Definições de Criptografia

        - Problema do Logaritmo Discreto
    - Problema da Fatorização de Números Primos
    - Protocolos de Acordo de Chaves
        - Protocolo de Acordo de Chaves Diffie-Hellman
        - Puzzels de Merkel
- RSA
    - ElGamal
    - Cifras de Chave Pública
    - Assinatura Digital
    - Certificados Digitais
    - Enters PKI
        - O Certificado X.509
    - PKI
        - Protocolos da PKI
    - *Pretty Good Privacy*(PGP)
        - Confiança em PGP
    - Cartão de Cidadão da República Portuguesa
- + O PIN de assinatura digital tem 4 dígitos (introduzido até 3 vezes).
- + O PUK da assinatura dígital tem 4 dígitos.

Criptografia

> Conjunto de técnicas que procuram tornar possível a comunidade secreta entre dois agentes, sobre um canal aberto.

A criptografia é uma ciência rigorosa que elabora em 3 passos:

1. **Especificar**, de modo preciso, **o modelo de ataque** a que o sistema estará sujeito;
2. **Propor uma construção** / sistema que simultaneamente preencha os requisitos e tenha em atenção o modelo de ataque;
3. **Provar que o compremetimento** da construção / sistema proposto **é aquivalente a resolver uma problema matemático reconhecidamente difícil e que lhe está subjacente**.

Este conjunto de técnicas, que formam no fundo o núcleo da criptografia, é constituído por **cifras**, **mecanismo de integridade** e de **troca de chaves** ou **segredos criptográficos**.

Criptanálise

> É constituida pelo **estudo de técnicas que visam** gorar os objetivos da Criptografia, isto é, **quebrar a segurança de comunicação**.

Criptologia

> **Conjuntamente**, a Criptografia e a Criptanálise **formam uma disciplina a que podemos chamar Criptologia**.

Criptografia Clássica

> A palavra Criptografia vem do grego *Kryptos* (oculto, segredo) e *graph* (escrita).

A chave de cifra dos Espartanos era o diâmetro dos bastões.

A principal necessidade para a criptografia era guerra.

Espartanos

Primeiros a usar criptografia por motivos bélicos.

Cifra de César (ROT)

Cifra Monoalfabética: usamos um só alfabeto e cada letra do texto-limpo é transformado sempre da mesma maneira numa mesma letra no criptograma.

Cifra de Vigenére

Cifra Polialfabética: porque uma mesma letra no texto-limpo pode ser transdormada em letras diferentes no criptograma conforme a sua posição na mensagem.

Cifra de Substituição

Cifra Monoalfabética (Ver **Cifra de César**)

Cifra Enigma

Cifra Polialfabética (Ver **Cifra de Vigenére**)

Modelos de Ataque a Cifras e a Primitivas Criptográficas

Brute Force

- Não é Criptanálise;
- É usado em último recurso;
- Normalmente não funciona com Criptografia moderna (se a chave for gerada aleatoriamente):
    - Chaves de 128 bit;

Ataque em que Apenas o Criptograma é Conhecido

Matematicamente é representado por um jogo:

1. O jogador manda duas mensagens para serem cifradas;
2. O jogador recebe uma das mensagens, agora cifrada;
3. O jogador ganha se conseguir advinhar qual é a mensagem original que resultou na mensagem que recebeu cifrada;

Ataques com Texto Limpo Original Escolhido

Neste ataque, o atacante pode escolher textos-limpos e pedir ao iniciante para os cifrar **antes** do jogo começar.

Ataque com Conhecimento de Parte do Texto-Limpo

Apenas é conhecida parte da mensagem cifrada.

Ataques com Texto Limpo Original Escolhido Adaptativamente

Neste ataque, o atacante pode escolher textos-limpos e pedir ao iniciante para os cifrar **antes** e **depois** do jogo começar.

Ciptografia de Chave Simétrica

Conceitos de Difusão e Confusão

Para que as cifras sejam boas têm de implementar dois conceitos:

- Confusão:
  - Quaisquer relações entre o texto-limpo e a chave não são notáveis no criptograma.
- Difusão:
  - Quando ciframos, tudo o que forem padrões no texto-limpo são disfarçados e espalhados por todo o criptograma.

As cifras modernas implementam estes conceitos.

One Time Pad

- Positivos:
  - Cifra com secretismo perfeito.
- Negativos:
  - A chave tem de ter o mesmo tamanho da mensagem.
  - A chave tem de ser gerada completamente aleatoriamente.

Outras Contribuições Interessantes neste Contexto

A segurança por obscurantismo não existe(*Kerckhoffs, 1883*).

Segundo Kerckhoffs, um sistema criptográfico é seguro se **tudo** (exceto a chave), inclusive o algoritmo e a sua descrição, forem conhecidos e mesmo assim, ninguém o conseguir quebrar.

Definição de Cifra de Chave Simétrica

Uma **Cifra** de Chave Simétrica é um par de algoritmos:

1. Algoritmo de Cifra: $c = E(k,m)$
2. Algoritmo de Decifra: $m = D(k,c)$

A palavra 'Simétrica' refere-se ao facto da chave usada para cifrar ser a **mesma** que é usada para cifrar.

Isto tem de ser verdade: D(k,E(k,m)) = m
Para **todas** as mensagens e chaves.

Cifra de Chave Simétrica Contínua

A cifra é continua porque à medida que vai crescendo o ficheiro este é cifrado de foram contínua.

Rivest Cipher 4 (RC4)

```
          128 bits
             |
             V
           RC4(k)
             |
             V
     bits pseudo aleatórios
```

Cifra de Chave Simétrica **por Blocos**

História

Feistel trabalhava na IBM nos anos 70. Fez uma cifra chamada Lucifer. Rede de Feistel

- chaves de 128 bits
- blocos de 64 bytes

Alterou o algoritmo para funcionar com 56 bits.
Resultou no *Data Encruption Standard* (DES)

Em 1997 um grupo de investigadores numa universidade juntaram várias *PlayStation* e correram muitas chaves de cifra de 56 bits.

A solução temporária foi de cifrar, decifrar e cifrar com 3 chaves diferentes.

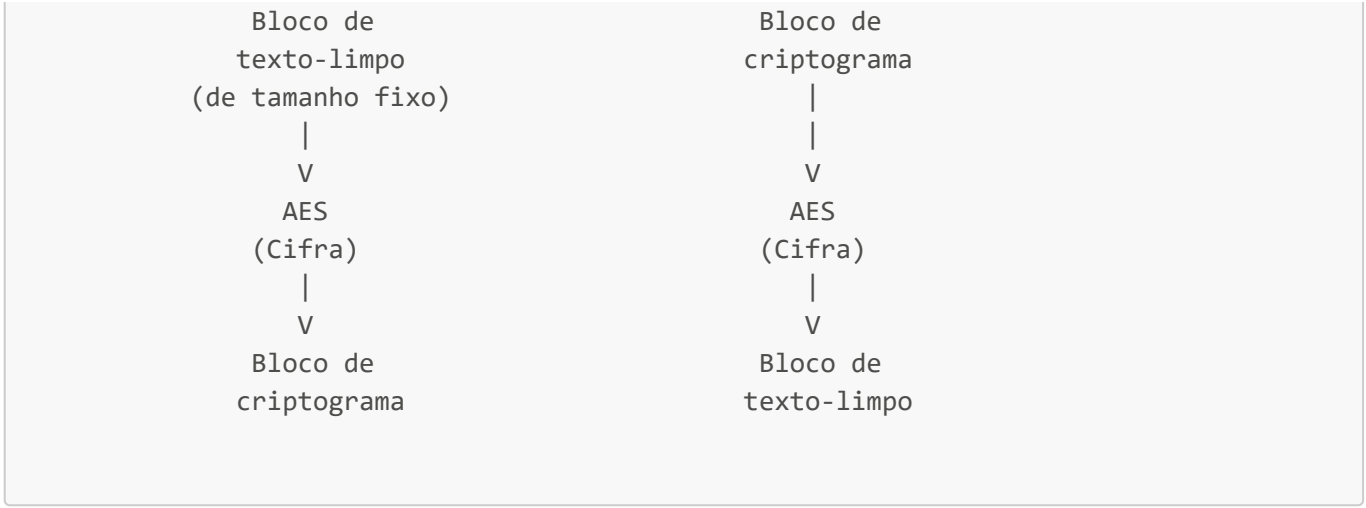
```
    E-DES(k3,D-DES(k2,E-DES(k1,M)))
```

Esta cifra, a 3DES, era de 168bits a custa de 3 vezes a *performance*

Mais tarde foi desenvolvido o novo *Encryption Standard* através de uma competição aberta que resultou na cifra *Advanced Encryption Standard* (AES).

Aspetos Gerais

```
            Chave                          Chave
              +                              +
```



As funções usadas nestas cifras (*Pseudo Random Permutations*) são funções que implementam os dois conceitos de Shannon para cifras de qualidade.

Como estas cifras operam por blocos, é preciso (infelizmente) saber como funcionam os modos.
Esses modos são:

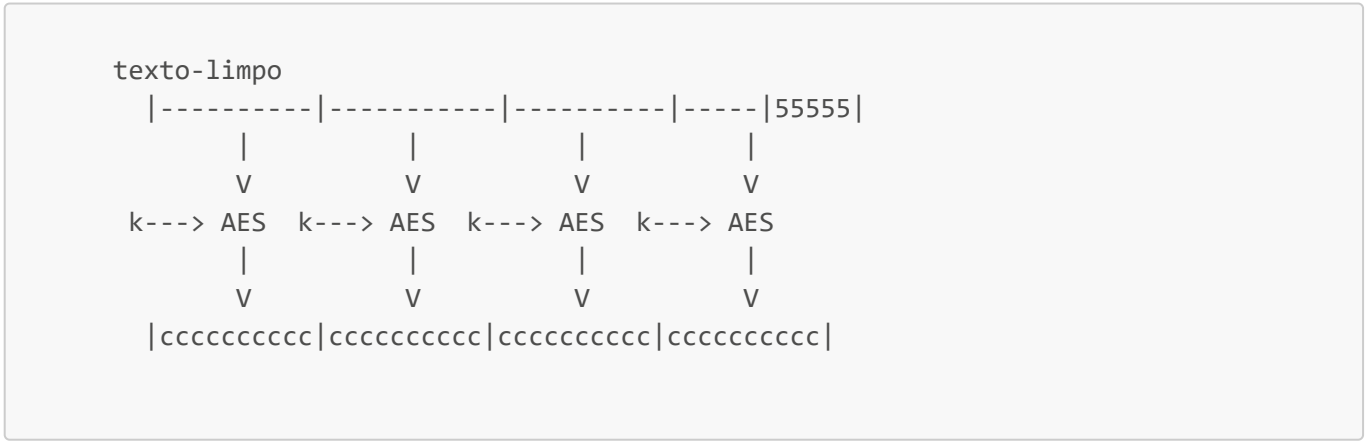
- *Electronic Code Book* (ECB)
- *Cipher Block Chaining* (CBC)
- *Counter Mode* (CTR)

Sempre que se utiliza uma cifra de Chave Simétrica por blocos nos modos CBC e ECB, é obrigatório usar *padding*.

Electronic Code Book

O Electronic Code Book(ECB) é o modo mais simples de usar uma cifra de chave Simétrica por blocos.

Apesar de ser o modo mais simples de usar uma cifra de chave simétrica por blocos não se deve usar ECB.



O *padding* mais utilizado é ***PKCS1.5(Public Key Criptography Standard) padding***. Este é composto por valores iguais ao tamanho que falta para completar um bloco. Caso o último bloco venha cheio adiciona-se um bloco só de *padding*.

O problema do ECB é que se existirem dois (ou mais) blocos de texto-limpo então o criptograma também conterá também blocos iguais.

Um caso onde se pode utilizar o ECB sem risco de comprometer informação é utilizando esta cifra em texto-limpo mais curto que um bloco

Cipher Block Chaining

Este modo supera o problema de dois blocos iguais darem exatamente os mesmos blocos de criptograma.

- O circuito de cifra diz que é preciso cifrar o que está para trás para cifrar determinado bloco.
- O circuito de decifrar permite aceder a determinado bloco diretamente (usando o criptograma respetivo e o anterior).

Um dos grandes problemas desta cifra é a falta de suporte para processamento em paralelo pois a cifra de um bloco é dependente da cifra do bloco anterior.

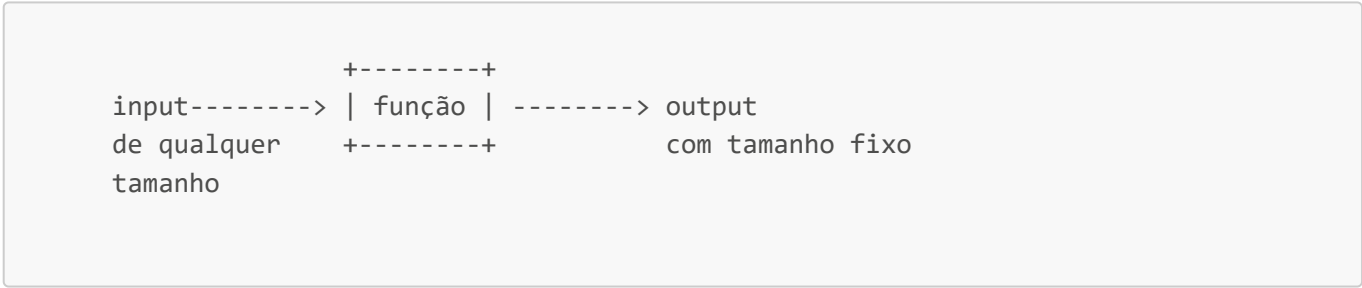
Counter Mode

Um dos modos mais usados no planeta Terra.
Modo ideal para ir cifrando e decifrando à medida que os dados vão sendo gerados ou recebidos.

O CTR transforma a Cifra de Chave Simétrica por Blocos numa Cifra de Chave Simétrica Contínua.

Funções de *Hash* Criptográficas

Integridade



Propriadades de Funções de *Hash* sem serem Criptográficas

- São fáceis de computar no sentido direto (podem ter inverção);
- Dado um *input* de qualquer tamanho, devolvem sempre um *output* com tamanho fixo (que é sempre o mesmo).

Propriadades de Funções de *Hash* Criptográficas

- Resistência a Colisões.

É praticamente impossível encontrar dois ficheiros com o mesmo valor de *hash*.
O tamanho do *output* das funções de *hash* nunca pode ser inferior a 128 bits na modernidade.

Esta propriadade é a mais fácil de abordar (mas ainda impraticável em funções de *hash* modernas) das 3 prorpiedades, graças ao chamado paradoxo

- Resistência à Descoberta de uma Mensagem Original.

Se tivermos um valor de *hash*, não conseguimos descobrir, em tempo útil, qual a mensagem que lhe deu origem.
O que sabemos? Um valor de *hash*.

- Resistência à Descoberta de uma Segunda Mensagem Original.

Se tivermos um ficheiro e o valor de *hash*, não conseguimos encontrar, em tempo útil, outro ficheiro com o mesmo valor de *hash*.

Exemplos Canónicos

- MD5 (desencorajada, perdeu o estatuto), devolve 128 bits
- SHA1 (desencorajada), devolve 160 bits
- SHA256, devolve 256 bits
- SHA512, devolve 512 bits
- SHA3, it does stuff

Utilidades das Funções de *Hash*

```
Alice               Micro-ondas               Bob
  | ola! ----------~~~~~~~~~~~~~----------> ole! |

Alice               Micro-ondas               Bob
  | ola!                                      |
  |SHA256(ola!) ----~~~~~~~~~~~~~~------> oli! h1 |
                                    SHA256(oli!) = h2
                                    Se h1 != h2
                                       descarta a mensagem
```

As funções de *hash* podem **então** ser usadas para proteger a integridade de mensagens em casos de erros **aleatórios**.

Códigos de Autenticação de Mensagens

- As cifras garantem a confidencialidade;
- As funções de *hash* garantem a integridade em casos de erros aleatórios;
- Os *message authentication codes* garantem a integridade em casos de modificações intencionais.

Construções dos MACs

Códigos de Autenticação de Mensagens Códigos de Autenticação da Origem da Informação

Que construções existem? Muitas.

A mais popular é a **HMAC** - *Hash based Message Authentication Code*

```
    HMAC-SHA256(k,m) = SHA256(k || SHA256(k || m))
```

A fórmula anterior produz um código de tamanho 256bits

```
    HMAC-SHA1(k,m) = SHA1(k xOR mascara_externa || SHA1(k xOR mascara_interna ||
m))
```

A fórmula anterior produz um código de tamanho 160bits

ECBC-MAC: *Encrypted Cipher Block Chaining* MAC

Para o cálculo de um ECBC-MAC são usadas 2 (ou talvez 1 se for 256bits para a podermos repartir em 2) chaves.

Se for usado AES para o cálculo de ECBC-MAC, o MAC terá 128bits.

MAC + Cifra, como Combinar

Imaginemos que temos uma mensagem e queremos:

- cifrar;
- calcular o MAC.

Hipótese 1 (*MAC-and_Encrupt*):
Calcular primeiro o MAC: t = MAC(ik,m) Ciframos a mensagem: c = E(ek, m) Juntamos as duas coisas: (c, t)

Não é a mais segura,mas é usada no SSH.

Hipótese 2 (*MAC-then-Encrypt*): Calcular primeiro o MAC: t = MAC (ik, m) Juntamos o MAC com a mensagem e ciframos as duas coisas: c = E(ek, m t)

Não é a mais segura, mas é usada no TLS.

Hipótese 3 (*Encrypt-then-MAC*): Ciframos a mensagem: c = E(ek,m) Calculamos o MAC do criptograma: t = MAC(ik, c) Juntamos as duas coisas: (c, t)

Mais segura. Sempre correta. Usada no IPSec.

Códigos de Autenticação de Mensagens

...

Esquemas de Distribuição de Chaves

...

Criptografia de Chave Pública

Definição de Chave Pública

> Uma **Cifra** de Chave Simétrica é um terno (3) de algorimos:

1. *c = E(pk,m)*
2. *m = D(sk, c)*

A expressão 'chave pública' refere-se ao facto da chave esada para cifrar ser **diferente** da que é usada para decifrar.

Isto tem de ser verdade: D(sk, E(pk, m)) = m

Contrução *Merkle-Damgard*

Forma engenhosa de construir funções de *hash*

```
   m1      m2      ...                    mn
|-------|-------|-------|-------|-------|-------|

g(m,h) --> h

g(m1,s) --> h1
g(m2,h1) -> h2
g(m3,h2) -> h3
  ...
        hn
        é o valor de *hash*.
```

SHA1, MD5, SHA usa esta construção

Problemas Intratáveis

Há problemas matemáticos que se acreditam serem de intratável resolução. Significa que não há algoritmos eficientes (que nos deem em tempo útil uma resposta) para resolver um determinado problema.

Isto significa que, no mínimo, precisamos de mais do que 2^80 iterações para resolver o problema.

Problema do Logaritmo Discreto

Se tiverem um número primo P extremamente grande ($>$ 2^1024), portanto com mais do que 1024bits, um gerador g para o grupo ZP gerar um x aleatório entre 1 e P

É fácil calcular

```
X = g^x mod P
```

É intratável obter o x (pequeno) sabendo o X, g e o P.

Problema da Fatorização de Números Primos

Se eu arranjar dois números primos grandes ($>$ 2^1024), portanto com mais de 1024bits.

Protocolos de Acordo de Chaves

Protocolo de Acordo de Chaves Diffie-Hellman

Baseado no Problema do Logaritmo Discreto. Servem para trocar chaves de cifra simétricas, mas é um mecanismo da Criptografia de Chave Pública.

```
Alice                                                Bob
 g,P ------------------------------------------------> g,P
 gera 1<x<P                                          gera 1<y<P
 X=g^x mod P                                         Y=g^y mod P
 X------------------------------------------------->X
 Y<-------------------------------------------------Y
 K=Y^x mod P                                         K=X^y mod p
 =(g^y mod P)^x mod P                                = (g^x)^y mod P
 =g^(y*x) mod P                                      = g^(x*y) mod P

                          Claire
                           g,P
                           X,Y
```

No final, a Alice e o Bob calculam K. É um número grande que parece aleatório e pode ser usado como chave de cifra ou de integridade em MACs. Só funciona em cenários de ataque de homem no meio **passivo**.

```
Alice                                                Bob
g,P
gera 1<x<P
X=g^x mod P
Z
K1 = Z^x mod P

                          Claire
                          g,P
                          X
                          gera 1<z<P
                          Z = g^z mod P
                          K1 = X^z mod P

                          gera 1<w<P
                                                     g,P
                                                     gera 1<y<P
                                                     Y=g^y mod P
                                                     W
                                                     K2 = W^y mod P
                          W=g^w mod P
```

```
                                Y
                                K2=Y^w mod P
```

A Alice cifra as mensagens todas com K1 e envia para o Bob. A Claire captura e decifra todas as mensagens. Mais, se a Claire quiser, cifra novamente as mensagens com K2 e envia-as para o Bob.

Puzzels de Merkel

Servem para trocar chaves de cifra simétricas, e são esquemas da criptografia de chave simétrica.

```
    Alice                                                   Bob
    puzzle x = 1    k1
    puzzle y = 2    k2
    ...         3   k3
    ...  ...  ...   ...
    puzzle w = 2^32 k2^32
    AES(puzzle x, 0000000000000000 | K)
    A Alice faz 2^32 cifras
    A Alice envia todas as cifras ao Bob.                   O Bob recebe
2^32puzzles

                            Claire
                            A Claire também recebe todos os 2^32 puzzles.

                                                            O Bob tenta decifrar 1
puzzle.
                                                            Tenta todas as chaves.
                                                            Ao decifrar um puzzle
encontra um indice e uma chave.
                                                            Envia o indice para
indicar a chave.
```

RSA

Rivest Shamir Adleman(RSA).

Baseada no Problema da Fatorização em Números Primos. É uma cifra de chave pública.

Definição: (Uma cifra de chave pública tem 3 algoritmos)

1. Algoritmo para Cifrar; $c <-- RSA(pk,m) = m^pk \mod N$
2. Algoritmo para Decifrar; $m <-- RSA(sk,c) = c^sk \mod N$
3. Algoritmo para Gerar Chaves:
    1. Gerar dois números primos muito grandes p,q
    2. Calcular $N = p \times q$ $\phi(N) = (p-1) \times (q-1)$
    3. Tentamos encontrar dois números pk e sk tal que $pk \times sk \mod \phi(N) = 1$

4. Devolvemos (pk,N)

ElGamal

Baseada no Problema do Logaritmo Discreto. É uma cifra de chave pública.

Cifras de Chave Pública

...

Assinatura Digital

m RSA(sk, SHA256(m))

Certificados Digitais

Como estabelecer confiança de que uma chave pública é de uma entidade.

- Identificação <----> Chave Pública

Uma entidade omnipresente era capaz de passar documentos que diziam

```
+--------------------------------+
| "f55fd3ffd" é do Pedro         |
|                                |
|                   A entidade   |
|                  da123fawe23d  |
+--------------------------------+
```

Considerem que já tinham a chave pública daquela entidade no computador. Cada vez que recebiam nova chave, bastava receberem também um certificado para imediatamente poderem verificar se a chave pertence ao utilizador ou não.

Enters PKI

```
+---------------------------------+
| O Rui tem esta chave pública: PK |
|                                 |
| Assinado                        |
| Pedro                           |
| sad8a78s7a89d7987a8sde          |
+---------------------------------+
```

Cada vez que alguém queria a chave do Rui, ia buscar aquele certificado, e usava a chave do Prof. para ver se o certificado era verdadeiro. Só depois é que usva a chave do Rui.

A outra pessoa do outro curso conhecia outro Professor e tinha a chave dele.

É preciso uma entidade superior para passar certificados. Neste cenário seria o reitor a passar certificados aos professores, que por sua vez passam certificados aos alunos.

A *International Telecommunications union*(ITU) e a *Internet Enginnering Task Force*(IETF) são as organizações que regulam o uso da infrastrutura de chave pública nas telecomunicações e Internet.

As chaves públicas são guardadas dentro de documentos chamados **certificados**.

Obtem-se uma chave pública por conta própria ou pedindo a uma autoridade de certificação.

A própria natureza do certificado faz com que a sua distribuição não necessite de precauções.

O Cartão de Cidadão(CC) contém um certificado digital X.509 para cada cidadão português.

Um certificado digital inclui um prazo de validade.

Se alguém comprometer uma chave privada antes da validade acabar, o certificado é adicionada a Lista de Revogação de Certificados.

O Certificado X.509

Assumimos que a AC verrificou a identidade e a pertença da chave pública **antes** de assinar o certificado.

O certificado liga uma chave pública a uma entidade.

O certificado não pode ser falsificado, porque só as autoridades de certificação

---

PKI

A Infraestrutura de Chave Pública (*Public Key Infrastructure*(PKI)) é o conjunto de *hardware*, *software*, pessoas, políticas e procedimentos necessários à criação, gestão, distribuição, utilização, armazenamento e revogação de certificados digitais.

A PKI é composta por 5 entidades:

1. **Entidade terminal ou Cliente**
2. more stuff
3. **Autoridades de Registo Organizacional(ARO)** - um sistema opcional ao qual as ACs delegam certas funções de gestão.
4. **Emissor de Listas de Revogação de Certificados(LRC)**
5. **Repositório**

Protocolos da PKI

1. **Protocolos Operacionais**
2. **Protocolos de Gestão**:
    1. **registo**;
    2. **Inicialização**;
    3. **Certificação**;
    4. **Recuperação / atualização de chaves**;
    5. ***Pedido de Revogação**;

6. **Certificação Mútua entre ACs**.

Pretty Good Privacy(PGP)

- É uma forma de trazer criptografia às pessoas.
- Foi desenvolvido por Phil Zimmerman.
- GPG é uma implementação *opensource* para *Unix* ou *Unix like* do PGP.
- É possível instalar *plugins* para clientes de e-mail.

Quando um utilizador inicializa o PGP, gera **normalmente** 4 chaves:

- 2 pares de chaves:
  - 1 par de chaves (pública e privada) para autenticação (assinatura digital);
  - 1 par de chaves (pública e privada) para cifra (confidencialidade).
- Geram-se 2 chaveiros:
  - Privado, que guarda as chaves privadas;
  - Público, que guarda as chaves pública.

Privado (está cifrado por uma *password*)

ID	**chave**
primeiros_pk1	BASE64(sk1)
primeiros_pk2	BASE64(sk2)

Público

ID	**PK**	**Legitimidade**	**Certificados**	**Confiança**	**Nome**
primeiros_pk1	BASE64(pk1)	TOTAL	----------	TOTAL	Pedro
primeiros_pk2	BASE64(pk2)	TOTAL	----------	TOTAL	Pedro
primeiros_pkx	BASE64(pkx)	TOTAL	Pedro	Trusted Introducer	Manel
primeiros_pky	BASE64(pky)	TOTAL	Pedro	Trusted Introducer	Manel
...	...	...	...	...	...
primeiros_pkw	BASE64(pkw)	TOTAL	Manel	Marginal	Rita
primeiros_pkz	BASE64(pkz)	TOTAL	Manel	Marginal	Rita
...	...	...	...	...	...
primeiros_pk3	BASE64(pk3)	TOTAL		Marginal	Rui
primeiros_pk4	BASE64(pk4)	TOTAL		Marginal	Rui
...	...	...	...	...	...
primeiros_pk5	BASE64(pk5)	TOTAL		Marginal	Carlos

ID	PK	Legitimidade	Certificados	Confiança	Nome
primeiros_pk6	BASE64(pk6)	TOTAL		Marginal	Carlos
...	...	...	...	...	...
primeiros_pk7	BASE64(pk7)	TOTAL	Rita, Rui, Carlos		Ana
primeiros_pk8	BASE64(pk8)	TOTAL	Rita, Rui, Carlos		Ana

Os níveis de Confiança são:

- Sem Confiança;
- Marginal;
- *Trusted Introducer*.

Confiança em PGP

Há 3 formas de establecer confiança em PGP:

- Contacto direto:
    - Eu assino uma chave pública com a minha chave privada em sinal de que sei que a chave é, de facto, daquela pessoa.
- É recebermos uma chave assinada por outra pessoa que nós conhecemos e em quem temos confiança absoluta (*Trusted Introducer*);
- Se recerbermos 1 chave assinada por 3 pessoas que conhecemos e em quem confiamos marginalmente (*Web of Trust*):
    - Para enviar a mensagem:
        1. Comprime a mensagem;
        - m
        2. Gera uma chave de 128 bits AES k;
            - $c1 = AES(k, m')$;
        3. Cifra a chave k com RSA;
            - $c2 = RSA(pkx, k)$;
        4. Calcula a assinatura digital:
        5. (pede *password*);
        6. $s = RSA(sk1, SHA256(m))$;
        7. Codifica tudo em BASE64;
            - $c1-64$, $c2-64$, $s-64$;
    - Para ler a mensagem recebida:
        1. Recebe $c1-64$, $c2-64$, $s-64$;
        2. Descodifica a chave simétrica:
            1. $sk = RSA^{-1}(skx, c2)$;
            2. pede *password*;
        3. Agora pode decifrar o criptograma c1:
            - $m' = AES^{-1}(k, c1)$
        4. Descomprime o m'
            - m
        5. Verifica-se a assinatura digital com a chave pública de quem mandou a mensagem.

Se se perder as chaves ou desconfiar que alguém as roubou:

Cartão de Cidadão da República Portuguesa

- O PIN de autenticação tem 4 dígitos (introduzido até 3 vezes);
- O PIN de morada tem 4 dígitos (introduzido até 3 vezes);
- O PIN de assinatura digital tem 4 dígitos (introduzido até 3 vezes).

- O PUK de autenticação tem 4 dígitos;
- O PUK da morada tem 4 dígitos;
- O PUK da assinatura dígital tem 4 dígitos.

- A ativação do *smartcard* tem 8 dígitos;
- Ativação de assinatura digital tem 4 dígitos;
- O PIN de cancelamento do cartão tem 8 dígitos.