Crypt(?) -

Objectives

Requirements / Goals / Attributes / ...

CIA Triad	
	Confidentiality
	Integrity
	Availability

Authentication

Non-repudiation

Terminology

Cryptology	*Attack*	*Defences*
Cryptography	*Adversary*	*Mitigations*
Cryptanalysis	*Corrupted / Malicious Party*	*Countermeasures*
Cryptogram		

Cryptology

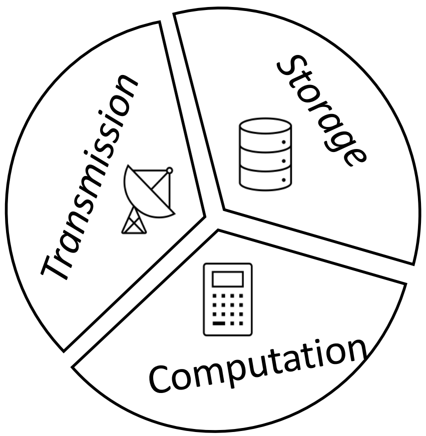
Security

Pen & paper

Mechanical computation

Electronical computation

Quantum computation



Secure all!

Attacks

Passive	vs.	*Active*
Outsider		*Insider*

Deteriorate

Corrupt

Impersonate

Alice Bob Charlie Daisy ... Oscar / Eve

Pages on SecuRity by Ruxandra F. Olimid

https://pagesonsecurity.blogspot.com/

Principles -

Kerckhoffs's principle

Only keep hidden the key.
(e.g., make the design public)

Principle of sufficient keys

The number of possible keys must be large.
(e.g., avoid brute force)

Principle of (key) separation

Use different keys for different contexts, compartmentalize.
(e.g., minimise the damage of a leak)

Principle of simplicity

Keep everything simple.
(e.g., unnecessary complexity brings in risks)

Principle of diversity

Use different types of … e.g., cryptographic algorithms.
(e.g., avoid same attacks against all)

Security by default

Keep default configuration as secure as possible.
(e.g., deny access by default)

Principle of minimal trust

Minimise the number of trusted entities, don't trust easily.
(e.g., do not say your secrets to anyone)

Principle of the weakest link

A system cannot be more secure than its weakest component (link).
(e.g., secure all components)

Principle of least privilege

Grant the exact privileges required to perform the job.
(e.g., do not grand less or more privileges)

Security by design

Build in security from start.
(e.g., integrate security in all design and development stages)

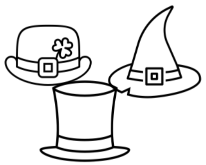
Principle of modularization

Keep things modular.
(e.g., easily change one component with another)

Defence in depth

Use diverse security strategies at different layers.
(e.g., use physical and technological security)

Ethics!



Security through obscurity (?)

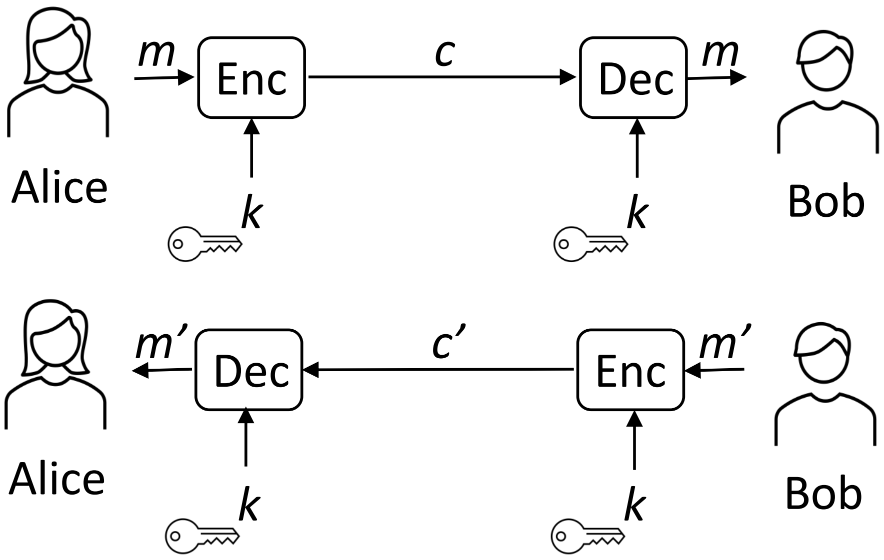
Oblivious Transfer, Obfuscation, Covert Channels, … ; Kleptography; Standardisation …

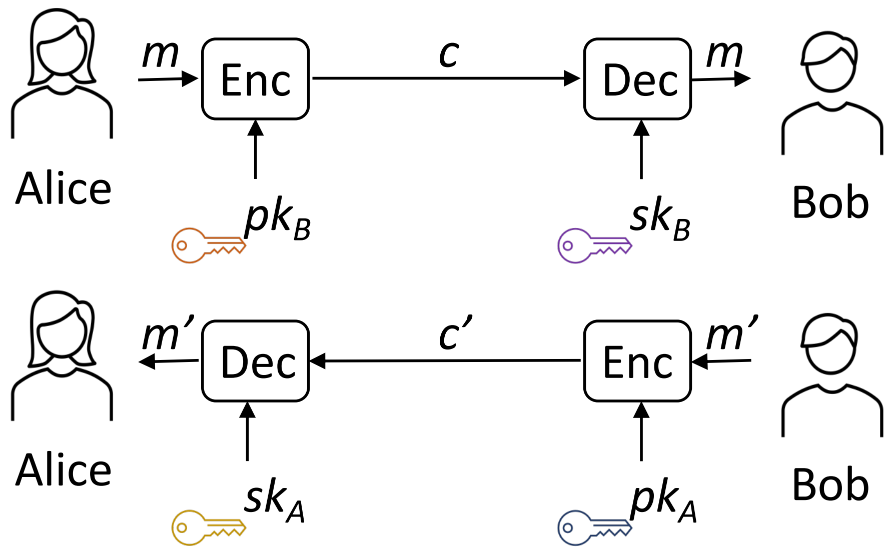
Symmetric vs. Asymmetric Encryption -

https://pagesonsecurity.blogspot.com/

Symmetric	*Asymmetric*

… encryption





Encryption: $c = \mathrm{Enc}(k,m)$
Decryption: $m=\mathrm{Dec}(k,c)$
Correctness: $\forall m \in \mathcal{M},\ k \in \mathcal{K}$
$\mathrm{Dec}(k,\mathrm{Enc}(k,m)) = m$

\+ Shorter keys

− Key establishment

\+ Private keys never leave the owner

− Computational cost & speed

Encryption: $c = \mathrm{Enc}(pk_B,m)$
Decryption: $m=\mathrm{Dec}(sk_B,c)$
Correctness: $\forall m \in \mathcal{M},\ (pk_B,\ sk_B) \in \mathcal{K}$
$\mathrm{Dec}(sk_B,\mathrm{Enc}(pk_B,m)) = m$

Terminology

k: symmetric key	*m*: plaintext
pk: public key	*c*: ciphertext
sk: private (secret) key	Enc: encryption alg.
(pk,sk): public-private key pair	Dec*:* decryption alg.
	Cryptanalysis

Confidentiality

No. of keys

for N bi-directional communicating parties

Each: N-1 [k]	vs.	*Each: 1 [sk], N-1 [pk]*
Total: N(N-1)/2 [k]		*Total: N [sk], N [pk]*

Pages on SecuRity by Ruxandra F. Olimid

Unconditional vs. Conditional Security -

Unconditional (Information-theoretic)	Conditional (Computational)

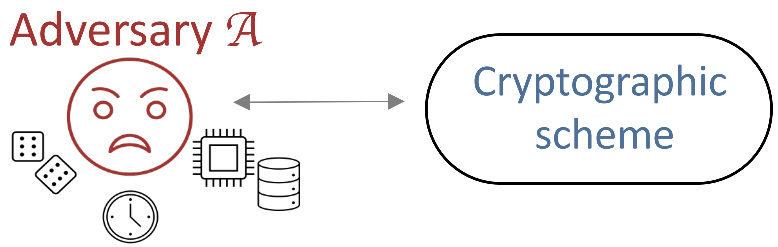
… security

*An **adversary** with **no restrictions** (**unbounded** computational resources - time, memory) **cannot break** the scheme.*

*An **adversary** with **computational restrictions** (**bounded** time, memory) **can break** the scheme with **some (negligible) probability**.*

Stands against brute force +

Good in theory, poor in practice −



+ Suitable for practice

− Weaker than unconditional security

A cryptographic construction satisfies ***computational security*** if any adversary $\mathcal{A}$ that runs the attack in a time $t(n)$ succeeds the attack with probability at most $\varepsilon(n)$; t and ε are functions of a ***computational security parameter*** n.

Statistical Security

A cryptographic construction satisfies $\varepsilon(\lambda)$ ***statistical security*** if any unbounded adversary $\mathcal{A}$ succeeds the attack with probability at most $\varepsilon(\lambda)$; ε is function of a ***statistical security parameter*** λ.

- Introduces a *small* advantage $\varepsilon(\lambda)$ wrt the *a-priory probability* of winning

Statistical and *computational* security are both ***relaxations*** of information-theoretical security.

PPT (Probabilistic Polynomial in Time) Adversary

- $t(n)$ is **polynomial** in n
- $\varepsilon(n)$ is **negligible** in n

Negligibility:

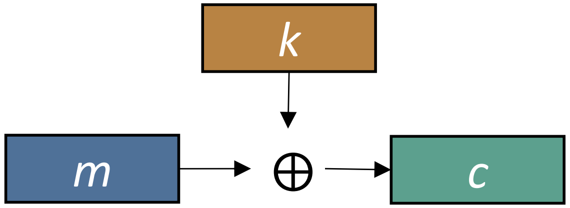
$\forall p(n)$, $\exists\, n_d$ such that $\forall n \geqslant n_d$ it holds $\varepsilon(n) < 1/p(n)$

$p(n) = n^d$ and d constant

Examples:

✗ $1/2$, $1/n^{100}$ ✓ $1/2^n$, $p(n)/2^n$

One Time Pad (OTP) -



Encryption: $c = k \oplus m$
Decryption: $m = k \oplus c$

$\oplus$ (XOR)	k: 0	k: 1
$m(c)$: 0	0	1
$m(c)$: 1	1	0

The key k:

- is as long as the plaintext m and the ciphertext c
- is uniformly random chosen in $\mathcal{K}$
- must be used only once

Perfect Secrecy

For all m possible plaintext (i.e., all m in $\mathcal{M}$) and any c ciphertext (i.e., all c in $\mathcal{C}$) such that $Pr[C=c]>0$, it holds:

$$Pr[M=m|C=c] = Pr[M=m]$$

Theorem (key length bounding):

Let (Enc, Dec) be a perfectly-secret encryption scheme over a plaintext space $\mathcal{M}$ and a key space $\mathcal{K}$. Then it holds that $|\mathcal{K}| \geqslant |\mathcal{M}|$ (i.e., the length of the key is larger or equal to the length of the message).

Examples

k: 0 1 1 0 1 1 0 0 $\oplus$
m: 1 0 1 1 1 0 0 1
c: 1 1 0 1 0 1 0 1

k: G F N O M $\oplus$
m: P A G E S (mod 26)
c: V F T S E

\+ Easy, fast encryption and decryption

− Long key length

Multiple use of the same key k

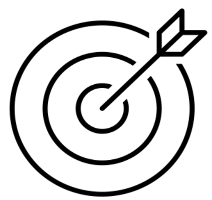
$c_1 = k \oplus m_1$, $c_2 = k \oplus m_2$, ...

Attack 1. $\mathcal{A}$ knows the ciphertexts c_1, c_2

$\mathcal{A}$ finds a relation between the plaintexts: $m_1 \oplus m_2 = c_1 \oplus c_2$

Attack 2: $\mathcal{A}$ knows (at least) the pair (m_1, c_1)

$\mathcal{A}$ finds the key $k = m_1 \oplus c_1$, then decrypts $m_2 = k \oplus c_2$



Perfect secrecy

Pages on SecuRity
by Ruxandra F. Olimid

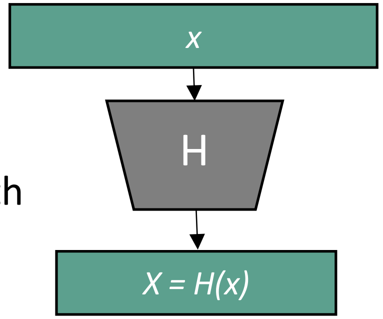
Cryptographic Hash Function -

https://pagesonsecurity.blogspot.com/

Cryptographic Hash Function

$H: \{0,1\}^* \to \{0,1\}^{l(n)}$

- arbitrary input length, fixed output length
- deterministic
- “easy” to compute, “difficult” to invert



Attacks: Birthday attack

l(n) = poly(n), with n the security parameter
$\{0,1\}^*$: sequence on bits, regardless its size
s.t.: such that
$\mathcal{A}$: adversary

Security

Collision resistance

$Hash^{coll}_{\mathcal{A},H}(n)=1$ if
$\mathcal{A}$ outputs $x,y \in \{0,1\}^*$ s.t.
$x \neq y$ and $H(x) = H(y)$
$Hash^{coll}_{\mathcal{A},H}(n)=0$, otherwise

H is *collision resistant* if
$\forall\mathcal{A}$ PPT, $\exists\ \varepsilon(n)$ negligible s.t.:
$\Pr[Hash^{coll}_{\mathcal{A},H}(n)=1] \le \varepsilon(n)$

Second pre-image resistance

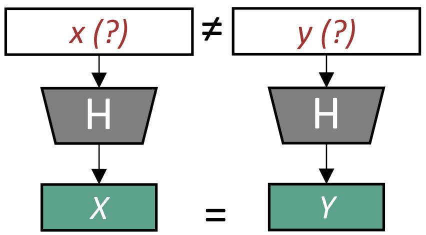
$Hash^{2nd\text{-}pre\text{-}img}_{\mathcal{A},H}(n)=1$ if
given $x \xleftarrow{R} \{0,1\}^*$,
$\mathcal{A}$ outputs $y \in \{0,1\}^*$ s.t.
$x \neq y$ and $H(x) = H(y)$
$Hash^{2nd\text{-}pre\text{-}img}_{\mathcal{A},H}(n)=0$, otherwise

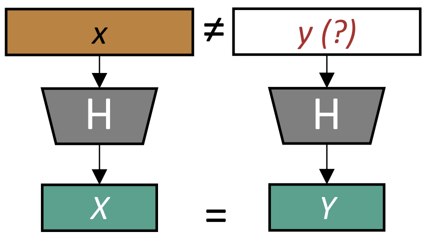
H is *second pre-image resistant* if
$\forall\mathcal{A}$ PPT, $\exists\ \varepsilon(n)$ negligible s.t.:
$\Pr[Hash^{2nd\text{-}pre\text{-}img}_{\mathcal{A},H}(n)=1] \le \varepsilon(n)$

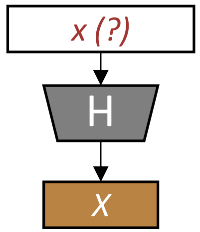
First pre-image resistance

$Hash^{1st\text{-}pre\text{-}img}_{\mathcal{A},H}(n)=1$ if
given $X = H(x')$, $x' \xleftarrow{R} \{0,1\}^*$,
$\mathcal{A}$ outputs $x \in \{0,1\}^*$ s.t.
$H(x) = X$
$Hash^{1st\text{-}pre\text{-}img}_{\mathcal{A},H}(n)=0$, otherwise

H is *first pre-image resistant* if
$\forall\mathcal{A}$ PPT, $\exists\ \varepsilon(n)$ negligible s.t.:
$\Pr[Hash^{1st\text{-}pre\text{-}img}_{\mathcal{A},H}(n)=1] \le \varepsilon(n)$







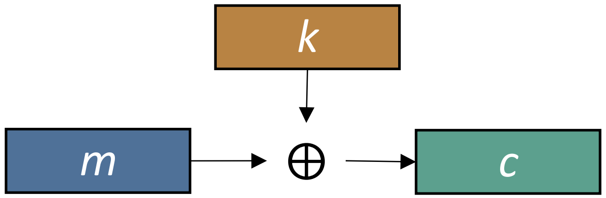
one-way function

higher security ← *lower security*

One Time Pad (OTP)

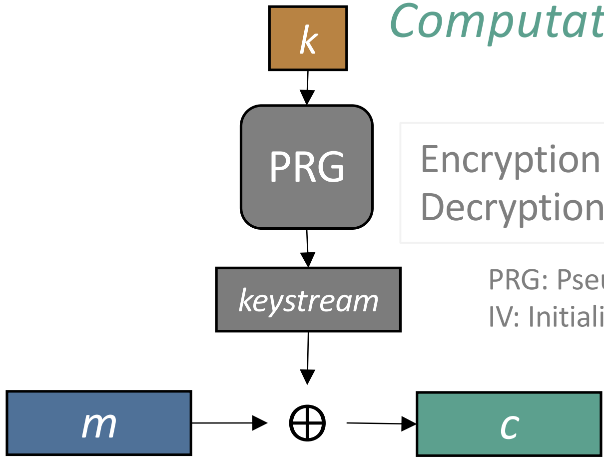
Perfect secrecy

Encryption: $c = k \oplus m$
Decryption: $m = k \oplus c$



Stream Ciphers

Computational secrecy

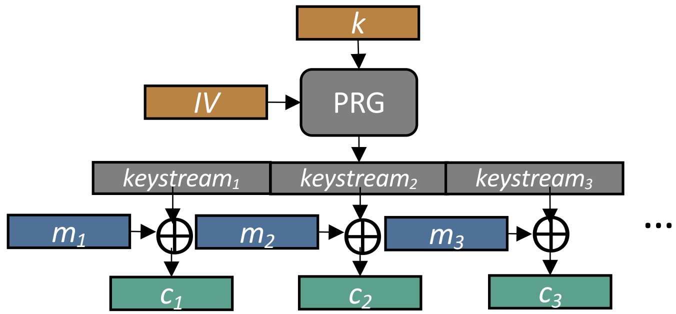


Encryption: $c = PRG(k) \oplus m$
Decryption: $m = PRG(k) \oplus c$

PRG: Pseudo-Random Generator
IV: Initialization Vector

Synchronized Mode

Encryption: $c_1 || c_2 || c_3 \ldots = (IV, PRG(k,IV) \oplus m_1 || m_2 || m_3 \ldots)$
Decryption: $m_1 || m_2 || m_3 \ldots = PRG(k,IV) \oplus c_1 || c_2 || c_3 \ldots$
IV chosen uniformly at random



Unsynchronized Mode

Encryption: $c_i = (IV_i, PRG(k,IV_i) \oplus m_i)$
Decryption: $m_i = PRG(k,IV_i) \oplus c_i$
$IV_1, IV_2, \ldots$ chosen uniformly at random
(and thus independent)