Introduction to Cryptography

Ayelet Yablon Daniela Yablon

May 22, 2022

Abstract

Cryptography is a cornerstone of modern communication systems, and is crucial to ensure security and privacy. In this paper, we describe two important encryption schemes — the RSA (Rivest–Shamir–Adleman) and Diffie Hellman encryption schemes. As quantum computers become more powerful, there is a very real possibility that these encryption systems will no longer remain secure, due to Shor's algorithm, developed by Peter Shor in 1994. We also describe how Shor's algorithm, using properties of quantum computers, can attack the RSA encryption scheme.

Contents

1 Introduction

In this expository paper, we discuss two important encryption schemes – the RSA (Rivest–Shamir–Adleman) encryption scheme in Section 4, and the Diffie Hellman encryption scheme in Section 5. In Section 6 we describe how Shor's algorithm attacks the RSA encryption scheme. The background knowledge and preliminaries are detailed in Section 2. Our main reference for this paper is [Aum17].

2 Preliminaries

2.1 Randomness

Randomness is key for secure encryption algorithms. Without random numbers, attackers can find the patterns in the numbers chosen for encryption and easily break the algorithm. However, random numbers are difficult to obtain, so computer scientists have developed technology called pseudorandom generators.

Definition 1 (Pseudorandom generator)**.** *A pseudorandom generator is a generator that generates seemingly random numbers given a small amount of truly random numbers.*

The pseudorandom generator takes in a small amount of truly random numbers, called a seed, and based off of that seed it produces a lot of seemingly random numbers. Encryption algorithms often need random numbers, so cryptographers use pseudorandom generators to create those random numbers.

2.2 Asymptotic Notation

Theoretical computer scientists analyze efficiency of algorithms in terms of the Big O notation, which captures the asymptotic growth of a function.

Example 1. *If an algorithm takes* $2n^3 + 5n^2 + 9n - 4$*, the algorithm's efficiency written in asymptotic notation would be* $O(n^3)$*. That is to say, Big O Notation ignores all coefficients of the terms, and only produces the fastest-growing term. In our case,* $2n^3$ *was the fastest growing term. It's a bit counter-intuitive to only take the fastest growing term, but since it's the* fastest growing term*, as* n *goes to infinity,* n^3 *grows much faster than* n^2*.*

Example 2. *Let's say that an algorithm takes* $2^{\log_2 n} + 2^{\sqrt{n}}$*. We can simplify this equation to* $n + 2^{\sqrt{n}}$ *to see that the fastest growing term is* $2^{\sqrt{n}}$*. Thus, we would write the efficiency of this algorithm in Big O Notation as* $O(2^{\sqrt{n}})$*.*

Example 3. *Imagine an algorithm that takes* $2 + 1/n^2$ *time to run. The fastest growing term here is actually* 2*. Thus, Big O Notation would be written as* $O(2)$*. However, when it's just integer without any coefficient we just write it as* $O(1)$*. Thus, the time it takes for this algorithm to run would just be* $O(1)$*.*

2.3 Introduction to Complexity Theory

Here, we briefly describe the importance of complexity theory in cryptography and define important complexity classes. The field of cryptography relies on the computational hardness of certain problems (factoring for RSA and discrete logarithms for Diffie–Hellman) to ensure security of encryption system against realistic adversaries that are limited in the amount of time and computational resources they have. Complexity theory, which studies the computational resources required to solve different problems, is therefore a foundational aspect of cryptography. For example, the problem finding the shortest path between two vertices in a graph might be easily solvable, but for more difficult problems like finding satisfying assignments to boolean formulas (SAT), we do not know how to solve them in a reasonable amount of time. Complexity theory tries to classify problems based on how difficult they are, or how difficult we believe them to be. It achieves this by defining complexity classes — sets of problems that need different amounts of computational resources,

relative to the size of the input of the problem. We use n to denote the size (number of bits) of the input for the problem.

Definition 2 (TIME)**.** $\mathsf{TIME}(f(n))$ *is the class of the solvable computation problems the time* $f(n)$*.*

Algorithms that are efficiently solvable are usually characterized by a polynomial runtime.

Definition 3 (P)**.** *This is the class of problems solvable in polynomial time. Formally, it is the union of all* $\mathsf{TIME}(O(n^k))$ *where* $k \in \mathbb{N}$*.*

Definition 4 (SPACE)**.** $\mathsf{SPACE}(f(n))$ *is the class of the solvable computation using* $f(n)$ *bits of memory.*

Definition 5 (PSPACE)**.** *This is the class of problems that is solvable using polynomial memory. This is the union of the types of* $\mathsf{SPACE}(n^k)$ *problems, where* $k \in \mathbb{N}$*.*

Definition 6 (NP)**.** *Also known as nondeterministic polynomial time, this is the class of problems verifiable in polynomial time.*

3 Introduction to Group Theory and Number Theory

In this section, we give the background knowledge in group theory and number theory required to understand the encryption algorithms we talk about later.

Definition 7 (Group)**.** *A set of elements that respects a binary operation with the following properties*

- Associativity - given three real numbers a, b, and c, $a \star (b \star c) = (a \star b) \star c$
- Closure - $a \in G$ and $b \in G$, $a \star b \in G$
- Identity Existence - given group G, there exists element $i \in G$ such that $a \star i = a$
- Inverse Existence - given group G, there exists element $e \in G$ such that $a \star e = i$

To be a group, it must satisfy these four *group axioms*. In addition, groups may be classified in other ways too; a *cyclic* group means that a group contains the powers of a single element g mod p and a *commutative* group is a group where two elements in the group x and y satisfy $x * y = y * x$.

Suppose that p is prime. The group $\mathbb{Z}_p^*$ is defined as the set of positive integers from 1 to $p-1$ mod some number, with the operation being multiplication mod p. For example, the group $\mathbb{Z}_5^*$ is the set $1, 2, 3, 4$.

Two numbers a and b are congruent to each other $\mod c$ if they leave the same remainder after dividing by c.

Here are a series of examples that show the modulo operator:

$$17 \equiv 3 \mod 7$$

$$22 \equiv 5 \mod 17$$

$$27 \equiv 6 \mod 21$$

$$39 \equiv 7 \mod 32$$

3.1 Euler's Totient Function

The group $\mathbb{Z}_n^*$ is the group that contains all the integers from 1 to n that are *coprime* with n, the operation being multiplication mod n. The size of the group is denoted by the totient function,

$$\varphi(n) = |\mathbb{Z}_n^*| = |\{a : 1 \leq a \leq n, \mathsf{gcd}(a, n) = 1\}|.$$

Here $\mathsf{gcd}(a, n)$ denotes the greatest common divisor of a and n. If $n = p$ is prime, then $\mathbb{Z}_p^*$ will contain the set of integers from 1 to $p-1$. However, if n is not prime, then not all integers from 1 to $n-1$ will be coprime to n. Integers with non-trivial common divisors with n will not have inverse mod n, violating a group axiom. For example, consider $\mathbb{Z}_{15}^*$, which contains the elements $\{1, 2, 4, 7, 8, 11, 13, 14\}$. We can derive this by first looking at all the numbers in the set $\{1, 2, \ldots, 15\}$ and crossing off the elements that do not have inverses. The elements that do not have inverses are the numbers in which the greatest common divisor of 15 and that number is not one. Thus, all the multiples of 5 and 3 are not in this set. The number of multiples of 3, ignoring 15 is 4, and the number of multiples of 5, ingoring 15 again is 2. There is 1 multiple of 15 that will also not have an inverse. So we can write that the number of elements in this set is $15 - 4 - 2 - 1 = 8$. In other words, this is the total minus the multiples of 3 in the set excluding 15 which is 4, minus the multiples of 5 excluding 15 which is 2, minus multiples of 15 which in our set is only 1.

We can generalize this method for a set of any length. If we have a set $\mathbb{Z}_{pq}^*$ where p and q are primes, the length of the set is $pq-(p-1)-(q-1)-1$ (we followed this form in the example above: $3{\cdot}5-(3-1)-(5-1)-1 = 8$). This simplifies to $pq-p-q+1 = (p-1)(q-1)$. Thus the number of elements in $\mathbb{Z}_n^*$ where n is a product of two primes p, q is

$$\varphi(n) = \varphi(pq) = (p-1)(q-1).$$

Thus, Euler's Totient Theorem states that if $n = p_1 p_2 \ldots p_k$ is a product of k primes, the size of the group $\mathbb{Z}_n^*$ is $\varphi(n) = \varphi(p_1 \cdot p_2 \cdot \ldots \cdot p_k) = (p_1 - 1)(p_2 - 1) \cdot (p_k - 1)$.

3.2 Euler's Totient Theorem

Theorem 1 (Euler's Theorem)**.** *For two positive integers x, n, such that x, n are relatively prime*

$$x^{\varphi(n)} \equiv 1 \mod n.$$

Proof. To begin to understand how this works, we can introduce a set of integers: $A = \{a_1, a_2, \ldots, a_\varphi\}$ to be the elements a_i such that $\mathsf{gcd}(a_i, n) = 1$. We know there are $\varphi(n)$ number of elements in the set because of Euler's Totient Function. Multiplying each element in A by x, which is coprime with n, yields the set $B = \{xa_1, xa_2, \ldots xa_\varphi\}$. We can $\mod$ each value by n to get a modified version of set B. So $C = \{xa_1 \pmod n, xa_2 \pmod n, \ldots, xa_\varphi \pmod n\}$.

We know that applying the modulus function will not result in any of these elements being equal to 0, because if $x \cdot a_i \equiv 0 \mod n$, that would imply that $a_i \equiv 0 \mod n$. But this cannot be true since we know that the elements of A have to come from the set $\{1, 2, \ldots, n-1\}$. This means that the elements in C also come from the set $\{1, 2, \ldots, n-1\}$.

The elements in set A are all the numbers that are relatively prime with n, $\mod n$, and these elements are not congruent to each other, $\mod n$. We know this also applies to set C becuase the elements of set C are all the elements relatively prime to n and are not congruent to each other $\mod n$. So each element in C is in A, and each element in A is in C, thus $A = C$.

We know that for every element in C, there is an element in B that is congruent to this element $\mod n$. In other terms,

$$xa_1 \equiv xa_1 \pmod n \mod n^{1}.$$

And we had just stated that every element in C is in A, so that means that there is an element in A that is congruent to this element $\mod n$, or

$$a_{i_1} \equiv xa_1 \mod n.$$

This is true for every element in A, where that element exists somewhere in set B.

We can multiply all the the elements in A to get product u, and we can multiply all the elements in B to get products v. We know that

$$v \equiv u \mod n$$

because each element in A is equal to one of the elements in B, and that applies to all elements in A. We can take a closer look at v. We know that v is the product of all the elements of B which is the product of all the elements of A and x. We know that there are $\varphi(n)$ number of elements in A, so we are multiplying x by itself $\varphi(n)$ number of times. We can rewrite v as

$$v = u \cdot x^{\varphi(n)}.$$

Substituting this back into our first equation yields

$$u \cdot x^{\varphi(n)} = u \mod n.$$

Dividing both sides by u gives us

$$x^{\varphi(n)} \equiv 1 \mod n.$$

□

3.3 Fermat's Little Theorem

Fermat's little theorem is a special case of Euler's Totient Theorem when n is a prime number. We can label this number as p. We already know that $\varphi(p) = p - 1$.

Theorem 2 (Fermat's Little Theorem)**.** *For any integer x such that p does not divide x,*

$$x^{p-1} \equiv 1 \mod p$$

3.4 Lagranges Theorem

Definition 8 (Order of a Group)**.** *The order of group A is number of elements in the group, denoted with $|A|$*

Definition 9 (Order of an Element)**.** *Given a group G, if $g \in G$, then the order of g also denoted by ord(g) is the smallest number x that satisfies $g^x = e$ (where e is the identity element of the group).*

Definition 10 (Subgroup)**.** *A subgroup is part of a group and all of the elements in a subgroup are in the group. Both groups are defined under the same operations. This is denoted with the symbol $\trianglelefteq$.*

This theorem is named after Joseph-Louis Lagrange, an Italian astronomer and mathematician. In a group G, there are two basic subgroups: itself and the identity element, often denoted as e. But how do you know if there are any more subgroups?

Theorem 3 (Lagrange's Theorem)**.** *If J is a subgroup of G, denoted by $J \trianglelefteq G$, then the order of J is a divisor of the order of G.*

[1]Note that the first element in A might not be equal to the first element in C

Proof. Let G be a finite group and have $|G| = x$. We know that the identity element, denoted as e, is a subgroup as well as G. This is because $|e| = 1$, and 1 divides of x, and $|G| = x$, and x divides x. Additionally, let J be a subgroup of G. This means that J will also contain the same identity element, e, or 1. Next, we find an element g_1 that is an element in G, but not an element in J. We can then multiply every element in J by g_1 to get g_1J. This is called a 'left coset' because the elements of J are being multiplied by g_1 on the left. Additionally, it is important to understand that J and g_1J have no elements in common; they do not overlap.

The reason why is because of the proof that follows: let's assume that there is an element in both J and g_1J. This would mean that

$$g_1 \cdot j_a = j_b,$$

where $j_a, j_b \in J$. Multiplying this equation by j_a^{-1}, the inverse of j_a gives us

$$g_1 \cdot j_a \cdot j_a^{-1} = j_b \cdot j_a^{-1}.$$

Following the inverse existence axiom defined above, this simplifies to

$$g_1 \cdot e = j_b \cdot j_a^{-1}.$$

We know the identity element is 1, so the right hand side is just g_1. Also, because j_a and $j_b \in J$, that means that their product, or g_1, is also in J. But this doesn't make sense, because we chose g_1 as an element that is not in group J. Thus, the intersection of J and g_1 is $\varnothing$.

To continue our proof, we can pick another element $g_2 \in G$ that is not in J or g_1J. The left coset of the set is g_2J, or every $j \in J$ multiplied by g_2. Using the same argument as above, the intersection of the subgroup J and the set g_2J is $\varnothing$. In addition, g_1J and g_2J do not overlap either.

This is as follows: we can start by assuming that there is an element that the sets g_1J and g_2J share. This would imply that $g_1 \cdot j_c = g_1 \cdot j_d$, for some $j_c, j_d \in J$. Multiplying both sides by the inverse of j_d computes to

$$g_1 \cdot j_c \cdot j_d^{-1} = g_2 \cdot j_d \cdot j_d^{-1}.$$

However, we know that the product of two inverses of a group is e, or 1. So now we have

$$g_1 \cdot j_c \cdot j_d^{-1} = g_2 \cdot 1.$$

We also know that the product of j_c and j_d^{-1} is in J because of the closure axiom. We can call this product j_x ($j_x \in J$). Substituting this in is

$$g_1 \cdot j_x = g_2.$$

But this doesn't make sense. If this equation were true, that would mean that $g_2 \in g_1J$. However, we specifically chose a value for g_2 that is not in this coset. Thus, g_2 is not in g_1J.

We can continue this process until we reach the point where there are no elements left to be in a coset. The order of these cosets are equal to the order of the subgroup J. We can understand this easier if we assume the left coset gJ has a duplicate. This means that for some $j_m, j_n \in J$,

$$g \cdot j_m = g \cdot j_n.$$

When we multiply both sides by g^{-1}, we get

$$g^{-1} \cdot g \cdot j_m = g^{-1} \cdot g \ \cdot j_n.$$

Just like above,

$$g^{-1} \cdot g = e = 1.$$

Now we have

$$j_n = j_m.$$

But these to values are different, thus proving that the orders of the cosets are equal to each other.

This results in G being split into subgroup J and sets $g_1J, g_2J, \ldots, g_nJ$. The reason why J is a subgroup and the rest are sets is because J contains the identity element, and the other sets do not. Containing the identity element satisfies the group axiom of identity existence.

We can call the number of cosets a group has as w, which is also the index. So the index of J in $G = w = |G : J|$. So we have

$$w \cdot |J| = |G|,$$

or the order of J divides the order of G. □

Example 4. *Let A be a group with $|A| = 403$. The factors of 403 are 13 and 31. According to Lagranges theorem, the only possible subgroups of 403 have orders of 403, the identity element which is 1, 13, and 31. This doesn't necessarily mean that there is a subgroup of order 13, the theorem just yields the possibilities of the order of a subgroup.*

3.5 Linear Algebra

In this section, we give a brief introduction to basic linear algebra that comes in handy in understanding basic quantum mechanics and Shor's algorithm, which we describe in Section 6.

Definition 11 (Matrix)**.** *A matrix is an array of numbers organized into columns and rows, and allows for computation in linear algebra.*

Definition 12 (Vector)**.** *A vector is a quantity with both magnitude and direction. A vector is a matrix with one column and a number of rows.*

Adding Matrices To add two matrices of equal dimensions, you add the corresponding parts to each other and the sum goes into the corresponding spot of the new matrix. For example, $\begin{Bmatrix} 1 & 2 \\ 3 & 4 \end{Bmatrix} + \begin{Bmatrix} 5 & 6 \\ 7 & 8 \end{Bmatrix} = \begin{Bmatrix} 1+5 & 2+6 \\ 3+7 & 4+8 \end{Bmatrix} = \begin{Bmatrix} 6 & 8 \\ 10 & 12 \end{Bmatrix}$.

Multiplying by a Scale Factor To multiply a matrix of a scale factor, you multiply each number in the matrix by the scale factor, and the product goes in the corresponding spot. As an example, $3 \cdot \begin{Bmatrix} 1 & 2 \\ 3 & 4 \end{Bmatrix} = \begin{Bmatrix} 1 \cdot 3 & 2 \cdot 3 \\ 3 \cdot 3 & 4 \cdot 3 \end{Bmatrix} = \begin{Bmatrix} 3 & 6 \\ 9 & 12 \end{Bmatrix}$.

This can be used to **subtract** matrices by multiplying one matrix by the scale factor "-1", and then adding the two matrices.

Matrix-Vector Multiplication To multiply a matrix, the number of columns of the first matrix needs to be equal to the number of rows of the second matrix. You multiply the column of the first matrix by the first row of the second matrix, the second column of the first matrix by the second row of the second matrix, and continue this way, with their product aligning to the corresponding spot of the new matrix. So, $\begin{Bmatrix} 3 & 6 \\ 9 & 12 \end{Bmatrix} \cdot \begin{Bmatrix} 5 \\ 10 \end{Bmatrix} = \begin{Bmatrix} 3 \cdot 5 + 6 \cdot 10 \\ 9 \cdot 5 + 12 \cdot 10 \end{Bmatrix} = \begin{Bmatrix} 75 \\ 165 \end{Bmatrix}$.

A specific kind of matrix multiplication is matrix-vector multiplication, which is a vector multiplied by a matrix to yield a new vector.

Definition 13 (Linear Independence)**.** *A set of vectors $\{v_1, v_2, \ldots, v_n\}$ is said to be linearly independent if the following is true: If $a_1v_1 + a_2v_2 + \ldots + a_nv_n = 0$ for scalars a_i, then that implies $a_1 = a_2 = \ldots = a_n = 0$.*

Example 5. *Let* $A = \begin{Bmatrix} 1 & 2 \\ 1 & 0 \end{Bmatrix}$. *Are the vectors* $\begin{Bmatrix} 1 \\ 2 \end{Bmatrix}, \begin{Bmatrix} 1 \\ 0 \end{Bmatrix}$, *and* $\begin{Bmatrix} 1 \\ 2 \end{Bmatrix}$, *independent? We can write this a different way: is there such values for* α_1, α_2, *and* α_1 *that satisfy* $\alpha_1 \cdot \begin{Bmatrix} 1 \\ 2 \end{Bmatrix} + \alpha_2 \cdot \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} + \alpha_3 \cdot \begin{Bmatrix} 1 \\ 2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$, *where* α_1, α_2, *and* $\alpha_3 \neq 0$*(but are real numbers)? The answer is no, so these vectors are independent. Conversely, if the answer to this question was yes, that would mean that the vectors are dependent.*

Definition 14 (Column Span)**.** *Given a matrix of any size, with the columns labeled* $v_1, v_2, \ldots, v_n$, *the column span of this matrix is the set of all vectors that can be written in the form* $a_1 \cdot v_1 + a_2 \cdot v_2 + \ldots + a_n \cdot v_n$, *where* $a_1, a_2, \ldots, a_n = \mathbb{R}$.

The column span of independent 2-dimensional vectors is the whole plane and is infinite, and the column span of dependent 2-dimensional vectors is a line, which is also infinite.

Definition 15 (Null Space)**.** *For a given matrix* B, *the null space of that matrix is the set of all the vectors* v *that satisfy* $Bv = 0$.

Identity Matrix An identity matrix is an m by m matrix of all zeroes, except for the diagonal of the top left corner to the bottom right, which is just ones. Let's call it A in our example, and let N be a matrix of m rows and k columns. This means that $A \cdot N = N$, for any m and k in the set of real numbers.

4 RSA

Many encryption algorithms have been developed, but have also been solved. It is crucial for cryptography to have encryption methods which are difficult to solve, or impossible to solve. Most encryption algorithms have a key between two people. That is to say, if Person A wants to communicate with Person B, and Person A wants to communicate with Person C, Person A would have two separate keys: a key with Person B and a key with Person C.

This can become extremely inefficient if Person A wants to communicate with multiple people. Person A would then have to generate individual keys for every single person. This posed a question for many cryptographers: Is there an algorithm where Person A only has one key that everyone can use?

Cryptographers then developed a clever technique: public key cryptography.

Definition 16 (Public Key Cryptography)**.** *Rather than using one key per person, public key cryptography is a pair of keys, one that is public and one that is private. The public key is accessible to the whole public, and used for encryption. The private key is only accessible to the person receiving the message, and is used for decryption.*

Another important technique cryptographers developed was trapdoor functions.

Definition 17 (Trapdoor Function)**.** *A trapdoor function is a function in which it is easy to go in one direction, but harder to go the other direction. In other words, it's easy to go from* x *to* y, *but hard to go from* y *to* x.

An example of a trapdoor function is the modulus function, which was previously defined in Intro to Group Theory.

4.1 Key Generation

RSA uses both public key cryptography, and trapdoor functions to work. In the RSA algorithm, two people, Alice and Bob, wish to communicate. Here is how Bob would go about encrypting his message.

Alice establishes her key, consisting of a private key only she knows, and a public key everyone else knows. We can call her private key d. Her public key consists of two numbers, which we can call e and N.

e represents the exponent that Bob has to raise his message to, while N is the modulus. N is achieved by multiplying two large primes together. To establish the keys, e and d must be inverses of each other in the group $\varphi(N)$.

4.2 Encryption Algorithm

Bob takes his message, which we can call x, raises it to the eth power, then takes the remainder when divided by N like so:

$$x^e \mod N$$

We can call this number y. Bob sends y, the encrypted message, to Alice.

4.3 Decryption Algorithm

Alice receives Bob's message, which we can call y. She raises it to the dth power and takes the remainder when dividing by N:

$$y^d \mod N$$

This then gives Alice Bob's original message, x.

4.4 Proof of Correctness

Even though Bob publicly announces y, and e and N are public, without Alice's secret key d, it is really hard to find what x originally was, because of the discrete log problem.

But e and d need to are deliberately chosen so that they are inverses of each other in mod $\varphi(n)$ so that Euler's Theorem can be used. Recall that Euler's Theorem states:

$$a^{\varphi(N)} \equiv 1 \mod N$$

Mathematically writing the fact that e, d are inverses of each other in group $\varphi(N)$ looks like:

$$ed \equiv 1 \mod \varphi(N)$$

We can rewrite this equation to obtain (where k is an integer):

$$ed = k \cdot \varphi(N) + 1$$

Back to Alice and Bob, if Alice raises y^d, we can substitute $y = x^e$ to get $(x^e)^d = 1 \mod n$. Using the Power Rule we can rewrite this as:

$$y^d \equiv x^{ed} \mod n$$

We can substitute $ed = k \cdot \varphi(N) + 1$ to get:

$$x^{k \cdot \varphi(N)+1}$$

Applying the Product Rule gives us

$$x^{k \cdot \varphi(N)} \cdot x^1 \mod N$$

Since $x^1 = x$, we now have:

$$x^{k \cdot \varphi(N)} \cdot x \mod N$$

Earlier we proved that $x^{\varphi(N)} = 1 \mod N$, so substituting we get:

$$1 \cdot x \mod n$$

Simplifying results in:

$$x \mod N$$

Thus, Alice can achieve Bob's original message with her secret key, since raising $y^d = x$.

4.5 Proof of Security

It is extremely hard to discover Bob's encrypted message without Alice's private key d. The only way to find d is to know $\varphi(N)$ However, to find $\varphi(N)$, then you would need to know the factors of N, since $\varphi(N) = (p-1)(q-1)$, assuming N has two factors. The way to find the factors of N, one would need to factor N. Therefore, this algorithm is so secure as long as there's no efficient way to factor N.

5 Diffie Hellman

Diffie-Hellman is an algorithm to generate a shared secret. It is named after two inventors, Whitfield Diffie and Martin Hellman, who published this idea of using a private key and a corresponding public key in 1976. This encryption algorithm was a major breakthrough in the history of cryptography as it was one of the first algorithm to implement public key cryptography.

5.1 How It Works

Let's say two people, Alex and Ben, want to generate information that only they have access to. How would they go about that? To start off, let Alex have a secret key a and Ben have secret key b. This means that only Alex has access to a, and only Ben has access to b. Additionally, these keys must be chosen from $\mathbb{Z}_p^*$, for some prime number p. Alex and Ben then do similar operations to their private key; they raise a public generatr, g, the their secret-key's power. So that means that

$$A = g^a \mod p$$

$$B = g^b \mod p$$

After completing this, Alex sends A to Ben, and Ben sends B to Alex, publicly. Alex and Ben do similar computations again:

$$B^a = (g^b)^a = g^{ab} \mod p$$

$$A^b = (g^a)^b = g^{ab} \mod p$$

Now Alex and Ben ended with the same result, g^{ab}, so they have a shared secret that only they can view.

The reason why this works is because of the **Discrete Logarithm Problem**. The Discrete Logarithm Problem states that given the numbers a, b, and p, it is easier and takes less time to calculate the value of $a^b \mod p$, and given a, a^b, and p, it's harder to calculate the value of b in $a^b \mod p$.

6 Shor's Algorithm

If quantum computers were to ever be invented, mathematicians have already developed algorithms that utilize the strengths of those computers to break modern-day encryption algorithms, specifically RSA. Mathematician Peter Shor developed an algorithm, Shor's Algorithm, that can break the encyprtion algorithm of RSA through quantum computers. Recall that the security of RSA relies on the computational hardness of the factoring problem. That is to say, given the RSA modulus $N = pq$, if one can factor N, one breaks the security of the encryption algorithm. Indeed, Shor's algorithm attempts to factor N. Here, we describe the high-level idea of Shor's algorithm.

Shor's algorithm works by reducing the problem of factoring to the problem of order finding. Specifically, it uses the fact that if we can reliably and efficiently find the period of the function $f(x) = g^x \pmod N$ where $g \in \mathbb{Z}_N^*$, then we can factor N into its two prime factors. Concretely, it takes the following steps.

1. Pick a random number $g \in \mathbb{Z}_N^*$. This can be done by picking random integers $1 < g < N$ until the picked number g is coprime with N. That is, $\gcd(g, N) = 1$.

2. Find the smallest r such that $g^r \equiv 1 \pmod{N}$ using the quantum Fourier transform.

3. If r is even and if $g^{r/2} \not\equiv 1 \pmod{N}$, then compute $g^{r/2} \pm 1$ by repeated squaring and using Euclid's algorithm, find $\gcd(g^{r/2} + 1, N)$ and $\gcd(g^{r/2} - 1, N)$.

Step 2 is where we leverage a quantum algorithm called the quantum Fourier transform to efficiently find the period of the function $f(x) = g^x \pmod{N}$. We will not get into the details of the quantum part of the algorithm, however we will explain the basics of the (classical) reduction from factoring to order-finding.

Suppose that $g \in \mathbb{Z}_N^*$ and $r = \mathsf{ord}(g)$, which means that r is the smallest positive integer such that $g^r \equiv 1 \pmod{N}$. Equivalently, there is some integer m such that

$$g^r - 1 = m \cdot N.$$

Now, if r is even, then we can factor the left-hand side of the equation using the formula for the difference of squares,

$$(g^{r/2} + 1)(g^{r/2} - 1) = m \cdot N.$$

Now, if $N = pq$ does not divide either of $g^{r/2} \pm 1$, then we know that p has to divide one of $g^{r/2} \pm 1$ and q divides the other. Without loss of generality, suppose that p divides $g^{r/2} + 1$ and q divides $g^{r/2} - 1$. Since N divides neither, q does not divide $g^{r/2}+1$ and p does not divide $g^{r/2}-1$. This means that $\gcd(g^{r/2}+1, N) = p$ and $\gcd(g^{r/2} - 1, N) = q$. Thus computing the GCDs in Step 3 gives the factorization of N. Finally, we note that the "bad" cases of choosing g such that its order r is odd, or N divides one of $g^{r/2} \pm 1$ happens only with a small constant probability. As a result, repeating the experiment with several random picks of g gives a polynomial-time quantum algorithm that succeeds reasonably well.

7 Acknowledgements.

We would like to thank Aparna for helping us learn cryptography and answering all our questions! She was a great mentor and we were so lucky to be able to work with her!

Also a thanks to Mary and Marisa for running this program and providing this opportunity for us!

References

[Aum17] Jean-Philippe Aumasson. *Serious cryptography: a practical introduction to modern encryption.* No Starch Press, 2017. 3