

Zeroizing attacks against Evasive and Circular Evasive LWE

Shweta Agrawal

IIT Madras

Anuja Modi

IIT Madras

Anshu Yadav

IST Austria

Shota Yamada

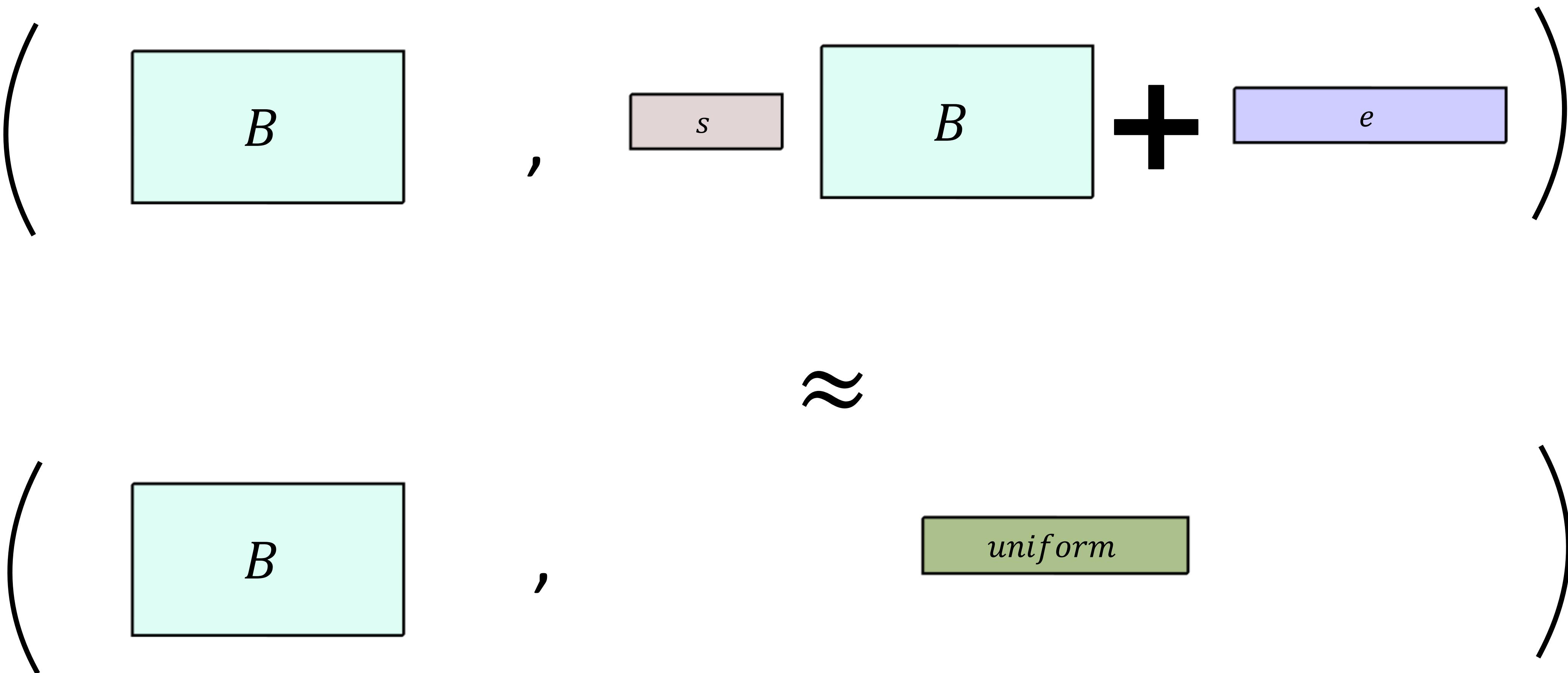
AIST Tokyo

@SG CRYPTO workshop

Slides mostly made by Shweta and Anuja,
with multiple changes added here and there

Learning With Errors Assumption (LWE) [Reg05]

Let $B \leftarrow \mathbb{Z}_q^{n \times m}, s \leftarrow \mathbb{Z}_q^n, e \leftarrow \mathcal{X}_q^m$



Evasive LWE [Wee22, Tsa22]

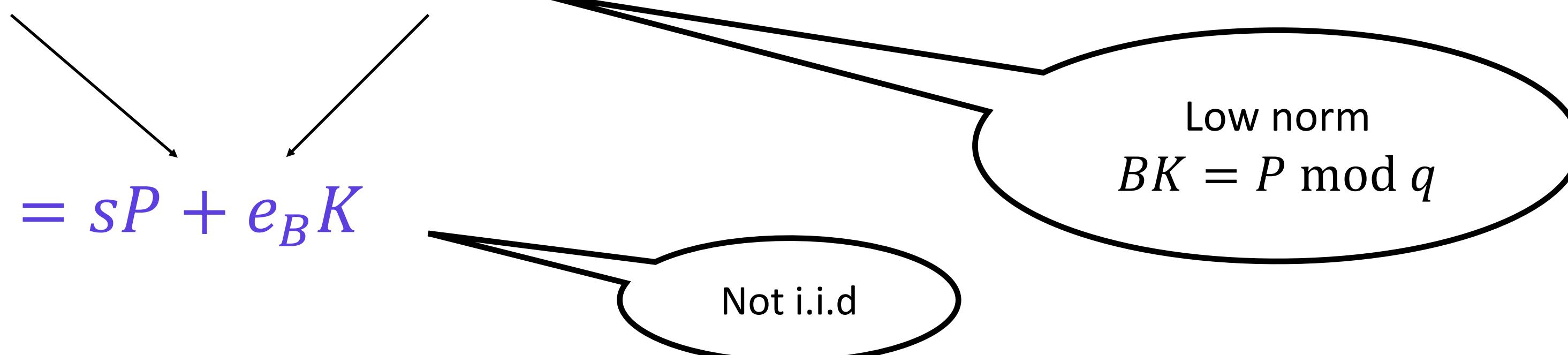
$(P, aux) \leftarrow Samp$

If

$$(B, P, sB + e_B, sP + e_P, aux) \approx (B, P, \$, \$, aux)$$

Then

$$(B, P, sB + e_B, K = B^{-1}(P), aux) \approx (B, P, \$, K = B^{-1}(P), aux)$$



Evasive LWE [Wee22, Tsa22]

$(P, aux) \leftarrow Samp$

Public-coin: Adv knows Sampler's random coins

Private-coin: Adv does not know Sampler's random coins.

If

$(B, P, sB + e_B, sP + e_P, aux)$

i.i.d

$\approx (B, P, \$, \$, aux)$

Insecure in general
(Wee22, VWW22,
BUW24, BDJ+24,
HHY25).

Then

$(B, P, sB + e_B, K = B^{-1}(P), aux)$

$\approx (B, P, \$, K = B^{-1}(P), aux)$

$= sP + e_B K$

Low norm
 $BK = P \bmod q$

Not i.i.d

Applications of Evasive LWE

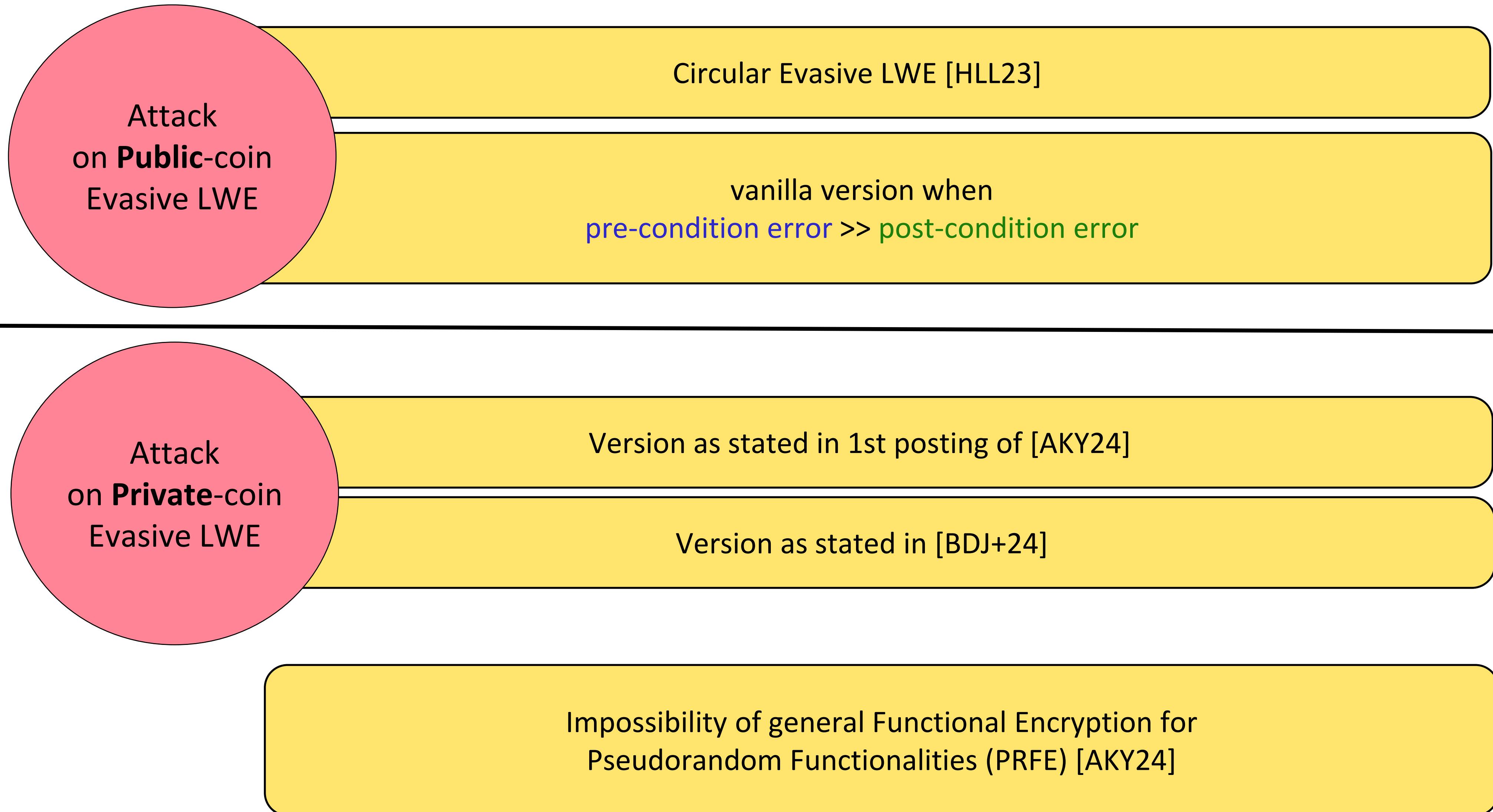
- Optimal Broadcast Encryption [Wee22]
- Witness Encryption [Tsa22, VWW22]
- Unbounded depth ABE for circuits [HLL23]
- Optimal Broadcast and Trace [AKYY23]
- Constant-input Attribute Based Encryption [ARYY23]
- ABE for Turing Machines from Lattices [AKY24]
- Adaptively secure ABE from WE [WW24]
- Multi-authority ABE from lattices without random oracles [WWW22]
- Adaptively sound zero-knowledge SNARKS for UP [MPV24]
- SNARGs for NP [JKLM24]
- Pseudorandom Obfuscation [DJM⁺25]
- Pseudorandom Functional Encryption [AKY24]
- Succinct iO for Turing Machines [JJMP25]

Applications of Evasive LWE

- Optimal Broadcast Encryption [Wee22]
- Witness Encryption [Tsa22, VWW22]
- Unbounded depth ABE for circuits [HLL23]
- Optimal Broadcast and Trace [AKYY23]
- Constant-input Attribute Based Encryption [ARYY23]
- ABE for Turing Machines from Lattices [AKY24]
- Adaptively secure ABE from WE [WW24]
- Multi-authority ABE from lattices without random oracles [WWW22]
- Adaptively sound zero-knowledge SNARKS for UP [MPV24]
- SNARGs for NP [JKLM24]
- Pseudorandom Obfuscation [DJM⁺25]
- Pseudorandom Functional Encryption [AKY24]
- Succinct iO for Turing Machines [JJMP25]

Only handful from public coin!

Our Results



Comparison with Concurrent and Independent Work

Our results	[HJL25] attack	[DJMMV25] attack
<p>Attack on Public-coin Evasive LWE</p>	<p>Circular Evasive LWE [HLL23]</p> <p>vanilla version when pre-condition error >> post-condition error</p>	<p>None</p>
<p>Attack on Private-coin Evasive LWE</p>	<p>Version as stated in 1st posting of [AKY24]</p> <p>Version as stated in [BDJ+24]</p>	<p>=</p> <p>1st version of [AKY24]</p> <p>1 st version of [BDJ+24] (Mentioned)</p>
	<p>Impossibility of general Functional Encryption for Pseudorandom Functionalities (PRFE) [AKY24]</p>	<p>None</p>

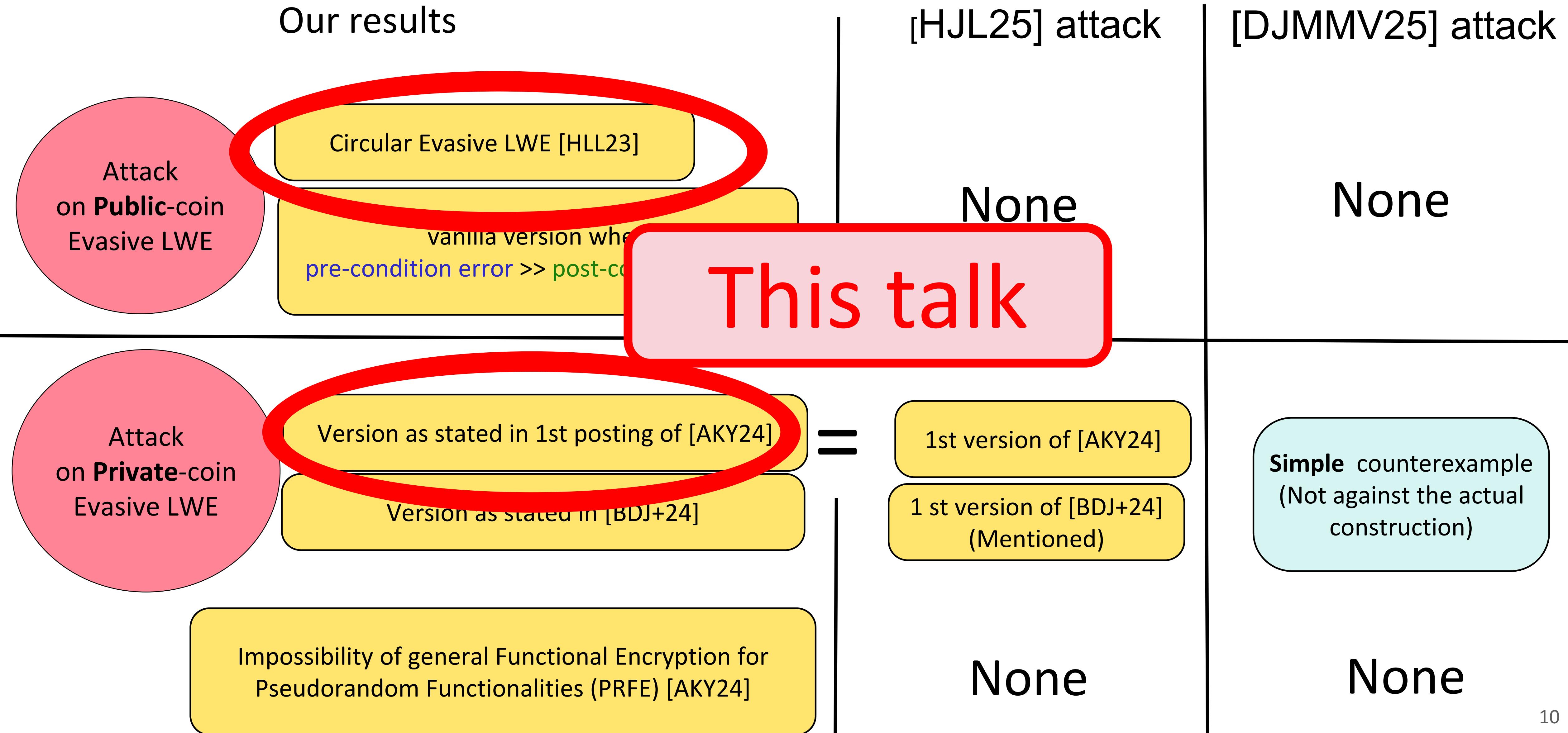
More on the Comparison: [DJMMV25] and Ours/[HJL25]

Classification of the private-coin evasive by BUW: Whether B and P are given or not

		(B, \neg P)	(B, P)
[DJMMV25]	Ours	[DJMMV25]	Ours
Broken by [BUW24]	(easy modification)		
		(\neg B, \neg P)	(\neg B, P)
[DJMMV25]	Ours	[DJMMV25]	Ours
	(easy modification)		

Ours: Against specific scheme 1st version of [AKY24]/DJMMV25: Not for a scheme

Comparison with Concurrent and Independent Work



Attack on Private-coin Evasive LWE as stated in [AKY24]

Prelims for attack

Recall: GSW FHE



Approximate Decryption is inner Product :

$$sE_{pkfhe}(f(x)) = e_{fhe} + f(x)$$

Notation:

$$\hat{x} := E_{pkfhe}(x), \hat{f}(ct) := \text{Eval}_{pkfhe}(ct)$$

$$\text{Hence, } \hat{f}(\hat{x}) = \widehat{f(x)}, s\widehat{f(x)} = e_{fhe} + f(x)$$

Prelims for attack

Recall: [BGG+14] Encoding

Encoding of attribute x :

Public matrix

$$s(A - x \otimes G) + e_A$$

2 deterministic algo outputs:

$$H_f, H_{f,x}$$

publicly computable
& low norm

S.t $(A - x \otimes G)H_{f,x} = AH_f - f(x)$

Automatic Decryption [BTW17]

Reuse FHE secret key as BGG+14 LWE secret!

Prelims for attack

Recall: [BGG+14] Encoding

Encoding of attribute x : $s(A - x \otimes G) + e_A$

2 deterministic algo outputs: $H_f, H_{f,x}$

publicly computable & low norm

S.t $((A - x \otimes G) + e_A)H_{f,x} = AH_f - f(x)$

Automatic Decryption [BTWV17]

Reuse FHE secret key as BGG+14 LWE secret!

$$\begin{aligned} & (s(A - \hat{x} \otimes G) + e_A)H_{\hat{f}, \hat{x}} \\ &= s\hat{A} - s\widehat{f(x)} + e_A H_{\hat{f}, \hat{x}} \\ &= s\hat{A} - f(x) + e_{fhe} + e_A H_{\hat{f}, \hat{x}} \end{aligned}$$

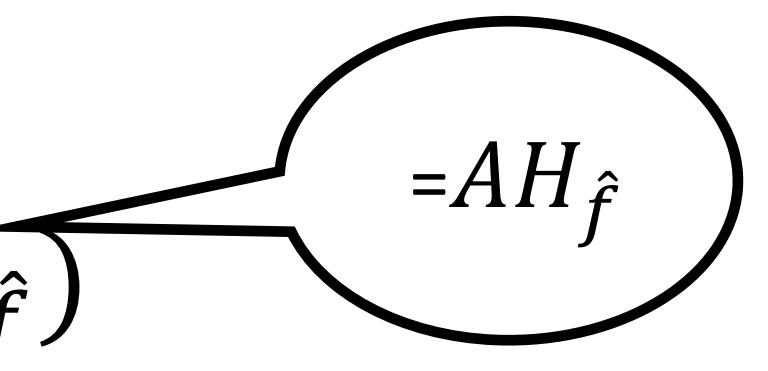
Mask Error

\hat{x}, \hat{f} are FHE CT and homomorphic evaluation resp.

Prelims for attack

[AKY24] PRFE construction

$$ct(x): \underbrace{c_B = sB + e_B}_{\text{LWE instance}}, c_A = s(A - X \otimes G) + e_A, X = E_{pk_{fhe}}(x)$$

$$sk_f: K \leftarrow B^{-1}(A_{\hat{f}})$$


\hat{f} :Homomorphically compute $f(x)$

$$\text{Dec: } c_B K - c_A H_{\hat{f},X} = sA_{\hat{f}} + e_B K - sA_{\hat{f}} + f(x) + e_{fhe} - e_A H_{\hat{f},X}$$

$$= f(x) + \underbrace{e_B K - e_A H_{\hat{f},X}}_{\text{Can extract}} + e_{fhe}$$

Can extract
Approximately
(i.e., higher bits)

Hope is $f(x)$ floods error - vulnerability

Prelims for attack

[AKY24] PRFE security definition

If $f(x)$ is pseudorandom given aux

Then CT is pseudorandom, given aux & sk

Security proof

Invoke Evasive LWE w.r.t the sampler:

Samp:

1. Compute PRFE $\text{ct}(x)$
2. $K = B^{-1}(P)$ & $P = AH_{\hat{f}}$
3. Output (P, aux) , $\text{aux} = (X, c_A, f, \text{other info})$

Prelims for attack

[AKY24] PRFE security definition

If $f(x)$ is pseudorandom given aux

Then CT is pseudorandom, given aux & sk

Security proof

Invoke Evasive LWE w.r.t the sampler:

Suffices to prove pre-condition i.e.

i.i.d

$$(aux, B, P, A, f, c_B, c_A, X, c_P = sP + e_P)$$

$$\approx (aux, B, P, A, f, c_B, c_A, X, c_A H_{\hat{f}, X} + f(x) + e_P) \quad \because \text{By flooding}$$

$$\approx (aux, B, P, A, f, \$, \$, \$, \text{known terms} + f(x)) \quad \because \text{By LWE}$$


$$\approx (aux, B, P, A, f, \$, \$, \$, \$) \quad \because \text{By the pseudorandomness of } f$$

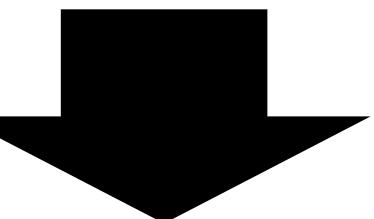
Samp:

1. Compute PRFE ct(x)
2. $K = B^{-1}(P) \& P = AH_{\hat{f}}$
3. Output (P, aux) , $aux = (X, c_A, f, other info)$

Prelims for attack

Take any function f

Recall we have $sE_{pk_{fhe}}(f(x)) = e_{fhe} + f(x)$



Can choose contrived circuit implementation of f (following the idea of [HJL21])



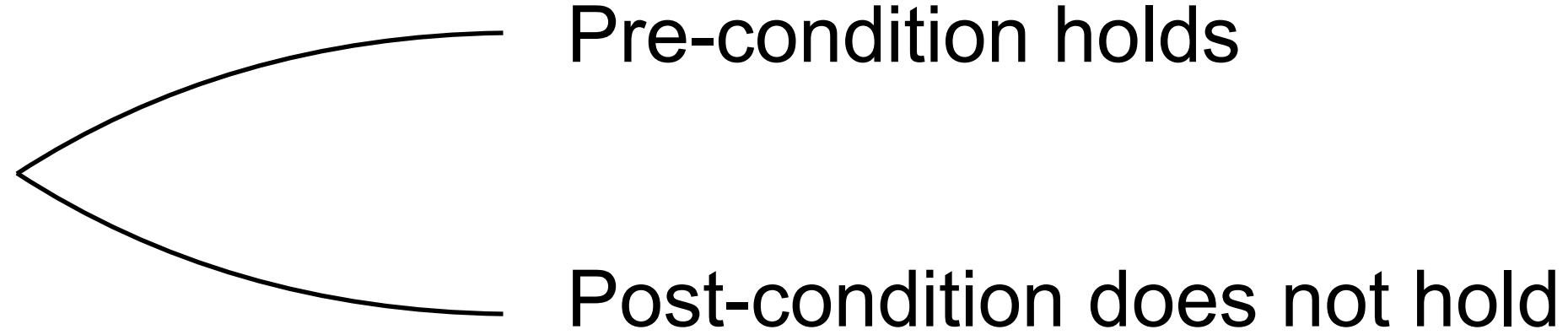
We have $e_{fhe} \equiv f(x) \pmod{2}$

Correlation between the encrypted value
and the noise/error!

Attack against [AKY24] sampler

To show attack, we need to prove

For $(P, aux) \leftarrow Samp$



Samp:

1. Compute PRFE $ct(x)$
2. $K = B^{-1}(P) \& P = AH_{\hat{f}}$
3. Output (P, aux) , $aux = (X, c_A, f, other info)$

1. Pre-condition holds

$$(B, P, sB + e_B, sP + e_P, aux)$$

\approx

$$(B, P, \$, \$, aux)$$



::By AKYY PRFE
security

2. Post-condition is distinguishable

$$(B, P, sB + e_B, K = B^{-1}(P), aux)$$

$\not\approx$

$$(B, P, \$, K = B^{-1}(P), aux)$$

Attack against [AKY24] sampler

Distinguishing post-condition

Given $(B, P, c_B, c_A, X, K = B^{-1}(P))$, distinguisher tries to distinguish if

$$c_B = sB + e_B, c_A = s(A - X \otimes G) + e_A, X = E_{pk_{fhe}}(x)$$

Or $c_B = \$, c_A = \$, X = \$$

Distinguishing strategy

1.

Compute $v = c_B K - c_A H_{\hat{f},X} \bmod q$

If $\underbrace{v}_{\text{Pseudorandom over } \mathbb{Z}_q} = \underbrace{f(x) + e_B K - e_A H_{\hat{f},X} + e_{fhe}}_{\text{Small } \ll q}$

PRFE Dec
eq

Pseudorandom
over \mathbb{Z}_q

Small $\ll q$

Key observation:

Wraparound occurs only with negl prob

\Rightarrow Can retrieve the value **over the integer** (w.h.p)

Attack against [AKY24] sampler

Distinguishing strategy

2.

Get $v = f(x) + e_B K + e_{fhe} - e_A H_{\hat{f}, X}$

Get the value over
the integers

Cannot separate $f(x)$: lower order bits mask error terms

Idea of [HJL21]

$$v = \cancel{f(x)} + e_B K + \cancel{e_{fhe}} - e_A H_{\hat{f}, X} \quad mod 2$$

choose contrived
ckt implementing
homomorphic
computation of
PRG

3.

Distinguisher solves linear eq for e_B and e_A , outputs



if solution is found.

Else outputs



w.h.p

Attack against [AKY24] sampler

Distinguishing strategy

2.

Get $v = f(x) + e_B K + e_{fhe} - e_A H_{\hat{f}, X}$

Get the value over
the integers

Cannot separate $f(x)$: lower order bits mask error terms

Idea of [HJL21]

$$v = \cancel{f(x)} + e_B K + \cancel{e_{fhe}} - e_A H_{\hat{f}, X} \quad mod 2$$

choose contrived
ckt implementing
homomorphic
computation of
PRG

Hence, attack against private-coin Evasive LWE
assumption used by 1st version of [AKY24] is found.

3.

Distinguisher solves linear eq

w.h.p

Attack on Circular Evasive LWE [HLL23]

Circular Evasive LWE Assumption [HLL23]

If

$$(B, pk_{fhe}, A, c_B = \textcolor{red}{s}B + e_B, c_A = \textcolor{red}{s}(A - \textcolor{red}{S} \otimes G) + e_A, \textcolor{red}{S} = \mathsf{E}_{pk_{fhe}}(\textcolor{red}{s}), c_P = \textcolor{red}{s}P + e_P, aux)$$

\approx

$$(B, \$, A, c_B = \$, c_A = \$, S = \$, c_P = \$, aux)$$

Circular Evasive LWE Assumption [HLL23]

If

$$(B, pk_{fhe}, A, c_B = \textcolor{red}{s}B + e_B, c_A = \textcolor{red}{s}(A - \textcolor{red}{S} \otimes G) + e_A, \textcolor{red}{S} = \mathsf{E}_{pk_{fhe}}(\textcolor{red}{s}), c_P = \textcolor{red}{s}P + e_P, aux)$$

\approx

$$(B, \$, A, c_B = \$, c_A = \$, S = \$, c_P = \$, aux)$$

Then

$$(B, pk_{fhe}, A, c_B = \textcolor{red}{s}B + e_B, c_A = \textcolor{red}{s}(A - \textcolor{red}{S} \otimes G) + e_A, \textcolor{red}{S} = \mathsf{E}_{pk_{fhe}}(\textcolor{red}{s}), K = B^{-1}(P), aux)$$

\approx

$$(B, \$, A, c_B = \$, c_A = \$, S = \$, K = B^{-1}(P), aux)$$

Where $(A, P, aux) \leftarrow Samp(1^\lambda; coins_{pub})$

Comparing Evasive LWE as in [AKY24] and Circular Evasive LWE

Terms in LHS of precondition:

Private-coin Evasive LWE	Circular Evasive LWE
$c_B = sB + e_B$	$c_B = sB + e_B$
$c_A = s(A - S \otimes G) + e_A$	$c_A = s(A - S \otimes G) + e_A$
$X = \mathsf{E}_{pk_{fhe}}(x)$	$S = \mathsf{E}_{pk_{fhe}}(s)$
$c_P = sP + e_P$	$c_P = sP + e_P$
$(A, P, aux) \leftarrow Samp(1^\lambda)$	$(A, P, aux) \leftarrow Samp(1^\lambda; coins_{pub})$

Circular Evasive LWE - public OR private coin?

categorized as “public-coin” in [HLL23, BDJ+24, CW25].

[BUW24] - does not fall in public-coin regime in strict sense.

Comparing Evasive LWE as in [AKY24] and Circular Evasive LWE

Terms in LHS of precondition:

Private-coin Evasive LWE	Circular Evasive LWE
$c_B = sB + e_B$	$c_B = sB + e_B$
$c_A = s(A - S \otimes G) + e_A$	$c_A = s(A - S \otimes G) + e_A$
$X = \mathsf{E}_{pk_{fhe}}(x)$	$S = \mathsf{E}_{pk_{fhe}}(s)$
$c_P = sP + e_P$	$c_P = sP + e_P$
$(A, P, aux) \leftarrow Samp(1^\lambda)$	$(A, P, aux) \leftarrow Samp(1^\lambda; coins_{pub})$

Circular Evasive LWE - public OR private coin?

categorized as “public-coin” in [HLL23, BDJ+24, CW25].

[BUW24] - does not fall in public-coin regime in strict sense.

We show attack against circular evasive LWE!

Circular Evasive LWE Assumption [HLL23]

If

$$(B, pk_{fhe}, A, c_B^\top = \textcolor{red}{s}B + e_B, c_A = \textcolor{red}{s}(A - \textcolor{red}{S} \otimes G) + e_A, \textcolor{red}{S} = \mathsf{E}_{pk_{fhe}}(\textcolor{red}{s}), c_P = \textcolor{red}{s}^\top P + e_P, aux)$$

$$(B, \$, A, c_B = \$, c_A = \$, S = x, c_P = \$, aux) \approx$$

Then

consider this as AKY ciphertext encrypting “s”

$$(B, pk_{fhe}, A, c_B^\top = \textcolor{red}{s}B + e_B, c_A = \textcolor{red}{s}(A - \textcolor{red}{S} \otimes G) + e_A, \textcolor{red}{S} = \mathsf{E}_{pk_{fhe}}(\textcolor{red}{s}), K = B^{-1}(P), aux)$$

$$(B, \$, A, c_B = \$, c_A = \$, S = \$, K = B^{-1}(P), aux) \approx$$

P set s.t. K is sk
for function f

Attack against post-condition same as for AKY

Proving Pre-condition: Overview

In AKY24,

$$(c_B, c_A, S, pk_{fhe}, f(x) = PRF(x)) \approx (c_B, c_A, S, pk_{fhe}, \$)$$

In HLL23,

$$(c_B, c_A, S, pk_{fhe}, f(s)) \stackrel{?}{\approx} (c_B, c_A, S, pk_{fhe}, \$)$$

Correlated with other terms!

Failed Idea : Let's make f randomized and set $f(s) = sF + \text{noise}$

Joint pseudorandomness follows from circular LWE

The randomness of f should be kept hidden – Sampler becomes private-coin!

Working Idea: $f(s)$ - learning with rounding instance [BPR12]

⇒ Derive the pseudo-randomness deterministically



Thank you!