Randomization in FHE and TFHE

FHE.org 2026: Mar 8th 2026

Speaker: Nigel P. Smart (Zama and KU Leuven)

Author: Nigel P. Smart and Michael Walter

Security and Correctness of FHE Schemes

In recent years it has become apparent that there are subtle issues with respect to security and correctness of FHE schemes.

In addition, randomization forms a key aspect of many of these issues

Randomization is, after all, inherent in the method for LWE encryption.

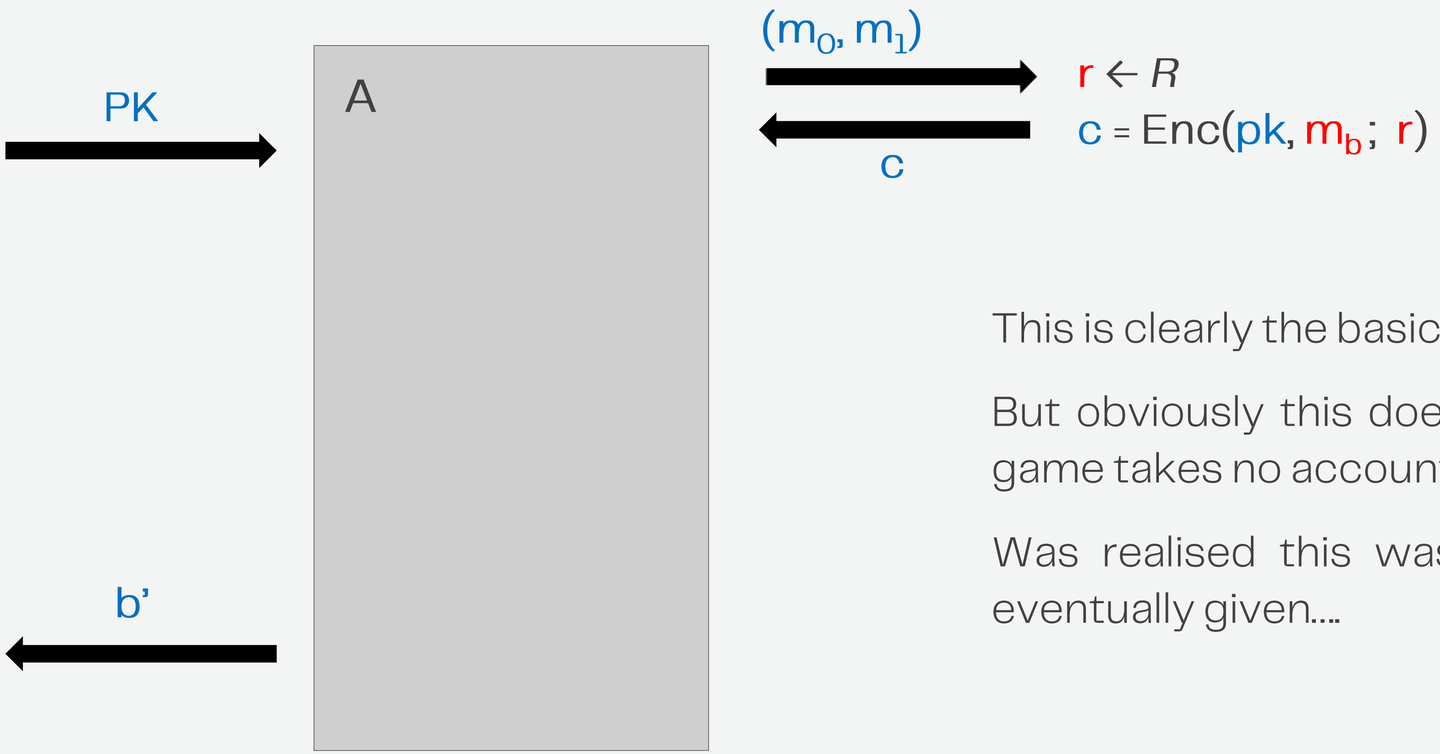
- It is needed to ensure both mask, the a values in LWE, and error values, the e values in LWE, are suitably random

Without good randomization a scheme is not secure

What is less obvious is that same is true for correctness, an issue which we will explore in this talk

Security Notions for FHE

FHE Security Model : IND-CPA



This is clearly the basic requirement

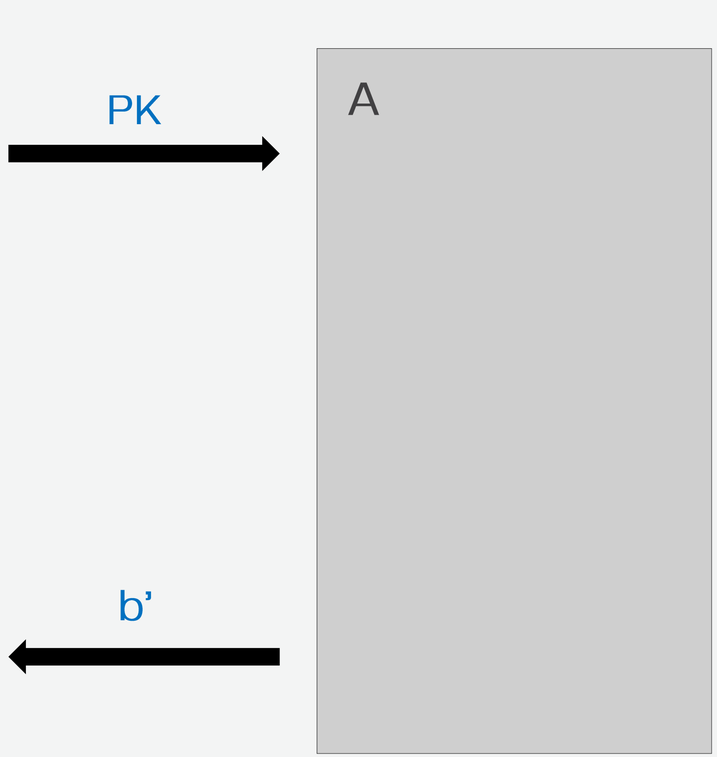
But obviously this does not really capture FHE at all, as the game takes no account of homomorphic evaluation queries.

Was realised this was not enough so a new model was eventually given....

Adversary wins the game if b' = b

4

FHE Security Model : IND-CPA-D



(m_0, m_1)

$r \leftarrow R$

$c = Enc(pk, m_b; r)$

c

$state \leftarrow state \cup \{ (m_0, m_1, c) \}$

j

If $state[j].m_0 \neq state[j].m_1$ then $m = \perp$

Else $m = Dec(state[j].c, sk)$

m

$(g, j_1, ..., j_m)$

$(c_1, ..., c_n) \leftarrow Eval(g, state[j_1].c, ..., state[j_m].c)$

$(m_{0,1}, ..., m_{0,n}) \leftarrow g\,(state[j_1].m_0, ..., state[j_m].m_0)$

$(m_{1,1}, ..., m_{1,n}) \leftarrow g\,(state[j_1].m_1, ..., state[j_m].m_1)$

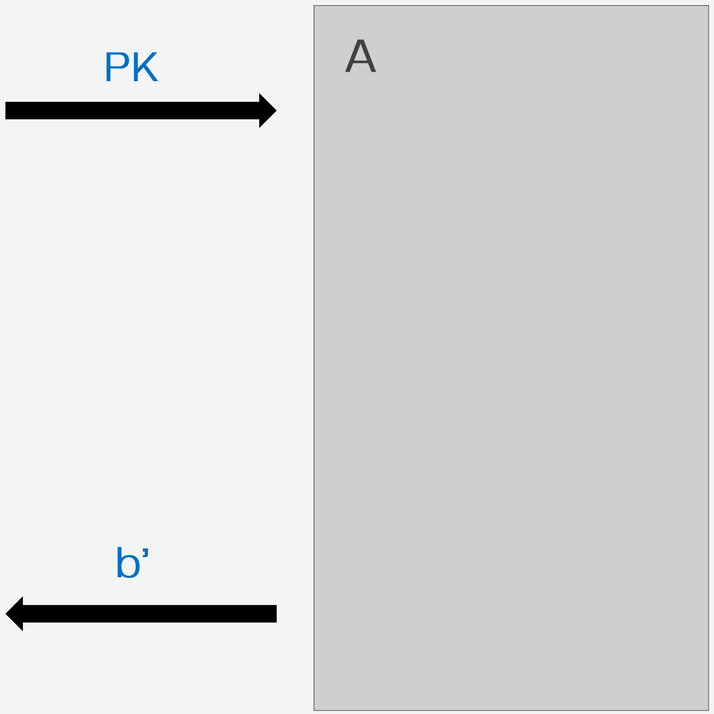
$state \leftarrow state \cup \{ (m_{0,k}, m_{1,k}, c_k) \}_{k=1,...,n}$

$(c_1, ..., c_n)$

Adversary wins the game if $b' = b$

5

FHE Security Model : IND-CPA-D



This model is better as it captures issues related to evaluations of ciphertexts.

Note the adversary can select the functions to be evaluated, and on what ciphertexts.

- This captures real attacks on actual schemes which have been found in the literature

It does not however capture the ability for the adversary to generate his own ciphertexts with randomness that he controls

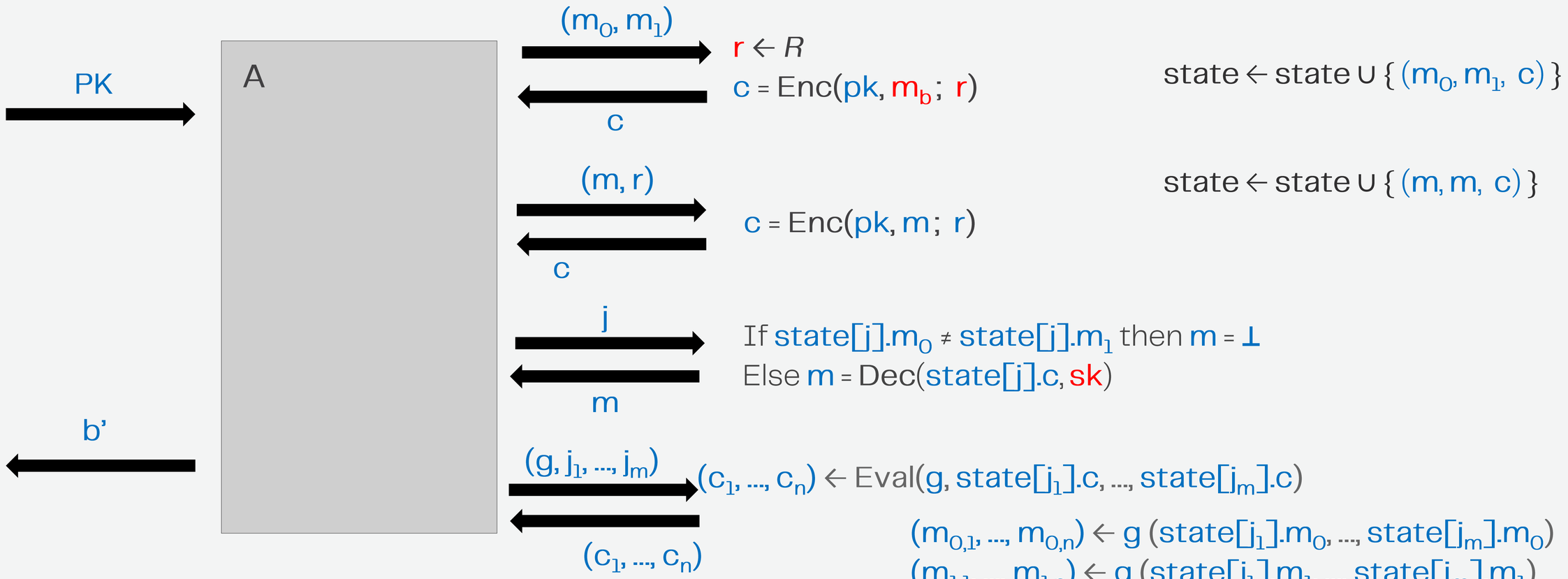
- Thus IND-CPA-D is really about secret key FHE and not public key FHE

This leads us to our final security notion (strong) IND-CPA-D.....

Adversary wins the game if b' = b

6

FHE Security Model : sIND-CPA-D



Adversary wins the game if b' = b

7

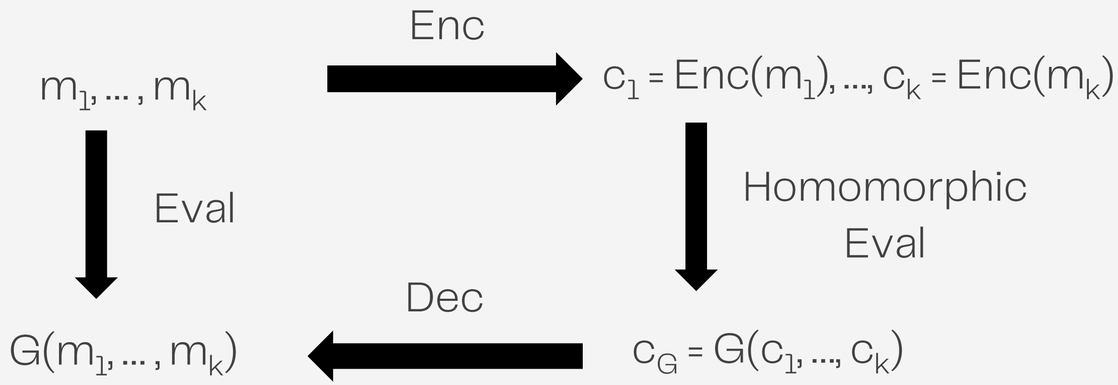
Correctness Notions for FHE

Traditional Correctness Notion

The traditional correctness notion for FHE is very simplistic

- If you encrypt some data, then apply a function homomorphically, and then decrypt, you should get the same result as if you had applied the function to the data in the clear.

Said another way, the following diagram holds with probability one (perfect correctness), or with probability $1 - \epsilon$ (statistical correctness)

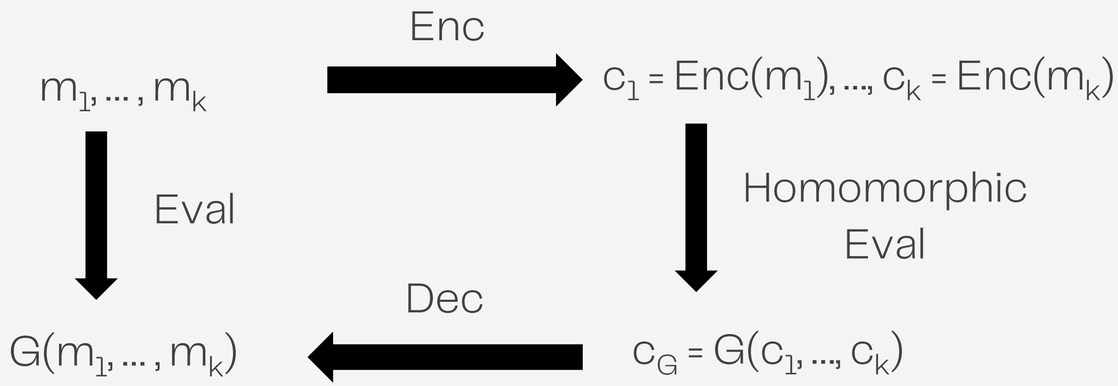


Traditional Correctness Notion

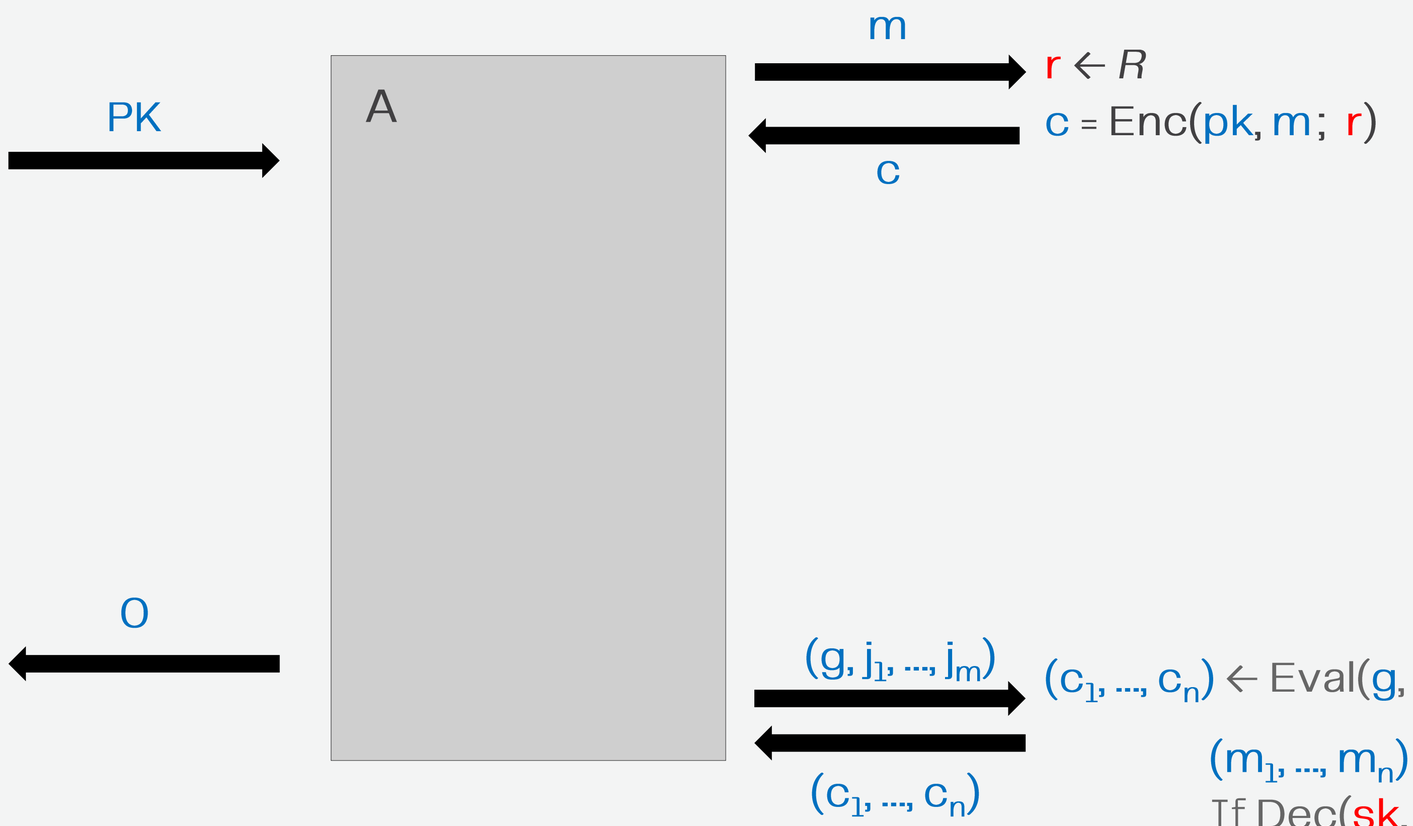
This is not however how FHE is used in practice

Adversaries can select functions to evaluate dependent on previous function evaluations, encrypt new data, even obtain selective decryptions

Thus, a reactive notion of correctness is needed.....



FHE Computational Correctness



state ← state ∪ { (m, c) }

If Dec(sk,c) ≠ m the game terminates and outputs 1

(m1, ..., mn) ← g (state[j1].m, ..., state[jm].m)
If Dec(sk, ci) ≠ mi for any i then terminate and output 1
state ← state ∪ { (mk, ck) }k=1,...,n

Adversary wins the game if the game outputs 1.

11

FHE Computational Correctness and IND-CPA-D

The computational correctness definition looks a lot like (bits of) the IND-CPA-D security model

In fact, the only thing missing is the notion of indistinguishable encryptions

Indeed, Alexandru et al 2024 proved the following....

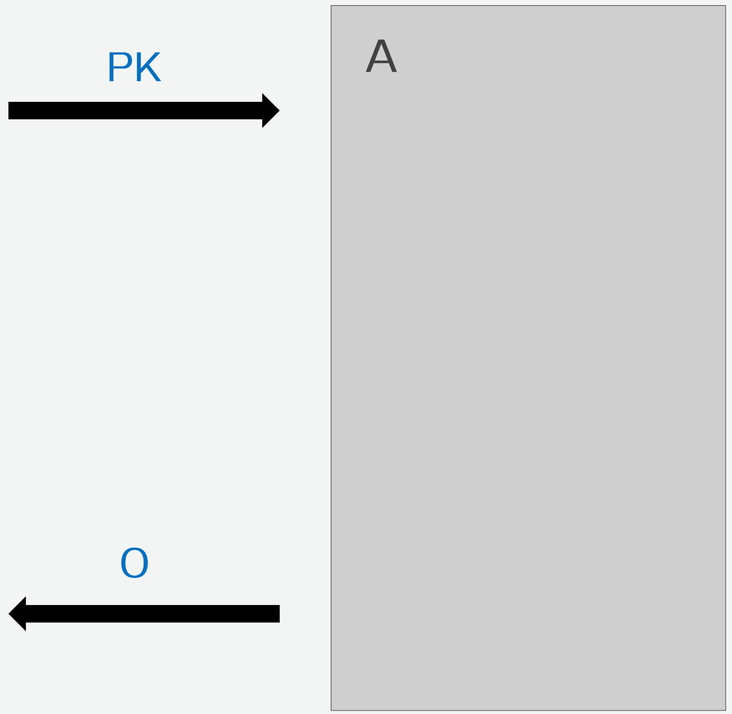
An FHE scheme which is IND-CPA and computationally correct is IND-CPA-D

So, the issue with IND-CPA-D security is really about poor correctness definitions in the past

Can we do a similar trick with sIND-CPA-D security?

12

FHE ACER Correctness



$c = Enc(pk, m; r)$

state ← state ∪ { (m, c) }

If Dec(sk,c) ≠ m the game terminates and outputs 1

$(c_1, ..., c_n) \leftarrow Eval(g, state[j_1].c, ..., state[j_m].c)$

$(m_1, ..., m_n) \leftarrow g\ (state[j_1].m, ..., state[j_m].m)$

If $Dec(sk, c_i) \neq m_i$ for any i then terminate and output 1

$state \leftarrow state \cup \{ (m_k, c_k) \}_{k=1,...,n}$

Adversary wins the game if the game outputs 1.

FHE ACER Correctness and sIND-CPA-D

In eprint 2025/2005 we show

An FHE scheme which is IND-CPA and ACER correct is sIND-CPA-D

This is done by modifying the argument from Bernard et al 2024

However, to achieve ACER correctness seems to require a randomized evaluation operation

- As the adversary can select functions to evaluate reactively one needs to ensure this does not alter probability distributions used to argue correctness.
- This means evaluation needs to be modified to be executed as follows

$$RandEval(g, c_1, \dots, c_m) = Eval\big(g, ReRand(c_1), \dots, ReRand(c_m)\big)$$

where ReRand(c_i) is a function which produces a ciphertext encrypting the same message as c_i

Note the evaluation is performed by the evaluator, and not the adversary, so the adversary does not know HOW the re-randomization was performed.

14

Setting FHE Parameters

When setting FHE parameters we also make a correctness assumption:

Parameters are set in a way which assumes the mask values are uniform and the error values satisfy some Gaussian assumption for specific basic operations

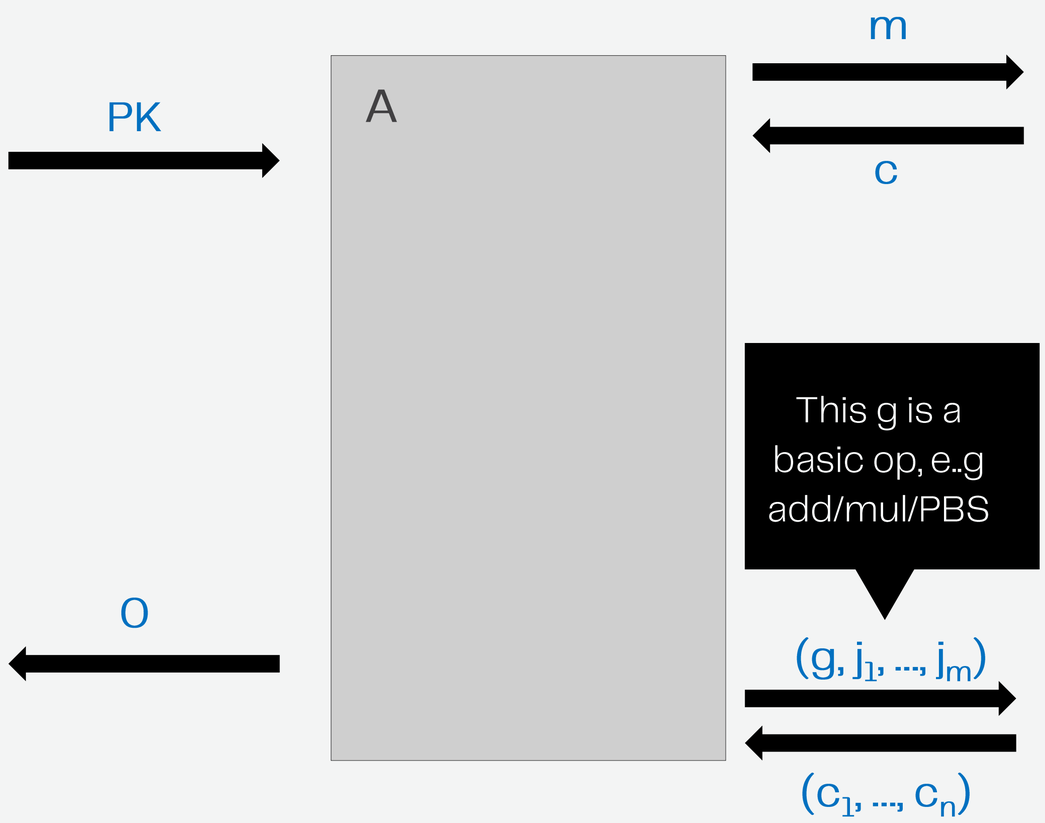
- Add and Mul in BGV, BFV
- PBS in TFHE

But we never actually use a single Add, Mul or PBS on their own. We chain these operations together.

So, when setting parameters, we assume that this chaining respects our assumptions on the mask and error values.

This correctness assumption we capture in a game we denote by AvgCor, as it captures the average case correctness assumptions people implicitly make when setting parameters.

AvgCor Correctness



state ← state ∪ { m }

The evaluation is performed on fresh encryptions of the underlying messages

$(c_1', ..., c_m') \leftarrow (Enc(state[j_1]),...,Enc(state[j_m])$

$(m_1,...,m_n) \leftarrow g(state[j_1],...,state[j_m])$

$state \leftarrow state \cup \{(m_1,...,m_n)\}$

$(c_1, ..., c_n) \leftarrow Eval(g, c_1', ..., c_m')$

If $Dec(sk, c_i) \neq m_i$ for any i then terminate and output 1

Adversary wins the game if the game outputs 1.

16

The Problem with sIND-CPA-D

To achieve sIND-CPA-D security at a security level of 128 bits one needs:

- IND-CPA security at a level of 128-bit security
- A failure probability for the evaluation operation of at most 2^{-128}, as per definition AvgCor
- The evaluation operation needs to be randomized (i.e. input ciphertexts are randomized before evaluation is performed).

This last point is a problem in practice as for verifiability one wants the FHE evaluation operation to be deterministic.

Can we de-randomize the randomization?

We show for TFHE this is possible in eprint 2025/2005

- Our method probably applies to other FHE schemes, but we utilize specific properties of TFHE in our proof

17

De-Randomization of TFHE

ReRandomization

$$RandEval(g, c_1, \dots, c_m) = Eval\big(g, ReRand(c_1), \dots, ReRand(c_m)\big)$$

Our goal is to de-randomize the above operation

For technical reasons we need each ciphertext to be accompanied by the ZK proof of correct encryption, when it is fresh encryption (i.e. not the output of an Eval operation)

- This is needed for the simulator to be able to extract the underlying message for adversarially produced ciphertexts.
- For outputs of Eval we assume this auxiliary information is $\perp$

19

DeRandomized – ReRandomization

Step 1

- On input of $\big((c_1, aux_1), \dots, (c_m, aux_m)\big)$ we compute for $\mathrm{i} = 1, \dots, \mathrm{m}$

$$seed_i \leftarrow \mathrm{Hash}\big((c_1, aux_1), \dots, (c_m, aux_m), i, g\big)$$

Note, the function, index and ALL inputs are passed into the hash function

Step 2

- Compute for $\mathrm{i} = 1, \dots, \mathrm{m}$ $\quad z_i \leftarrow Enc(0, \boldsymbol{pk}; seed_i)$

Step 3

- Compute for $\mathrm{i} = 1, \dots, \mathrm{m}$ $\quad \mathrm{c}_i' \leftarrow c_i + z_i$

Step 4

- Compute $\quad (r_1, \dots, r_n) \leftarrow g(c_1', \dots, c_m')$

20

Theorem

Assume

- Noise components in the ACER Corectness game are indistinguishable from those in the AvgCor game
- If the input mask to TFHE.DimensionSwitch operation (used in public key encryption) is uniformly random then the output mask is also uniformly random
- A specific HintLWE problem instance (particular to TFHE) is hard
- The output of the TFHE evaluation function is “unpredictable”, a technical definition which basically says that if you see the output of an evaluation but not the inputs it is hard to come up with inputs which give that evaluation.
- The ZKPoKs for fresh encryptions are complete, knowledge sound and straight-line extractable.
- The hash function above is a random oracle

Then

The de-randomized randomized evaluation algorithm is ACER correct

Hence with this evaluation algorithm TFHE is sIND-CPA-D and deterministic

Thank you

ZAMA

CONTACT

nigel@zama.org

zama.org

github.com/zama-ai

zama.org/community