Higher-Order Masking of Lattice-Based Signatures

Sonia BELAÏD

CryptoExperts

Outline

Introduction

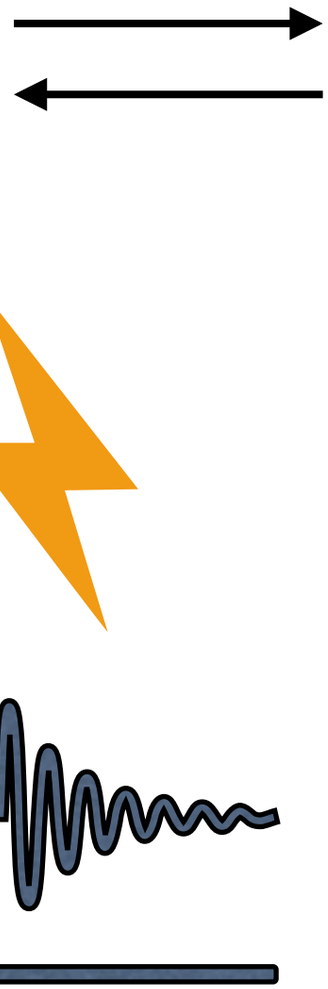
- Side-Channel Attacks
- Side-Channel Attacks Against Lattice-Based Signatures
- Masking

Masking Lattice-Based Signatures

- An Example: Masking GLP
- Other Signature Schemes

Introduction

Side-Channel Attacks

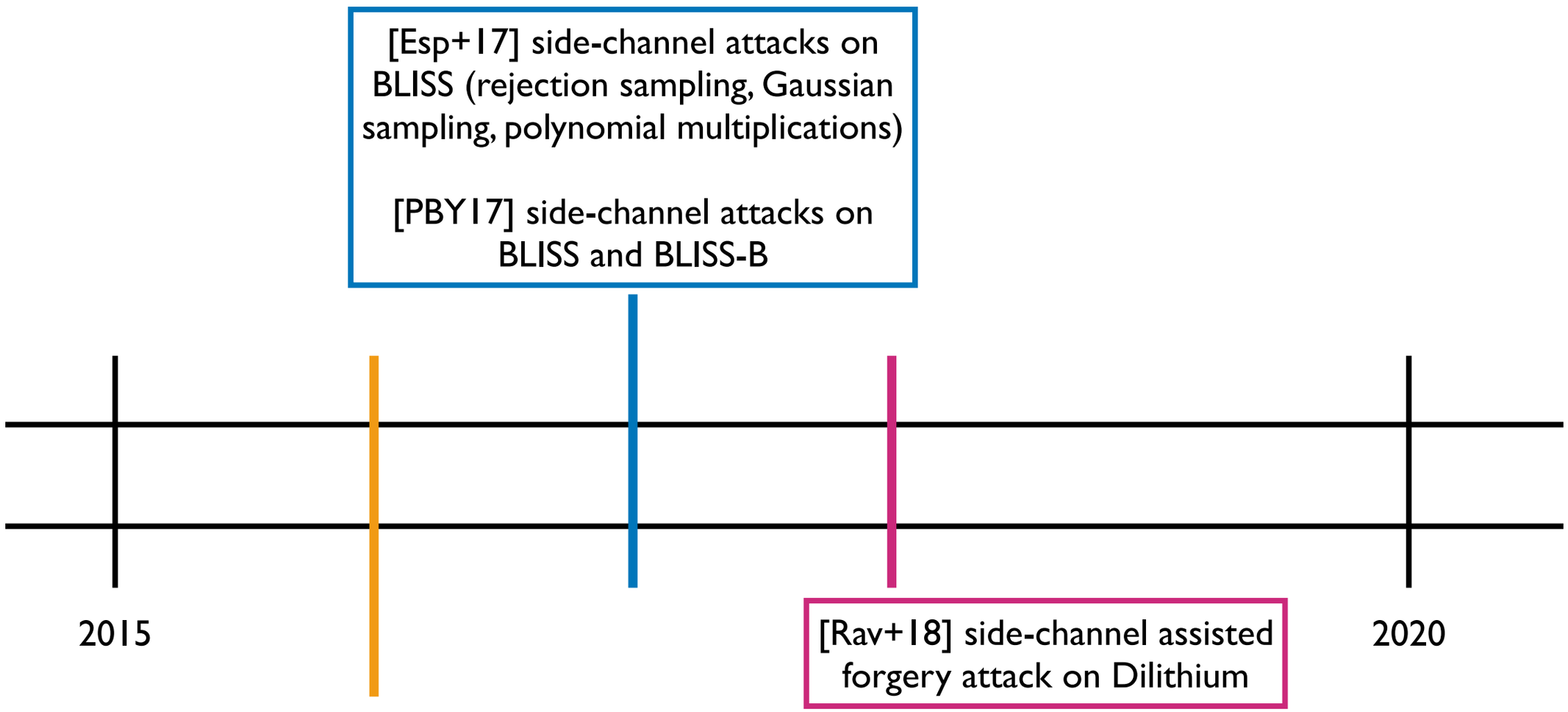


inputs / outputs

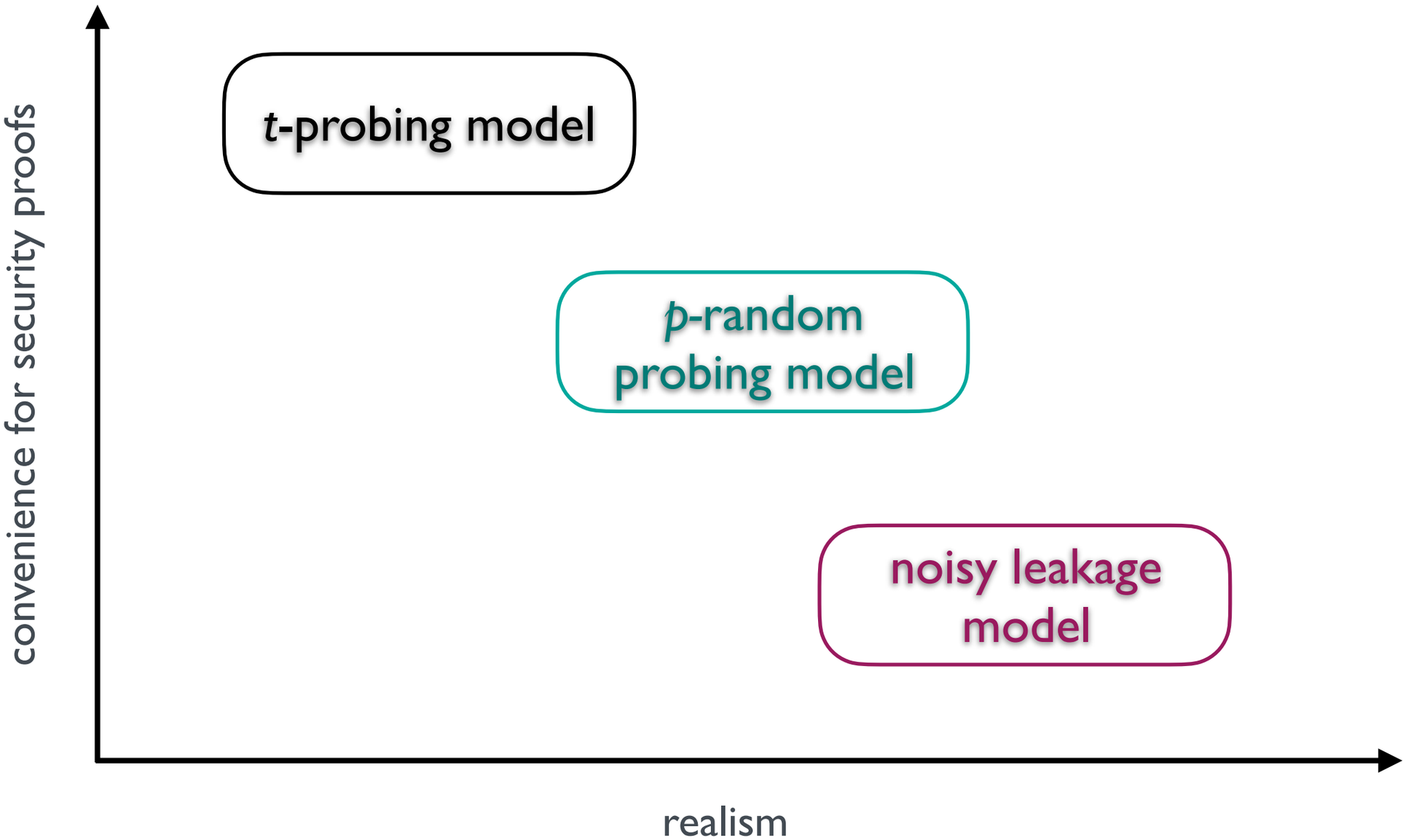
Power consumption

$$\approx \left|\left|_i \left(f(v_i) + \eta\right)\right.\right.$$

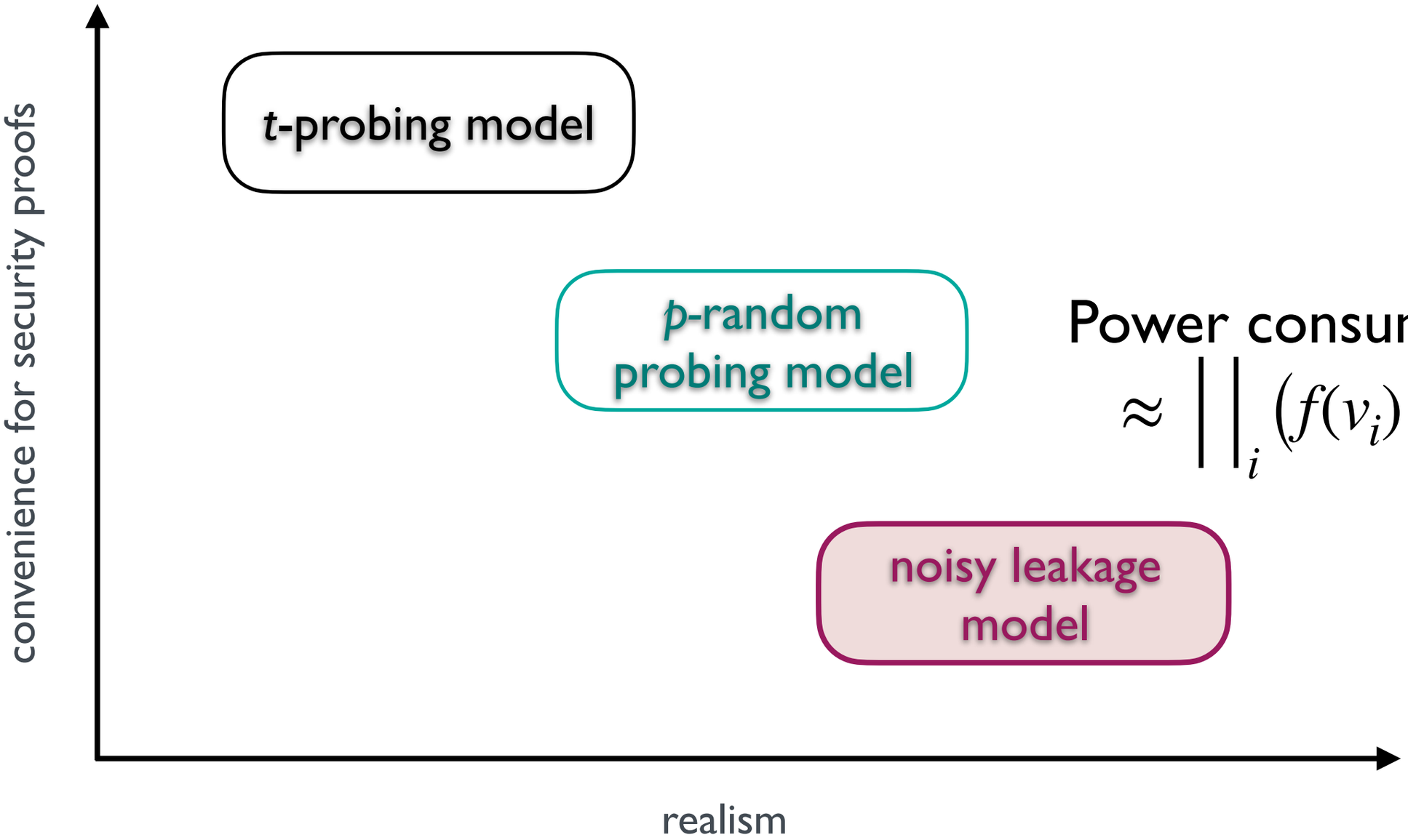
Side-Channel Attacks against Lattice-Based Signatures



Leakage Models



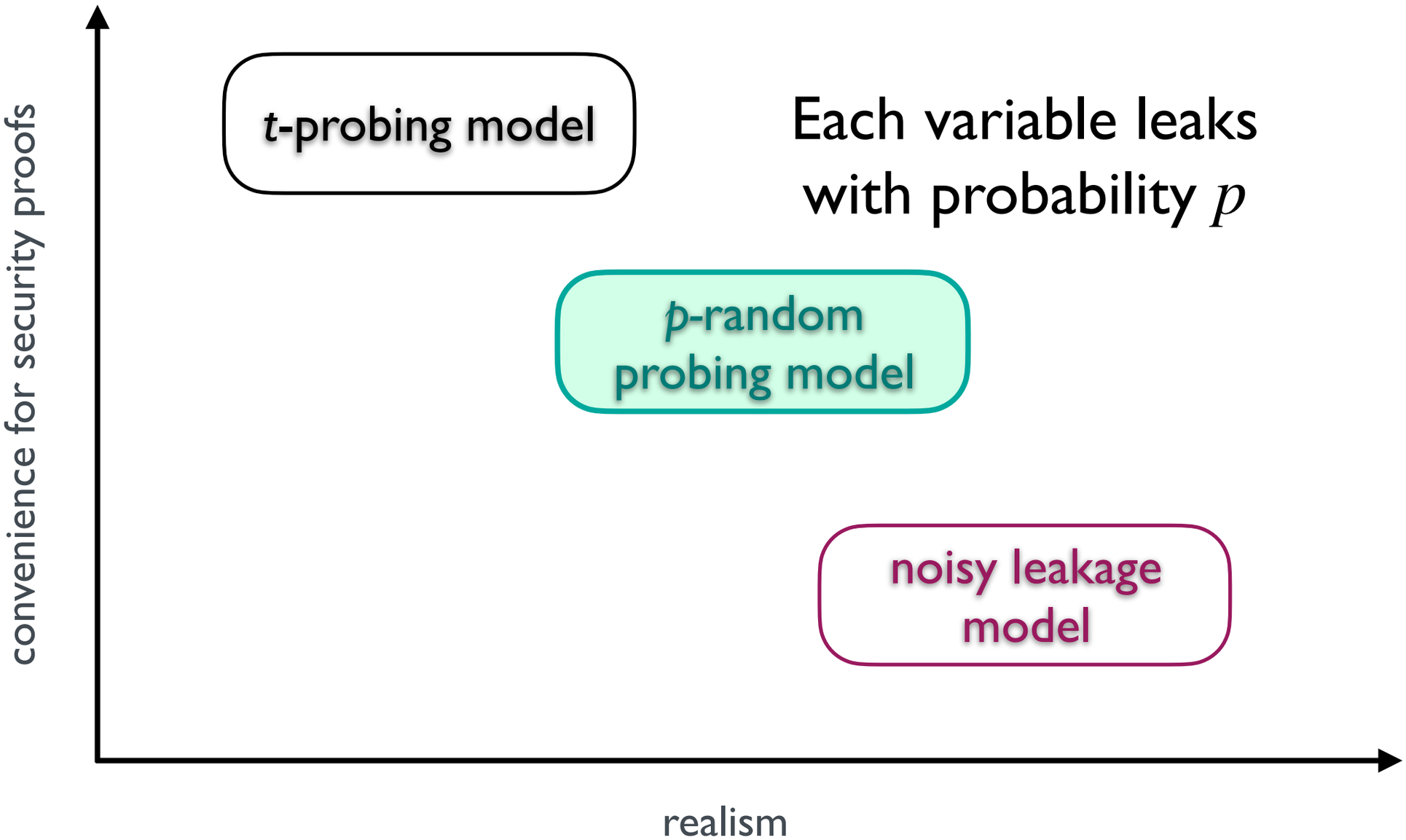
Leakage Models



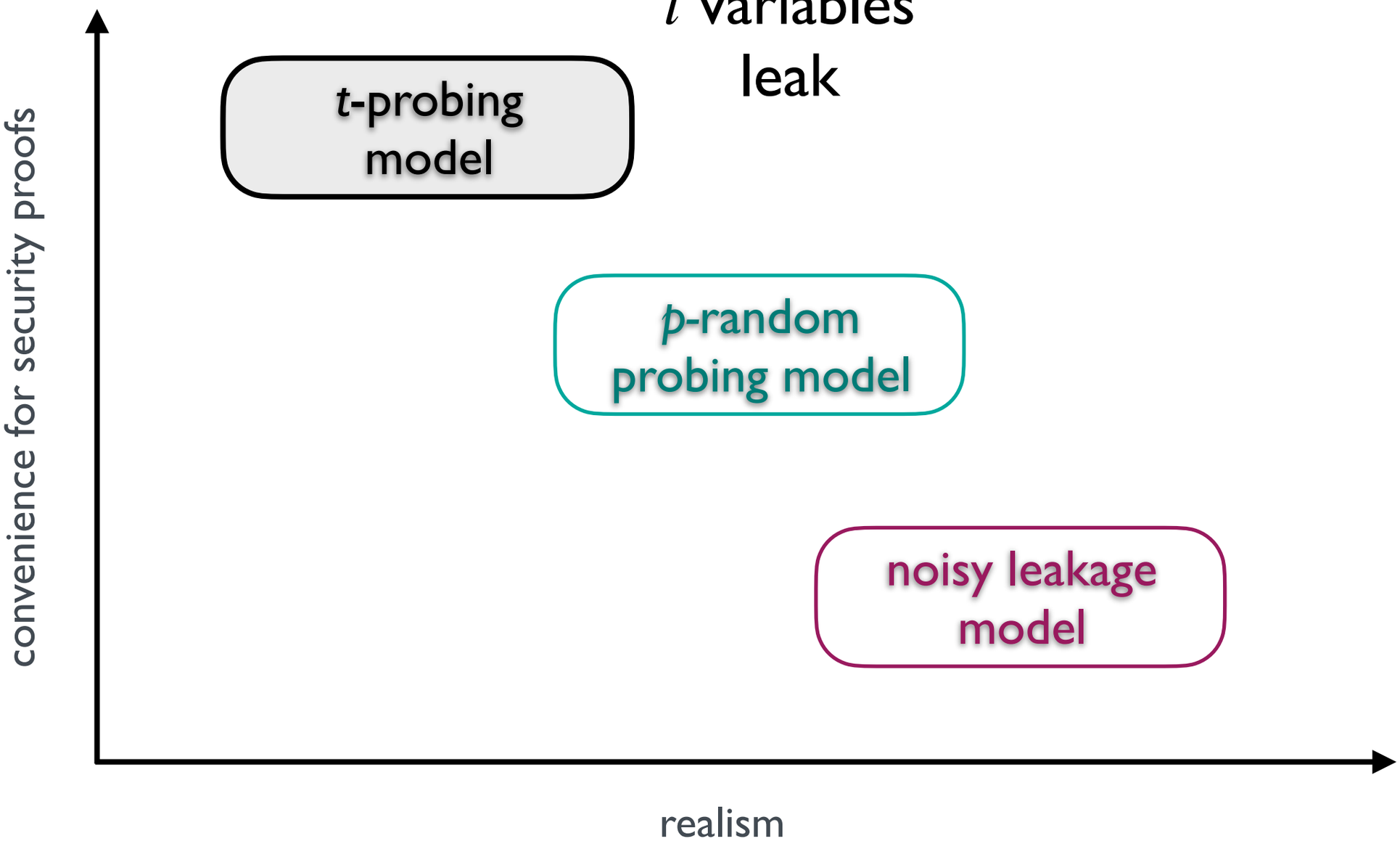
Power consumption

$$\approx \left|\left|_i \left(f(v_i) + \eta\right)\right.\right.$$

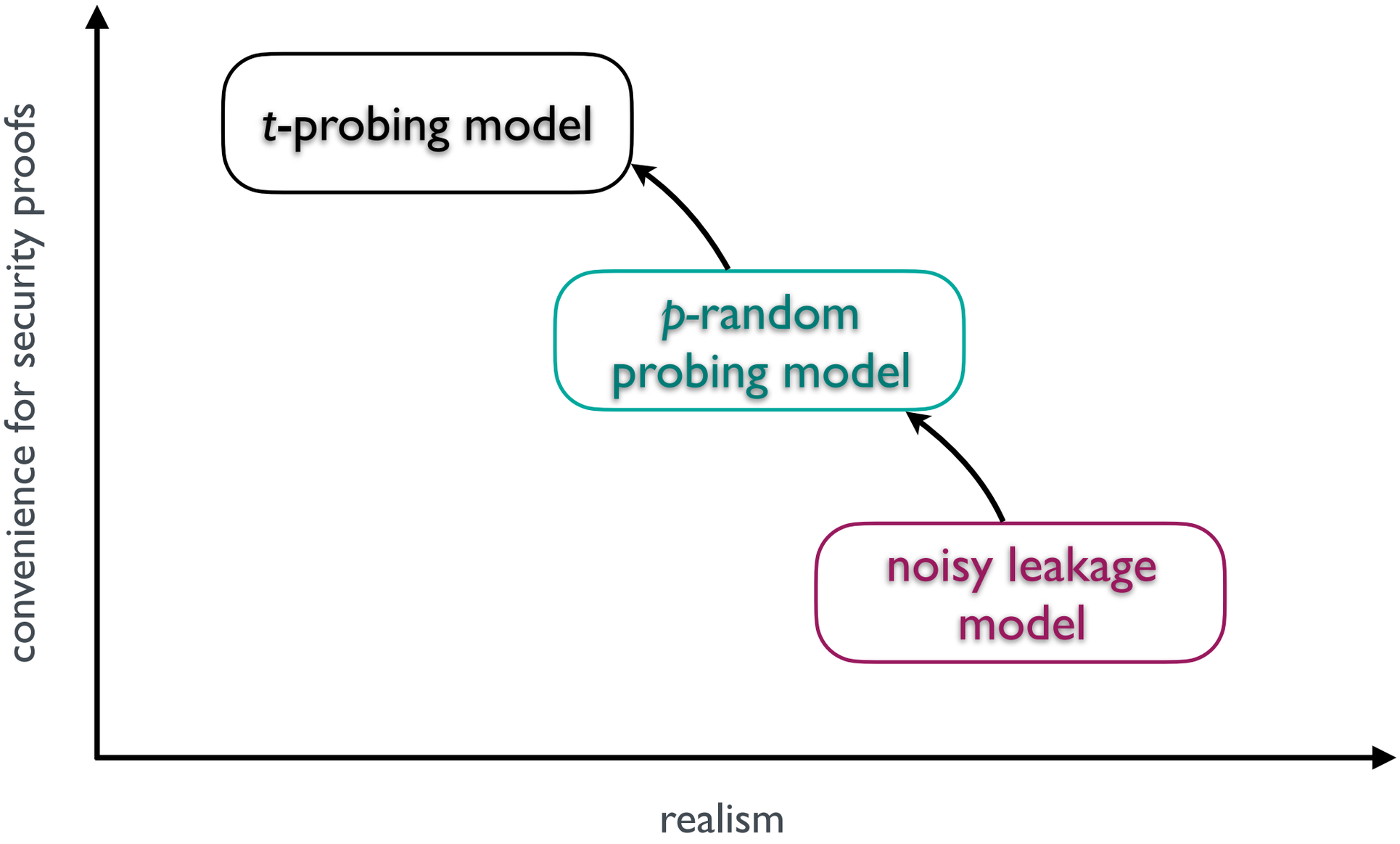
Leakage Models



Leakage Models



Leakage Models



[EC:DDF14] *Unifying Leakage Models: From Probing Attacks to Noisy Leakage*. Alexandre Duc, Stefan Dziembowski, Sebastian Faust. Eurocrypt 2014

Probing Security [ISW03]

Attacker model: adversary gets the exact values of t intermediate variables

Security: any set of *at most* t intermediate variables must be independent of the secrets

Masking

Sensitive variable v

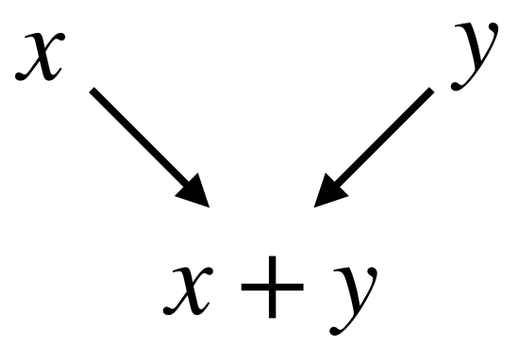
$$v_1, \ldots, v_{n-1} \leftarrow \$ \text{ } \textit{(uniform distribution)}$$

$$v_n \leftarrow v + v_1 + \ldots + v_{n-1}$$

Each strict subset of $(v_i)_{1 \leq i \leq n}$ is independent from v

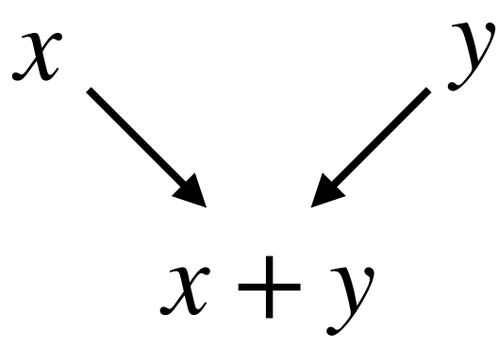
Masking

Additions:



Masking

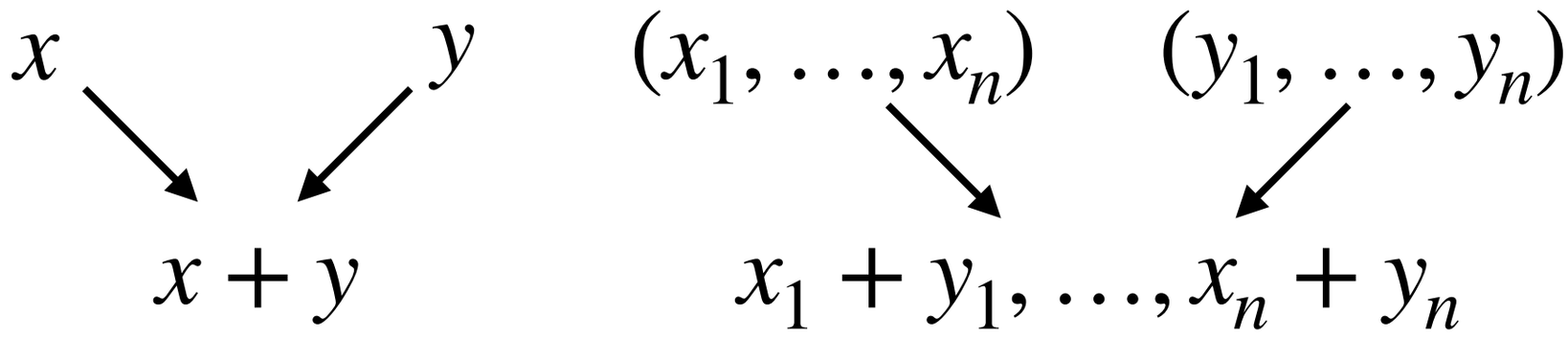
Additions:



$(x_1, \ldots, x_n)$ $(y_1, \ldots, y_n)$

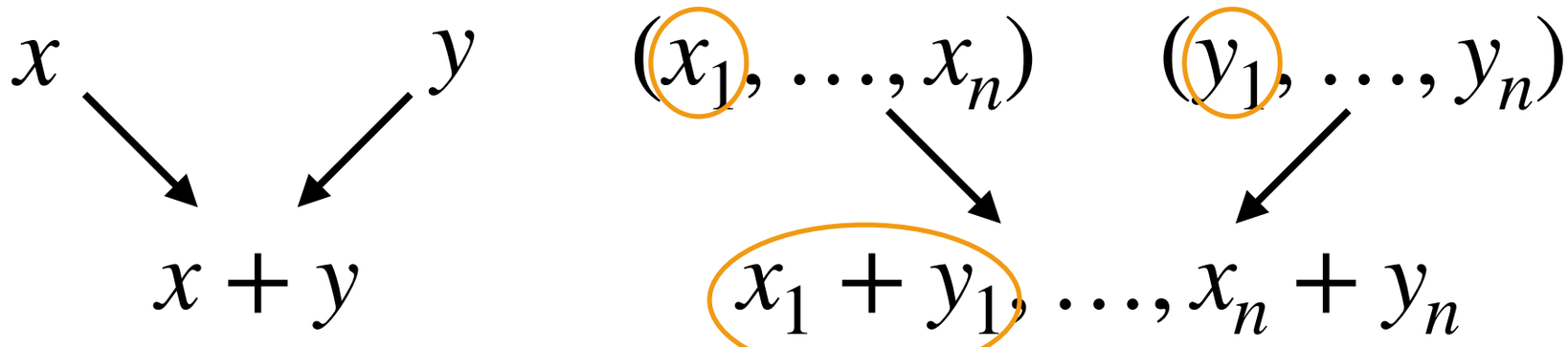
Masking

Additions:



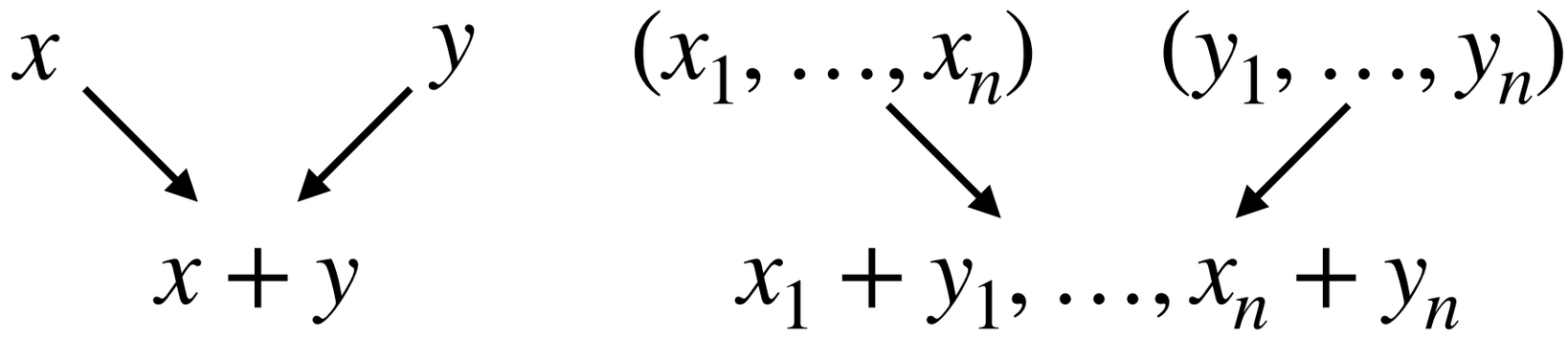
Masking

Additions:



Masking

Additions:



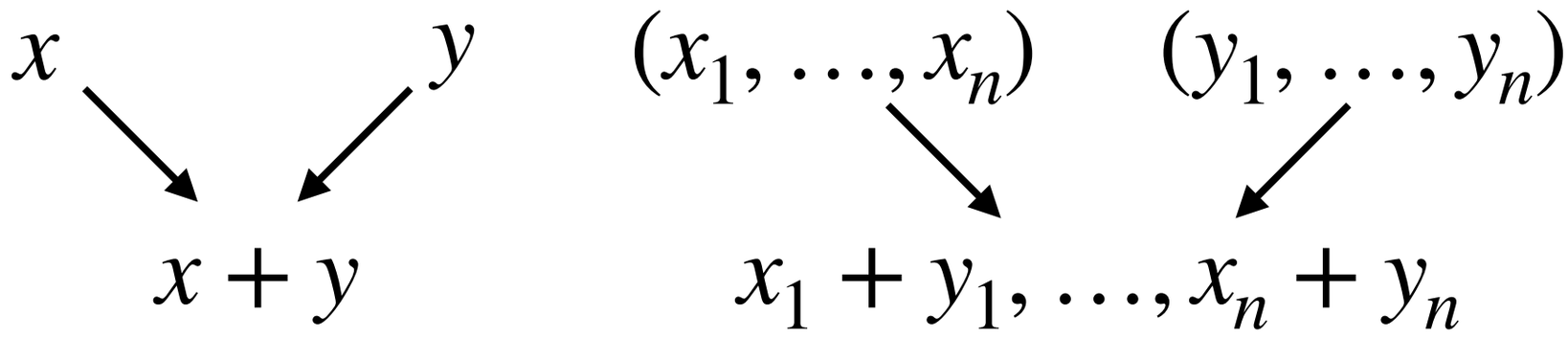
Multiplications:

$$x \cdot y \rightarrow ?$$

Need for extra randomness to mix shares without introducing a bias

Masking

Additions:



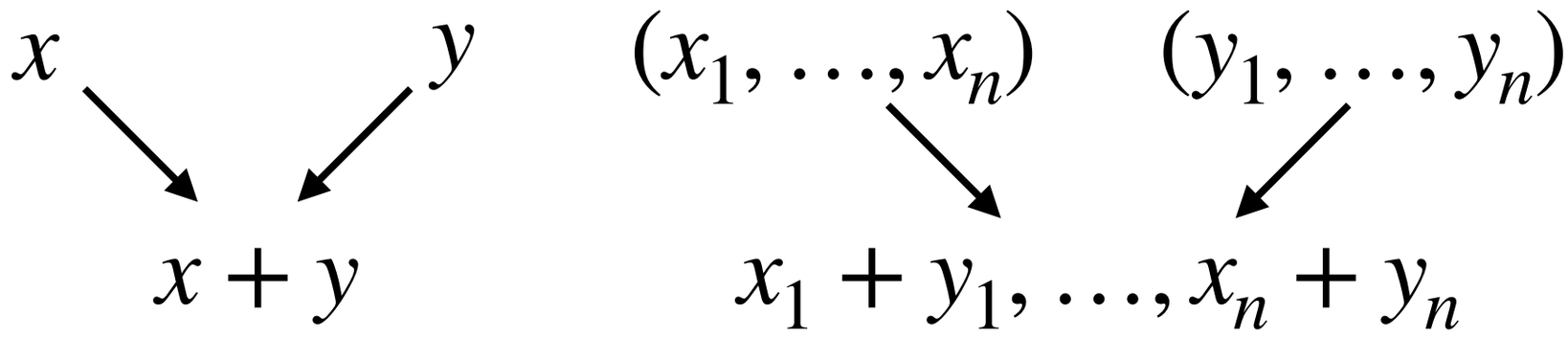
Multiplications:

$$x \cdot y \rightarrow ?$$

Small example for $n = 2$

Masking

Additions:



Multiplications:

$$x \cdot y \rightarrow ?$$

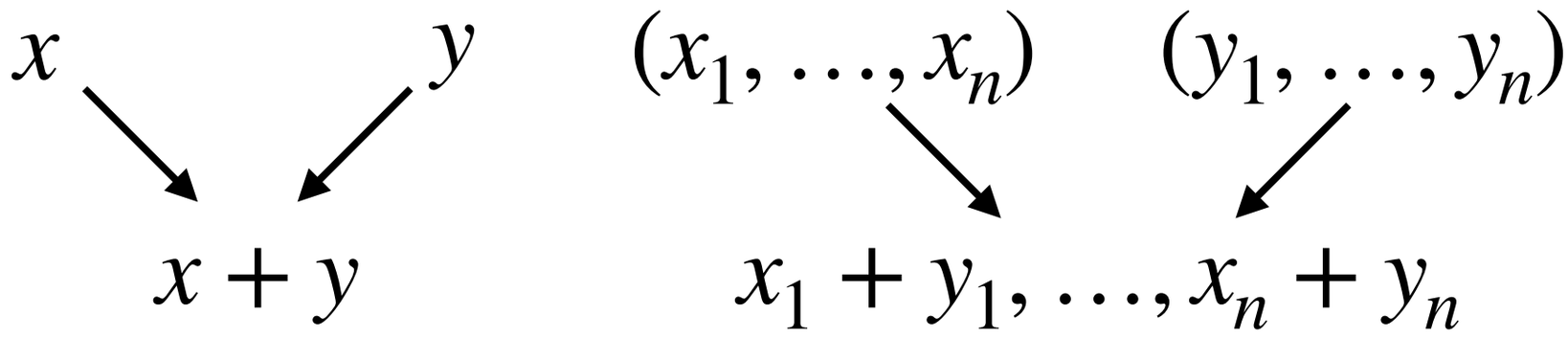
Small example for $n = 2$

$$x = x_0 + x_1$$

$$y = y_0 + y_1$$

Masking

Additions:



Multiplications:

$$x \cdot y \rightarrow ?$$

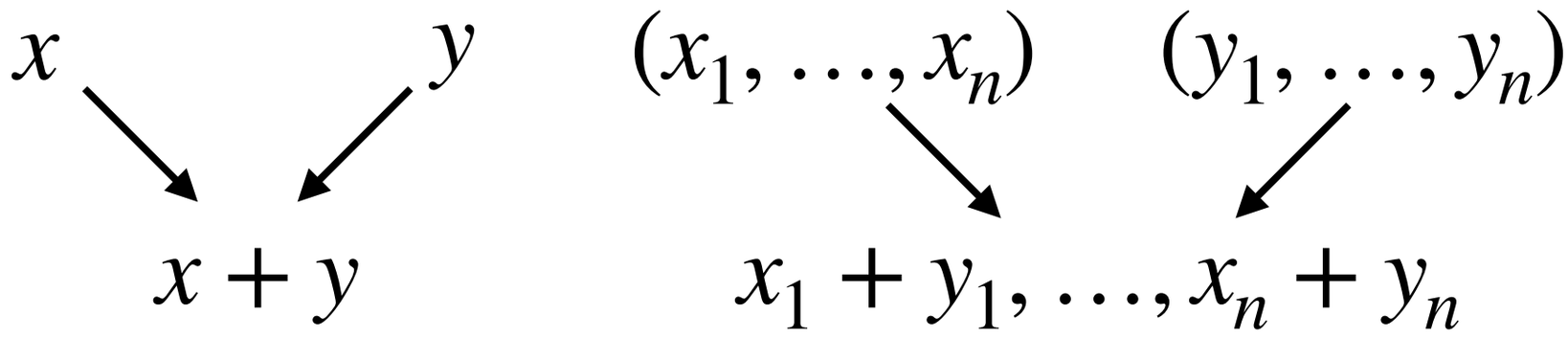
Small example for $n = 2$

$$x = x_0 + x_1 \qquad z_0 \leftarrow x_0 y_0 + x_0 y_1$$

$$y = y_0 + y_1 \qquad z_1 \leftarrow x_1 y_1 + x_1 y_0$$

Masking

Additions:



Multiplications:

$$x \cdot y \rightarrow ?$$

Small example for $n = 2$

$$x = x_0 + x_1 \qquad z_0 \leftarrow x_0 y_0 + r + x_0 y_1$$

$$y = y_0 + y_1 \qquad z_1 \leftarrow x_1 y_1 + r + x_1 y_0$$

Outline

Introduction

- Side-Channel Attacks
- Side-Channel Attacks Against Lattice-Based Signatures
- Masking

Masking Lattice-Based Signatures

- An Example: Masking GLP
- Other Signature Schemes

Masking Lattice-Based Signatures

Main Challenges

Symmetric implementations

Linear operations

- Boolean masking

Non-linear operations

- Boolean masking
- Generic (non-linear) multiplications

Lattice-Based Signatures

Linear operations

- Boolean & Arithmetic masking
- Conversions

Non-linear operations

- Boolean & Arithmetic masking
- Conversions
- Secret-dependent branches
- New generic algorithms to exhibit

Example: GLP Signature Scheme

Only uniform distributions

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$

Steps

1. Identify sensitive variables
2. Identify operations to mask
3. Mask atomic operations & convert Boolean / arithmetic masking
4. Safely compose operations

[CHES:GLP12] *Practical lattice-based cryptography: A signature scheme for embedded systems.* T. Güneysu, V. Lyubashevsky, and T. Pöppelmann. CHES 2012.

Example: GLP Signature Scheme

Only uniform distributions

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$

Steps

1. Identify sensitive variables
2. Identify operations to mask
3. Mask atomic operations & convert Boolean / arithmetic masking
4. Safely compose operations

[EC:BBEFGRT18] *Masking the GLP Lattice-Based Signature Scheme at Any Order.* Gilles Barthe, Sonia Belaïd, Thomas Espitau, Pierre-Alain Fouque, Benjamin Grégoire, Mélissa Rossi, and Mehdi Tibouchi. Eurocrypt 2018

Example: GLP Signature Scheme

Only uniform distributions

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$

Steps

1. Identify sensitive variables
2. Identify operations to mask
3. Mask atomic operations & convert Boolean / arithmetic masking
4. Safely compose operations

Example: GLP Signature Scheme

Only uniform distributions

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$

Steps

1. Identify sensitive variables
2. Identify operations to mask
3. Mask atomic operations & convert Boolean / arithmetic masking
4. Safely compose operations

Example: GLP Signature Scheme

Only uniform distributions

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$

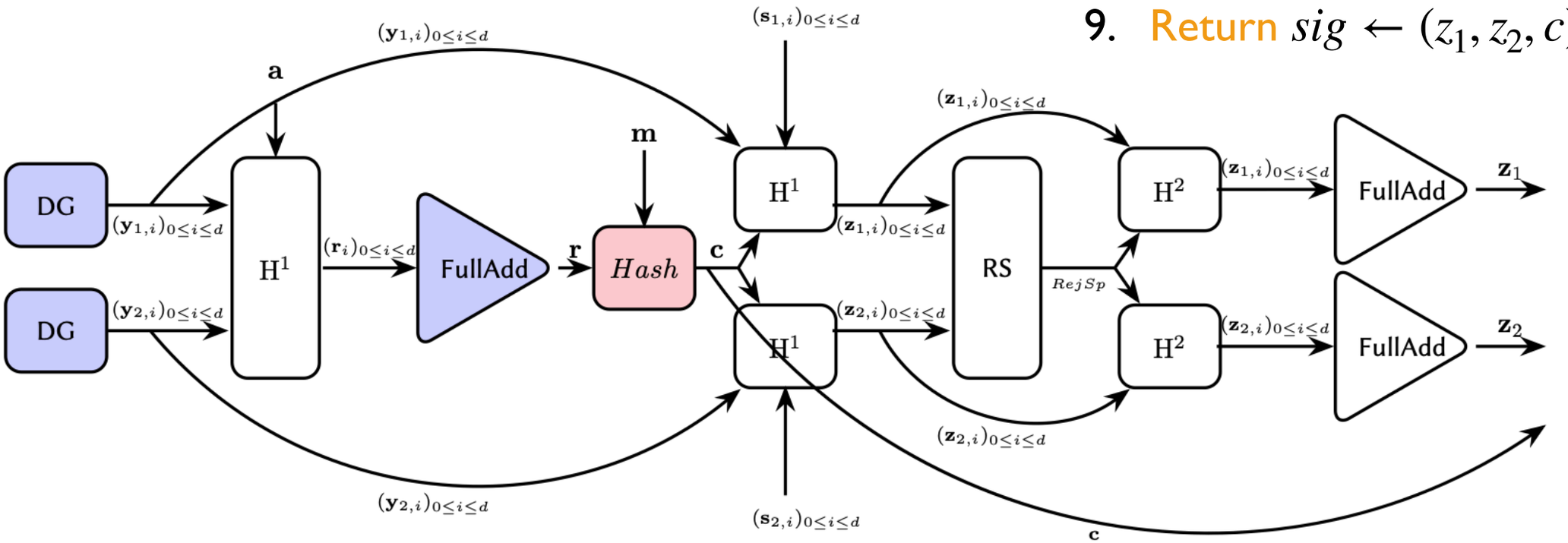
Steps

1. Identify sensitive variables
2. Identify operations to mask
3. Mask atomic operations & convert Boolean / arithmetic masking
4. Safely compose operations

Example: GLP Signature Scheme

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

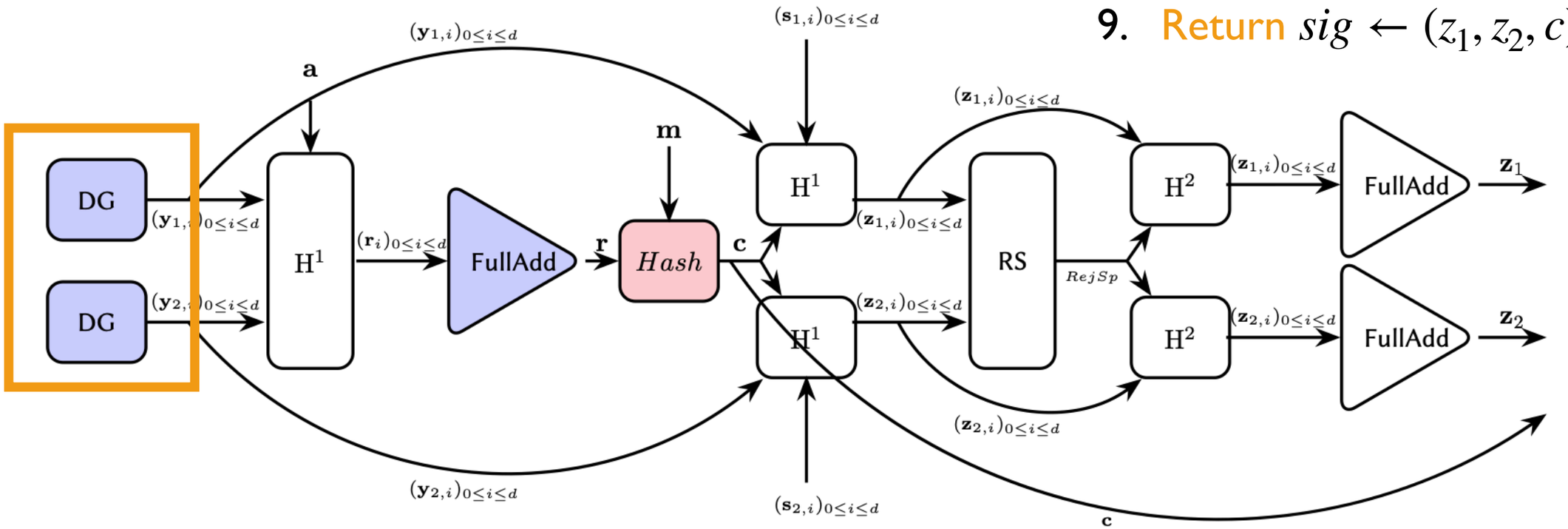
1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$



Example: GLP Signature Scheme

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$

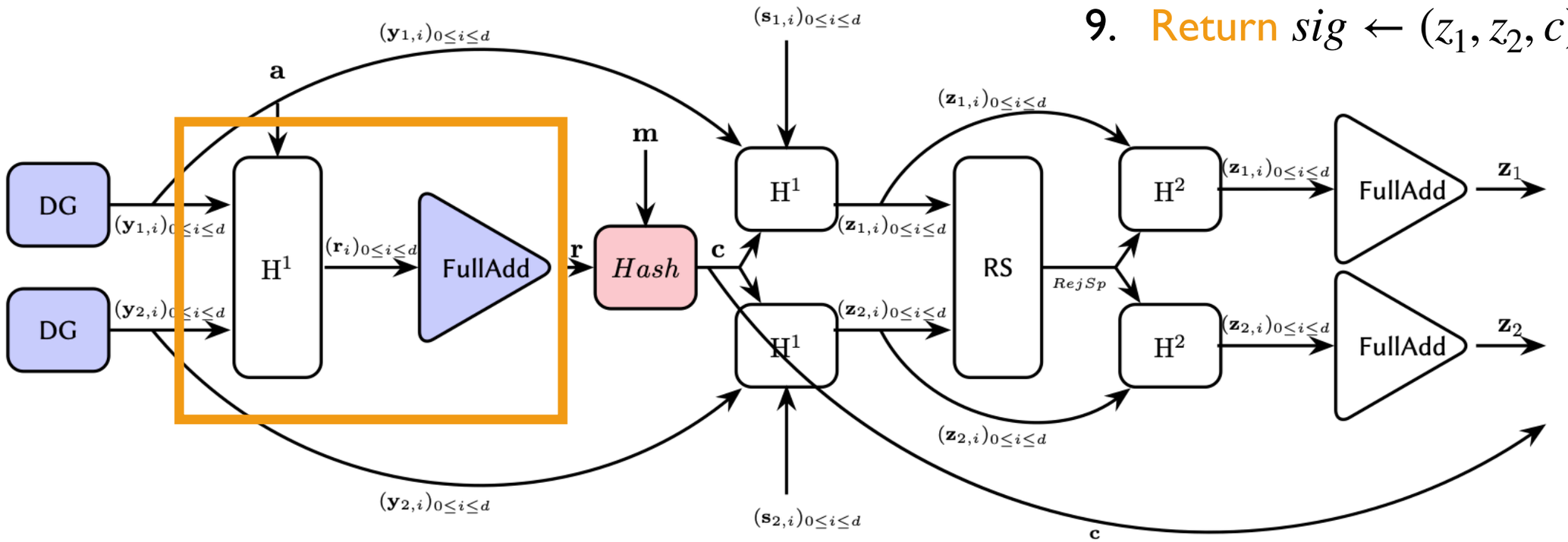


CryptoExperts

Example: GLP Signature Scheme

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$

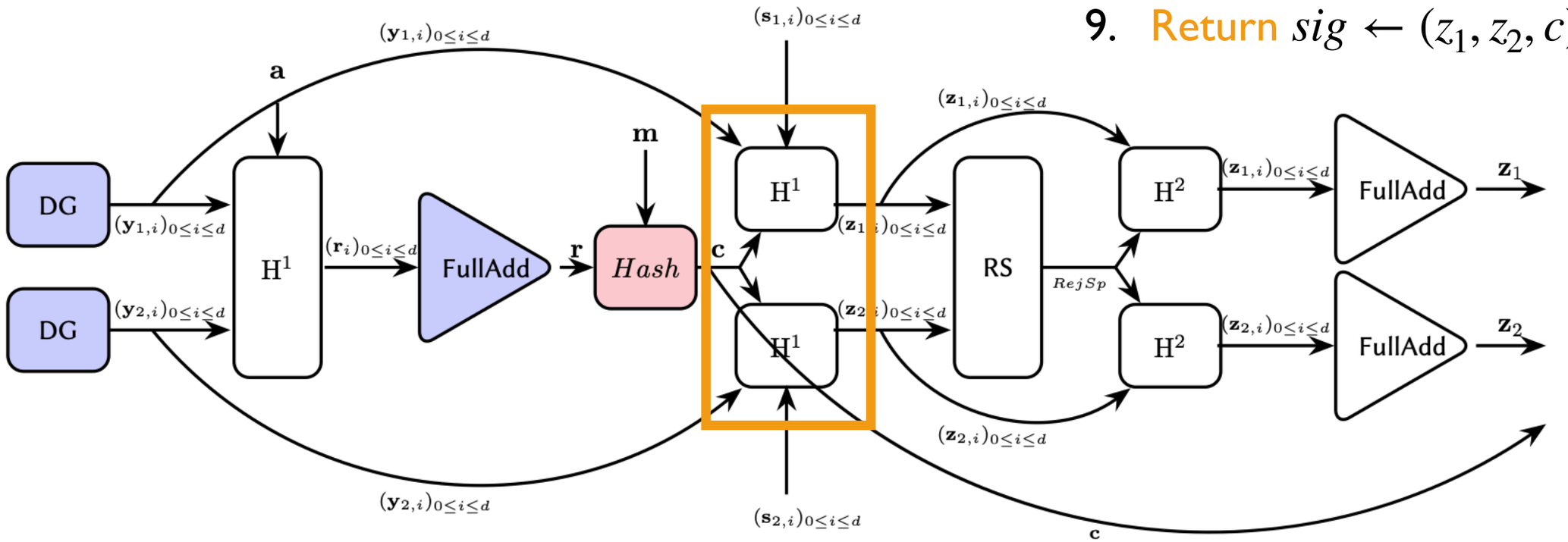


CryptoExperts

Example: GLP Signature Scheme

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$

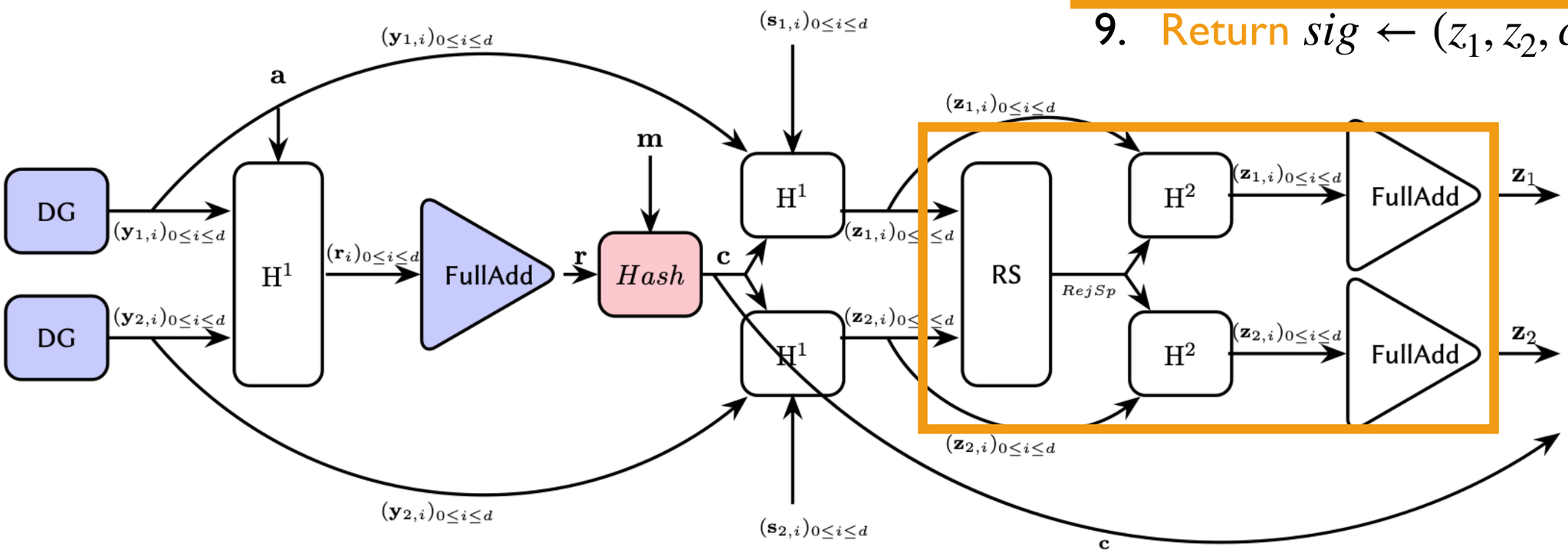


CryptoExperts

Example: GLP Signature Scheme

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$



Example: GLP Signature Scheme

Only uniform distributions

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$

Steps

1. Identify sensitive variables
2. Identify operations to mask
3. Mask atomic operations & convert Boolean / arithmetic masking
4. Safely compose operations

Example: GLP Signature Scheme

Only uniform distributions

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$

Steps

1. Identify sensitive variables
2. Identify operations to mask
3. Mask atomic operations & convert Boolean / arithmetic masking
4. Safely compose operations

Example: GLP Signature Scheme

Only uniform distributions

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$

Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$

Generate each coefficient in $[-k, k]$:

- generate Boolean masking of a uniformly random value
  - with rejection
- convert the Boolean masking into an arithmetic masking
  - variant of existing conversions but with arbitrary modulus

Example: GLP Signature Scheme

Only uniform distributions

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$

- Convert both arithmetic maskings into Boolean maskings
- Compare the maskings by safely checking the most significant bit of the difference
- Unmask the final result

Example: GLP Signature Scheme

Only uniform distributions

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$

Steps

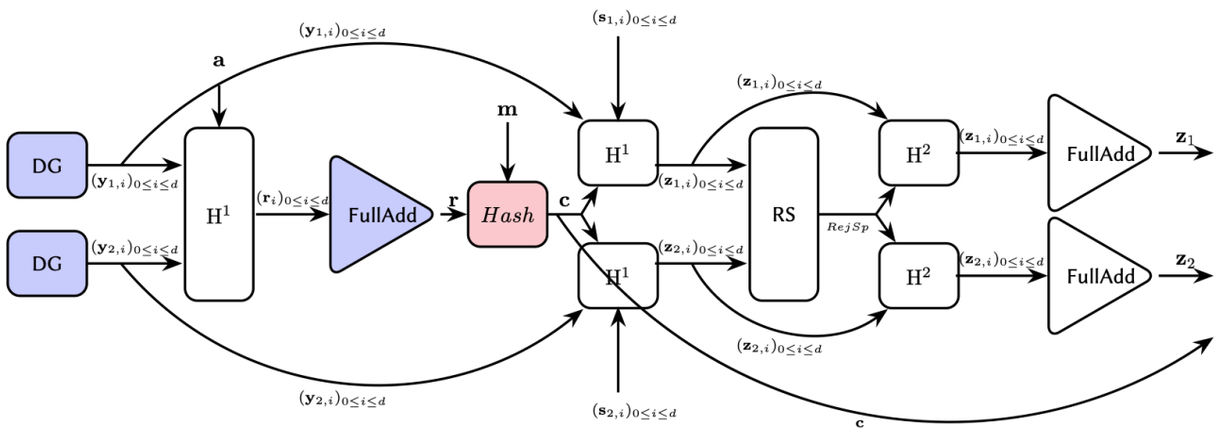
1. Identify sensitive variables
2. Identify operations to mask
3. Mask atomic operations & convert Boolean / arithmetic masking
4. Safely compose operations

Example: GLP Signature Scheme

Only uniform distributions

Inputs: $m, pk = (a, t), sk = (s_1, s_2)$
Outputs: signature sig

1. $y_1, y_2 \leftarrow R_{q,\kappa}$
2. $r \leftarrow a \cdot y_1 + y_2$
3. $c \leftarrow \text{hash}(r,m)$
4. $z_1 \leftarrow s_1 \cdot c + y_1$
5. $z_2 \leftarrow s_2 \cdot c + y_2$
6. If z_1 or $z_2 \notin R_{q,\kappa-\alpha}$ then
7. Restart
8. End
9. Return $sig \leftarrow (z_1, z_2, c)$



Steps

1. Identify sensitive variables
2. Identify operations to mask
3. Mask atomic operations & convert Boolean / arithmetic masking
4. Safely compose operations
   - Add randomness at careful locations

Other Signature Schemes

With Gaussian distributions

- Gaussian distributions
  - Pros: better parameters and security reductions
  - Cons: secure implementations are difficult to build for discrete Gaussians (timing attacks, power analysis attacks)

Almost all (PQC NIST) candidates chose to stay away from Gaussian distributions

Other Signature Schemes

With Gaussian distributions

- Gaussian distributions
  - Pros: better parameters and security reductions
  - Cons: secure implementations are difficult to build for discrete Gaussians (timing attacks, power analysis attacks)

Almost all (PQC NIST) candidates chose to stay away from Gaussian distributions

- Example
  - **BLISS**: Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. *Lattice Signatures and Bimodal Gaussians*. In CRYPTO 2013

Other Signature Schemes

With Gaussian distributions

- Main Challenges
  - Gaussian sampling $\mathscr{D}_\sigma$
  - Rejection Sampling

Other Signature Schemes

With Gaussian distributions

- Main Challenges
  - Gaussian sampling $\mathscr{D}_\sigma$
  - Rejection Sampling

[CCS:BBEFRT] G. Barthe, S. Belaïd, T. Espitau, P.-A. Fouque, M. Rossi, and M. Tibouchi. *GALACTICS: Gaussian Sampling for Lattice-Based Constant-Time Implementation of Cryptographic Signatures, Revisited.* In CCS 2019.

Other Signature Schemes

With Gaussian distributions

- Main Challenges
  - Gaussian sampling $\mathscr{D}_\sigma$

Other Signature Schemes

With Gaussian distributions

- Main Challenges
  - Gaussian sampling $\mathscr{D}_\sigma$
    - Naive method: cumulative distribution table (CDT) sampling
      - Generate a random value α in [0,1] with high precision
      - Return the index of the first entry greater than α

Other Signature Schemes

With Gaussian distributions

- Main Challenges
  - Gaussian sampling $\mathscr{D}_\sigma$
    - Naive method: cumulative distribution table (CDT) sampling
      - Generate a random value α in [0,1] with high precision
      - Return the index of the first entry greater than α
    - Better strategy: construct a distribution somehow close to $\mathscr{D}_\sigma^+$ (with smaller standard deviation) then use rejection sampling
      - Less entries for the CDT
      - Relies on uniform generation + rejection sampling
      - Masking of the comparisons in the CDT

Other Signature Schemes

With Gaussian distributions

- Main Challenges
  - Rejection Sampling
    - *Main idea: compare uniform values in [0,1] to some probability values*

Other Signature Schemes

With Gaussian distributions

- Main Challenges
  - Rejection Sampling
    - *Main idea: compare uniform values in [0,1] to some probability values*
    - Naive method: repeated Bernoulli trials with known constant probabilities
      - Not in constant time

Other Signature Schemes

With Gaussian distributions

- Main Challenges
  - Rejection Sampling
    - *Main idea: compare uniform values in [0,1] to some probability values*
    - Naive method: repeated Bernoulli trials with known constant probabilities
      - Not in constant time
    - Better method: approximate the probabilities of rejection with a sufficiently close polynomial
      - Integer arithmetic only
      - High precision with the methodology [AC:Prest17] based on the Rényi divergence
      - Masking of the comparison using Boolean sharing

[AC:Prest17] Thomas Prest. *Sharper Bounds in Lattice-Based Cryptography Using the Rényi Divergence*. In ASIACRYPT 2017.

Conclusion

Conclusion

- Operations in lattice-based signatures require new masking techniques
  - Conversion between Boolean and arithmetic masking (with different modulus)
  - Sampling from different distributions
    - Uniform distributions
    - More challenging Gaussian distributions
    - etc.

- Remaining Challenges
  - Many schemes are not investigated yet
  - Maybe different operations to mask
  - Design masking-friendly signature schemes (e.g., Mitaka, AsiaCCS 2021)

Related Works

[EC:BBEFGRT18] *Masking the GLP Lattice-Based Signature Scheme at Any Order.* Gilles Barthe, Sonia Belaïd, Thomas Espitau, Pierre-Alain Fouque, Benjamin Grégoire, Mélissa Rossi, and Mehdi Tibouchi. Eurocrypt 2018

[CCS:BBEFRT19] *GALACTICS: Gaussian Sampling for Lattice-Based Constant-Time Implementation of Cryptographic Signatures, Revisited.* Gilles Barthe, Sonia Belaïd, Thomas Espitau, Pierre-Alain Fouque, Mélissa Rossi, and Mehdi Tibouchi. CCS 2019

[CARDIS:GR19] *An Efficient and Provable Masked Implementation of qTESLA.* François Gérard, Mélissa Rossi. CARDIS 2019

[ACNS:MGTF19] *Masking Dilithium - Efficient Implementation and Side-Channel Evaluation.* Vincent Migliore, Benoît Gérard, Mehdi Tibouchi, Pierre-Alain Fouque. ACNS 2019

[AsiaCCS:E21] *Mitaka: Faster, Simpler, Parallelizable and Maskable Hash-and-Sign Signatures on NTRU Lattices.* Thomas Espitau. AsiaCCS 2021