Practical CCA2-Secure and Masked Ring-LWE Implementation

Tobias Oder[1], Tobias Schneider[2], Thomas Pöppelmann[3], Tim Güneysu[1,4]

[1]Ruhr-University Bochum, [2]Université Catholique de Louvain, [3]Infineon Technologies AG, [4]DFKI

CHES 2018

10.09.2018

Horst Görtz Institut für IT-Sicherheit

Motiviation

Ring-LWE

- NIST post-quantum standardization project
- Various NIST submissions are based on Ring-LWE including
  - NewHope
  - LIMA
  - (Kyber)
  - …

Ring-LWE

RUB

- NIST post-quantum standardization project
- Various NIST submissions are based on Ring-LWE including
  - NewHope
  - LIMA
  - (Kyber)
  - …

Previous work

- A masked ring-LWE implementation. *O. Reparaz, S. Sinha Roy, F. Vercauteren, I. Verbauwhede.* CHES 2015
- Additively homomorphic ring-LWE masking. *O. Reparaz, S. Sinha Roy, R. de Clercq, F. Vercauteren, I. Verbauwhede.* PQCrypto 2016

CCA2-Security

- Plain Ring-LWE encryption is only secure against chosen-plaintext attackers (CPA)

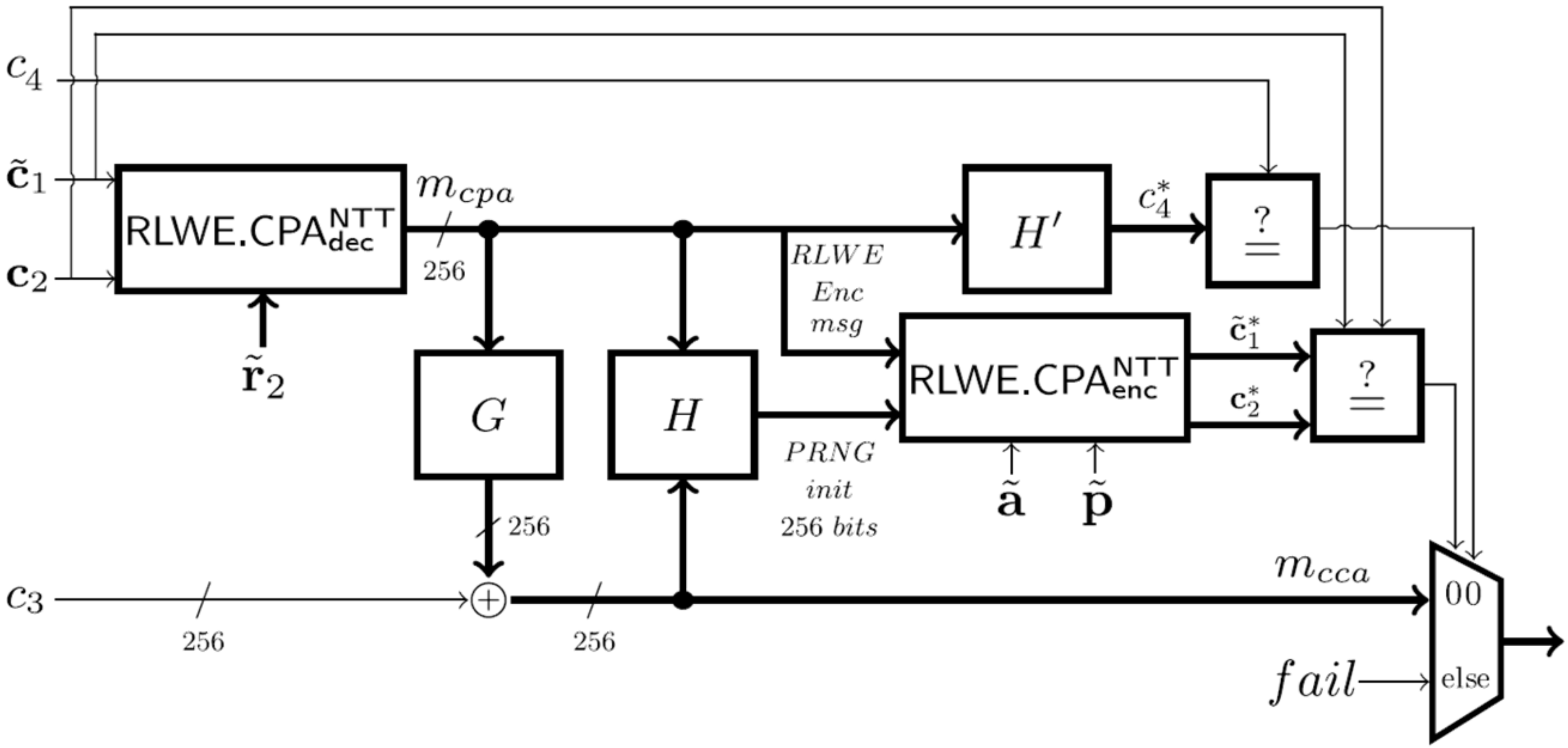
- Many use cases require security against chosen-ciphertext attackers (CCA)

- Generic Fujisaki-Okamoto transform
  - Assumes negligible decryption error
  - Tweak by Targhi and Unruh for post-quantum security [TU16]
  - Expensive re-encryption in decryption

[TU16] E. E. Targhi and D. Unruh. *Post-quantum security of the Fujisaki-Okamoto and OAEP transforms*. TCC 2016

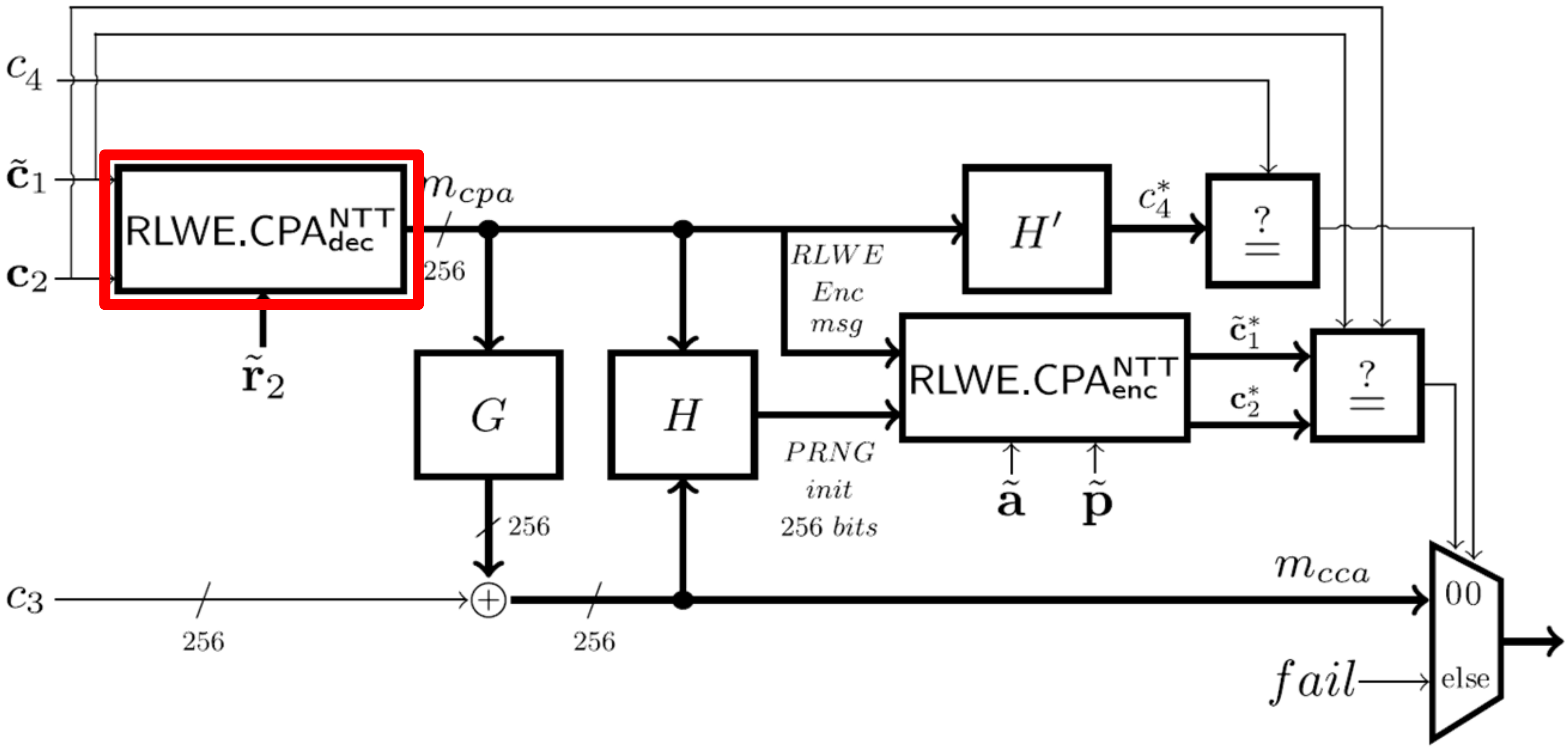
CCA2-Security

CCA2-secure Decryption



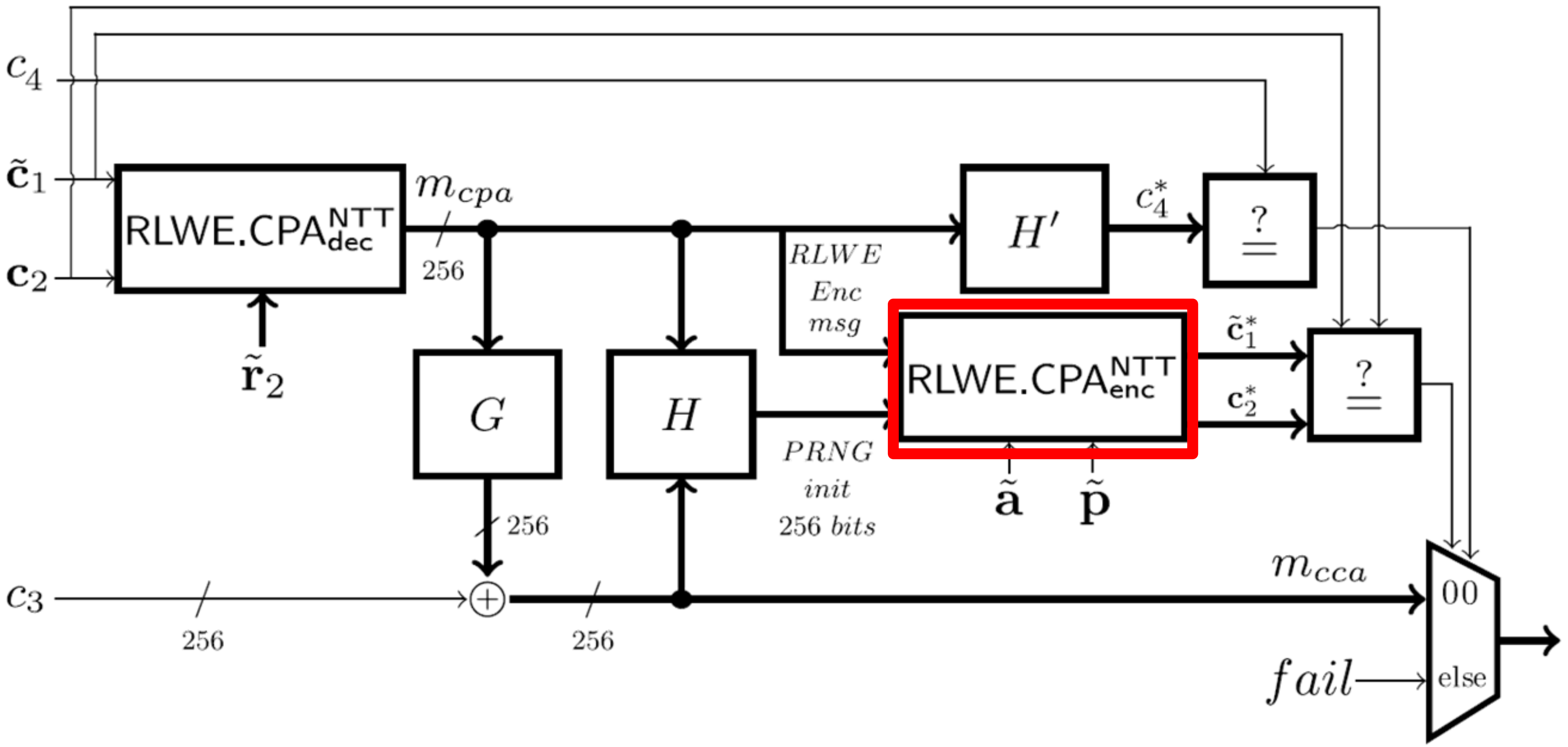
CCA2-Security

CCA2-secure Decryption



CCA2-Security

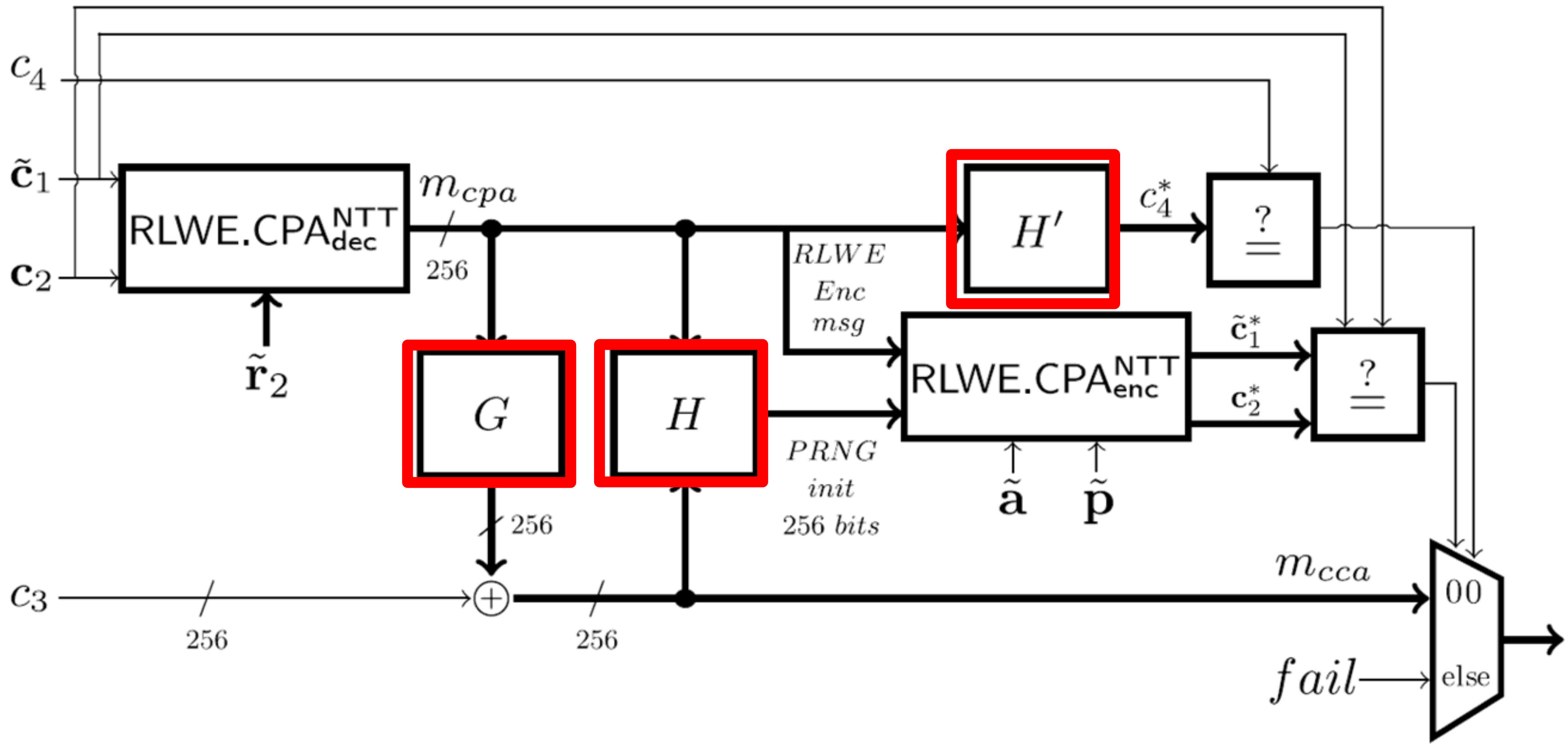
CCA2-secure Decryption



CCA2-Security

RUB

CCA2-secure Decryption



Contribution

Embedded Implementation

- **Our contribution:**

  CCA2-secure first-order masked Ring-LWE implementation

Embedded Implementation

- **Our contribution:**

  CCA2-secure first-order masked Ring-LWE implementation

- Target platform ARM Cortex-M4
  - Constrained computing capabilities/memory

Embedded Implementation

- **Our contribution:**

  CCA2-secure first-order masked Ring-LWE implementation

- Target platform ARM Cortex-M4
  - Constrained computing capabilities/memory

- Secret-independent execution time as countermeasure against timing attacks

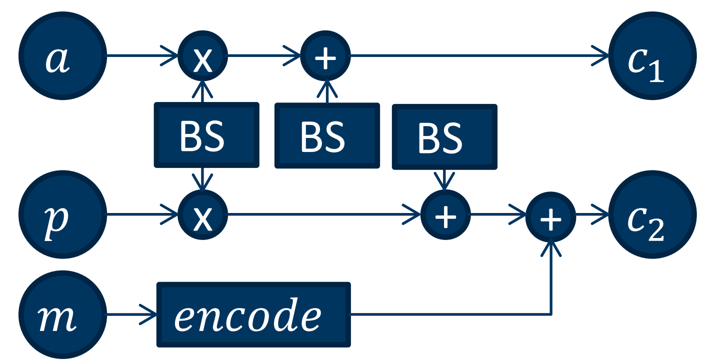
- Masking as countermeasure against Differential Power Analysis
  - Boolean vs. arithmetic

Masking Ring-LWE

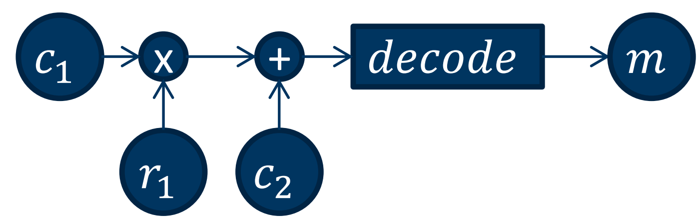
Components to be masked in CCA2-secure Ring-LWE

- PRNG/Hash
- NTT
  - Polynomial multiplication
- Binomial sampler (BS)
- Encoding/Decoding

Ring-LWE CPA Encryption



Ring-LWE CPA Decryption

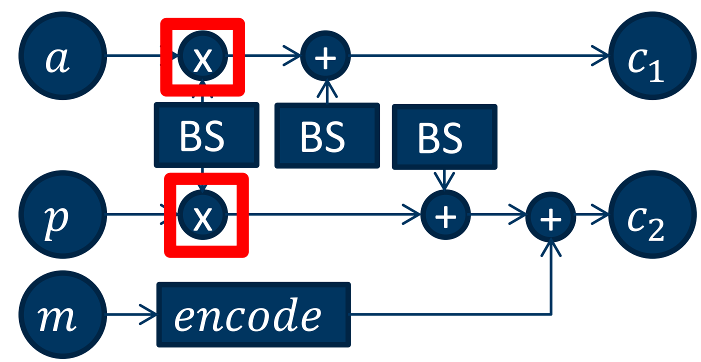


Masking Ring-LWE

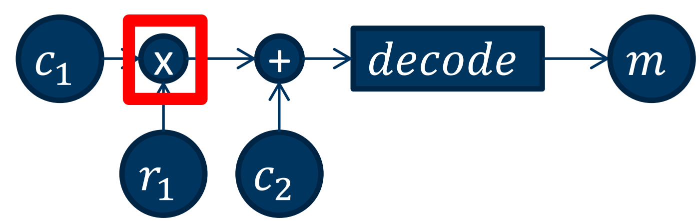
Components to be masked in CCA2-secure Ring-LWE

- PRNG/Hash → [BDPVA10]
- NTT → straight-forward
  - Polynomial multiplication
- Binomial sampler (BS)
- Encoding/Decoding

Ring-LWE CPA Encryption



Ring-LWE CPA Decryption



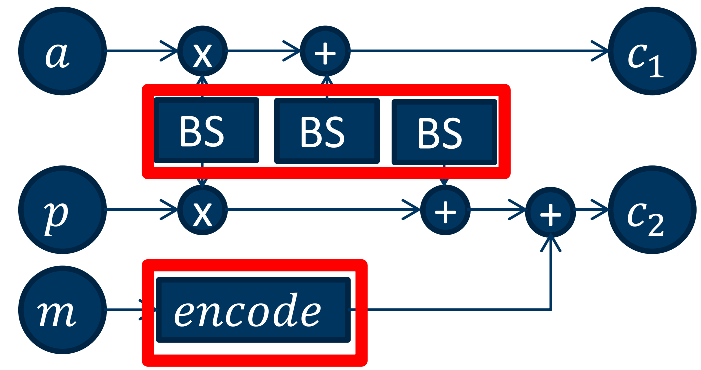
[BDPVA10] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. *Building power analysis resistant implementations of Keccak*. Second SHA-3 candidate conference, 2010

Masking Ring-LWE

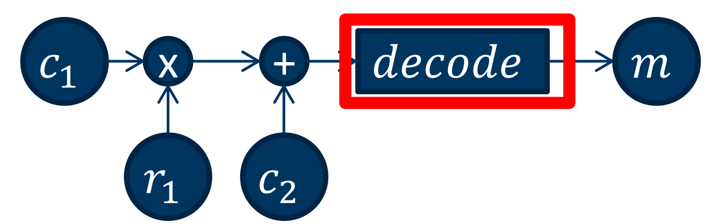
Components to be masked in CCA2-secure Ring-LWE

- PRNG/Hash → [BDPVA10]
- NTT → straight-forward
  - Polynomial multiplication
- Binomial Sampler (BS)
- Encoding/Decoding

Ring-LWE CPA Encryption



Ring-LWE CPA Decryption



[BDPVA10] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. *Building power analysis resistant implementations of Keccak*. Second SHA-3 candidate conference, 2010

Encoding

Masked encoding

- *Encoding* transforms a bit string into a polynomial
  - Without masking:

$$coeff = bit \cdot \lfloor \frac{q}{2} \rceil$$

Masked encoding

RUB

- *Encoding* transforms a bit string into a polynomial
  - Without masking:

$$coeff = bit \cdot \lfloor \frac{q}{2} \rfloor$$

  - With $bit' \oplus bit'' = bit$:

$$coeff' = bit' \cdot \lfloor \frac{q}{2} \rfloor$$

$$coeff'' = bit'' \cdot \left\lfloor \frac{q}{2} \right\rfloor$$

Masked encoding

- *Encoding* transforms a bit string into a polynomial
  - Without masking:

$$coeff = bit \cdot \lfloor \frac{q}{2} \rfloor$$

  - With $bit' \oplus bit'' = bit$:

$$coeff' = bit' \cdot \lfloor \frac{q}{2} \rfloor$$

$$coeff'' = bit'' \cdot \left\lfloor \frac{q}{2} \right\rfloor$$

- q is a odd → $\left\lfloor \frac{q}{2} \right\rfloor + \left\lfloor \frac{q}{2} \right\rfloor \neq q$

Problem: Result is off by one if $bit' = 1$ and $bit'' = 1$

Masked encoding

Solution: Add $bit' \cdot bit''$ to the result

- Compute $bit' \cdot bit''$ by splitting into subshares

$$\left(bit'^{(1)} + bit'^{(2)}\right) \cdot \left(bit''^{(1)} + bit''^{(2)}\right)$$

$$= bit'^{(1)} \cdot bit''^{(1)} + bit'^{(1)} \cdot bit''^{(2)} + bit'^{(2)} \cdot bit''^{(1)} + bit'^{(2)} \cdot bit''^{(2)}$$

- Use fresh randomness to securely sum the cross-products

Decoding

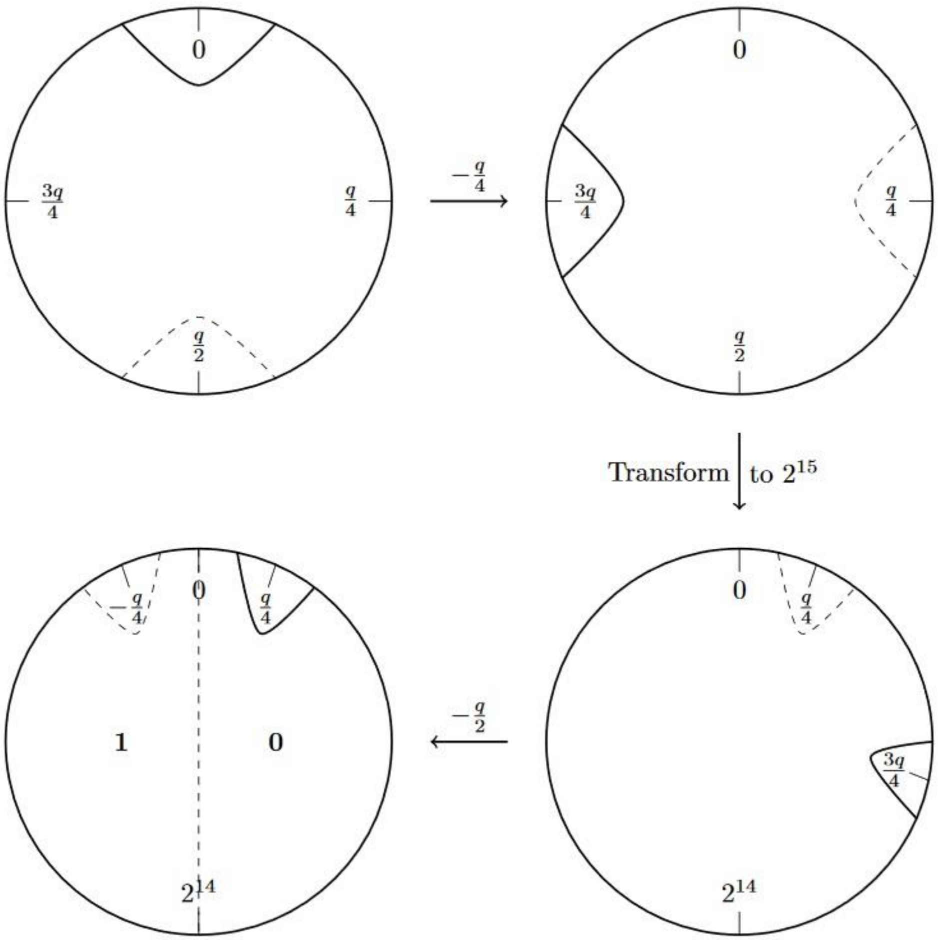
Masked decoding

Input: Coefficient $\in [0, q-1]$

Output: Decoded bit

Idea:

- Shift distribution of coefficients
- Apply arithmetic-to-Boolean conversion
- Extract sign bit



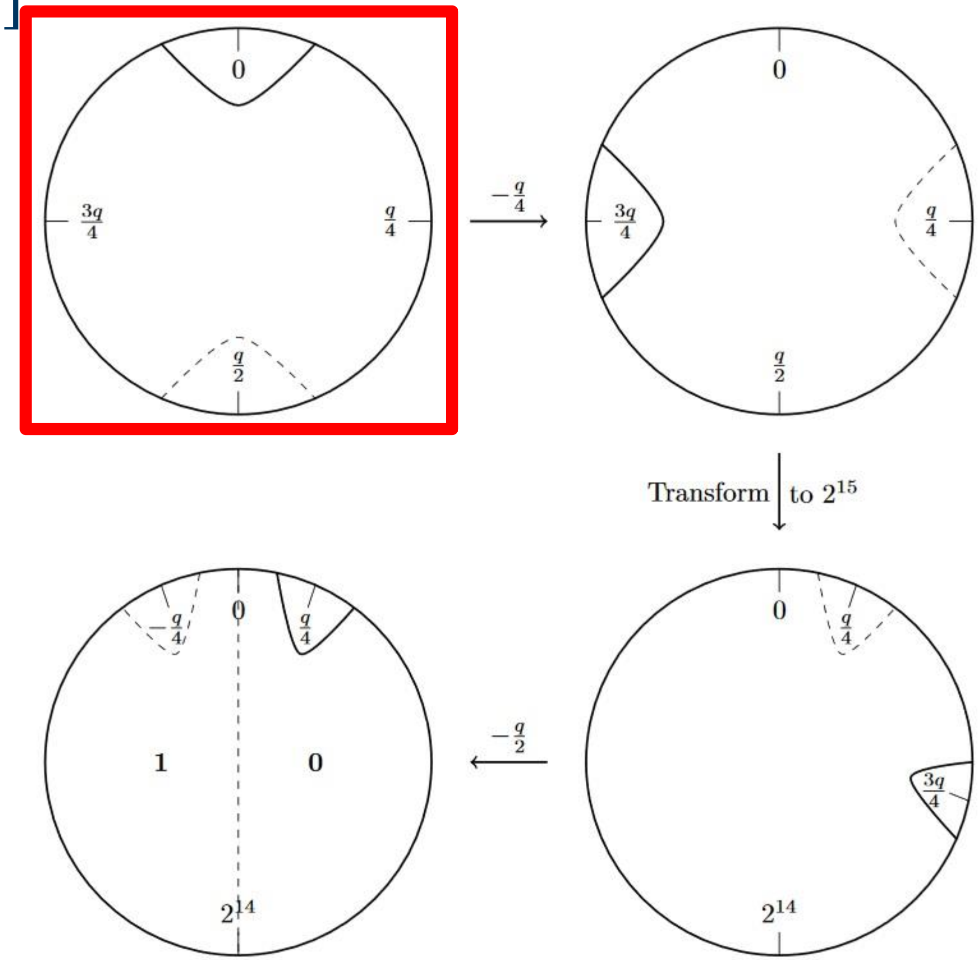
Masked decoding

Input: Coefficient $\in [0, q-1]$

Output: Decoded bit

Idea:

- Shift distribution of coefficients
- Apply arithmetic-to-Boolean conversion
- Extract sign bit



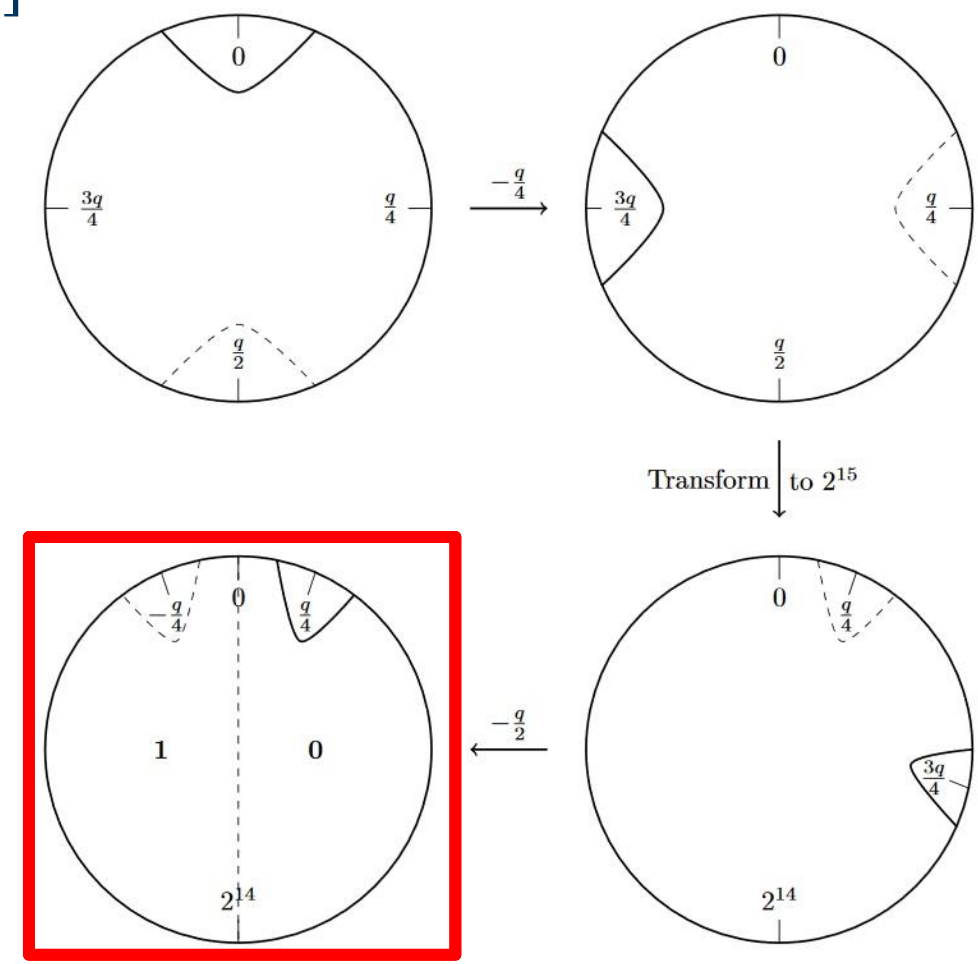
Masked decoding

Input: Coefficient $\in [0, q-1]$

Output: Decoded bit

Idea:

- Shift distribution of coefficients
- Apply arithmetic-to-Boolean conversion
- Extract sign bit



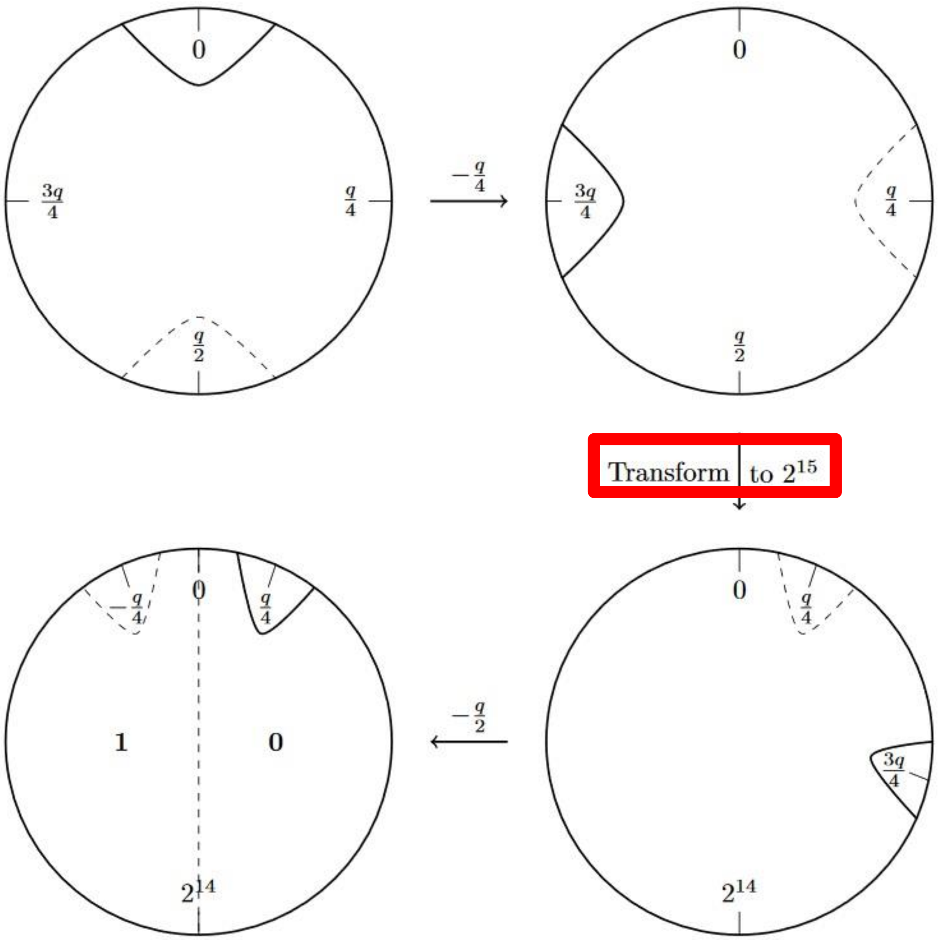
Masked decoding

Input: Coefficient $\in [0, q-1]$

Output: Decoded bit

Idea:

- Shift distribution of coefficients
- Apply arithmetic-to-Boolean conversion
- Extract sign bit



Binomial Sampler

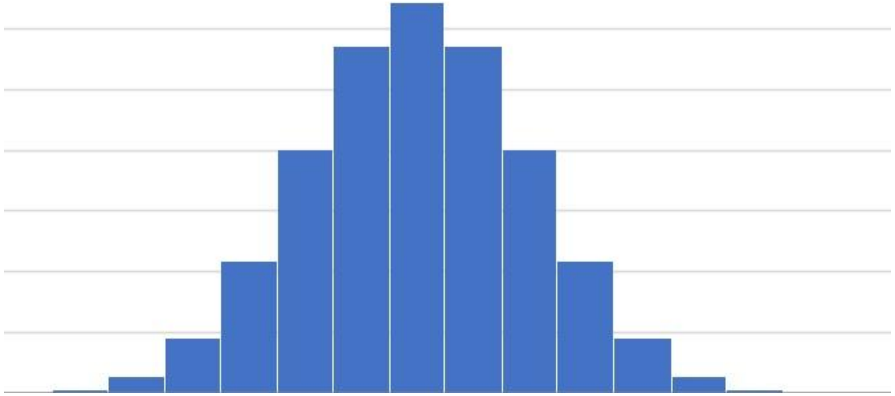
Masked sampler

- **Input**: Boolean shares; **Output**: Arithmetic shares

- Count Hamming weight as

$$\sum_{i=0}^{7}(bit'(i) \oplus bit''(i))$$
$$= \sum_{i=0}^{7} bit'(i) + bit''(i) - 2bit'(i)bit''(i)$$

- Compute $bit'(i) \cdot bit''(i)$ by splitting into subshares

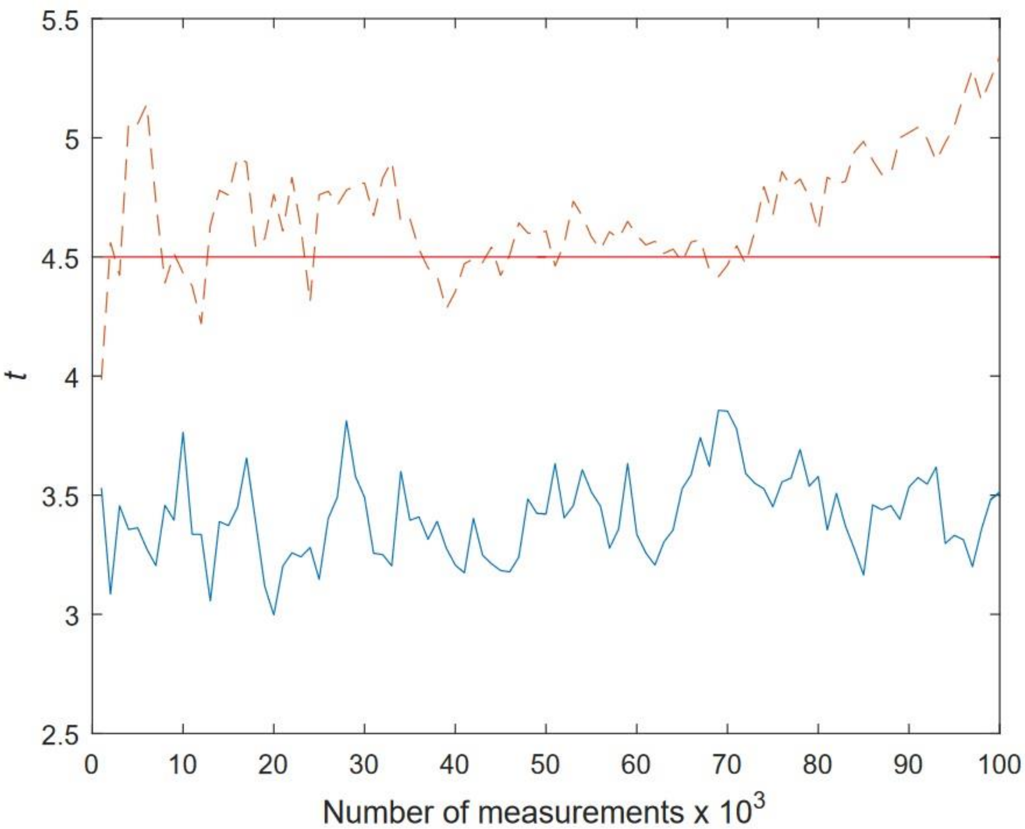


Results

Side-Channel Evaluation

T-test evaluation of the decoding (example)

- *Blue*: first-order evaluation
- *Dashed red*: second-order evaluation



Cortex-M4 Performance

- Dimension n = 1024
- Modulus q = 12289
- Standard deviation ς = 2

Operation	Cycle Counts Unmasked	Cycle Counts Masked
Key Generation	2,669,559	-
CCA2-secured Encryption	4,176,684	-
CCA2-secured Decryption	4,416,918	25,334,493
CPA-RLWE Encryption	3,910,871	19,315,432
CPA-RLWE Decryption	163,887	550,038
Shake-128	87,738	201,997
NTT	83,906	-
INTT	104,010	-
Uniform Sampling (TRNG)	60,014	-
SampleNoisePoly (PRNG)	1,142,448	6,031,463
PRNG (64 bytes)	88,778	202,454

Cortex-M4 Performance

RUB

- Dimension n = 1024
- Modulus q = 12289
- Standard deviation ς = 2

Operation	Cycle Counts	
	Unmasked	Masked
Key Generation	2,669,559	-
CCA2-secured Encryption	4,176,684	-
CCA2-secured Decryption	4,416,918	25,334,493
CPA-RLWE Encryption	3,910,871	19,315,432
CPA-RLWE Decryption	163,887	550,038
Shake-128	87,738	201,997
NTT	83,906	-
INTT	104,010	-
Uniform Sampling (TRNG)	60,014	-
SampleNoisePoly (PRNG)	1,142,448	6,031,463
PRNG (64 bytes)	88,778	202,454

Cortex-M4 Performance

- Dimension n = 1024
- Modulus q = 12289
- Standard deviation ς = 2

Operation	Cycle Counts Unmasked	Cycle Counts Masked
Key Generation	2,669,559	-
CCA2-secured Encryption	4,176,684	-
CCA2-secured Decryption	4,416,918	25,334,493
CPA-RLWE Encryption	3,910,871	19,315,432
CPA-RLWE Decryption	163,887	550,038
Shake-128	87,738	201,997
NTT	83,906	-
INTT	104,010	-
Uniform Sampling (TRNG)	60,014	-
SampleNoisePoly (PRNG)	1,142,448	6,031,463
PRNG (64 bytes)	88,778	202,454

Cortex-M4 Performance

- Dimension n = 1024
- Modulus q = 12289
- Standard deviation ς = 2

Operation	Cycle Counts Unmasked	Cycle Counts Masked
Key Generation	2,669,559	-
CCA2-secured Encryption	4,176,684	-
CCA2-secured Decryption	4,416,918	25,334,493
CPA-RLWE Encryption	3,910,871	19,315,432
CPA-RLWE Decryption	163,887	550,038
Shake-128	87,738	201,997
NTT	83,906	-
INTT	104,010	-
Uniform Sampling (TRNG)	60,014	-
SampleNoisePoly (PRNG)	1,142,448	6,031,463
PRNG (64 bytes)	88,778	202,454

Conclusion

- First masking of a Ring-LWE-based scheme that covers CCA2-security with first-order proof
- New masked encoder & decoder
- New masked sampler
- *Future work*: Higher-order masking

RUB

Thank You For Your Attention!

hgi Horst Görtz Institut für IT-Sicherheit