Practical q-IND-CPA-D-security for Approximate Homomorphic Encryption

JEAN-PHILIPPE BOSSUAT, ANAMARIA COSTACHE, CHRISTIAN MOUCHET, **LEA NÜRNBERGER**, JUAN RAMÓN TRONCOSO-PASTORIZA

Overview

1. Noise in FHE
2. Approximate Homomorphic Encryption
3. The Li and Micciancio Attack on CKKS
4. IND-CPA-D Security
5. Achieving IND-CPA-D security for CKKS

Preliminaries

What we actually do in the paper

Overview

1. **Noise in FHE**

2. Approximate Homomorphic Encryption

3. The Li and Micciancio Attack on CKKS

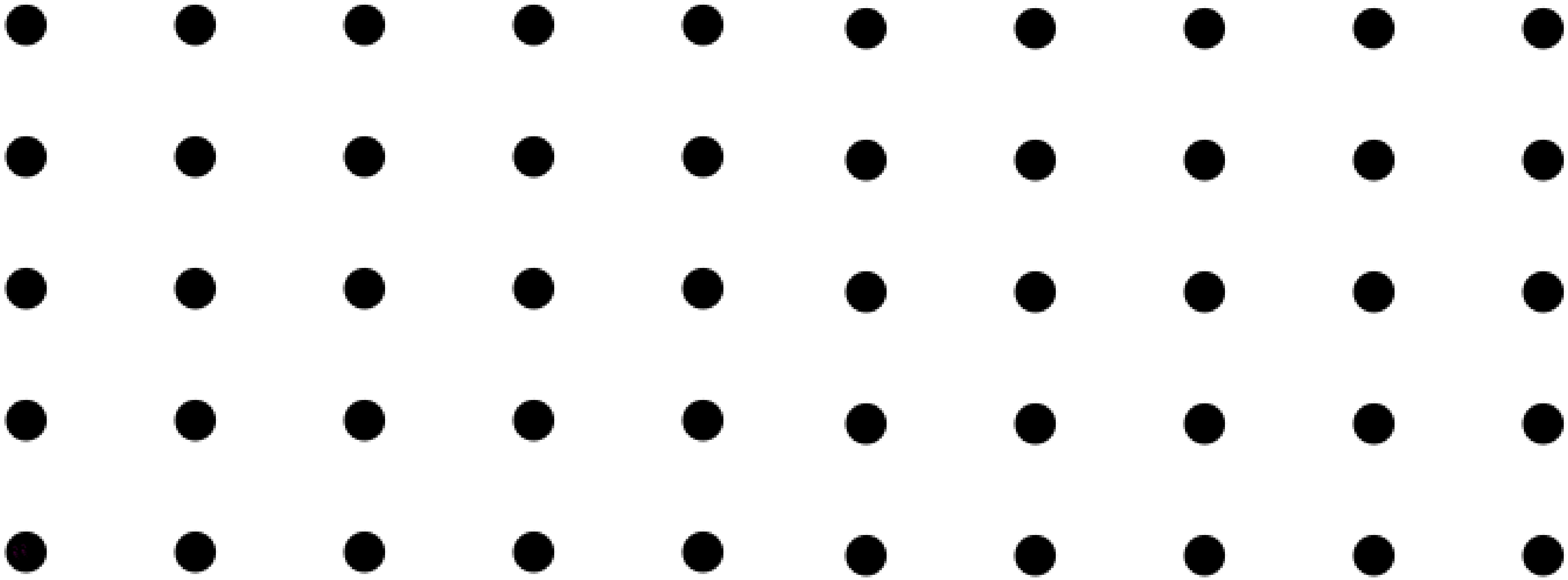
4. IND-CPA-D Security

5. Achieving IND-CPA-D security for CKKS

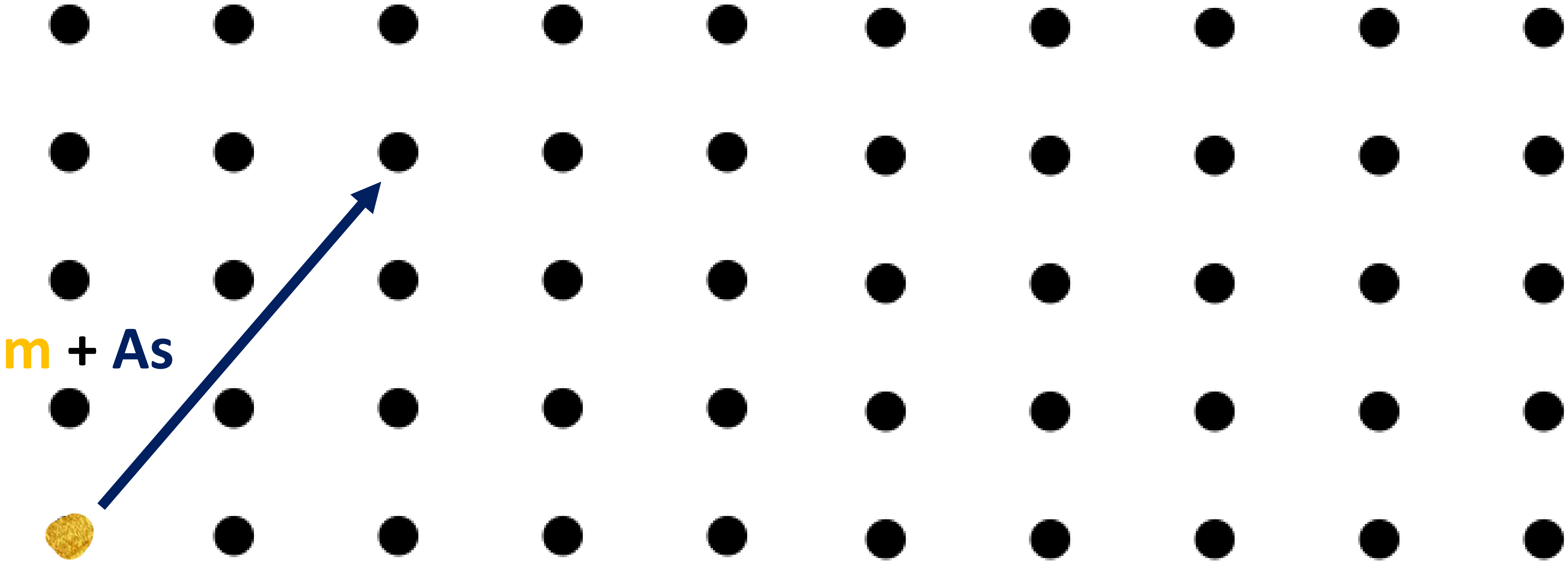
Preliminaries

What we actually do in the paper

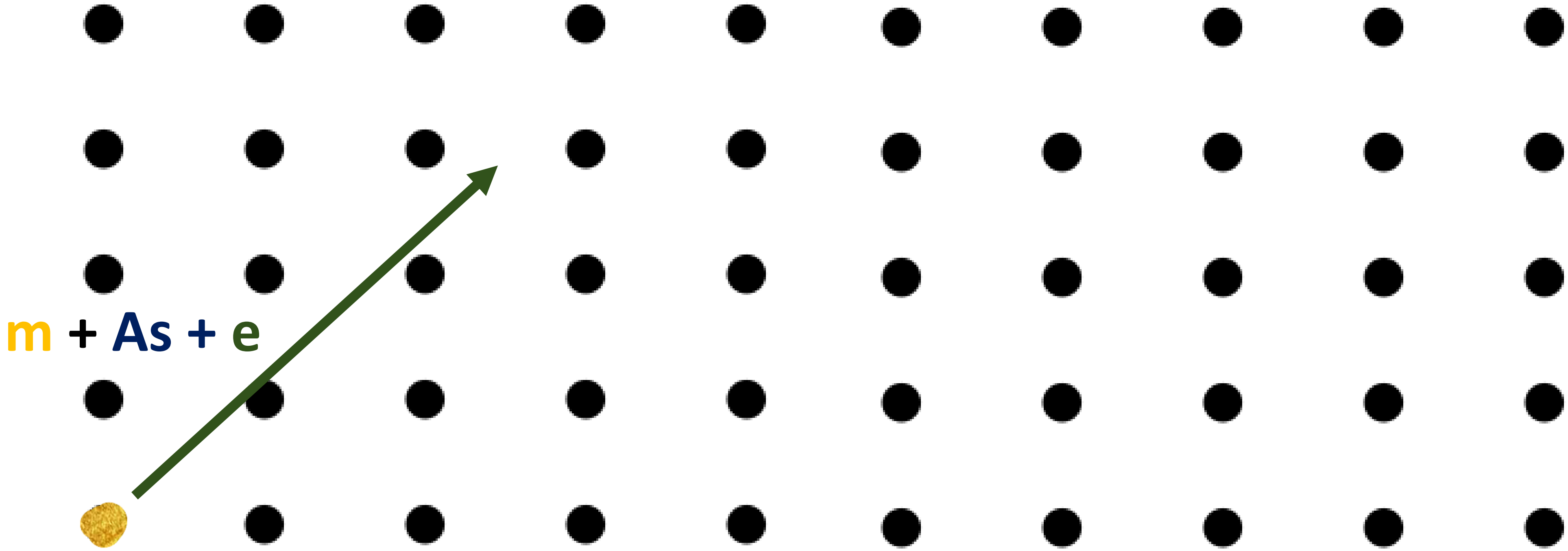
Noise in FHE



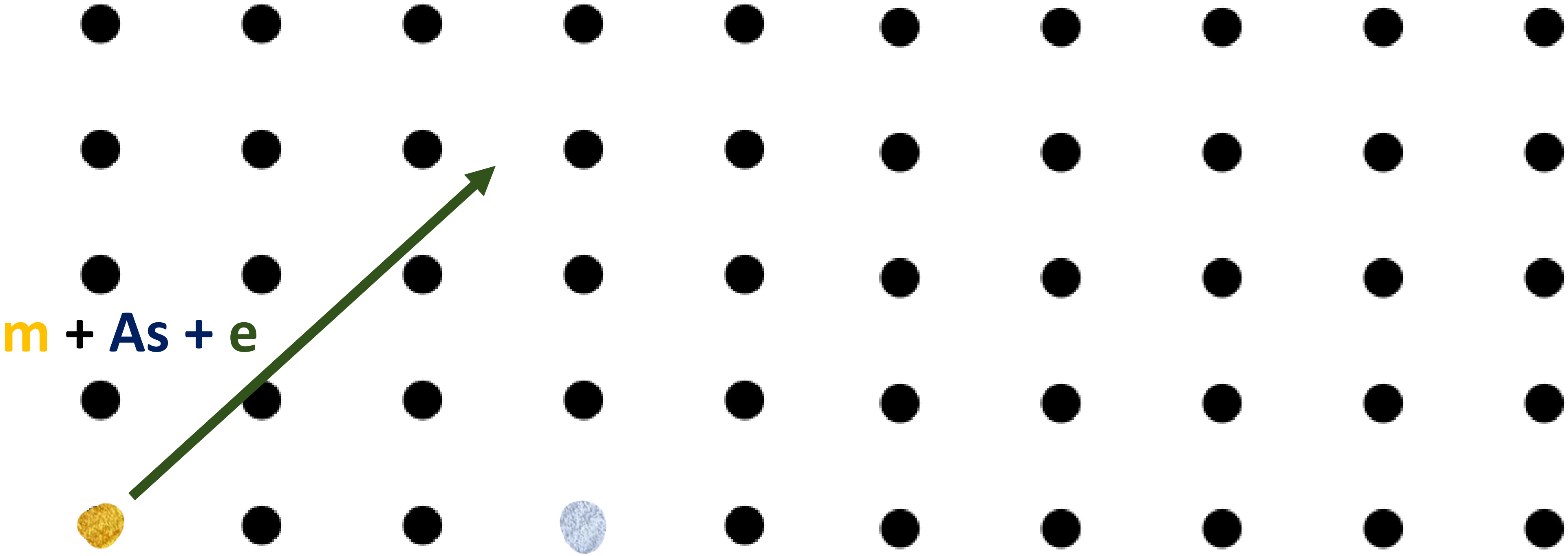
Noise in FHE



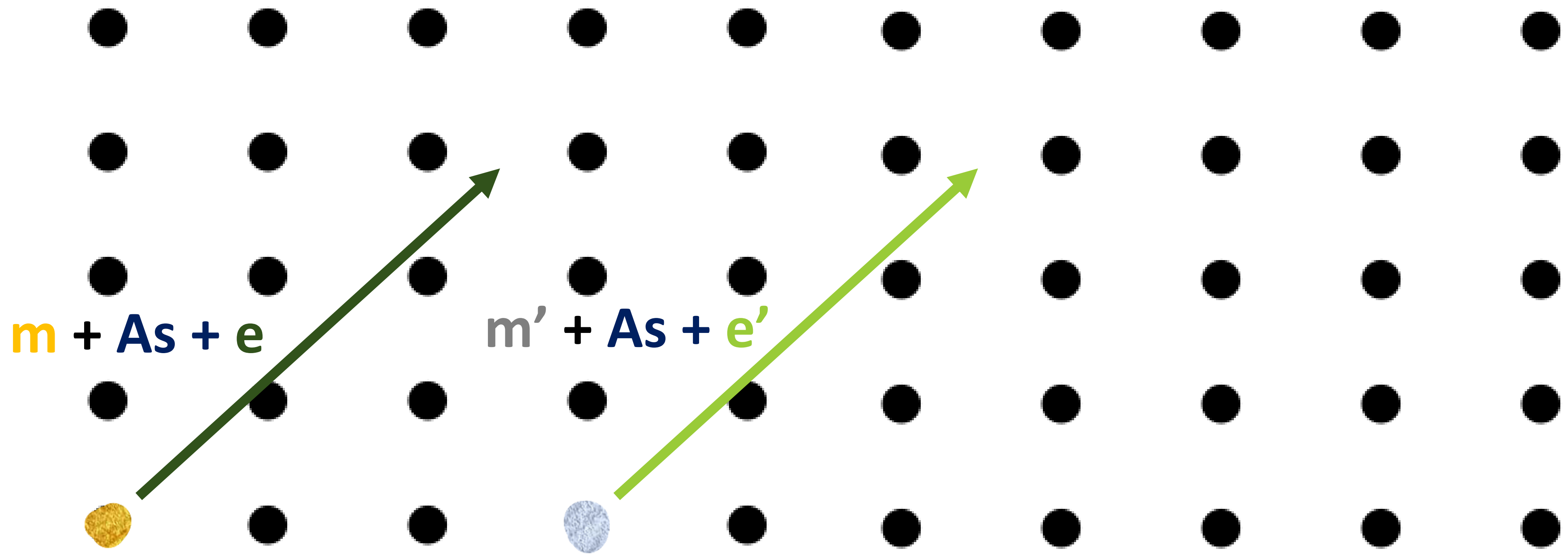
Noise in FHE



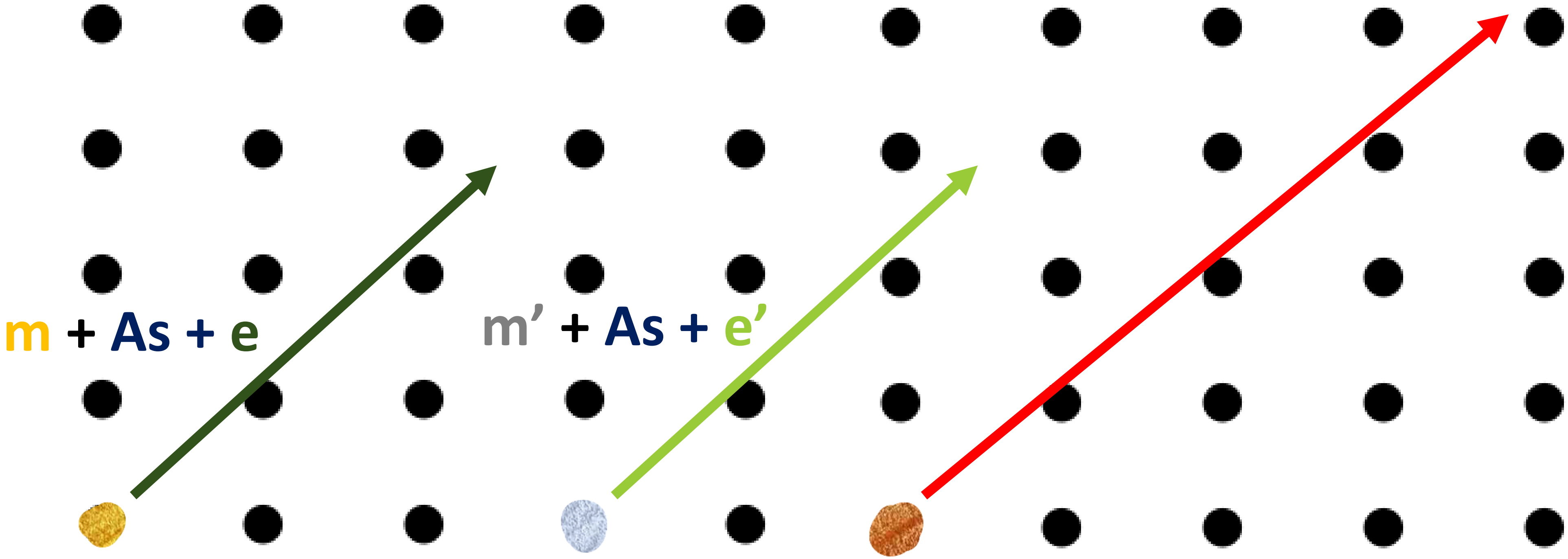
Noise in FHE



Noise in FHE



Noise in FHE



Noise in FHE

FHE Noise Dilemma

- Without noise the scheme would be insecure. But with too much noise eventually we will not be able to decrypt correctly.
- To know whether decryption is still correct, we need to know exactly how much noise the ciphertext has, but if we know it exactly the scheme is no longer secure.

Different Ways of Dealing With Noise

The different schemes have different methods of removing the randomness during decryption to recover the message.

- In BGV the randomness gets multiplied by the plaintext modulus. During decryption a reduction modulo the plaintext modulus is performed that removes the randomness under certain conditions.
- In BFV the message is scaled so that the noise in comparison is small and can under certain conditions be deterministically rounded off during decryption.

Dif ith Noise

$$\mathtt{ct}_{\mathtt{BGV}} = ([\overset{\text{message}}{m} + \underset{\text{Public element}}{a}\overset{\text{Secret key}}{s} + te]_Q, a)$$

$$\mathtt{Dec}_{\mathtt{BGV}}(\mathtt{ct}) = [[\langle \mathtt{ct}, (1, s)\rangle]_Q]_t = [[m + te]_Q]_t$$

The moving the randomness during

- ➢ In BGV the ra ed by the plaintext modulus. During decryption a r ion modulo the plaintext modulus is performed that removes the randomness under certain conditions.
- ➢ In BFV the message is scaled so that the noise in comparison is small and can under certain conditions be deterministically rounded off during decryption.

Different Ways of D

scaling factor

secret key

$$\mathtt{ct}_{\mathtt{BFV}} = ([\Delta m + as + e]_Q, a)$$

message

public element

$$\mathtt{Dec}_{\mathtt{BFV}}(\mathtt{ct}) = \lceil \Delta^{-1}[\langle \mathtt{ct}, (1, s)\rangle]_Q \rfloor = \lceil \Delta^{-1}[\Delta m + e]_Q \rfloor$$

The different schemes have d
during decryption to recover

- In BGV the randomness gets
decryption a reduction modulo t
removes the randomness under certa
- In BFV the message is scaled so that the n
is small and
can under certain conditions be determinis
lly rounded off during
decryption.

Different Ways of Deali

The different schemes have different meth during decryption to recover the message.

- ➢ In BGV the randomness gets multiplied by the pla uring decryption a reduction modulo the plaintext modu is performed that removes the randomness under certain conditions.
- ➢ In BFV the message is scaled so that the noise in comparison is small and can under certain conditions be deterministically rounded off during decryption.

All those conditions require careful bounds on the noise and its growth and take a toll on efficiency.

Overview

1. Noise in FHE

2. Approximate Homomorphic Encryption

3. The Li and Micciancio Attack on CKKS

4. IND-CPA-D Security

5. Achieving IND-CPA-D security for CKKS

Approximate Homomorphic Encryption

At Asiacrypt 2017 Cheon, Kim, Kim and Song proposed a different way of dealing with the noise:

Let it be.

- They proposed HEAAN (Homomorphic Encryption for Arithmetic on Approximate Numbers) today more commonly called CKKS after its authors.
- It is based on the BGV scheme, but the message space is different: the messages are no longer integers modulo t, but in the real or complex numbers.
- Therefore, there is no plaintext modulus by which the randomness could be multiplied, and thus modulus reduction does not remove the randomness.

Approximate Homomorphic Encryption

- The authors show that the randomness does not grow as fast as in other schemes and therefore mostly is very small.
- They hence do not remove it at all, but just consider it a part of the message.
- Encryption has become approximate: the decryption of the encryption of a message will only be approximately the original message, and not exactly.
- The authors argue that most real-world data is approximate in nature, and that therefore exact decryption is not required.

CKKS – Encryption and Decryption

Secret Key Encryption

$$\boxed{\mathtt{ct}_{\mathtt{CKKS}} = ([\mathtt{pt} + as + e]_Q, a)}$$

plaintext

secret key

public element

Decryption

$$\boxed{\begin{aligned}\mathtt{Dec}_{\mathtt{CKKS}}(\mathtt{ct}) &= [\langle \mathtt{ct}, (1, s)\rangle]_Q \\ &= [\mathtt{pt} + as + e - as]_Q \\ &= [\mathtt{pt} + e]_Q\end{aligned}}$$

The error is now part of the plaintext.

Overview

1. Noise in FHE

2. Approximate Homomorphic Encryption

3. The Li and Micciancio Attack on CKKS

4. IND-CPA-D Security

5. Achieving IND-CPA-D security for CKKS

Li and Micciancio Attack on CKKS

- CKKS has been proven IND-CPA secure. IND-CPA is the standard notion of security used in FHE.

- There has been the suspicion for some time that for approximate schemes as CKKS IND-CPA security might not fully cover all the things a passive adversary can do.

- This suspicions has been confirmed by a passive key-recovery attack on CKKS presented by Li and Micciancio at Eurocrypt 2021.

Li and Micciancio Attack on CKKS

Let ct be a secret-key encryption of 0.

$$\mathtt{ct} = ([as + e]_Q, -a)$$

Then the decryption is simply the error term e. Subtracting the decryption from the first component of the ciphertext gives a product between a known polynomial and the secret key.

$$\mathtt{Dec}(\mathtt{ct}) = [\langle \mathtt{ct}, (1, s) \rangle]_Q = [as + e - as]_Q = [e]_Q$$

If the error e is subtracted from the first ciphertext component, as remains. a has an inverse with high probability which can be found efficiently. This returns the secret key.

Li and Micciancio Attack on CKKS

- For “exact” schemes, this attack should not work: decryption simply returns the message, which is already known to the attacker.
- Only in approximate homomorphic encryption does decryption return more information than just the message: the attacker additionally learns the error term if they know both the original message and the approximate decrypted result.
- This is not captured by the IND-CPA notion.
- Therefore, Li and Micciancio propose IND-CPA-D, which enhances the IND-CPA game with a (very restricted) decryption oracle.

Overview

1. Noise in FHE

2. Approximate Homomorphic Encryption

3. The Li and Micciancio Attack on CKKS

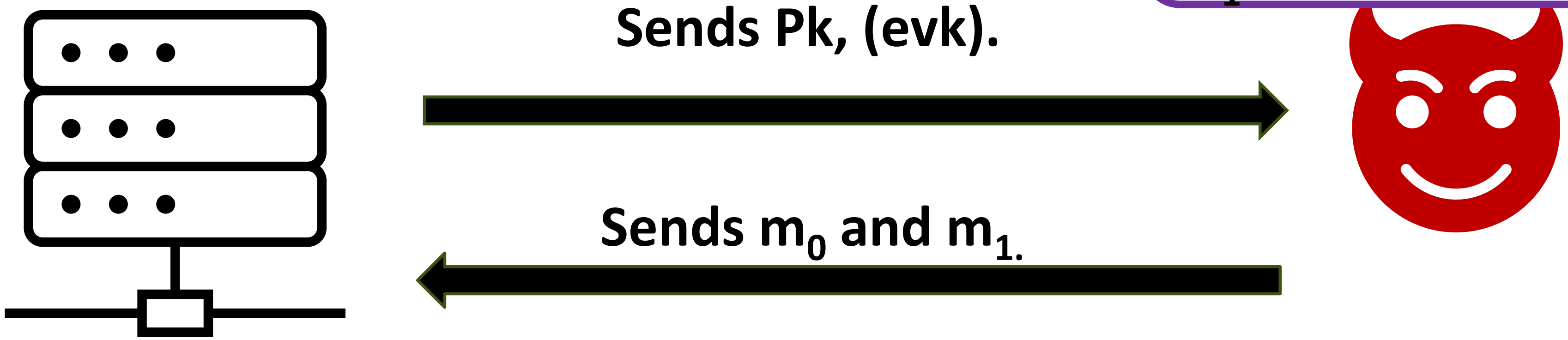
4. IND-CPA-D Security

5. Achieving IND-CPA-D security for CKKS

IND-CPA Security

- Generates the keys sk,pk, (evk).
- Chooses a secret bit b.

- Sees pk, (evk).
- Picks messages m_0 and m_1.



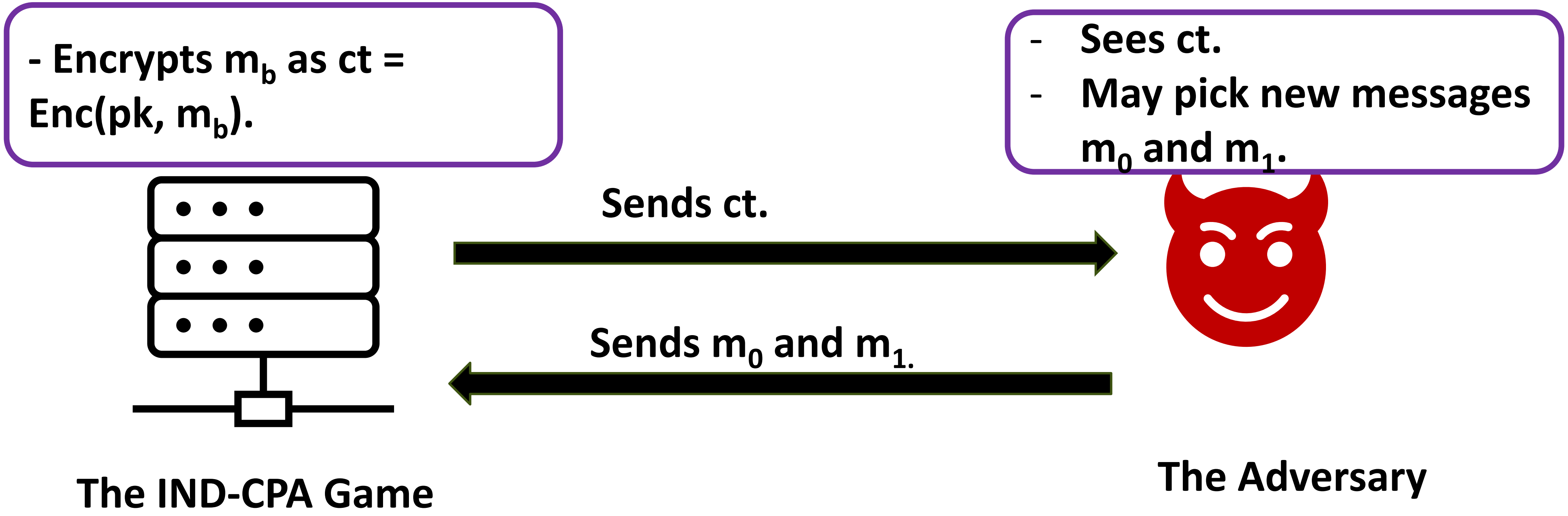
Sends Pk, (evk).

Sends m_0 and $m_{1.}$

The IND-CPA Game

The Adversary

IND-CPA Security



- Encrypts m_b as ct = Enc(pk, m_b).

- Sees ct.
- May pick new messages m_0 and m_1.

Sends ct.

Sends m_0 and $m_{1.}$

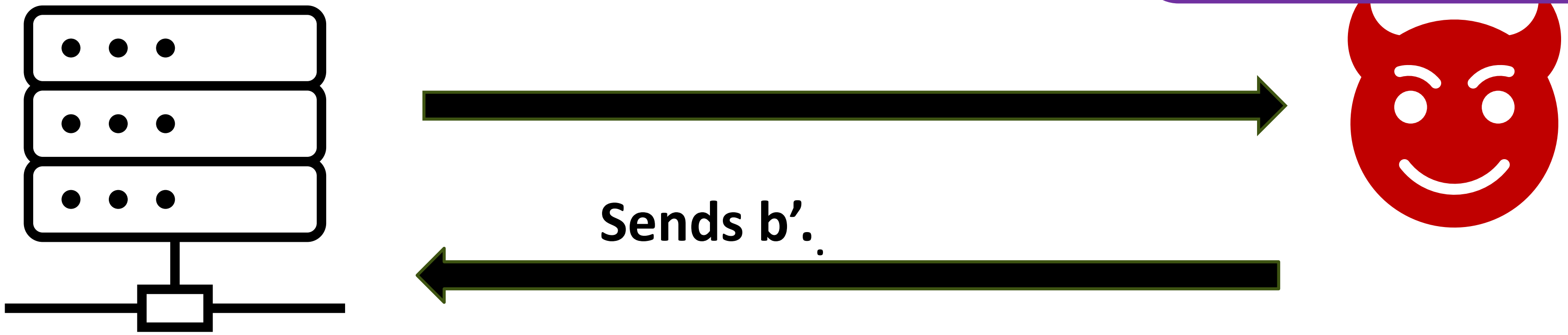
The IND-CPA Game

The Adversary

IND-CPA Security

- **Returns yes if b’ = b.**
- **Returns no else.**

- **Eventually makes a guess b’ for the secret bit b.**



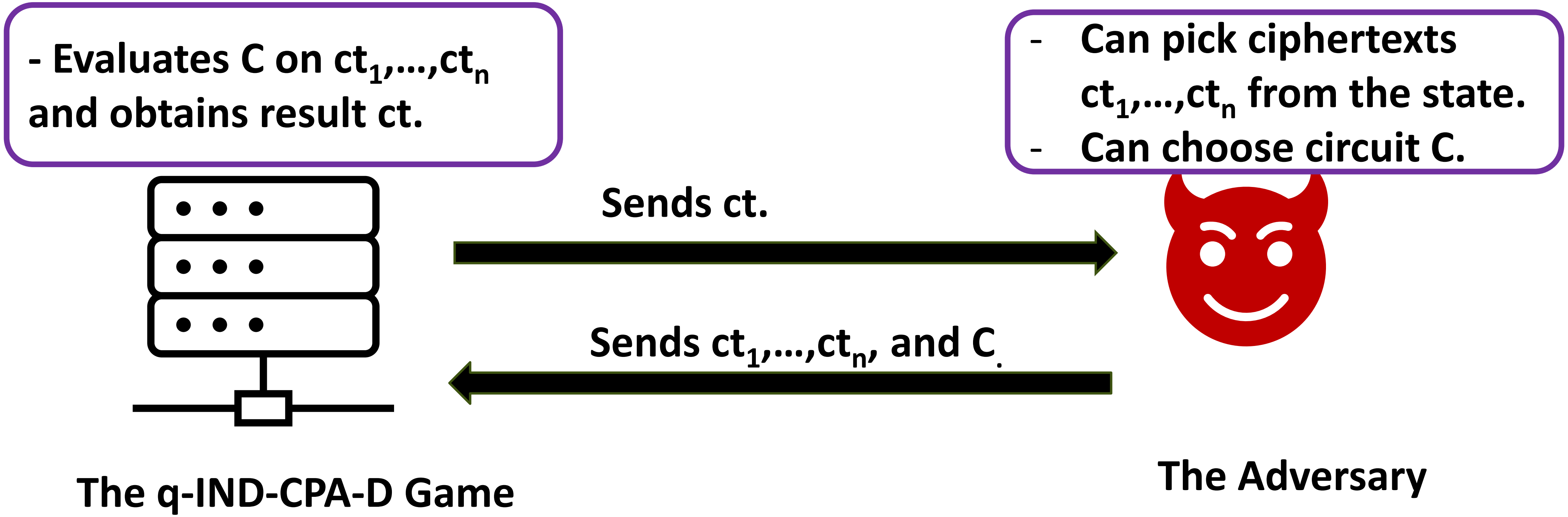
Sends b’.

The IND-CPA Game

The Adversary

The adversary wins the IND-CPA game, if the game returns yes. A scheme is said to be IND-CPA secure if over repeated iterations of this game the adversary’s probability of winning is only negligibly bigger than ½.

(q)-IND-CPA-D Security



- Evaluates C on $ct_1,\ldots,ct_n$ and obtains result ct.

- Can pick ciphertexts $ct_1,\ldots,ct_n$ from the state.
- Can choose circuit C.

Sends ct.

Sends $ct_1,\ldots,ct_n$, and C.

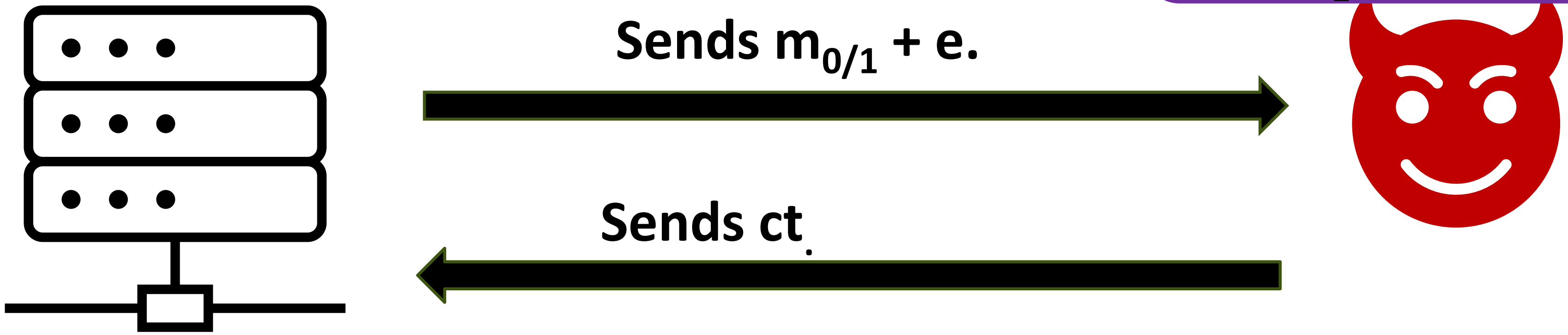
The q-IND-CPA-D Game

The Adversary

(q)-IND-CPA-D Security

- **Decrypts ct**

- **Can pick ciphertext from state such that $m_0 = m_1$.**



Sends $m_{0/1}$ + e.

Sends ct.

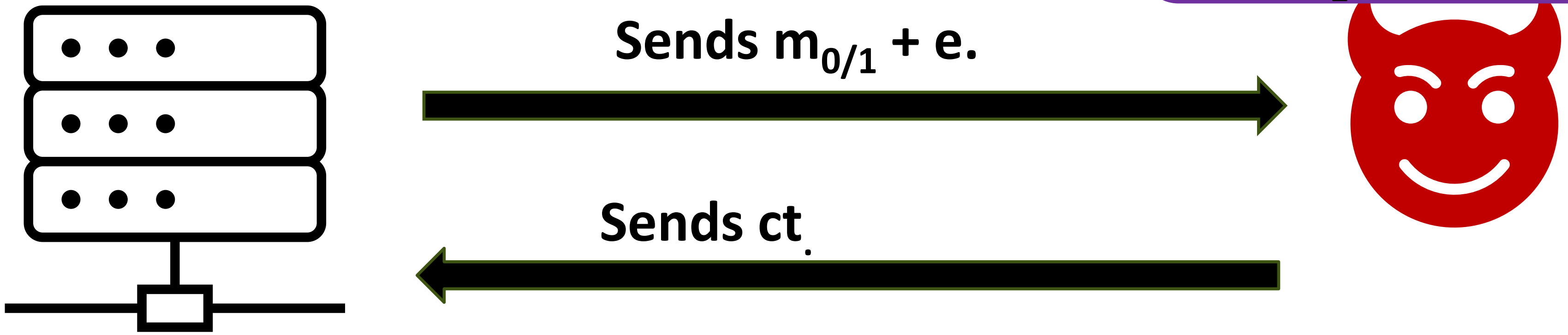
The q-IND-CPA-D Game

The Adversary

(q)-IND-CPA-D Security

- Decrypts ct

- Can pick ciphertext from state such that $m_0 = m_1$.



Sends $m_{0/1}$ + e.

Sends ct.

The q-IND-CPA-D Game

The Adversary

The adversary wins the q-IND-CPA-D game, if the game returns yes. A scheme is said to be q-IND-CPA-D secure if over repeated iterations of this game the adversary’s probability of winning is only negligibly bigger than ½.

Overview

1. Noise in FHE

2. Approximate Homomorphic Encryption

3. The Li and Micciancio Attack on CKKS

4. IND-CPA-D Security

5. Achieving IND-CPA-D security for CKKS

Approaches for Achieving (q)-IND-CPA-D security

- The attack is made possible because if we know what the correct message would be, seeing an approximate encryption gives us the complete error/encryption randomness.

- Knowing the encryption randomness breaks the security of the underlying problem.

- To secure CKKS we therefore need to hide the encryption randomness, even if the attacker knows a correct and an approximate decryption.

- There are two possible techniques:

  1. Adding extra randomness to hide the original one (noise flooding).

  2. Remove the randomness during decryption (exact CKKS).

Approaches for Achieving (q)-IND-CPA-D security

- The attack is made possible because if we know what the correct message would be, seeing an approximate encryption gives us the complete error/encryption randomness.
- Knowing the encryption randomness breaks the security of the underlying problem.
- To secure CKKS we therefore need to hide the encryption randomness, even if the attacker knows a correct and an approximate decryption.
- There are two possible techniques:

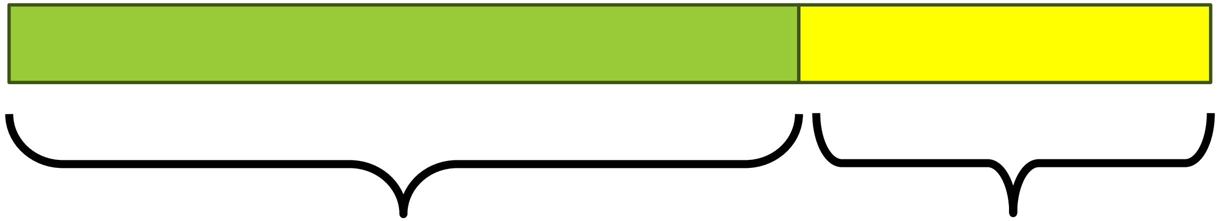
  1. Adding extra randomness to hide the original one (noise flooding).
  2. **Remove the randomness during decryption (exact CKKS).**

Our Contributions

We define an exact version of CKKS.

- ➢We show how to round off the error to return an exact decryption result with high probability.
- ➢We show how to do this without having a large impact on the efficiency of CKKS.
- ➢To do this, we provide a tight analysis of the growth of the randomness of CKKS. We additionally present an estimator that allows to track this growth for any circuit.
- ➢ We use those results to prove that our version of CKKS provably achieves IND-CPA-D security.

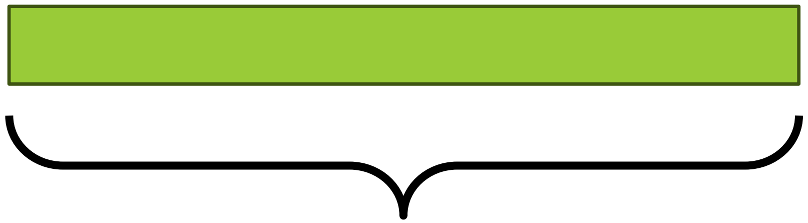
Our Contributions



Message bits untouched by the error

Message bits polluted by the error

Our Contributions



Message bits untouched by the error

We can achieve this by scaling the message down so that all the error bits are “after the comma” and then rounding it up to the nearest integer.

- ➢ We need tight bounds on the error, so that we round off neither too many nor too few bits.

Contributions in More Detail
- Tight Noise Analysis for CKKS -

- The original encryption randomness is changed when ciphertexts are multiplied, added, rotated,…

- Very soon the behaviour of this encryption randomness gets very complex.

- Knowing the distribution and therefore the likely magnitude of the encryption randomness is important outside of this work too, since it influences the setting of the parameters and thereby the efficiency of the scheme.

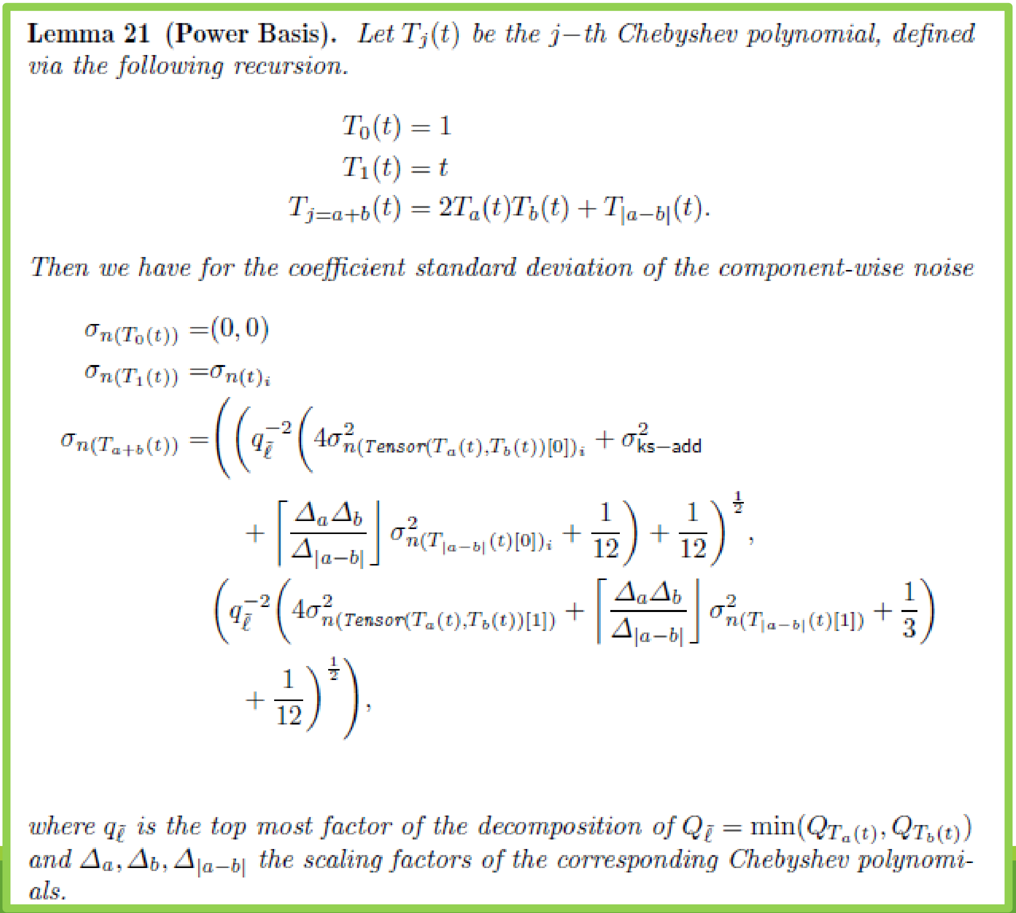
- We provide tight bounds on the noise for all basic operations and show how to combine them into estimates for larger circuits as for example bootstrapping.

How do we bound the error?

- It can be shown that the distribution of the error can be approximated by a normal distribution (assuming independence).
- Then the infinity norm of the error can be bounded using the standard deviation of this distribution.
- We therefore calculate the standard deviation of the error distribution for all basic operations and show how to combine this into estimates on the standard deviation for large circuits.

Contributions in More Detail
- Tight Noise Analysis for CKKS -

This leads to beautiful formulas…



Lemma 21 (Power Basis). *Let $T_j(t)$ be the $j-$th Chebyshev polynomial, defined via the following recursion.*

$$\begin{aligned} T_0(t) &= 1 \\ T_1(t) &= t \\ T_{j=a+b}(t) &= 2T_a(t)T_b(t) + T_{|a-b|}(t). \end{aligned}$$

Then we have for the coefficient standard deviation of the component-wise noise

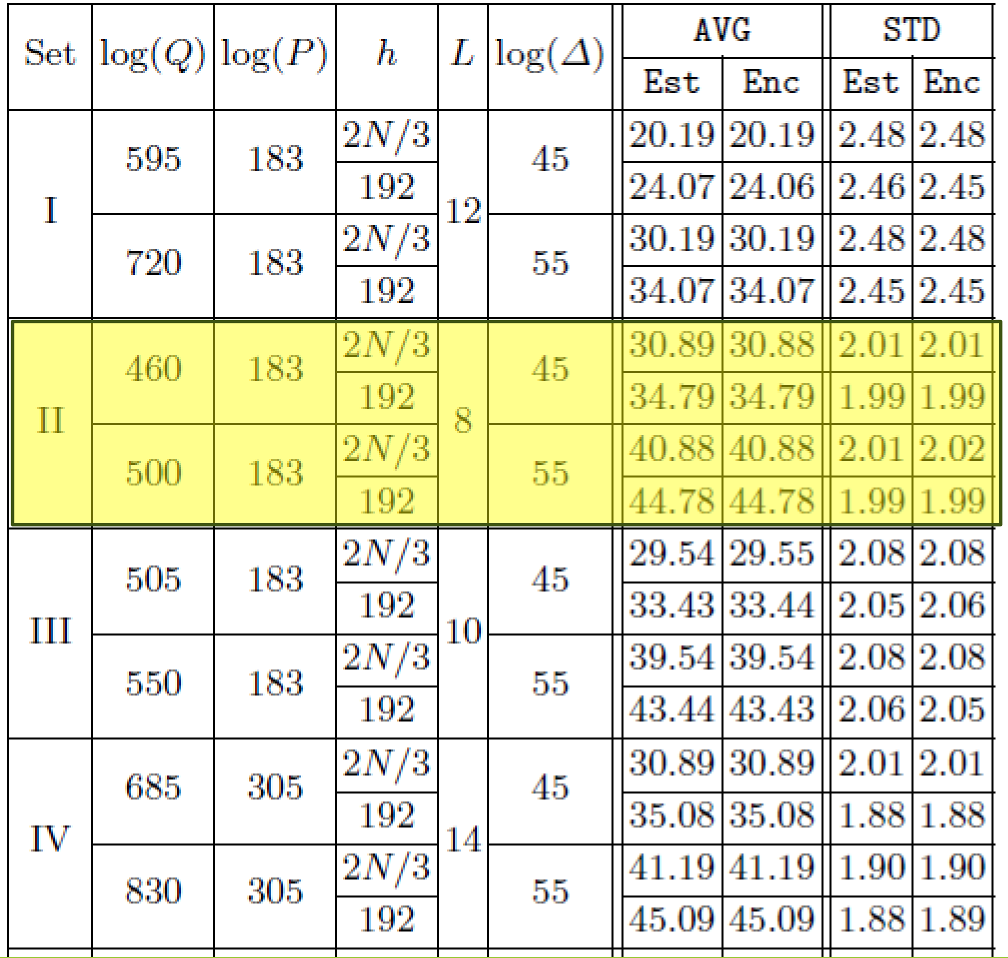
$$\begin{aligned} \sigma_{n(T_0(t))} =& (0,0) \\ \sigma_{n(T_1(t))} =& \sigma_{n(t)_i} \\ \sigma_{n(T_{a+b}(t))} =& \Bigg(\Bigg(q_{\bar{\ell}}^{-2} \Bigg(4\sigma^2_{n(\mathit{Tensor}(T_a(t),T_b(t))[0])_i} + \sigma^2_{\mathsf{ks-add}} \\ & + \left\lceil \frac{\Delta_a \Delta_b}{\Delta_{|a-b|}} \right\rfloor \sigma^2_{n(T_{|a-b|}(t)[0])_i} + \frac{1}{12} \Bigg) + \frac{1}{12} \Bigg)^{\frac{1}{2}}, \\ & \Bigg(q_{\bar{\ell}}^{-2} \Bigg(4\sigma^2_{n(\mathit{Tensor}(T_a(t),T_b(t))[1])} + \left\lceil \frac{\Delta_a \Delta_b}{\Delta_{|a-b|}} \right\rfloor \sigma^2_{n(T_{|a-b|}(t)[1])} + \frac{1}{3} \Bigg) \\ & + \frac{1}{12} \Bigg)^{\frac{1}{2}} \Bigg), \end{aligned}$$

where $q_{\bar{\ell}}$ is the top most factor of the decomposition of $Q_{\bar{\ell}} = \min(Q_{T_a(t)}, Q_{T_b(t)})$ and $\Delta_a, \Delta_b, \Delta_{|a-b|}$ the scaling factors of the corresponding Chebyshev polynomials.

Contributions in More Detail
- Tight Noise Analysis for CKKS -

…and tight estimates.

These are the estimates for approximating the sigmoid function by a Chebysheff interpolation and evaluating this on ciphertexts.

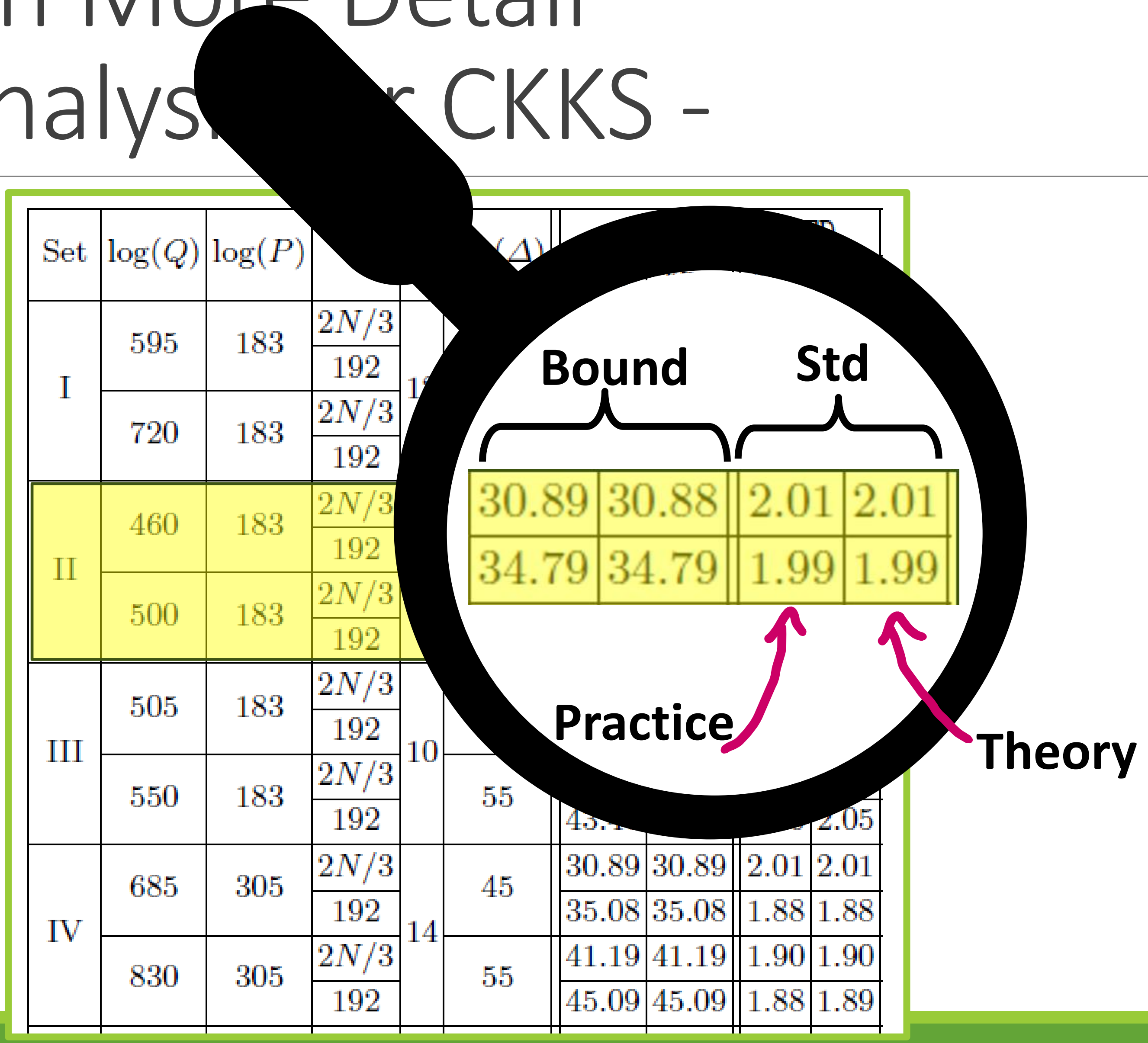


Set	$\log(Q)$	$\log(P)$	h	L	$\log(\Delta)$	AVG		STD	
						Est	Enc	Est	Enc
I	595	183	$2N/3$	12	45	20.19	20.19	2.48	2.48
			192			24.07	24.06	2.46	2.45
	720	183	$2N/3$		55	30.19	30.19	2.48	2.48
			192			34.07	34.07	2.45	2.45
II	460	183	$2N/3$	8	45	30.89	30.88	2.01	2.01
			192			34.79	34.79	1.99	1.99
	500	183	$2N/3$		55	40.88	40.88	2.01	2.02
			192			44.78	44.78	1.99	1.99
III	505	183	$2N/3$	10	45	29.54	29.55	2.08	2.08
			192			33.43	33.44	2.05	2.06
	550	183	$2N/3$		55	39.54	39.54	2.08	2.08
			192			43.44	43.43	2.06	2.05
IV	685	305	$2N/3$	14	45	30.89	30.89	2.01	2.01
			192			35.08	35.08	1.88	1.88
	830	305	$2N/3$		55	41.19	41.19	1.90	1.90
			192			45.09	45.09	1.88	1.89

Contributions in More Detail
- Tight Noise Analysis for CKKS -

…and tight estimates.

These are the estimates for approximating the sigmoid function by a Chebysheff interpolation and evaluating this on ciphertexts.



Set	$\log(Q)$	$\log(P)$	[illegible]	[illegible]	[illegible](Δ)	[illegible]	[illegible]	[illegible]	[illegible]
I	595	183	$2N/3$	1[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
			192			[illegible]	[illegible]	[illegible]	[illegible]
	720	183	$2N/3$		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
			192			[illegible]	[illegible]	[illegible]	[illegible]
II	460	183	$2N/3$	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
			192			[illegible]	[illegible]	[illegible]	[illegible]
	500	183	$2N/3$		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
			192			[illegible]	[illegible]	[illegible]	[illegible]
III	505	183	$2N/3$	10	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
			192			[illegible]	[illegible]	[illegible]	[illegible]
	550	183	$2N/3$		55	[illegible]	[illegible]	[illegible]	[illegible]
			192			43.[illegible]	[illegible]	[illegible]	2.05
IV	685	305	$2N/3$	14	45	30.89	30.89	2.01	2.01
			192			35.08	35.08	1.88	1.88
	830	305	$2N/3$		55	41.19	41.19	1.90	1.90
			192			45.09	45.09	1.88	1.89

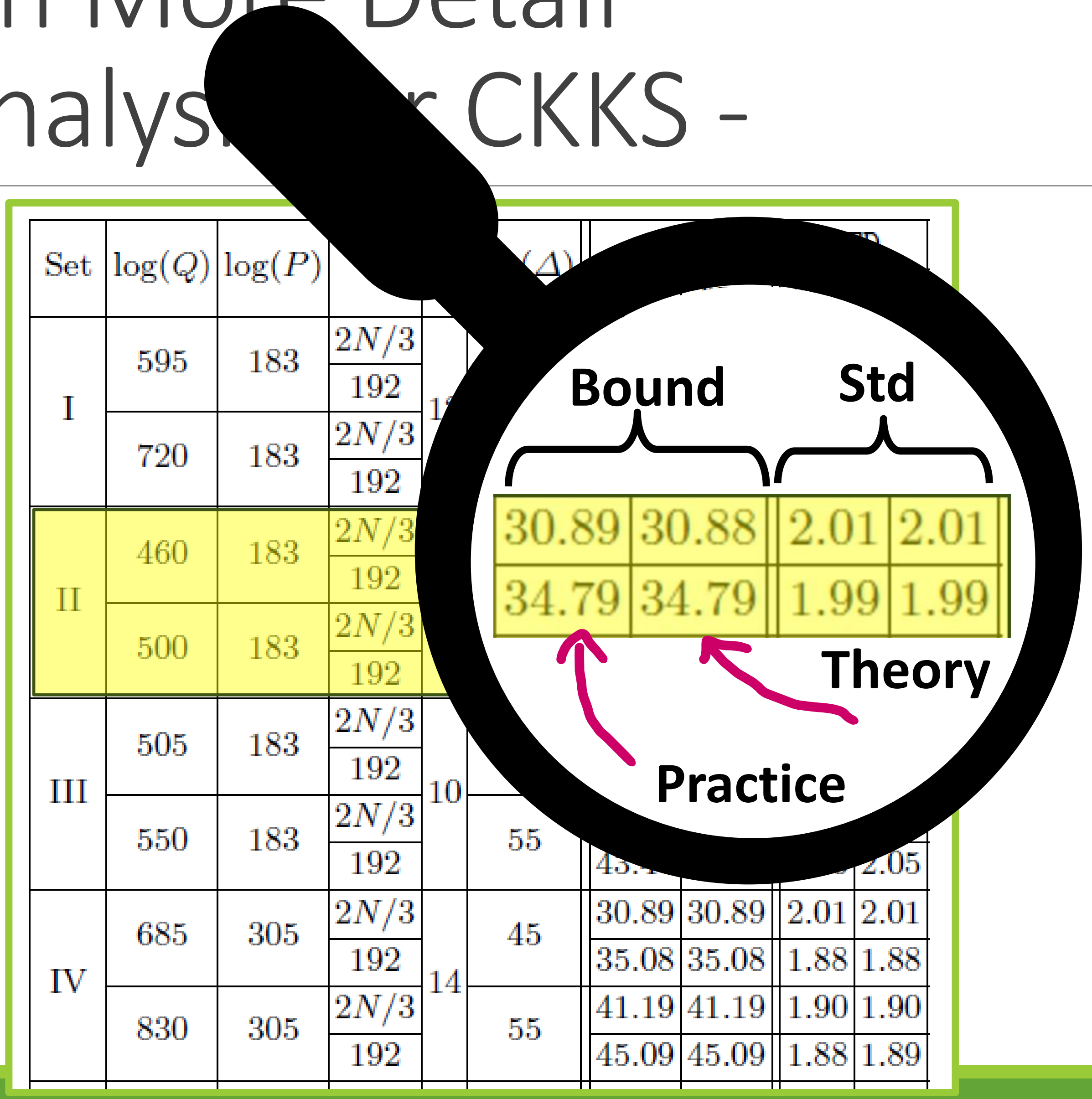
Bound		Std	
30.89	30.88	2.01	2.01
34.79	34.79	1.99	1.99

Practice
Theory

Contributions in More Detail
- Tight Noise Analys r CKKS -

…and tight estimates.

These are the estimates for approximating the sigmoid function by a Chebysheff interpolation and evaluating this on ciphertexts.



Set	$\log(Q)$	$\log(P)$	[illegible]	[illegible]	(Δ)	[illegible]	[illegible]	[illegible]	[illegible]
I	595	183	$2N/3$	1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
			192			[illegible]	[illegible]	[illegible]	[illegible]
	720	183	$2N/3$		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
			192			[illegible]	[illegible]	[illegible]	[illegible]
II	460	183	$2N/3$	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
			192			[illegible]	[illegible]	[illegible]	[illegible]
	500	183	$2N/3$		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
			192			[illegible]	[illegible]	[illegible]	[illegible]
III	505	183	$2N/3$	10	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
			192			[illegible]	[illegible]	[illegible]	[illegible]
	550	183	$2N/3$		55	[illegible]	[illegible]	[illegible]	[illegible]
			192			43.1	[illegible]	[illegible]	2.05
IV	685	305	$2N/3$	14	45	30.89	30.89	2.01	2.01
			192			35.08	35.08	1.88	1.88
	830	305	$2N/3$		55	41.19	41.19	1.90	1.90
			192			45.09	45.09	1.88	1.89

30.89	30.88	2.01	2.01
34.79	34.79	1.99	1.99

Bound
Std
Theory
Practice

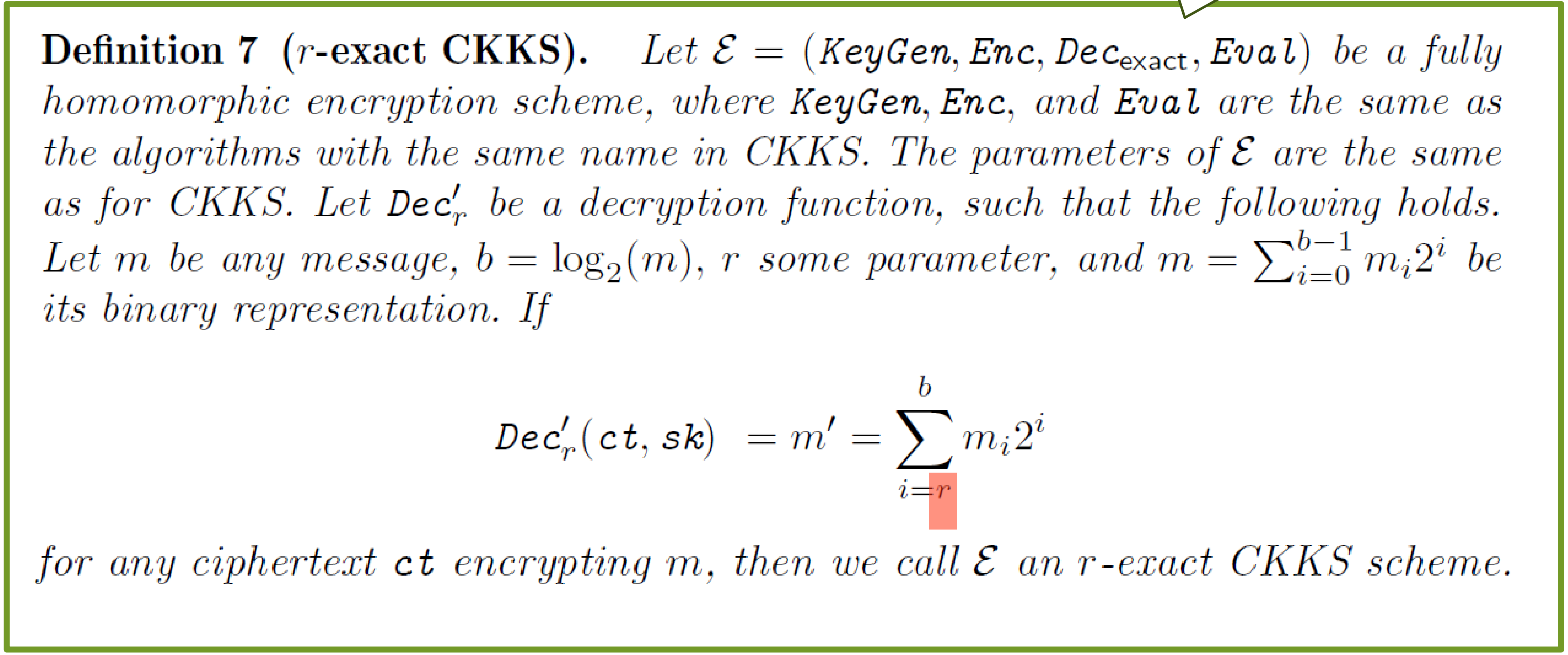
Contributions in More Detail
- The Estimator -

We use these theoretical estimations to provide an estimator for the Lattigo library.

- For any function the estimates of the basic operations are combined into estimates of the larger function, with no loss in tightness.
- This allows to get a good idea of the development of the error for any function and therefore an efficient setting of the parameters without having to run the computations a couple of times in advance or having to combine complex theoretical formulas by hand.

Contributions in More Detail - (δ,r)-exact CKKS -

This is what we would like in an ideal world.

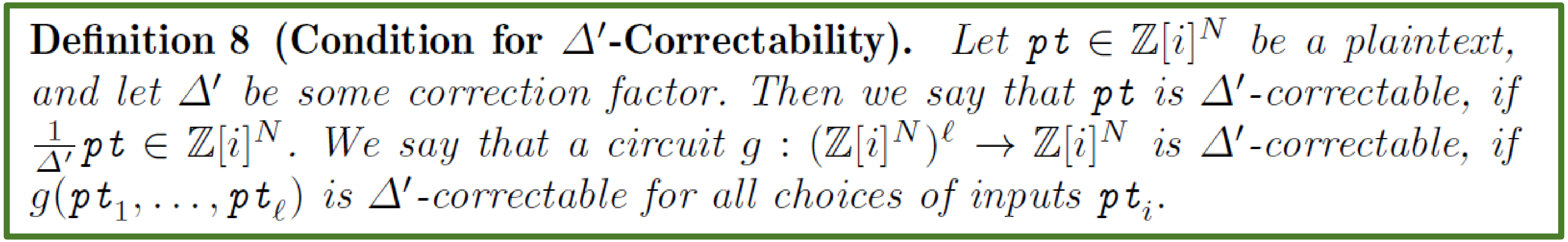


Definition 7 (*r*-**exact CKKS**). *Let* $\mathcal{E} = (\texttt{KeyGen}, \texttt{Enc}, \texttt{Dec}_{\text{exact}}, \texttt{Eval})$ *be a fully homomorphic encryption scheme, where* $\texttt{KeyGen}, \texttt{Enc}$, *and* $\texttt{Eval}$ *are the same as the algorithms with the same name in CKKS. The parameters of* $\mathcal{E}$ *are the same as for CKKS. Let* $\texttt{Dec}'_r$ *be a decryption function, such that the following holds. Let* m *be any message,* $b = \log_2(m)$, r *some parameter, and* $m = \sum_{i=0}^{b-1} m_i 2^i$ *be its binary representation. If*

$$\texttt{Dec}'_r(\texttt{ct}, \texttt{sk}) = m' = \sum_{i=r}^{b} m_i 2^i$$

for any ciphertext $\texttt{ct}$ *encrypting* m, *then we call* $\mathcal{E}$ *an* r*-exact CKKS scheme.*

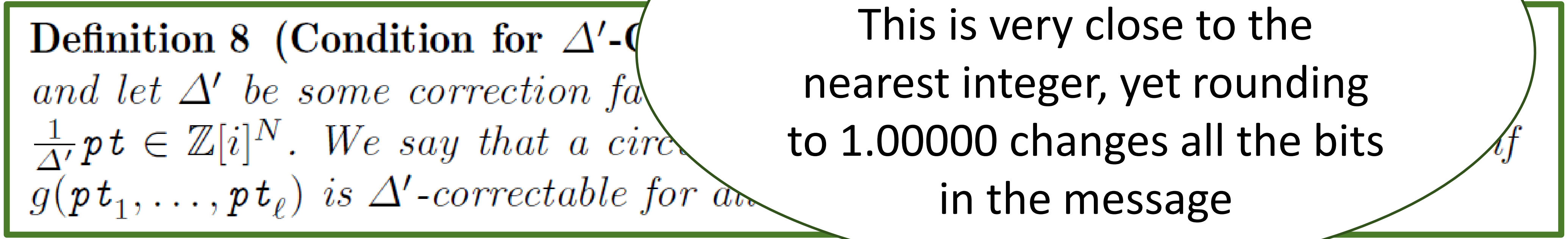
Contributions in More Detail - (δ,r)-exact CKKS -



Definition 8 (Condition for Δ'-Correctability). *Let $\boldsymbol{pt} \in \mathbb{Z}[i]^N$ be a plaintext, and let Δ' be some correction factor. Then we say that $\boldsymbol{pt}$ is Δ'-correctable, if $\frac{1}{\Delta'}\boldsymbol{pt} \in \mathbb{Z}[i]^N$. We say that a circuit $g : (\mathbb{Z}[i]^N)^\ell \to \mathbb{Z}[i]^N$ is Δ'-correctable, if $g(\boldsymbol{pt}_1, \ldots, \boldsymbol{pt}_\ell)$ is Δ'-correctable for all choices of inputs $\boldsymbol{pt}_i$.*

- This condition is necessary, since even a small error may change all the bits in a ciphertext when it is rounded off, if it is close to the cutting off point.
- This is not predictable and is a problem in other schemes too.

Contributions in More Detail
- (δ,r)-exact CKKS -

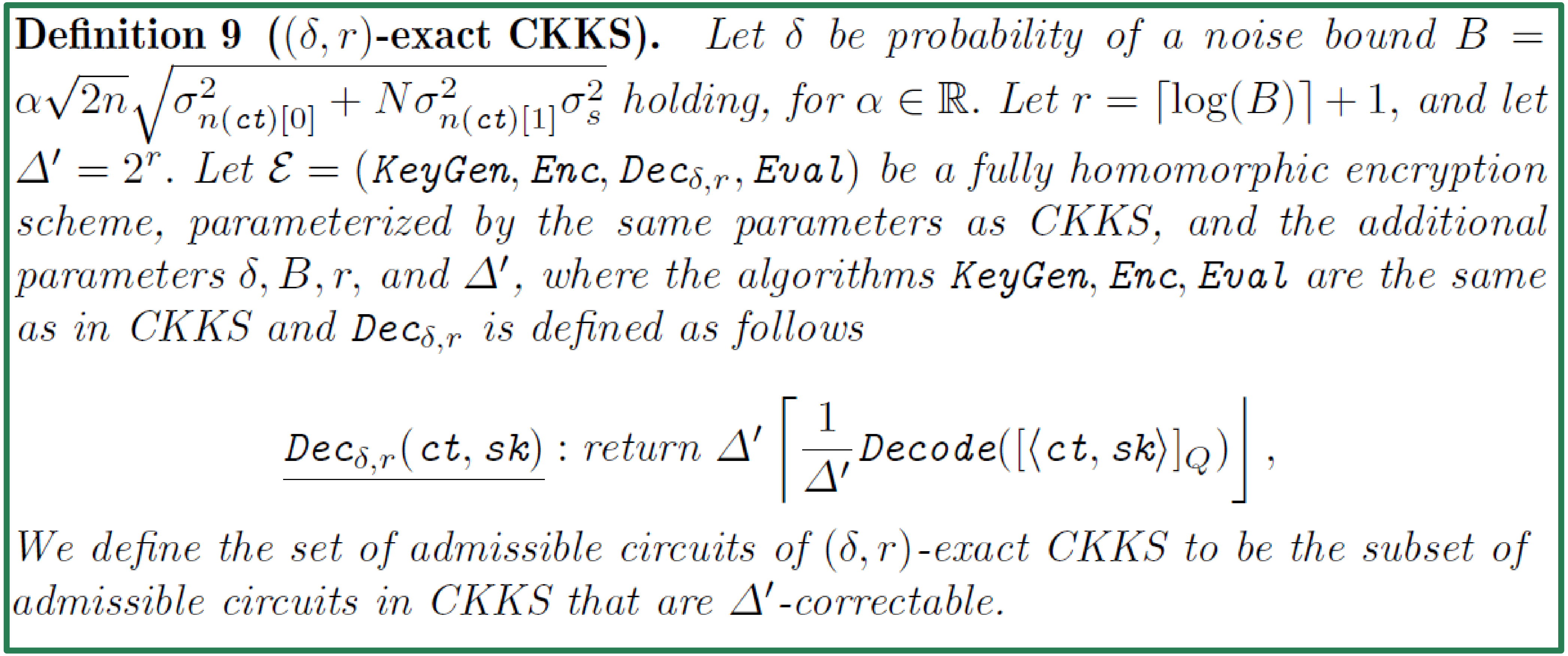


Definition 8 (Condition for Δ'-C *and let* Δ' *be some correction fa* $\frac{1}{\Delta'}\boldsymbol{pt} \in \mathbb{Z}[i]^N$. *We say that a circ* *if* $g(\boldsymbol{pt}_1, \ldots, \boldsymbol{pt}_\ell)$ *is* Δ'*-correctable for al*

To give an example, assume we have the value 0.99999. This is very close to the nearest integer, yet rounding to 1.00000 changes all the bits in the message

- ➢ This condition is necessary, since even a small error may change all the bits in a ciphertext when it is rounded off, if it is close to the cutting off point.
- ➢ This is not predictable and is a problem in other schemes too.

Contributions in More Detail
- (δ,r)-exact CKKS -

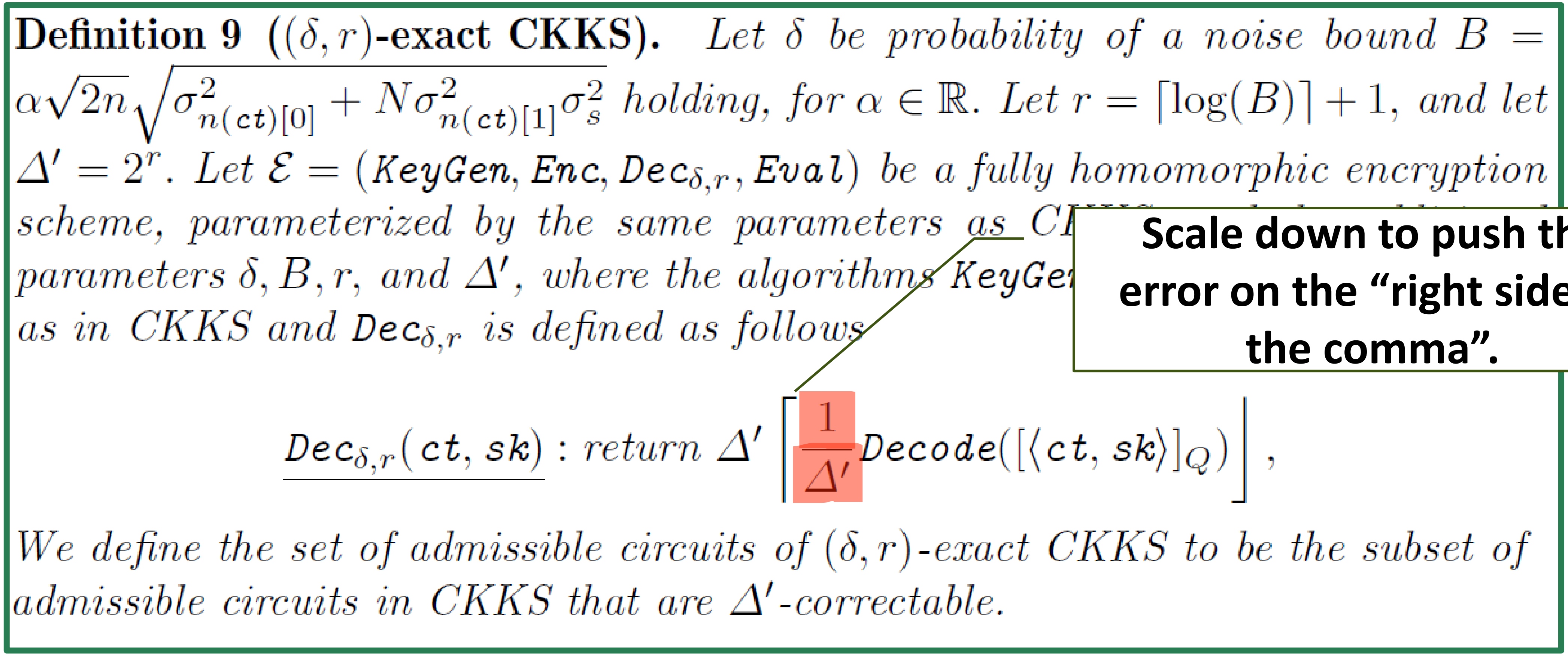


Definition 9 **((δ, r)-exact CKKS).** *Let δ be probability of a noise bound $B = \alpha\sqrt{2n}\sqrt{\sigma^2_{n(\mathtt{ct})[0]} + N\sigma^2_{n(\mathtt{ct})[1]}\sigma^2_s}$ holding, for $\alpha \in \mathbb{R}$. Let $r = \lceil \log(B) \rceil + 1$, and let $\Delta' = 2^r$. Let $\mathcal{E} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec}_{\delta,r}, \mathtt{Eval})$ be a fully homomorphic encryption scheme, parameterized by the same parameters as CKKS, and the additional parameters $\delta, B, r,$ and Δ', where the algorithms $\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Eval}$ are the same as in CKKS and $\mathtt{Dec}_{\delta,r}$ is defined as follows*

$$\underline{\mathtt{Dec}_{\delta,r}(\mathtt{ct}, \mathtt{sk})} : \textit{return } \Delta' \left\lceil \frac{1}{\Delta'} \mathtt{Decode}([\langle \mathtt{ct}, \mathtt{sk} \rangle]_Q) \right\rfloor ,$$

We define the set of admissible circuits of (δ, r)-exact CKKS to be the subset of admissible circuits in CKKS that are Δ'-correctable.

Contributions in More Detail
- (δ,r)-exact CKKS -



Definition 9 ((δ, r)-exact CKKS). *Let δ be probability of a noise bound* $B = \alpha\sqrt{2n}\sqrt{\sigma^2_{n(\mathtt{ct})[0]} + N\sigma^2_{n(\mathtt{ct})[1]}\sigma^2_s}$ *holding, for* $\alpha \in \mathbb{R}$. *Let* $r = \lceil \log(B) \rceil + 1$, *and let* $\Delta' = 2^r$. *Let* $\mathcal{E} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec}_{\delta,r}, \mathtt{Eval})$ *be a fully homomorphic encryption scheme, parameterized by the same parameters as C*

parameters δ, B, r, *and* Δ', *where the algorithms* KeyGe

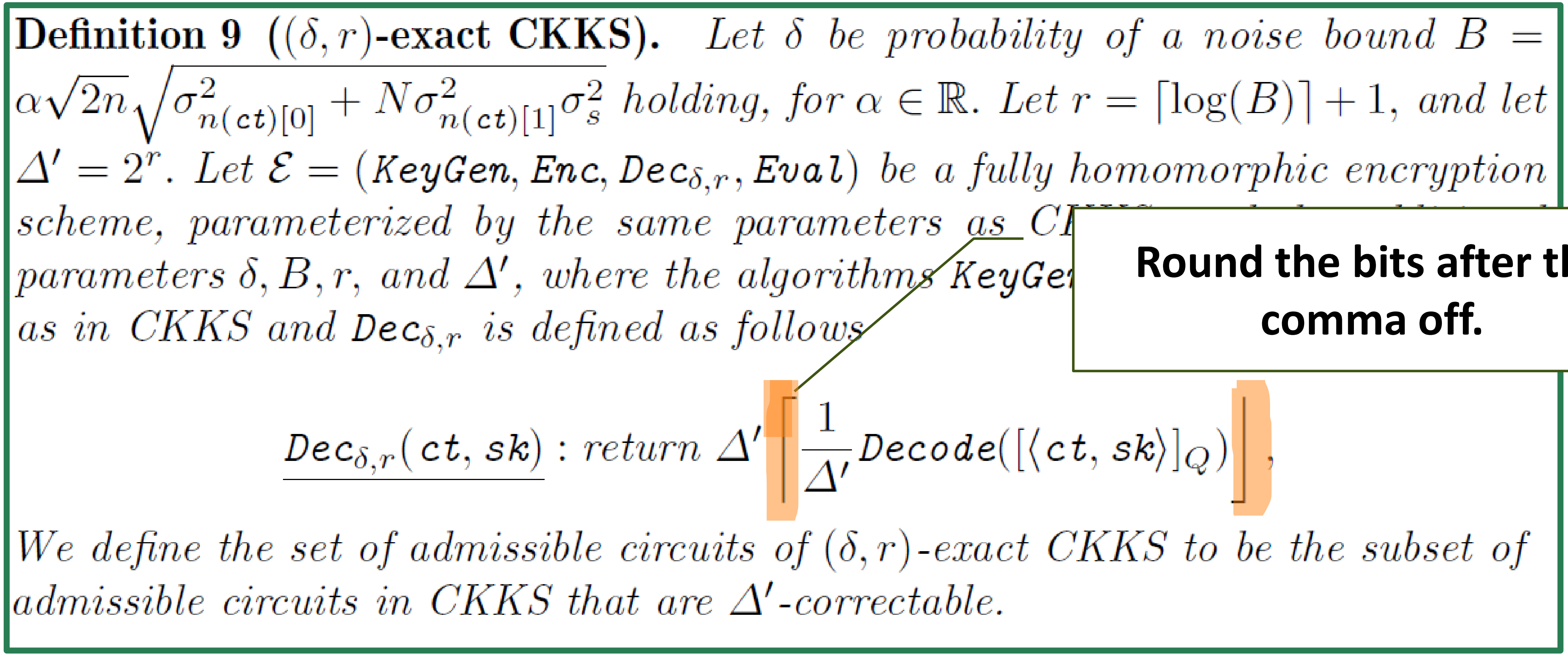
as in CKKS and $\mathtt{Dec}_{\delta,r}$ *is defined as follows*

$$\mathtt{Dec}_{\delta,r}(\mathtt{ct},\mathtt{sk}) : \textit{return } \Delta' \left\lceil \frac{1}{\Delta'} \mathtt{Decode}([\langle \mathtt{ct}, \mathtt{sk} \rangle]_Q) \right\rfloor ,$$

We define the set of admissible circuits of (δ, r)*-exact CKKS to be the subset of admissible circuits in CKKS that are* Δ′*-correctable.*

Scale down to push the error on the "right side of the comma".

Contributions in More Detail - (δ,r)-exact CKKS -



Definition 9 ((δ, r)-exact CKKS). *Let δ be probability of a noise bound* $B = \alpha\sqrt{2n}\sqrt{\sigma^2_{n(\mathtt{ct})[0]} + N\sigma^2_{n(\mathtt{ct})[1]}\sigma^2_s}$ *holding, for* $\alpha \in \mathbb{R}$*. Let* $r = \lceil \log(B) \rceil + 1$*, and let* $\Delta' = 2^r$*. Let* $\mathcal{E} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec}_{\delta,r}, \mathtt{Eval})$ *be a fully homomorphic encryption scheme, parameterized by the same parameters as C*

parameters $\delta, B, r,$ *and* Δ'*, where the algorithms* $\mathtt{KeyGe}$

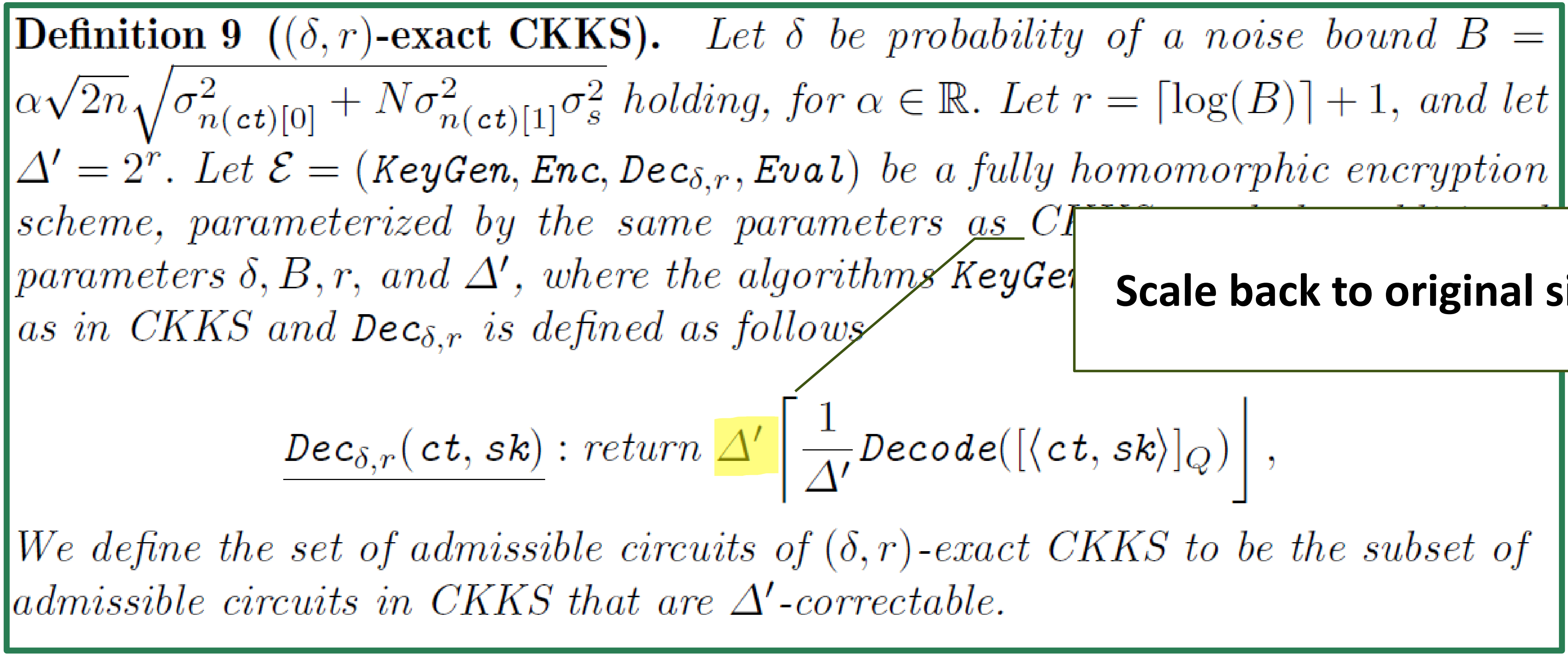
as in CKKS and $\mathtt{Dec}_{\delta,r}$ *is defined as follows*

$$\underline{\mathtt{Dec}_{\delta,r}(\mathtt{ct}, \mathtt{sk})} : \textit{return } \Delta' \left\lceil \frac{1}{\Delta'} \mathtt{Decode}([\langle \mathtt{ct}, \mathtt{sk} \rangle]_Q) \right\rfloor ,$$

We define the set of admissible circuits of (δ, r)*-exact CKKS to be the subset of admissible circuits in CKKS that are* Δ'*-correctable.*

Round the bits after the comma off.

Contributions in More Detail
- (δ,r)-exact CKKS -



Definition 9 **((δ, r)-exact CKKS).** *Let* δ *be probability of a noise bound* $B = \alpha\sqrt{2n}\sqrt{\sigma^2_{n(\mathtt{ct})[0]} + N\sigma^2_{n(\mathtt{ct})[1]}\sigma_s^2}$ *holding, for* $\alpha \in \mathbb{R}$. *Let* $r = \lceil \log(B) \rceil + 1$, *and let* $\Delta' = 2^r$. *Let* $\mathcal{E} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec}_{\delta,r}, \mathtt{Eval})$ *be a fully homomorphic encryption scheme, parameterized by the same parameters as CI*

parameters $\delta, B, r,$ *and* Δ', *where the algorithms* $\mathtt{KeyGe}$

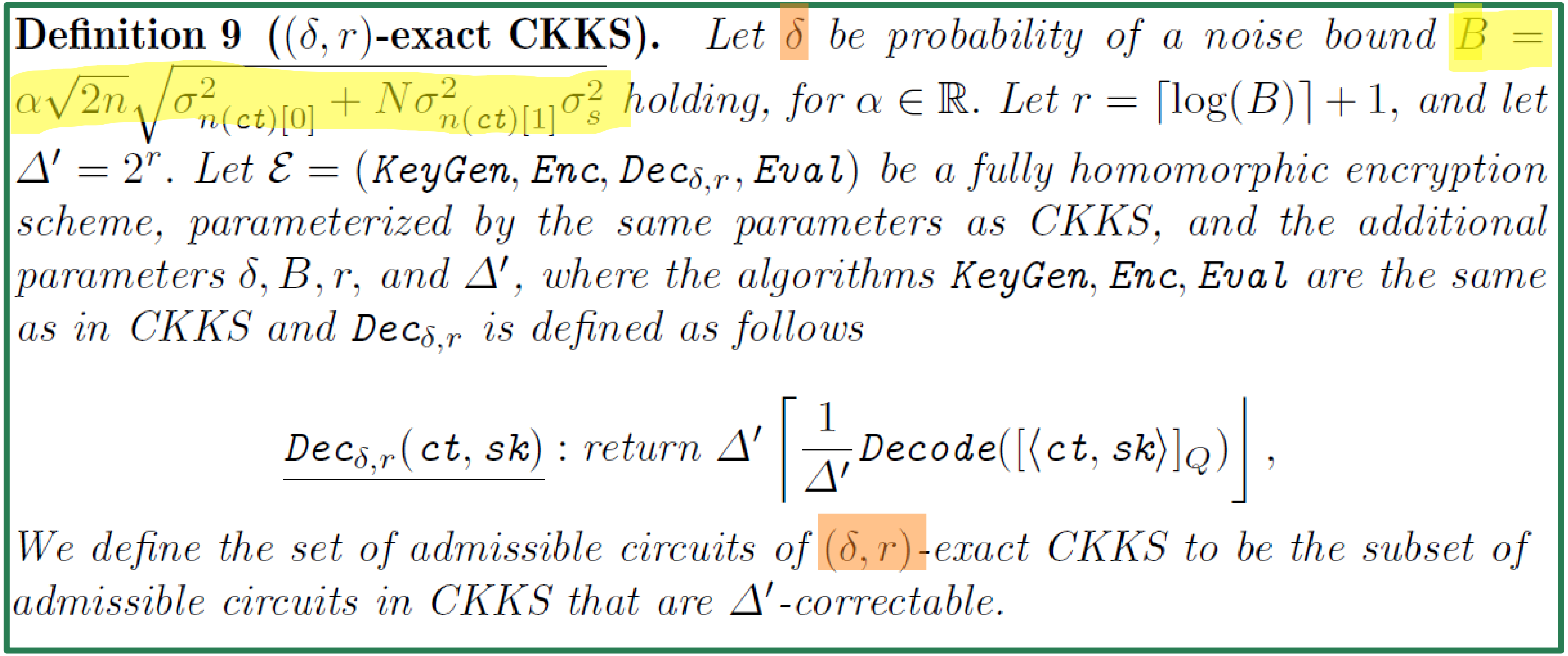
as in CKKS and $\mathtt{Dec}_{\delta,r}$ *is defined as follows*

$$\underline{\mathtt{Dec}_{\delta,r}(\mathtt{ct}, \mathtt{sk})} : \textit{return } \Delta' \left\lceil \frac{1}{\Delta'} \mathtt{Decode}([\langle \mathtt{ct}, \mathtt{sk} \rangle]_Q) \right\rfloor,$$

We define the set of admissible circuits of (δ, r)*-exact CKKS to be the subset of admissible circuits in CKKS that are* Δ'*-correctable.*

Scale back to original size.

Contributions in More Detail
- (δ,r)-exact CKKS -

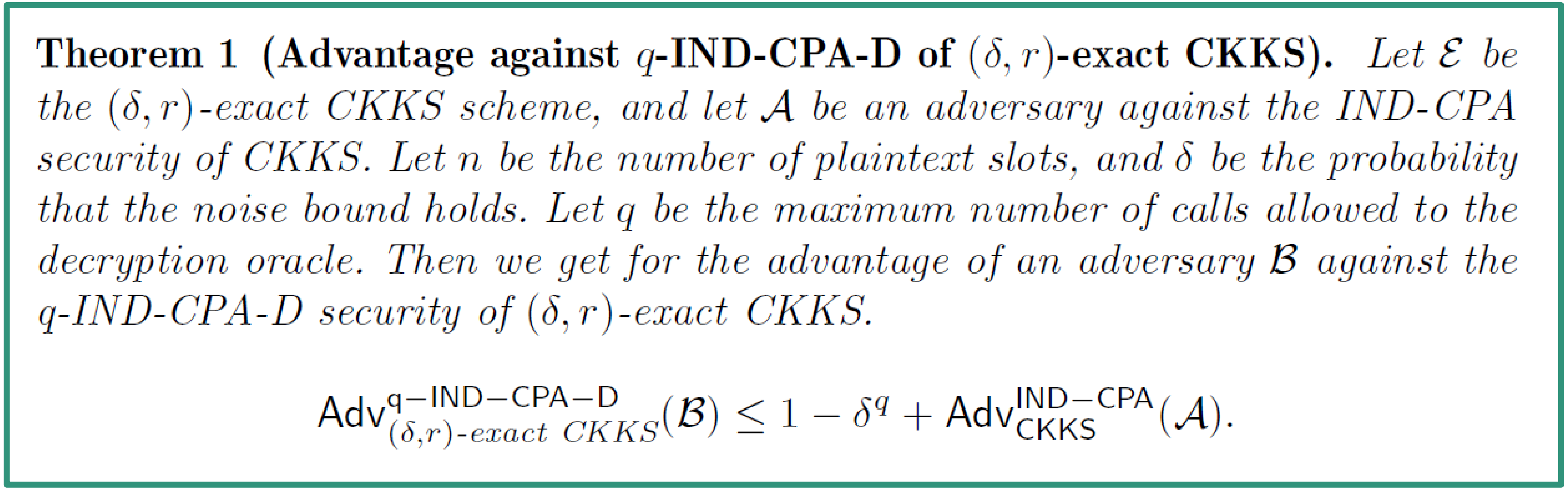


Definition 9 ((δ, r**)-exact CKKS).** *Let* δ *be probability of a noise bound* $B = \alpha\sqrt{2n}\sqrt{\sigma^2_{n(\mathtt{ct})[0]} + N\sigma^2_{n(\mathtt{ct})[1]}\sigma^2_s}$ *holding, for* $\alpha \in \mathbb{R}$*. Let* $r = \lceil \log(B) \rceil + 1$*, and let* $\Delta' = 2^r$*. Let* $\mathcal{E} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec}_{\delta,r}, \mathtt{Eval})$ *be a fully homomorphic encryption scheme, parameterized by the same parameters as CKKS, and the additional parameters* δ, B, r*, and* Δ'*, where the algorithms* $\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Eval}$ *are the same as in CKKS and* $\mathtt{Dec}_{\delta,r}$ *is defined as follows*

$$\underline{\mathtt{Dec}_{\delta,r}(\mathtt{ct}, \mathtt{sk})} : \textit{return } \Delta' \left\lceil \frac{1}{\Delta'} \mathtt{Decode}([\langle \mathtt{ct}, \mathtt{sk} \rangle]_Q) \right\rfloor ,$$

We define the set of admissible circuits of (δ, r)*-exact CKKS to be the subset of admissible circuits in CKKS that are* Δ'*-correctable.*

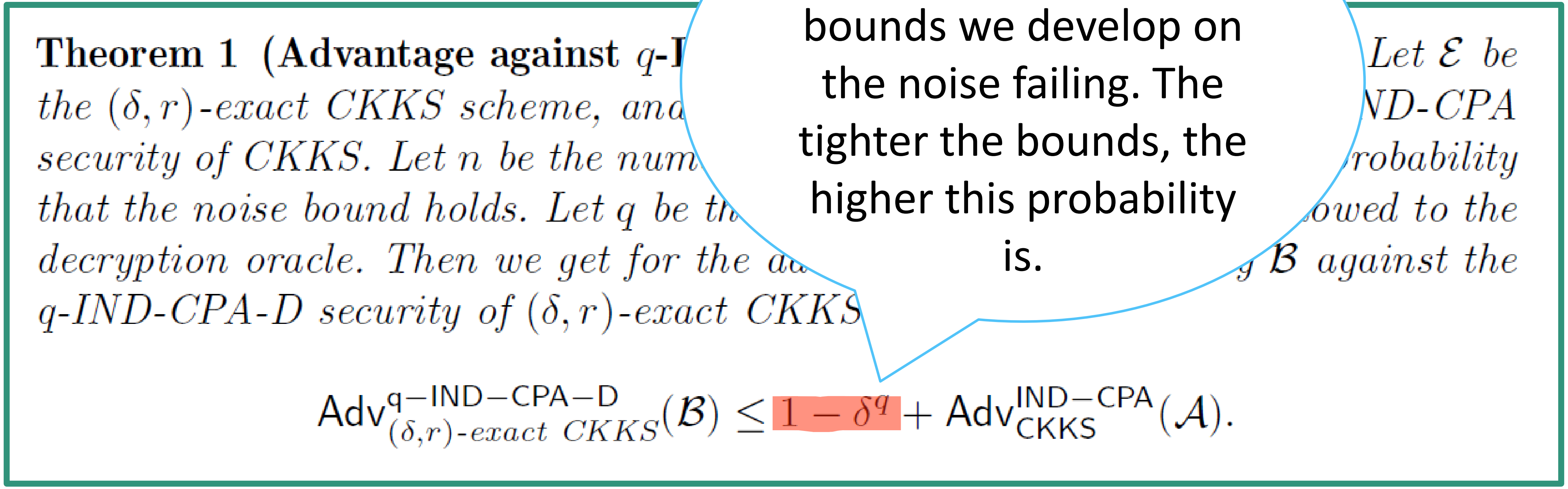
Contributions in More Detail
- (δ,r)-exact CKKS -



Theorem 1 (Advantage against q-IND-CPA-D of (δ, r)-exact CKKS). *Let $\mathcal{E}$ be the (δ, r)-exact CKKS scheme, and let $\mathcal{A}$ be an adversary against the IND-CPA security of CKKS. Let n be the number of plaintext slots, and δ be the probability that the noise bound holds. Let q be the maximum number of calls allowed to the decryption oracle. Then we get for the advantage of an adversary $\mathcal{B}$ against the q-IND-CPA-D security of (δ, r)-exact CKKS.*

$$\mathsf{Adv}^{\mathsf{q-IND-CPA-D}}_{(\delta,r)\text{-}exact\ CKKS}(\mathcal{B}) \leq 1 - \delta^q + \mathsf{Adv}^{\mathsf{IND-CPA}}_{\mathsf{CKKS}}(\mathcal{A}).$$

Contributions in More Detail
- (δ,r)-exact CKKS -



Theorem 1 (Advantage against *q*-**I** … *Let ℰ be*
the (δ, *r*)-*exact CKKS scheme, and* … *ND-CPA*
security of CKKS. Let n be the num … *robability*
that the noise bound holds. Let q be th … *owed to the*
decryption oracle. Then we get for the ad … *g ℬ against the*
q-IND-CPA-D security of (δ, *r*)-*exact CKKS*

$$\mathsf{Adv}^{q-\mathsf{IND}-\mathsf{CPA}-\mathsf{D}}_{(\delta,r)\text{-}exact\ CKKS}(\mathcal{B}) \le 1-\delta^{q} + \mathsf{Adv}^{\mathsf{IND}-\mathsf{CPA}}_{\mathsf{CKKS}}(\mathcal{A}).$$

This is the probability of the theoretical bounds we develop on the noise failing. The tighter the bounds, the higher this probability is.

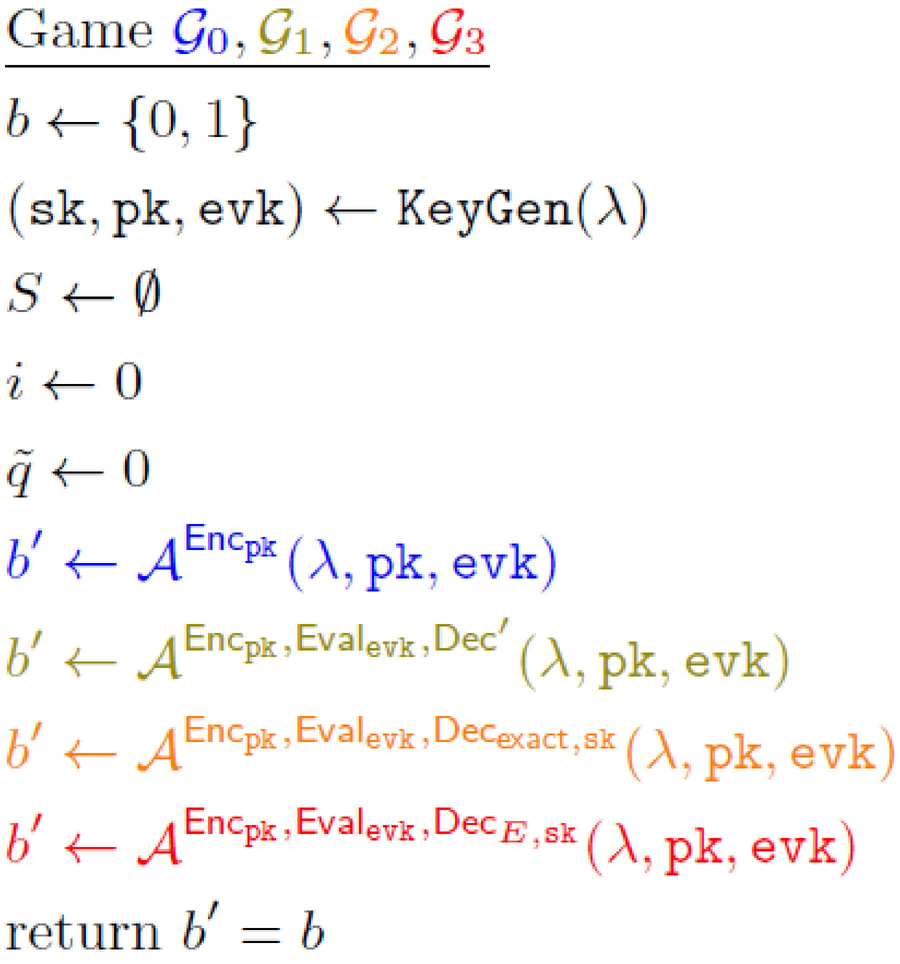
Contributions in More Detail
- (δ,r)-exact CKKS -

We prove the theorem by a game based proof, where we reduce in two steps first to r-exact CKKS and then to “normal” CKKS.

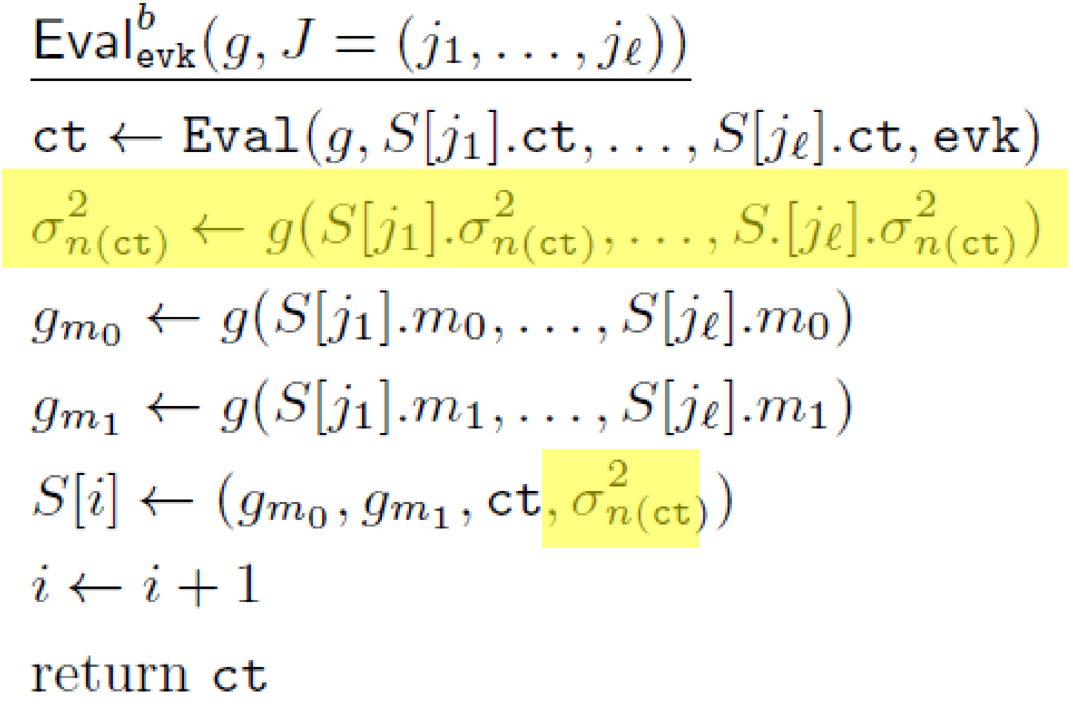
```
Game $\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$
$b \leftarrow \{0, 1\}$
$(\mathsf{sk}, \mathsf{pk}, \mathsf{evk}) \leftarrow \mathtt{KeyGen}(\lambda)$
$S \leftarrow \emptyset$
$i \leftarrow 0$
$\tilde{q} \leftarrow 0$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}}(\lambda, \mathsf{pk}, \mathsf{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}'}(\lambda, \mathsf{pk}, \mathsf{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}_{\mathsf{exact}, \mathsf{sk}}}(\lambda, \mathsf{pk}, \mathsf{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}_{E, \mathsf{sk}}}(\lambda, \mathsf{pk}, \mathsf{evk})$
return $b' = b$
```

```
$\mathsf{Enc}^{b}_{\mathsf{pk}}(m_0, m_1)$
$\mathtt{ct} \leftarrow \mathsf{Enc}(m_b, \mathsf{pk})$
$\sigma^2_{n(\mathtt{ct})} \leftarrow \left(\frac{1}{6}, \frac{1}{12}\right)$
$S[i] \leftarrow (m_0, m_1, \mathtt{ct}, \sigma^2_{n(\mathtt{ct})})$
$i \leftarrow i + 1$
return $\mathtt{ct}$
```

Game G_3



```
Game $\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$
$b \leftarrow \{0,1\}$
$(\mathtt{sk}, \mathtt{pk}, \mathtt{evk}) \leftarrow \mathtt{KeyGen}(\lambda)$
$S \leftarrow \emptyset$
$i \leftarrow 0$
$\tilde{q} \leftarrow 0$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}}(\lambda, \mathrm{pk}, \mathrm{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}'}(\lambda, \mathrm{pk}, \mathrm{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}_{\mathsf{exact}, \mathsf{sk}}}(\lambda, \mathrm{pk}, \mathrm{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}_{E, \mathsf{sk}}}(\lambda, \mathrm{pk}, \mathrm{evk})$
return $b' = b$
```



```
$\mathsf{Eval}^{b}_{\mathsf{evk}}(g, J = (j_1, \ldots, j_\ell))$
$\mathtt{ct} \leftarrow \mathtt{Eval}(g, S[j_1].\mathtt{ct}, \ldots, S[j_\ell].\mathtt{ct}, \mathtt{evk})$
$\sigma^2_{n(\mathtt{ct})} \leftarrow g(S[j_1].\sigma^2_{n(\mathtt{ct})}, \ldots, S.[j_\ell].\sigma^2_{n(\mathtt{ct})})$
$g_{m_0} \leftarrow g(S[j_1].m_0, \ldots, S[j_\ell].m_0)$
$g_{m_1} \leftarrow g(S[j_1].m_1, \ldots, S[j_\ell].m_1)$
$S[i] \leftarrow (g_{m_0}, g_{m_1}, \mathtt{ct}, \sigma^2_{n(\mathtt{ct})})$
$i \leftarrow i + 1$
return $\mathtt{ct}$
```

Game G_3

```
Game $\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$
$b \leftarrow \{0,1\}$
$(\mathtt{sk}, \mathtt{pk}, \mathtt{evk}) \leftarrow \mathtt{KeyGen}(\lambda)$
$S \leftarrow \emptyset$
$i \leftarrow 0$
$\tilde{q} \leftarrow 0$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}}(\lambda, \mathtt{pk}, \mathtt{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}'}(\lambda, \mathtt{pk}, \mathtt{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}_{\mathsf{exact},\mathsf{sk}}}(\lambda, \mathtt{pk}, \mathtt{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}_{E,\mathsf{sk}}}(\lambda, \mathtt{pk}, \mathtt{evk})$
return $b' = b$
```

```
$\mathsf{Dec}^b_{\mathsf{E},\mathsf{sk}}(i)$
If $S[i].m_0 = S[i].m_1$ and $\tilde{q} \leq q$ :
    $B \leftarrow \alpha\sqrt{2n}\sqrt{\sigma^2_{S[i].n(\mathtt{ct})[0]} + N\sigma^2_{S[i].n(\mathtt{ct})[1]}\sigma^2_s}$
    $\Delta' \leftarrow 2^{\lceil \log B \rceil + 1}$
    $\tilde{q} \leftarrow \tilde{q} + 1$
    If $||\mathtt{Decode}([\langle S[i].\mathtt{ct}, \mathtt{sk}\rangle]_Q) - S[i].m_b||_\infty \leq B$ :
        return $\mathtt{Dec}_{\delta,r}(S[i].\mathtt{ct}, \mathtt{sk})$
    Else
        $\mathsf{BAD} \leftarrow \mathtt{true}$
        return $\mathtt{Dec}_{\delta,r}(S[i].\mathtt{ct}, \mathtt{sk})$
Else:
    return $\perp$
```

Game G$_3$

```
Game $\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$
$b \leftarrow \{0,1\}$
$(\mathtt{sk}, \mathtt{pk}, \mathtt{evk}) \leftarrow \mathtt{KeyGen}(\lambda)$
$S \leftarrow \emptyset$
$i \leftarrow 0$
$\tilde{q} \leftarrow 0$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}}(\lambda, \mathrm{pk}, \mathrm{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}'}(\lambda, \mathrm{pk}, \mathrm{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}_{\mathsf{exact},\mathsf{sk}}}(\lambda, \mathrm{pk}, \mathrm{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}_{E,\mathsf{sk}}}(\lambda, \mathrm{pk}, \mathrm{evk})$
return $b' = b$
```

```
$\mathsf{Dec}^{b}_{\mathsf{exact},\mathsf{sk}}(i)$
If $S[i].m_0 = S[i].m_1$ and $\tilde{q} \leq q$ :
    $B \leftarrow \alpha\sqrt{2n}\sqrt{\sigma^2_{S[i].n(\mathtt{ct})[0]} + N\sigma^2_{S[i].n(\mathtt{ct})[1]}\sigma^2_s}$
    $\Delta' \leftarrow 2^{\lceil \log B \rceil + 1}$
    $\tilde{q} \leftarrow \tilde{q} + 1$
    If $||\mathtt{Decode}([\langle S[i].\mathtt{ct}, \mathtt{sk}\rangle]_Q) - S[i].m_b||_\infty \leq B$ :
        return $\mathtt{Dec}_r(S[i].\mathtt{ct}, \mathtt{sk})$
    Else
        $\mathsf{BAD} \leftarrow \mathtt{true}$
        return $\mathtt{Dec}_r(S[i].\mathtt{ct}, \mathtt{sk})$
Else:
    return $\perp$
```

$$|\Pr(\mathcal{G}_3 \Rightarrow 1) - \Pr(\mathcal{G}_2 \Rightarrow 1)| = \Pr(\mathsf{BAD} \leftarrow \mathsf{true}) = 1 - \delta^q$$

Game G_2

```
Game $\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$
$b \leftarrow \{0,1\}$
$(\mathtt{sk}, \mathtt{pk}, \mathtt{evk}) \leftarrow \mathtt{KeyGen}(\lambda)$
$S \leftarrow \emptyset$
$i \leftarrow 0$
$\tilde{q} \leftarrow 0$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}}(\lambda, \mathrm{pk}, \mathrm{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}'}(\lambda, \mathrm{pk}, \mathrm{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}_{\mathsf{exact},\mathsf{sk}}}(\lambda, \mathrm{pk}, \mathrm{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathsf{pk}}, \mathsf{Eval}_{\mathsf{evk}}, \mathsf{Dec}_{E,\mathsf{sk}}}(\lambda, \mathrm{pk}, \mathrm{evk})$
return $b' = b$
```

```
$\mathsf{Dec}'^{b}(i)$
If $S[i].m_0 = S[i].m_1 \wedge \tilde{q} \leq q$:
    $B \leftarrow \alpha\sqrt{2n}\sqrt{\sigma^2_{n(\mathtt{ct})[0]} + N\sigma^2_{n(\mathtt{ct})[1]}\sigma_s^2}$
    $\tilde{q} \leftarrow \tilde{q} + 1$
    $\Delta' \leftarrow 2^{\lceil \log(B) \rceil + 1}$
    return $\Delta' \left\lceil \frac{1}{\Delta'} S[i].m_b \right\rfloor$
Else:
    return $\perp$
```

$$\left|\Pr(\mathcal{G}_2 \Rightarrow 1) - \Pr(\mathcal{G}_1 \Rightarrow 1)\right| = 0$$

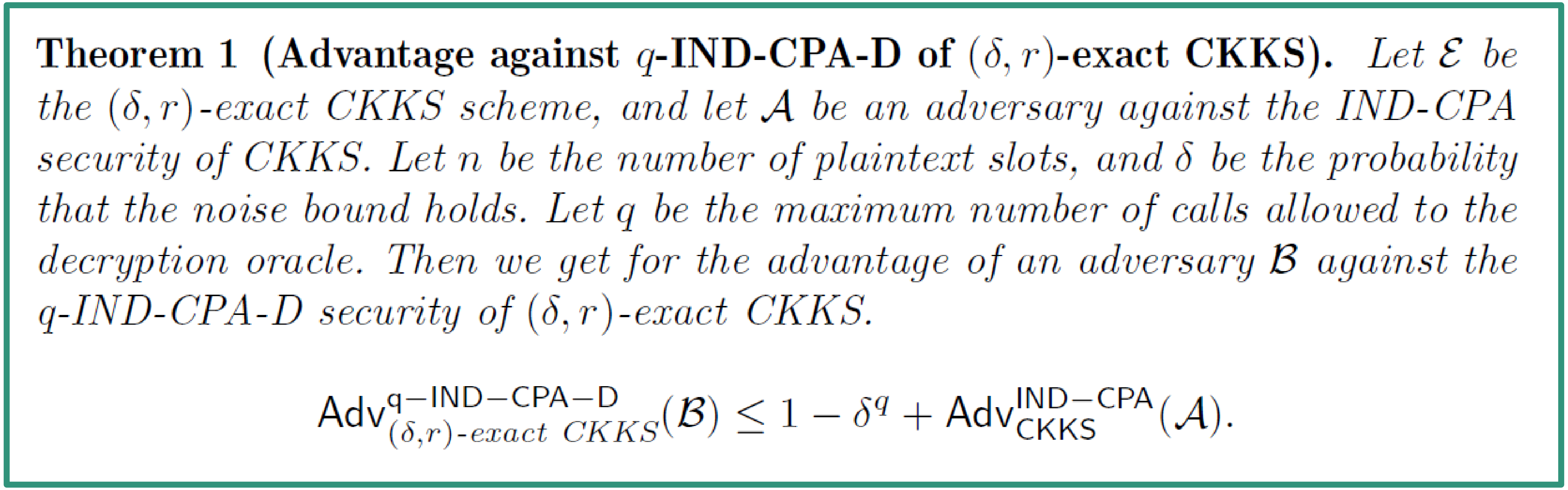
Game G_1

```
Game $\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$
$b \leftarrow \{0, 1\}$
$(\mathtt{sk}, \mathtt{pk}, \mathtt{evk}) \leftarrow \mathtt{KeyGen}(\lambda)$
$S \leftarrow \emptyset$
$i \leftarrow 0$
$\tilde{q} \leftarrow 0$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathtt{pk}}}(\lambda, \mathtt{pk}, \mathtt{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathtt{pk}}, \mathsf{Eval}_{\mathtt{evk}}, \mathsf{Dec}'}(\lambda, \mathtt{pk}, \mathtt{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathtt{pk}}, \mathsf{Eval}_{\mathtt{evk}}, \mathsf{Dec}_{\mathsf{exact}, \mathsf{sk}}}(\lambda, \mathtt{pk}, \mathtt{evk})$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{\mathtt{pk}}, \mathsf{Eval}_{\mathtt{evk}}, \mathsf{Dec}_{E, \mathsf{sk}}}(\lambda, \mathtt{pk}, \mathtt{evk})$
return $b' = b$
```

$$\Pr(\mathcal{G}_1 \Rightarrow 1) = \Pr(\mathcal{G}_0 \Rightarrow 1) = \Pr(\text{IND-CPA} \Rightarrow 1)$$

Game G_0

Contributions in More Detail
- (δ,r)-exact CKKS -



Theorem 1 (Advantage against q-IND-CPA-D of (δ, r)-exact CKKS). *Let $\mathcal{E}$ be the (δ, r)-exact CKKS scheme, and let $\mathcal{A}$ be an adversary against the IND-CPA security of CKKS. Let n be the number of plaintext slots, and δ be the probability that the noise bound holds. Let q be the maximum number of calls allowed to the decryption oracle. Then we get for the advantage of an adversary $\mathcal{B}$ against the q-IND-CPA-D security of (δ, r)-exact CKKS.*

$$\mathsf{Adv}^{\mathsf{q-IND-CPA-D}}_{(\delta,r)\text{-}exact\ CKKS}(\mathcal{B}) \leq 1 - \delta^q + \mathsf{Adv}^{\mathsf{IND-CPA}}_{\mathsf{CKKS}}(\mathcal{A}).$$

Open Questions

➢Investigate the condition of correctability.

- How large is the class of correctable circuits?
- Can any circuit be made correctable?
- If so, at what cost?

➢Is it possible to make the proof go through without the condition of correctability?

Thank you for your attention!

Any Questions?

https://eprint.iacr.org/2024/853

Lea.nurnberger@ntnu.no