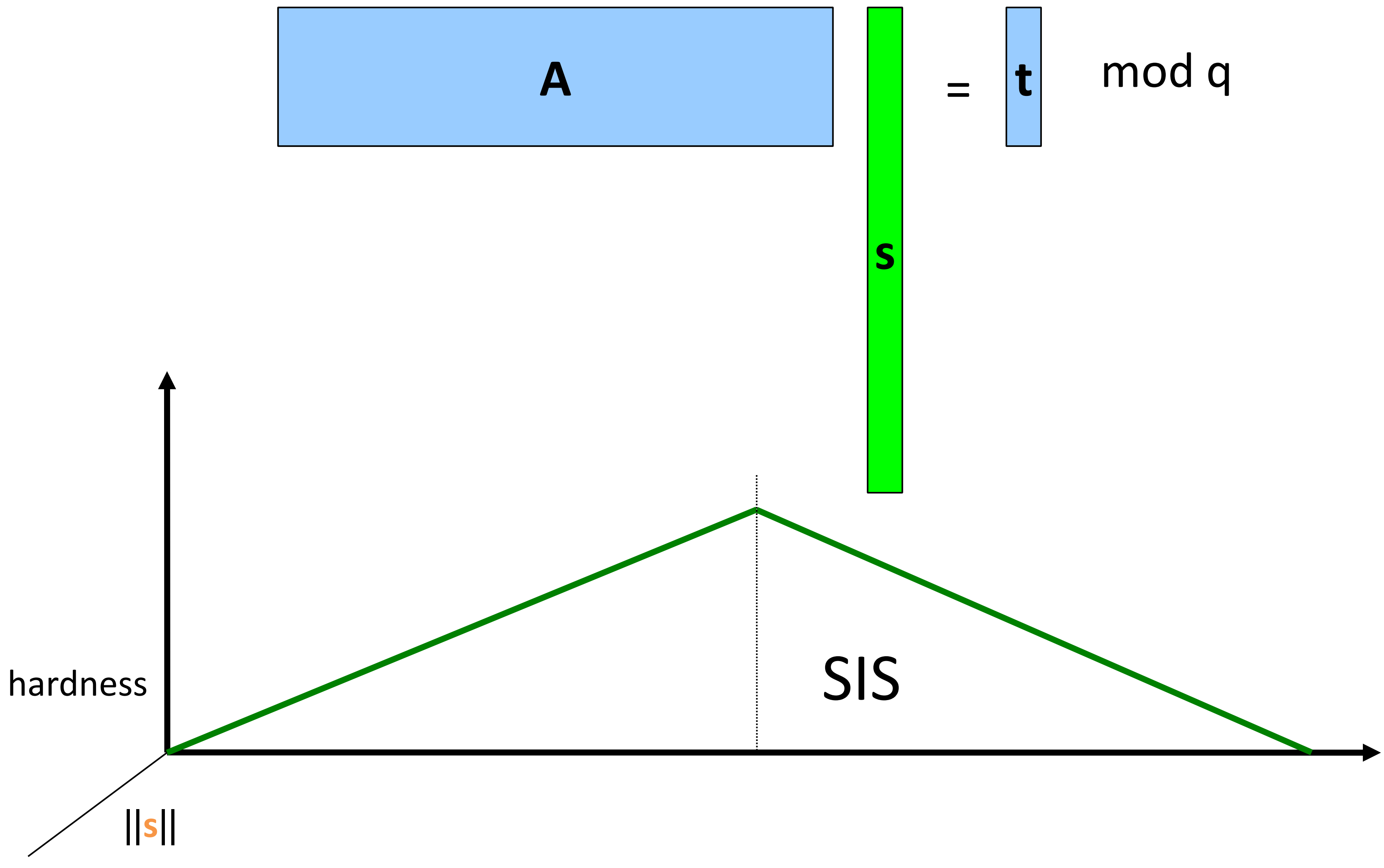
Given (**A**,**t**), find small **s'** such that

**A**s'=**t** mod q

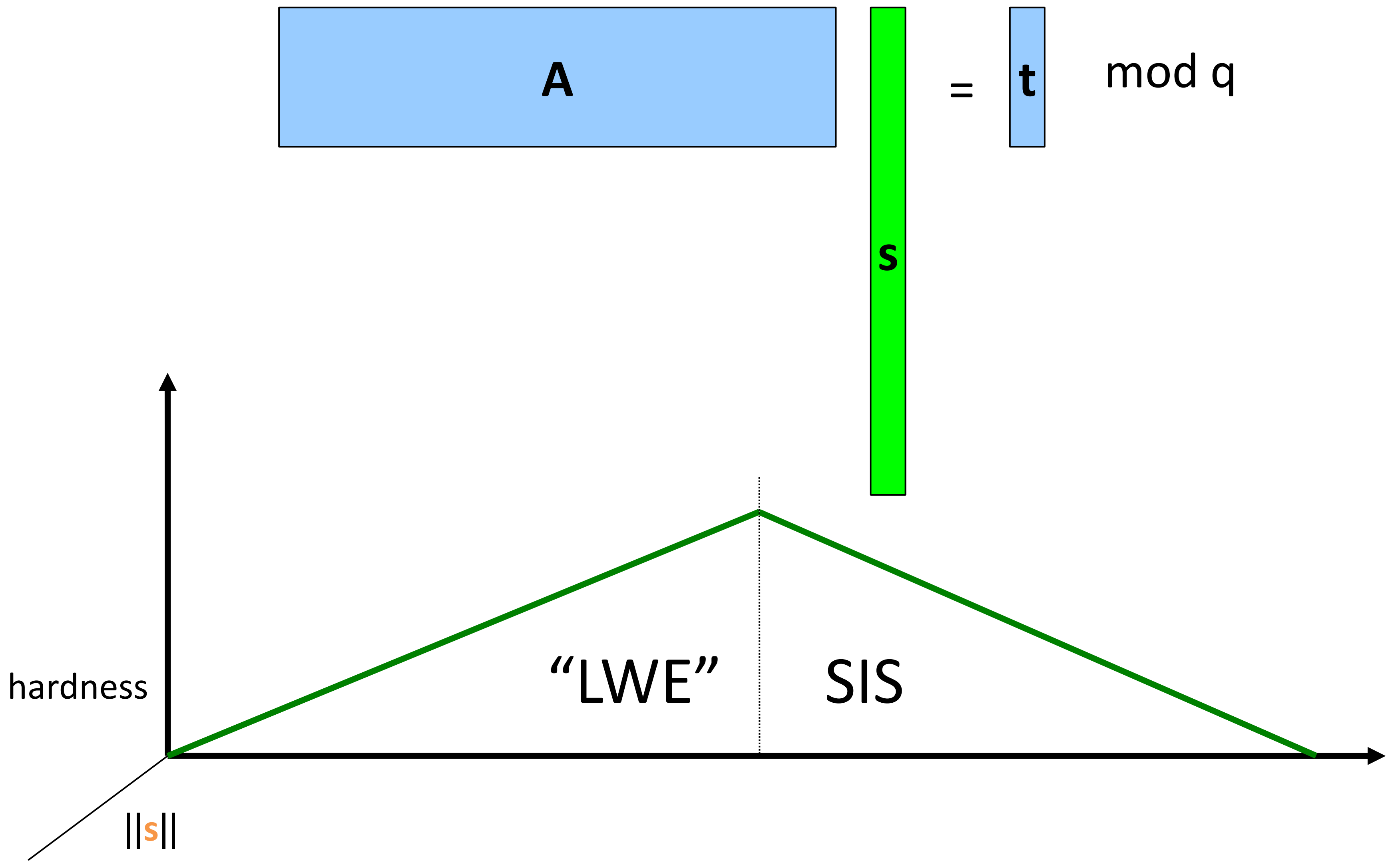
# Hardness of the Knapsack Problem



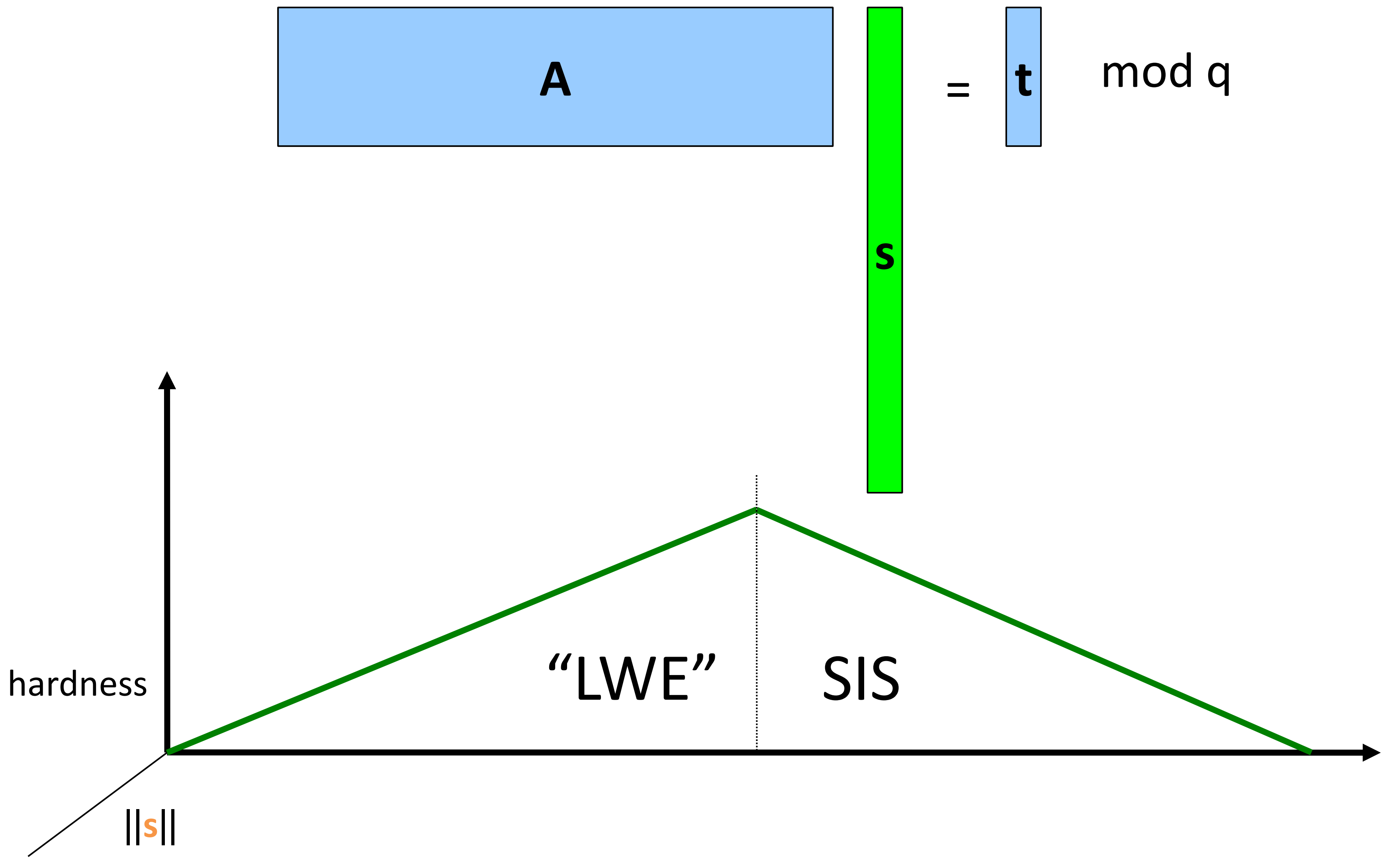
# Hardness of the Knapsack Problem



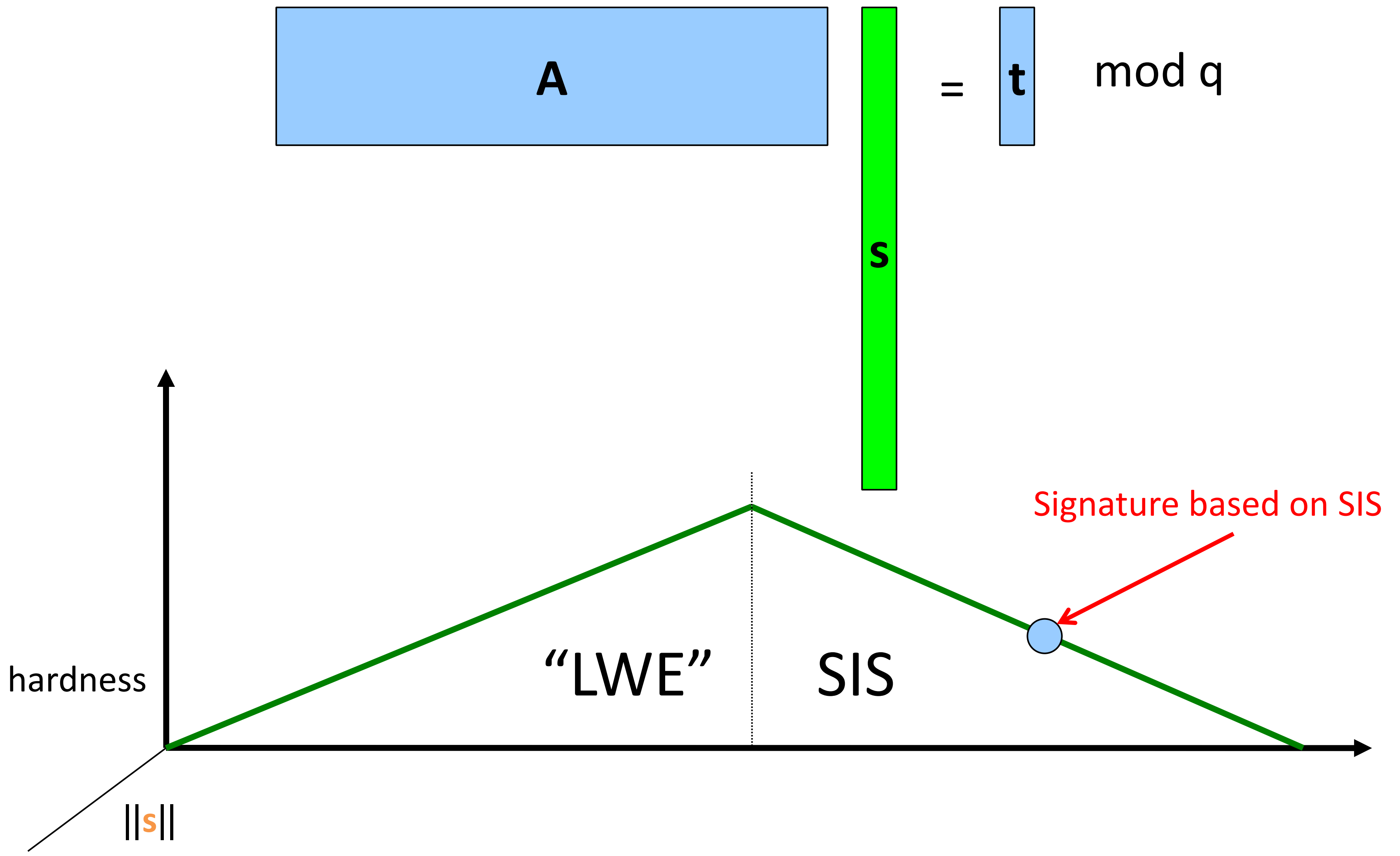
# Hardness of the Knapsack Problem



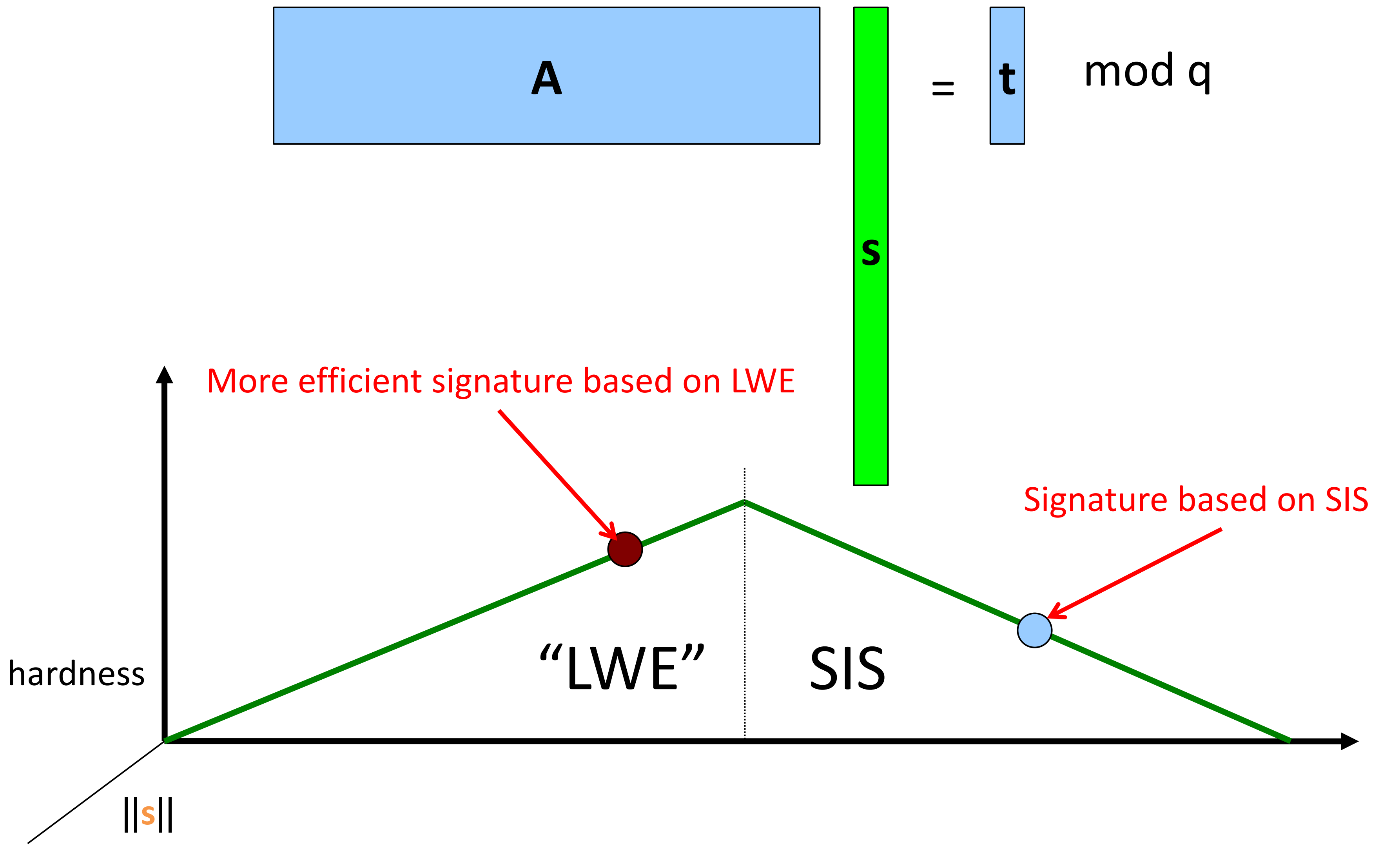
# Our Results



# Our Results

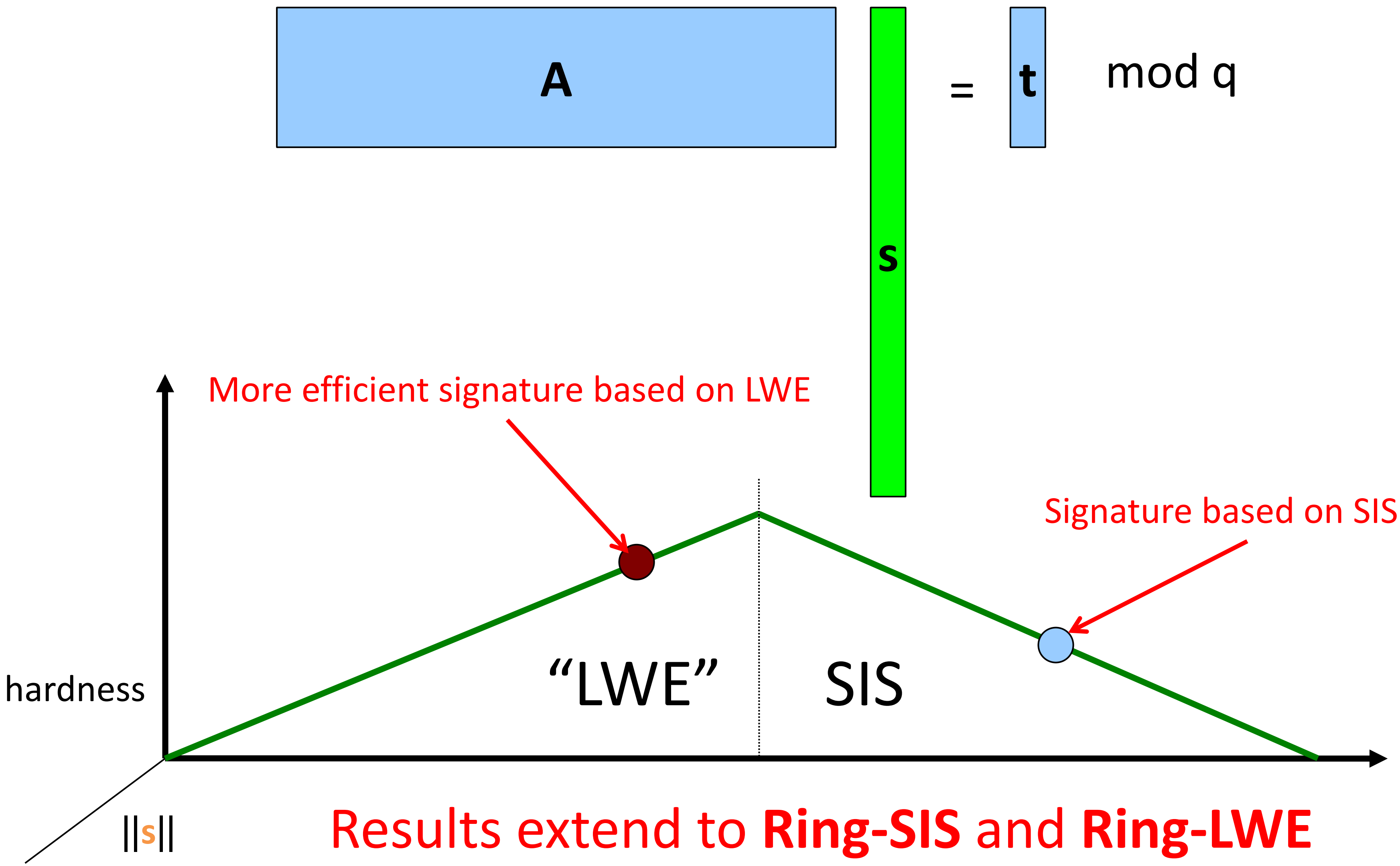


# Our Results



A
s
=
t
mod q
More efficient signature based on LWE
Signature based on SIS
hardness
"LWE"
SIS
||s||

# Our Results




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** mod q

# Signature Scheme (Main Idea)

Secret Key: $\mathbf{S}$

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

<u>Sign</u>(μ)

Pick a random $\mathbf{y}$

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

# Signature Scheme (Main Idea)

Secret Key: $\mathbf{S}$

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

<u>Sign</u>($\mu$)

Pick a random $\mathbf{y}$

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

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

# Signature Scheme (Main Idea)

Secret Key: $\mathbf{S}$

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

<u>Sign</u>(μ)

Pick a random $\mathbf{y}$

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

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

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

# Signature Scheme (Main Idea)

Secret Key: $S$

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

<u>Sign</u>(μ)

Pick a random $y$

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

$z=Sc+y$

Output($z$,$c$)

<u>Verify</u>($z$,$c$)

# Signature Scheme (Main Idea)

Secret Key: $S$

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

Sign($\mu$)

Pick a random $y$

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

$z=Sc+y$

Output($z$,$c$)

Verify($z$,$c$)

Check that $z$ is "small"

and

$c = H(\mathbf{A}z - \mathbf{T}c \bmod q, \mu)$

# Security Reduction Requirements

Secret Key: $\mathbf{S}$

*Given the public key, the secret key is not unique*

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

<u>Sign</u>(μ)

Pick a random $\mathbf{y}$

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

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

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

*Signature is independent of the secret key*

<u>Verify</u>($\mathbf{z}$,$\mathbf{c}$)

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

and

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

# Security Reduction

<u>Simulator</u> <u>Adversary</u>

# Security Reduction

Simulator

Adversary

**A**

# Security Reduction

<u>Simulator</u> | <u>Adversary</u>

**A**

Pick random **S**

# Security Reduction

Simulator

Adversary

**A** →

Pick random **S**

**A,AS** →

# Security Reduction

Simulator | Adversary

$\xrightarrow{\mathbf{A}}$ Pick random $\mathbf{S}$ $\xrightarrow{\mathbf{A},\mathbf{AS}}$

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

$\xrightarrow{(z_i, c_i)}$

...

# Security Reduction

Simulator | Adversary

$\mathbf{A}$

Pick random $\mathbf{S}$

$\mathbf{A}, \mathbf{AS}$

$\mu_i$

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

$(\mathbf{z_i}, \mathbf{c_i})$

…

$\mu, (\mathbf{z}, \mathbf{c})$

$\mu, (\mathbf{z'}, \mathbf{c'})$

# Security Reduction

| | Simulator | | Adversary |
|---|---|---|---|
| $\xrightarrow{\mathbf{A}}$ | Pick random $S$ | $\xrightarrow{\mathbf{A},\mathbf{A}S}$ | |
| | $(z_i,c_i)=\mathrm{Sign}(\mu_i)$ | $\xleftarrow{\mu_i}$ | |
| | | $\xrightarrow{(z_i,c_i)}$ | |
| | | … | |
| | $\mathbf{A}(z-z'+Sc'-Sc)=0$ | $\xleftarrow{\mu,\ (z,c)}$ | |
| | | $\xleftarrow{\mu,\ (z',c')}$ | |

# Security Reduction

<u>Simulator</u> <u>Adversary</u>

$\mathbf{A}$

Pick random $\mathbf{S}$

$\mathbf{A}, \mathbf{AS}$

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

$\mu_i$

$(z_i, c_i)$

…

$\mathbf{A}(\underbrace{z - z' + \mathbf{S}c' - \mathbf{S}c}) = 0$

If this is not 0, then **SIS** is solved.

Important for adversary to not know $\mathbf{S}$.

$\mu, (z, c)$

$\mu, (z', c')$

# Security Reduction

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

# Security Reduction

$$\mathbf{A}\underbrace{(\mathbf{z}-\mathbf{z}'+\mathbf{S}\mathbf{c}'-\mathbf{S}\mathbf{c})}_{\text{Solution to } \textbf{SIS}}=\mathbf{0}$$

<u>We Want:</u>

# Security Reduction

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

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

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

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** mod q

<u>Sign</u>(μ)

Pick a random **y**

Compute **c**=H(**Ay** mod q,μ)

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

Output(**z**,**c**)

# Signature Scheme

Secret Key: $\mathbf{S}$

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

<u>Sign</u>(μ)

Pick a random $\mathbf{y}$ make $\mathbf{y}$ uniformly random mod q?

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

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

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

# Signature Scheme

Secret Key: **S**

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

<u>Sign</u>(μ)

Pick a random **y** make **y** uniformly random mod q?

Compute **c**=H(**Ay** mod q,μ)

**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** mod q

<u>Sign</u>(μ)

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

Compute **c**=H(**Ay** mod q,μ)

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

Output(**z**,**c**)

# Signature Scheme

Secret Key: **S**

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

<u>Sign</u>(μ)

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

Compute **c**=H(**Ay** mod q,μ)

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

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

# Rejection Sampling

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**

# Rejection Sampling

Secret Key: **S**

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

<u>Sign</u>(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**=**A**S mod q

<u>Sign</u>(μ)

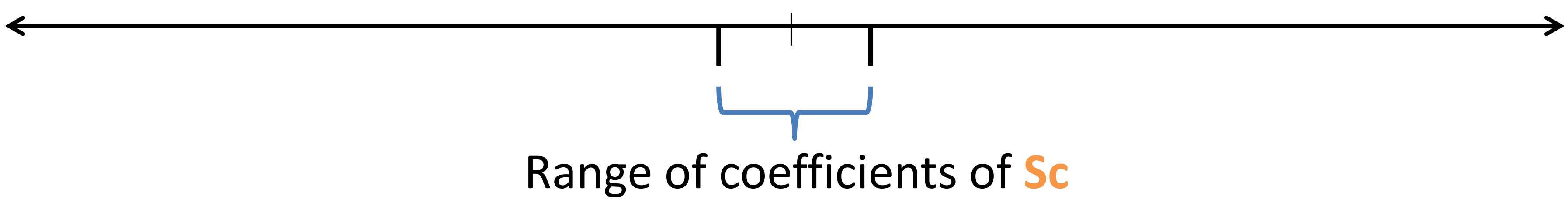
Pick a random **y** make **y** small

Compute **c**=H(**A**y mod q,μ)

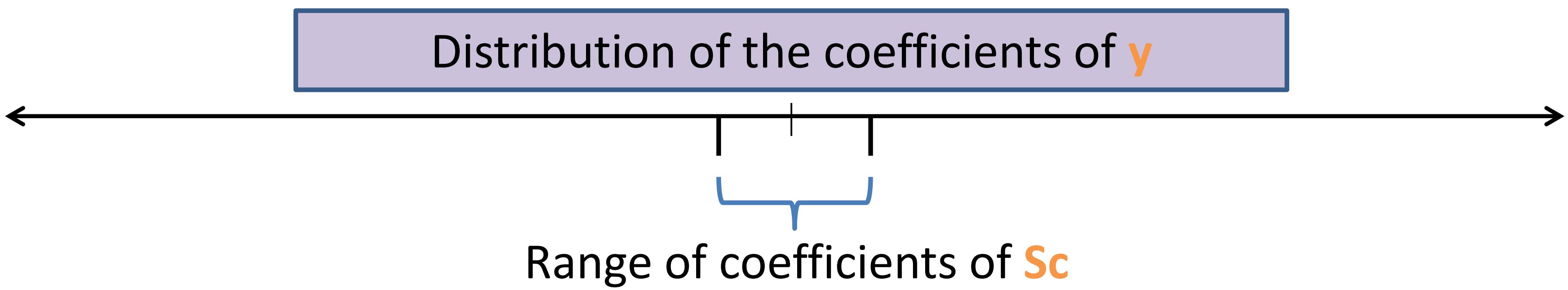
**z**=**Sc**+**y**

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

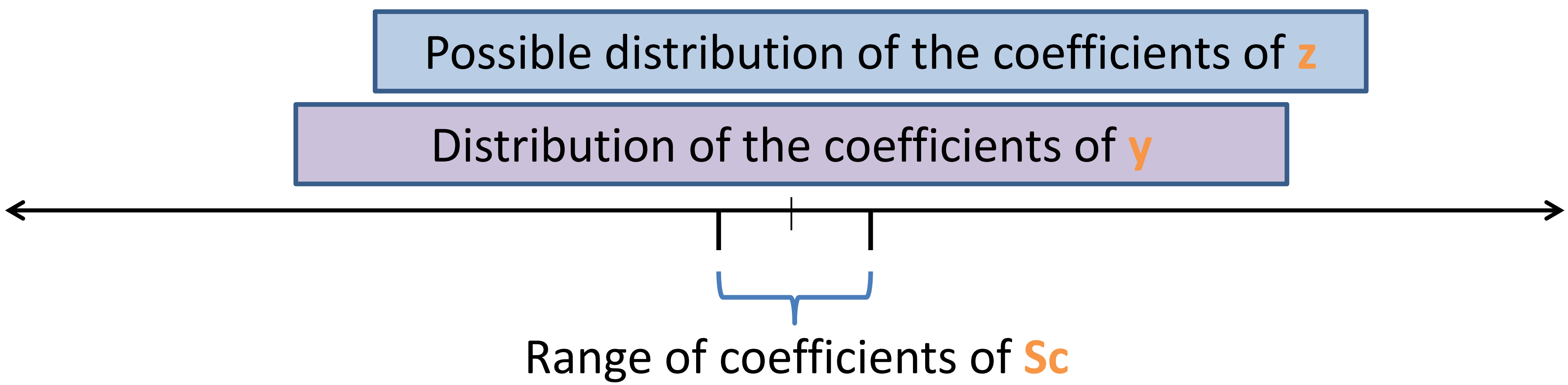
# Rejection Sampling (L ‘09)



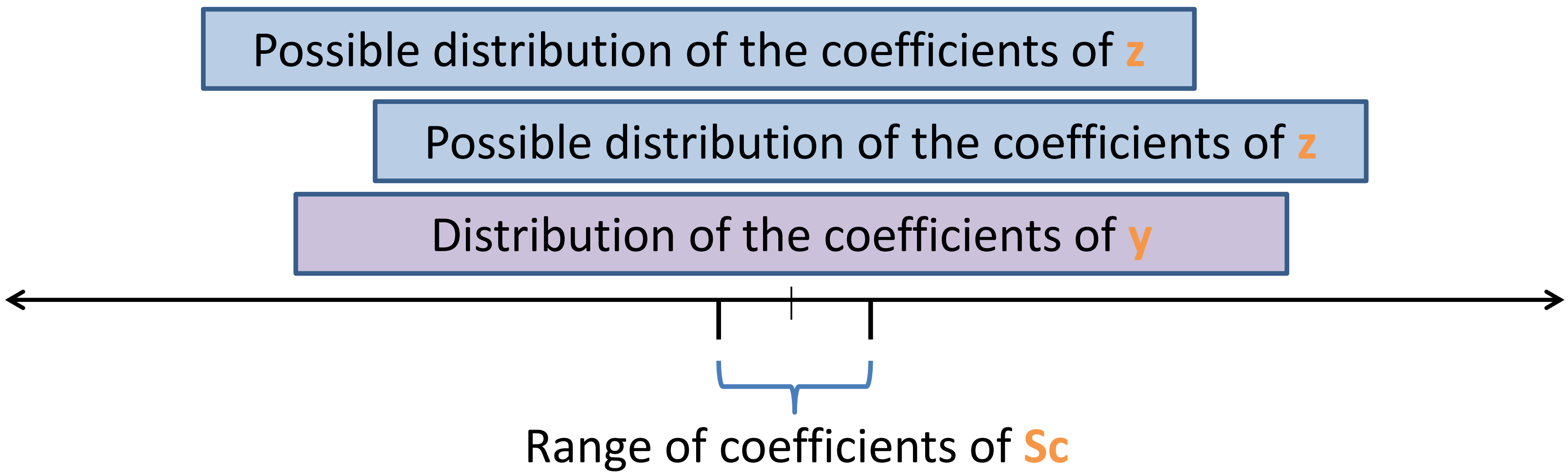
# Rejection Sampling (L '09)



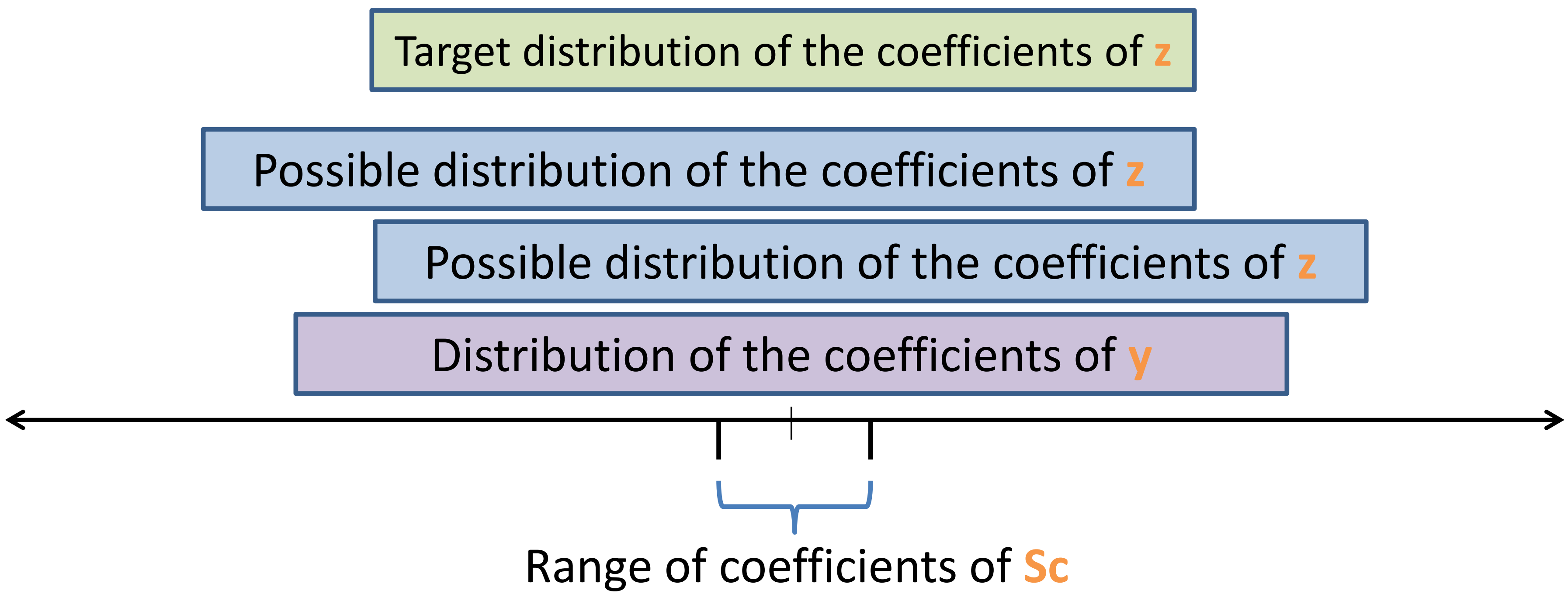
# Rejection Sampling (L '09)



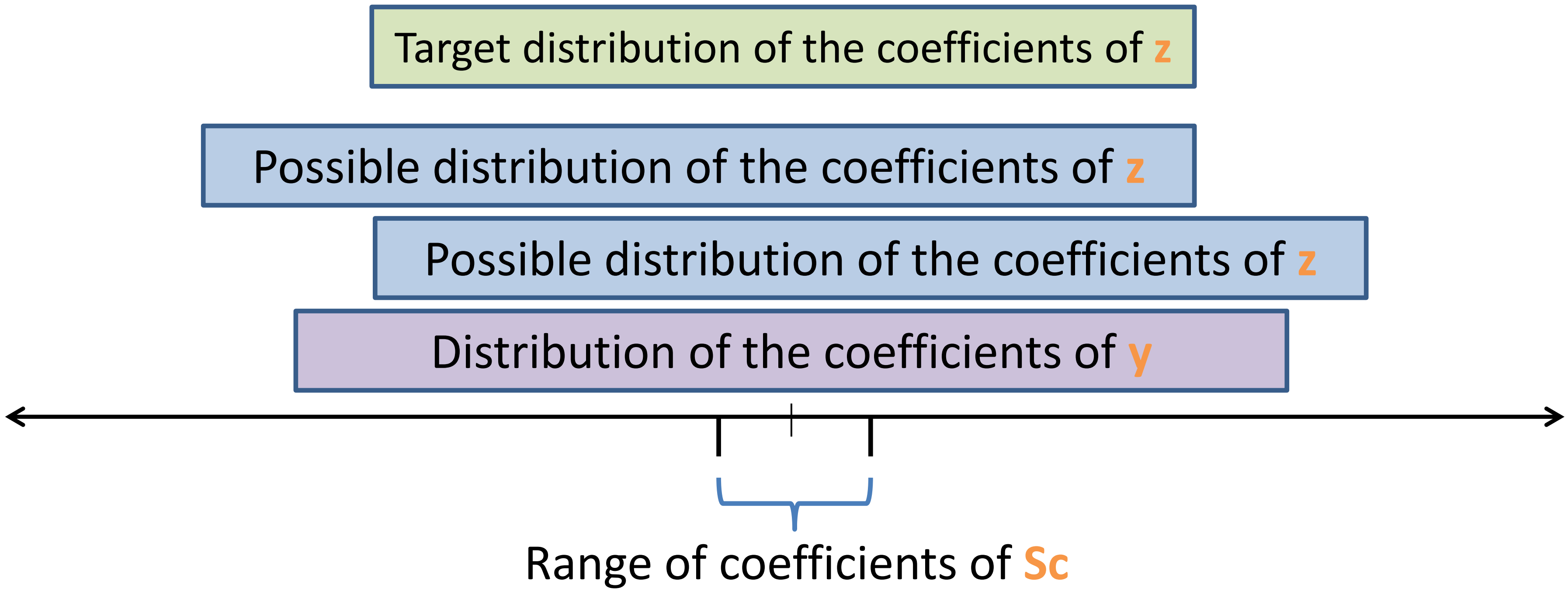
# Rejection Sampling (L '09)



# Rejection Sampling (L '09)



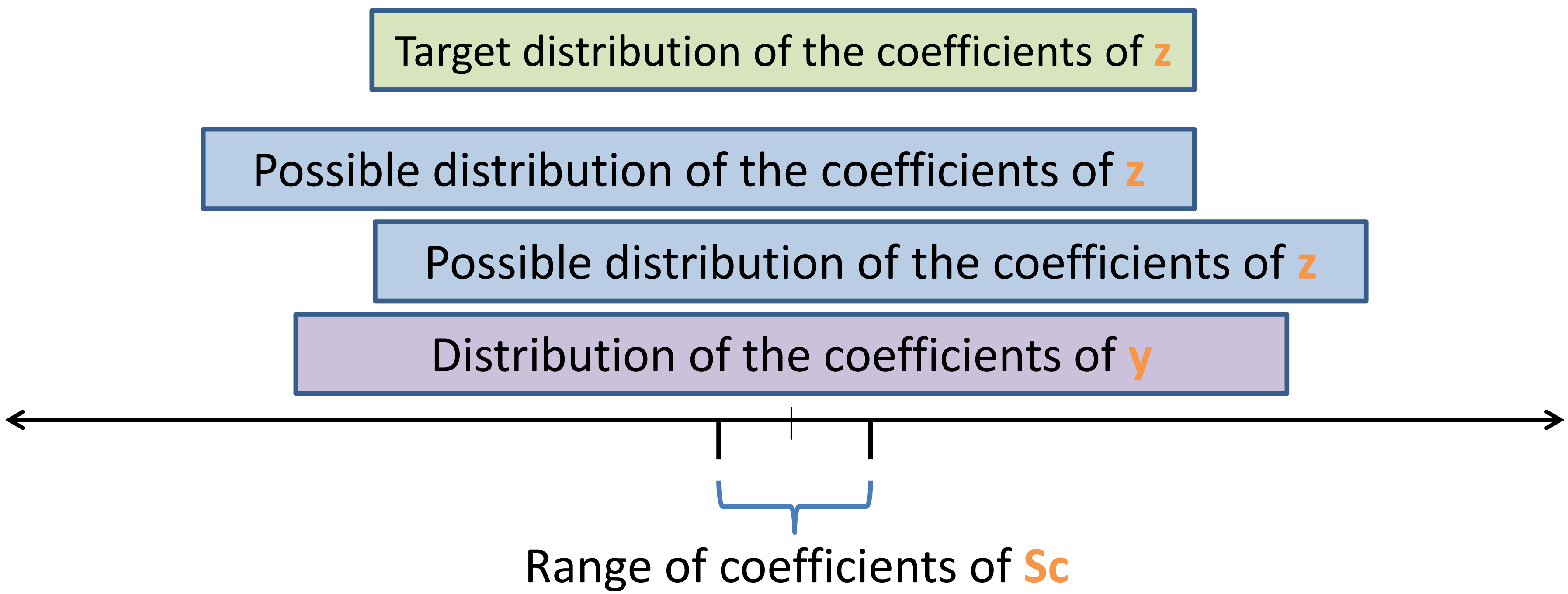
# Rejection Sampling (L '09)




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

Want $p^m \approx$ constant

# Rejection Sampling (L '09)




Probability each coefficient of **z** is in the target range = p
Want $p^m \approx$ constant
So $p \approx 1\text{-}1/m$
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 **Sc** = O(1)

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

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

# 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??

# 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??

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:

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

m-dimensional *discrete* normal distribution

$$D^{\mathbf{m}}_{\sigma,\mathbf{v}}(\mathbf{x}) = \rho^{\mathbf{m}}_{\sigma,\mathbf{v}}(\mathbf{x}) \;/\; \rho^{\mathbf{m}}_{\sigma,\mathbf{0}}(\mathbf{Z}^{\mathbf{m}})$$

# Different Rejection Sampling

Secret Key: $\mathbf{S}$

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

<u>Sign</u>(μ)

Pick a random $\mathbf{y}$

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

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

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

# Different Rejection Sampling

Secret Key: $\mathbf{S}$

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

Sign(μ)

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

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

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

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

# Different Rejection Sampling

Secret Key: **S**

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

<u>Sign</u>(μ)

Pick a random **y** ~ $D^{\mathbf{m}}_{\sigma,\mathbf{0}}$

Compute **c**=H(**Ay** mod q,μ)

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

Output(**z**,**c**)

# Different Rejection Sampling

Secret Key: **S**

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

<u>Sign</u>(μ)

Pick a random **y** ~ $D^{\mathbf{m}}_{\sigma,\mathbf{0}}$

Compute **c**=H(**Ay** mod q,μ)

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

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

# Different Rejection Sampling

Secret Key: **S**

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

<u>Sign</u>(μ)

Pick a random **y** ~ $D^{\mathbf{m}}_{\sigma,\mathbf{0}}$

Compute **c**=H(**Ay** mod q,μ)

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

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

Pick σ = O(√m), k = O(1) → ||**z**|| = O(m)

# Signature Based on **LWE**

# Security Reduction Requirements

Secret Key: $\mathbf{S}$

*Given the public key, the secret key is not unique*

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

<u>Sign</u>(μ)

Pick a random $\mathbf{y}$

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

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

Output($\mathbf{z}$,$\mathbf{c}$) (or reject)

*Signature is independent of the secret key*

<u>Verify</u>($\mathbf{z}$,$\mathbf{c}$)

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

and

$\mathbf{c} = H(\mathbf{A}\mathbf{z} - \mathbf{T}\mathbf{c} \bmod q, \mu)$

# Security Reduction Requirements

Secret Key: **S**

~~*Given the public key, the secret key is not unique*~~

Public Key: **A**, **T**=**A**S mod 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** mod q,μ)

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

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

*Signature is independent of the secret key*

Verify(**z**,**c**)

Check that **z** is "small"

and

**c** = H(**Az** – **Tc** mod q, μ)

# Security Intuition

Secret Key: $\mathbf{S}$

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

<u>Sign</u>($\mu$)

Pick a random $\mathbf{y}$

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

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

Output($\mathbf{z}$,$\mathbf{c}$) (or reject)

*Signature is independent of the secret key*

# Security Intuition

Secret Key: $\mathbf{S}$

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

Sign($\mu$)

Pick a random $\mathbf{y}$

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

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

Output($\mathbf{z}$,$\mathbf{c}$) (or reject)

*Signature is independent of the secret key*

Secret Key: $\mathbf{S}$

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

# Security Intuition

Secret Key: $\mathbf{S}$

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

Sign($\mu$)

Pick a random $\mathbf{y}$

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

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

Output($\mathbf{z}$,$\mathbf{c}$) (or reject)

*Signature is independent of the secret key*

Secret Key: $\mathbf{S}$

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

*The secret key is not unique*

# Security Intuition

Secret Key: $\mathbf{S}$

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

Sign($\mu$)

Pick a random $\mathbf{y}$

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

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

Output($\mathbf{z},\mathbf{c}$) (or reject)

*Signature is independent of the secret key*

Secret Key: $\mathbf{S}$

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

*The secret key is not unique*

Sign($\mu$)

Pick a random $\mathbf{c}$

Pick a random $\mathbf{z}$

# Security Intuition

Secret Key: $\mathbf{S}$

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

Sign(μ)

Pick a random $\mathbf{y}$

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

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

Output($\mathbf{z}$,$\mathbf{c}$) (or reject)

*Signature is independent of the secret key*

Secret Key: $\mathbf{S}$

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

*The secret key is not unique*

Sign(μ)

Pick a random $\mathbf{c}$

Pick a random $\mathbf{z}$

*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

$\mathbf{A} \rightarrow$

Pick random $\mathbf{S}$

$\xrightarrow{\mathbf{A},\mathbf{AS}}$

$\xleftarrow{\mu_i}$

Generate $(z,c)$

Program $H(\mathbf{A}z-\mathbf{T}c,\mu)=c$

$\xrightarrow{(z_i,c_i)}$

…

$\xleftarrow{\mu,\ (z,c)}$

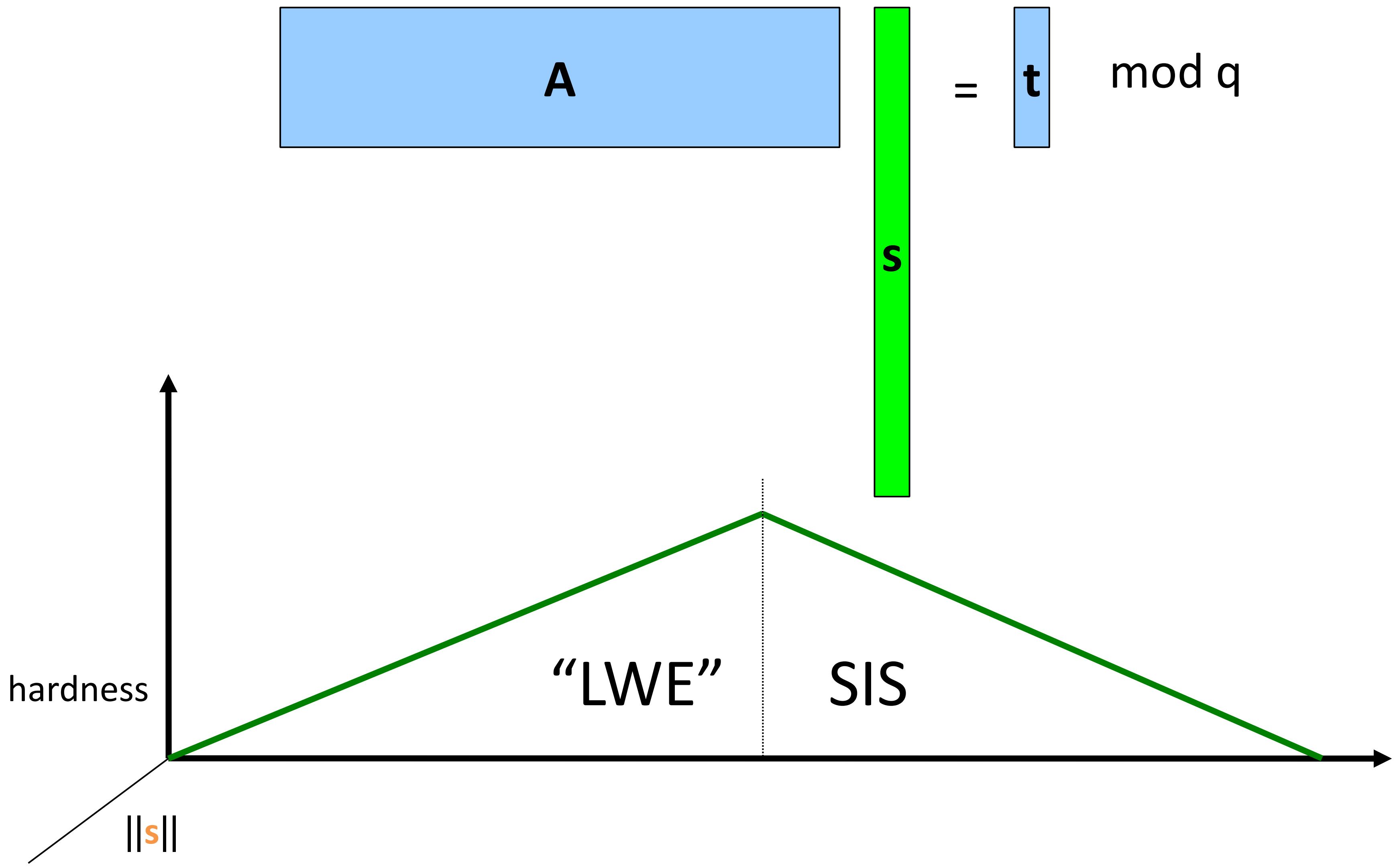
$\xleftarrow{\mu,\ (z',c')}$

$$\mathbf{A}(\underbrace{z-z'+\mathbf{S}c'-\mathbf{S}c}_{})=0$$

If this is not 0, then **SIS** is solved.

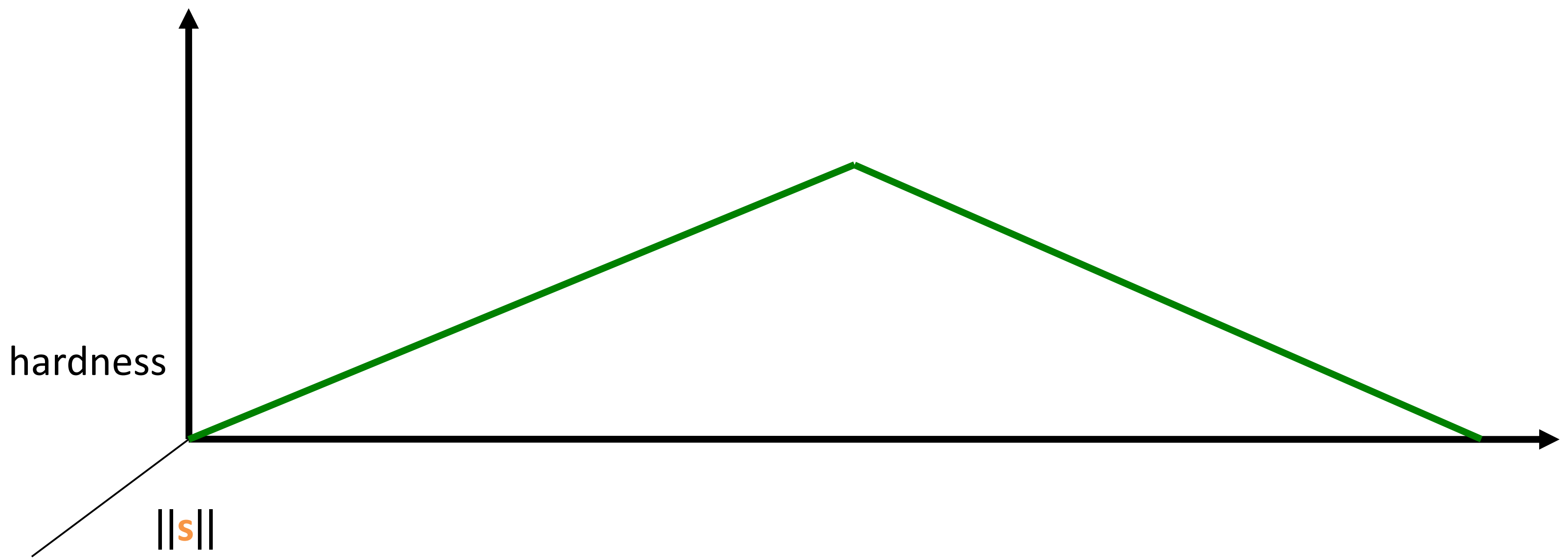
Important for adversary to not know **S**.

# Hardness of the Knapsack Problem



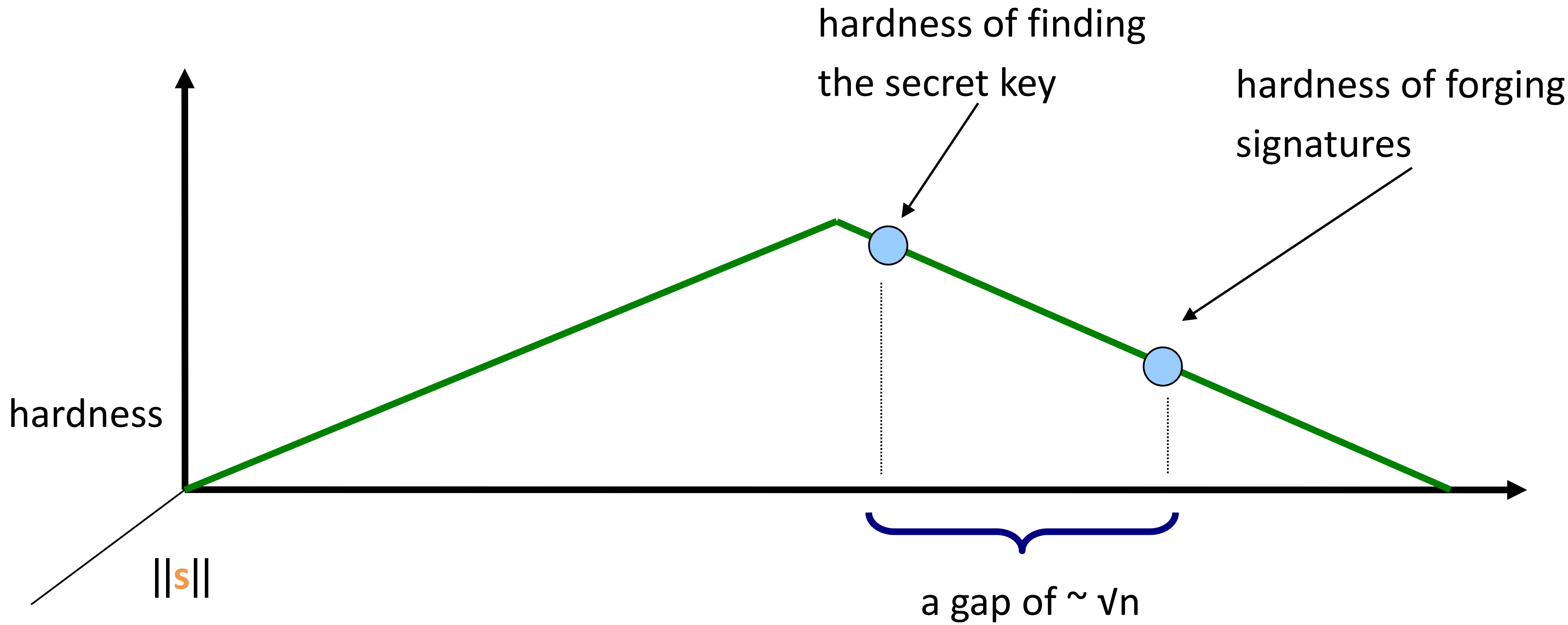
# Signature Hardness

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



# Signature Hardness

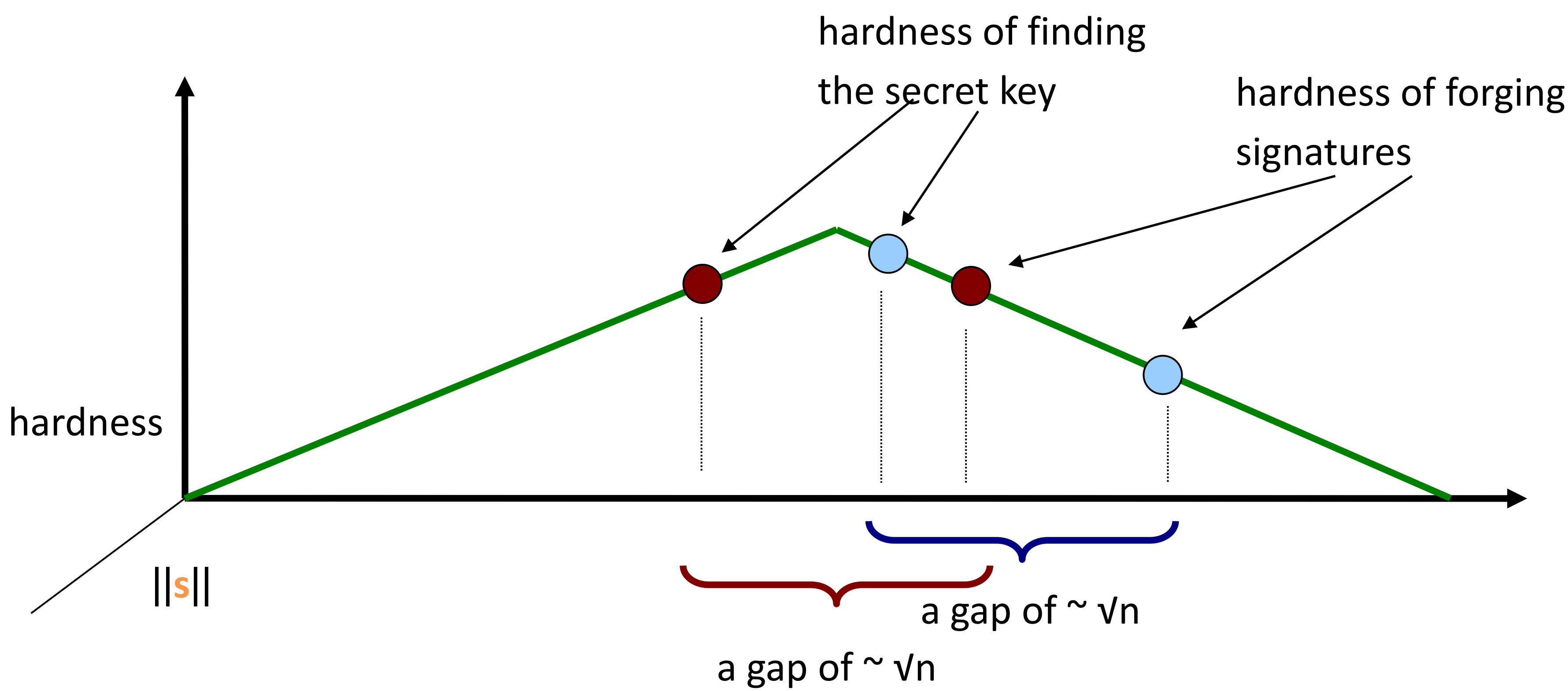
- Construction based on **SIS**
- Construction based on **LWE**



# Signature Hardness

Construction based on **SIS**

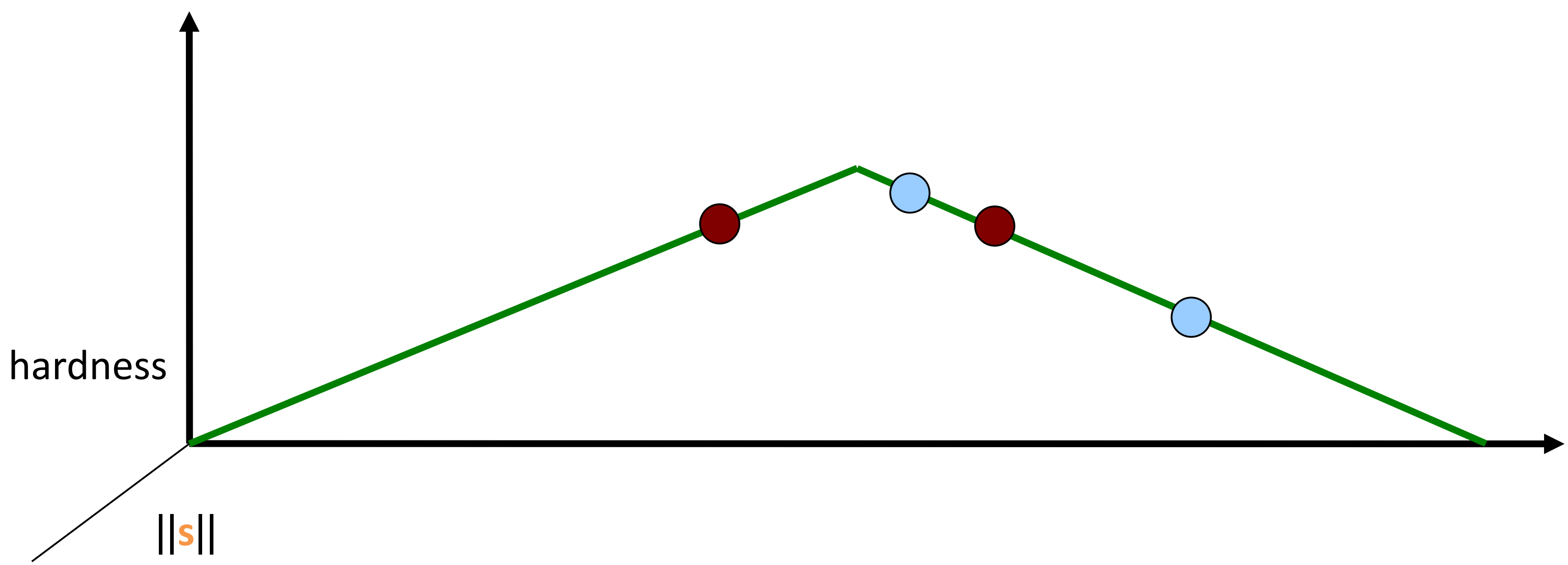
Construction based on **LWE**



# 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]