SIDE-CHANNEL ATTACKS AND COUNTERMEASURES FOR POST-QUANTUM CRYPTOGRAPHY

Fatemeh Gholizadeh
ECE 646
December 7, 2018
Dr. Kris Gaj
Outline

• Introduction

• Side-channel attack against post-quantum cryptography
  • Lattice-Based Cryptography
  • Code-Based Cryptography
  • Hash-Based Cryptography
  • Multivariate Cryptography
  • Isogeny-based Cryptography
Introduction

• Side-Channel Attacks (SCA)
  • Power analysis attack
  • Timing analysis attack

• Post-quantum cryptosystems (PQC) are susceptible to SCA

• Need to evaluate the security of the PQC against side-channel attacks
Lattice-Based Cryptography (LBC)

- The key building blocks of the LBC
  - Matrix-vector multiplication
  - Polynomial multiplication
  - Discrete Gaussian sampling
    - Lattice-based cryptography based on LWE and SIS problems require discrete Gaussian noise to mask the scheme’s secret key
Lattice-based key Exchange protocols

- NewHope: performs polynomial multiplication, rely on R-LWE
- Frodo: performs matrix multiplication, rely on LWE
- Horizontal DPA targets the hardware implementation of matrix and polynomial multiplication
- Implemented on a SAKURA-G FPGA Board
- Parameter
  - Frodo: uses matrix size of 752*8 or 8*752, multiplication of size 752*752, integer elements modulo of $2^{15}$
  - NewHope: uses polynomials of degree 1023 with integer coefficients modulo 12289
An Example of Horizontal DPA Attack on the Polynomial Multiplication

Aysu, A., Horizontal Side-Channel Vulnerabilities of Post-Quantum Key Exchange Protocols, HOST, 2018
DPA Against Key Exchange Protocol

- Targets intermediate state updates of matrix and polynomial multiplications

- extract the secret keys with over 99% probability with
  - 1024 horizontal tests for NewHope
  - 752 horizontal tests for Frodo

- Countermeasure
  - adding dummy steps in the computation
  - randomizing the order of computations
Code-Based Cryptography

- McEliece cryptosystem
  - based on (QC)-MDPC or binary Goppa codes
  - relies on the hardness of decoding

- DPA attack against a FPGA implementation of QC-MDPC McEliece

- 80-bit security level
  - n0 = 2, n = 9602, r = 4801, w = 90, t = 84

Resource requirements
- 64 slices and 1 block
- RAM (BRAM) to implement encryption and 159 slices
- 3 BRAMs to implement decryption on a Xilinx Spartan-6 XC6SLX4 FPGA
Targets of DPA Attacks on FPGA Implementation of QC-MDPC McEliece

Horizontal DPA attack
extract key with tens of traces

Vertical DPA attack
Extract key with thousands of chosen ciphertext trace

key row \((h_0 \ggg z||h_1 \ggg z)\) constantly rotated during the syndrome generation, \(z \in \{0, 4800\}\), so each row of \(H\) contains the same information.

Chen, C. Horizontal and Vertical Side Channel Analysis of a McEliece Cryptosystem, IEEE-TIFS, 2016
Power Analysis against Code-Based Cryptography

• Horizontal DPA attack
  • Target the leakage of a key rotation operation during the syndrome computation step of the decryption algorithm
  • Can succeed with tens of trace without the ciphertext

• Vertical attack
  • Target the leakage of the syndrome register during syndrome computation step of encryption
  • Recover key with thousands of chosen ciphertext trace

• Countermeasure
  • Masking and Hiding
Hash-Based Cryptography

- Rely on the security of an underlying hash function
- Use a binary hash tree structure

Hash-based signature schemes

- XMSS
  - Statefull, SCA resistant

- SPHINCS
  - Stateless, SCA vulnerability

- XMSS $^\text{MT}$
  - Statefull, SCA resistant
DPA attack on SPHINCS-256

- **SPHICS-256**
  - Uses a hyper tree height of h
  - upper layers use XMSS$^\text{MT}$ and W-OTS$^+$ to sign the roots of the child trees
  - the lowest layer uses a Merkle tree with HORST for signing messages (SPHINCS-256 uses $2^{60}$ HORST key pairs)

- **BLAKE-256 ($sk_1|| A$) : generate secret seeds of W-OTS$^+$ and HORST**
  - SPHINCS secret key
  - address of the instance

- Software implementation on the Atmel SAM3X8E Cortex-M3 CPU
6-DPA Attack on SPHINCS-256

Algorithm 1: Round $z = 0$ of BLAKE-256 compression algorithm
Input: $(s_0, \ldots, s_7)$ — secret key $sk_1$ split into 8 chunks of 32 bits each
Input: $(a_0, a_1)$ — address $A$ split into two chunks of 32 bits each

1: $Mix(v_0, v_4, v_8, v_{12}; s_0, s_1)$
2: $Mix(v_1, v_5, v_9, v_{13}; s_2, s_3)$
3: $Mix(v_2, v_6, v_{10}, v_{14}; s_4, s_5)$
4: $Mix(v_3, v_7, v_{11}, v_{15}; s_6, s_7)$

5: $Mix(v_0, v_5, v_{10}, v_{15}; a_0, a_1)$
6: $Mix(v_1, v_6, v_{11}, v_{12}; 0x80000000, 0x00000000)$
7: $Mix(v_2, v_7, v_8, v_{13}; 0x00000000, 0x00000001)$
8: $Mix(v_3, v_4, v_9, v_{14}; 0x00000000, 0x00000140)$

Algorithm 2: Mix procedure involved in Alg. 1.
$\sigma_z(i)$ — a permutation function
$C_i$ — constant $0 \leq i < 15$

Input: $(v_a, v_b, v_c, v_d)$ — intermediate values of 32 bits each
Input: $(M_{\sigma_z(e)}, M_{\sigma_z(e+1)})$ — hash function input chunks of 32 bits each

1: $v_a \leftarrow (v_a + v_b) + (M_{\sigma_z(e)} \oplus C_{\sigma_z(e+1)})$
2: $v_d \leftarrow (v_d \oplus v_a) \ll 16$
3: $v_c \leftarrow v_c + v_d$
4: $v_b \leftarrow (v_b \oplus v_d) \ll 12$

5: $v_a \leftarrow (v_a + v_b) + (M_{\sigma_z(e+1)} \cdot C_{\sigma_z(e)})$
6: $v_d \leftarrow (v_d \oplus v_a) \ll 8$
7: $v_c \leftarrow v_c + v_d$
8: $v_b \leftarrow (v_b \oplus v_d) \ll 7$
DPA attack on SPHINCS-256

- 6 - DPA attack on the BLAKE hash function that recovers one 32-bit chunk of the secret key sk1

- The BLAKE-256 compression procedure
  - takes 12 similar rounds during which the input is mixed
  - DPA focus on first two round

- Result of first two DPA attack
  - Collection of 10000 different real traces from signing of almost 2000 different messages resulted in extracting the key

- Countermeasures
  - Hiding the order of the Mix procedures
Multivariate Cryptography

• Based on multivariate quadratic polynomials

MQ-based signature schemes

(enTTS)

Unbalanced Oil-and-Vinegar (UOV)

Rainbow
enTTS (20,28) scheme

- enTTS(20, 28) Implemented on TSMC-0.18 μm standard cell CMOS ASICs
  - Hash size
  - Signature size

- DPA attacks on Affine transformations (L1 and L2)

- Countermeasure
  - Masking and hiding
The Flowchart of Signature Generation of enTTS

Affine transformations (matrix–vector multiplications and vector additions)

Yi, On the Importance of Checking Multivariate Public Key Cryptography for Side-Channel Attacks: The Case of enTTS Scheme, COMPUTER
DPA Attack on Affine Transformation

Algorithm of Affine transformation
L1

1: var
2: i,j: Integer: = 0;
3: begin
4: for i = 0 to 19 do \( y_i = y_i + b_i \);
5: for i = 0 to 19 do
6: for j = 0 to 19 do
7: \( a_{ij} = a_{ij} \times y_j \);
8: for i = 0 to 19 do
9: \( y_i = \left(00, \ldots, 00\right)_2 \);
10: for j = 0 to 19 do
11: \( y_i = y_i + a_{ij} \);
12: end.

- Vector \( b \) with size 20, part of private key
- Matrix \( a \) with size 20×20, part of private key
- 20 bytes Message

Adversary analyzes power consumption of these lines to extract the secret key countermeasure: hiding and masking
Isogeny-Based Cryptography

- Security is based on the difficulty of computing isogenies between supersingular elliptic curves

- Supersingular isogeny Diffie-Hellman (SIDH) key exchange
  - small keys
  - forward secrecy
  - Diffie-Hellman key exchange
Supersingular isogeny Diffie-Hellman (SIDH) Key Exchange

- SIDH can be broken down into

  - secret kernel generation
    - $R = h \langle [m]P + [n]Q \rangle$ for torsion basis points $\{P, Q\}$, where $m$ and $n$ are private keys
    - involves the secret key as a scalar and susceptible for SCA
  
  - Computation of large-degree isogeny over the secret kernel
    $\phi : E \longrightarrow E/ \langle R \rangle$
    - iteratively computing isogenies of a base degree to perform a isogeny graph walk based on the secret kernel
    - isogeny path decision susceptible for SCA
RPA Attack against SIDH

- SIDH uses Montgomery curves in its implementations
- zero-point attacks on the three-point Montgomery ladder

Zero-point Attack
- each step of the three-point ladder produces \([t]Q, [t+1]Q, P+[t]Q\)
- predict each bit of the key as a '0' or '1' and then validate that assumption with a forced zero point

Countermeasures
- Dynamic keys
- Initial random isogeny
- Private key representation randomization
- Point blinding
Attacks on Three-point Differential Ladder

Three-point differential ladder to compute $P + [t]Q$.

$dadd( P, Q, (P - Q).x)$” represent a differential point addition of $P$ and $Q$, where the x-coordinate of $P$-$Q$ is known.

Input: Point $P$ and $Q$ on an elliptic curve $E$, scalar $d$ which is $k$ bits
1: Set $A=0$, $B=Q$, $C=P$
2: Compute $Q - P$
3: for $I$ decreasing from $|d|$ downto 1 do
4: Let $d_i$ be the i-th bit of $d$
5: if $d_i = 0$ then
6: $B = dadd(A, B, Q)$, $C = dadd(A, C, P)$, $A = 2A$
7: else
8: $A = dadd(A, B, Q)$, $C = dadd(B, C, Q - P)$, $B = 2B$
9: end if
10: end for
Ensure: $C = P + [t]Q$

Zero-Point Attack
Partial-Zero Attack

$Q - P$ is partial-zero, $P$ is non-zero
Results in a power difference for $d_i = 0$ and $d_i = 1$

Countermeasure
- Reject a partial-zero $P$ or a partial-zero $Q - P$
- Randomize representation of $P$ and $Q - P$ to non-zero elements

Kozie, B. Side-Channel Attacks on Quantum-Resistant Supersingular Isogeny Die-Hellman, SAC, 2017
Zero-value attack on the large-degree isogeny computation

- **Zero-value isogeny coefficient attack**
  - Force an isogeny to compute a curve with a full-zero coefficient ($A = 0$ or $B = 0$ for an elliptic curve)
  - Can be mounted against the second round of static-key SIDH

- **Zero-value isogeny point attack**
  - Force an isogeny to compute on a point with a zero-value ($x = 0$ or $y = 0$)
  - Can target torsion basis points in first round of SIDH or intermediate kernel point in either round

- **Countermeasure**
  - randomize the resulting isogenous curves
Conclusion

• Lattice-based cryptography and code-based cryptography
  • Received most investigations
  • Encryption, strong candidates
  • Signature

• Hash-based cryptography
  • Signature, strong candidate
  • No effective timing analysis attack exist so far

• Multivariate cryptography
  • Encryption
  • Signature, strong candidate

• Supersingular Isogeny-based cryptography
  • Newest and need more investigation
  • Encryption