PRIMES
Introduction to Cryptography

Liubov Slesarenko and Chloe Zhong

Spring 2023

Abstract

In this paper, we will research the fundamental rules that cryptography is based on to better understand the mathematics standing behind it. Then we will describe a foundational algorithm in cryptography: the Diffie-Hellman key exchange protocol. This algorithm enables public-key encryption, which means two parties can communicate privately over a channel without ever having to meet to share a secret key. We will build up to this cryptosystem by first talking about secret-key encryption and introducing the necessary tools in number theory, group theory, and complexity theory.

Contents

1 Euclid's algorithm

The importance of this algorithm for our research is that it is the key to unlocking many of the deeper properties of natural numbers. These properties are interesting in themselves and pivotal in appreciating the applications of number theory to cryptography.

1.1 The Division Algorithm

If $a,\ b \in \mathbb{Z}, b > 0$, then there exist unique $q, r \in \mathbb{Z}$ such that $a = qb + r,\ 0 \leq r < b$. Here q is called the *quotient* of the integer division of a by b, and r is called the *remainder.*

Given two integers $a,\ b,\ b \neq 0$, we say that b divides a, written $b \mid a$, if there is some integer q such that $a = bq$: $b \mid a \leftrightarrow \exists q,\ a = bq$. We also say that b divides or is a divisor of a, or that a is a multiple of b.

There are a number of elementary **divisibility properties**, some of which we list in the following proposition. Let $a,\ b,\ c \in \mathbb{Z}$ be integers.
(a) If $a \mid b$ and $b \mid c$, then $a \mid c$.
(b) If $a \mid b$ and $b \mid a$, then $a = b$.
(c) If $a \mid b$ and $a \mid c$, then $a \mid b + c$ and $a \mid b - c$.

A positive integer d is called a *common divisor* of the integers a and b if d divides a and b. The greatest possible such d is called the *greatest common divisor* of a and b, denoted $\gcd(a, b)$. If $\gcd(a, b) = 1$ then a, b are called *relatively prime.*

Example 1. *The set of positive divisors of 12 and 30 is 1, 2, 3, 6. The greatest common divisor of* 12 *and* 30 *is* $\gcd(12,\ 30) = 6$.

1.2 The proof of the Euclidean Algorithm

Now we examine an alternative method to compute the *gcd* of two given positive integers a, b. It is based on the following fact: given two integers $a \geq 0$ and $b > 0$, and $r = a \bmod b$, then $\gcd(a,\ b) = \gcd(b,\ r)$.

The Euclidean algorithm is as follows. First, we divide a by b, obtaining quotient q and a remainder r. Then we divide b by r, obtaining a new quotient q_1 and a reminder c_1 . Next, we divide r by c_1 , which gives a quotient q_2 and another remainder c_2. We continue dividing each reminder by the next one until obtaining a zero reminder, at which point we stop. The last non-zero remainder is the *gcd.*

Proof: The Euclidean algorithm consists of a sequence of divisions with remainder Let $d = gcd(a,\ b)$, and assume it takes three steps to finish the algorithm, then

$$\begin{aligned} a &= bq_1 + c_1 \\ b &= c_1 q_2 + c_2 \\ c_1 &= c_2 q_3 + 0 \end{aligned} \tag{1}$$

So we know $d \mid c_2$, WTS that $c_2 \mid b$:
Let's replace c_1 by c_2q_3 into the second equation, then we have $b = c_2(q_2q_3 + 1)$. We can see that $c_2 \mid b$. Repeat the same with the first line, so that $a = c_2(q_1q_2q_3 + q_1 + q_3)$, and $c_2 \mid a$. Therefore, $c_2 \mid a$ and $c_2 \mid b$, which means that $c_2 \mid d$.

After proving the Euclidean algorithm, we illustrate it with an example.

Example 2. *Assume that we wish to compute* $gcd(500,\ 222)$. *Then we arrange the computations in the following way:*

$$\begin{aligned}500 &= 2 \cdot 222 + 56 \to c_0 = 56\\ 222 &= 3 \cdot 56 + 54 \to c_1 = 54\\ 56 &= 1 \cdot 54 + 2 \to c_2 = 2\\ 54 &= 27 \cdot 2 + 0 \to c_3 = 0\end{aligned} \tag{2}$$

The last non-zero remainder is $c_2 = 2$, hence $gcd(500, 222) = 2$.

1.3 The fast powering algorithm

The fast powering algorithm, also known as exponentiation by squaring, is an efficient algorithm for computing large powers of a number. It works by repeatedly squaring the base and reducing the exponent by half until the exponent becomes 0.

Algorithm:

Given a base b and an exponent n, the fast powering algorithm is as follows:

1. If the exponent n is 0, return 1.
2. If the exponent n is odd, return b times the result of the fast powering algorithm with the base b, the exponent $n-1$, and halved (i.e. compute $b \cdot (b^{(n-1)/2})^2$).
3. If the exponent n is even, return the result of the fast powering algorithm with the base b^2, the exponent $n/2$, and halved (i.e. compute $(b^2)^{n/2}$).

Explanation:

The fast powering algorithm takes advantage of the fact that b^n can be expressed as $(b^2)^{n/2}$ for even n, and $b \cdot b^{n-1}$ for odd n. By repeatedly squaring the base and halving the exponent, we can reduce the number of multiplications needed to compute b^n. For example, to compute b^{13}, we can do the following:

1. Compute b^2, which is b times b, and set n to 6.
2. Compute $(b^2)^3$, which is $(b^2)^2$ times b^2, and set n to 3.
3. Compute $b \cdot (b^2)^2$, which is b times $(b^2)^2$, and set n to 1.
4. Compute $b \cdot (b^2)^2 \cdot b$, which is b times $b \cdot (b^2)^2$, which gives us b^{13}.

Using the fast powering algorithm, we were able to compute b^{13} with only 3 multiplications, rather than the 12 multiplications that would be required using the naive algorithm.

Examples:

Here are some examples of using the fast powering algorithm to compute large powers of a number:

- Compute 2^{20}: Using the fast powering algorithm, we can compute 2^{20} as follows: $2^{20} = (2^2)^{10} = 4^{10} = ((4^2)^2)^2 = 16^4 = ((16^2)^2)^1 = 65536$.
- Compute 3^{25}: Using the fast powering algorithm, we can compute 3^{25} as follows: $3^{25} = 3 \cdot (3^2)^{12} = 3 \cdot (9^6)^2 = 3 \cdot (531441)^2 = 3 \cdot ((531441^2)^1) = 847288609443$.

As we can see, the fast powering algorithm allows us to efficiently compute large powers of a number using only a small number of multiplications.

1.4 Fermat's little theorem

Theorem 1. *Let p be a prime number and let a be any integer. Then* $a^{p-1} \equiv \begin{cases} 1(\mod p), if p \nmid a \\ 0(\mod p), if p \mid a \end{cases}$

Proof. If $p \mid a$, then it is clear that every power of a is divisible by p. And it means that $a^{p-1} \equiv 0(\mod p)$. So we only need to consider the case that $p \nmid a$. To prove this theorem let's show that $x^{p-1} \equiv 1(\mod p)$, $x \in \mathbb{Z}/\{0\}$.

The list x, $2x$, $3x$, ..., $(p-1)x$ contains $p-1$ numbers, and clearly, none of them are divisible by p. Suppose that we take two numbers jx and kx in this list, and suppose that they happen to be congruent $jx = kx \pmod p$.
Then $p \mid (j-k)x$, so that $p \mid (j-k)$, since we are assuming that p does not divide x. We know that if a prime divides a product then it divides one of the factors. On the other hand, we know that $1 \leq j, k \leq p-1$, so $\mid j-k \mid < (p-1)$. There is only one number with an absolute value less than $p-1$ that is divisible by p and that number is zero. Hence, $j-k=0$. This shows that different multiples in the list x, $2x$, $3x$, ..., $(p-1)x$ are distinct modulo p.

So we now know that the list x, $2x$, $3x$, ..., $(p-1)x$ contains $p-1$ distinct nonzero values modulo p. But there are only $p-1$ distinct nonzero values modulo p. Hence, the list x, $2x$, $3x$, ..., $(p-1)x$ and the list 1, 2, 3, ..., $(p-1)$ must contain the same numbers modulo p.

This means that

$$\begin{aligned} x, 2x, 3x, \ldots, (p-1)x = 1, 2, 3, \ldots, (p-1) \pmod p \\ x^{p-1}(p-1)! \equiv (p-1)! \pmod p \\ x^{p-1} \equiv 1 \pmod p \end{aligned} \tag{3}$$

□

1.5 The Fundamental Theorem of Arithmetic

Theorem 2. *Every positive integer greater than 1 can be uniquely expressed as a product of primes.*

Proof. We prove the theorem by induction. Let n be a positive integer greater than 1.

Base case: If n is a prime number, then it can be expressed as a product of primes in a unique way (itself).

Inductive step: Suppose that every positive integer less than n can be expressed as a product of primes in a unique way. We need to show that n can also be expressed as a product of primes.

If n is prime, then we are done. Otherwise, there exist positive integers a and b such that $n = ab$, $1 < a \leq b < n$. By the induction hypothesis, a and b can each be expressed as a product of primes in a unique way. Therefore, $n = ab$ can also be expressed as a product of primes in a unique way.

To prove uniqueness, suppose that $n = p_1 p_2 \cdots p_r = q_1 q_2 \cdots q_s$ are two different factorizations of n into primes. Without loss of generality, we may assume that $p_1 \leq p_2 \leq \cdots \leq p_r$ and $q_1 \leq q_2 \leq \cdots \leq q_s$. Since p_1 divides n and n is a product of primes, p_1 must be equal to one of the q_i. Similarly, p_2 must be equal to one of the remaining q_j, and so on. Since the p_i's and q_i's are in increasing order, the two factorizations must be identical after a suitable reordering. Therefore, the factorization is unique. □

Proof of the Fundamental Theorem of Arithmetic

Proof. Suppose that an integer n has two prime factorizations:

$$n = p_1^{a_1} p_2^{a_2} \cdots p_m^{a_m} = q_1^{b_1} q_2^{b_2} \cdots q_k^{b_k}, \tag{4}$$

where $p_1, p_2, \ldots, p_m, q_1, q_2, \ldots, q_k$ are all prime numbers, and $a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_k$ are all positive integers.

Without loss of generality, assume that $p_1 \leq q_1$. Then p_1 divides the left-hand side of the equation, so it must also divide the right-hand side. Since q_1 is prime, either $p_1 = q_1$ or p_1 divides one of the other q_i's.

If $p_1 = q_1$, then we can divide both sides of the equation by $p_1^{a_1}$ and $q_1^{b_1}$, and we are left with:

$$p_2^{a_2} \cdots p_m^{a_m} = q_2^{b_2} \cdots q_k^{b_k}. \tag{5}$$

By repeating this argument, we can cancel all of the common prime factors from both sides of the equation, and we are left with $1 = 1$. Thus, the two prime factorizations are identical. If p_1 divides one of the q_i's, say q_j, then we can divide both sides of the equation by $p_1^{a_1}$ and $q_j^{b_j}$, and we are left with:

$$p_2^{a_2} \cdots p_m^{a_m} q_1^{b_1} \cdots q_j^{\hat{b}_j} \cdots q_k^{b_k} = q_1^{b_1} \cdots q_j^{\hat{b}_j} \cdots q_k^{b_k}, \tag{6}$$

where the hat over $q_j^{b_j}$ indicates that it has been removed from the product. Again, by repeating this argument, we can cancel all of the common prime factors from both sides of the equation, and we are left with $1 = 1$. Thus, the two prime factorizations are identical. Therefore, any integer $n > 1$ can be uniquely expressed as a product of primes, up to the order in which the prime factors appear. $\square$

2 Discrete Logarithms and Diffie–Hellman

2.1 Diffie-Hellman key exchange

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.

1. **Setup:** Alice and Bob agree on a prime number p and a base g such that g is a primitive root modulo p. These values are made public.

2. **Alice's Private Key:** Alice chooses a secret integer a, and calculates $A = g^a \bmod p$. She sends A to Bob and Charlie.

3. **Bob's Private Key:** Bob chooses a secret integer b, and calculates $B = g^b \bmod p$. He sends B to Alice and Charlie.

4. **Charlie's Interception:** Charlie intercepts the values A and B sent by Alice and Bob. However, he cannot compute the shared secret key s_{AB} without knowledge of either Alice's or Bob's secret key a or b, due to the discrete logarithm problem.

5. **Shared Secret:** Alice computes the shared secret key with Bob as $s_{AB} = B^a \bmod p$. Bob computes the shared secret key with Alice as $s_{BA} = A^b \bmod p$. Since g is a primitive root modulo p, both Alice and Bob will arrive at the same value for s_{AB} and s_{BA}, which they can use as a symmetric encryption key.

2.2 The discrete logarithm problem

Let g be a primitive root for F_p and let h be a nonzero element of F_p. The Discrete Logarithm Problem (DLP) is the problem of finding an exponent x such that

$$g^x \equiv h(\mod p) \tag{7}$$

The number x is called the discrete logarithm of h to the base g and is denoted by $log_g(h)$.

The discrete logarithm problem is a well-posed problem, namely to find an integer exponent x such that $g^x = h$. However, if there is one solution, then there are infinitely many because Fermat's little theorem tells us that $g^{p-1} \equiv 1(\mod p)$. Hence if x is a solution to $g^x = h$, then $x + k(p-1)$ is also a solution for every value of k, because

$$g^{x+k(p-1)} = g^x (g^{p-1})^k \equiv h \cdot 1^k \equiv h(\mod p) \tag{8}$$

3 An Overview of the Theory of Group

Definition. A group consists of a set G and a rule, which we denote by $\star$, for combining two elements $a, b \in G$ to obtain an element $a \star b \in G$. The composition operation $\star$ is required to have the following three properties:

- **Identity Law:**
  There is an $e \in G$ such that $e \star a = a \star e = a$ for every $a \in G$.
- **Inverse Law:**
  For every $a \in G$ there is a (unique) $a^{-1} \in G$ satisfying $a \star a^{-1} = a^{-1} \star a = e$.
- **Associative Law:**
  $a \star (b \star c) = (a \star b) \star c$ for all a, b, $c \in G$.

If G has finitely many elements, we say that G is a *finite group*. The *order* of G is the number of elements in G; it is denoted by $|G|$ or $\#G$.

Example 3. *Here are a few examples of groups:*

1. *$G = \mathbb{Z}$ and $\star$ = addition. The identity element is $e = 0$ and the inverse of a is $-a$. This group G is an infinite group.*
2. *$G = \mathbb{R}^*$ and $\star$ = multiplication is a group since all elements have multiplicative inverses inside $\mathbb{R}^*$.*
3. *However, $G = \mathbb{Z}$ and $\star$ = multiplication is not a group, since most elements do not have multiplicative inverses inside $\mathbb{Z}$.*

3.1 Lagrange's Theorem

Let G be a finite group. Then every element of G has finite order. Further, if $a \in G$ has order d and if $a^k = e$, then $d|k$. Using this we can prove the following proposition, which is also called Lagrange's Theorem.

Theorem 3. *Let G be a finite group and let $a \in G$. Then the order of a divides the order G. More precisely, let $n = |G|$ be the order of G and let d be the order of a, i.e., a^d is the smallest positive power of a that is equal to e. Then $a^n = e$ and $d|n$.*

Proof. Since G is finite, we can list its elements as $G = \{g_1, g_2, \dots, g_n\}$.

We now multiply each element of G by a to obtain a new set, which we call S_a:

$$S_a = \{a * g_1, a * g_2, \dots, a * g_n\} \tag{9}$$

We claim that the elements of S_a are distinct. To see this, suppose that $a * g_i = a * g_j$. Multiplying both sides by a^{-1} yields $g_i = g_j$. Thus, S_a contains n distinct elements, which is the same as the number of elements of G. Therefore, $S_a = G$, so if we multiply together all of the elements of S_a, we get the same answer as multiplying together all of the elements of G. (Note that we are using the assumption that G is commutative.) Thus,

$$(a * g_1) * (a * g_2) * \ldots * (a * g_n) = g_1 * g_2 * \ldots * g_n \tag{10}$$

We can rearrange the order of the product on the left-hand side (again using the commutativity) to obtain

$$a^n * g_1 * g_2 * \ldots * g_n = g_1 * g_2 * \ldots * g_n \tag{11}$$

Now multiplying by $(g_1 * g_2 * \ldots * g_n)^{-1}$ yields $a^n = e$.
Now, multiplying by $(g_1 \cdot g_2 \cdot \ldots \cdot g_n)^{-1}$ on both sides of the equation $a \cdot g_1 \cdot g_2 \cdot \ldots \cdot g_n = g_1 \cdot g_2 \cdot \ldots \cdot g_n$ yields:

$$(a \cdot g_1 \cdot g_2 \cdot \ldots \cdot g_n) \cdot (g_1 \cdot g_2 \cdot \ldots \cdot g_n)^{-1} = (g_1 \cdot g_2 \cdot \ldots \cdot g_n) \cdot (g_1 \cdot g_2 \cdot \ldots \cdot g_n)^{-1} \tag{12}$$

Simplifying both sides using the properties of inverses, we have:

$$a \cdot (g_1 \cdot g_2 \cdot \ldots \cdot g_n) \cdot (g_1 \cdot g_2 \cdot \ldots \cdot g_n)^{-1} = e \tag{13}$$

Since $(g_1 \cdot g_2 \cdot \ldots \cdot g_n) \cdot (g_1 \cdot g_2 \cdot \ldots \cdot g_n)^{-1} = e$ (the inverse of a product is the product of inverses), we can rewrite the equation as:

$$a \cdot e = e \tag{14}$$

Therefore, we have $a = e$. This shows that $a^n = e$ for some positive integer n, which means that a has finite order.

Moreover, since $a^k = e$ for some positive integer k, by Lagrange's theorem, we know that the order of a, denoted as d, divides n. Therefore, d divides the order of the group G. Hence, the order of a divides the order of G, as required.
Hence, Lagrange's Theorem is proved. □

4 Acknowledgements

We want to express our gratitude to our advisor Aparna, for her invaluable help and support throughout our investigation into cryptography. Your guidance and expertise have been instrumental in our understanding and analysis of mathematical concepts and in improving our knowledge of cryptography, and we greatly appreciate all your time and dedication.
We would also like to extend many thanks to the organizers Mary and Marisa for providing us with the wonderful opportunity to explore math through the PRIMES Circle program. The resources, guidance, and encouragement provided by this program have been invaluable to our growth as math enthusiasts, and we are grateful for the chance to be a part of this community.
Thank you again, Aparna, Mary, and Marisa, for all that you have done for us. We look forward to continuing our journey in math and hope to collaborate with you in the future.

References

[1] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman. *An Introduction to Mathematical Cryptography. Undergraduate Texts in Mathematics,* 2014.