The Mathematics of Cryptography – Securing the Future of Digital Communication

By Ayan Singh

Introduction

Whether we are messaging our friends or sending highly confidential information to MI6, cryptography helps keep our information secret.

So – (you might be wondering) What Exactly is Cryptography?

In simple terms, cryptography is a technique of securing information and communications with the use of codes that only the authorized parties can decode. This keeps the information we want to be kept private – private. Sometimes people think cryptography is overly complicated using noticeably big sums to create a code – however, anything can be encrypted and as simply as possible!

For Example, a basic type of encryption would be as follows:

A-1, B-2, C-3, D-4, E-5, F-6, G-7, H-8, I-9, J-10 K-11, L-12, M-13, N-14, O-15, P-16, Q-17, R-18, S-19, T-20, U-21, V-22, W-23, X-24, Y-25, Z-26

9, 12-15-22-5, 4-15-9-14-7, 13-1-20-8-19! - Decode and Find the Message

Though encryptions can be as simple as this, it wouldn't do an extremely good job keeping your data safe. That is why companies that offer end-to-end encryption (such as websites where payments need to be made) often have their own encryption.

In this essay, we will be discussing different famous encryption methods, what encryption is and how it actually works and what challenges encryption can face in the future

RSA Algorithm

The RSA algorithm is a form of encryption that is widely used as it is practically unbreakable – apart from quantum computers (but there is more about that later).

The RSA algorithm is a widely used asymmetric cryptography algorithm that relies on the mathematical difficulty of factorizing large numbers.

The generation of the keys work in this format:

1. 2 large prime numbers get chosen –- assume to be *x* and *y*
2. Compute a modulus for both the keys (*assume n) -- n=x*y*
3. Calculate *(\phi(n)) -- (\phi(n)) = (x-1) * (y-1)* -- The function "*phi*" represents Eulers totient function. It counts how many integers from 1 to *n* are coprime to *n*
4. Choose an integer to become the public exponent (*assume e*) -- *e should be that: (1 < e <* \phi(n)) and *({GCD} (e, \phi(n)) = 1*
5. Determine (*d*) -- this is the modulus multiplicative inverse of *e* modulo *(\phi(n)*. This means that
   *d*e = 1 (mod* (\phi(n))

After working all this out, the public key become ((e,n)) and the private key becomes ((d,n))

So – How Does it Encrypt?

To encrypt a message (*assume m) ---- c = m^e mod n*

M – plaintext message ------- *e* – encryption exponent (public key) ------- *n* – modulus -------- *c* - ciphertext

To decrypt a message ---- *m = c^d mod n*

D – decryption exponent (private key) ------- *c* – ciphertext ---------- *m* – original message after decryption

What are the Advantages and Disadvantages?

The RSA is considered very secure due to the difficulty of factoring large numbers. It uses separate keys for encrypting and decrypting – enhancing its security. It can also be used for digital signatures, ensuring data integrity and authenticity

RSA is (on the other hand) slower compared to other algorithms – especially when crunching large amounts of data. It required large keys for security which demands more computational power. Future quantum computers could potentially be able to break the encryption.

Elliptic Curve Cryptography

Elliptic curve cryptography is a form of encryption that is based on the mathematics of elliptic curves over a finite number of fields. It provides strong security with smaller key sizes compared to the traditional methods like RSA. Let us go into more detail:

Basis of Elliptic Curves

An elliptic curve is defined by an equation with this form:

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

Where *a* and *b* are constants, and the curve is defined over a finite field *Fp* (a set of integers modulo *p*, where *p* is prime). This curve has special properties: any two points on the curve can be added together to get another point of the curve and a point can be multiplied by a number to produce another point.

How Does it generate Keys?

The private key is easy to make – a random number (presume *d)* is the private key. It gets chosen from a large range

The public key on the other hand is generated differently – a known base point (*G)* is multiplied by the private key

*P = d * G – where P is the public key*

The security of the ECC comes from the ECDLP – given P *and G,* it is computationally impossible to find *d*

How is it used in Cryptography?

1. Key Exchange

This is used for securely exchanging a shared secret. Eg,

- Alice has a private key - (dA) and computes a public key – pA = dAG
- Bob has a private key - (dB) and computes a public key – pB = dBG
- They exchange public keys with each other
- Both compute the shared secret -

S = dApB = dA(dBG) = dB(dAG) = dBpA

- Since only Alice and Bob know their private key, no-one else can compute *S*

2. Digital Signatures

This is used for verifying authenticity Eg,

- 1. Signing
    - Alice generates a random number *k*
    - Compute *R* = *kG*
    - Uses a private key dA, and message *m* to create a signature
- 2. Verification
    - Bob checks if the signature is valid using Alice's public key pA

Since solving dA from pA = dAG is hard, the signature cannot be forged

What are the Advantages and Disadvantages?

The best-known attack against the ECDLP takes exponential time, while RSA's integer factorization problem can be solved in sub-exponential time. This means ECC keys can be much smaller for the same level of security

- 256–bit ECC ≈ 3072–bit RSA
- 384–bit ECC ≈ 7680-bit RSA

This makes ECC faster and more efficient, especially for constrained devices such as smartphones and IoT

However, on the other hand – ECC is harder to implement correctly meaning even the smallest of errors can make it vulnerable to attacks if randomness or calculations are mishandled. Moreover, Quantum computers could break ECC using Shor's algorithm – making it obsolete in the long term

Public and Private Key Infrastructure

So, I know I've been rattling on about public and private keys – however, it is important that we know what it is and what it's actually about!

What are these keys?

So, by now I hope you have understood that these keys are a bit different than the ones you open your house doors with. These special keys are used for securing digital communications. This is the difference:

- Public Key – shared openly and used for encryption
- Private Key – kept secret and used for decryption

What are Key Components of this Infrastructure?

- Certificate Authority – issues and verifies digital certificates and keeps public keys secure
- Registration Authority – this acts as an intermediary between users and CA. Also verifies users before certificates are issued
- OCSP – this provides real time certificate status checks

How does Encryption Work with Public and Private Key Infrastructure?

- Encryption
  - The sender encrypts the data with the receiver's **public key**
  - Only the receives **private key** can decrypt it
- Digital Signatures
  - The sender signs a message using their **private key**
  - The receiver verifies it using the sender's **public key**

What are the most use cases for Public and Private Key Infrastructure?

- Secure websites
- Email Encryption
- Secure software updates
- Digital identities and authentication

What Are the Threats that Could Come in the Future?

- Quantum Computers and Shor's Algorithm (could break RSA and ECC)

RSA encryption is based on the difficulty of factoring a large integer

$N = p \times q$

where p and q are large prime numbers. Classical algorithms for factorization, such as the General Number Field Sieve (GNFS), run in sub-exponential time:

$$O\left(e^{(c(\log N)^{1/3}(\log\log N)^{2/3})}\right)$$

Shor's algorithm, however, uses Fourier transforms to solve the problem in polynomial time, roughly:

O ((log N)^3)

This makes RSA highly vulnerable as quantum computers can factorise 2048-bit keys in seconds – breaking the security.

For ECC, Shor's algorithm can compute logarithms in:

O (m^3)

Where m is the is the bit length of the elliptic curve order. This effectively breaks ECC encryption.

- Lattice-Based Attacks and Learning with Error (LWE)

Lattice-based cryptography is still considered to be resistant to quantum attacks. These rely on the difficulty of problems like the SVP – where given a basis *B* of a lattice *A* – the goal is to find the shortest non-zero vector.

$$\min_{x \neq 0} \|Bx\|$$

This problem is hard in high dimensions but can be approximates using LLL Reduction or BKZ algorithm. Recent improvements have significantly weakened some cryptographic schemes.

In the LWE problem, encryption is based on solving linear equations:

$Ax + e = b \pmod q$

Where *A* is a known matrix, *x* is the secret and *e* is a small error vector. Solving for *x* without *e* is believed to be hard but improvements in lattice reduction techniques may reduce is security margin

Both of these threats pose serious risks to modern encryption and decryption. The best-known threat is quantum computing, but breakthroughs in classical algorithms, artificial intelligence, and mathematical breakthroughs could also undermine encryption in unexpected ways. Cryptographers are working every day on post-quantum cryptography to help counter these risks, but nothing is entirely future proof.

Mathematical Challenges – How to Defend against Quantum Computing

Even though new recent developments have introduced the fact that our methods of cryptography might not be safe anymore, don't worry! For still we have some mathematical problems that quantum computing cannot yet solve. (These aren't exactly 100% known to

counter quantum computers but we think these are mathematical problems that they will not be able to break.

- Hard Mathematical Problems in PQC

SVP – given a basis *B (b1, b2, b3….)* of a lattice A, find the shortest non-zero vector v (= a

$$\text{SVP:} \quad v = \arg \min_{v \in \Lambda, v \neq 0} \|v\|$$

This problem is still considered to be hard to solve by the quantum computers

Ring-LWE Variant – this works in polynomial rings such as $\mathbb{Z}_q[x]/(f(x)),$, reducing the key sizes while maintaining security

- Code – Based Cryptography

This relies on the syndrome decoding problem in error-correcting codes.

Given a random generator matrix *G and* a corrupted codeword *y=xG+e*, recover *x*

This is believed to be hard for quantum computing to break. McEliece cryptosystem is based on this problem but has very large key sizes (sometime 100 KB)

- Isogeny Based Cryptography

This uses elliptic curve isogenies and the difficulty of finding isogenies between supersingular elliptic curves.

Given two elliptic curves e1, e2 over fP find an isogeny (/phi): e1 --> e2 such that:

(/phi) (x,y) = g ((x,y), h(x,y))

For some rational functions – g,h

SIDH was broken recently using torsion point attacks, but CSIDH and SIKE are still being studied (these are all types of isogeny based cryptographic encryption methods)

Conclusion

Cryptography has been a pillar in facilitating secure digital communication, from simple ciphers to more advanced encryption methods like RSA and ECC. However, as technology

evolves, so do the threats. Quantum computing, in particular using Shor's algorithm, poses an existential risk to traditional cryptographic measures by rendering RSA and ECC useless. To counter such threats, post-quantum cryptography (PQC) is emerging as the new frontier in digital security. Cryptography founded on mathematical issues such as lattice-based cryptography (LWE and SVP), code-based cryptography, and isogeny-based cryptography are among the strongest contenders to be quantum-resistant. While there are ongoing advancements in PQC, there are also obstacles such as large key sizes and computational efficiency.

The future of cryptography depends on continuous research and innovation. Governments, organizations, and cryptographers worldwide are standardizing quantum-resistant encryption techniques. The National Institute of Standards and Technology (NIST) has already shortlisted some PQC algorithms for real-world implementation, so that our digital infrastructure remains secure in the decades ahead. No encryption method is fully future-proof, however, because unforeseen mathematical breakthroughs or advancements in artificial intelligence could introduce new weaknesses.

Finally, the integrity of our digital realm remains a constant competition between cryptographers and adversaries. As we move toward a quantum-capable era, adaptability and forward-thinking research will be essential to keeping cryptography a pillar of digital security. From protecting personal correspondence, financial data, or state secrets, encryption will always be at the center of protecting information within our increasingly interlinked world.