Side-channel Security of Public-Key Cryptography

Cryptography on Hardware Platform

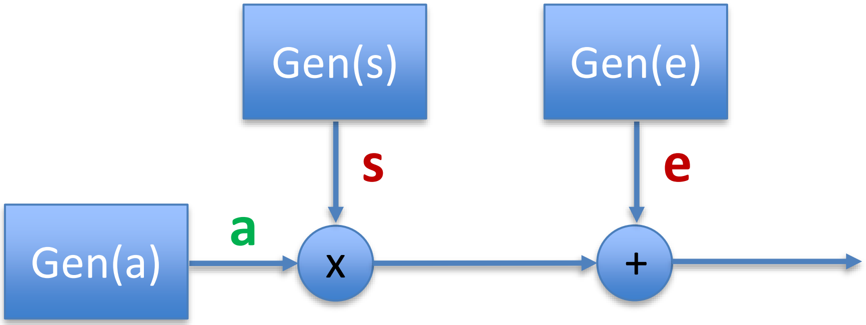
Sujoy Sinha Roy

sujoy.sinharoy@iaik.tugraz.at

Recap of Ring-LWE Public-key Encryption

Ring LWE Public-Key Encryption (PKE)

- **Key Generation:**
  - **Output:** public key (pk), secret key (sk)



$b = a.s + e$ (Ring LWE sample)

Arithmetic operations are performed in a polynomial ring R_q

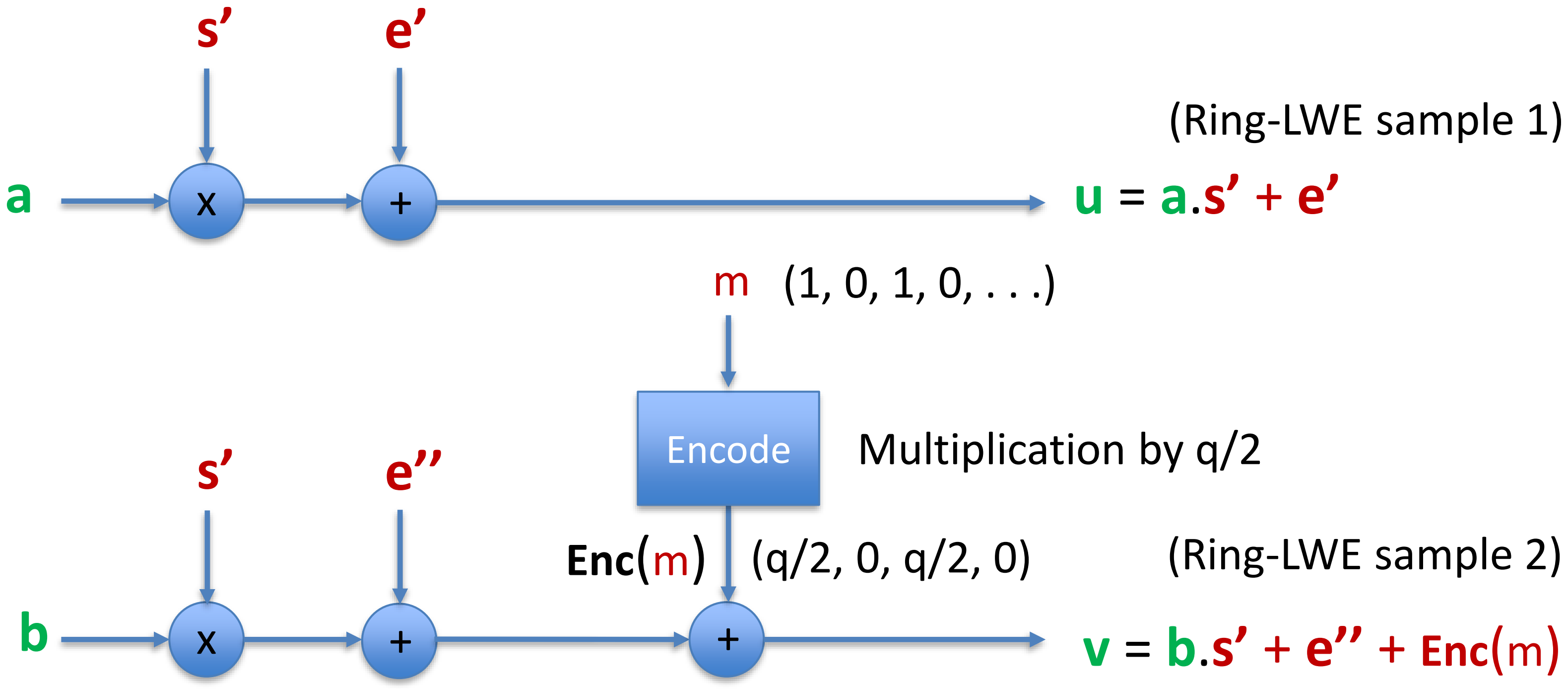
Public Key (pk): (a,b)

Secret Key (sk): (s)

V. Lyubashevsky, C. Peikert, and O. Regev. "On Ideal Lattices and Learning with Errors Over Rings". IACR ePrint 2012/230.

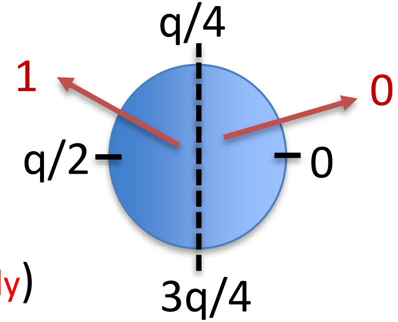
Ring LWE Public-Key Encryption (PKE)

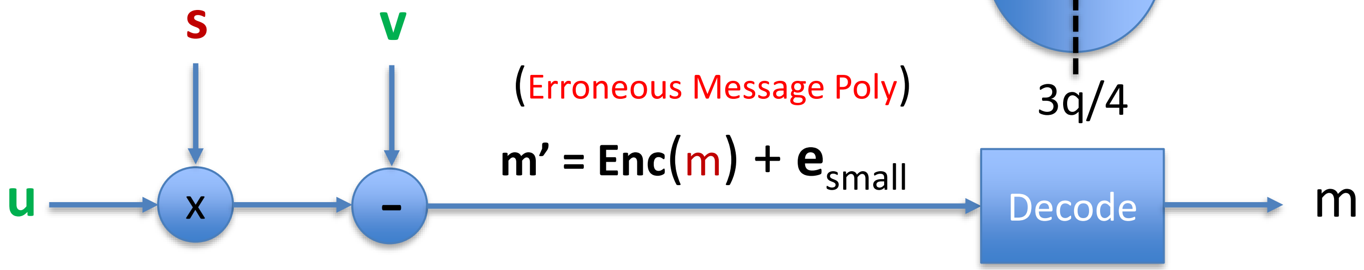
- **Encryption:**
  - **Input: pk** = (a,b), message m
  - **Output: ct** = (u,v)



Ring LWE Public-Key Encryption (PKE)

- **Decryption:**
  - **Input:** ct = (u, v), sk = s
  - **Output:** m after decoding





$$\mathbf{v} - \mathbf{u}.\mathbf{s} = \mathbf{m'} = \mathbf{Enc}(m) + (\mathbf{e}.\mathbf{s'} + \mathbf{e''} + \mathbf{e'}.\mathbf{s})$$
$$= \mathbf{Enc}(m) + \mathbf{e}_{\text{small}}$$

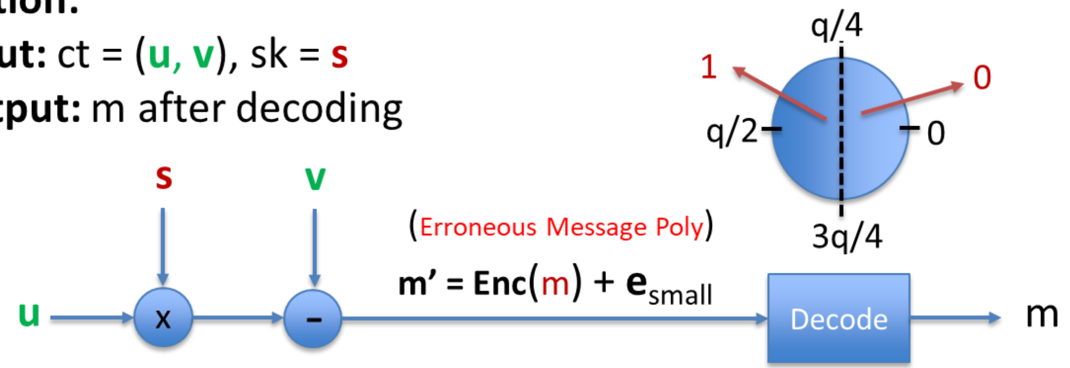
Select most significant bit of each coefficient as the message bits

Ring LWE Public-Key Encryption (PKE)

Describe a side-channel attack given the following:

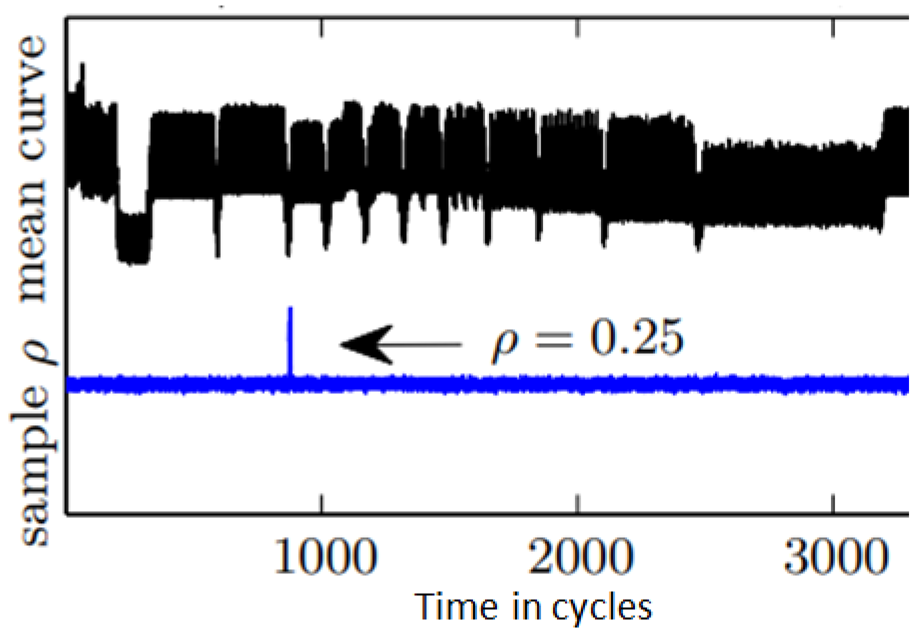
- A decryption device has a long-term and constant secret **s**.
- Attacker has a copy of the same device but doesn't know **s**.
- Attacker can do decryption queries

- **Decryption:**
  - **Input:** ct = (**u**, **v**), sk = **s**
  - **Output:** m after decoding



$$\mathbf{v} - \mathbf{u}.\mathbf{s} = \mathbf{m'} = \mathbf{Enc}(m) + (\mathbf{e}.\mathbf{s'} + \mathbf{e''} + \mathbf{e'}.\mathbf{s})$$
$$= \mathbf{Enc}(m) + \mathbf{e}_{small}$$

Correlation power analysis for u[0]*s[0]



Power trace from target device

Pearson's correlation

1. Attacker uses her identical device to obtain power traces for all possible s[0].
2. Attacker computes correlation between the power traces obtained from the two devices.
3. For the correct guess of s[0], the correlation will be noticeably high.

What causes this guessing possible?

How to make this guessing harder?

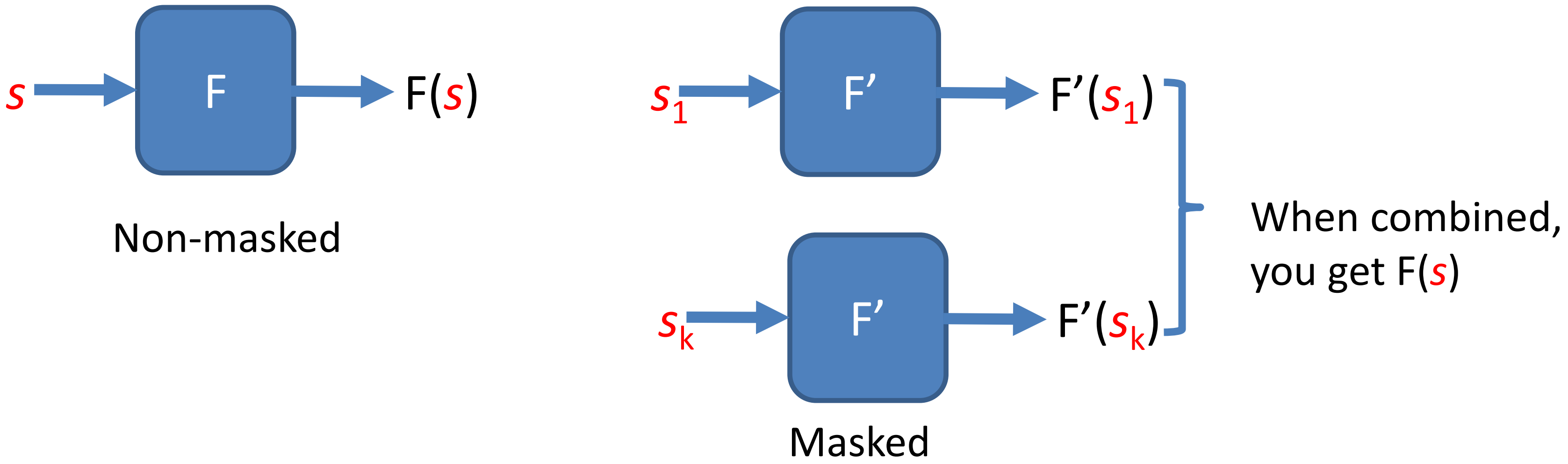
What is masking countermeasure?

- Countermeasure against differential power analysis (DPA)

- Randomizes computation by splitting secret data into random shares

$$s = s_1 + s_2 + s_3 + \ldots + s_k$$

- No information about s can be obtained by observing a proper subset



Arithmetic and Boolean shares

- Two common ways of splitting a secret into shares

- Boolean shares: secret bit is split in GF(2)

$$s = s_1 \oplus s_2 \oplus s_3 \oplus \ldots \oplus s_k \quad \text{mod } 2$$

… applicable to words (vector of bits)

- Arithmetic shares: secret is split in GF(p) where p>2

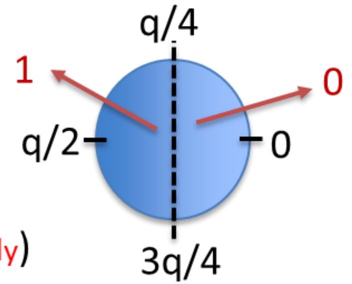
$$s = s_1 + s_2 + s_3 + \ldots + s_k \quad \text{mod } p$$

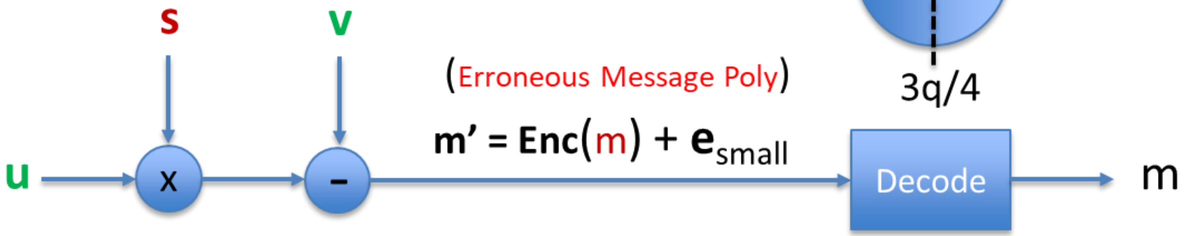
E.g., 7 = 8 + 10 mod 11

- Some cryptographic algorithms require working with both types

Design a masking scheme for this the decryption?

- **Decryption:**
  - **Input:** ct = (u, v), sk = s
  - **Output:** m after decoding

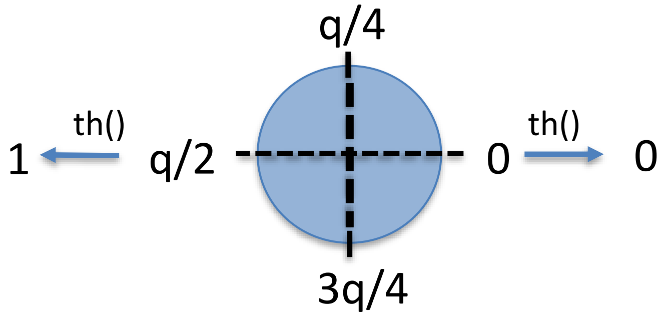


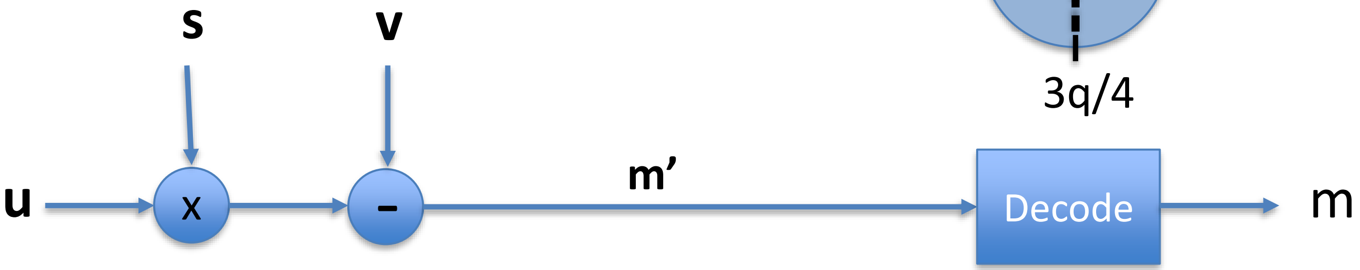


$$v - u.s = \mathbf{m'} = \mathbf{Enc}(m) + (\mathbf{e.s'} + \mathbf{e''} + \mathbf{e'.s})$$
$$= \mathbf{Enc}(m) + \mathbf{e}_{\text{small}}$$

Ring LWE Decryption

- **Decryption:**
  - **Input:** ct = (**u, v**), sk = **s**
  - **Output:** m





$$\mathbf{m'} = \mathbf{v} - \mathbf{u.s} = \mathbf{Enc}(m) + \mathbf{e}_{\text{small}}$$

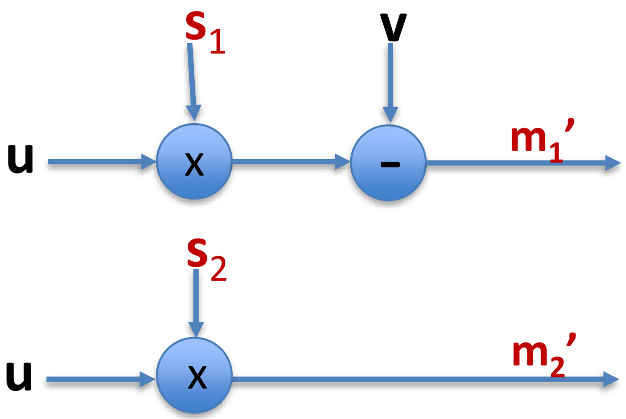
Note: ct = (**u, v**) is controlled by attacker

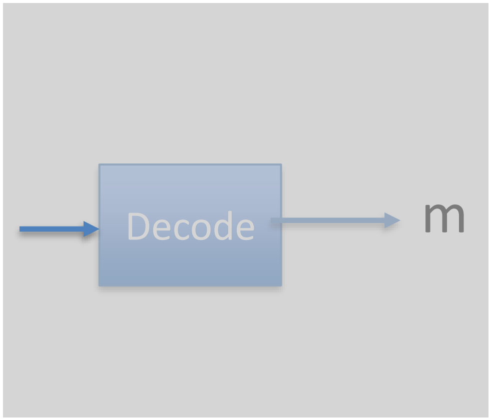
Masking Idea: Split **s** into random shares and randomize computation

1^{st} Order Masking for Ring-LWE Decryption

- Step1: Split **s** into two random arithmetic shares

$$s = s_1 - s_2 \bmod q$$



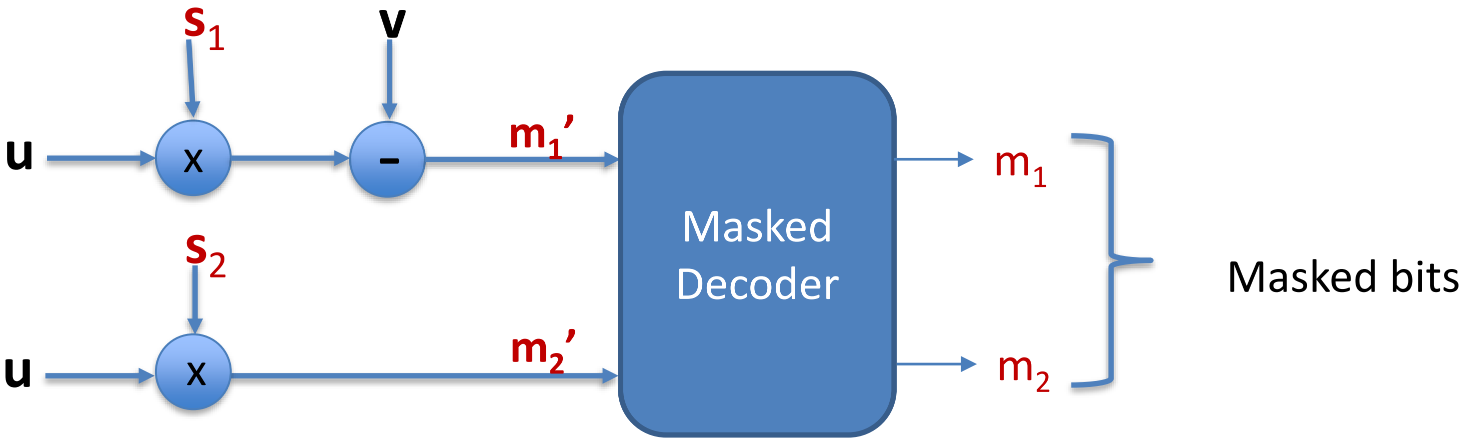


$m_1' = v - u.s_1$
$m_2' = u.s_2$

How to compute decoding on two shares?

Easy to check **$m_1' + m_2' = v - u.s = m'$**

Masked Decoding



What we want:

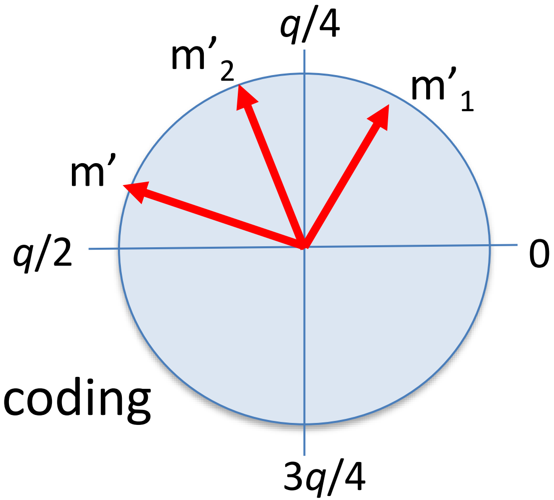
1. Compute mask-message pair (m_1, m_2) s.t. $m = m_1 + m_2$ mod 2
2. No combination of the two input shares $\mathbf{m_1'}$ and $\mathbf{m_2'}$

There are several approaches to design masked decoders

Masked Decoder of [RRVV15]

- Observation: Only a few most significant bits of the shares are helpful to perform threshold decoding

- Example:
  If $0 < m'_1 < q/4$ and $q/4 < m'_2 < q/2$
  then $q/4 < m' < 3q/2$
  → $th(m') = 1$



- This observation is used to simplify masked decoding

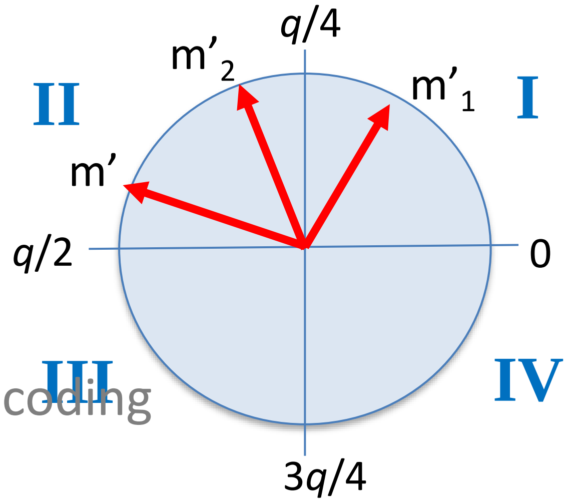
[RRVV15] O. Reparaz, S. Sinha. Roy, F. Vercauteren, I. Verbauwhede. "A Masked Ring-LWE Implementation". CHES 2015.

Masked Decoder of [RRVV15]

- Observation: Only a few most significant bits of the shares are helpful to perform threshold decoding

- Example:
  If $0 < m'_1 < q/4$ and $q/4 < m'_2 < q/2$
  then $q/4 < m' < 3q/2$
  $\rightarrow$ th(m') = 1

- This observation is used to simplify masked decoding

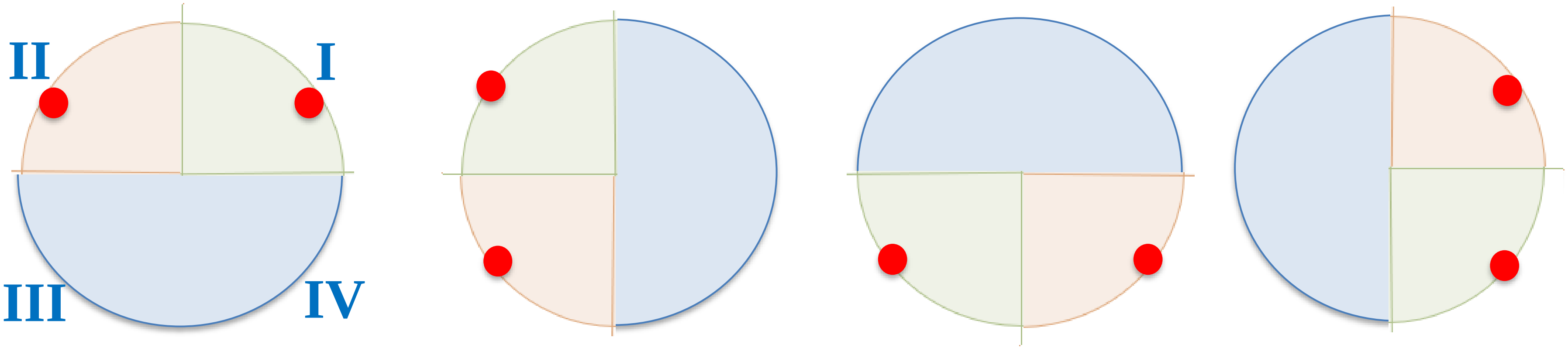


quad() function is used to output quadrant of a share.

[RRVV15] O. Reparaz, S. Sinha Roy, F. Vercauteren, I. Verbauwhede. "A Masked Ring-LWE Implementation". CHES 2015.

Masked Decoder of [RRVV15]

Quad-based decoding **works** if two shares are in adjacent quadrants.



Otherwise, this approach fails.

Solution proposed in [RRVV15]: Refresh shares and try again.

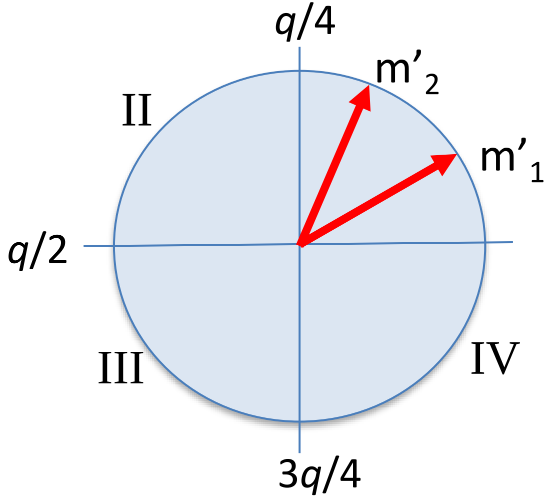
1. Take a constant δ_i from a table
2. $m'_1 := m'_1 - \delta_i$
3. $m'_2 := m'_2 + \delta_i$
4. Check if they are in adjacent quadrants

Iterated a fixed number of times

Masked Decoder of [RRVV15]: Resolving ambiguity

Shares are refreshed and then rule checking is performed

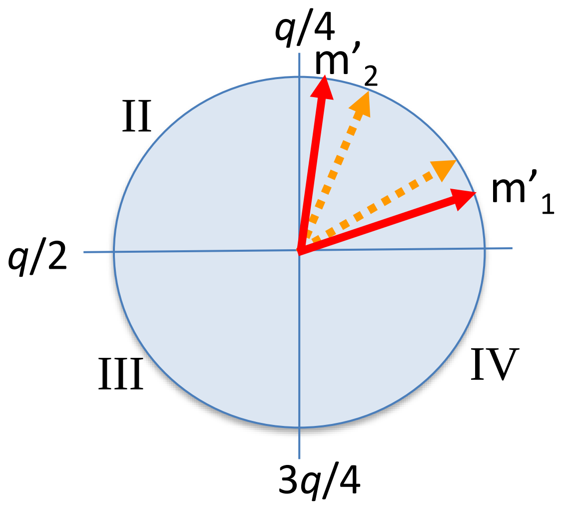
- Take a constant δ_i from a table
- $m'_1 := m'_1 + \delta_i$
- $m'_2 := m'_2 - \delta_i$
- Check if they are in adjacent quadrants



Masked Decoder of [RRVV15]: Resolving ambiguity

Shares are refreshed and then rule checking is performed

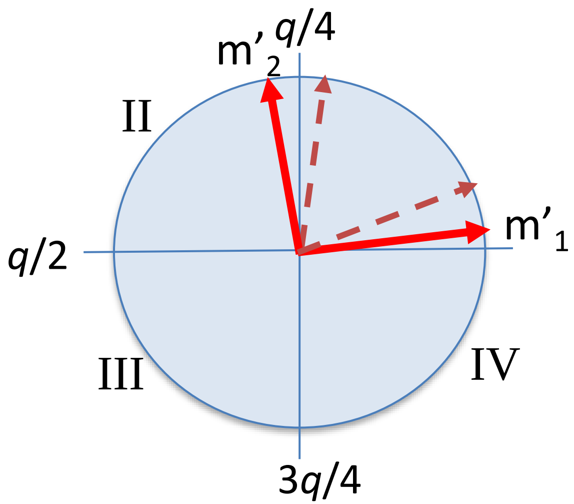
- Take a constant δ_i from a table
- $m'_1 := m'_1 + \delta_i$
- $m'_2 := m'_2 - \delta_i$
- Check if they are in adjacent quadrants



Masked Decoder of [RRVV15]: Resolving ambiguity

Shares are refreshed and then rule checking is performed

- Take a constant δ_i from a table
- $m'_1 := m'_1 + \delta_i$
- $m'_2 := m'_2 - \delta_i$
- Check if they are in adjacent quadrants



As soon as a valid decoding rule is hit, the quadrants are recorded.

Estimate the performance overhead of masked Ring-LWE decryption?