Cryptography: A brief overview

Supreme Sharma

March 2021

Introduction

Cryptography, or cryptology, is the practice and study of techniques for secure communication in the presence of third parties called adversaries. Cryptography has a very prominent position in our modern world. But secure communication between humans is not the only place where we see it's application. From securing financial transaction by encrypting between machines to producing new types of currency such as cryptocurrency, cryptography is part of ever modern humans lives. There are also efforts to encrypt DNA [18] for the privacy conscious individuals which is different from using DNA for creating encryption system [17].

History

Figure 1: Reconstructed Scytala [14]

"It is believed that the oldest known text to contain one of the essential components of cryptography, a modification of the text, occurred some 4000 years ago in the Egyptian town of MENET KHUFU where the hieroglyphic inscriptions on the tomb of the nobleman KHNUMHOTEP II were written with a number of unusual symbols to confuse or obscure the meaning of the inscriptions." [16]. There were also several other systems like the Spartian scytala used to send secret messages.

David Kahn notes in The Codebreakers that modern cryptology originated among the Arabs, the first people to systematically document cryptanalytic methods.Al-Khalil (717–786) wrote the Book of Cryptographic Messages, which contains the first use of permutations and combinations to list all possible Arabic words with and without vowels.[13]

Caesar shit cypher is one of the simpler and fairly well known encryption techniques. It is name after Julius Caesar. [20]. This technique is still used in ROT13.[13]

Another example of a cypher system is the Enigma used by the Germans during World War II that wrecked havoc by providing a secure method of communication for them for a long time. The cypher system was complex [12] enough that a simple brute force effort was not enough to crack this system and required the invention of sophisticated machinery. Although a lot of theoretical groundwork was done by several Polish mathematicians [19] , the final machine was constructed in USA with the help of Alan Turin et al [2].

Example 1

Definition. *A cryptographic system (or cypher system) is a model that allows for obfuscation of data (encryption) such that it can only be un-obfuscated (de-*

cryption) in presence of certain information about the system.[11]

Definition. *Encryption is a method of taking plain-text (data we don't want certain people to see) and turning it into cypher-text (data from which the original information is hard to extract without special information about the system such as a key.)*

Encryption can also be thought of as a function whose domain is plain-text and whose codomain is a set of symbols as seen in figure 2.

$$E : W \to S$$

where E is the encryption function, W a set of words (it could be numbers, letters or even symbols), and S a set of symbols.

Definition. *Decryption is a method of of getting plain-text back from a cypher-text.*

Definition. *In symmetric-key cryptography, a key is the shared knowladge between parties and is required to encrypt and decrypt messages. The keys needed to encrypt or decrypt might be different or same knowladge.*

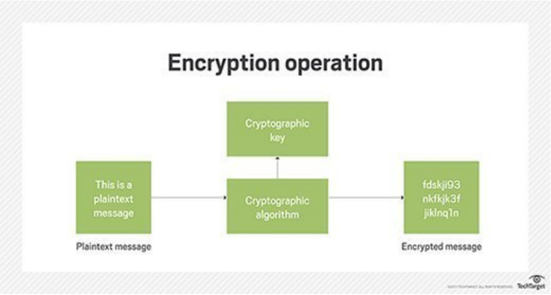

Figure 2: Encryption

The process generally requires some knowledge about the system for example a decryption key. Decryption can be thought of as the inverse to the encryption function. But this is not necessarily true as a bijective E actually makes an encryption system weaker to adviserial attacks.

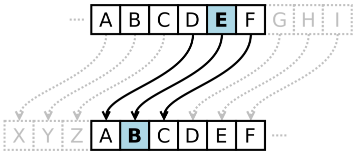

Figure 3: Shift cypher [14]

An example of a crypto-system [11] called Caesar shift can be constructed using set of English alphabets P as the domain and codomain C of a shift function E that does the encryption.

$$E_n : P \to C$$

where $n \in \{0, .., 25\}$ is the distance of the shift. For example $n = 1$ means each alphabet gets shifted by one, i.e a is replaced by b, b by c and so on.

$$E_1(a) = b; E_1(hello) = ifmmp$$

In this system, E is a bijective function hence E^{-1} is a function that shifts the alphabet backward.

$$E_n^{-1} : C \to P$$

This is a simple crypto-system that it can be done by hand on a piece of paper. But it can also be done with a computer by encoding the alphabets in a way that we can add and substract shifts. For example, using ASCII encoding standard.[**?**]

Algebra

Definition. *A group is a set G together with a binary operation*

$$\circ : G \times G \to G$$

that obeys the following axioms.

- *Closure: If $a, b \in G$ then $a \circ b \in G$*
- *Associativity: $(a \circ b) \circ c = a \circ (b \circ c)$*
- *Identity: There exists an element $e \in G$ such that $e \circ a = a \circ e = a$ for all $a \in G$*
- *Inverse: For all $a \in G$ exists $a^{-1} \in G$ such that $a \circ a^{-1} = a^{-1} \circ a = e$*

Definition. *A symmetric group S_n of degree n is the group of all permutation on n symbols.*

We see that when we consider all possible shifts of alphabets in our shift cypher, there is an underlying group structure of permutations. With this knowledge, it is possible to add arbitrary symbols to the domain of the function and create different variations in shift cypher. Algebraically, we are dealing with modular arithmetic in a $Z/26Z$

Definition. *Fix a group G and an element g. The Discrete Logarithm Problem (DLP) for G is : Given an element h in the subgroup generated by g, find an integer m satisfying*

$$h = g^m$$

The smallest integer m satisfying this condition is called the logarithm (or index) of h with respect to g, and is denoted

$$m = log_g(h)$$

The DLP is used as the underlying hard problem in many cryptographic constructions, including key exchange, encryption, digital signature and hash functions.

Definition. *An elliptical curve is a smooth, projective, algebriac curve of genus one which there is a specified point O.*

Informally, an elliptic curve is the set of points satisfying

$$y^2 = x^3 + ax + b$$

Definition. *The group on an elleptic curve which has been transformed to form*

$$y^2 = x^3 + ax + b$$

is a set of K-rational points, including the single point at infinity. The group law (addition) is defined as follows: Take 2 K-rational point P and Q. Now 'draw' a straight line through them and compute the third point of intersection R (also a K-rational point). Then

$$P + Q + R = 0$$

gives the identity point at infinity. Now find the inverse of R, which can be done by setting $R = (a, b)$ giving $-R = (a, -b)$. [10]

Definition. *Let G and H be groups. A homomorphism is a structure preserving map $\phi : G \to H$ such that:*

$$\phi(g_1 \circ g_2) = \phi(g_1) \circ \phi(g_2)$$

i.e the group operation is preserved.

Definition. *A trapdoor function[9] is a collection of one-way functions $f_k : D_k \to R_k (k \in K)$, in which all of K,D_k,R_k are subsets of binary string $0, 1^*$, satisfying the following conditions:*

- *There exists a probabilistic polynomial time (PPT) sampling algorithm Gen such that $Gen(1^n) = (k, t_k)$ with $k \in K \cap 0, 1^n$ and $t_k \in 0, 1^*$ satisfies $|t_k| < p(n)$, in which p is some polynomial. Each t_k is called the trapdoor corresponding to k. Each trapdoor can be efficiently sampled.*
- *Given input k, there also exists a PPT algorithm that outputs $x \in D_k$. That is, each D_k can be efficiently sampled.*
- *For any $k \in K$, there exists a PPT algorith that correctly computes f_k.*
- *For any $k \in K$, there exists a PPT algorithm A such that for any $x \in D_k$, let $y = A * k, f_k(x), t_k)$, and then we have $f_k(y) = f_k(x)$. That is, given trapdoor, it is easy to invert.*
- *For any $k \in K$, without trapdoor t_k, for any PPT algorithm, the probability to correctly invert f_k is negligible.*

Definition. *The fundamental theorem of arithmetic, also called unique-prime-factorization-theorem, states that every integer greater than 1 is either a prime number or can be represented as the unique product of primes up to reordering.*

Example 2

The RSA algorithm [15] is named after Ron Rivest, Adi Shamir and Len Adleman who invented it in 1977. It relies on a public key and requires no shared secret to encrypt a message to send to a recipient who can then use their private key to decrypt the cypher text to plain text. The key generation algorithm is as follows:

- Generate two large primes, p and q, of approximately equal size such that their product $n = pq$ is of the required lenth, e.g. 1024 bits.
- Compute $n = pq$ and $\phi = (p-1)(q-1)$.
- Choose an integer $e, 1 < e < \phi$, such that $gcd(e, \phi) = 1$
- Compute the secret exponent d, $1 < d < \phi$, such that $ed \equiv 1 \text{mod } \phi$.
- The public key is (n, e) and the private key is (d, p, q). Keep all the values d, p, q, ϕ secret.

n is known as the modulus. e is known as the public exponent and d is known as the secret exponent.

Sender A does the following:

- Obtains the recipient B's public key (n, e).
- Represent the plaintext message as a positive integer m with $1 < m < n$.
- Compute the ciphertext $c = m^e \text{ mod } n$.
- Sends the ciphertext c to B.

Recipient B does the following:

- Uses his private key (n, d) to compute $m = c^d \text{ mod } n$.
- Extracts the plaintext from the message representative m.

Example 3

Diffie-Hellman key exchange establishes a shared secret between two parties that is used for secret communication over an insecure communications channel. The original implementation of the protocol uses the multiplicative group of integers G modulo p where p is a prime and $g \in G$ is a primitive root module p. Any choice of value from 1 to $p-1$ is valid for non secret values. In the following algorithm, only a, b and s are secrets.

- Alice and Bob publicly agree to use modulus $p = 23$ and base $g = 5$.
- Alice chooses a secret integer $a = 4$, then sends Bob $A = g^a \text{ mod } p$.

$$A = 5^4 \, mod \, 23 = 4$$

- Bob chooses a secret integer $b = 3$, then sends Alice $B = g^b \text{ mod } p$.

$$B = 5^3 \, mod \, 23 = 10$$

- Alice computes $s = B^1 \text{ mod } p$.

$$s = 10^4 \, mod \, 23 = 18$$

- Bob computes $s = A^b \bmod p$.

$$s = 4^3\, mod\, 23 = 18$$

- Alice and Bob now share a secret (the number 18).

Both Alice and Bob have arrived at the same values under mod p,

$$A^b\, mod\, p = g^{ab}\, mod\, p = g^{ba}\, mod\, p = B^a\, mod\, p$$

More specifically,

$$(g^a\ mod\ p)^b\ mod\ p = (g^b\ mod\ p)^a\ mod\ p$$

Even if p is 600 digits, modern classical algorithm struggles to figure out the secret. This is due to the discrete logarithm problem.[6]

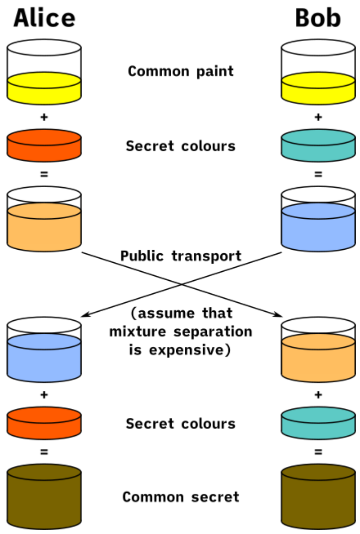

Figure 4: Diffie-Hellman key exchange.
[1]

Example 4

A lot of cryptography systems rely on the fact that prime factorization is a generally hard problem and that there is no known classical algorithm for it not including brute force methods. Tjat said, factoring is not the hardest problem on a bit for bit basis. Specialized algorithms like Quadratic Sieve and the General Number Field Sieve were created to tackle the problem of prime factorization and has been moderately successful. These factoring algorithm get more efficient as the size of the number being factored get larger.

As such, research explored other mathematics-based cryptographic solutions looking for algorithm beyond factoring that would serve as a good Trapdoor function. In 1985, cryptographic algorithms were purposed based on an esoteric branch of mathematics called elliptic curve. For example, if the domain of elliptic curve over real place R^2 is restricted to integers, the necessary requirement for RSA is met. Or if you take elliptic curves over finite field, you get finite elliptic group and you get finite elliptic curve cryptography based on the discrete logarithm on that group. This would give an elliptic Diffie-Hellman cryptography.

The elliptic curve discrete logarithm is the hard problem underpinning elliptic curve cryptography. Despite almost three decades of research, mathematicians still haven't found an algorithm to solve this problem beyond the naive approach.

Probably the most famous use case of this system is the Tor project. But it also appears in digital signature verification and even proof of ownership for Bitcoins. [21]

Example 5

Homomorphic Encryption (HE) refers to a special type of encryption technique that allows for computation to be done on encrypted data, without requiring access to a secret (decryption) key. The results of the computations are encrypted and can be revealed only by the owner of the secret key.[4]

The idea is that there is a group homomorphism between plain text and cypher-text so that you can work with the cypher-text group directly.

If the RSA public has modulus n and encryption exponent e, then the encryption of a message m given by $\phi(m) = m^e \bmod n$. The homomorphic property is then:

$$\begin{aligned}\phi(m_1) \circ \phi(m_2) &= m_1^e m_2^e \, mod \, n \\ &= (m_1 m_2)^e \, mod \, n \\ &= \phi(m_1 \circ m_2)\end{aligned}$$

This is sometimes called unpaded RSA. [8]

Quantum Computers and Cryptography

In 1994, an American mathematician named Peter Shore came up with an algorithm for then mostly theoretical quantum computers that has been reshaping cryptography. Shor's algorithm is a polynomial-time quantum computer algorithm for integer factorization. Informally, it solves the following problem: Given an integer N, find it's prime factorization. This number N can be factor in $O((logN)3)$ time and $O(logN)$ space.[3]

The algorithm is significant because it implies that public key cryptography might be easily broken, given a sufficiently large quantum computer. RSA, for example, uses a public key N which is the product of two large prime numbers. Like all quantum algorithm, Shor's algorithm is probabilistic: it gives the correct answer with high probability, and the probability of faliusre can be decreased by repeating the algorithm.

Shor's algorithm was demonstrated in 2001 by a group at IBM, which factored 15 into 3 and 5, using a quantum computer with 7 qbits.

Shor's algorithm consists of two parts:

- A reduction of the factoring problem to the problem of order-finding, which can be done on a classical computer.
- A quantum algorithm to solve the order finding problem.

The algorithm and the proof for it can be found here.shor and also on Wikipedia.

Shor is not the only algorithm out there that might break some crypto-systems either. Grover's algorithm is another such algorithm that can attack a crypto-system in a different way. Grover's algorithm could brute-force a $128-bit$ symmetric cryptography key in roughly 2^64 iterations.[7]

Suppose you have a function $f : 0, 1^n \to 0, 1^n$ and $x \in 0, 1^n$ and we are looking for a y such that $f(y) = x$. On a classical computer, unless we know something special about f, we need to iterate over all possible inputs to find such a y such that $f(y) = x$. So, if $N = |0, 1^n$, on a classical computer, you need to do $O(N)$ operations. On the other hand, if out f is given as a quantum circuit and we can run our code on a quantum computer, we only need $O(\sqrt{(}N))$ operations to find such a y.

For example, supposed we have a known plaintext problem where we have a message P, and encrypted message S and we are looking for a key K such that $f(P, K) = S$. On a classical computer, a brute force algorithm requires $O(N)$ steps where N is the size of the key space. While on a quantum computer, you only need $O(N^{1/2})$ steps. So if you use AES with 128-bit key, Grover's algorithm can break it in $O(2^6 4)$ steps so we only get half as many bits of security as we thought we had.

But it's not all bad news when it comes to quantum computer. Quantum mechanics (namely Quantum Field Theory) itself can be used to produce cryptographic systems. Quantum key distribution (QKD) uses a series of photons (light particles) to transmit data from one location to another over a fiber optic cable. By comparing measurements of the properties of a fraction of these photons, the two endpoints can determine what the key is and if it is safe to use.

- The sender transmits photons through a filter (or polarize) which randomly gives them one of four possible polarizations and bit designations: Vertical(One bit), Horizontal(Zero bit), 45 degree right(One bit), or 45 degree left(Zero bit).

- The photons travel to a receiver, which uses two beam splitters (horizontal/vertical and diagonal) to "read" the polarization of each photon. The receiver does not know which beam splitter to use for each photon and has to guess which one to use.

- Once the stream of photon has been sent, the receiver tell the sender which splitter was used for each of the photons in the sequence they were sent, and the sender compares that information with the sequence of polarizers used to send the key. The photons that were read using the wrong beam splitter are discarded, and the resulting sequence of bits becomes the key.

If the photon is read or copied in a way by an eavesdroppr, the photon's state will change. The change will be detected by the endpoints. In other words, this means you cannot read the photon and forward it on or make a copy of it without being detected.[5]

Conclusion

References

[1] File:diffie-hellman key exchange.

[2] How alan turing cracked the enigma code.

[3] Shor's algorithm.

[4] Homomorphic encryption, Aug 2020.

[5] Quantum cryptography, explained, Jan 2020.

[6] Diffie–hellman key exchange, Apr 2021.

[7] Grover's algorithm, Apr 2021.

[8] Homomorphic encryption, Apr 2021.

[9] Trapdoor function, Apr 2021.

[10] Wolfram Alpha. Elliptic curve group law.

[11] Corinne Bernstein. What is a cryptosystem? definition from whatis.com, Jun 2019.

[12] G. Brassard. Brief history of quantum cryptography: a personal perspective. In *IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security, 2005.*, pages 19–23, 2005.

[13] Lyle D. Broemeling. An account of early statistical inference in arab cryptology. *The American Statistician*, 65(4):255–257, 2011.

[14] Wikimedia Commons. File:skytalaemptystrip-shaded.png — wikimedia commons, the free media repository, 2020. [Online; accessed 31-March-2021].

[15] DI Management Services Pty Limited David Ireland. Rsa.

[16] Donald Davies. A brief history of cryptography. *Information Security Technical Report*, 2(2):14–17, 1997.

[17] Zhang et al. Dna origami cryptography for secure communication, Nov 2019.

[18] https://en.wikipedia.org/wiki/DNA$_{e}ncryption.Dnaencryption,Feb$2021.

[19] John Simkin. Enigma machine.

[20] Suetonius. Svetoni tranqvilii vita divi ivli. page 56.6.

[21] Nick Sullivan. A (relatively easy to understand) primer on elliptic curve cryptography, Feb 2019.