Lattice-Based Cryptography in a Quantum Setting: Security Proofs & Attacks

PhD Defense

Pouria Fallahpour

supervisors:
Damien Stehlé & Gilles Villard

UNIVERSITÉ DE LYON

ENS DE LYON

Outline

Introduction

Knowledge Assumption

Fiat-Shamir Transform

Conclusion

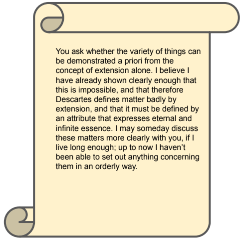
Outline

- Introduction
- Knowledge Assumption
- Fiat-Shamir Transform
- Conclusion

How do signatures work?

Spinoza

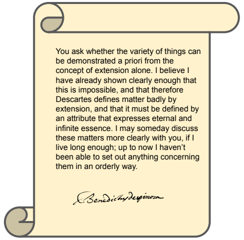
von Tschirnhaus



How do signatures work?

Spinoza

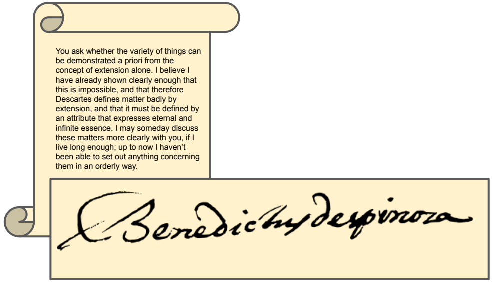
von Tschirnhaus



How do signatures work?

Spinoza

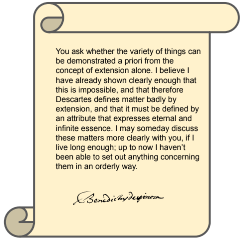
von Tschirnhaus



How do signatures work?

Spinoza

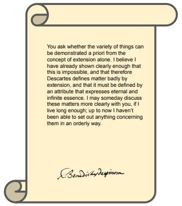
von Tschirnhaus



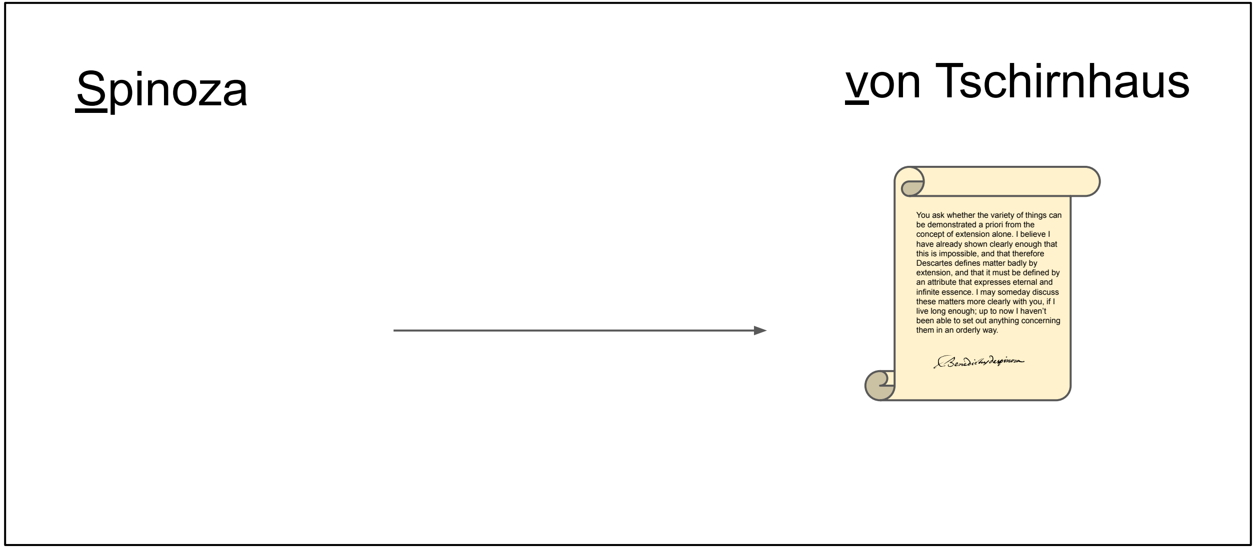
How do signatures work?

Spinoza

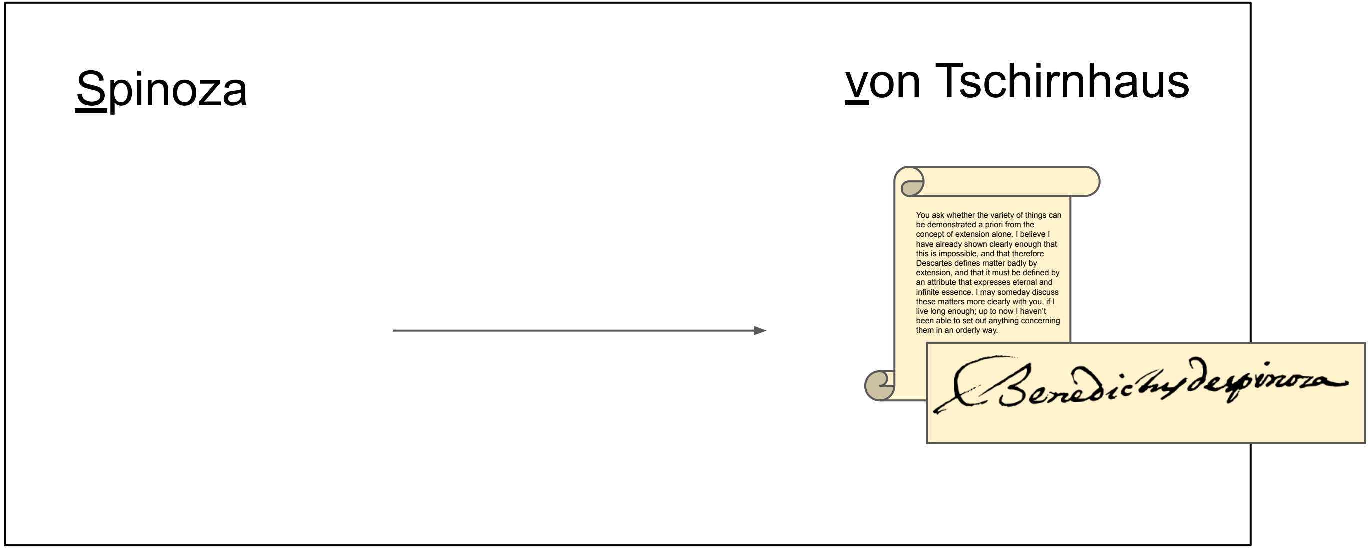
von Tschirnhaus



How do signatures work?



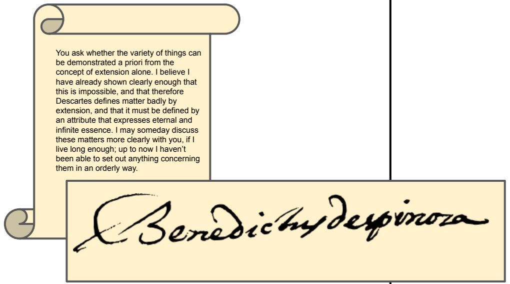
How do signatures work?



How do signatures work?

Spinoza

von Tschirnhaus



Can verify the authenticity

Cryptographic signatures

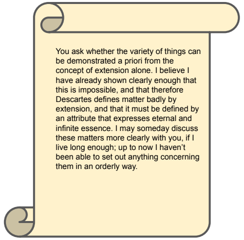
Spinoza

(pk, sk)

$\xrightarrow{\quad pk \quad}$

von Tschirnhaus

$\mu :=$



Cryptographic signatures

Spinoza

von Tschirnhaus

(pk, sk)

$$\xrightarrow{pk}$$

$\mu \in \{0,1\}^*$

Cryptographic signatures

Spinoza | von Tschirnhaus

(pk, sk)

$\xrightarrow{pk}$

$\sigma \leftarrow \mathsf{Sign}(sk, \mu)$

$\xrightarrow{(\mu, \sigma)}$

$\mathsf{Verify}(pk, \mu, \sigma) \in \{0, 1\}$

Can verify the authenticity

Cryptographic signatures

Signature syntax

Signer

(pk, sk)

pk

Verifier

$\sigma \leftarrow \mathsf{Sign}(sk, \mu)$

(μ, σ)

$\mathsf{Verify}(pk, \mu, \sigma) \in \{0, 1\}$

Can verify the authenticity

Security definition

CMA security game

Challenger		Adversary
(pk, sk)	$\xrightarrow{\quad pk \quad}$	
	$\xleftarrow{\quad \mu_i \quad}$	
$\sigma_i \leftarrow \mathsf{Sign}(sk, \mu_i)$	$\xrightarrow{\quad \sigma_i \quad}$	
		forges(μ^*, σ^*)
	$\xleftarrow{\quad (\mu^*, \sigma^*) \quad}$	

adversary wins if $\forall i: \ \mu^* \neq \mu_i$ and $\mathsf{Verify}(pk, \mu^*, \sigma^*) = 1$

Security definition

CMA security game

Challenger		Adversary
(pk, sk)	$\xrightarrow{\quad pk \quad}$	
	$\xleftarrow{\quad \mu_i \quad}$	
$\sigma_i \leftarrow \mathsf{Sign}(sk, \mu_i)$	$\xrightarrow{\quad \sigma_i \quad}$	
		forges (μ^*, σ^*)
	$\xleftarrow{\quad (\mu^*, \sigma^*) \quad}$	

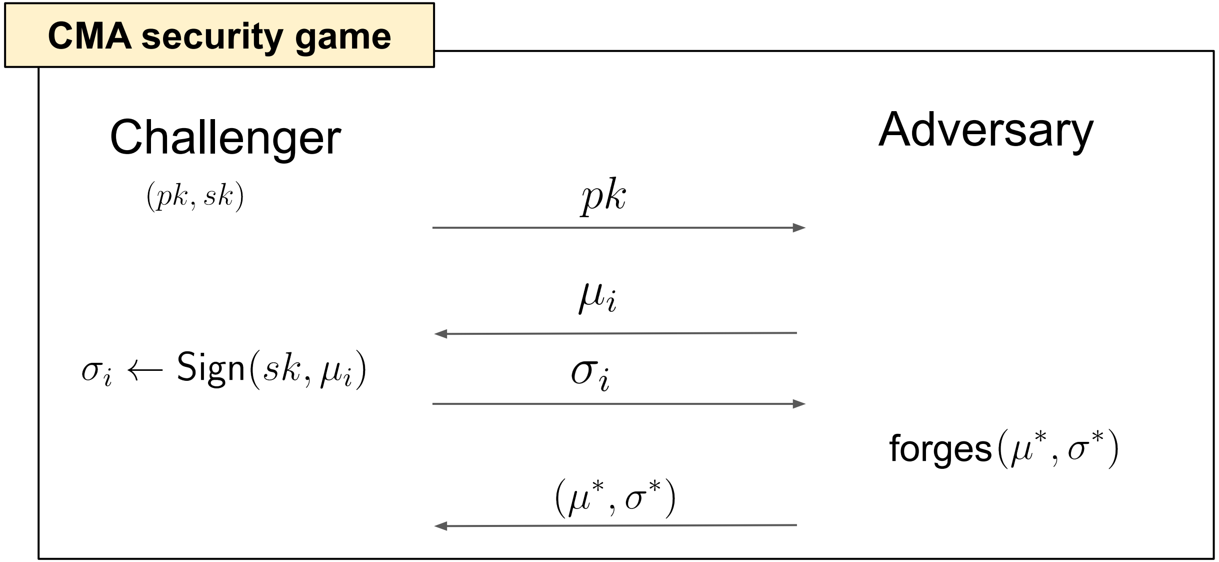
adversary wins if $\forall i: \ \mu^* \neq \mu_i$ and $\mathsf{Verify}(pk, \mu^*, \sigma^*) = 1$

CMA security

A signature scheme is CMA-insecure if a poly-time adversary wins the CMA game with non-negligible probability; otherwise it is CMA-secure

Security proof

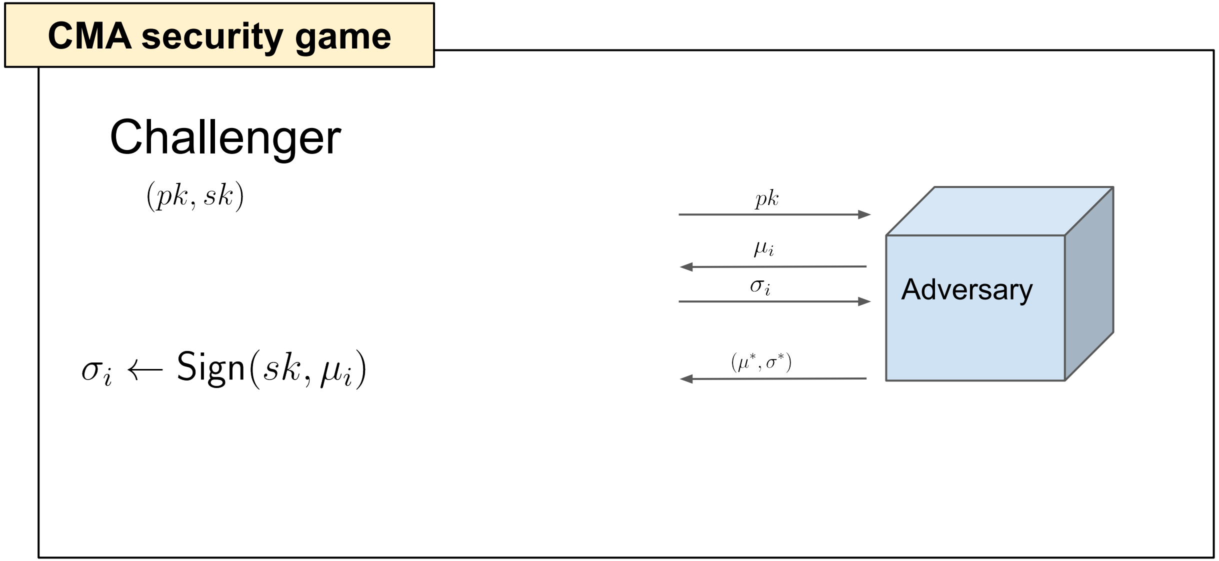
How to prove the security? By contradiction.



Assume that adversary wins, i.e., $\forall i: \ \mu^* \neq \mu_i$, and $\mathsf{Verify}(pk, \mu^*, \sigma^*) = 1$

Security proof

How to prove the security? By contradiction.



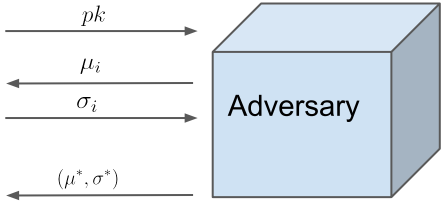
Assume that adversary wins, i.e., $\forall i: \ \mu^* \neq \mu_i$, and $\mathsf{Verify}(pk, \mu^*, \sigma^*) = 1$

Security proof

How to prove the security? By contradiction.

What can we do with a machine?

- Feed it simulated data
- Measure its wires
- Tweak its randomness
- Rewind it
- …



such that $\forall i: \ \mu^* \neq \mu_i$ and $\text{Verify}(pk, \mu^*, \sigma^*) = 1$

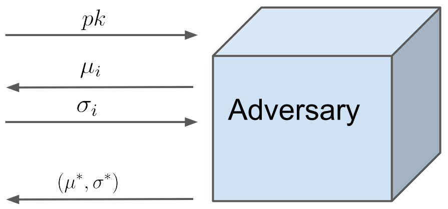
Security proof

How to prove the security? By contradiction.

A.k.a. cryptographic assumption

What can we do with a machine?

- Feed it simulated data
- Measure its wires
- Tweak its randomness
- Rewind it
- …



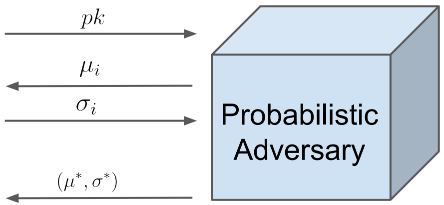
such that $\forall i: \ \mu^* \neq \mu_i$ and $\mathsf{Verify}(pk, \mu^*, \sigma^*) = 1$

we use the machine to solve a computational problem that is assumed to be hard-to-solve for poly-time algorithms.

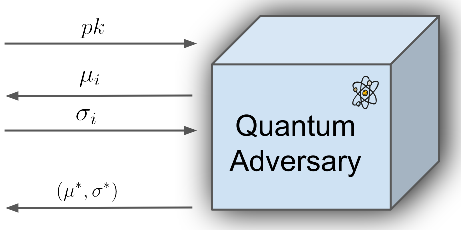
Since the adversary is poly-time, this is a contradiction.

Probabilistic vs Quantum machines

Are they the same?



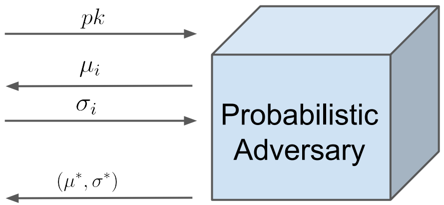
such that $\forall i: \ \mu^* \neq \mu_i$ and Verify$(pk, \mu^*, \sigma^*) = 1$



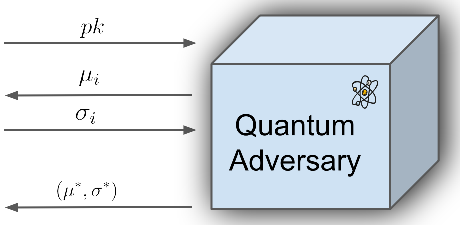
such that $\forall i: \ \mu^* \neq \mu_i$ and Verify$(pk, \mu^*, \sigma^*) = 1$

Probabilistic vs Quantum machines

Are they the same?



such that $\forall i: \ \mu^* \neq \mu_i$ and Verify$(pk, \mu^*, \sigma^*) = 1$

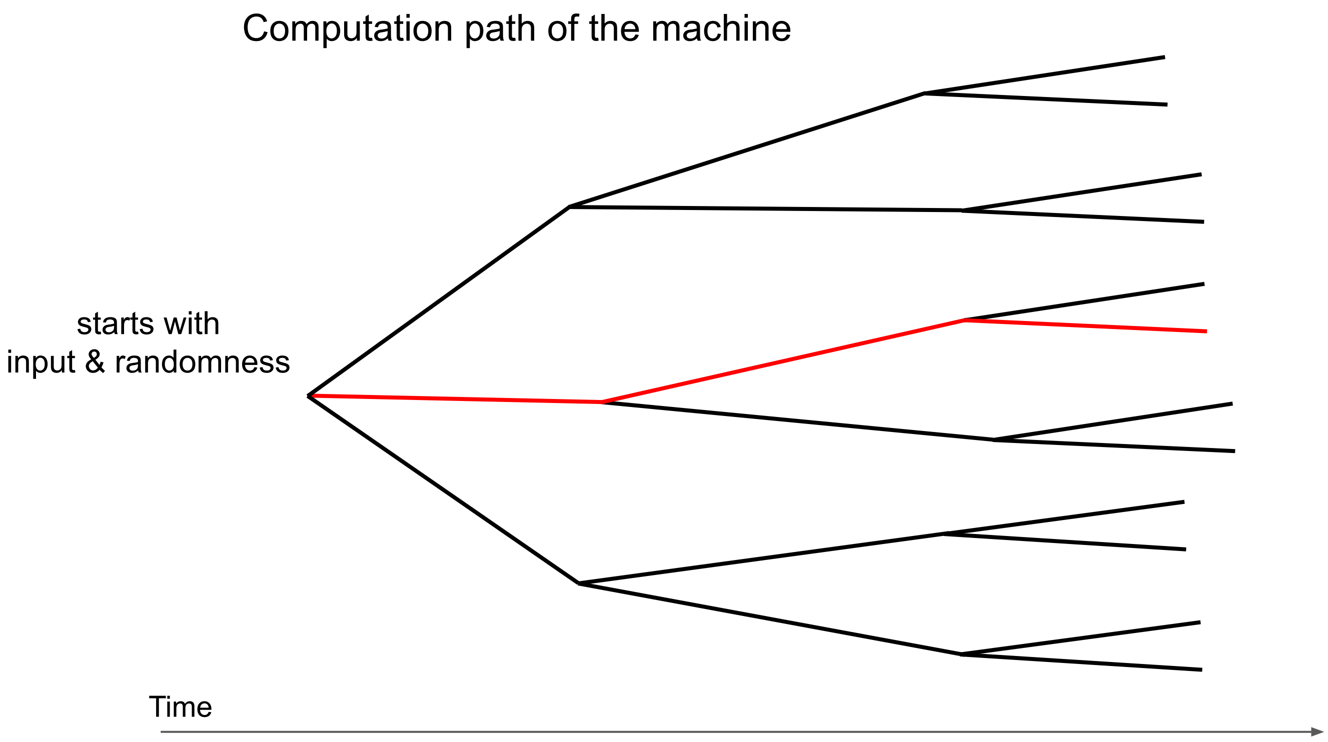


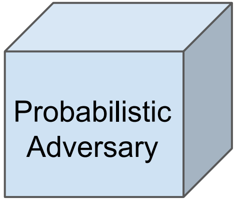
such that $\forall i: \ \mu^* \neq \mu_i$ and Verify$(pk, \mu^*, \sigma^*) = 1$

DLog assumption is not broken yet by probabilistic poly-time adversaries

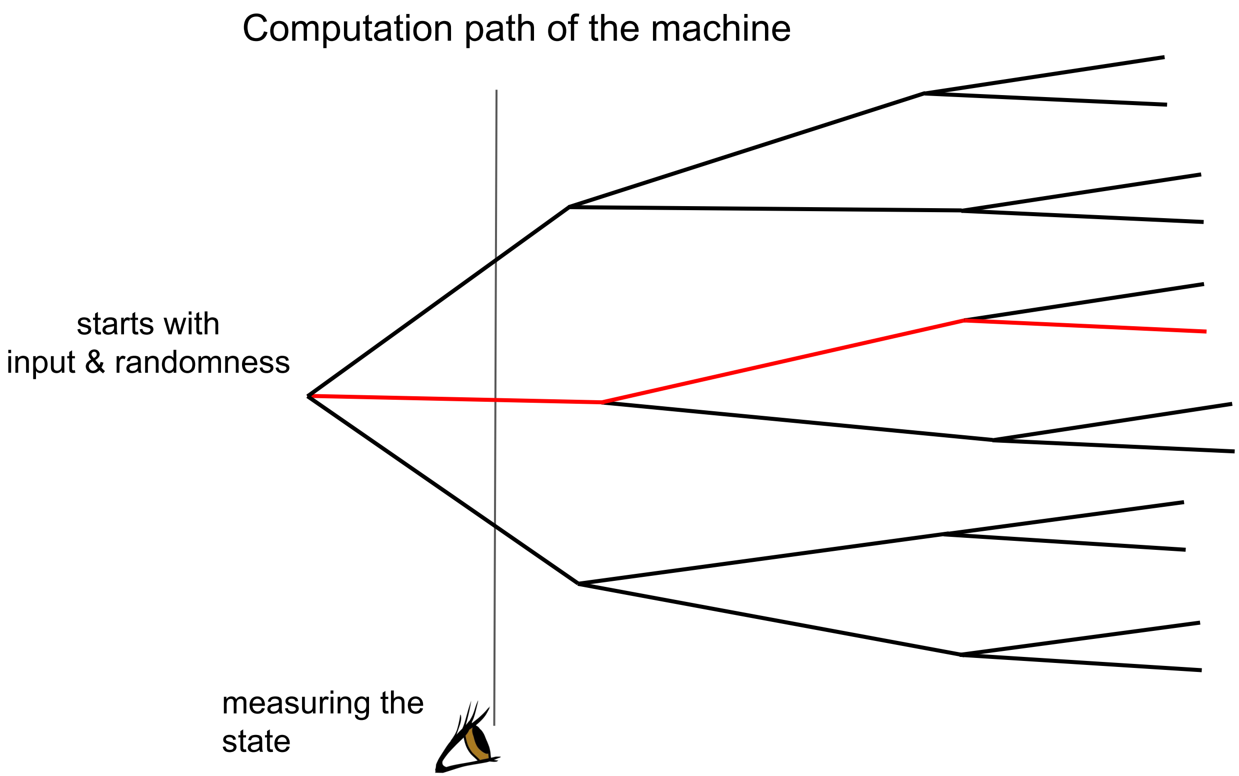
quantum poly-time adversaries can break DLog assumption (Shor's algorithm)

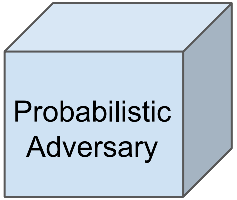
Probabilistic vs Quantum machines





Probabilistic vs Quantum machines





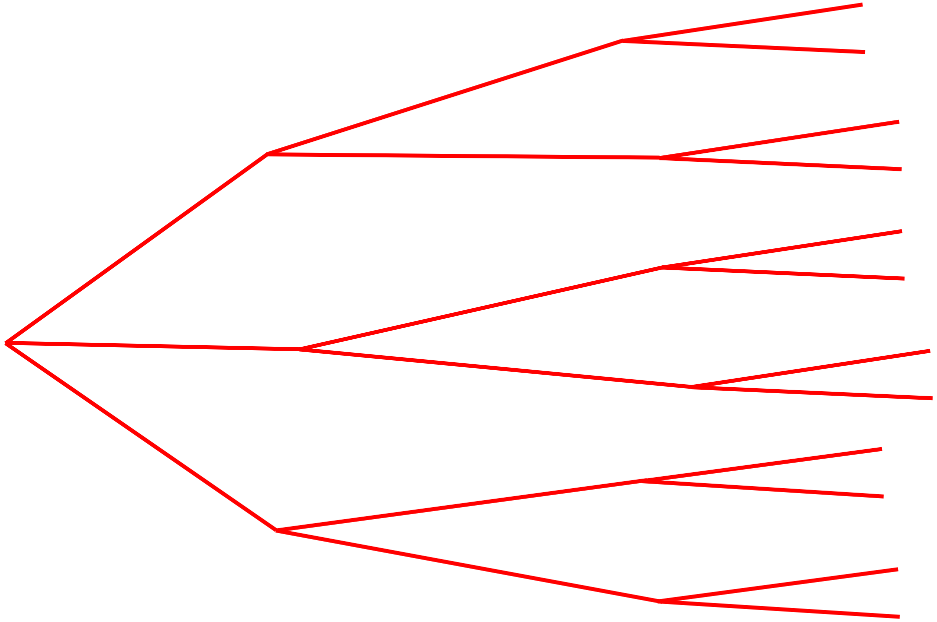
Probabilistic behaviour:

- Single path
- No measurement effect

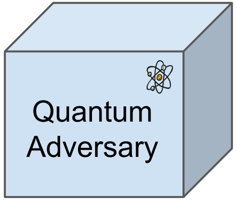
Probabilistic vs Quantum machines

Computation path of the machine

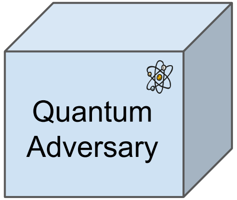
starts with input

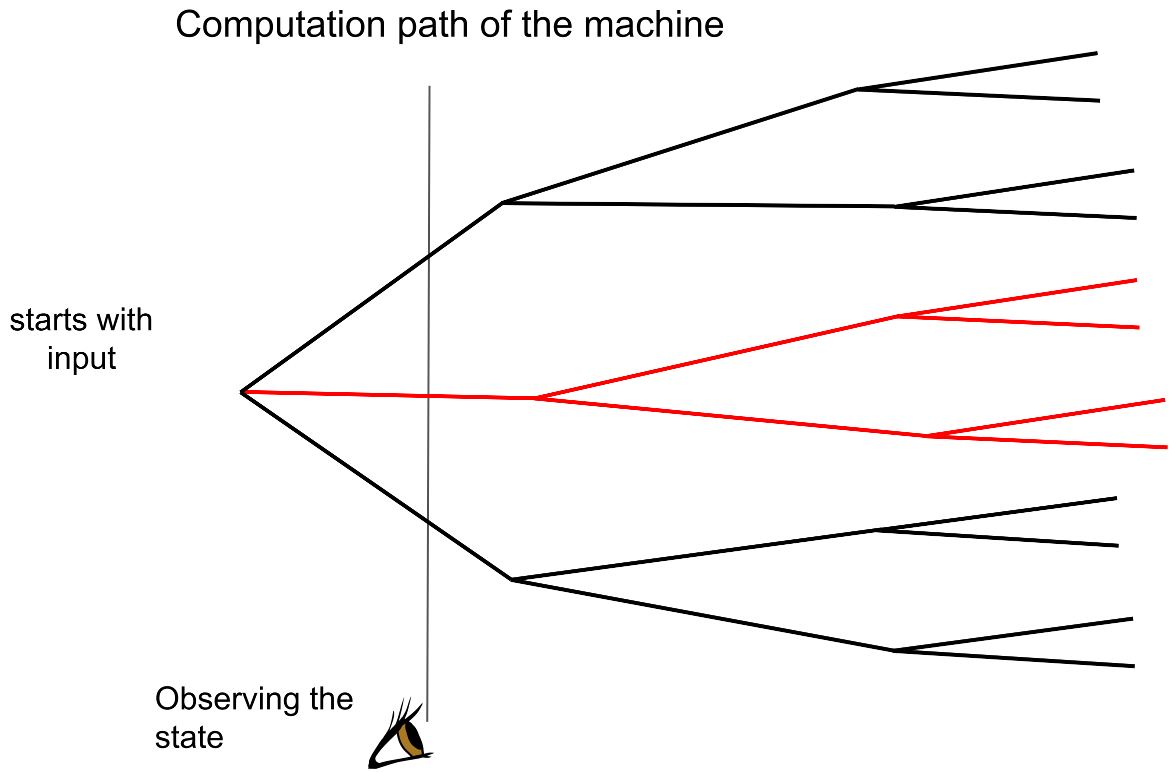


superposition state



Probabilistic vs Quantum machines

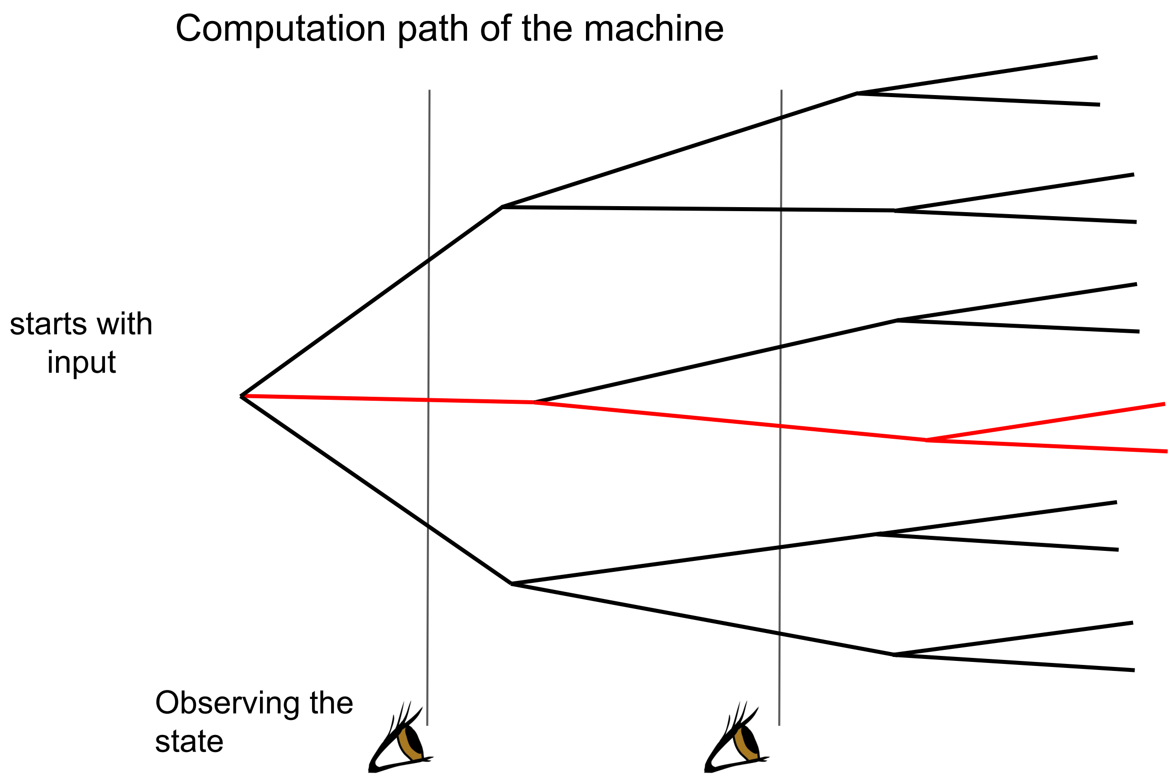


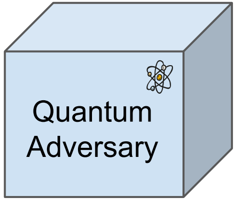


Quantum behaviour:

- Multiple paths in superposition
- Collapsing

Probabilistic vs Quantum machines





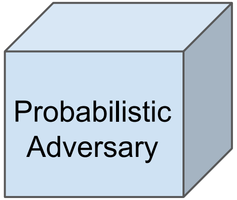
Quantum behaviour:

- Multiple paths in superposition
- Collapsing

Cryptographic Impacts of Quantum Computation

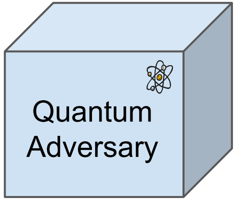
Probabilistic behaviour:

- Single path
- No collapse



Quantum behaviour:

- Multiple paths
- Collapsing



Any proof that uses the probabilistic behaviour of the adversary becomes invalid and must be revised
(in the quantum setting)

Our contributions

Analysis of two cryptographic tools against quantum adversaries:

- [DFPS23]: the Fiat-Shamir transform with aborts (revision of the proof)
  we thoroughly analyzed its security, correctness, and runtime

- [DFS24]: an LWE knowledge assumption (breaking the assumption)
  we demonstrated how to obliviously sample LWE instances
  in poly-time

[DFPS23]: A detailed analysis of Fiat-Shamir with aborts, J. Devevey, P. Fallahpour, A. Passelègue, D. Stehlé, CRYPTO'23
[DFS24]: Quantum Oblivious LWE Sampling and Insecurity of Standard Model Lattice-Based SNARKs, T. D. Alazard, P. Fallahpour, D. Stehlé, STOC'24

Outline

- Introduction
- Knowledge Assumption
- Fiat-Shamir Transform
- Conclusion

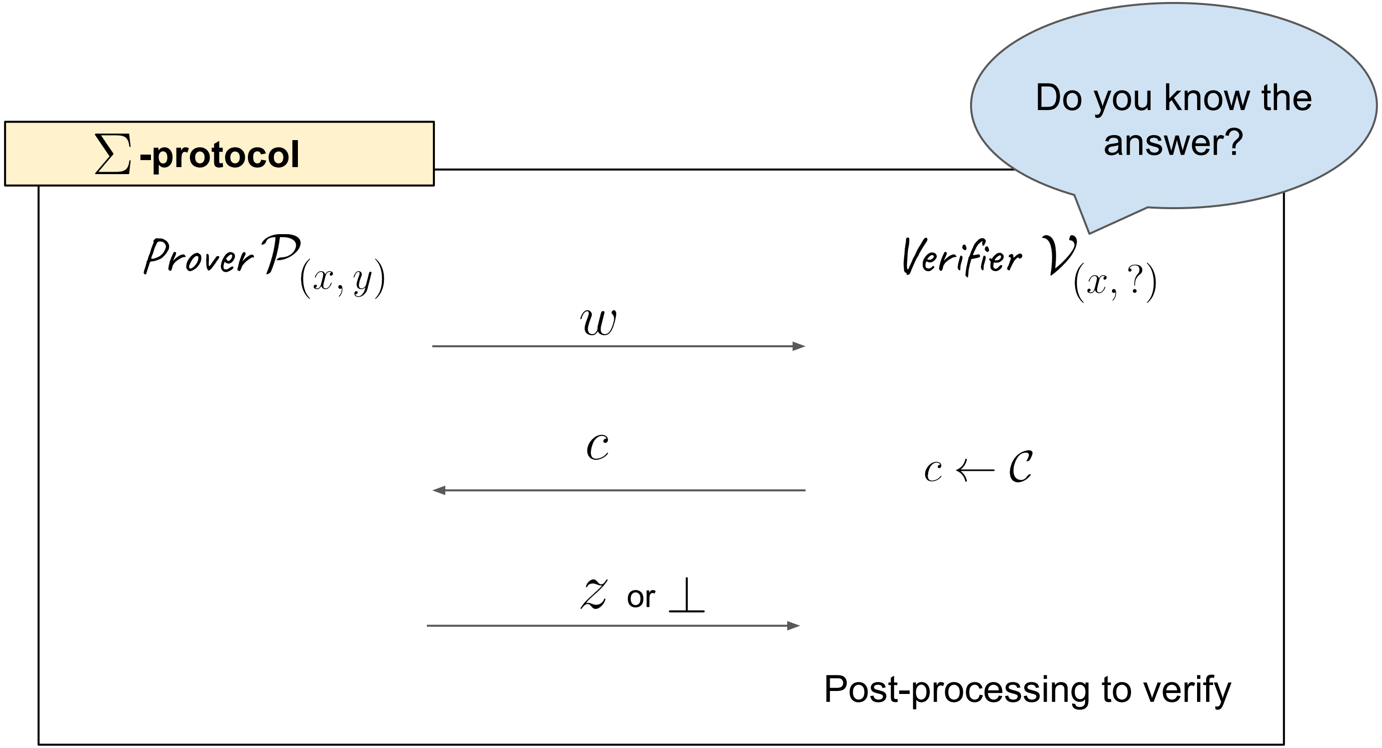
Fiat-Shamir in practice

One of the main paradigms to construct practical signature schemes is the Fiat-Shamir transform

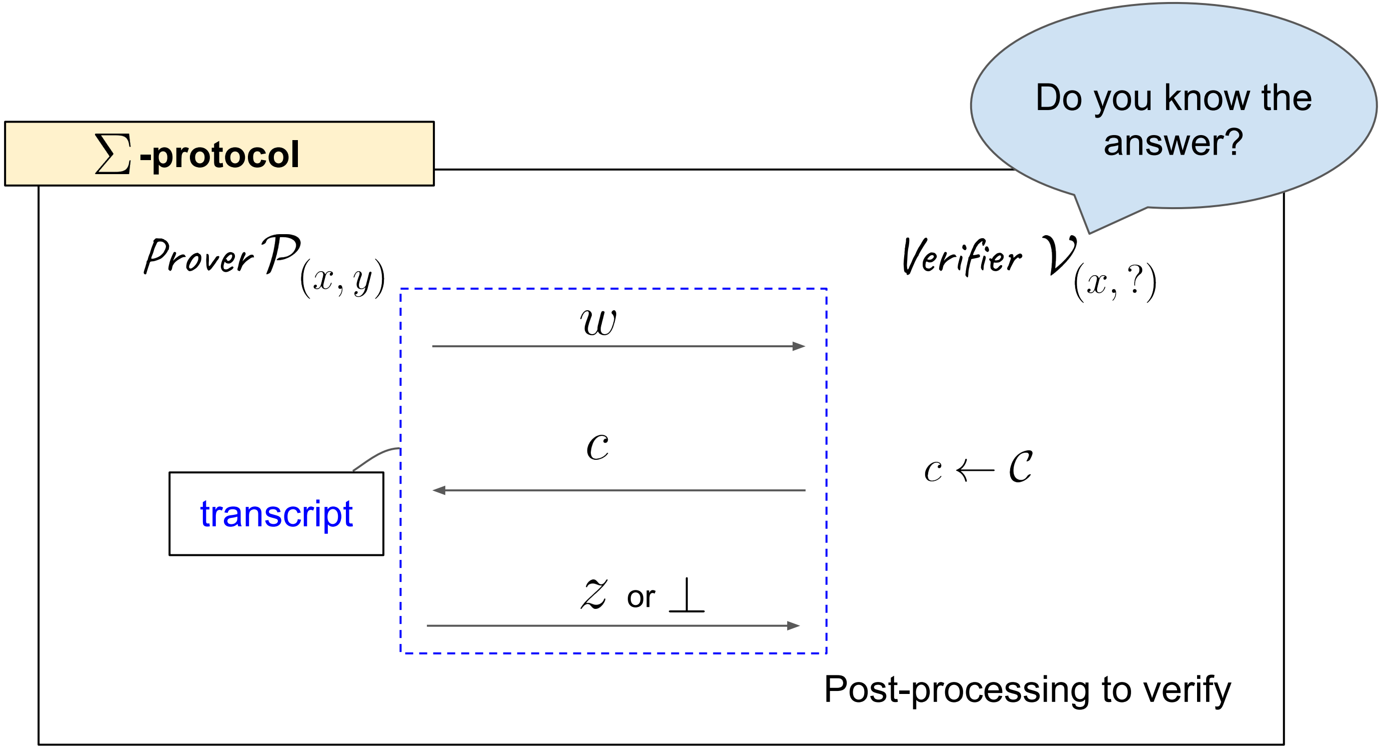
Some examples:

- Schnorr's signature based on the DLog Problem
- Lyubashevsky's signature based on the Short Integer Solution (SIS) or Learning with errors (LWE) problems
- Dilithium signature is a Fiat-Shamir-based signature that won the NIST competition for post-quantum secure signatures

Σ-protocol



Σ-protocol



Σ-protocol

Σ-protocol

Prover $\mathcal{P}_{(x,y)}$ | Verifier $\mathcal{V}_{(x,?)}$

$\xrightarrow{\quad w \quad}$

$\xleftarrow{\quad c \quad}$ $c \leftarrow \mathcal{C}$

$\xrightarrow{\quad z \text{ or } \perp \quad}$

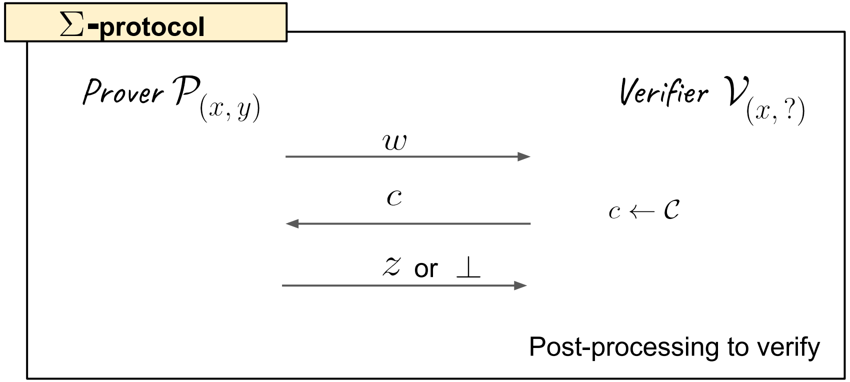
Post-processing to verify

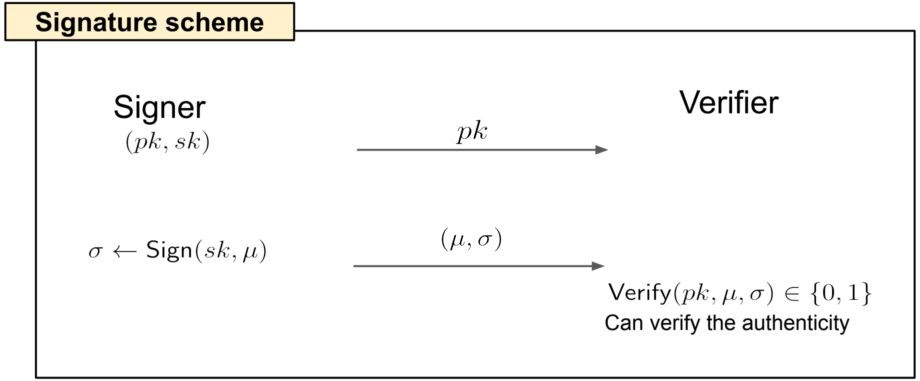
Soundness: $\mathcal{V}$ is not convinced when $\mathcal{P}$ does not know y

Zero-knowledge: $\mathcal{V}$ learns nothing beyond the fact that $\mathcal{P}$ knows y

$$\exists\, \text{PPT Sim} : \; \text{Sim}\,(x,c) \;\approx_{stat}\; (w,c,z)$$
conditioned on $z \neq \perp$

Fiat-Shamir transform with aborts (FSwA)





Soundness: $\mathcal{V}$ is not convinced when $\mathcal{P}$ does not know y

Zero-knowledge: $\mathcal{V}$ learns nothing beyond the fact that $\mathcal{P}$ knows y

$$\exists\, \text{PPT Sim} : \ \text{Sim}\,(x, c) \approx_{stat} (w, c, z)$$
conditioned on $z \neq \perp$

Fiat-Shamir transform with aborts (FSwA)

Σ-protocol

Prover $\mathcal{P}_{(x,y)}$ — Verifier $\mathcal{V}_{(x,?)}$

$\xrightarrow{w}$

$\xleftarrow{c}$ $\quad c \leftarrow \mathcal{C}$

$\xrightarrow{z \text{ or } \perp}$

Post-processing to verify

Soundness: $\mathcal{V}$ is not convinced when $\mathcal{P}$ does not know y

Zero-knowledge: $\mathcal{V}$ learns nothing beyond the fact that $\mathcal{P}$ knows y

$$\exists \, \text{PPT Sim}: \ \text{Sim}\,(x, c) \approx_{stat} (w, c, z)$$

conditioned on $z \neq \perp$

Signature scheme

Signer $(pk, sk) := (x, y)$ — Verifier

$\xrightarrow{pk}$

repeat B **times**

$w \leftarrow \mathcal{P}$

$c := H(w, \mu)$

$z \leftarrow \mathcal{P}(w, c)$

break when $z \neq \perp$

If all aborted

$\sigma := (\perp, \perp)$

else

$\sigma := (w, z)$

$\xrightarrow{(\mu, \sigma)}$

Post-processing to verify the signature

H : is a hash function, e.g., SHA-3

Fiat-Shamir transform with aborts (FSwA)

Σ-protocol

Prover $\mathcal{P}_{(x,y)}$ — Verifier $\mathcal{V}_{(x,?)}$

$\xrightarrow{\;w\;}$

$\xleftarrow{\;c\;}$ $\quad c \leftarrow \mathcal{C}$

$\xrightarrow{\;z \text{ or } \perp\;}$

Post-processing to verify

Signature scheme

Signer $(pk, sk) := (x, y)$ $\xrightarrow{\;pk\;}$ Verifier

repeat B **times**

$w \leftarrow \mathcal{P}$

$c := H(w, \mu)$

$z \leftarrow \mathcal{P}(w, c)$

break when $z \neq \perp$

If all aborted

$\sigma := (\perp, \perp)$ $\xrightarrow{\;(\mu, \sigma)\;}$

else

$\sigma := (w, z)$

Post-processing to verify the signature

Soundness: $\mathcal{V}$ is not convinced when $\mathcal{P}$ does not know y

Zero-knowledge: $\mathcal{V}$ learns nothing beyond the fact that $\mathcal{P}$ knows y

$$\exists \text{ PPT Sim}: \ \text{Sim}(x, c) \approx_{stat} (w, c, z)$$

conditioned on $z \neq \perp$

in the security proof we assume that H is a **random function/oracle** to which both parties have oracle access

Fiat-Shamir transform with aborts (FSwA)

Σ-protocol

Prover $\mathcal{P}_{(x,y)}$ — Verifier $\mathcal{V}_{(x,?)}$

w →

← c — $c \leftarrow \mathcal{C}$

z or $\perp$ →

Post-processing to verify

Signature scheme

Signer $(pk, sk) := (x, y)$ — Verifier

pk →

repeat B **times**

$w \leftarrow \mathcal{P}$

$c := H(w, \mu)$

$z \leftarrow \mathcal{P}(w, c)$

break when $z \neq \perp$

If all aborted

$\sigma := (\perp, \perp)$

else

$\sigma := (w, z)$

(μ, σ) →

Post-processing to verify the signature

We expect that

Soundness $\Longrightarrow$ NMA security

Zero-knowledge $\Longrightarrow$ CMA $\leq$ NMA

Similar to CMA, except that the adversary is not allowed to ask for any signatures

Fiat-Shamir transform with aborts (FSwA)

Σ-protocol

Prover $\mathcal{P}_{(x,y)}$ Verifier $\mathcal{V}_{(x,?)}$

w →

c ← $c \leftarrow \mathcal{C}$

z or $\perp$ →

Post-processing to verify

Signature scheme

Signer $(pk, sk) := (x, y)$ Verifier

pk →

repeat B **times**

$w \leftarrow \mathcal{P}$

$c := H(w, \mu)$

$z \leftarrow \mathcal{P}(w, c)$

break when $z \neq \perp$

If all aborted

$\sigma := (\perp, \perp)$

else

$\sigma := (w, z)$

(μ, σ) →

Post-processing to verify the signature

We expect that

~~**Soundness** $\Longrightarrow$ NMA security~~

Zero-knowledge $\Longrightarrow$ CMA $\leq$ NMA

Similar to CMA, except that the adversary is not allowed to ask for any signatures

Our contribution: a detailed and correct proof of

$$\text{Zero-knowledge} \implies \text{CMA} \leq \text{NMA}$$

In the process we also analyze the runtime and correctness.

All previous proofs are flawed
(even in the classical setting)

Roadmap for the CMA-to-NMA reduction

Explain the proof of [KLS18] → Point out the flaws → Explain how to fix the flaws

They claim post-quantum security

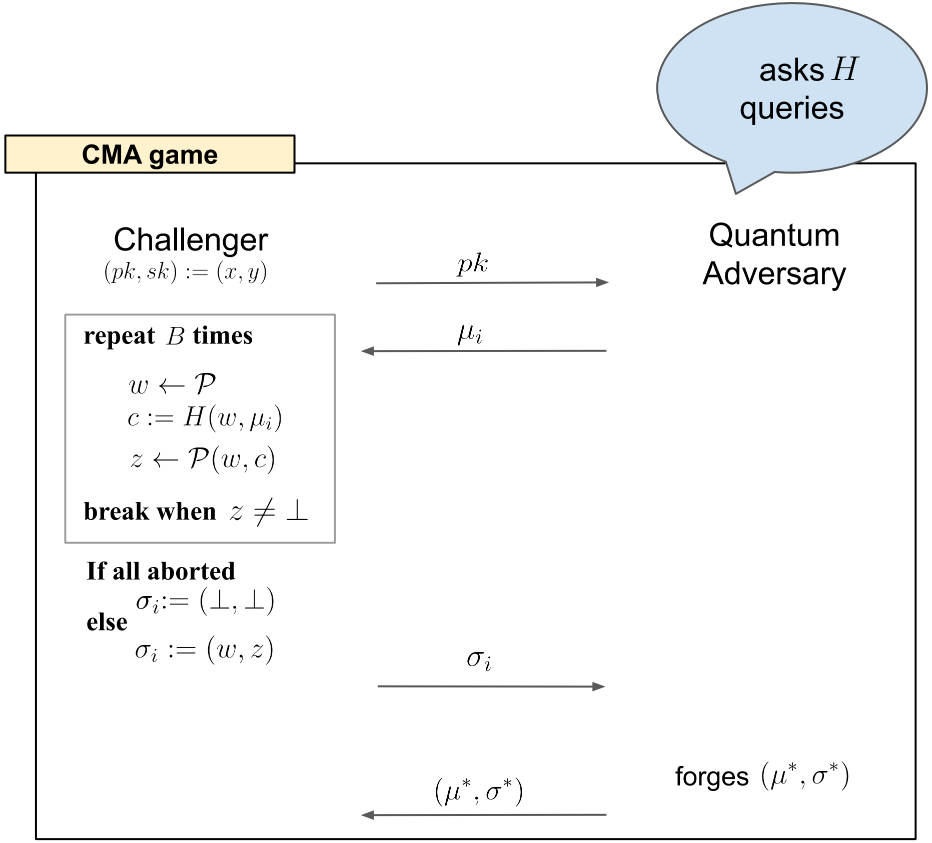
[KLS18]: E. Kiltz, V. Lyubashevsky, C. Schaffner, Eurocrypt'18

Roadmap for the CMA-to-NMA reduction

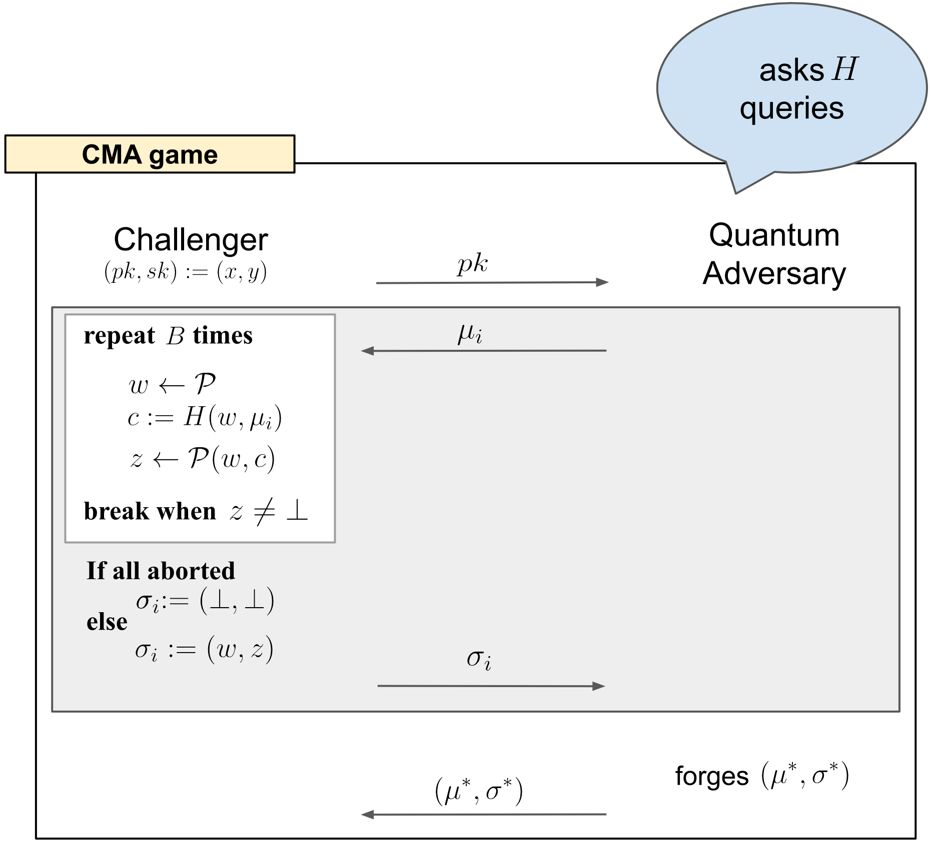
Explain the proof of [KLS18] → Point out the flaws → Explain how to fix the flaws

[KLS18]: E. Kiltz, V. Lyubashevsky, C. Schaffner, Eurocrypt'18

How to reduce CMA to NMA?



How to reduce CMA to NMA?



How to reduce CMA to NMA?

Goal: How to fake the signatures without having $sk := y$, consistently with H?

repeat B **times**

$w \leftarrow \mathcal{P}$

$c := H(w, \mu_i)$

$z \leftarrow \mathcal{P}(w, c)$

break when $z \neq \perp$

$\xleftarrow{\mu_i}$

If all aborted

$\sigma_i := (\perp, \perp)$

else

$\sigma_i := (w, z)$

$\xrightarrow{\sigma_i}$

NMA game

asks H queries

Challenger

$(pk, sk) := (x, y)$

$\xrightarrow{pk}$

Quantum Adversary

forges (μ^*, σ^*)

$\xleftarrow{(\mu^*, \sigma^*)}$

[KLS18] analysis of FSwA

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Game 0

repeat B **times**

$w \leftarrow \mathcal{P}$
$c := H(w, \mu)$
$z \leftarrow \mathcal{P}(w, c)$

break when $z \neq \perp$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

[KLS18] analysis of FSwA

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Game 0

repeat B **times**

$$w \leftarrow \mathcal{P}$$
$$c := H(w, \mu)$$
$$z \leftarrow \mathcal{P}(w, c)$$

break when $z \neq \perp$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

=

Game 0

$$w_1 \leftarrow \mathcal{P}$$
$$c_1 := H(w_1, \mu)$$
$$z_1 \leftarrow \mathcal{P}(w_1, c_1)$$

$$w_2 \leftarrow \mathcal{P}$$
$$c_2 := H(w_2, \mu)$$
$$z_2 \leftarrow \mathcal{P}(w_2, c_2)$$

$$\vdots$$

$$w_B \leftarrow \mathcal{P}$$
$$c_B := H(w_B, \mu)$$
$$z_B \leftarrow \mathcal{P}(w_B, c_B)$$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

[KLS18] analysis of FSwA

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Game 0

repeat B **times**

$w \leftarrow \mathcal{P}$
$c := H(w, \mu)$
$z \leftarrow \mathcal{P}(w, c)$

break when $z \neq \perp$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

$=$

Game 0

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$\boxed{\begin{array}{l} w_2 \leftarrow \mathcal{P} \\ c_2 := H(w_2, \mu) \\ z_2 \leftarrow \mathcal{P}(w_2, c_2) \end{array}}$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

First non-aborting round, e.g., for $\kappa = 2$

[KLS18] analysis of FSwA

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Game 0

repeat B **times**

$w \leftarrow \mathcal{P}$
$c := H(w, \mu)$
$z \leftarrow \mathcal{P}(w, c)$

break when $z \neq \perp$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

$=$

Game 0

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$w_2 \leftarrow \mathcal{P}$
$c_2 := H(w_2, \mu)$
$z_2 \leftarrow \mathcal{P}(w_2, c_2)$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

First non-aborting round, e.g., for $\kappa = 2$

if $\kappa \leq B$ simulate & reprogram $\rightarrow$

Game 1

$c_\mu \leftarrow \mathcal{C}$
$(w_\mu, z_\mu) \leftarrow \mathsf{Sim}(pk, c_\mu)$
$H(w_\mu, \mu) := c_\mu$

[KLS18] analysis of FSwA

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Game 0

repeat B **times**

$w \leftarrow \mathcal{P}$
$c := H(w, \mu)$
$z \leftarrow \mathcal{P}(w, c)$

break when $z \neq \perp$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

First non-aborting round, e.g., for $\kappa = 2$

$=$

Game 0

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$\boxed{\begin{array}{l} w_2 \leftarrow \mathcal{P} \\ c_2 := H(w_2, \mu) \\ z_2 \leftarrow \mathcal{P}(w_2, c_2) \end{array}}$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

if $\kappa \leq B$ simulate & reprogram $\rightarrow$

Game 1

$\boxed{\begin{array}{l} c_\mu \leftarrow \mathcal{C} \\ (w_\mu, z_\mu) \leftarrow \mathsf{Sim}(pk, c_\mu) \\ H(w_\mu, \mu) := c_\mu \end{array}}$

$H(w, \mu)$:
if $w = w_\mu$
return c_μ
else use the random function honestly

[KLS18] analysis of FSwA

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Game 0

repeat B **times**

$w \leftarrow \mathcal{P}$
$c := H(w, \mu)$
$z \leftarrow \mathcal{P}(w, c)$

break when $z \neq \perp$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

$=$

Game 0

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$w_2 \leftarrow \mathcal{P}$
$c_2 := H(w_2, \mu)$
$z_2 \leftarrow \mathcal{P}(w_2, c_2)$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

First non-aborting round, e.g., for $\kappa = 2$

simulate & reprogram $\rightarrow$

Game 1

if $\kappa \leq B$
$c_\mu \leftarrow \mathcal{C}$
$(w_\mu, z_\mu) \leftarrow \mathsf{Sim}(pk, c_\mu)$
$H(w_\mu, \mu) := c_\mu$
else
$(w_\mu, z_\mu) := (\perp, \perp)$

$H(w, \mu)$:
if $w = w_\mu$
return c_μ
else use the random function honestly

[KLS18] analysis of FSwA

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Game 0

repeat B **times**

$w \leftarrow \mathcal{P}$
$c := H(w, \mu)$
$z \leftarrow \mathcal{P}(w, c)$

break when $z \neq \perp$

If all aborted $\sigma := (\perp, \perp)$
else $\sigma := (w, z)$

$=$

Game 0

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$w_2 \leftarrow \mathcal{P}$
$c_2 := H(w_2, \mu)$
$z_2 \leftarrow \mathcal{P}(w_2, c_2)$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted $\sigma := (\perp, \perp)$
else $\sigma := (w, z)$

First non-aborting round, e.g., for $\kappa = 2$

simulate & reprogram

Game 1

if $\kappa \leq B$
$c_\mu \leftarrow \mathcal{C}$
$(w_\mu, z_\mu) \leftarrow \mathsf{Sim}(pk, c_\mu)$
$H(w_\mu, \mu) := c_\mu$

else
$(w_\mu, z_\mu) := (\perp, \perp)$

$H(w, \mu)$:
if $w = w_\mu$
return c_μ
else use the random function honestly

It is flawed!

[KLS18] analysis of FSwA

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Flaw A

If z is det. with respect to w and c, and if w leads to $z = \perp$, then w will never be part of a signature.
It is different in Game 1.

Game 0

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$w_2 \leftarrow \mathcal{P}$
$c_2 := H(w_2, \mu)$
$z_2 \leftarrow \mathcal{P}(w_2, c_2)$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

simulate & reprogram $\rightarrow$

Game 1

if $\kappa \leq B$
$c_\mu \leftarrow \mathcal{C}$
$(w_\mu, z_\mu) \leftarrow \mathsf{Sim}(pk, c_\mu)$
$H(w_\mu, \mu) := c_\mu$
else
$(w_\mu, z_\mu) := (\perp, \perp)$

$H(w, \mu)$:
if $w = w_\mu$
return c_μ
else use the random function honestly

[KLS18] analysis of FSwA

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Flaw A

If z is det. with respect to w and c, and if w leads to $z = \perp$, then w will never be part of a signature.
It is different in Game 1.

Game 0

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$w_2 \leftarrow \mathcal{P}$
$c_2 := H(w_2, \mu)$
$z_2 \leftarrow \mathcal{P}(w_2, c_2)$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

simulate & reprogram

Game 1

if $\kappa \leq B$
$c_\mu \leftarrow \mathcal{C}$
$(w_\mu, z_\mu) \leftarrow \mathsf{Sim}(pk, c_\mu)$
$H(w_\mu, \mu) := c_\mu$
else
$(w_\mu, z_\mu) := (\perp, \perp)$

$H(w, \mu)$:
if $w = w_\mu$
return c_μ
else use the random function honestly

Flaw B

an estimation of κ is required, i.e, a correctness and runtime analysis

[KLS18] analysis of FSwA

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Flaw A

If z is det. with respect to w and c, and if w leads to $z = \perp$, then w will never be part of a signature.
It is different in Game 1.

Game 0

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$w_2 \leftarrow \mathcal{P}$
$c_2 := H(w_2, \mu)$
$z_2 \leftarrow \mathcal{P}(w_2, c_2)$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

simulate & reprogram $\rightarrow$

Game 1

if $\kappa \leq B$
$c_\mu \leftarrow \mathcal{C}$
$(w_\mu, z_\mu) \leftarrow \mathsf{Sim}(pk, c_\mu)$
$H(w_\mu, \mu) := c_\mu$
else
$(w_\mu, z_\mu) := (\perp, \perp)$

$H(w, \mu)$:
if $w = w_\mu$
return c_μ
else use the random function honestly

Flaw B

an estimation of κ is required, i.e, a correctness and runtime analysis

Flaw C

adversary's **quantum** access to transcripts is neglected

Roadmap for the CMA-to-NMA reduction

Explain the proof of [KLS18] → Point out the flaws → Explain how to fix the flaws

[KLS18]: E. Kiltz, V. Lyubashevsky, C. Schaffner, Eurocrypt'18

Our fix - a middle game

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Game 0

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$\boxed{w_2 \leftarrow \mathcal{P} \quad c_2 := H(w_2, \mu) \quad z_2 \leftarrow \mathcal{P}(w_2, c_2)}$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

handling collisions →

Game 0.5

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$\boxed{w_2 \leftarrow \mathcal{P} \quad c_2 := H(w_2, \mu) \quad z_2 \leftarrow \mathcal{P}(w_2, c_2)}$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

if $w_i = w_j$ **abort**

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

Our fix - a middle game

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Game 0

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$\boxed{\begin{array}{l} w_2 \leftarrow \mathcal{P} \\ c_2 := H(w_2, \mu) \\ z_2 \leftarrow \mathcal{P}(w_2, c_2) \end{array}}$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

$\xrightarrow[\text{collisions}]{\text{handling}}$

Game 0.5

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$\boxed{\begin{array}{l} w_2 \leftarrow \mathcal{P} \\ c_2 := H(w_2, \mu) \\ z_2 \leftarrow \mathcal{P}(w_2, c_2) \end{array}}$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B :=$
$z_B \leftarrow$

If all abort
$\sigma := ($
else
$\sigma := (w, z)$

Flaw A resolved

if $w_i = w_j$ **abort**
$\mathbb{P}[w_i = w_j] \leq B^2 \cdot 2^{-\alpha-1}$

Our fix - a stronger simulator

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Game 0

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$w_2 \leftarrow \mathcal{P}$
$c_2 := H(w_2, \mu)$
$z_2 \leftarrow \mathcal{P}(w_2, c_2)$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

handling collisions →

Game 0.5

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$w_2 \leftarrow \mathcal{P}$
$c_2 := H(w_2, \mu)$
$z_2 \leftarrow \mathcal{P}(w_2, c_2)$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H$
$z_B \leftarrow$

If all abort
$\sigma := ($
else
$\sigma := (w, z)$

Flaw A resolved

if $w_i = w_j$ **abort**
$\mathbb{P}[w_i = w_j] \leq B^2 \cdot 2^{-\alpha-1}$

simulate & reprogram →
all transcripts even the aborting ones

Game 1

$c_{\mu,1} \leftarrow \mathcal{C}$
$(w_{\mu,1}, z_{\mu,1}) \leftarrow \mathsf{Sim}(pk, c_{\mu,1})$
$H(w_{\mu,1}) := c_{\mu,1}$

$c_{\mu,2} \leftarrow \mathcal{C}$
$(w_{\mu,2}, z_{\mu,2}) \leftarrow \mathsf{Sim}(pk, c_{\mu,2})$
$H(w_{\mu,2}) := c_{\mu,2}$

$\vdots$

$c_{\mu,B} \leftarrow \mathcal{C}$
$(w_{\mu,B}, z_{\mu,B}) \leftarrow \mathsf{Sim}(pk, c_{\mu,B})$
$H(w_{\mu,B}) := c_{\mu,B}$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

Our fix - a stronger simulator

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Game 0

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$w_2 \leftarrow \mathcal{P}$
$c_2 := H(w_2, \mu)$
$z_2 \leftarrow \mathcal{P}(w_2, c_2)$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

handling collisions →

Game 0.5

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$w_2 \leftarrow \mathcal{P}$
$c_2 := H(w_2, \mu)$
$z_2 \leftarrow \mathcal{P}(w_2, c_2)$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H$
$z_B \leftarrow \mathcal{P}$

Flaw A resolved

if $w_i = w_j$ **abort**
$\mathbb{P}[w_i = w_j] \leq B^2 \cdot 2^{-\alpha-1}$

If all abort
$\sigma :=$
else
$\sigma := (w, z)$

simulate & reprogram →
all transcripts even the aborting ones

Game 1

$c_{\mu,1} \leftarrow \mathcal{C}$
$(w_{\mu,1}, z_{\mu,1}) \leftarrow \mathsf{Sim}(pk, c_{\mu,1})$
$H(w_{\mu,1}) := c_{\mu,1}$

$c_{\mu,2} \leftarrow \mathcal{C}$
$(w_{\mu,2}, z_{\mu,2}) \leftarrow \mathsf{Sim}(pk, c_{\mu,2})$
$H(w_{\mu,2}) := c_{\mu,2}$

$\vdots$

$c_{\mu,B} \leftarrow \mathcal{C}$
$(w_{\mu,B}, z_{\mu,B}) \leftarrow \mathsf{Sim}(pk, c_{\mu,B})$
$H(w_{\mu,B}) := c_{\mu,B}$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

$H(w, \mu)$:
if $w = w_{\mu,j}$
return $c_{\mu,j}$
else use the random function honestly

Our fix - a stronger simulator

Flaw B resolved

By simulating all transcripts, there is no need to approximate κ

We provide such a stronger simulator for Lyubashevsky Σ-protocol

Goal: How to fake the signatures without having $sk := y$, consistently with H?

Game 0

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$w_2 \leftarrow \mathcal{P}$
$c_2 := H(w_2, \mu)$
$z_2 \leftarrow \mathcal{P}(w_2, c_2)$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

$\xrightarrow[\text{collisions}]{\text{handling}}$

Game 0.5

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$w_2 \leftarrow \mathcal{P}$
$c_2 := H(w_2, \mu)$
$z_2 \leftarrow \mathcal{P}(w_2, c_2)$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H$
$z_B \leftarrow \mathcal{P}$

If all abort
$\sigma :=$
else
$\sigma := (w, z)$

Flaw A resolved

if $w_i = w_j$ **abort**
$\mathbb{P}[w_i = w_j] \leq B^2 \cdot 2^{-\alpha-1}$

simulate & reprogram $\longrightarrow$
all transcripts even the aborting ones

Game 1

$c_{\mu,1} \leftarrow \mathcal{C}$
$(w_{\mu,1}, z_{\mu,1}) \leftarrow \mathsf{Sim}(pk, c_{\mu,1})$
$H(w_{\mu,1}) := c_{\mu,1}$

$c_{\mu,2} \leftarrow \mathcal{C}$
$(w_{\mu,2}, z_{\mu,2}) \leftarrow \mathsf{Sim}(pk, c_{\mu,2})$
$H(w_{\mu,2}) := c_{\mu,2}$

$\vdots$

$c_{\mu,B} \leftarrow \mathcal{C}$
$(w_{\mu,B}, z_{\mu,B}) \leftarrow \mathsf{Sim}(pk, c_{\mu,B})$
$H(w_{\mu,B}) := c_{\mu,B}$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

$H(w, \mu)$:
if $w = w_{\mu,j}$
return $c_{\mu,j}$
else use the random function honestly

Our fix - flaw C

Goal: How to fake the signatures without having $sk := y$, consistently with H?

It suffices that:

$$\sum_k \gamma_k \left| \begin{matrix} w_k \\ c_k \\ z_k \end{matrix} : \text{REAL} \right\rangle \approx_{tr} \sum_k \gamma_k \left| \begin{matrix} w_k \\ c_k \\ z_k \end{matrix} : \text{SIM} \right\rangle$$

Game 0.5

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$\boxed{\begin{array}{l} w_2 \leftarrow \mathcal{P} \\ c_2 := H(w_2, \mu) \\ z_2 \leftarrow \mathcal{P}(w_2, c_2) \end{array}}$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted $\sigma := (\perp, \perp)$
else $\sigma := (w, z)$

simulate & reprogram $\longrightarrow$

all transcripts even the aborting ones

$$\overbrace{\text{Game 1}}$$

$\boxed{\begin{array}{l} c_{\mu,1} \leftarrow \mathcal{C} \\ (w_{\mu,1}, z_{\mu,1}) \leftarrow \mathsf{Sim}(pk, c_{\mu,1}) \\ H(w_{\mu,1}) := c_{\mu,1} \end{array}}$

$\boxed{\begin{array}{l} c_{\mu,2} \leftarrow \mathcal{C} \\ (w_{\mu,2}, z_{\mu,2}) \leftarrow \mathsf{Sim}(pk, c_{\mu,2}) \\ H(w_{\mu,2}) := c_{\mu,2} \end{array}}$

$\vdots$

$\boxed{\begin{array}{l} c_{\mu,B} \leftarrow \mathcal{C} \\ (w_{\mu,B}, z_{\mu,B}) \leftarrow \mathsf{Sim}(pk, c_{\mu,B}) \\ H(w_{\mu,B}) := c_{\mu,B} \end{array}}$

If all aborted $\sigma := (\perp, \perp)$
else $\sigma := (w, z)$

$H(w, \mu)$:
if $w = w_{\mu,j}$
return $c_{\mu,j}$

else use the random function honestly

[Zha12]: M. Zhandry, FOCS'12

Our fix - flaw C

Goal: How to fake the signatures without having $sk := y$, consistently with H?

[Zha12]:

$$f \approx_{stat} g \implies |f\rangle \approx_{tr} |g\rangle$$

It suffices that:

$$\sum_k \gamma_k \left| \begin{matrix} w_k \\ c_k \\ z_k \end{matrix} : \text{REAL} \right\rangle \approx_{tr} \sum_k \gamma_k \left| \begin{matrix} w_k \\ c_k \\ z_k \end{matrix} : \text{SIM} \right\rangle$$

Game 0.5

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$w_2 \leftarrow \mathcal{P}$
$c_2 := H(w_2, \mu)$
$z_2 \leftarrow \mathcal{P}(w_2, c_2)$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

simulate & reprogram

all transcripts even the aborting ones

Game 1

$c_{\mu,1} \leftarrow \mathcal{C}$
$(w_{\mu,1}, z_{\mu,1}) \leftarrow \text{Sim}(pk, c_{\mu,1})$
$H(w_{\mu,1}) := c_{\mu,1}$

$c_{\mu,2} \leftarrow \mathcal{C}$
$(w_{\mu,2}, z_{\mu,2}) \leftarrow \text{Sim}(pk, c_{\mu,2})$
$H(w_{\mu,2}) := c_{\mu,2}$

$\vdots$

$c_{\mu,B} \leftarrow \mathcal{C}$
$(w_{\mu,B}, z_{\mu,B}) \leftarrow \text{Sim}(pk, c_{\mu,B})$
$H(w_{\mu,B}) := c_{\mu,B}$

If all aborted
$\sigma := (\perp, \perp)$
else
$\sigma := (w, z)$

$H(w, \mu)$:
if $w = w_{\mu,j}$
return $c_{\mu,j}$
else use the random function honestly

[Zha12]: M. Zhandry, FOCS'12

Our fix - flaw C

Goal: How to fake the signatures without having $sk := y$, consistently with H?

[Zha12]:

$$f \approx_{stat} g \implies |f\rangle \approx_{tr} |g\rangle$$

Flaw C resolved

It suffices that:

$$\sum_k \gamma_k \left| \begin{matrix} w_k \\ c_k \\ z_k \end{matrix} : \text{REAL} \right\rangle \approx_{tr} \sum_k \gamma_k \left| \begin{matrix} w_k \\ c_k \\ z_k \end{matrix} : \text{SIM} \right\rangle$$

Game 0.5

$w_1 \leftarrow \mathcal{P}$
$c_1 := H(w_1, \mu)$
$z_1 = \perp$

$w_2 \leftarrow \mathcal{P}$
$c_2 := H(w_2, \mu)$
$z_2 \leftarrow \mathcal{P}(w_2, c_2)$

$\vdots$

$w_B \leftarrow \mathcal{P}$
$c_B := H(w_B, \mu)$
$z_B \leftarrow \mathcal{P}(w_B, c_B)$

If all aborted $\sigma := (\perp, \perp)$
else $\sigma := (w, z)$

simulate & reprogram →

all transcripts even the aborting ones

Game 1

$c_{\mu,1} \leftarrow \mathcal{C}$
$(w_{\mu,1}, z_{\mu,1}) \leftarrow \mathsf{Sim}(pk, c_{\mu,1})$
$H(w_{\mu,1}) := c_{\mu,1}$

$c_{\mu,2} \leftarrow \mathcal{C}$
$(w_{\mu,2}, z_{\mu,2}) \leftarrow \mathsf{Sim}(pk, c_{\mu,2})$
$H(w_{\mu,2}) := c_{\mu,2}$

$\vdots$

$c_{\mu,B} \leftarrow \mathcal{C}$
$(w_{\mu,B}, z_{\mu,B}) \leftarrow \mathsf{Sim}(pk, c_{\mu,B})$
$H(w_{\mu,B}) := c_{\mu,B}$

If all aborted $\sigma := (\perp, \perp)$
else $\sigma := (w, z)$

$H(w, \mu)$:
if $w = w_{\mu,j}$
return $c_{\mu,j}$
else use the random function honestly

Table of results

Analysis of $\mathsf{CMA} \leq \mathsf{NMA}$	Fixed proof of [KLS18]	Adaptive reprogramming (extension of [GHHM21])
Reduction loss	$2^{-\alpha/2} B Q_S Q_H + \varepsilon_{zk}^{1/2} B^{1/2} Q_H^{3/2}$	$2^{-\alpha/2} B Q_S Q_H^{1/2} + \varepsilon_{zk} B Q_S$
Runtime	$B Q_S Q_H$	$Q_H \log(B Q_S)$

Q_S : number of sign queries

Q_H : number of hash queries

ε_{zk} : zero-knowledge simulator error

α : min-entropy of commitments

B : upper bound for the number of trials in signing algorithm

[KLS18]: E. Kiltz, V. Lyubashevsky, C. Schaffner, Eurocrypt'18
[GHHM21]: A. B. Grilo, K. Hövelmanns, A. Hülsing, C. Majenz, Asiacrypt'21

Outline

- Introduction
- Knowledge Assumption
- Fiat-Shamir Transform
- Conclusion

LWE instance

An LWE instance:

- given $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- sample $(\mathbf{s}, \mathbf{e}) \sim \mathcal{U}(\mathbb{Z}_q^n) \times \chi_\sigma^m$
- return $\mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$

$\mathbf{s}$: **secret**
$\mathbf{e}$: **noise**

χ_σ^m: m-dimensional discrete Gaussian with standard deviation $\sigma > 0$

LWE instance

An LWE instance:

always tuned to be injective

- given $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- sample $(\mathbf{s}, \mathbf{e}) \sim \mathcal{U}(\mathbb{Z}_q^n) \times \chi_\sigma^m$
- return $\mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$

$\mathbf{s}$: secret
$\mathbf{e}$: noise

χ_σ^m: m-dimensional discrete Gaussian with standard deviation $\sigma > 0$

LWE instance

LWE problem: given $\mathbf{A}$ and $\mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$, find the secret

LWE assumption: when $\mathbf{A}$ is sampled uniformly, it is hard to find the secret

An LWE instance:

always tuned to be injective

- given $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- sample $(\mathbf{s}, \mathbf{e}) \sim \mathcal{U}(\mathbb{Z}_q^n) \times \chi_\sigma^m$
- return $\mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$

$\mathbf{s}$: secret
$\mathbf{e}$: noise

χ_σ^m: m-dimensional discrete Gaussian with standard deviation $\sigma > 0$

LWE sampler

An LWE instance **sampler**:

- given $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- sample $(\mathbf{s}, \mathbf{e}) \sim \mathcal{U}(\mathbb{Z}_q^n) \times \chi_\sigma^m$
- return $\mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$

χ_σ^m: m-dimensional discrete Gaussian with standard deviation $\sigma > 0$

LWE sampler

- Can we sample an LWE instance without knowing its secret?

*We call such a sampler **oblivious***

The naive sampler:

- given $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- sample $(\mathbf{s}, \mathbf{e}) \sim \mathcal{U}(\mathbb{Z}_q^n) \times \chi_\sigma^m$
- return $\mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$

χ_σ^m: m-dimensional discrete Gaussian with standard deviation $\sigma > 0$

LWE sampler

- Can we sample an LWE instance without knowing its secret?

*We call such a sampler **oblivious***

A candidate:

- sample $\mathbf{b} \sim \mathcal{U}(\mathbb{Z}_q^m)$
- return $\mathbf{b}$

It is far from the correct distribution

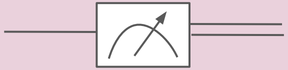
LWE sampler

- Can we sample an LWE instance without knowing its secret?

We call such a sampler ***oblivious***

The superposition sampler:

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n}\sum_{\mathbf{e}\in\mathbb{Z}_q^m}\sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$



It is not known how to build this state

LWE sampler

- Can we sample an LWE instance without knowing its secret?

*We call such a sampler **oblivious***

Another candidate?
We are not aware of any!

LWE Knowledge Assumption: there is no poly-time oblivious sampler for LWE

Used to analyze the security of several SNARK protocols [GMNO18, NYI+ 20, ISW21, SSEK22, CKKK23, GNSV23]

Our contribution: a quantum polynomial-time oblivious LWE sampler

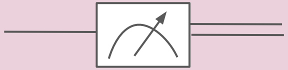
Invalidates the security analyses of the mentioned SNARKs in the context of quantum adversaries

LWE state

We use the framework of the superposition sampler

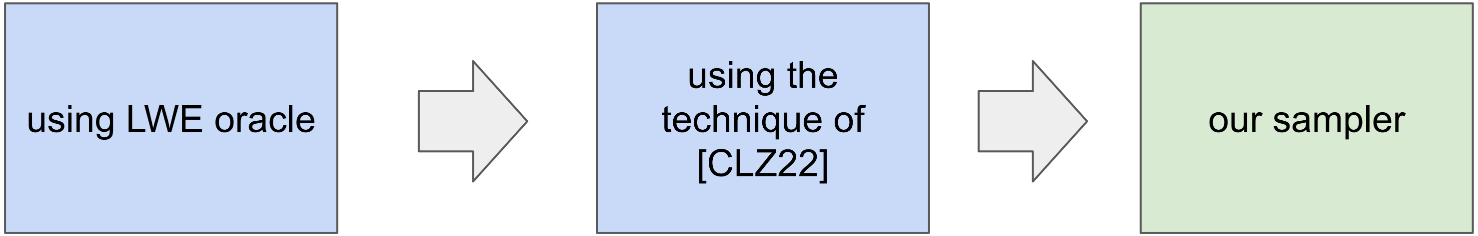
The superposition sampler:

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n}\sum_{\mathbf{e}\in\mathbb{Z}_q^m}\sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$



m-dimensional discrete Gaussian with standard deviation σ

Roadmap to LWE state

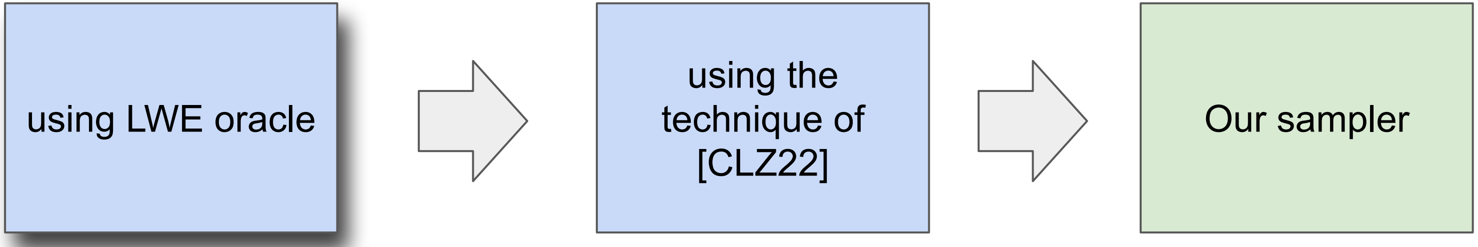


A toy example

The constraints on the parameters cannot be satisfied

[CLZ22]: Y. Chen, Q. Liu, M. Zhandry, Eurocrypt'22

Roadmap to LWE state



A toy example

The constraints on the parameters cannot be satisfied

[CLZ22]: Y. Chen, Q. Liu, M. Zhandry, Eurocrypt'22

LWE state with LWE oracle [Regev 05, SSTX 09]

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{e}\rangle$$

[Regev05]: O. Regev, STOC'05
[SSTX]: D. Stehlé, R. Steinfeld, K. Tanaka, K. Xagawa, Asiacrypt'09

LWE state with LWE oracle [Regev 05, SSTX 09]

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{e}\rangle$$

$$\longrightarrow \quad \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

[Regev05]: O. Regev, STOC’05
[SSTX]: D. Stehlé, R. Steinfeld, K. Tanaka, K. Xagawa, Asiacrypt’09

LWE state with LWE oracle [Regev 05, SSTX 09]

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{e}\rangle$$

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

Using an LWE solver

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}-\mathrm{solve}(\mathbf{A},\mathbf{A}\mathbf{s}+\mathbf{e})\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{0}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

[Regev05]: O. Regev, STOC’05
[SSTX]: D. Stehlé, R. Steinfeld, K. Tanaka, K. Xagawa, Asiacrypt’09

LWE state with LWE oracle [Regev 05, SSTX 09]

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{e}\rangle$$

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

Using an LWE solver

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}-\mathrm{solve}(\mathbf{A},\mathbf{A}\mathbf{s}+\mathbf{e})\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{0}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

we do not know how to do it in poly-time

[Regev05]: O. Regev, STOC’05
[SSTX]: D. Stehlé, R. Steinfeld, K. Tanaka, K. Xagawa, Asiacrypt’09

Roadmap to LWE state

using LWE oracle → using the technique of [CLZ22] → Our sampler

A toy example

The constraints on the parameters cannot be satisfied

[CLZ22]: Y. Chen, Q. Liu, M. Zhandry, Eurocrypt'22

LWE state with [CLZ22]

Let $\mathbf{A}$ be a single row $\mathbf{a}^T$

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{e\in\mathbb{Z}_q} \sqrt{\chi_\sigma}(e)|e\rangle$$

$$\longrightarrow \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{e\in\mathbb{Z}_q} \sqrt{\chi_\sigma}(e)|\mathbf{a}^T\mathbf{s}+e\rangle$$

Notation

$$|\psi_j\rangle \propto \sum_{e\in\mathbb{Z}_q} \sqrt{\chi_\sigma}(e)|j+e\rangle$$

"superposition of Gaussian distribution centered around j"

[CLZ22]: Y. Chen, Q. Liu, M. Zhandry, Eurocrypt'22

LWE state with [CLZ22]

Let $\mathbf{A}$ be a single row $\mathbf{a}^T$

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{e\in\mathbb{Z}_q} \sqrt{\chi_\sigma}(e)|e\rangle$$

$$\longrightarrow \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{e\in\mathbb{Z}_q} \sqrt{\chi_\sigma}(e)|\mathbf{a}^T\mathbf{s}+e\rangle$$

$$\propto \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes |\psi_{\mathbf{a}^T\mathbf{s}}\rangle$$

Notation

$$|\psi_j\rangle \propto \sum_{e\in\mathbb{Z}_q} \sqrt{\chi_\sigma}(e)|j+e\rangle$$

"superposition of Gaussian distribution centered around j"

[CLZ22]: Y. Chen, Q. Liu, M. Zhandry, Eurocrypt'22

LWE state with [CLZ22]

Let $\mathbf{A}$ has arbitrarily many rows

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m}(\mathbf{e})|\mathbf{e}\rangle$$

$$\longrightarrow \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m}(\mathbf{e})|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

$$\propto \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes |\psi_{\mathbf{a}_1^T\mathbf{s}}\rangle \otimes \cdots \otimes |\psi_{\mathbf{a}_m^T\mathbf{s}}\rangle$$

Notation

$$|\psi_j\rangle \propto \sum_{e\in\mathbb{Z}_q} \sqrt{\chi_\sigma}(e)|j+e\rangle$$

"superposition of Gaussian distribution centered around j"

[CLZ22]: Y. Chen, Q. Liu, M. Zhandry, Eurocrypt'22

LWE state with [CLZ22]

Let $\mathbf{A}$ has arbitrarily many rows

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{e}\rangle$$

$$\longrightarrow \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

$$\propto \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \underbrace{|\psi_{\mathbf{a}_1^T\mathbf{s}}\rangle \otimes \cdots \otimes |\psi_{\mathbf{a}_m^T\mathbf{s}}\rangle}_{\text{Extract } \mathbf{s} \text{ from these}}$$

Notation

$$|\psi_j\rangle \propto \sum_{e\in\mathbb{Z}_q} \sqrt{\chi_\sigma(e)}|j+e\rangle$$

"superposition of Gaussian distribution centered around j"

[CLZ22]: Y. Chen, Q. Liu, M. Zhandry, Eurocrypt'22

LWE state with [CLZ22]

Let $\mathbf{A}$ has arbitrarily many rows

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m}(\mathbf{e})|\mathbf{e}\rangle$$

$$\longrightarrow \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m}(\mathbf{e})|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

$$\propto \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \underbrace{|\psi_{\mathbf{a}_1^T\mathbf{s}}\rangle \otimes \cdots \otimes |\psi_{\mathbf{a}_m^T\mathbf{s}}\rangle}_{\text{Extract } \mathbf{s} \text{ from these}}$$

$$\longrightarrow \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{0}\rangle \otimes |\psi_{\mathbf{a}_1^T\mathbf{s}}\rangle \otimes \cdots \otimes |\psi_{\mathbf{a}_m^T\mathbf{s}}\rangle \quad \propto \quad \sum_{\mathbf{s}\in\mathbb{Z}_q^n} \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m}(\mathbf{e})|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

Notation

$$|\psi_j\rangle \propto \sum_{e\in\mathbb{Z}_q} \sqrt{\chi_\sigma}(e)|j+e\rangle$$

"superposition of Gaussian distribution centered around j"

[CLZ22]: Y. Chen, Q. Liu, M. Zhandry, Eurocrypt'22

LWE state with [CLZ22]

Let $\mathbf{A}$ has arbitrarily many rows

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m}(\mathbf{e})|\mathbf{e}\rangle$$

$$\longrightarrow \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m}(\mathbf{e})|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

$$\propto \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \underbrace{|\psi_{\mathbf{a}_1^T\mathbf{s}}\rangle \otimes \cdots \otimes |\psi_{\mathbf{a}_m^T\mathbf{s}}\rangle}_{\text{Extract } \mathbf{s} \text{ from these}}$$

$$\longrightarrow \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{0}\rangle \otimes |\psi_{\mathbf{a}_1^T\mathbf{s}}\rangle \otimes \cdots \otimes |\psi_{\mathbf{a}_m^T\mathbf{s}}\rangle \quad \propto \quad \sum_{\mathbf{s}\in\mathbb{Z}_q^n}\sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{\chi_\sigma^m}(\mathbf{e})|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

Notation

$$|\psi_j\rangle \propto \sum_{e\in\mathbb{Z}_q} \sqrt{\chi_\sigma}(e)|j+e\rangle$$

Our observation

This is an instance of *unambiguous state discrimination*

[CLZ22]: Y. Chen, Q. Liu, M. Zhandry, Eurocrypt'22

Unambiguous state discrimination

$|\psi_1\rangle, |\psi_2\rangle, \cdots, |\psi_q\rangle \in \mathbb{C}^q$ $\quad |\psi_j\rangle := \sum_{e \in \mathbb{Z}_q} f(e)|j+e\rangle$ $\quad f : \mathbb{Z}_q \to \mathbb{R}$ is known

Challenger $\xrightarrow{|\psi_j\rangle}$ Distinguisher

Challenger $\xleftarrow{\widehat{j} \in \{1, 2, \cdots, q, \perp\}}$ Distinguisher

not allowed to make mistakes

$\perp$ represents “don’t know”

Distinguisher wins if it finds j

CLZ distinguisher

$|\psi_1\rangle, |\psi_2\rangle, \cdots, |\psi_q\rangle \in \mathbb{C}^q$ $\quad |\psi_j\rangle := \sum_{e \in \mathbb{Z}_q} f(e)|j+e\rangle$ $\quad f : \mathbb{Z}_q \to \mathbb{R}$ is known

$$\text{Challenger} \xrightarrow{|\psi_j\rangle} \text{Distinguisher}$$

$$\text{Challenger} \xleftarrow{\widehat{j} \in \{1, 2, \cdots, q, \perp\}} \text{Distinguisher}$$

not allowed to make mistakes

$\perp$ represents “don’t know”

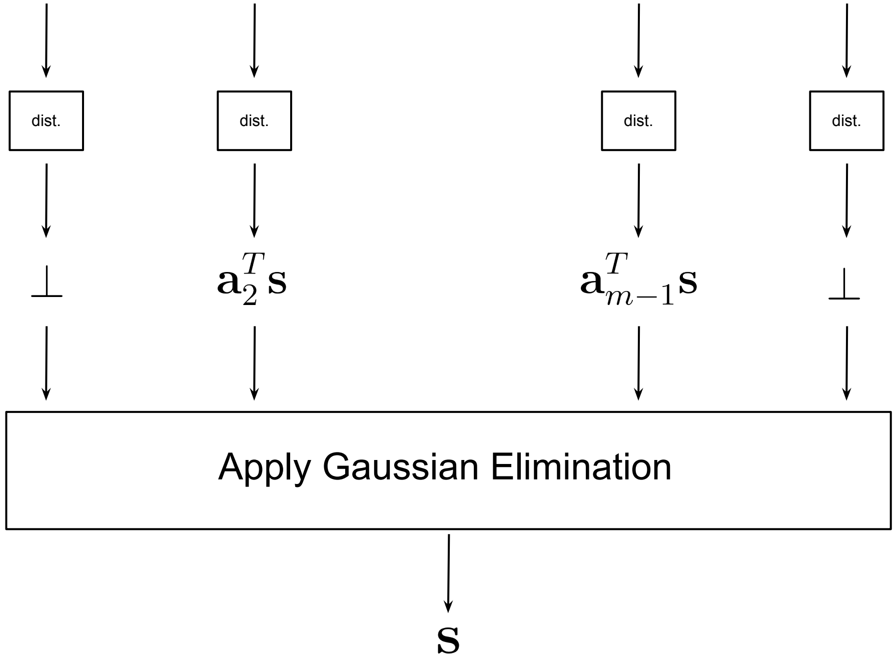
Distinguisher wins if it finds j

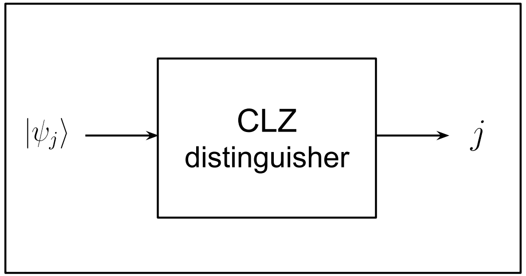
[CLZ22] distinguisher:

$$p_{\text{succ}}^{\text{CLZ}} = \min_y |\hat{f}(y)|^2 / q$$

Extraction with CLZ distinguisher

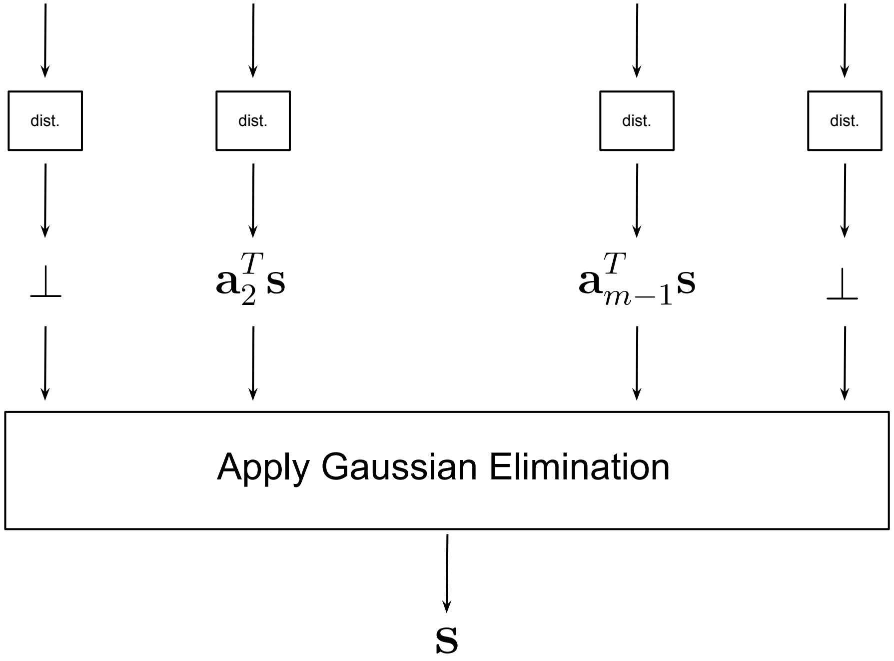
$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes |\psi_{\mathbf{a}_1^T\mathbf{s}}\rangle \otimes |\psi_{\mathbf{a}_2^T\mathbf{s}}\rangle \otimes \cdots \otimes |\psi_{\mathbf{a}_{m-1}^T\mathbf{s}}\rangle \otimes |\psi_{\mathbf{a}_m^T\mathbf{s}}\rangle$$

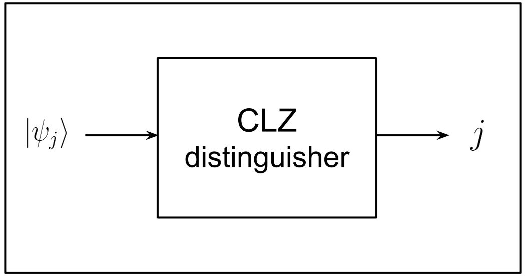




Extraction with CLZ distinguisher

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes |\psi_{\mathbf{a}_1^T\mathbf{s}}\rangle \otimes |\psi_{\mathbf{a}_2^T\mathbf{s}}\rangle \otimes \cdots \otimes |\psi_{\mathbf{a}_{m-1}^T\mathbf{s}}\rangle \otimes |\psi_{\mathbf{a}_m^T\mathbf{s}}\rangle$$





Requirement

$$m \gtrsim n/p_{\text{succ}}^{\text{CLZ}}$$
$$\approx nq^2 \cdot e^{\pi\sigma^2}$$

Summary of CLZ

	Distinguisher	[CLZ22]
	Success probability	$p_{\text{succ}}^{\text{CLZ}} = \min_y \lvert\hat{f}(y)\rvert^2/q$
when $f \propto \sqrt{\chi_\sigma}$	Requirement for GE[1]	$m \gtrsim nq^2 \cdot e^{\pi\sigma^2}$
	Circuit size	not specified

1: Gaussian Elimination

Summary of CLZ

	Distinguisher	[CLZ22]
	Success probability	$p_{\text{succ}}^{\text{CLZ}} = \min_y \lvert \hat{f}(y) \rvert^2 / q$
when $f \propto \sqrt{\chi_\sigma}$	Requirement for GE[1]	$m \gtrsim nq^2 \cdot e^{\pi\sigma^2}$
	Circuit size	naive implementation: poly(m, q)

1: Gaussian Elimination

How to improve it?

		Distinguisher	[CLZ22]	[CB98]
		Success probability	$p_{\mathrm{succ}}^{\mathrm{CLZ}} = \min_y \lvert\hat{f}(y)\rvert^2/q$	$p_{\mathrm{succ}}^{\mathrm{CB}} = q \cdot \min_y \lvert\hat{f}(y)\rvert^2$
when	$f \propto \sqrt{\chi_\sigma}$	Requirement for GE	$m \gtrsim nq^2 \cdot e^{\pi\sigma^2}$	$m \gtrsim n \cdot e^{\pi\sigma^2}$
		Circuit size	Naive implementation: poly(m, q)	not specified

[CB98]: A. Chefles, S. M. Barnett, Phys. Lett. A, 1998

How to improve it?

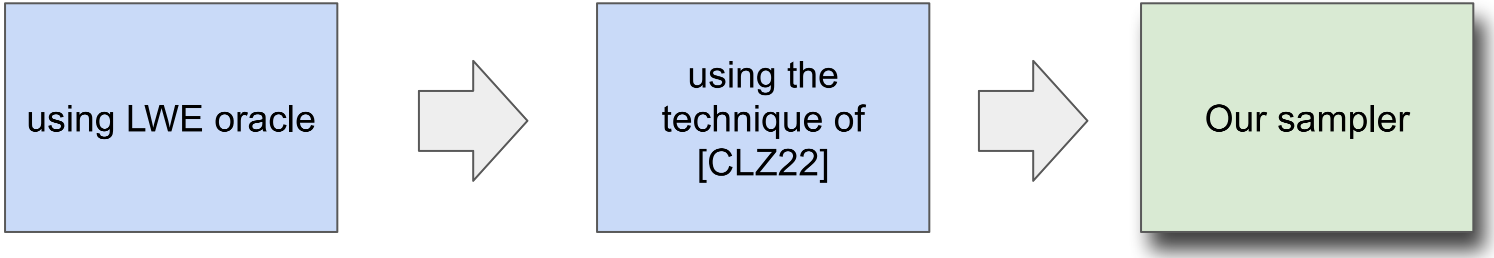
	Distinguisher	[CLZ22]	[CB98]
	Success probability	$p_{\text{succ}}^{\text{CLZ}} = \min_y \lvert\hat{f}(y)\rvert^2 / q$	$p_{\text{succ}}^{\text{CB}} = q \cdot \min_y \lvert\hat{f}(y)\rvert^2$
when $f \propto \sqrt{\chi_\sigma}$	Requirement for GE	$m \gtrsim nq^2 \cdot e^{\pi\sigma^2}$	$m \gtrsim n \cdot e^{\pi\sigma^2}$
	Circuit size	Naive implementation: poly(m, q)	our implementation: poly$(m, \log(q))$

[CB98]: A. Chefles, S. M. Barnett, Phys. Lett. A, 1998

Barrier

We can build the LWE state when $m \gtrsim n \cdot e^{\pi\sigma^2}$. For typical choices of σ, we need exponentially-large m!

Roadmap to LWE state



A toy example

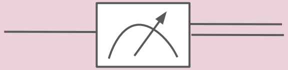
The constraints on the parameters cannot be satisfied

[CLZ22]: Y. Chen, Q. Liu, M. Zhandry, Eurocrypt'22

A new strategy

The superposition sampler:

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n}\sum_{\mathbf{e}\in\mathbb{Z}_q^m}\sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

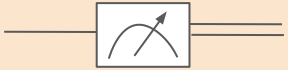


m-dimensional discrete Gaussian with standard deviation σ

A new strategy : LWE state with phase

The superposition sampler with phase:

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n}\sum_{\mathbf{e}\in\mathbb{Z}_q^m} e^{i\theta(\mathbf{e})}\sqrt{\chi_\sigma^m}(\mathbf{e})|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$



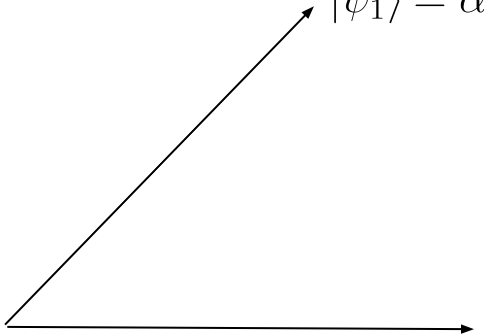
m-dimensional discrete Gaussian with standard deviation σ

The phase does not have any effects
on the distribution of the outcome

Do phases help the distinguisher?

$\boxed{\text{Assume that } q = 2}$

$$|\psi_1\rangle = \alpha|0\rangle + \beta|1\rangle$$



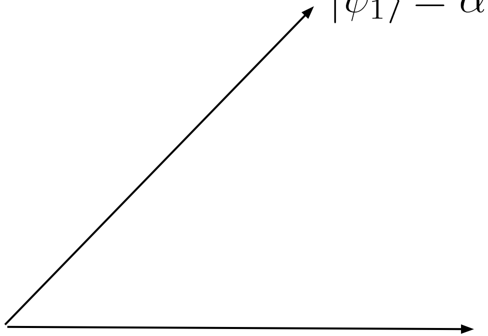
$$|\psi_2\rangle = \beta|0\rangle + \alpha|1\rangle$$

$$\alpha, \beta \in \mathbb{R} \setminus \{0\}$$

Do phases help the distinguisher?

Assume that $q = 2$

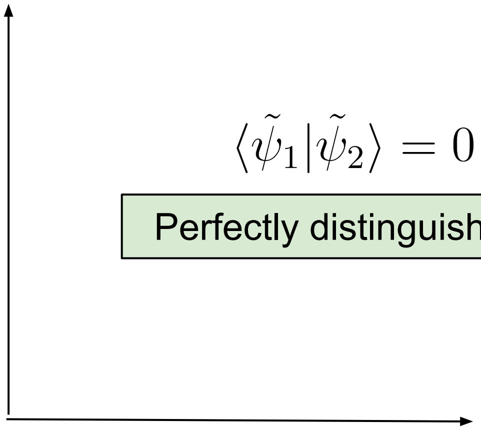
$$|\psi_1\rangle = \alpha|0\rangle + \beta|1\rangle$$



$$|\psi_2\rangle = \beta|0\rangle + \alpha|1\rangle$$

$$\alpha, \beta \in \mathbb{R} \setminus \{0\}$$

$$|\tilde{\psi}_1\rangle = \alpha|0\rangle + i\beta|1\rangle$$



$$\langle\tilde{\psi}_1|\tilde{\psi}_2\rangle = 0$$

Perfectly distinguishable

$$|\tilde{\psi}_2\rangle = i\beta|0\rangle + \alpha|1\rangle$$

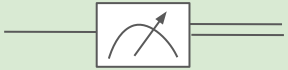
A new strategy : LWE state with sign

Observation:
sign exponentially improves the lower bound

$$\mathrm{sign}(e) := \begin{cases} +1 & e \in [0, \frac{q}{2}] \\ -1 & e \in (-\frac{q}{2}, 0) \end{cases}$$

Our quantum LWE sampler:

$$\sum_{\mathbf{s} \in \mathbb{Z}_q^n} \sum_{\mathbf{e} \in \mathbb{Z}_q^m} \mathrm{sign}(\mathbf{e}) \sqrt{\chi_\sigma^m(\mathbf{e})} |\mathbf{A}\mathbf{s} + \mathbf{e}\rangle$$



m-dimensional discrete Gaussian with standard deviation σ

LWE state with sign

Let $\mathbf{A}$ have arbitrarily many rows

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \mathrm{sign}(\mathbf{e})\sqrt{\chi_\sigma^m}(\mathbf{e})|\mathbf{e}\rangle$$

$$\longrightarrow \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \mathrm{sign}(\mathbf{e})\sqrt{\chi_\sigma^m}(\mathbf{e})|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

$$\propto \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes |\widetilde{\psi}_{\mathbf{a}_1^T\mathbf{s}}\rangle \otimes \cdots \otimes |\widetilde{\psi}_{\mathbf{a}_m^T\mathbf{s}}\rangle$$

Notation

$$|\widetilde{\psi}_j\rangle \propto \sum_{e\in\mathbb{Z}_q} \mathrm{sign}(e)\sqrt{\chi_\sigma}(e)|j+e\rangle$$

"superposition of signed Gaussian distribution centered around j"

LWE state with sign

Let $\mathbf{A}$ have arbitrarily many rows

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \mathrm{sign}(\mathbf{e})\sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{e}\rangle$$

$$\longrightarrow \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \mathrm{sign}(\mathbf{e})\sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

$$\propto \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \underbrace{|\widetilde{\psi}_{\mathbf{a}_1^T\mathbf{s}}\rangle \otimes \cdots \otimes |\widetilde{\psi}_{\mathbf{a}_m^T\mathbf{s}}\rangle}_{\text{Extract } \mathbf{s} \text{ from these}}$$

Notation

$$|\widetilde{\psi}_j\rangle \propto \sum_{e\in\mathbb{Z}_q} \mathrm{sign}(e)\sqrt{\chi_\sigma(e)}|j+e\rangle$$

"superposition of signed Gaussian distribution centered around j"

LWE state with sign

Let $\mathbf{A}$ have arbitrarily many rows

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n}|\mathbf{s}\rangle\otimes\sum_{\mathbf{e}\in\mathbb{Z}_q^m}\mathrm{sign}(\mathbf{e})\sqrt{\chi_\sigma^m}(\mathbf{e})|\mathbf{e}\rangle$$

$$\longrightarrow\sum_{\mathbf{s}\in\mathbb{Z}_q^n}|\mathbf{s}\rangle\otimes\sum_{\mathbf{e}\in\mathbb{Z}_q^m}\mathrm{sign}(\mathbf{e})\sqrt{\chi_\sigma^m}(\mathbf{e})|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

$$\propto\sum_{\mathbf{s}\in\mathbb{Z}_q^n}|\mathbf{s}\rangle\otimes\underbrace{|\widetilde{\psi}_{\mathbf{a}_1^T\mathbf{s}}\rangle\otimes\cdots\otimes|\widetilde{\psi}_{\mathbf{a}_m^T\mathbf{s}}\rangle}_{\text{Extract } \mathbf{s} \text{ from these}}$$

$$\longrightarrow\sum_{\mathbf{s}\in\mathbb{Z}_q^n}|\mathbf{0}\rangle\otimes|\widetilde{\psi}_{\mathbf{a}_1^T\mathbf{s}}\rangle\otimes\cdots\otimes|\widetilde{\psi}_{\mathbf{a}_m^T\mathbf{s}}\rangle\quad\propto\quad\sum_{\mathbf{s}\in\mathbb{Z}_q^n}\sum_{\mathbf{e}\in\mathbb{Z}_q^m}\mathrm{sign}(\mathbf{e})\sqrt{\chi_\sigma^m}(\mathbf{e})|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

Notation

$$|\widetilde{\psi}_j\rangle\propto\sum_{e\in\mathbb{Z}_q}\mathrm{sign}(e)\sqrt{\chi_\sigma}(e)|j+e\rangle$$

"superposition of signed Gaussian distribution centered around j"

LWE state with sign

Let $\mathbf{A}$ have arbitrarily many rows

$$\sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \mathrm{sign}(\mathbf{e})\sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{e}\rangle$$

$$\longrightarrow \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \mathrm{sign}(\mathbf{e})\sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

$$\propto \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{s}\rangle \otimes \underbrace{|\widetilde{\psi}_{\mathbf{a}_1^T\mathbf{s}}\rangle \otimes \cdots \otimes |\widetilde{\psi}_{\mathbf{a}_m^T\mathbf{s}}\rangle}_{\text{Extract } \mathbf{s} \text{ from these}}$$

$$\longrightarrow \sum_{\mathbf{s}\in\mathbb{Z}_q^n} |\mathbf{0}\rangle \otimes |\widetilde{\psi}_{\mathbf{a}_1^T\mathbf{s}}\rangle \otimes \cdots \otimes |\widetilde{\psi}_{\mathbf{a}_m^T\mathbf{s}}\rangle \quad \propto \quad \sum_{\mathbf{s}\in\mathbb{Z}_q^n} \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \mathrm{sign}(\mathbf{e})\sqrt{\chi_\sigma^m(\mathbf{e})}|\mathbf{A}\mathbf{s}+\mathbf{e}\rangle$$

Notation

$$|\widetilde{\psi}_j\rangle \propto \sum_{e\in\mathbb{Z}_q} \mathrm{sign}(e)\sqrt{\chi_\sigma(e)}|j+e\rangle$$

How does it work?

- Apply CB distinguisher

$$p_{\mathrm{succ}}^{\mathrm{sign}} \approx 1/\sigma$$

- Apply Gaussian elimination which requires $m \gtrsim n\sigma$

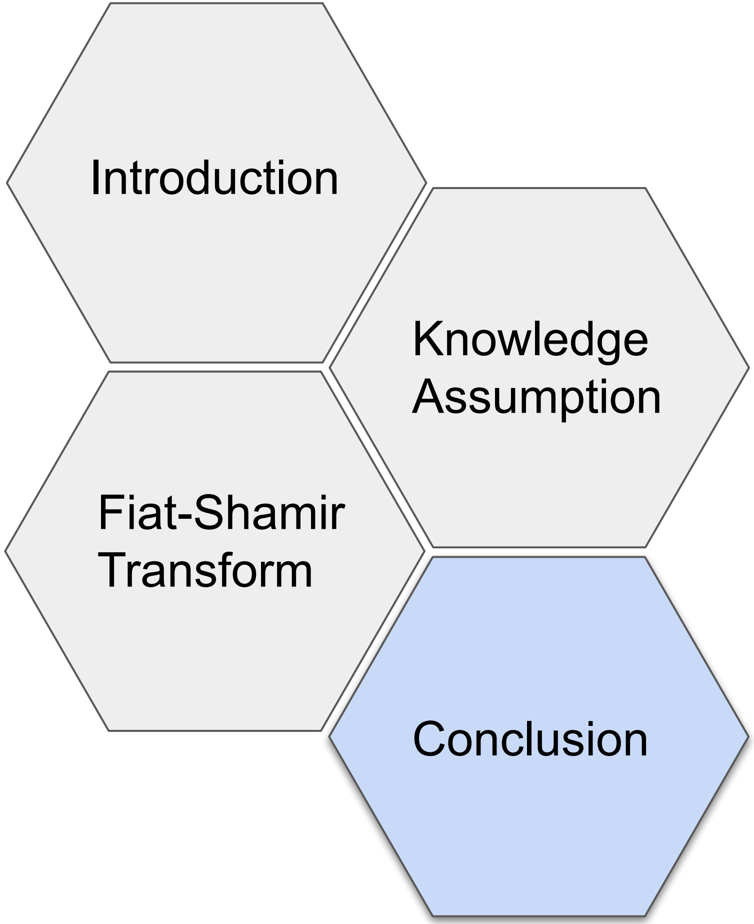
Table of results

	Distinguisher	[CLZ22]	[CB98]
	Success probability	$p_{\text{succ}}^{\text{CLZ}} = \min_y \lvert\hat{f}(y)\rvert^2/q$	$p_{\text{succ}}^{\text{CB}} = q \cdot \min_y \lvert\hat{f}(y)\rvert^2$
when $f \propto \sqrt{\chi_\sigma}$	Requirement for GE	$m \gtrsim nq^2 \cdot e^{\pi\sigma^2}$	$m \gtrsim n \cdot e^{\pi\sigma^2}$
	Circuit size	naive implementation: poly(m, q)	our implementation: poly$(m, \log(q))$

Table of results

				Requirement: $f(x) = f(-x \bmod q)$
	Distinguisher	[CLZ22]	[CB98]	with signs
	Success probability	$p_{\text{succ}}^{\text{CLZ}} = \min_y \lvert\hat{f}(y)\rvert^2/q$	$p_{\text{succ}}^{\text{CB}} = q \cdot \min_y \lvert\hat{f}(y)\rvert^2$	$p_{\text{succ}}^{\text{sign}} = \lvert f(0)\rvert^2$
when $f \propto \sqrt{\chi_\sigma}$	Requirement for GE	$m \gtrsim nq^2 \cdot e^{\pi\sigma^2}$	$m \gtrsim n \cdot e^{\pi\sigma^2}$	$m \gtrsim n\sigma$
	Circuit size	naive implementation: poly(m, q)	our implementation: poly$(m, \log(q))$	our implementation: poly$(m, \log(q))$

Outline


Introduction
Knowledge Assumption
Fiat-Shamir Transform
Conclusion

Conclusion

- A CMA-to-NMA reduction for FSwA signatures in the QROM
  - A detailed correctness and runtime analysis
  - We also provide a similar reduction from the strong variant of CMA

Open question: Is the reduction tight? Can we achieve a tighter one in terms of runtime and reduction loss?

Analysis of CMA $\leq$ NMA	Adaptive reprogramming (extension of [GHHM21])
Reduction loss	$2^{-\alpha/2} B Q_S Q_H^{1/2} + \varepsilon_{zk} B Q_S$
Runtime	$Q_H \log(B Q_S)$

Conclusion

- Obliviously sampling instances of LWE with poly-large standard deviation
  - Extendable to exponentially-large standard deviation
  - Generalizable to structured variants of LWE (Module-LWE)

Open question: Can we extend it to other distribution of matrices, and therefore other classes of lattices?

Our oblivious LWE sampler:

- given $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- returns $\mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$

does not require any special property of the matrix

Thank you for attending and/or listening!

[BISW17] Dan Boneh, Yuval Ishai, Amit Sahai, and David J. Wu. Lattice-based SNARGs and their application to more efficient obfuscation. In EUROCRYPT, 2017.

[CB98] Anthony Chefles and Stephen M. Barnett. Optimum unambiguous discrimination between linearly independent symmetric states. Phys. Lett. A, 1998.

[GMNO18] Rosario Gennaro, Michele Minelli, Anca Nitulescu, and Michele Orrù. Lattice-based ZK-SNARKs from square span programs. In CCS, 2018.

[GNSV23] Chaya Ganesh, Anca Nitulescu, and Eduardo Soria-Vazquez. Rinocchio: SNARKs for ring arithmetic. J. Cryptol., 2023.

[ISW21] Yuval Ishai, Hang Su, and David J. Wu. Shorter and faster post-quantum designated-verifier zkSNARKs from lattices. In CCS, 2021.

[NYI+20] Ken Naganuma, Masayuki Yoshino, Atsuo Inoue, Yukinori Matsuoka, Mineaki Okazaki, and Noboru Kunihiro. Post-quantum zk-SNARK for arithmetic circuits using QAPs. In AsiaJCIS, 2020.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. J. ACM, 2009.

[SSEK22] Ron Steinfeld, Amin Sakzad, Muhammed F. Esgin, and Veronika Kuchta. Private re-randomization for module LWE and applications to quasi-optimal ZK-SNARKs, 2022. Available at https://eprint.iacr.org/2022/1690.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In ASIACRYPT, 2009.

[CKKK23] Heewon Chung, Dongwoo Kim, Jeong Han Kim, and Jiseung Kim. Amortized efficient zk-SNARK from linear-only RLWE encodings. J. Comm. Netw., 2023.

[CLZ22] Yilei Chen, Qipeng Liu, and Mark Zhandry. Quantum algorithms for variants of average-case lattice problems via filtering. In EUROCRYPT, 2022.

[KLS18] E. Kiltz, V. Lyubashevsky, and C. Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In EUROCRYPT, 2018.

[GHHM21] A. B. Grilo, K. Hövelmanns, A. Hülsing, and C. Majenz. In ASIACRYPT, 2022.

The observer picture (the eye) and the atom picture are borrowed from wikipedia