Revisiting the Security of Approximate FHE with Noise-Flooding Countermeasures

PKC'25

Flavio Bergamaschi, **Anamaria Costache**, Dana Dachman-Soled, Hunter Kippen, Lucas LaBuff and Rui Tang

May 14, 2025

NTNU

Introduction

What is (Fully) Homomorphic Encryption?

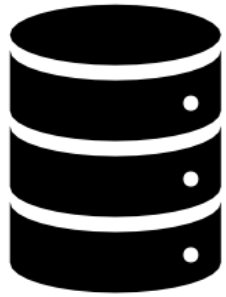
$(sk, pk) \leftarrow KeyGen(1^{\lambda})$

$ct \leftarrow Enc(m, pk)$

$ct' \leftarrow f(ct) = Enc(f(m), pk)$

(ct, f, pk)

ct'



$f(m) \leftarrow Dec(ct', sk)$

The CKKS scheme

- FHE approximate scheme introduced in 2017 [CKKS17]

Homomorphic Encryption for Arithmetic of Approximate Numbers, Jung Hee Cheon, Andrey Kim, Miran Kim and Yongsoo Song, Asiacrypt'17

On the Security of Homomorphic Encryption on Approximate Numbers, Baiyu Li and Daniele Micciancio, Eurocrypt 2021

The CKKS scheme

- FHE approximate scheme introduced in 2017 [CKKS17]
- In particular,

$$\mathtt{Dec}(\mathtt{Enc}(m, \mathtt{pk})) = m + e,$$

  for some noise e, even for fresh encryptions!

Homomorphic Encryption for Arithmetic of Approximate Numbers, Jung Hee Cheon, Andrey Kim, Miran Kim and Yongsoo Song, Asiacrypt'17

On the Security of Homomorphic Encryption on Approximate Numbers, Baiyu Li and Daniele Micciancio, Eurocrypt 2021

The CKKS scheme

- FHE approximate scheme introduced in 2017 [CKKS17]
- In particular,

$$\mathtt{Dec}(\mathtt{Enc}(m, \mathtt{pk})) = m + e,$$

  for some noise e, even for fresh encryptions!
- Approximate correctness leads to an efficient key recovery attack [LM21]

Homomorphic Encryption for Arithmetic of Approximate Numbers, Jung Hee Cheon, Andrey Kim, Miran Kim and Yongsoo Song, Asiacrypt'17

On the Security of Homomorphic Encryption on Approximate Numbers, Baiyu Li and Daniele Micciancio, Eurocrypt 2021

The CKKS scheme

- FHE approximate scheme introduced in 2017 [CKKS17]
- In particular,

$$\mathtt{Dec}(\mathtt{Enc}(m, \mathtt{pk})) = m + e,$$

  for some noise e, even for fresh encryptions!
- Approximate correctness leads to an efficient key recovery attack [LM21]
- Authors capture this in a new security model, IND-CPA-D

Homomorphic Encryption for Arithmetic of Approximate Numbers, Jung Hee Cheon, Andrey Kim, Miran Kim and Yongsoo Song, Asiacrypt'17

On the Security of Homomorphic Encryption on Approximate Numbers, Baiyu Li and Daniele Micciancio, Eurocrypt 2021

Countermeasure: noise flooding

- In a follow-up work, Li et al. [LMSS22] analyse the effectiveness of noise flooding as a countermeasure to the attack presented in [LM21]

Securing Approximate Homomorphic Encryption Using Differential Privacy, Baiyu Li, Daniele Micciancio, Mark Schultz and Jessica Sorrell, Crypto 2022

Countermeasure: noise flooding

- In a follow-up work, Li et al. [LMSS22] analyse the effectiveness of noise flooding as a countermeasure to the attack presented in [LM21]
  - Using Differential Privacy techniques, can prove security

---

Securing Approximate Homomorphic Encryption Using Differential Privacy, Baiyu Li, Daniele Micciancio, Mark Schultz and Jessica Sorrell, Crypto 2022

Countermeasure: noise flooding

- In a follow-up work, Li et al. [LMSS22] analyse the effectiveness of noise flooding as a countermeasure to the attack presented in [LM21]
  - Using Differential Privacy techniques, can prove security
- But this leads to a large efficiency loss

Securing Approximate Homomorphic Encryption Using Differential Privacy, Baiyu Li, Daniele Micciancio, Mark Schultz and Jessica Sorrell, Crypto 2022

Countermeasure: noise flooding

- In a follow-up work, Li et al. [LMSS22] analyse the effectiveness of noise flooding as a countermeasure to the attack presented in [LM21]
  - Using Differential Privacy techniques, can prove security
- But this leads to a large efficiency loss
  - The resulting modified CKKS loses message precision bits

Securing Approximate Homomorphic Encryption Using Differential Privacy, Baiyu Li, Daniele Micciancio, Mark Schultz and Jessica Sorrell, Crypto 2022

Our work

Objectives

- Goal is to investigate the concrete security degradation of the CKKS scheme in the presence of t decryptions, with noise flooding of some variance ρ^2

Objectives

- Goal is to investigate the concrete security degradation of the CKKS scheme in the presence of t decryptions, with noise flooding of some variance ρ^2
  - Optimal setting is to set ρ^2 equal to the noise already present (thus losing 1 bit of message precision)

Objectives

- Goal is to investigate the concrete security degradation of the CKKS scheme in the presence of t decryptions, with noise flooding of some variance ρ^2
  - Optimal setting is to set ρ^2 equal to the noise already present (thus losing 1 bit of message precision)
  - On the other side of the spectrum, noise flood for provable security

Objectives

- Goal is to investigate the concrete security degradation of the CKKS scheme in the presence of t decryptions, with noise flooding of some variance ρ^2
  - Optimal setting is to set ρ^2 equal to the noise already present (thus losing 1 bit of message precision)
  - On the other side of the spectrum, noise flood for provable security
- Our aim is to present trade-offs between:

Objectives

- Goal is to investigate the concrete security degradation of the CKKS scheme in the presence of t decryptions, with noise flooding of some variance ρ^2
  - Optimal setting is to set ρ^2 equal to the noise already present (thus losing 1 bit of message precision)
  - On the other side of the spectrum, noise flood for provable security
- Our aim is to present trade-offs between:
  - Number of allowed decryptions queries before refreshing keys

Objectives

- Goal is to investigate the concrete security degradation of the CKKS scheme in the presence of t decryptions, with noise flooding of some variance ρ^2
  - Optimal setting is to set ρ^2 equal to the noise already present (thus losing 1 bit of message precision)
  - On the other side of the spectrum, noise flood for provable security
- Our aim is to present trade-offs between:
  - Number of allowed decryptions queries before refreshing keys
  - Variance of the noise flooding noise (how much message precision we lose)

Objectives

- Goal is to investigate the concrete security degradation of the CKKS scheme in the presence of t decryptions, with noise flooding of some variance ρ^2
  - Optimal setting is to set ρ^2 equal to the noise already present (thus losing 1 bit of message precision)
  - On the other side of the spectrum, noise flood for provable security
- Our aim is to present trade-offs between:
  - Number of allowed decryptions queries before refreshing keys
  - Variance of the noise flooding noise (how much message precision we lose)
  - Concrete security of the scheme after a number of decryptions have been observed

LWE with side information

- A framework for cryptanalysis of lattice-based schemes when side information (hints) about the secret and/ or noise is available [DDGR20]

LWE with Side Information: Attacks and Concrete Security Estimation, Dana Dachman-Soled, Léo Ducas, Huijing Gong and Mélissa Rossi, Crypto'20

LWE with side information

- A framework for cryptanalysis of lattice-based schemes when side information (hints) about the secret and/ or noise is available [DDGR20]
- Allows progressive integration of hints on the secret s before running a final lattice reduction step

---

LWE with Side Information: Attacks and Concrete Security Estimation, Dana Dachman-Soled, Léo Ducas, Huijing Gong and Mélissa Rossi, Crypto'20

LWE with side information

- A framework for cryptanalysis of lattice-based schemes when side information (hints) about the secret and/ or noise is available [DDGR20]
- Allows progressive integration of hints on the secret s before running a final lattice reduction step
  - Perfect hints: $\langle s, v \rangle = l$

LWE with Side Information: Attacks and Concrete Security Estimation, Dana Dachman-Soled, Léo Ducas, Huijing Gong and Mélissa Rossi, Crypto'20

LWE with side information

- A framework for cryptanalysis of lattice-based schemes when side information (hints) about the secret and/ or noise is available [DDGR20]
- Allows progressive integration of hints on the secret s before running a final lattice reduction step
  - Perfect hints: $\langle s, v \rangle = l$
  - Modular hints : $\langle s, v \rangle = l \pmod{k}$

LWE with Side Information: Attacks and Concrete Security Estimation, Dana Dachman-Soled, Léo Ducas, Huijing Gong and Mélissa Rossi, Crypto'20

LWE with side information

- A framework for cryptanalysis of lattice-based schemes when side information (hints) about the secret and/ or noise is available [DDGR20]
- Allows progressive integration of hints on the secret s before running a final lattice reduction step
  - Perfect hints: $\langle s, v \rangle = l$
  - Modular hints : $\langle s, v \rangle = l \pmod{k}$
  - Approximate hints : $\langle s, v \rangle = l + \epsilon$

LWE with Side Information: Attacks and Concrete Security Estimation, Dana Dachman-Soled, Léo Ducas, Huijing Gong and Mélissa Rossi, Crypto'20

LWE with side information

- A framework for cryptanalysis of lattice-based schemes when side information (hints) about the secret and/ or noise is available [DDGR20]
- Allows progressive integration of hints on the secret s before running a final lattice reduction step
  - Perfect hints: $\langle s, v \rangle = l$
  - Modular hints : $\langle s, v \rangle = l \pmod{k}$
  - Approximate hints : $\langle s, v \rangle = l + \epsilon$
  - Short vector hints : $v \in \Lambda$

LWE with Side Information: Attacks and Concrete Security Estimation, Dana Dachman-Soled, Léo Ducas, Huijing Gong and Mélissa Rossi, Crypto'20

Methodology

- We model the leakage on the noise in CKKS decryption as hints

Methodology

- We model the leakage on the noise in CKKS decryption as hints
  - Decryption of a (fresh) CKKS ciphertext:

$$\texttt{Dec}(\texttt{sk}, \texttt{ct}) = m + s \cdot e + E,$$

for some noise terms e and E

Methodology

- We model the leakage on the noise in CKKS decryption as hints
  - Decryption of a (fresh) CKKS ciphertext:

$$\texttt{Dec}(\texttt{sk}, \texttt{ct}) = m + s \cdot e + E,$$

    for some noise terms e and E
- Noise-flood by $\rho^2, 100 \cdot \rho^2, t \cdot \rho^2$

Methodology

- We model the leakage on the noise in CKKS decryption as hints
  - Decryption of a (fresh) CKKS ciphertext:

$$\texttt{Dec}(\texttt{sk}, \texttt{ct}) = m + s \cdot e + E,$$

    for some noise terms e and E
- Noise-flood by $\rho^2, 100 \cdot \rho^2, t \cdot \rho^2$
  - ρ^2 is an average-case estimate of the variance of the underlying noise, as estimated by [CCHMOP22]

Methodology

- We model the leakage on the noise in CKKS decryption as hints
  - Decryption of a (fresh) CKKS ciphertext:

$$\texttt{Dec}(\texttt{sk}, \texttt{ct}) = m + s \cdot e + E,$$

    for some noise terms e and E
- Noise-flood by $\rho^2, 100 \cdot \rho^2, t \cdot \rho^2$
  - ρ^2 is an average-case estimate of the variance of the underlying noise, as estimated by [CCHMOP22]
- Validate by running estimates for noise flooding levels suggested in [LMSS22]

Methodology

- We model the leakage on the noise in CKKS decryption as hints
  - Decryption of a (fresh) CKKS ciphertext:

$$\texttt{Dec}(\texttt{sk}, \texttt{ct}) = m + s \cdot e + E,$$

    for some noise terms e and E
- Noise-flood by $\rho^2, 100 \cdot \rho^2, t \cdot \rho^2$
  - ρ^2 is an average-case estimate of the variance of the underlying noise, as estimated by [CCHMOP22]
- Validate by running estimates for noise flooding levels suggested in [LMSS22]
- Then run all those attacks for the parameters suggested by `homomorphicencryption.org`

On the precision loss in approximate homomorphic encryption, Anamaria Costache, Benjamin R. Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie and Rachel Player, SAC'22

IND-CPA-DSH

- We capture our threat model in a new security model, IND-CPA-DSH[1]: semi-honest attackers with access to a decryption oracle

[1] SH stands for Semi-Honest

IND-CPA-DSH

- We capture our threat model in a new security model, IND-CPA-DSH[1]: semi-honest attackers with access to a decryption oracle
- Restrict the adversary to evaluating circuits on fresh, independent ciphertext that are each used at most once

[1] SH stands for Semi-Honest

IND-CPA-DSH

- We capture our threat model in a new security model, IND-CPA-DSH[1]: semi-honest attackers with access to a decryption oracle
- Restrict the adversary to evaluating circuits on fresh, independent ciphertext that are each used at most once
- We restrict the adversary to certain classes of circuits

[1] SH stands for Semi-Honest

IND-CPA-DSH

- We capture our threat model in a new security model, IND-CPA-DSH[1]: semi-honest attackers with access to a decryption oracle
- Restrict the adversary to evaluating circuits on fresh, independent ciphertext that are each used at most once
- We restrict the adversary to certain classes of circuits
- We don't allow adaptive queries to `Eval` (state only released to the adversary at the end of the game)

[1] SH stands for Semi-Honest

IND-CPA-DSH

- We capture our threat model in a new security model, IND-CPA-DSH[1]: semi-honest attackers with access to a decryption oracle
- Restrict the adversary to evaluating circuits on fresh, independent ciphertext that are each used at most once
- We restrict the adversary to certain classes of circuits
- We don't allow adaptive queries to `Eval` (state only released to the adversary at the end of the game)
    - Otherwise this allows for adaptive attacks where the adversary can bias the noise distribution

[1] SH stands for Semi-Honest

IND-CPA-DSH

- We capture our threat model in a new security model, IND-CPA-DSH[1]: semi-honest attackers with access to a decryption oracle
- Restrict the adversary to evaluating circuits on fresh, independent ciphertext that are each used at most once
- We restrict the adversary to certain classes of circuits
- We don't allow adaptive queries to `Eval` (state only released to the adversary at the end of the game)
  - Otherwise this allows for adaptive attacks where the adversary can bias the noise distribution
- We release the encryption randomness to the adversary

[1] SH stands for Semi-Honest

IND-CPA-DSH

- We capture our threat model in a new security model, IND-CPA-DSH[1]: semi-honest attackers with access to a decryption oracle
- Restrict the adversary to evaluating circuits on fresh, independent ciphertext that are each used at most once
- We restrict the adversary to certain classes of circuits
- We don't allow adaptive queries to `Eval` (state only released to the adversary at the end of the game)
  - Otherwise this allows for adaptive attacks where the adversary can bias the noise distribution
- We release the encryption randomness to the adversary
  - Whenever releasing the radomness does not lead to a trivial distinguishing attack

[1] SH stands for Semi-Honest

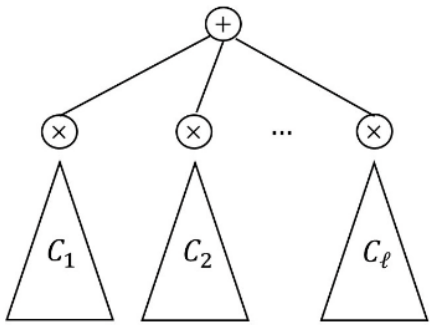
IND-CPA-DSH

- We capture our threat model in a new security model, IND-CPA-DSH[1]: semi-honest attackers with access to a decryption oracle
- Restrict the adversary to evaluating circuits on fresh, independent ciphertext that are each used at most once
- We restrict the adversary to certain classes of circuits
- We don't allow adaptive queries to `Eval` (state only released to the adversary at the end of the game)
  - Otherwise this allows for adaptive attacks where the adversary can bias the noise distribution
- We release the encryption randomness to the adversary
  - Whenever releasing the radomness does not lead to a trivial distinguishing attack
  - In that sense, we are somewhat orthogonal to IND-CPA-D

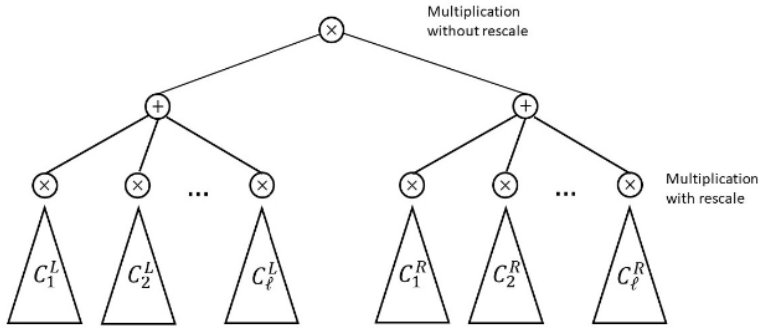
[1] SH stands for Semi-Honest

Classes of circuits

We consider identity circuits, together with Class 1 and 2 circuits.



(a) First Class of Circuits



(b) Second Class of Circuits

Fig. 1: A pictorial representation of the two classes of circuits we consider.

Lattice Attacks

- The hints consist of noisy linear equations on the LWE secret/error, where the noise is sampled form a Gaussian distribution

Lattice Attacks

- The hints consist of noisy linear equations on the LWE secret/error, where the noise is sampled form a Gaussian distribution
- One important difference from [DDGS20] is that we need to compute the **determinant** of a $2n \times 2n$ matrix obtained from t hints for very large values of n

Lattice Attacks

- The hints consist of noisy linear equations on the LWE secret/error, where the noise is sampled form a Gaussian distribution
- One important difference from [DDGS20] is that we need to compute the **determinant** of a $2n \times 2n$ matrix obtained from t hints for very large values of n
- We instead compute the expectation of the determinant

Lattice Attacks

- The hints consist of noisy linear equations on the LWE secret/error, where the noise is sampled form a Gaussian distribution
- One important difference from [DDGS20] is that we need to compute the **determinant** of a $2n \times 2n$ matrix obtained from t hints for very large values of n
- We instead compute the expectation of the determinant
  - Provide a closed-form expression for the determinant

Lattice Attacks

- The hints consist of noisy linear equations on the LWE secret/error, where the noise is sampled form a Gaussian distribution
- One important difference from [DDGS20] is that we need to compute the **determinant** of a $2n \times 2n$ matrix obtained from t hints for very large values of n
- We instead compute the expectation of the determinant
  - Provide a closed-form expression for the determinant
  - Provide experimental validation

Guessing Attacks

- After integrating t hints, can guess n out of $2n$ coordinates of the LWE secret/ error w.h.p.

Guessing Attacks

- After integrating t hints, can guess n out of $2n$ coordinates of the LWE secret/ error w.h.p.
- Then, solve the original LWE system for the remaining n coordinates

Guessing Attacks

- After integrating t hints, can guess n out of $2n$ coordinates of the LWE secret/ error w.h.p.
- Then, solve the original LWE system for the remaining n coordinates
- Keeping track of the covariance matrix requires a $2n \times 2n$ matrix inversion, which is computationally infeasible

Guessing Attacks

- After integrating t hints, can guess n out of $2n$ coordinates of the LWE secret/ error w.h.p.
- Then, solve the original LWE system for the remaining n coordinates
- Keeping track of the covariance matrix requires a $2n \times 2n$ matrix inversion, which is computationally infeasible
- Knowing the distribution of the matrix allows to derive w.h.p. bounds on the trace and eigenvalues

Guessing Attacks

- After integrating t hints, can guess n out of $2n$ coordinates of the LWE secret/ error w.h.p.
- Then, solve the original LWE system for the remaining n coordinates
- Keeping track of the covariance matrix requires a $2n \times 2n$ matrix inversion, which is computationally infeasible
- Knowing the distribution of the matrix allows to derive w.h.p. bounds on the trace and eigenvalues
- These can in turn be used to bound the success probability of the guessing attack

Hybrid Attacks

- Combines the two strategies

Hybrid Attacks

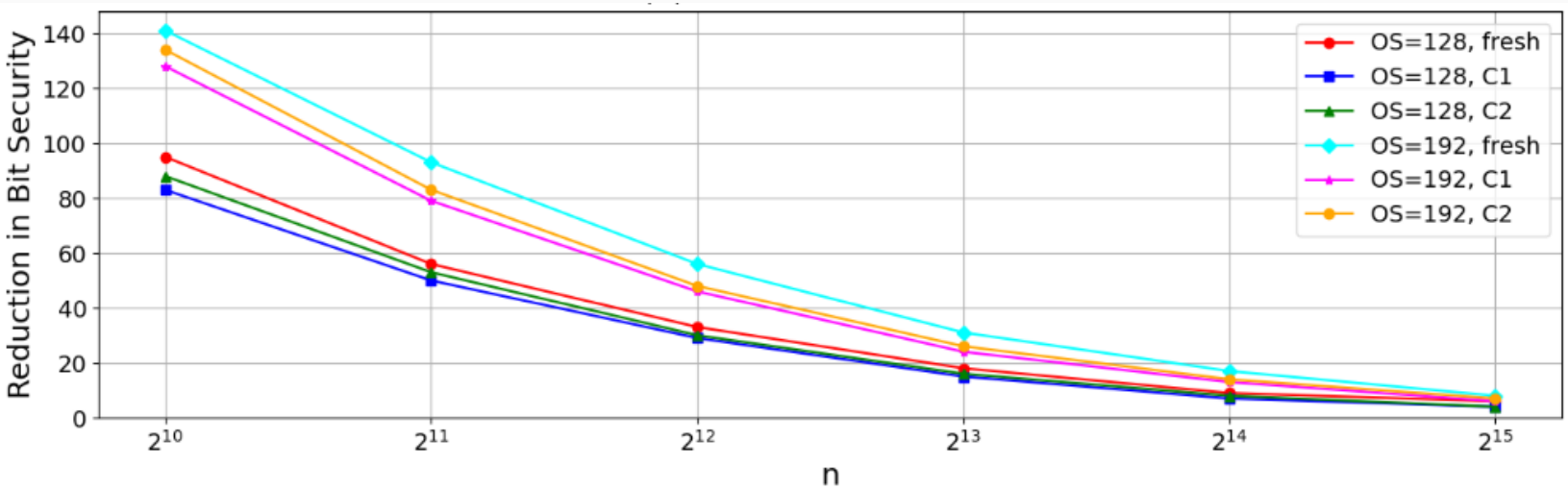
- Combines the two strategies
- Guess $g < n$ coordinates, but cannot guess n of them w.h.p.

Hybrid Attacks

- Combines the two strategies
- Guess $g < n$ coordinates, but cannot guess n of them w.h.p.
- Attacker integrates the g guesses as perfect hints

Results

Hybrid attack trends



(c) Hybrid Attack

General observations

- We validate that there is no security drop in our experiments when using the provably-secure noise-flooding levels

General observations

- We validate that there is no security drop in our experiments when using the provably-secure noise-flooding levels
- Greater reduction in bit-security for lattice attacks for identity circuits vs C1/2

General observations

- We validate that there is no security drop in our experiments when using the provably-secure noise-flooding levels
- Greater reduction in bit-security for lattice attacks for identity circuits vs C1/2
- Greater security reduction for higher target security level vs. lower target security level

General observations

- We validate that there is no security drop in our experiments when using the provably-secure noise-flooding levels
- Greater reduction in bit-security for lattice attacks for identity circuits vs C1/2
- Greater security reduction for higher target security level vs. lower target security level
- Guessing attacks perform significantly better for C1/2 circuits versus identity circuits

General observations

- We validate that there is no security drop in our experiments when using the provably-secure noise-flooding levels
- Greater reduction in bit-security for lattice attacks for identity circuits vs C1/2
- Greater security reduction for higher target security level vs. lower target security level
- Guessing attacks perform significantly better for C1/2 circuits versus identity circuits
- As the value of n increases, the security level drop decreases

Conclusion

Conclusions

- As expected, noise-flooding by the lowest level (ρ^2) incurs the biggest security drop

Conclusions

- As expected, noise-flooding by the lowest level (ρ^2) incurs the biggest security drop
- Noise-flooding by $t \cdot \rho^2$ leads to a very low reduction in the security level, if at all

Conclusions

- As expected, noise-flooding by the lowest level (ρ^2) incurs the biggest security drop
- Noise-flooding by $t \cdot \rho^2$ leads to a very low reduction in the security level, if at all
- Perhaps a less cautious approach is to noise-flood by $\alpha \cdot t \cdot \rho^2$, for some $\alpha \in (0, 1)$

Conclusions

- As expected, noise-flooding by the lowest level (ρ^2) incurs the biggest security drop
- Noise-flooding by $t \cdot \rho^2$ leads to a very low reduction in the security level, if at all
- Perhaps a less cautious approach is to noise-flood by $\alpha \cdot t \cdot \rho^2$, for some $\alpha \in (0, 1)$
    - Think of α as a fine-tuning parameter

Conclusions

- As expected, noise-flooding by the lowest level (ρ^2) incurs the biggest security drop
- Noise-flooding by $t \cdot \rho^2$ leads to a very low reduction in the security level, if at all
- Perhaps a less cautious approach is to noise-flood by $\alpha \cdot t \cdot \rho^2$, for some $\alpha \in (0, 1)$
  - Think of α as a fine-tuning parameter
  - Perhaps for some higher dimensions, a security loss of a few bits is acceptable?

Conclusions

- As expected, noise-flooding by the lowest level (ρ^2) incurs the biggest security drop
- Noise-flooding by $t \cdot \rho^2$ leads to a very low reduction in the security level, if at all
- Perhaps a less cautious approach is to noise-flood by $\alpha \cdot t \cdot \rho^2$, for some $\alpha \in (0, 1)$
  - Think of α as a fine-tuning parameter
  - Perhaps for some higher dimensions, a security loss of a few bits is acceptable?
- The techniques and results of this work can be used to establish key refreshing policies

Thank you!

anamaria.costache@ntnu.no

https://eprint.iacr.org/2024/424