CSCI 1510

This Lecture:

- Somewhat Homomorphic Encryption from LWE (GSW)
- Bootstrapping SWHE to FHE
- Program Obfuscation

FHE Constructions

Step 1: Somewhat Homomorphic Encryption (SWHE) from LWE (GSW)

Step 2: Bootstrapping

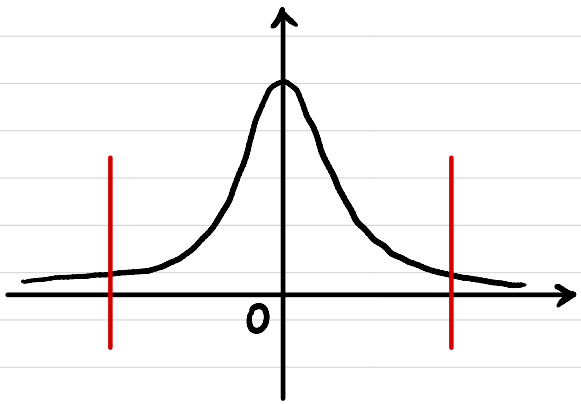
Post-Quantum Assumption: Learning With Errors (LWE)

n: security parameter

$q \sim 2^{n^\epsilon}$

$m = \Omega(n \log q)$

χ: distribution over $\mathbb{Z}_q$
(concentrated on "small integers")

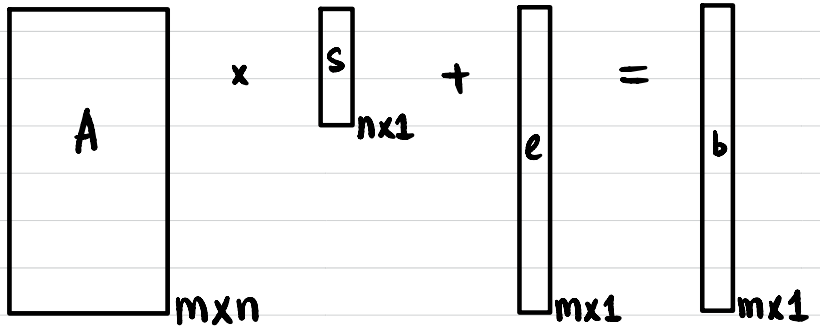


$$\Pr[|e| > \alpha \cdot q \mid e \leftarrow \chi] \leq \text{negl}(n)$$

$\alpha \ll 1$

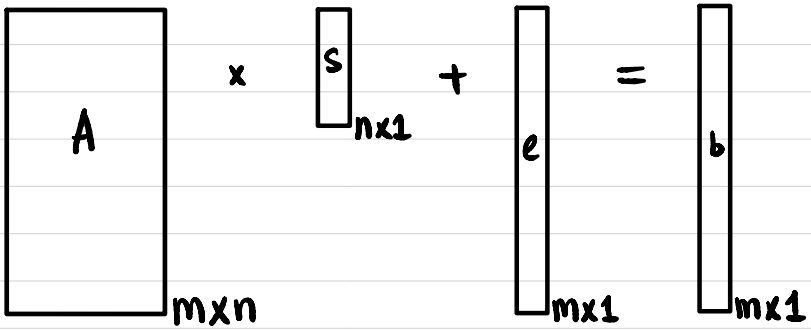
Def We say the decisional $\text{LWE}_{n,m,q,\chi}$ problem is (quantum) hard if $\forall$ (quantum) PPT $\mathcal{A}$, $\exists$ negligible function $\epsilon(\cdot)$ s.t.

$$\left| \Pr\left[\begin{matrix} A \xleftarrow{\$} \mathbb{Z}_q^{m \times n} \\ s \xleftarrow{\$} \mathbb{Z}_q^n \\ e \leftarrow \chi^m \end{matrix} : \mathcal{A}(A, [As + e \bmod q]) = 1 \right] - \Pr\left[\begin{matrix} A \xleftarrow{\$} \mathbb{Z}_q^{m \times n} \\ b' \xleftarrow{\$} \mathbb{Z}_q^m \end{matrix} : \mathcal{A}(A, b') = 1 \right] \right| \leq \epsilon(n).$$



Regev Encryption from LWE

$A \xleftarrow{\$} \mathbb{Z}_q^{m \times n} \quad s \xleftarrow{\$} \mathbb{Z}_q^{n} \quad e \leftarrow \chi^m$



$pk = (A, b)$

$sk = s$

$\text{Enc}_{pk}(\mu)$: $\mu \in \{0,1\}$

sample a random $S \subseteq [m]$

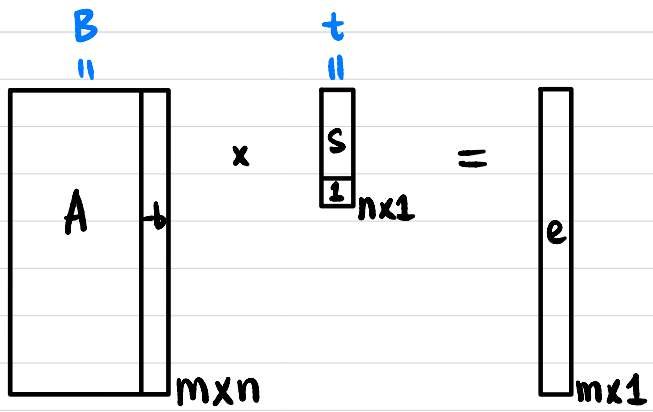
$$c = \left(\sum_{i \in S} A_i \,,\, \left(\sum_{i \in S} b_i \right) + \mu \cdot \lfloor \tfrac{q}{2} \rfloor \right)$$

A_i: i-th row of A

$\text{Dec}_{sk}(c)$: $c = [\,c_1 \,|\, c_2\,]$

$$c_2 - \langle c_1, s \rangle = \mu \cdot \lfloor \tfrac{q}{2} \rfloor + \sum_{i \in S} e_i$$

$\sum_{i \in S} e_i$: small noise



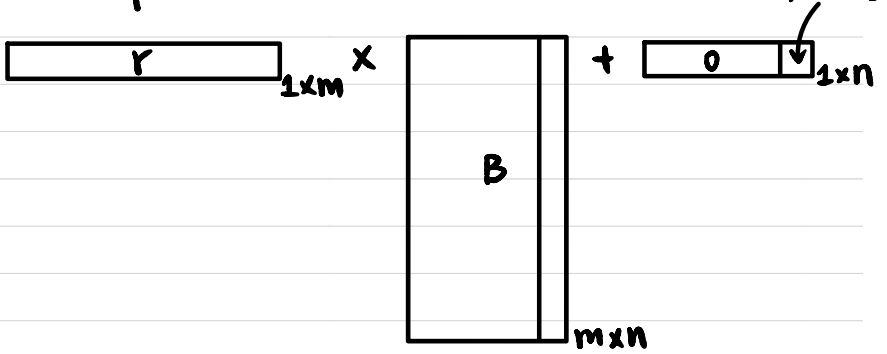
$pk = B_{m \times n}$

$sk = t_{n \times 1}$

$B \cdot t = \text{small}$

$\text{Enc}_{pk}(\mu)$: $\mu \in \{0,1\}$

sample $r \xleftarrow{\$} \{0,1\}^m$



$$c = r^T \cdot B + (0, \cdots, 0, \mu \cdot \lfloor \tfrac{q}{2} \rfloor)$$

$\text{Dec}_{sk}(c)$: $\langle c, t \rangle = \mu \cdot \lfloor \tfrac{q}{2} \rfloor + \text{small noise}$

Regev Encryption from LWE

Homomorphism:

$c_1 = Enc(\mu_1)$ $\quad \langle c_1, t \rangle = \text{"small"} + \mu_1 \cdot \lfloor \frac{q}{2} \rfloor$

$c_2 = Enc(\mu_2)$ $\quad \langle c_2, t \rangle = \text{"small"} + \mu_2 \cdot \lfloor \frac{q}{2} \rfloor$

Additive Homomorphism?

$c = c_1 + c_2$

$\langle c, t \rangle = \text{"small"} + (\mu_1 + \mu_2) \cdot \lfloor \frac{q}{2} \rfloor$

Multiplicative Homomorphism?

SWHE from LWE (GSW)

Attempt 1 (secret-key)

$$sk = t_{nx1} \quad \begin{bmatrix} s \\ 1 \end{bmatrix}_{nx1}$$

$Enc_{sk}(\mu)$: $\mu \in \{0,1\}$

Sample $C_0 \in \mathbb{Z}_q^{n \times n}$ s.t. $C_0 \cdot \vec{t} = small$

$$\underset{nxn}{C_0} \times \underset{nx1}{t} = \underset{nx1}{e}$$

$$C = C_0 + \mu \cdot I$$

C: $n \times n$; I: identity matrix

$Dec_{sk}(c)$: $C \cdot \vec{t} = (C_0 + \mu \cdot I) \cdot \vec{t} = small + \mu \cdot \vec{t}$

CPA Security?

SWHE from LWE (GSW)

Attempt 1 (secret-key)

Without Error: $c \cdot \vec{t} = \mu \cdot \vec{t}$

With Error: $c \cdot \vec{t} = \mu \cdot \vec{t} + \vec{e}$

Homomorphism:

$$c_1 \cdot \vec{t} = \mu_1 \cdot \vec{t}$$
$$c_2 \cdot \vec{t} = \mu_2 \cdot \vec{t}$$

Homomorphism:

$$c_1 \cdot \vec{t} = \mu_1 \cdot \vec{t} + \vec{e}_1$$
$$c_2 \cdot \vec{t} = \mu_2 \cdot \vec{t} + \vec{e}_2$$

Additive Homomorphism?

$c_1 + c_2$?

$$(c_1 + c_2) \cdot \vec{t} = (\mu_1 + \mu_2) \cdot \vec{t}$$

Additive Homomorphism?

$c_1 + c_2$?

$$(c_1 + c_2) \cdot \vec{t} = (\mu_1 + \mu_2) \cdot \vec{t} + (\vec{e}_1 + \vec{e}_2)$$

Multiplicative Homomorphism?

$c_1 \cdot c_2$?

$$(c_1 \cdot c_2) \cdot \vec{t} = c_1 \cdot \mu_2 \cdot \vec{t}$$
$$= \mu_2 \cdot (c_1 \cdot \vec{t})$$
$$= \mu_2 \cdot \mu_1 \cdot \vec{t}$$

Multiplicative Homomorphism?

$c_1 \cdot c_2$?

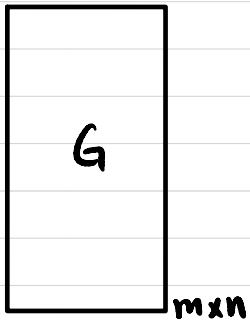
$$(c_1 \cdot c_2) \cdot \vec{t} = c_1 \cdot (\mu_2 \cdot \vec{t} + \vec{e}_2)$$
$$= \mu_2 \cdot c_1 \cdot \vec{t} + c_1 \cdot \vec{e}_2$$
$$= \mu_2 \cdot (\mu_1 \cdot \vec{t} + \vec{e}_1) + c_1 \cdot \vec{e}_2$$
$$= \mu_2 \cdot \mu_1 \cdot \vec{t} + \mu_2 \cdot \vec{e}_1 + c_1 \cdot \vec{e}_2$$

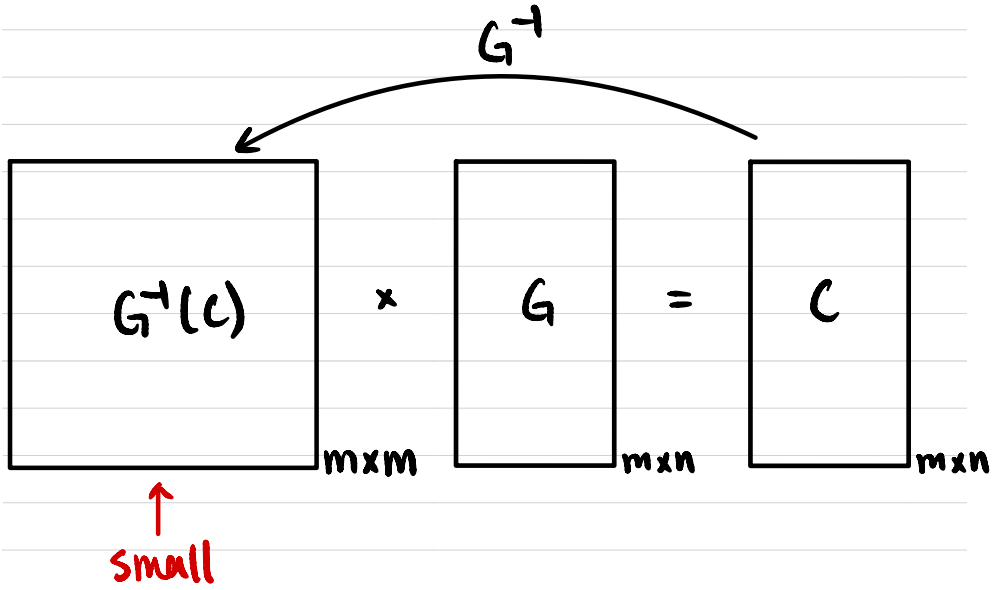
SWHE from LWE (GSW)

Attempt 2 (secret-key)

Flattering Gadget:

Gadget matrix $G \in \mathbb{Z}_q^{m \times n}$



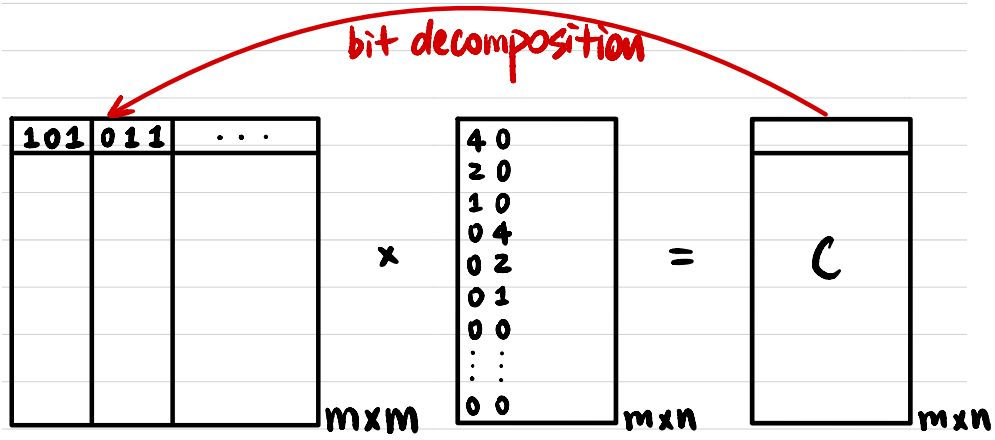


Inverse transformation

$$G^{-1}: \mathbb{Z}_q^{m \times n} \to \mathbb{Z}_q^{m \times m}$$

$\forall C \in \mathbb{Z}_q^{m \times n}$, $G^{-1}(C) = \text{small}$

$G^{-1}(C) \cdot G = C$



$m = n \cdot \log q$

SWHE from LWE (GSW)

Attempt 2 (secret-key)

$$sk = t_{n\times 1} \begin{bmatrix} s \\ 1 \end{bmatrix}_{n\times 1}$$

$Enc_{sk}(\mu)$: $\mu \in \{0,1\}$

Sample $C_0 \in \mathbb{Z}_q^{m\times n}$ s.t. $C_0 \cdot \vec{t} = small$

$$\underset{m\times n}{C_0} \times \underset{n\times 1}{t} = \underset{m\times 1}{e}$$

$$C = C_0 + \mu \cdot G$$

↑ gadget matrix

$Dec_{sk}(C)$: $C \cdot \vec{t} = (C_0 + \mu \cdot G)\cdot \vec{t}$

$= small + \mu \cdot (G \cdot \vec{t})$

CPA Security?

Homomorphism: $C_1 \cdot \vec{t} = \mu_1 \cdot (G \cdot \vec{t}) + \vec{e}_1$

$C_2 \cdot \vec{t} = \mu_2 \cdot (G \cdot \vec{t}) + \vec{e}_2$

Additive Homomorphism?

$C_1 + C_2$?

$(C_1 + C_2)\cdot \vec{t} = (\mu_1 + \mu_2)\cdot (G\cdot \vec{t}) + (\vec{e}_1 + \vec{e}_2)$

Multiplicative Homomorphism?

$G^{-1}(C_1)\cdot C_2$?

$G^{-1}(C_1)\cdot C_2 \cdot \vec{t} = G^{-1}(C_1)\cdot (\mu_2 \cdot (G\cdot \vec{t}) + \vec{e}_2)$

$= \mu_2 \cdot G^{-1}(C_1)\cdot G \cdot \vec{t} + G^{-1}(C_1)\cdot \vec{e}_2$

$= \mu_2 \cdot C_1 \cdot \vec{t} + G^{-1}(C_1)\cdot \vec{e}_2$

$= \mu_2 \cdot \mu_1 \cdot (G\cdot \vec{t}) + \mu_2 \cdot \vec{e}_1 + G^{-1}(C_2)\cdot \vec{e}_2$

How homomorphic is it?

$\#MULT \sim \log_m q$

SWHE from LWE (GSW)

Attempt 3 (public-key)

$sk = t_{n\times 1}$ 

public key: "encryptions of 0"

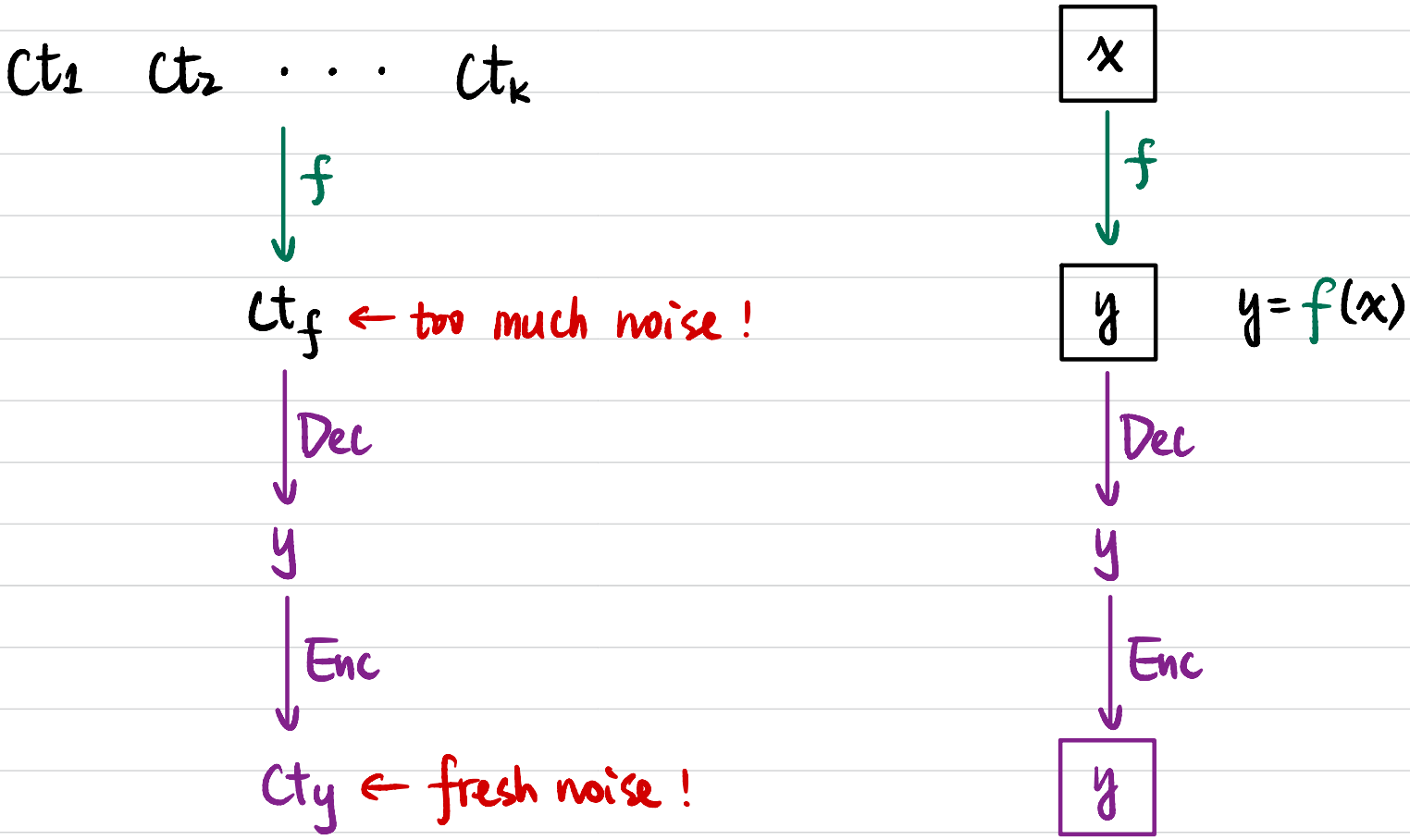
$$\left\{ c_0^i \in \mathbb{Z}_q^{m\times n} \;\middle|\; c_0^i \cdot \vec{t} = \text{small} \right\}_{i\in[n]}$$

$Enc_{sk}(\mu)$: $\mu \in \{0,1\}$

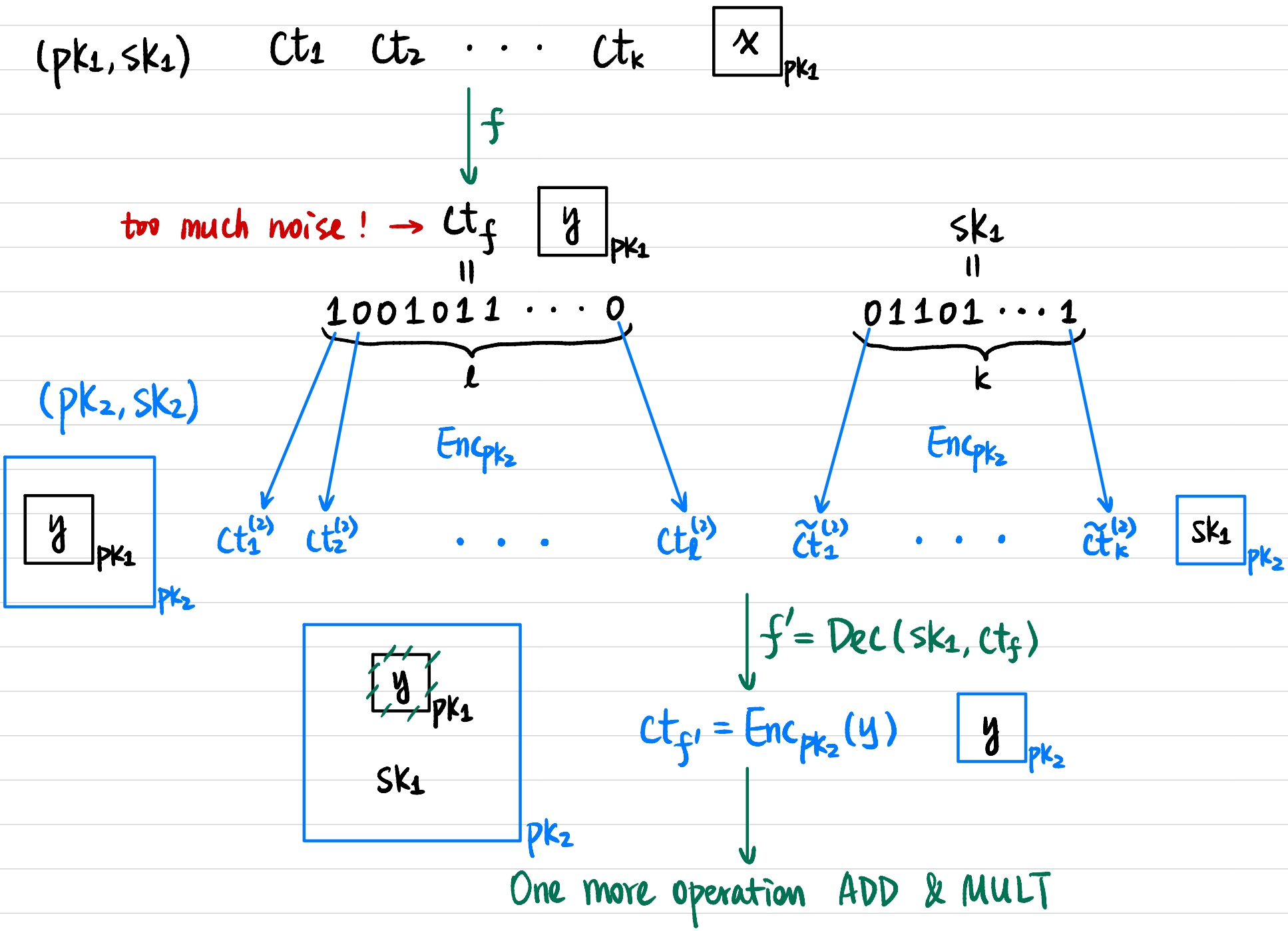
$C =$ (random subset sum of c_0^i's) $+ \mu \cdot G$

G: gadget matrix

Step 2: Bootstrapping



Leveled FHE


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

Step 2: Bootstrapping

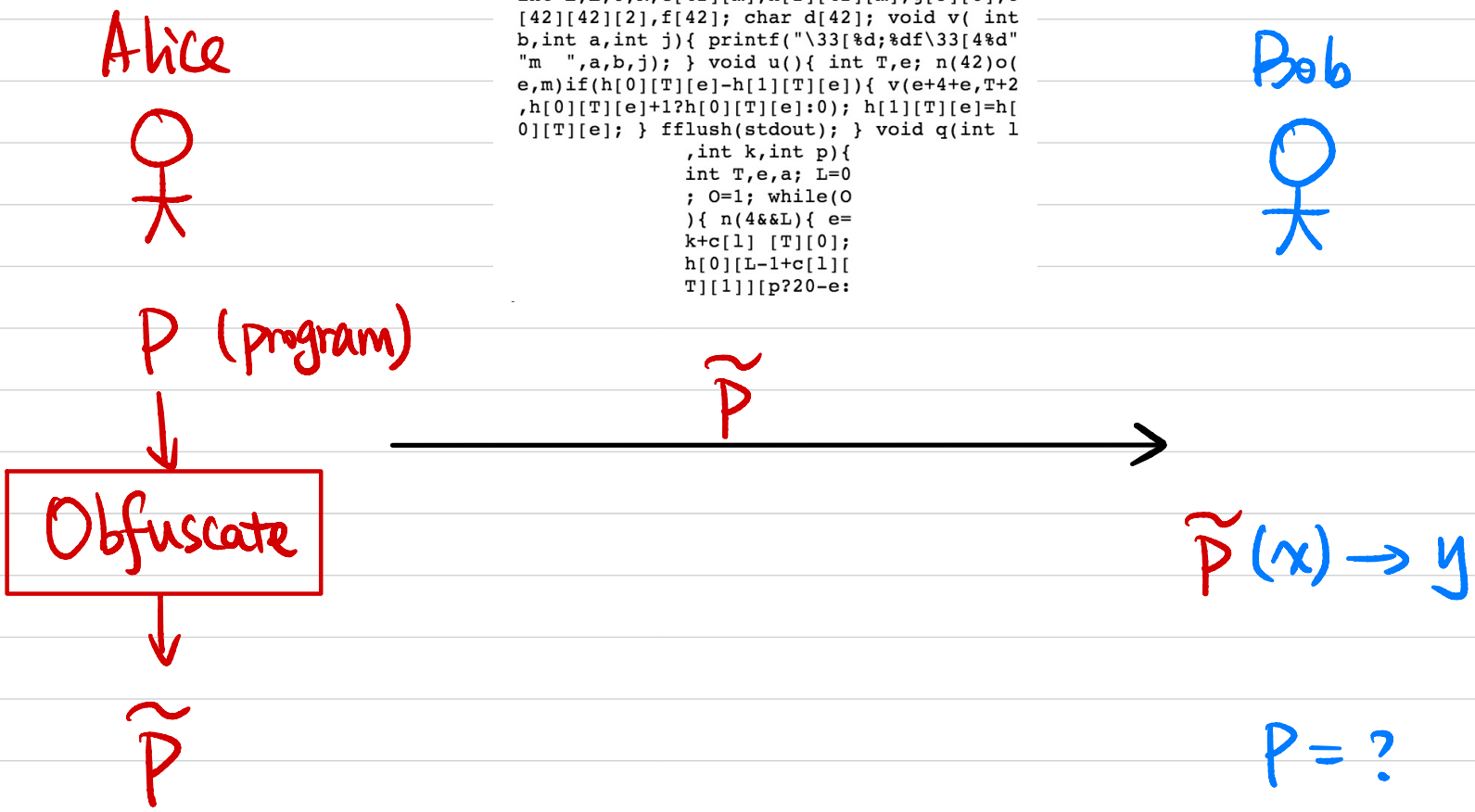
Leveled FHE: pk_1, pk_2, pk_3, . . . , pk_n

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

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

"circular secure" assumption

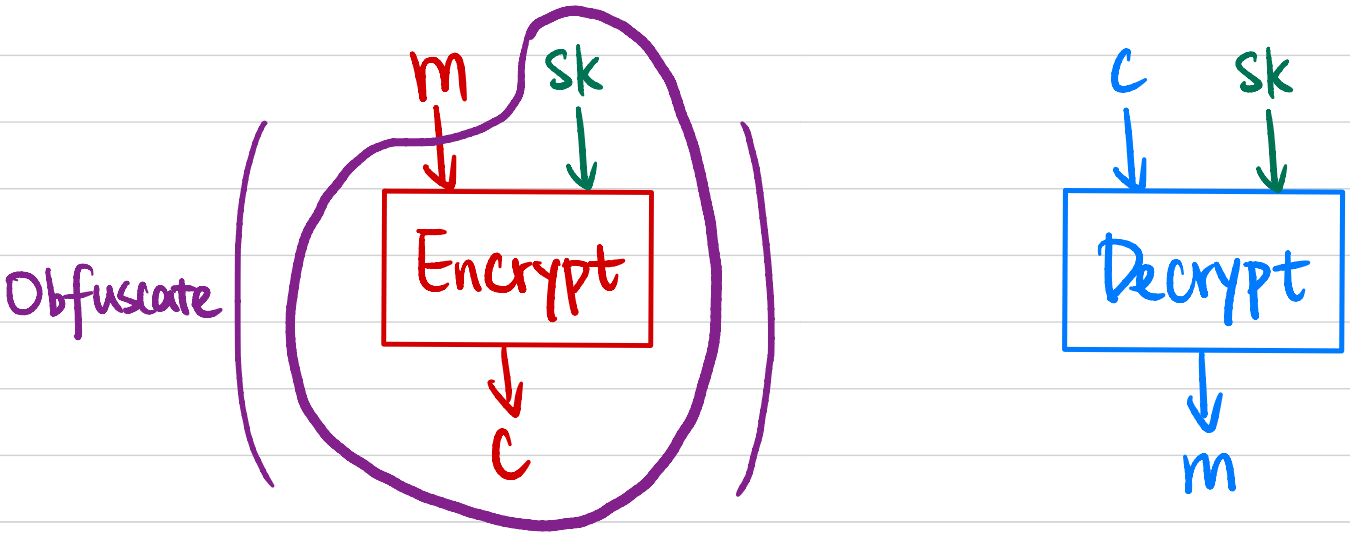
Program Obfuscation




```
int E,L,O,R,G[42][m],h[2][42][m],g[3][8],c
[42][42][2],f[42]; char d[42]; void v( int
b,int a,int j){ printf("\33[%d;%df\33[4%d"
"m  ",a,b,j); } void u(){ int T,e; n(42)o(
e,m)if(h[0][T][e]-h[1][T][e]){ v(e+4+e,T+2
,h[0][T][e]+1?h[0][T][e]:0); h[1][T][e]=h[
0][T][e]; } fflush(stdout); } void q(int l
                ,int k,int p){
                int T,e,a; L=0
                ; O=1; while(O
                ){ n(4&&L){ e=
                k+c[l] [T][0];
                h[0][L-1+c[l][
                T][1]][p?20-e:
```

Goal: Make the program "unintelligible" without affecting its functionality.

Symmetric-Key to Public-Key

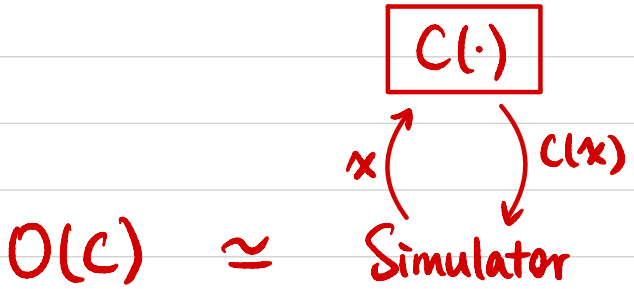


Formal Definition: Virtual Black Box (VBB)

Obfuscator O: $C \xrightarrow{O} O(C)$

- **Functionality:** $O(C)$ computes the same function as C.
- **Polynomial Slowdown:** $|O(C)| \leq \text{poly}(n) \cdot |C|$
- **Security (Virtual Black Box):**

  $\forall$ PPT A, $\exists$ PPT S, s.t. $\forall C$, $A(O(C)) \overset{c}{\simeq} S^{C(\cdot)}(1^{|C|})$.



Thm VBB obfuscator for all poly-sized circuits is impossible to achieve.

Formal Definition: Indistinguishability Obfuscation (iO)

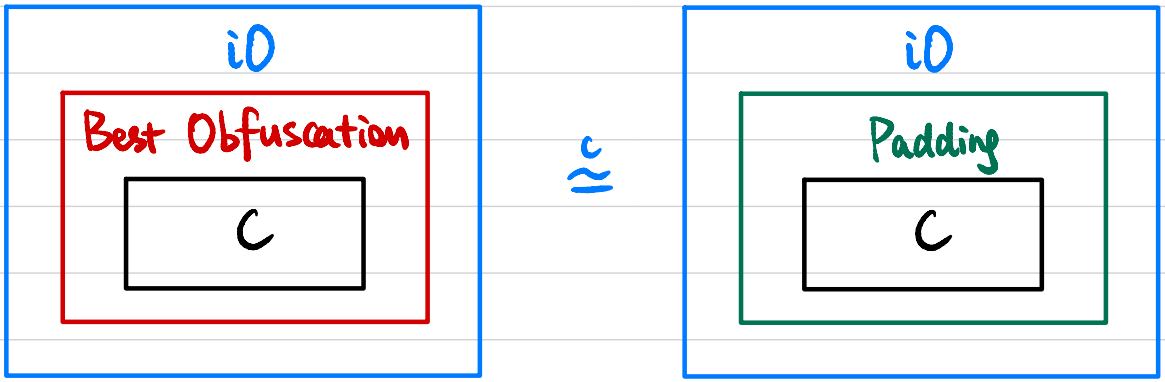
Obfuscator O: $C \xrightarrow{O} O(C)$

- **Functionality:** $O(C)$ computes the same function as C.
- **Polynomial Slowdown:** $|O(C)| \leq \text{poly}(n) \cdot |C|$
- **Security (indistinguishability obfuscation):**

  If C_0 & C_1 compute the same function and $|C_0| = |C_1|$.

  then $O(C_0) \overset{c}{\simeq} O(C_1)$

- **Best Possible Obfuscation**



Is it possible?

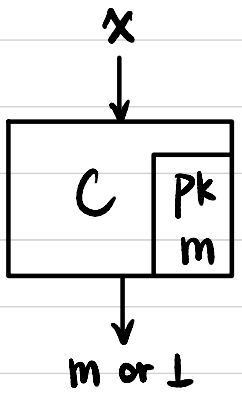
- 2001: Notion introduced
- 2013: Fist "candidate" construction from multilinear maps
- 2013-2020: Attack, fixes, new constructions from new assumptions
- 2020: New construction from well-founded assumptions

PKE from iO

Let $G: \{0,1\}^n \to \{0,1\}^{2n}$ be a length-doubling PRG.

- $\text{Gen}(1^n)$:
  $sk \xleftarrow{\$} \{0,1\}^n$
  $pk := G(sk)$

- $\text{Enc}_{pk}(m)$:
  $C_{pk,m}(x) := \begin{cases} m & \text{if } G(x) = pk \\ \perp & \text{otherwise} \end{cases}$
  Output $ct \leftarrow iO(C_{pk,m})$



- $\text{Dec}_{sk}(ct)$: $ct(sk) \to m$

Thm If G is a PRG and $iO(\cdot)$ is an idistinguishability obfuscator, then this PKE scheme is CPA-secure.

H_0: $sk \xleftarrow{\$} \{0,1\}^n$

$pk := G(sk)$

$C_{pk,m}(x) := \begin{cases} m & \text{if } G(x) = pk \\ \perp & \text{otherwise} \end{cases}$

Output $c \leftarrow iO(C_{pk,m})$

PRG

H_1: $pk \xleftarrow{\$} \{0,1\}^{2n}$

$C_{pk,m}(x) := \begin{cases} m & \text{if } G(x) = pk \\ \perp & \text{otherwise} \end{cases}$

Output $c \leftarrow iO(C_{pk,m})$

Stat. close

H_2: $pk \xleftarrow{\$} \{0,1\}^{2n}$

$C'_{pk,m}(x) := \perp$

Output $c \leftarrow iO(C'_{pk,m})$

$iO(C'_{pk,m_0}) \overset{c}{\approx} iO(C'_{pk,m_1})$

Thank You ^^