Applied Cryptography and Computer Security
CSE 664 Spring 2020

Lecture 22: Post-Quantum Cryptography

Department of Computer Science and Engineering
University at Buffalo

Overview

- We'll briefly discuss the implications of quantum computing on cryptography
  - quantum computing basics
  - impact of quantum computers on conventional cryptography
  - post-quantum cryptographic algorithms

Quantum Computing

- Classical computers process the input data sequentially
  - a bit is the elementary unit of information
  - computation can be represented as a Boolean circuit composed of elementary gates
  - an n-bit input x can take up to 2^n time to process
    - e.g., by performing computation on all possible n-bit values y and determining which $f(y)$ matches x
- Quantum computers can compute all 2^n values simultaneously
  - the basic information unit is a quantum bit, or qubit
  - quantum computing uses quantum circuits

Quantum Computing

- It is important to understand the computing model and its restrictions
  - each qubit can assume infinitely many states, but only one classical bit can be extracted (or measured)
  - each qubit measurement is probabilistic
  - the internal state of a quantum computer is inaccessible and only a single output can be extracted
  - because the output is probabilistic, quantum algorithms have to be carefully designed to be useful

Quantum Computing

- A qubit can assume infinitely many states between 0 and 1
  - the state is represented by a normalized vector in $\mathbb{C}^2$
  - using the standard basis $e_1 = (1, 0)$ and $e_2 = (0, 1)$, the basis states are denoted by $|0\rangle$ and $|1\rangle$
  - the state of a qubit $|\psi\rangle = a|0\rangle + b|1\rangle$ is a linear combination of the basis states $|0\rangle$ and $|1\rangle$, where $a, b \in \mathbb{C}$ and $|a|^2 + |b|^2 = 1$
  - coefficients a and b can be interpreted as probabilities and a qubit as a random variable
  - a measurement changes the state of qubit and yields a regular bit
  - the original state of a qubit (i.e., a and b) is lost after the measurement and cannot be directly extracted

Quantum Computing

- More on qubits
  - the state of a qubit determines the probability of the result of a measurement
    - the probability of 0 is $|a|^2$ and the probability of 1 is $|b|^2$
  - for instance, measurement of a qubit with state $|0\rangle = 1 \cdot |0\rangle + 0 \cdot |1\rangle$ always gives 0
  - however, a qubit with state $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ outputs both 0 and 1 with probability 1/2
    - we denote such a qubit that outputs a uniform random bit by $|+\rangle$

Quantum Computing

- Quantum gates
  - a quantum gate U with a single input and output qubit is described by a unitary 2×2 matrix $\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$
  - a state $|\psi\rangle = a|0\rangle + b|1\rangle$ is transformed into

$$U|\psi\rangle = U(a|0\rangle + b|1\rangle) = (c_{11}a + c_{12}b)|0\rangle + (c_{21}a + c_{22}b)|1\rangle$$

  - for example, the quantum analog of the NOT gate is given by matrix $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
    - it transforms state $|\psi\rangle = a|0\rangle + b|1\rangle$ into $|\overline{\psi}\rangle = b|0\rangle + a|1\rangle$

Quantum Computing

- Quantum gates
  - another useful gate is called the Hadamard gate, described by matrix

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

    - because $H \cdot \binom{1}{0} = \frac{1}{\sqrt{2}}\binom{1}{1}$, the state $|0\rangle$ is transformed into $|+\rangle$
    - this is very useful for producing a balanced superposition (linear combination) of basis states
    - i.e., it turns a 0 qubit into a qubit that is simultaneously 0 and 1
    - measuring $H|0\rangle$ gives a uniform random bit

Quantum Computing

- More interesting quantum operations require systems of multiple qubits
  - a system of n qubits can represent 2^n states simultaneously
  - the basis states are $|x_1 x_2 \ldots x_n\rangle$, where $x_i \in \{0, 1\}$
  - states in an n-qubit system are a superposition of the 2^n basis states
    - this is not the same as n individual qubits
  - states are represented by the n-fold tensor product of $\mathbb{C}^2$:

$$\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2 = (\mathbb{C}^2)^{\otimes n}$$

  - e.g., a 2-qubit system is represented by a state in $\mathbb{C}^2 \otimes \mathbb{C}^2$ with basis states $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$
    - states are $|\psi\rangle = a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle$, where $|a_{00}|^2 + |a_{01}|^2 + |a_{10}|^2 + |a_{11}|^2 = 1$

Quantum Computing

- Quantum algorithms
  - computation takes form of quantum circuits processing qubits
  - basic building blocks are quantum logic gates, which implement unitary (and therefore reversible) transformation
    - elementary gates in classical circuits are typically not reversible
  - one example is controlled-NOT gate $\text{CNOT}|x, y\rangle = |x, x \oplus y\rangle$
    - it leaves the first (control) bit unchanged and flips the second (target) bit if control bit is 1
    - the CNOT gate is represented by the unitary matrix

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Quantum Computing

- Theorem: Single qubit gates and the CNOT gate are sufficient to implement an arbitrary unitary operation on n qubits

- The Walsh-Hadamard transformation W generalizes the Hadamard gate to transform the 0 state into a balanced superposition of 2^n basis states
  - quantum algorithms can use this superposition to simultaneously compute all values of function $f : \{0,1\}^n \rightarrow \{0,1\}^m$

- Because f may not be invertible, it needs to be modified
  - when $n \neq m$, f is not invertible
  - given f, define invertible $F : \{0,1\}^{n+m} \rightarrow \{0,1\}^{n+m}$ as

$$F(x, y) = (x, y \oplus f(x))$$

Quantum Computing

- Quantum Fourier Transform is a key algorithm in quantum computing
  - classical Discrete Fourier Transform maps a sequence of N complex numbers into the frequency domain
    - the result reveals the periodic structure of the input
    - if the data is r-periodic and N is divisible by r, the Fourier coefficients y_k are non-zero only for multiples of N/r
    - more generally, a Fourier amplitude $|y_k| \gg 0$ indicates that N/k is an approximate multiple of the period

- Quantum Fourier Transform
  - the above allows for Quantum Fourier Transform to find a hidden period of input vector of size $N = 2^s$
    - indices k with Fourier coefficients $|y_k|^2 \gg 0$ reveal the period
    - measuring a state of Fourier amplitudes will give such indices k with significant probability
    - QFT has an efficient circuit and runs in $O(s^2)$ time

Quantum Factoring

- In 1994, Shor discovered a quantum polynomial-time factoring algorithm
  - the fastest classical algorithm – number field sieve – run in subexponential, but superpolynomial time
- Shor's algorithm combines QFT with second degree congruences
  - QFT finds a hidden period of a function
  - we use function $f(x) = a^x \bmod n$ to find the hidden period of x
  - the order of a mod n leads to the computation of factors p and q of n
  - it uses $\approx 3 \log n$ qubits and $O((\log n)^3)$ operations

Quantum Factoring

- The idea behind computing factors p and q of n is somewhat similar to that of computing factors from RSA's e and d
  - assume we have the ability to find the hidden period of $a^x \bmod n$, i.e., the order of $a \in \mathbb{Z}_n^*$ mod n
  - choose random $1 < a < n$
    - if $gcd(a, n) \neq 1$, this immediately gives us factors
    - otherwise, order r of a mod n divides $\phi(n) = (p-1)(q-1)$
    - by definition, $a^r \equiv 1 \pmod{n}$
  - if r is even, $a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1) \equiv 0 \pmod{n}$
    - this means that $n | (a^{r/2} - 1)(a^{r/2} + 1)$
    - also, because the order is not $r/2$, $n \not| (a^{r/2} - 1)$

Quantum Factoring

- Factoring of $n = pq$ given r such that $a^r \equiv 1 \pmod{n}$
  - based on the above, we obtain two possibilities
    - p divides one of $a^{r/2} - 1$ and $a^{r/2} + 1$ and q divides the other
      - in this case $\gcd(a^{r/2} + 1, n)$ gives p or q
    - $n | (a^{r/2} + 1)$ and the algorithm fails
      - we have to choose another base a
  - this means the algorithm is successful if r is even and $n \not| (a^{r/2} + 1)$
    - the probability of this is at least 50%
    - i.e., r is odd if and only iff the orders of a in both $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$ are odd
    - and if r is even, we must have $a^{r/2} \equiv -1 \pmod{p}$ and $a^{r/2} \equiv -1 \pmod{q}$ to have $a^{r/2} + 1 \equiv 0 \pmod{n}$

Quantum Factoring

- The remaining step is to determine the unknown order r of residue class $a \in \mathbb{Z}_n^*$
  - we prepare a superposition of input values $x = 0, 1, \ldots, N-1$ using Walsh-Hadamard transformation
  - we apply it to transformation for $a^x \bmod n$ to simultaneously compute all $a^x \bmod n$
  - because the values are r-periodic, $a^x \equiv a^{x+r}$, the QFT is applied to reveal the period with high probability
    - measuring the state gives k, which is an approximate multiple of N/r
    - the exact r is computed using the continued fraction expansion
  - setting $N = 2^s$, where $n^2 \leq N \leq 2n^2$, is a good choice

Post-Quantum Cryptography

- The discrete logarithm problem can also be solved using a period-finding algorithm
  - consider $h = g^y$ for some $G = \langle g \rangle$
  - function $f(x_1, x_2) = h^{x_1} g^{-x_2}$ has period $(1, y)$ because

$$f(x_1 + 1, x_2 + y) = h^{x_1+1} g^{-x_2-y} = g^{yx_1+y} g^{-x_2-y} = h^{x_1} g^{-x_2}$$

- This means that classical public-key cryptography algorithms can be broken by quantum computers

- Symmetric key algorithms are less severely affected
  - Grover's algorithm reduces work from 2^k to $2^{k/2}$ for k-bit keys
  - this means that post-quantum 256-bit AES has the strength of 128-bit AES

Post-Quantum Computing

- In the post-quantum world, we would need to use alternative algorithms for public-key cryptography
  - this includes public-key encryption, signatures, etc.
- Two prominent directions are
  - lattice-based cryptography
  - code-based cryptography

Lattice-Based Cryptography

- Lattices are discrete subgroups of $\mathbb{R}^n$
  - a subset of $\Lambda \subset \mathbb{R}^n$ is called discrete if for every point $v \in \Lambda$, v is the only point in the environment of radius $\epsilon > 0$ around it
  - a discrete subgroup of $\mathbb{R}^n$ is called a lattice
  - all nontrivial lattices are infinite sets, but they have a finite basis
  - in cryptographic constructions we normally use integers instead of real numbers

Lattice-Based Cryptography

- Examples of hard lattice-based problems used in cryptography
  - closest vector problem (CVP): given a target vector $w \in \mathbb{R}^n$, find the closest lattice point $v \in \Lambda$ to w
  - learning with errors (LWE): solving a random system of noisy linear equations modulo an integer
    - note that solving a system of linear equations is easy
- Examples of cryptosystems include
  - GGH (1997) public-key encryption and signature schemes
  - NTRU (1998) public-key encryption scheme that uses polynomials in the ring $\mathbb{Z}[x]/(x^N - 1)$
  - many recent somewhat and fully homomorphic encryption schemes

Code-Based Cryptography

- Codes play an important role in error detection and error correction when sending data over noisy channels
- For cryptographic applications, one can use very long codes with a secret structure
  - Goppa codes are an example of suitable linear codes
- There are similarities between lattice-based and code-based cryptography
  - both are linear subspaces of high-dimensional spaces and finding the closest vector to the target vector in the subspace can be hard
- McEliece and Niederreiter cryptosystems are promising candidates for post-quantum cryptography

Conclusions

- Many public-key cryptographic algorithms will lose their security in a post-quantum world
- Cryptographic techniques resilient to quantum computing cryptanalysis are an active area of research
  - lattice-based cryptography has particularly experienced a lot of progress