An essay on mathematical methods in cryptography

By Tobias Kreutz Wellsted.

1. Introduction

What is cryptography? Why do people study it and in what way does it help us? These may be the initial questions of any reader of this essay, and it is my absolute pleasure to help answer those very questions and explain what I can about this amazing study. As a computer scientist and mathematician, I find it fascinating how so much of our modern everyday lives, overshadowed by the use of highly advanced computer technologies, goes completely unnoticed, and how mathematics is one of the core subjects involved with building up the foundations of ideas such as data security and integrity.

What is cryptography and why is it important?

According to wikipedia: "*Cryptography is the study or practice of techniques for secure communication in the presence of adversarial behaviour*"[1]. What this really means is 'cryptography is the study of making sure people who are listening into communications between people on the internet can't understand what is being said'.
Let's say, you are talking with your friend, and you do not want others to be able to eavesdrop in on your conversation, cryptographers make sure that your wish becomes true, by using rigorously defined and tested concepts such as encryption.

To understand more about cryptography, you need to understand its main method: encryption; it is defined simply by the following three ideas:

Plaintext - This is the original message that you wish to send. This has not in any way been modified, and is what you want to communicate to the recipient.

Ciphertext - This is plaintext, except it has been 'encoded' or scrambled such that it is unrecognisable to any person or computer. It is now safe to share publicly as nobody will be able to decipher what the original message was.

Encryption Algorithm - This is what a computer uses to translate Plaintext into Ciphertext.

Encryption is the process of taking some **plaintext**, applying an **encryption algorithm** to it, transforming it into **ciphertext**.

Why is encryption important? Well, it is especially helpful when dealing with sensitive or confidential data such as credit card numbers, personal information or passwords. Communications across the internet that include this type of data are meant to be entirely safe from the eyes and ears of everyone who is not the sender or recipient.

Basically all data sent across the internet is intercepted; If any data is understood by another person, then it is considered to be ‘no longer safe’, someone else knows what they should not, and you can’t really do much about it. Data that is not securely encrypted, could lead to a lot of cybercrime such as fraud or identity theft, usually leaving the victims without any idea of what is happening, as it’s basically impossible to detect when someone has intercepted data.

In computer science, the most interesting part is often the theory behind how things work. With cryptography, this comes in understanding how things are encrypted - the different encryption algorithms.

The different encryption algorithms

There are hundreds of encryption algorithms that each have their own unique and precise methods, some are more complex than others, and others take more time to compute / work (although, most algorithms take a matter of milliseconds to process). Only a select few secure, tried and true methods are used - making sure that your everyday internet communications are entirely safe - due to the sheer mountain of resources cyber criminals have at their disposal.

All encryption algorithms work by taking in something known as a ‘key’. Keys determine the way in which the encryption algorithms encode plaintext into ciphertext; Data is always encrypted in the same way with the same key, but is never (or nearly never) the same when encrypted using different keys. This works in exactly the same way as your front door! When you open it, you use your key - things may only pass into your house from the outside world using your house’s key to open the door.

Methods are grouped by the way in which they share and use these keys. In symmetric encryption, both parties gain access to the same secret key used to both encrypt and decrypt messages, while with asymmetric encryption two keys come into circulation, one is allowed for public use, it allows anyone to encrypt and send their messages to you, and the second, only you have; it allows you to decrypt all of the messages sent using the public encryption key. Again, this is much like your postbox - people can send you their enveloped mail, which only you can then open and read.

Below, 4 encryption algorithms are listed, filtered by their general methods, with a (very) brief description of what they do.

Symmetric methods:

TDES (triple data encryption standard)

This is an enhanced version of another algorithm (DES), applying itself 3 times, producing three keys instead of one! It's not a very adaptable method, and has been largely replaced.

AES (advanced encryption standard)

AES, adopted by the U.S. National Institute in 2001, was built to outdo TDES, it's far faster, more secure and more adaptable. This is now one of the most commonly used encryption standards.

Asymmetric methods:

It is good to know that all asymmetric methods are considered "less secure" than their symmetric counterparts, as when their public key is shared, a small amount of vital information about their private key is also shared, which can be deciphered, leading to any messages being sent becoming compromised.

RSA (Rivest, Samir and Adleman)

RSA works by relying on the fact that multiplying prime numbers is easy, but factoring them is extremely difficult. It uses the neat mathematics of modulus' to encrypt and decrypt messages.

ECC (elliptic curve cryptography)

This relies on the maths of elliptic curves to produce small, quickly computable keys, fit for any device to process, balancing performance and security.

---

2. The mathematics of encryption

In computer science, cryptography is a highly regarded study, but to me, it starts to shine once you see the ingenious ideas it utilises, tackling these strangely abstract problems, finding some intricate way to make communication secure.

For this essay, I looked more closely at one of the most common methods, giving a precise explanation as to how it works and also proving that it always works as intended.

RSA (Rivest, Samir and Adleman)

Aside from RSA's simply amazing name (it gives full, direct credit to all of its creators just by mentioning it), it is one of the most used encryption standards of all time. It abuses simple yet effective facts and mathematics and computation to allow near watertight security for communication.

How it works:

Firstly, two large prime numbers are found - p & q (where $p \neq q$) - multiplying them together to form the first part of their public key, the modulus:
$pq = n$.

Then, the other part of the public key, the exponent e is chosen such that $e > 1$ (as message data would be lost) and less than some number $\varphi(n)$. Note that it must also be coprime to $\varphi(n)$, as the method would not hold up for all messages sent.

$\varphi(n) = (p-1)(q-1)$
$1 < e < \varphi(n)$

These two calculated numbers (n & e) are sent out as the public key that anyone can use to encrypt their messages. This works by applying the following equation to their message:
$C \equiv M^e \pmod{n}$
(where M represents the message they wish to send, and C, the corresponding ciphertext).

In the meantime, another exponent - d, the private key - is also calculated:
$ed = 1 \pmod{\varphi(n)}$, where $1 < d < \varphi(n)$.

Finally, any message, once received, can be decrypted using d in the following equation:
$M \equiv C^d \pmod{n}$
(again, where M represents the message that was sent, and C, the ciphertext received), ensuring the security of all messages sent to the holder of the private key.

Proof:

You can clearly see at a glance that RSA works, especially if you have tested a few numbers and messages, but, to be rigorous and know exactly that it always works, it needs to be proved. So, I have tried my best to make it clear as to why this method is correct.

The first step is simply seeing that
$C \equiv M^e \pmod{n}$ and $M \equiv C^d \pmod{n} \implies M \equiv (M^e)^d \pmod{n}$.
This works as numbers applied under a modulus are always equivalent, it does not matter when it is applied.

To prove $M \equiv M^{ed} \ (mod\ n)$, we need to prove $M \equiv M^{ed} \ (mod\ p)$, and $M \equiv M^{ed} \ (mod\ q)$ (however, the proof is the same for both p and q so, the proof for q has been omitted).

There are two cases, dependent on what the greatest common divisor of M and p is; We know, as p is prime, its only factors are p and 1:

If the greatest common divisor of M and p is p;
$M^{ed} \equiv m \ (mod\ p)$ as $M/p = 0,\ M \ (mod\ p) \equiv 0$, and $0^c = 0$ for all c.

If the greatest common divisor of M and p is 1;
First, aside, as $ed = 1 \ (mod\ \varphi(n))$, $ed = 1 + k\varphi(n)$

$M^{ed} \equiv M \ (mod\ p) \Longrightarrow M^{1+k\varphi(n)} \equiv M \ (mod\ p)$

$= \ M \cdot M^{k(p-1)(q-1)} \equiv M \ (mod\ p)$. Multiplying both sides by M's multiplicative inverse (M^{-1}) (it fulfils the equation: $M \cdot M^{-1} = 1 \ (mod\ p)$)

$= \ M^{-1} \cdot M \cdot M^{k(p-1)(q-1)} \equiv M^{-1} \cdot M \ (mod\ p) \ = \ M^{k(p-1)(q-1)} \equiv 1 \ (mod\ p)$.
Here, we can assume Fermat's little theorem to show that $M^{p-1} \equiv 1 \ (mod\ p)$ leaving us with $1^{k(q-1)} \equiv 1 \ (mod\ p)$, and simply, we know that $1^k \equiv 1$ for any k

$\Longrightarrow M^{ed} \equiv M \ (mod\ p)$.

Now, we need to prove that this also works under the modulus of n. Using the ideas above about p and q, we can state that $M^{ed} = M + k_1 p$ and $M^{ed} = M + k_2 q$ (this is implied by the definition of a modulus).

Combining the two equations: $M + k_1 p = M + k_2 q$, subtract M.
This implies that (in both ways, only here shown for the case of $k_1 p$) $k_1 p/q = k_1$. To check this statement, we have to use Euclid's Lemma showing $q \ divides \ k_1$ or $q \ divides \ p$, but as q and p are coprime, the second statement comes to a contradiction, implying:
$q \ must \ divide \ k_1 \ (k_1 = k_3 q)$.

In the final step of this proof, we simply take $M^{ed} = M + k_1 p$, and as

$k_1 = k_3 q,\ M^{ed} = M + k_3 qp$, and $qp = n$. Putting this all together, we can show that this is the definition of the original modulus; RSA's method is correct.
$M^{ed} = M + k_3 n \Longrightarrow M^{ed} \equiv M \ (mod\ n)$.

3. Conclusion

This essay was extremely fun to research and make - the time I had, has almost vanished! I wish the word count would have allowed me to be more rigorous, write more, and I wish I had more time, but If I want to submit this, then I unfortunately have to leave what's there as it is.

I want to give credit and thank my sister, Lucy, and my friend (anonymous) for helping me with this. I really hope that all who have read this essay feel the same excitement for cryptography as I do.

Sources:

[1]https://en.wikipedia.org/wiki/Cryptography

https://nordvpn.com/blog/rsa-encryption/?nc=21317059647&ns=google&nm=cpc&nt=&na=6656841114&gad_source=1&gad_campaignid=21317078835&gbraid=0AAAADwsNkS47C_PKDPRJQKiDamAhqlop&gclid=Cj0KCQjwpv7NBhCzARIsADkIfWysdlVwyhcb-fQa-TYa1hlKqPFzziw8c_11k53-4mes0WLKjkUYzOUaAsjQEALw_wcB#key-generation

https://www.geeksforgeeks.org/ethical-hacking/encryption-its-algorithms-and-its-future/

https://www.youtube.com/watch?v=qph77bTKJTM

https://www.geeksforgeeks.org/computer-networks/digital-signature-algorithm-dsa/

https://www.youtube.com/watch?v=AG5Q5H_nAWQ

https://en.wikipedia.org/wiki/Advanced_Encryption_Standard