History of Cryptography with a focus on Elliptic Curve Cryptography

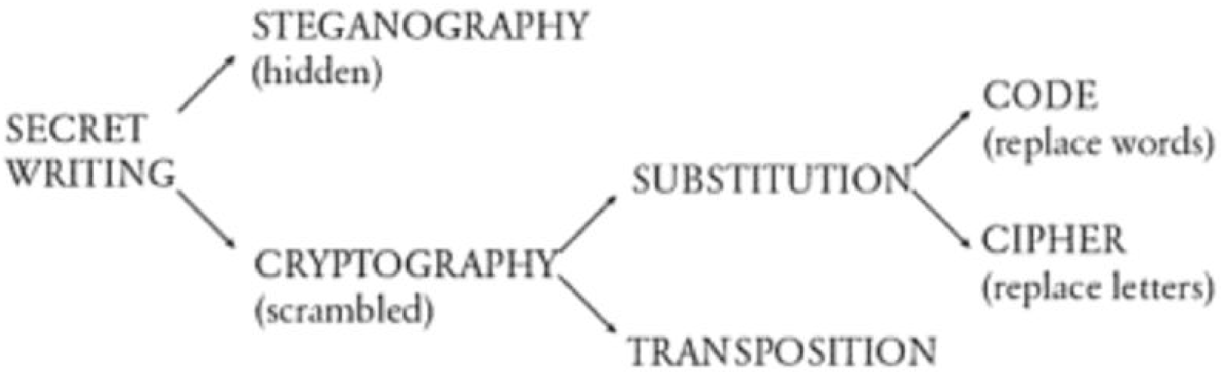
Poorva Sampat

Introduction

For centuries, various generals, members of royalty, and governments have used different modes of obscurity and encryption to ensure the security of their communications. Their methods constantly keep evolving as the flaws of the old methods are revealed. In this paper, we focus on an encryption system known as Elliptic Curve Cryptography, that was introduced about 20 years ago, and that is still being researched and modified.

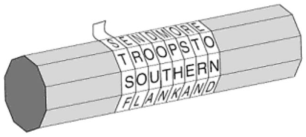
This paper is divided into three sections. Section I will cover a brief history of cryptography and explain the required terminology. Section II will focus on the method, benefits, and progress of Elliptic Curve Cryptography (ECC). Section III will provide a brief idea of work done after ECC, such as Hyperelliptic Curve Cryptography and Quantum Cryptography.

Section I



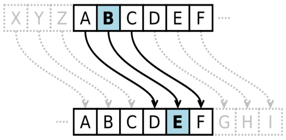
Secret writing dates back to the time of Herodotus, “the father of history”, who in his *The Histories* mentions how Demaratus, sent a message hidden underneath the wax of wooden folding tablets to warn the Spartans of Xerxes’ invasion plan. This form of secret communication that was achieved by hiding messages came to be known as *Steganography* (covered writing). Over the next two thousand years, various techniques of steganography have been used all over the world. However, confidentiality of a message in such a system is immediately compromised, should the message be discovered by say a guard. Thus, in parallel developed the field of *cryptography* (hidden writing).[2]

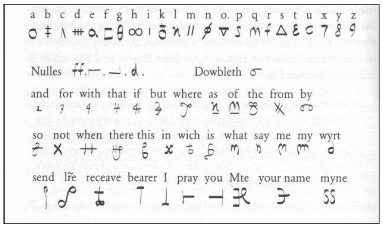
The aim of cryptography was to hide the meaning of a message instead of hiding the message itself. Cryptography itself is divided into two branches - *transposition* and *substitution.*



Transposition involves scrambling the letters in the message as seen in the “Spartan *scytale*”. The scytale is a wooden staff around which a parchment is wrapped and the message is written horizontally. The parchment is then unwound to get the strip containing the scrambled message.[2]

Substitution, on the other hand, involves changing the identity of each letter or word, instead of changing its position. Some of the earliest examples of this form of cryptography include the Caesar shift cipher, where the identity of each alphabet was shifted a certain distance, and Mary Queen of Scot's algorithm, where each alphabet and some common words were substituted by a symbol. Substitution can be further divided into two forms - *code* (substituting words) and *ciphers* (substituting alphabets). Usually, all the above-mentioned methods are applied to a message together to create a secure form of communication.




a b c d e f g h i k l m n o p q r s t u x y z
Nulles
Dowbleth
and for with that if but where as of the from by
so not when there this in wich is what say me my wyrt
send lre receave bearer I pray you Mte your name myne

The development of mathematical techniques such as frequency analysis, modular arithmetic, and so on; as well as the advancements in technology and machinery led to the constant creation of better ciphers as the old ones could easily be cracked. From the Vigenere square to Enigma machines, for thousands of years people trying to communicate, call them *Alice* and *Bob* (as is convention), depended on sharing a *secret key* otherwise known as a *symmetric key* that their enemy, *Eve,* did not know. The problem with this technique known as the *private key cryptosystem* was that it was necessary for Alice and Bob to meet to exchange the secret key which is not always possible in the current world. This introduced the *symmetric key distribution* problem. How can Alice and Bob exchange their secret key without having to physically meet?

In 1976, Whitfield Diffie and Martin Hellman published a paper called "*New directions in cryptography*" in which they introduced a new system for *key exchange* that erased these disadvantages of the previous system. This discovery was so astonishing that it completely changed the face of cryptography forever. In fact, Diffie and Hellman were well aware of the consequences of their discovery, so much so that they aptly began their paper with the sentence, "We stand today on the brink of a revolution in cryptography."[1] The concept they introduced, which allowed two people to securely exchange symmetric keys without having to physically meet was simple but very clever and it used basic modular arithmetics. Here is the Diffie-Hellman algorithm:

- A trusted party publishes a large prime p and an integer g that has a large prime order in the finite field $\mathbb{F}_p$. p and g are both publicly chosen and known to everyone.
- Alice chooses a secret integer a that is only known to herself and computes $A \equiv g^a \ (mod\ p)$. Similarly, Bob chooses a secret integer b that is only known to himself and computes $B \equiv g^b \ (mod\ p)$.
- Alice sends A to Bob, while Bob sends B to Alice.
- Alice privately computes $B^a \ (mod\ p)$, while Bob privately computes $A^b \ (mod\ p)$. Alice and Bob both now have the shared private key $B^a \equiv (g^a)^b \equiv g^{ab} \ (mod\ p) \equiv (g^b)^a \equiv A^b$ that they can use to send messages.

Security of such a key exchange protocol depends on its usage of a *one-way function* that has a *trapdoor*. A one-way function is an invertible function that is easy to compute, whose inverse is "difficult or hard to compute". This implies that it is impossible to compute the inverse in a reasonable amount of time, i.e. less than the age of the universe. A trapdoor is an extra piece of information that allows the inverse to be computed easily. Thus, Eve needs to solve the *Discrete Logarithm Problem* (DLP) to crack the message. The problem states that finding an integer solution for x in $g^x \equiv h \ (mod\ p)$, i.e. to find $log_g\ (h)$ is a hard problem.[4]

A problem that arose from the Diffie-Hellman algorithm was that it was susceptible to a *chosen plaintext attack*. So, with some modification, El-Gamal introduced the *El-Gamal key exchange* that used an additional variable called an *ephemeral key*. Both these systems, however, could only be used to establish a common shared key between both parties. The messages that actually need to be sent would still have to be encrypted with the established shared key. This introduced an extra step at the beginning of all communication. Apart from this, for Alice to communicate with multiple people - Bob, Charles, David, etc, she would need to store individual distinct keys for each receiver. Imagine trying to communicate with everyone in the United States - Alice would need to store 300 billion unique keys, which introduces the key storage issue.

In response to this issue, Ron Rivest, Adi Shamir, and Leonard Adleman introduced a *public key cryptosystem(PKC)* called *RSA*, which enabled the sender (Bob) to share encrypted messages with the receiver (Alice) without requiring to first establish the common shared key. Security of this algorithm was based on two “hard problems” - DLP and factorization of a number. Here is an algorithm that describes how Bob would send a message to Alice:

- Alice will choose two large primes p, q and compute $N = pq$. She will publish N publicly but keep p, q a secret only known by herself and then compute e, d such that $e \equiv d^{-1}\ (mod\ (p\text{-}1)(q\text{-}1))$. These are the two unique keys that belong to Alice - e is the *public encryption key* otherwise denoted as A_{pu} which is visible to everyone and d is the *private decryption key* also denoted as A_{pr} that is only known by Alice.
- If Bob wants to send a message m to Alice, he will use Alice’s public key A_{pu} or e to encrypt the message as $c \equiv m^e\ (mod\ N)$ before sending c to her.
- Alice can then use her private key A_{pr} or d to decrypt the message $m \equiv c^d\ (mod\ N)$.[4]

This algorithm uses a key property of the *Euler Phi function* $\Phi(x)$, which is - $a^{\Phi(x)}\ (mod\ x) \equiv 1$ where $a,\ x\ \in \boldsymbol{Z}^+$. In case of the above mentioned RSA algorithm where $N = pq$, $\Phi(N) = (p\text{-}1)(q\text{-}1)$. Thus, since $1 \equiv ed\ (mod\ (p\text{-}1)(q\text{-}1)) \ \Rightarrow\ ed = c(p\text{-}1)(q\text{-}1) + 1$ where c $\in \boldsymbol{Z}^+$.
We can now use this to show that the decryption algorithm leads to the initial message:

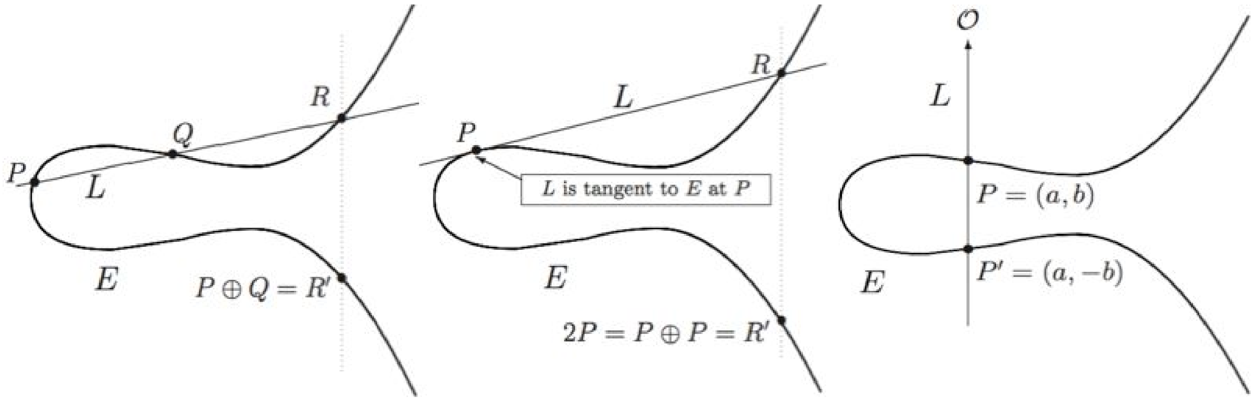
$$\begin{aligned} & c^d\ (mod\ N) \\ & \equiv (m^e)^d\ (mod\ N) \\ & \equiv m^{c(p-1)(q-1)\ +1}\ (mod\ N) \\ & \equiv m^{c\ \cdot\ \Phi(N)}\ \cdot\ m\ (mod\ N) \\ & \equiv 1\ \cdot\ m\ (mod\ N) \\ & \equiv m \end{aligned}$$

Following the invention of RSA, various modifications with Hash functions, digital signatures, padding schemes etc were introduced to ensure certain features such as authentication, integrity, confidentiality and nonrepudiation for the message. As security was added to communication techniques, various encryption cracking techniques such as man-in-the-middle attacks, collision algorithms, and the Pollard Rho methods were introduced. However, these methods are “hard” in terms of time taken to obtain results, thus, using larger sized keys resolves this issue. In 1985, Neal Koblitz and Victor Miller independently discovered the *Elliptic Curve Cryptosystem*, which is a PKC based on the structure of elliptic curves over finite fields. It requires smaller sized keys and is harder to crack.[3]

Section II

Before understanding how an Elliptic curve cryptosystem works, it is necessary to understand the “addition law” on elliptic curves. An elliptic curve is a set of solution to an equation of the form $Y^2 = X^3 + AX + B$.[4] The addition law for such a curve can be described using geometry.

Let P and Q be two points on the curve. If we draw a line L from P to Q and extend it, this line intersects the curve at three points - P, Q and R. The reflection of R across the x-axis gives a new point, R', which is considered to be the sum of P and Q. We can write this expression as $P \oplus Q = R'$.



There are, however, two properties of the addition law on elliptic curves that need to be mentioned. As the limit of Q approaches the P, the line L becomes tangent to P and $P \oplus P = 2P = R'$. However when P is added to its reflection over the x-axis, P', L is a vertical line, such that the third point created lies at infinity and is denoted as O, creating the expression $P \oplus P' = O,$ resulting in an unusable solution. Similarly we can perform point multiplication such that, $nP = P + P + P + \ldots n\ times.$[4]

For ECC, we only consider the solutions of an elliptic curve E, over a finite field $\mathbb{F}_p$, i.e. the solutions for $Y^2 \equiv X^3 + AX + B\ (mod\ p)$ which give us a set of points $E(\mathbb{F}_p)$. P, Q, R and R' all belong in $E(\mathbb{F}_p)$. The security of ECC depends on the *Elliptic Curve Discrete Logarithm Problem* (ECDLP), which is similar to the regular DLP problem. According to ECDLP, given P, Q, it is “hard” to find the value of n in the equation $Q = nP \Rightarrow n = log_P (Q)$. The first algorithm introduced for key exchange in Elliptic Curve Cryptography is known as the *Elliptic Diffie-Hellman key exchange* and it is as follows:

- A trusted third party publishes a large prime p, an elliptic curve E over $\mathbb{F}_p$, and a point P in $E(\mathbb{F}_p)$. These values are publicly known.

- Alice chooses a secret integer a that is only known to herself and computes $A = aP$. Similarly, Bob chooses a secret integer b that is only known to himself and computes $B = bP$.
- Alice sends A to Bob, while Bob sends B to Alice.
- Alice privately computes aB, while Bob privately computes bA. Alice and Bob both now have the shared private key $aB = (ab)P = bA$ that they can use to send messages.[4]

With some modification the Elliptic El-gamal PKC was introduced, following which various digital signature schemes based on ECC were also introduced.

The benefit of using an ECC compared to a conventional PKC is that ECC provides a higher level of security for the same bit size. This difference in size becomes more prominent as the key size increases. Since ECC provides higher strength per bit, in practical applications this increases transmission speed, uses less storage, reduces power consumption, which creates less heat and works on smaller hardware and software. This makes it ideal to use such an encryption technique on small devices such as smartphones, tablets, etc.[3]

Current research in this field is focusing on three main areas:

- Different forms of curves are useful for hardware versus software implementation. While binary elliptic curves are suitable for hardware implementation, prime elliptic curves are better suited for software implementation to create a fast encryption-decryption system.
- Choosing the right curve that is not susceptible to known successful pollard rho attacks is important for security, thus, scientists are constantly in search for a newer, more secure curves.
- The current point multiplication algorithm, “double and add method” is an expensive method. So, a section of ECC research is currently focusing on developing a faster, cost-effective algorithm for point multiplication.[3]

Section III

Following the introduction of ECC, here is a list of other prominent cryptosystems that have been introduced:

- **Knapsack cryptosystem** - First attempt to base a cryptosystem on an $\mathcal{NP}$-complete problem called the knapsack problem by Merkle and Hellman in 1978.[4]
- **Lattice-based cryptosystem** - Based on hard lattice problems. While ECC requires $O(k^3)$ operations to achieve k bits of security, lattice-based systems only require $O(k^2)$ operations. These systems are also easier to implement in hardware and software, however, due to their more recent introduction, security analysis on them is lacking and thus they do not have many real-world applications. The most important ones are:
    - **Ajtai-Dwork cryptosystem** - Introduced by Ajtai and Dwork. The system is secure as long as the key size is $O(n^4)$, which is not practical. All efficient, practical implementations of this system are insecure.[4]
    - **GGH cryptosystem** - Introduced by Goldreich, Goldwasser, and Halevi in 1997. The system was based on Lattice reduction using a trapdoor. However, in 1999

Nguyen described a flaw in the encryption scheme that reduced the original problem to an easier closest vector problem.[4]

  - **NTRU cryptosystem** - Proposed by Hoffstein, Pipher, and Silverman in 1996. Unlike RSA and ECC, this cryptosystem is not vulnerable to quantum attacks. However, for a special class of lattices, the problem can be reduced to an easier Shortest vector problem (key recovery) or a Closest Vector Problem (message recovery).[4]

- **Hyperelliptic Curve Cryptosystem** - Based on the equation $y^2 + g(x)y = f(x)$, where $g(x)$ and $f(x)$ are polynomials. Unlike ECC, the additive law applies to a set of points instead of a single point. The addition of two sets of g points, $\{P_1, P_2, \dots , P_g\}$ and $\{Q_1, Q_2, \dots , Q_g\}$ give a resulting set of g points $\{R_1, R_2, \dots , R_g\}$.[4]
- **Quantum Cryptography** - At limits at finding a theoretical breakthrough to crack ECC and RSA, code-breakers looked for a technological contraption. Running out of options with classical physics, scientists began looking into quantum physics and came up with the idea of quantum computers. A quantum computer relies on non-classical or quantum physics which relies on two opposing ideas.
  - The first is the idea of a computer that runs on modifying qubits (quantum bits) using quantum logic gates to simulate properties such as *superposition* and *entanglement*. These properties can be defined as the uncertainty of the state of a particle, resulting in it being in several states at once and the ability of particles at a distance to exchange quantum information.
  - The second idea is based on the *multiverse theory* (multiple universes) which states that at each point of uncertainty of n possibilities, the universe divides into n universes with the particle following through with a possibility per universe. The universes all converge once more at the next point of certainty This allows the quantum computer to process multiple states at the same time, just as the first idea of superposition does.

While this will easily compromise the security of RSA and ECC, in the 1980s Bennett and Brassard concocted a secure communication system that allowed for the distribution of *quantum keys* and encrypted messages. This system was based on the principle behind quantum cash which was first introduced by Stephen Wiesner in the 1960s. This system came to be known as quantum cryptography and is as of now the most secure form of encryption. Not just is it impossible for an adversary to decrypt a message without the key, it is also impossible for Eve to intercept a message without Alice and Bob finding out.

In 1988 Bennett witnessed the first ever quantum cryptographic exchange between two computers separated at a distance of 30 cm. In 1995, scientists at the University of Geneva implemented this via fiber optics cable between two locations at a distance of 23 km. Recently, researchers at the Los Alamos National Laboratory succeeded in a transmission over air at a distance of 1 km. [2]

The question now remains - In the race between code-makers and code-breakers, could the makers make quantum cryptography a practical technology, implemented on every device? Or, will the breakers successfully build a quantum computer first that breaks the currently used system?

Citations

[1] Diffie W. and Hellman M.E., "*New directions in cryptography*". IEEE Transaction of Information Theory, 22:644-454,1976.

[2] S. Singh, *The Code Book*, 1th ed. New York: Anchor, 2000.

[3] Kalra S., Sood S.K. (2011) "*Elliptic Curve Cryptography: Current Status and Research Challenges*". In: Mantri A., Nandi S., Kumar G., Kumar S. (eds) High Performance Architecture and Grid Computing. Communications in Computer and Information Science, vol 169. Springer, Berlin, Heidelberg

[4] J. Hoffstein, J. Pipher, J. H. Silverman, *An Introduction to Mathematical Cryptography*, New York: Springer, 2008