# CSCI 1515 Applied Cryptography

This Lecture:

- Private Information Retrieval
- Bootstrapping SWHE to FHE
- Secure Hardware: Intel SGX

# Fully Homomorphic Encryption (FHE)

$Enc(m_1)$, $Enc(m_2)$ → $Enc(m_1 + m_2)$

Additively Homomorphic

$Enc(m_1)$, $Enc(m_2)$ → $Enc(m_1 \cdot m_2)$

Multiplicatively Homomorphic

$c$, $Enc(m)$ → $Enc(c \cdot m)$

Homomorphic Scalar Multiplication

# FHE Constructions

**Step 1:** Somewhat Homomorphic Encryption (SWHE)

- over Integers
- from RLWE (BFV)

**Step 2:** Bootstrapping

# Application: Privacy-Preserving Query

Server

Client

Query $x$

Key $sk$

$ct \leftarrow Enc(x)$

$ct$

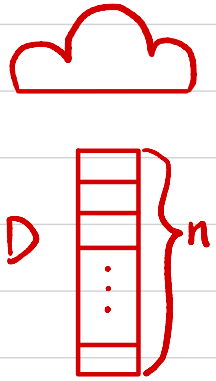
Search / ML / GPT / ...

$ct' \leftarrow Eval(f, ct)$

$ct'$

$f(x) \leftarrow Dec_{sk}(ct')$

# Application: Private Information Retrieval (PIR)





WANT: D[i]

While hiding i against Server

Query i

Key sk

$ct \leftarrow Enc(i)$

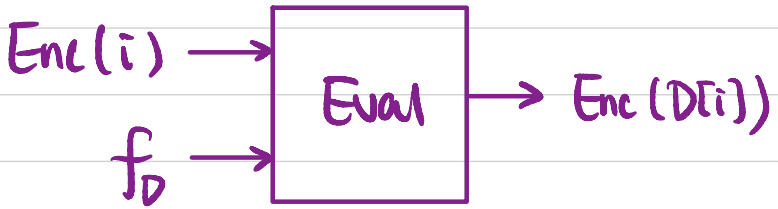
$ct$ →

$ct' \leftarrow Eval(f, ct)$

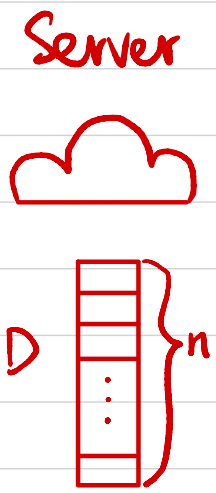
$f_D(i) = D[i]$

$ct'$ →

$D[i] \leftarrow Dec_{sk}(ct')$



# Private Information Retrieval (PIR)





WANT: D[i]

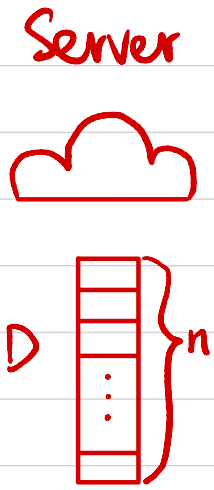
While hiding i against Server

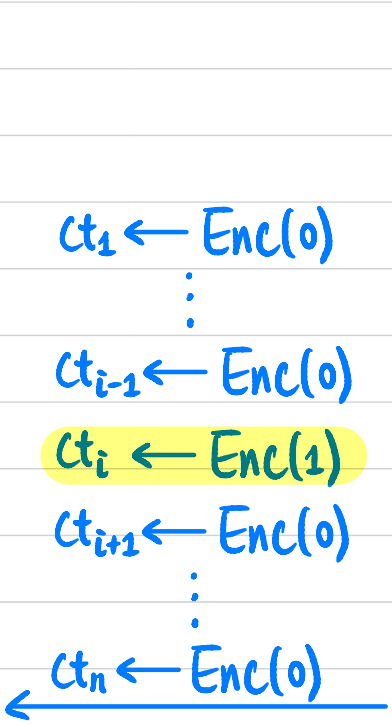
Trivial Solution:


Communication complexity $O(n)$

Goal: Communication complexity $o(n)$

# Private Information Retrieval (PIR)



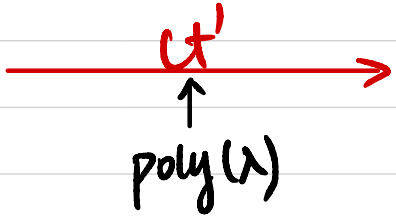

WANT: D[i]

While hiding i against Server

Homomorphic Scalar Mult

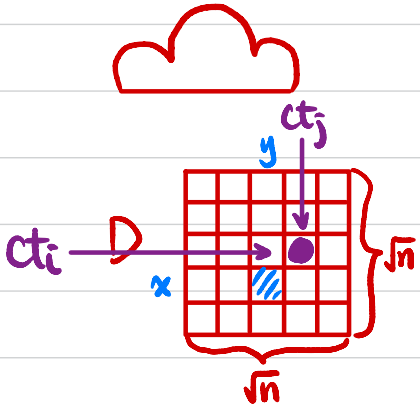
$$ct' \leftarrow \sum_{i=1}^{n} D[i] \cdot ct_i$$

Homomorphic Add




$D[i] = Dec(ct')$

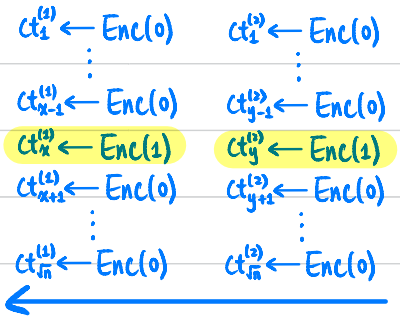
# Private Information Retrieval (PIR)







WANT: $D[x,y]$




While hiding (x,y) against Server

↑

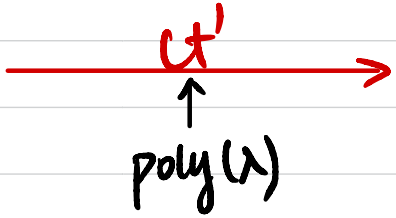
Why?

Homomorphic Scalar Mult

$$ct' \leftarrow \sum_{i,j=1}^{\sqrt{n}} D[i,j] \cdot ct_i^{(1)} \cdot ct_j^{(2)}$$

↑ Homomorphic Add

↑ Homomorphic Mult




$D[x,y] = Dec(ct')$

↑

Why?

**Extend to dimension d ?**

# Homomorphic Mult = $(d-1) \cdot n$

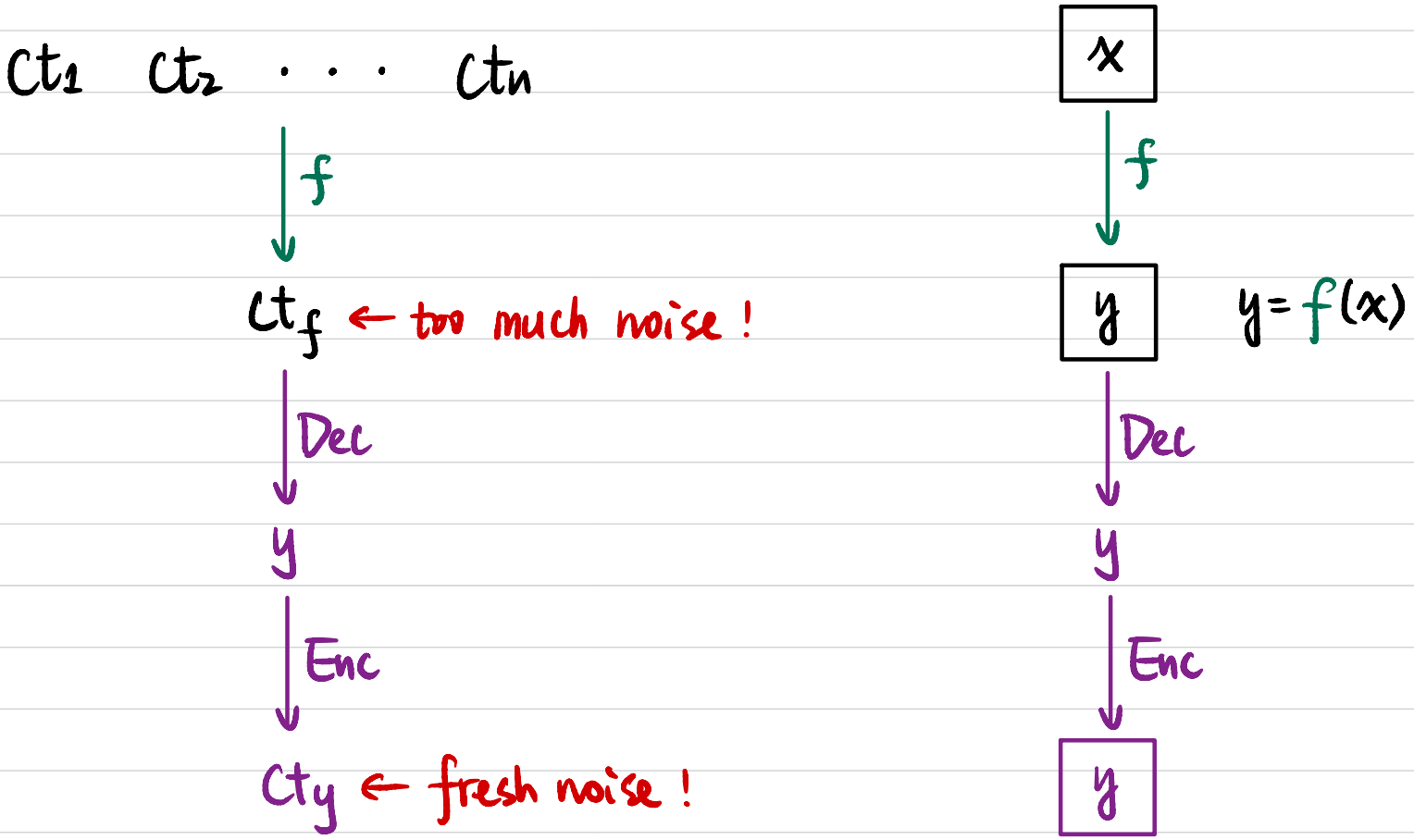
# Homomorphic Scalar Mult = $n$

# Homomorphic Add = $n$

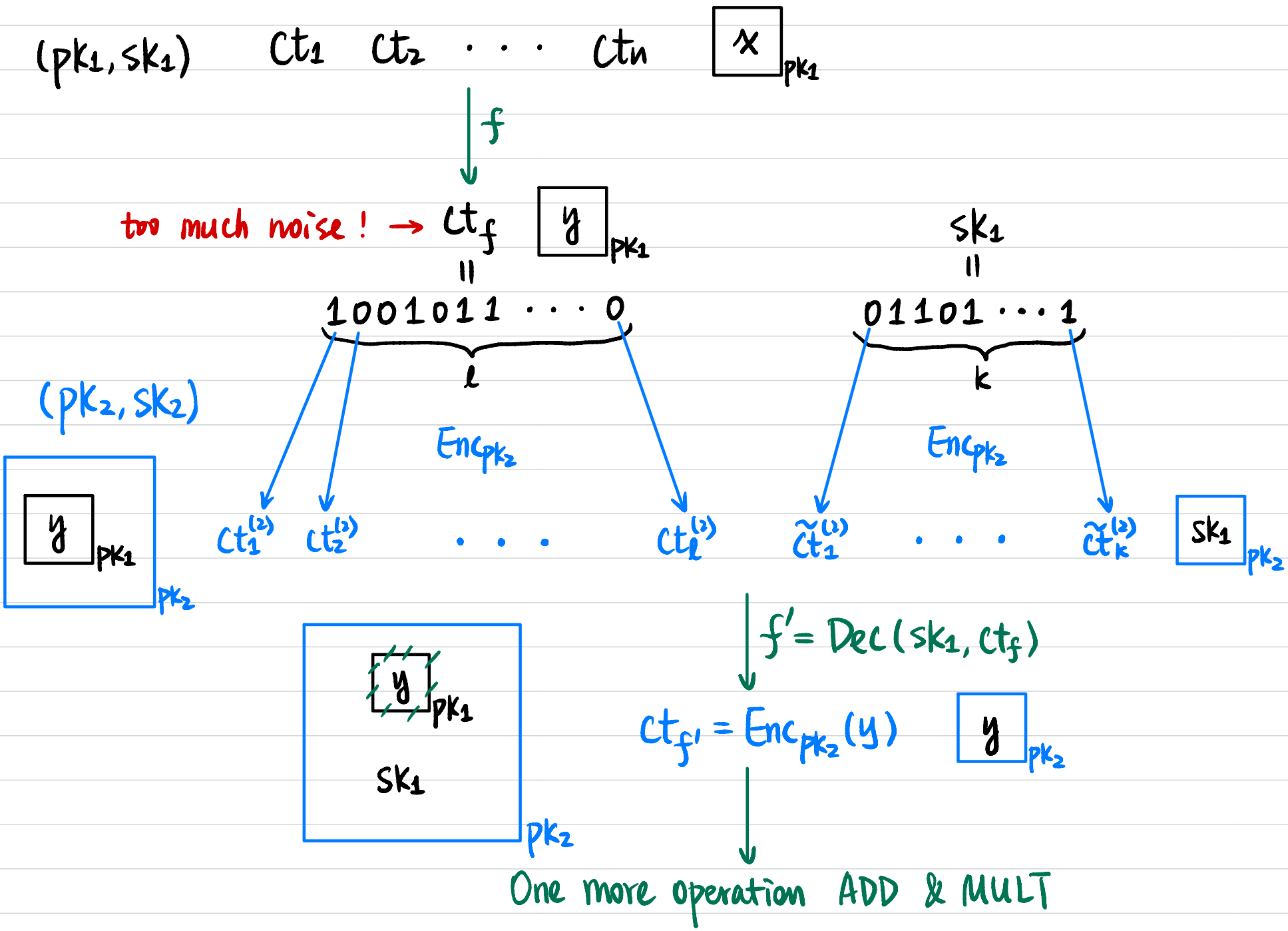
Communication = $d \cdot n^{1/d} \cdot poly(\lambda)$

$d \leq \log_2 n$

# Step 2: Bootstrapping



# Leveled FHE


$(pk_1, sk_1)$
$Ct_1$ $Ct_2$ · · · $Ct_n$
$x_{pk_1}$
$f$
too much noise ! → $Ct_f$
$y_{pk_1}$
$sk_1$
=
1001011 · · · 0
$l$
01101 · · · 1
$k$
$(pk_2, sk_2)$
$Enc_{pk_2}$
$Enc_{pk_2}$
$y_{pk_1}$ $pk_2$
$Ct_1^{(2)}$ $Ct_2^{(2)}$ · · · $Ct_l^{(2)}$
$\tilde{Ct}_1^{(2)}$ · · · $\tilde{Ct}_k^{(2)}$
$sk_1$ $pk_2$
$y_{pk_1}$
$sk_1$
$pk_2$
$f' = Dec(sk_1, Ct_f)$
$Ct_{f'} = Enc_{pk_2}(y)$
$y_{pk_2}$
One more operation ADD & MULT

## Step 2: Bootstrapping

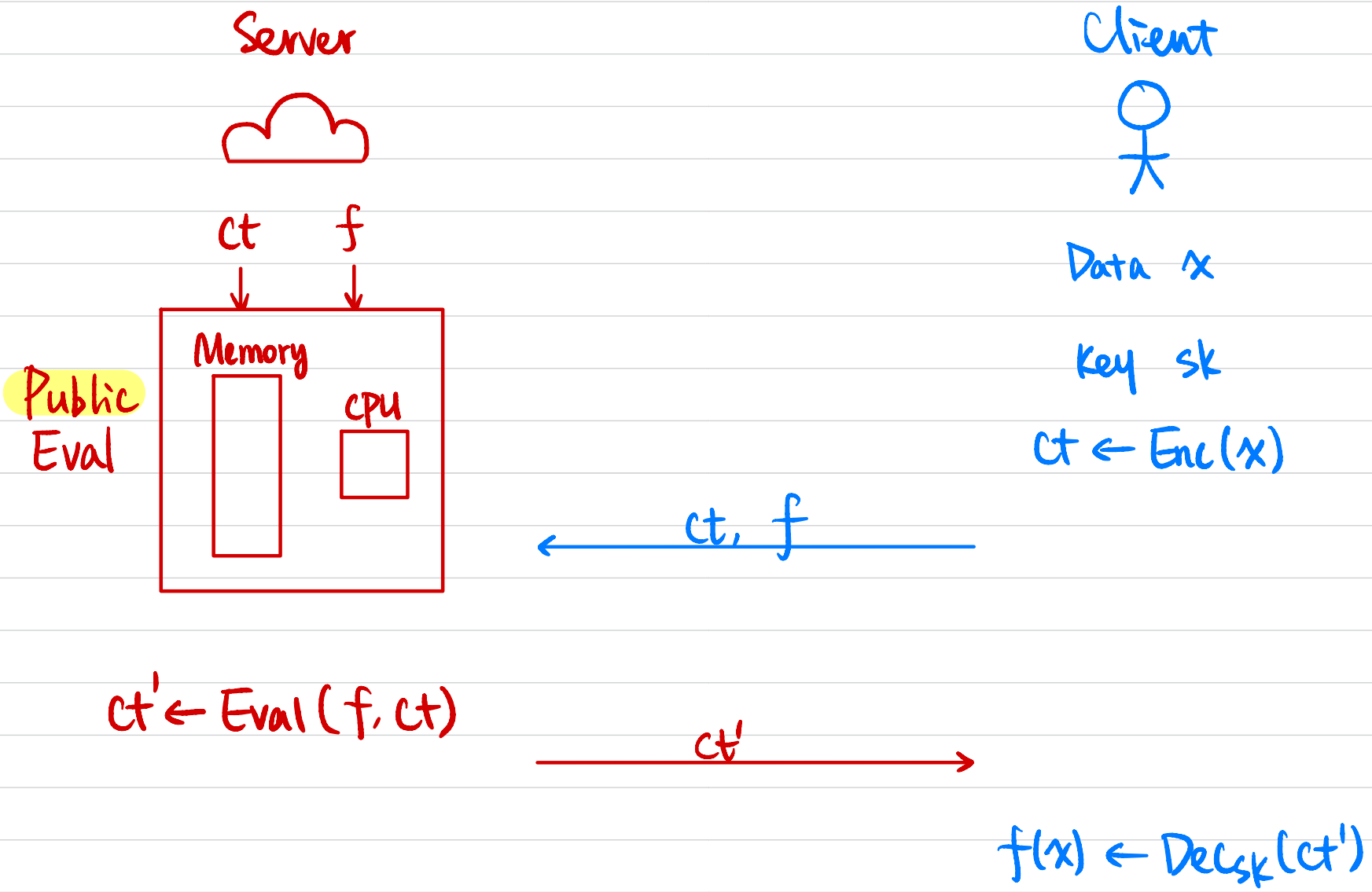
Leveled FHE: $pk_1$, $pk_2$, $pk_3$, . . . , $pk_n$

$Enc_{pk_2}(sk_1)$ $Enc_{pk_3}(sk_2)$ $Enc_{pk_n}(sk_{n-1})$

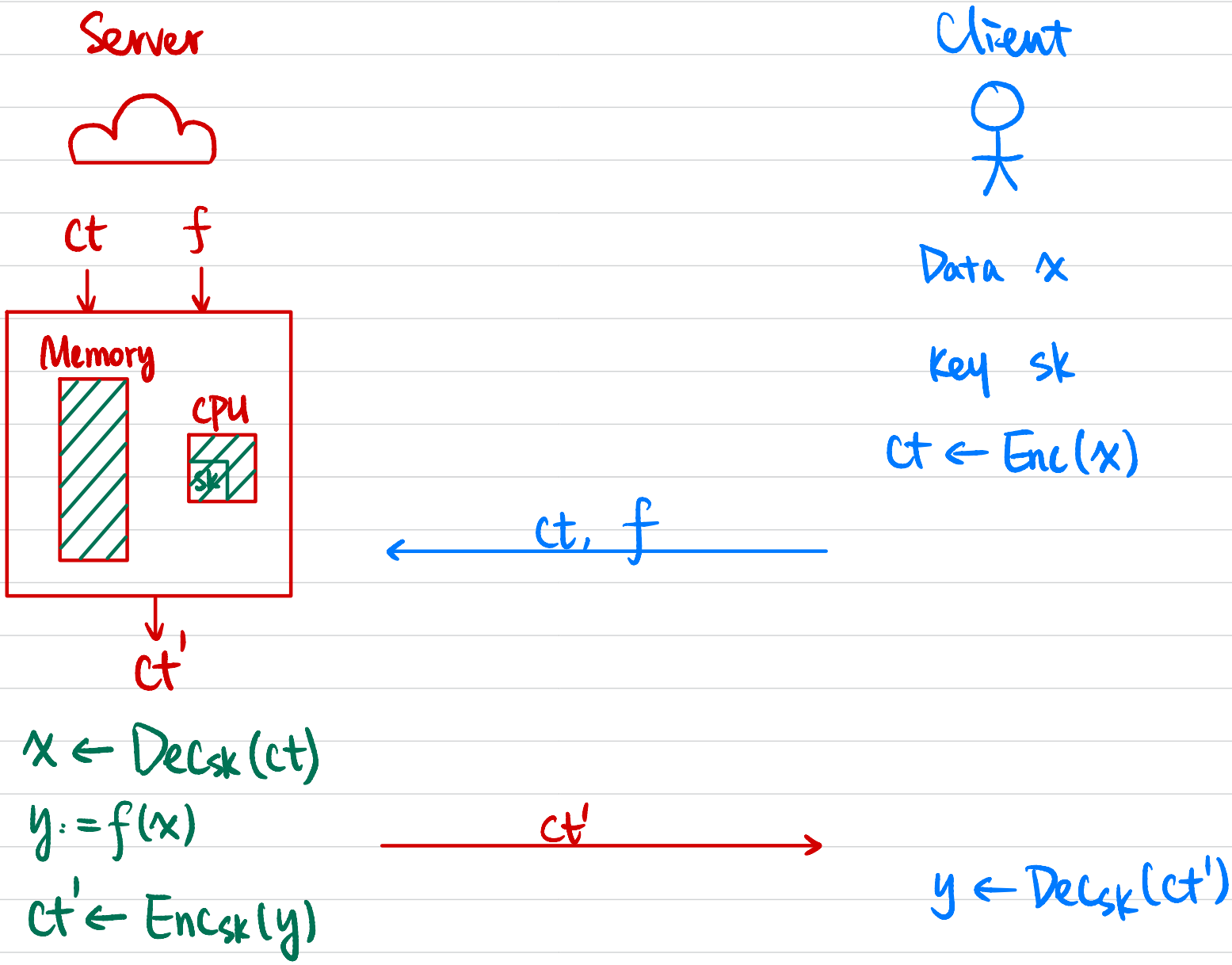
FHE: $pk$. $Enc_{pk}(sk)$

"circular secure" assumption

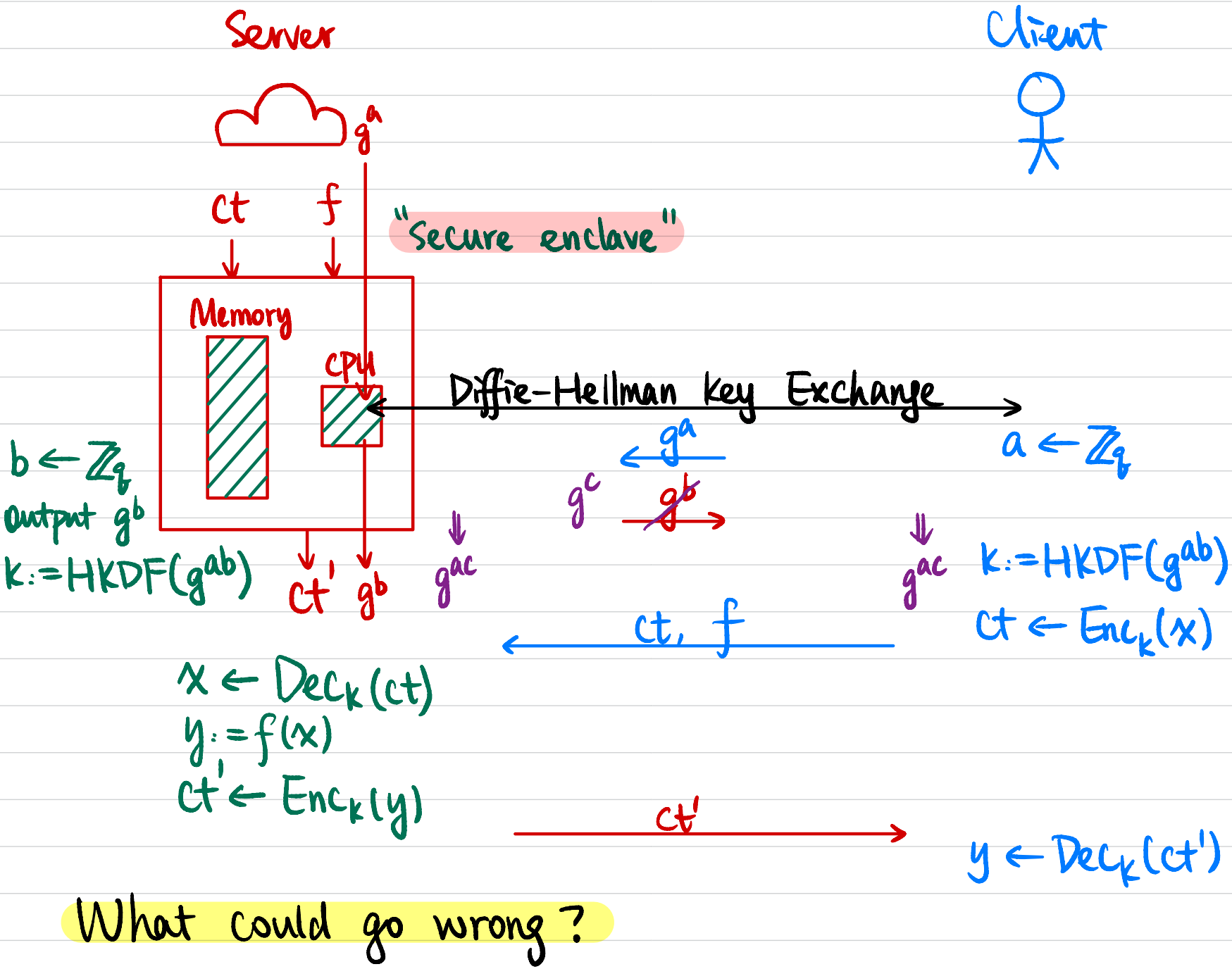
# Outsourcing Computation by FHE



# Outsourcing Computation by Secure Hardware

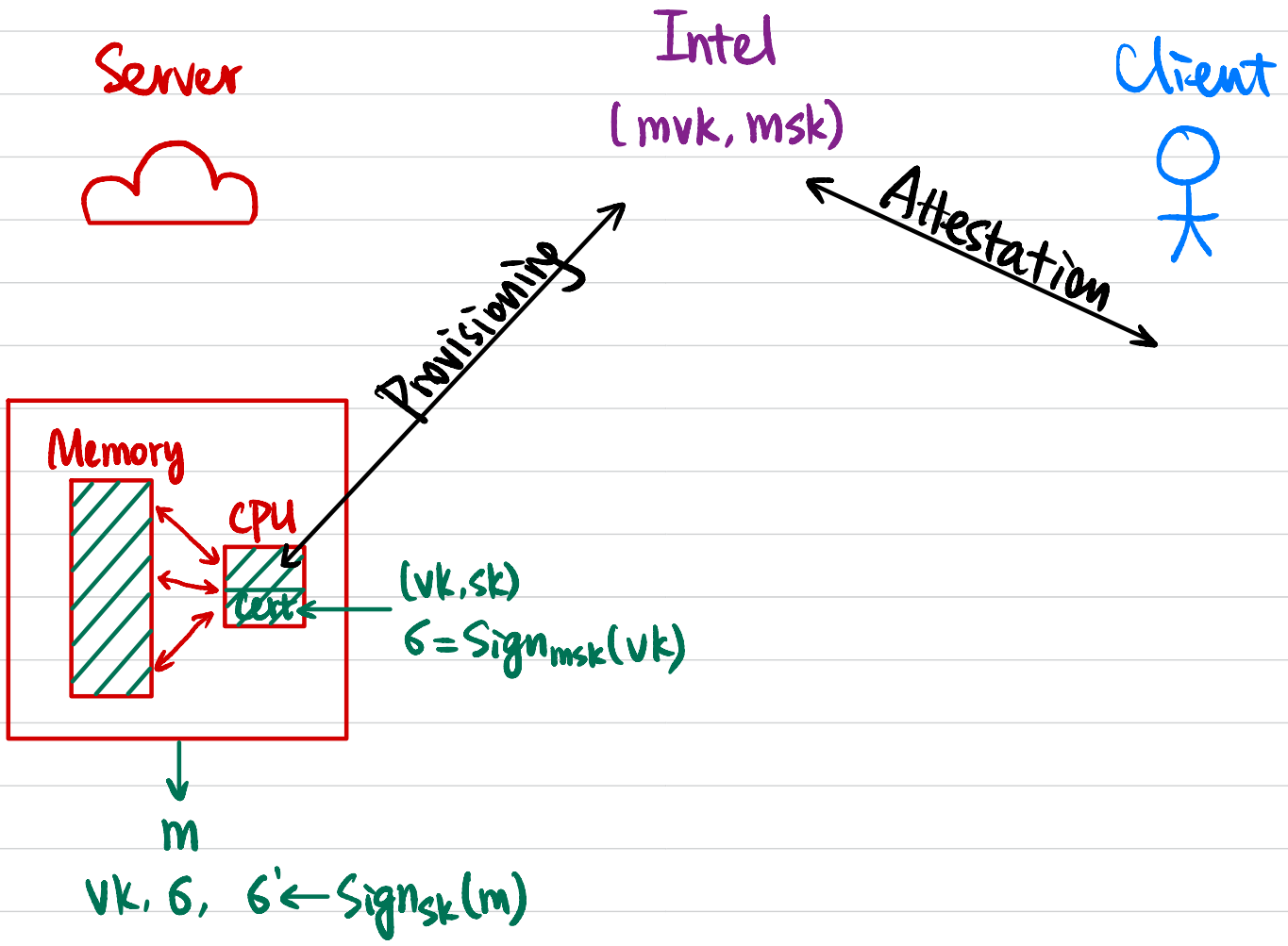


# Intel Software Guard Extension (SGX)




What could go wrong?

# Intel Software Guard Extension (SGX)




m

vk, σ, σ'

# Constraints & Attacks

- Trust in hardware
- Trust in Intel
- Limited memory size: 128 MB
- Replay attacks
- Side-channel attacks: memory access pattern
  - ↳ fix: Oblivious RAM (ORAM)
    overhead $\Theta(\log N)$