

- DESIGNING
SYMMETRIC
CRYPTOGRAPHY

\$whoami

Kostas Mpenos

- Linux Enthusiast
- Programmer
- Aiming to become a security engineer



1

INTRODUCTION

A few things on symmetric crypto

● INTRODUCTION

○ Symmetric cryptographic algorithms are the ones that use the same key for both encryption and decryption.

This requires that the sender and the receiver of a message both have access to the key.

The above requirement is the biggest drawback of symmetric crypto.

A teal circle containing the number 2, positioned on the left side of the slide. A thin white vertical line runs through the center of the circle and extends above and below it.


2

PRINCIPLES

Design principles of symmetric crypto algorithms



● PRINCIPLES



Designing symmetric-key algorithms is a tough and sensitive task.

The designer of a cryptosystem should be aware of the best practices used and apply them to his new system while enhancing them to maximize security.



PRINCIPLES

A good rule of thumb when it comes to designing your own cryptosystem is:

● PRINCIPLES



NEVER DESIGN YOUR OWN CRYPTO

...or at least don't until you *know how to break crypto.*



PRINCIPLES

In the 19th century Auguste Kerckhoff stated one of the most known design principles(*called Kerckhoff's desideratum/assumption/axiom/law*).



PRINCIPLES

In the 20th century Claude Shannon reformulated Kerckhoff's Law, with bigger focus on cryptosystem design (*called the Shannon's maxim*).

● PRINCIPLES

○ In 1945 Shannon published his classified report “*A Mathematical Theory of Cryptography*” in which he gave the definition of perfect secrecy and identified the two properties of a secure cipher.



PERFECT
SECRECY

● PRINCIPLES - PERFECT SECRECY

○ A cryptosystem is Shannon secure (of perfect secrecy) if for any message **x** and any encipherment **y**:

$$\rightarrow \text{Prob}[x|y] = \text{Prob}[x]$$

This implies that for every (message, cipher) pair there is at least a unique key that connects them.

$$\rightarrow |\text{Keys}| \geq |\text{Ciphers}| \geq |\text{Messages}|$$

● PRINCIPLES - PERFECT SECRECY

- Shannon's perfect secrecy principle is non-implementable since in the current system's storage and speed is of high importance but gives us the conclusion that we can't create perfect security cryptosystems.



CONFUSION AND DIFFUSION

● PRINCIPLES - CONFUSION AND DIFFUSION

○ **Confusion**

The ability of a cryptosystem to produce a ciphertext in which each bit depends on several parts of the key, in a complex way.

Diffusion

The ability of a cryptosystem to produce very different output with very similar (but not identical) inputs.

● PRINCIPLES - CONFUSION AND DIFFUSION

○ Diffusion is also known as avalanche effect.

One way to test a system's avalanche capabilities is the **Strict Avalanche Criterion (SAC)**.



STRICT AVALANCHE CRITERION

● PRINCIPLES - STRICT AVALANCHE CRITERION

○ A cryptosystem is said to satisfy the strict avalanche criterion if, whenever a single bit is complemented in the system's input each of the output bits should change with a probability of 50%. This applies to both encryption and decryption.

● PRINCIPLES - CONFUSION AND DIFFUSION

○ EXAMPLE - AES

- How is confusion applied to AES?
- How is diffusion applied to AES?

Any ideas?



● PRINCIPLES - CONFUSION AND DIFFUSION

EXAMPLE - AES

Confusion → Byte Substitution

Diffusion → Row Shifting, Column Mixing



SUMMING UP

● PRINCIPLES - SUMMING UP

○ In the previous slides we explained the principles of symmetric-key crypto design:

- Avoid security through obscurity
- Shannon's perfect secrecy
- Confusion
- Diffusion (Avalanche effect)

So what's next?



CRYPTANALYSIS

● CRYPTANALYSIS

○ In cryptography there are no mathematical proofs/security measurements for the majority of the available systems.

The levels of secureness of a cryptosystem are decided by the results of a series of attacks on it.

● CRYPTANALYSIS

○ Applying cryptanalysis attacks on a cryptosystem is fundamental to test and enhance its security.

A designer should be aware of the known attacks and their sophisticated versions, combine them, refactor them and run them against his system multiple times, in order to find its weaknesses.



3

CONCLUSIONS

What we have learned



CONCLUSIONS

In this presentation we have discussed about the principles that every symmetric-key cryptosystem should comply to.

If you are confident enough to attempt to design your own crypto you should keep these in mind while combining them with more sophisticated methods in order to build an acceptable system.



Thanks!

ANY QUESTIONS?

CREDITS

Sources:

- https://en.wikipedia.org/wiki/Kerckhoffs%27s_principle
- https://simple.wikipedia.org/wiki/Avalanche_effect
- http://cryptography.wikia.com/wiki/Confusion_and_diffusion
- <https://www.iacr.org/museum/shannon/shannon45.pdf>
- <http://www.cs.miami.edu/home/burt/learning/Csc609.011/Perfect/>

Presentation Design:

- Presentation template by SlidesCarnival
- Photographs by Unsplash

● CREDITS

○ A fun comic explaining how AES works:

- <http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html>

A Github project implementing a symmetric-key algorithm in Python, **for educational purposes**:

- <https://github.com/DaKnOb/CryptoAlgorithmChallenge>