MH4311 Cryptography

Lecture 20

Post-Quantum Cryptography

Wu Hongjun

Lecture Outline

- Classical ciphers
- Symmetric key encryption
- Hash function and Message Authentication Code
- Public key encryption
- Digital signature
- Key establishment and management
- Elliptic curve public key cryptosystems
- Introduction to other cryptographic topics
  - **Post-quantum cryptography**
  - Side-channel attacks

Recommended Reading

- Wikipedia:
  - Post-quantum Cryptography
    https://en.wikipedia.org/wiki/Post-quantum_cryptography
  - NIST post-quantum cryptography project
    https://csrc.nist.gov/Projects/Post-Quantum-Cryptography

Quantum Computer

Quantum Computer

- Different from the computers we are using
- Based on quantum entanglement and superposition
- A register containing *n* quantum bits (qubits) can contain 2^n different values at the same time
  - An *n*-bit register on our computer contains only one value at any time
- An operation performed on an *n*-qubit quantum register is applied to all those 2^n values in the register at the same time
  - An operation performed on an *n*-bit register on our computer is applied to a single value in the register

Theory on quantum computer

- In the early 1980s, it was realized that quantum computer can be used to significantly improve the efficiency of quantum simulation
  - You can quickly simulate any chemical reaction on quantum computer
  - Important to the chemical and pharmaceutical industries

Theory on quantum computer

- In 1994, Peter Shor showed that: on quantum computers, integer factorization and discrete logarithm problem can be solved with very low complexity
  - Complexity: $O((\log N)^2(\log\log N)(\log\log\log N))$
    - N is the value of modulus in RSA or the value of the prime number p in ElGamal cryptosystem
    - RSA is unsafe even with one-million bit modulus
  - Shor's algorithm can be applied to break almost all the public key cryptosystems we are using today
  - Governments started to build quantum computers

Theory on quantum computer

- In 1996, Lov Grover showed that on quantum computer,
  - breaking a cipher with *n*-bit secret key requires about $2^{n/2}$ computations of the cipher;
  - breaking a hash function with *n*-bit message digest requires about $2^{n/3}$ computations of the hash function.

Building a quantum computer

- It is hard to build a quantum computer with large registers because the entanglement of quantum bits is fragile
  - To break RSA with 2048-bit modulus, at least a 4096-bit quantum register is needed (in order to correct errors, the register should be much larger than 4096 bits)
  - In contrary, the widely used Intel Core-i7 CPU uses only 16 64-bit registers for all the applications

Building a quantum computer

- In 2001, IBM built 7-qubit quantum computer
  - Factorized number 15
- Starting from 2011, the Canadian company D-Wave Systems started to announce several “fake” quantum computers with large registers (ranging from 128 qubits to 2000 qubits)
  - NASA and Google bought a quantum computer from D-Wave with $15 million USD
- May 2017: IBM built 16-qubit quantum computer
- Nov 2017: IBM announced 50-qubit quantum computer
- Jan 2018: Intel announced 49-qubit quantum computer
- Mar 2018: Google announced 72-qubit quantum computer

➔ Be cautious of the recent claims/announcements

➔ Quantum computer is still in its infancy at the moment

Post-Quantum Cryptography

Post-Quantum Cryptography

- Due to the threat that quantum computer can be applied to break most of the public key cryptosystems being used today, it is necessary to design public key cryptosystems that are secure against an attack launched on quantum computer
- Post-quantum cryptography mainly refers to post-quantum public key cryptosystems

Post-Quantum Cryptography

- There are three main approaches to design post-quantum public key cryptosystems
  - Lattice-based PKC
    - Based on the difficulty of finding the shortest vector
    - Example: NTRUEncrypt (1998)
  - Multivariate PKC
    - Based on the difficulty of solving systems of nonlinear multivariate equations
    - Example: Unbalanced Oil and Vinegar (1999)
  - Coding-based PKC (based on error-correction code)
    - Based on the difficulty of decoding a error-added codeword without knowing the decoding algorithm
    - Example: McEliece (1978)

Post-Quantum Cryptography

- In 2017, NIST organized a competition on post-quantum cryptography (now ongoing)
  - 69 candidates in Round 1
    - public key encryption
    - key exchange
    - digital signature
  - The winner(s) will be standardized

https://en.wikipedia.org/wiki/Post-Quantum_Cryptography_Standardization

NTRUEncrypt

- We introduce one version of NTRUEncrypt
- NTRUEncrypt Public Parameters: ***N, p, q***
  - N a prime ($250 < N < 2500$)
  - q large modulus (say $q = 128$ or 256)
  - p small modulus (say $p = 3$)

    (p and q are coprime)

NTRUEncrypt

- NTRUEncrypt private key: ***f***

  F a secret polynomial with degree N-1
  each coefficient randomly chosen from {-1, 0, 1}

  G a secret polynomial with degree N-1
  each coefficient randomly chosen from {-1, 0, 1}

  $f = 1+pF$

  $g = pG$

- NTRUEncrypt public key: ***h***

  $h = f^{-1} \cdot g \pmod{X^N-1} \pmod{q}$

  (f is chosen so that the inverse of f exists in the polynomial ring with reduction polynomial $X^N - 1$, the addition and multiplication of the coefficients are computed modulo q)

NTRUEncrypt

- NTRUEncrypt encryption

  m the plaintext is encoded as a polynomial with degree N-1 with coefficients from {-1, 0, 1}

  r a one-time secret polynomial with degree N-1

  each coefficient randomly chosen from {-1, 0, 1}

To encrypt m, randomly generate a one-time secret polynomial r, compute the ciphertext c as:

$$c = r \cdot h + m \pmod{X^N - 1} \pmod{q}$$

NTRUEncrypt

- NTRUEncrypt decryption

  To decrypt the ciphertext c,

  Step 1. Compute $a = f \cdot c \pmod{X^N - 1} \pmod{q}$

  Step 2. Write the coefficients of a in range [-q/2, q/2]

  Step 3. $m = a \pmod{p}$

NTRUEncrypt

- NTRUEncrypt decryption recovers the plaintext

$$a = f \cdot c \pmod{X^N - 1} \pmod{q}$$
$$= f \cdot (r \cdot h + m) \pmod{X^N - 1} \pmod{q}$$
$$= f \cdot (r \cdot f^{-1} \cdot g + m) \pmod{X^N - 1} \pmod{q}$$
$$= r \cdot g + f \cdot m \pmod{X^N - 1} \pmod{q}$$

Write the coefficients of a in the range $[-q/2, q/2]$

The coefficients of f, r, g and m are very small, so the chance is extremely high that a can be written as:

$$a = r \cdot g + f \cdot m \pmod{X^N - 1}$$

So

$$a \bmod p = r \cdot pG + (1 + pf) \cdot m \pmod{X^N - 1} \pmod{p} = m$$

Summary

- Quantum computer
- Attack on quantum computer
  - Efficient against factorization and discrete logarithm problem
  - Complexity $2^{n/2}$ against a symmetric key cipher with n-bit key
  - Complexity $2^{n/3}$ against a hash function with n-bit message digest
- Post-quantum cryptography
  - Secure against the attacks on quantum computer
  - Lattice-based PKC
    - NTRUEncrypt
  - Multivariate PKC
  - Coding-based PKC