Cryptography

Samuel Henshaw

October 20, 2020

1 Introduction

Cryptography is a bipartite subject. It consists of the creation of methods of encoding and their testing and exploitation. The former is called cryptology and the latter is cryptanalysis. This paper is primarily about cryptology. In it, we will go over some primitive encryption methods, see what makes a given cryptosystem suitable for application, learn how to implement the RSA system, go over some basic integer factoring methods, see the Diffie-Hellman key exchange, and observe a simple elliptic curve based group used in many cryptosystems.

2 Early cryptography

The oldest cryptographic method is the Caesar cipher, so named because it was used by Julius and Augustus to protect notes containing information about the affairs of the state. To encypt a message, one assigns to the letters A through Z values 0 through 25, respectively. One adds a constant value to each term, then reduces modulo 26. This string of numbers is then translated to letters by the original assignment.

Here is an example. Suppose the message to be encrypted is

$$IVE\ SEEN\ THINGS\ YOU\ PEOPLE\ WOULDNT\ BELIEVE$$

dropping the punctuation marks. We assign 0 to A, 1 to B, 2 to C, and so forth. Replacing the letters in this string by their images and dropping the spaces, we get the string

$$8\ 21\ 4\ 18\ 4\ 4\ 13\ 19\ 7\ 8\ 13\ 6\ 18\ 24\ 14\ 20\ 15\ 4\ 14\ 15\ 11\ 4\ 22\ 14\ 20\ 11\ 3\ 13\ 19\ 1\ 4\ 11\ 8\ 4\ 21\ 4$$

Let's shift each value to the right by 7 (notice that left shifts correspond to negative constants.) Applying the shift and reducing modulo 26, we get the string

$$15\ 2\ 11\ 25\ 11\ 11\ 20\ 0\ 14\ 15\ 20\ 13\ 25\ 5\ 21\ 1\ 22\ 11\ 21\ 22\ 18\ 11\ 3\ 21\ 1\ 18\ 10\ 20\ 0\ 8\ 11\ 18\ 15\ 11\ 2\ 11$$

Reassigning these numbers to letters with the inverse of our original function, we get the final message

$$PCLZLLUAOPUNZFVBWLVWSLDVBSKUAILSPLCL$$

Another ancient form of encryption is the use of the scytale. This is a rod around which a strip of paper with an encrypted message is wrapped. Both this and the Caesar cipher can be generalized as permutation ciphers: they replace each letter with another letter according to a fixed permutation of the Latin alphabet. Using statistical analysis, such ciphers are easily cracked. This is a problem had by all early cryptosystems, as they are all just permutation-based.

3 Basics

A cryptosystem is a 5-tuple (P, C, K, E, D). P is the set of things that can be encrypted; it is called the plaintext space. C is the set of images of plaintexts under encryption; this is the ciphertext space. E is the set of encryptions, which are functions from P to C. Likewise, D is the set of decryptions: functions in the opposite direction. K is the set of keys, which are used in encryptions and decryptions in various ways depending on the cryptosystem.

For example, consider the Caesar cipher. P here is not the set of all messages with the Latin alphabet. It is the alphabet itself, a set of order 26. This is because we apply the encryption to each individual letter rather than applying it to the whole message as an indivisible object. This is called a monoalphabetic cipher. Likewise, C is the Latin alphabet. E and D are both the set of shifts modulo 26 parameterized by K, which we may identify with $\mathbb{Z}_{26}$.

The totient ϕ is important to a cryptosystem that will be examined in this paper. Euler's famous function ϕ counts for a positive integer n its number of coprime residues, or, equivalently, the number of units in the ring $\mathbb{Z}_n$. It is easiest to compute ϕ for simpler numbers first, then to build to more complex ones. For prime p, there are no nontrivial divisors and thus no non-coprime residues, so

$$\phi(p) = p - 1$$

Now, for a prime power p^n, the only residues that are not coprime are the multiples of p. There are $p^n/p - 1$ of them strictly less than n, so

$$\phi(p^n) = p^n - 1 - (p^{n-1} - 1) = p^{n-1}(p - 1)$$

Finally, for a number with prime factorization $\prod_k p_i^{n_i}$, any coprime residue is coprime to all the prime power factors $p_i^{n_i}$ and any k-tuple of residues each coprime to its respective $p_i^{n_i}$ gives a system of k linear congruences that can be solved for exactly one value modulo $\prod_k p_i^{n_i}$. We have injections between the finite sets $\mathbb{Z}_{\prod_k p_i^{n_i}}$ and $\prod_k \mathbb{Z}_{p_i^{n_i}}$, meaning there is a bijection between them. We thus have that

$$\phi(\prod_k p_i^{n_i}) = \prod_k \phi(p_i^{n_i}) = \prod_k (p_i^{n_i-1}(p_i - 1))$$

ϕ is known as a multiplicative function.

Encryption is a digital process, and information is able to be encrypted because it is stored as a string of bits. This string is taken to be a binary number or multiple binary numbers.

4 Good and bad cryptosystems

Why is the Caesar cipher bad? For starters, there are only 26 keys. One could just try each one. The ciphertext space is also very small. Since it is monoalphabetic, one could guess the plaintext letter to which each ciphertext letter corresponds by using the fact that e is the most common letter, a the next, and so forth. The fact that the ciphertext space is so small enables us to exhaust it in a relatively small amount of time.

What makes a good cryptosytem? The space of keys should be large so that they can't be guessed one-by-one. It should not be monoalphabetic. One should not be able to guess the plaintext of any given ciphertext, and one should not be able to guess the appropriate decryption function of any ciphertext.

5 The RSA system

The RSA cryptosystem is one of the most well-known modern cryptosystems. It is a public key cryptosystem, meaning that it uses one key known to the public to encrypt messages and another that is kept private to decrypt them. Predictably, these are known as public and private keys.

Suppose we have a message A to encrypt. It shall be represented as a residue in $\mathbb{Z}_m$ for some integer m that is a product of two very large primes a and b. Since these primes are known to you, you may factor m and calculate $\phi(m)$. Pick a new large prime e modulo m; it shall be the public key. Perform the Euclidean algorithm on $\phi(m)$ and e; at the end, you have the relation $de + j\phi(m) = 1$, which means that d and e are multiplicative inverses modulo $\phi(m)$. d is designated as the private key.

When e and d are multiplicative inverses modulo $\phi(m)$, we have that, for any residue b of m, $b^{ed} = b^{k\phi(m)+1} = b^{k\phi(m)}b = b \pmod{m}$ by Lagrange's theorem: the order of a subgroup divides the order $\phi(m)$ of the group $\mathbb{Z}_m^\times$ and thus any element raised to the power $\phi(m)$ is the identity.

E and D are the encryption and decryption functions. The process is as below.

$$A \xrightarrow{E} A^e \pmod{m} \xrightarrow{D} A^{ed} \pmod{m} = A \pmod{m}$$

Here is how it is applied. Suppose Bob is to send Alice a message. He keeps his public key (e, m) in the open so that anyone can use it. Alice encrypts the message with his public encrpyption key, then sends it over a public channel. Only Bob can decrypt it, for only Bob knows the private key (d, m).

It is safe to make known to the public the values e and m. To compute d from those two values as we did, it was necessary to factor m so that $\phi(m)$ could be computed, and there is no known efficient algorithm to factor arbitrary positive integers. The integers chosen for this scheme are so large that it would take modern computers thousands of year at the quickest to factor them. Another relevant problem is known simply as the RSA problem. This is to compute a decryption key from the public encryption key. It is slightly different from the problem of factorization in that it doesn't specifically say that this decryption must be done by computing $\phi(m)$ first. Like the problem of factorization, though, no modern computer can solve it efficiently by known methods. A quantum computer, though, could quickly factor a positive integer with Shor's algorithm, so a hot area of cryptographic research right now is that of post-quantum cryptography: how to make cryptosystems that a quantum computer couldn't crack.

Another application of the RSA system is in digital signatures. It is like the usual application, but in reverse order. If Alice wants to prove that she is the one sending the message A to Bob, she first encrypts it with her private key d_a. She then sends it to Bob. The only way for this message to be made intelligible is to use her public key e_a, so Bob applies it to decode the message. This can be applied in conjunction with encryption with Bob's keys to get a series (where all is done modulo m)

$$A \xrightarrow{D_a} A^{d_a} \xrightarrow{E_b} A^{d_a e_b} \xrightarrow{D_b} A^{d_a} \xrightarrow{E_a} A$$

Note that these operations are commutative, so Bob can apply D_b and E_a in either order once he has the message.

6 Integer factorization

Integer factorization is often hard, but there are instances when it is not. When choosing a number for RSA, one has to be careful; as we are about to see, there are simple factorization methods for integers of specified forms.

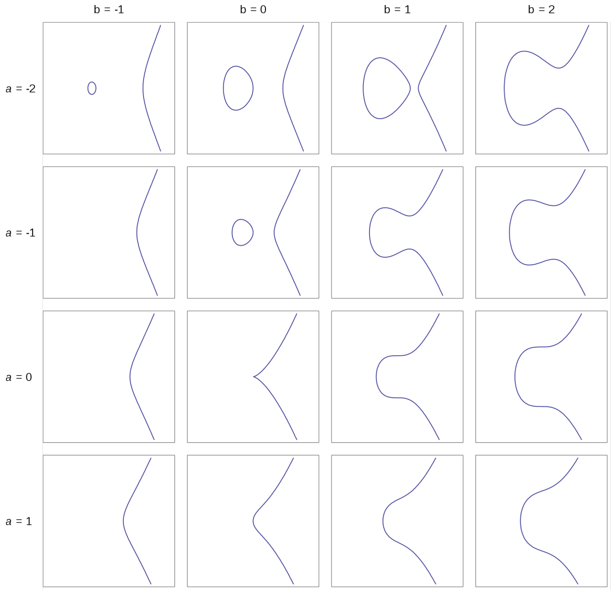
The first such method is Fermat factorization. If we can write an integer k as $a^2 - b^2$ for two positive integers a and b, we have that $k = (a+b)(a-b)$. This is a guess-and-check method; one starts by guessing $a = \lceil \sqrt{k} \rceil$ and computing $b^2 = a^2 - k$. If b^2 is not a perfect square, a is augmented by 1, and the process repeats. If k has a factor that is close in size to its algebraic square root, this method will find it quickly.

A similar method is the difference of squares method. If one can find different residues x and y modulo n such that $x^2 \equiv y^2$, then as before we have $(x-y)(x+y) \equiv 0$. One then computes the greatest common divisors of the pairs $x - y, n$ and $x + y, n$, which may yield a nontrivial factor of n. It is, however, often difficult to find such x and y.

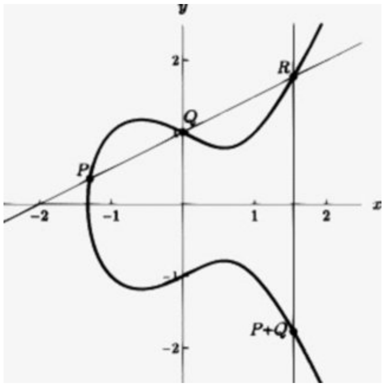
The simplest and worst way to factor a number is simply by trial and error. One just tries to divide it by smaller numbers until something works or all the numbers get used without success. This method may be improved by only trying prime numbers less than the positive square root of the given number. For every factor above the square root, there must be one below it, and trying nonprimes is superfluous. However, even with these improvements, this factorization method runs in exponential time. It is still too slow. It is often used in conjunction with Fermat factorization to create a process more efficient than either one alone, but even this is too slow.

7 Elliptic curves

An elliptic curve is the set of solutions of a polynomial equation of the form $y^2 = x^3 + ax + b$ over a field. Their shapes vary as seen below.



A group law can be defined on the curves on the right half and the curves with the unconnected nodes, i. e. nonsingular curves. The elements of the group are the points on the curve and the law is as follows: to combine two points P and Q, draw a straight line through them. It will intersect the elliptic curve at another point. The product $P + Q$ is the point mirrored below the x-axis of this intersection. Here is a picture.



To add a point to itself, one uses the tangent line at that point. The identity is taken as infinity, corresponding to a line parallel to the y-axis. This entire process can be generalized to arbitrary fields; for cryptography, it is applied on finite fields. Many elliptic curves over finite fields have been standardized for use in cryptography; a famous example is the curve $y^2 = x^3 + 7$ over the field of order $2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$. It is known as secp256k1, and it is used in Bitcoin's public key system.

8 Diffie-Hellman key exchange

The Diffie-Hellman key exchange is used to create a secret key over a public medium. Here is how it works: Alice and Bob agree on a number m. It can be made public. Each of them picks a positive value to keep secret; call these values a and b. Bob computes m^b and Alice computes m^a, and they send them to each other. Each then computes $m^{ba} = m^{ab}$.

$$m \longrightarrow m^b \longrightarrow m^{ba} = m^{ab} \longleftarrow m^a \longleftarrow m$$

This secure because it exploits another computationally difficult problem known as the discrete logarithm problem: given an arbitrary finite group and elements x, y, it is hard to find a number k such that $x^k = y$. Over elliptic curves, this problem is extremely difficult; it is even harder than it is over the groups $\mathbb{Z}_i^\times$.

9 Sources

https://www.researchgate.net/profile/Wim_Van_Dam/publication/23552588/figure/fig1/AS:269352175927296@1441229971343/

https://prateekvjoshi.com/2015/02/07/why-are-they-called-elliptic-curves/

http://www.ccn.com/wp-content/uploads/2014/10/elliptic_curve_commmons.png

https://en.bitcoin.it/wiki/Secp256k1

I took a cryptography class abroad last semester.