Autenticação: mecanismos e protocolos

Autenticação (Authn)

Provar que uma entidade possui um atributo que diz ter

- **Identidade**
  1. E: Olá, sou o João
  2. A: Prova-o
  3. E: Aqui estão as minhas credenciais
  4. A: Credenciais aceites/recusadas

- **Propriedade**
  1. E: Olá, tenho mais de 18 anos
  2. A: Prova-o
  3. E: Aqui está a prova
  4. A: 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 (2FA)**
  - Utilização simultânea de diferentes tipos
  - 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

USE CASE

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**
  - Apresentar credenciais
  - Esperar pelo veredicto

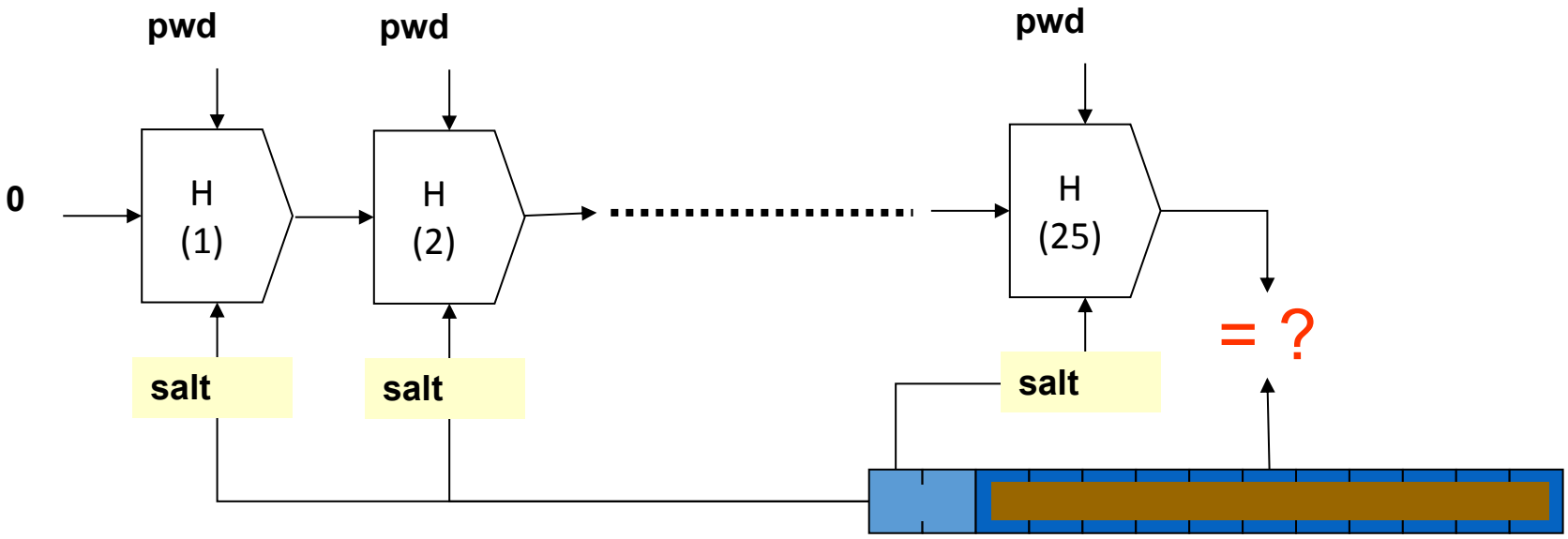
- **Aproximação com desafio-resposta**
  - Obter desafio
  - Calcular e fornecer uma resposta calculada com base no desafio e nas credenciais
  - Esperar pelo veredicto

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**

Sujeitos: Aproximação Direta com Senha Memorizada



Tradicionalmente em Unix, H=DES

DES *hash* = $DES_{pwd}^{25}(0)$

$DES_k^n(x) = DES_k(DES_k^{n-1}(x))$

Permutação de 12 subchaves com sal (12 bits)

O mesmo método pode ser utilizado com outra cifra (ex, AES)

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

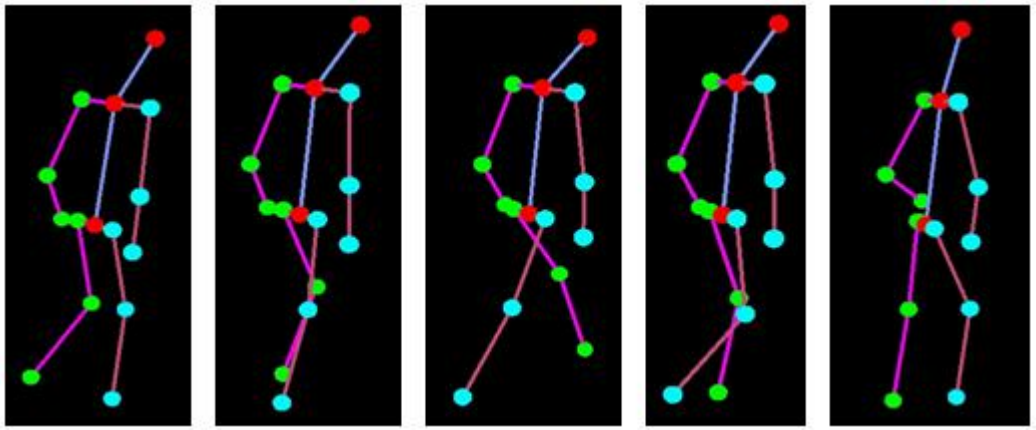
1. 123456
2. Password
3. 12345678
4. qwerty
5. 12345
6. 123456789
7. letmein
8. 1234567
9. football
10. iloveyou

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
  - 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

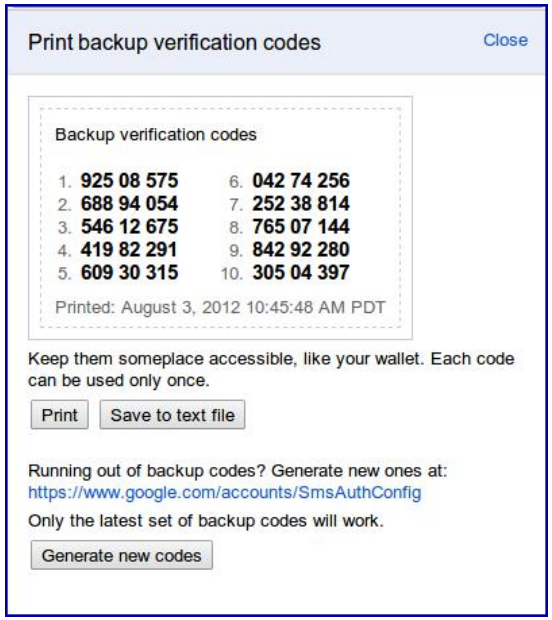
- **Alguns métodos ainda são incipientes**
  - Podendo ser ultrapassados com facilidade (ex, Reconhecimento Facial)
- **Sujeitos não podem alterar as credenciais**
  - A exposição das credenciais tem impacto duradouro
- **Credenciais não podem ser transferidas a outros**
  - Por vezes necessário em situações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

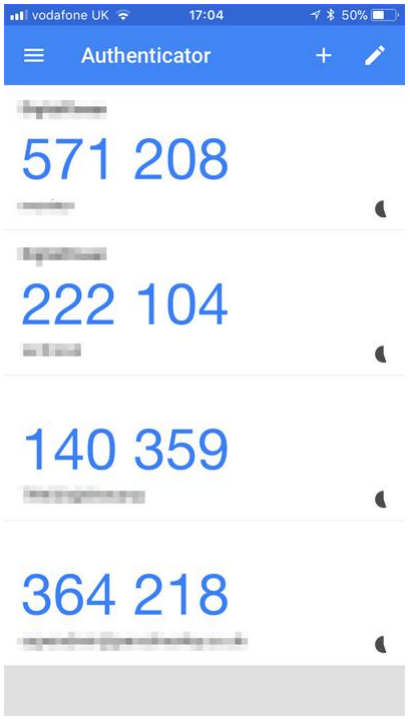
Sujeitos: Aproximação direta com Biometria: Desvantagens

- **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





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 a 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**
  - OTP = UserID | 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 carateres**
  - Emula um teclado USB
  - Suporta HOTP (Eventos) ou TOTP (Temporal)
  - Se for fornecido um desafio, utilizador tem de tocar no botão para que o resultado seja fornecido
  - AES 128

cccjgjgkhcbbirdrfdnlnghhfgrtnnlgedjlftrbdeut

cccccbcgujhingjrdejhgfnuetrgigvejhhgbkugded

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

cccccbcgujh | ingjrdejhgfnuetrgigvejhhgbkugded

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

Encrypted One Time Passcode

YubiKEY ID | Unique Passcode | Counter

Yubico Server

Match ID to Server

Decrypt Token With Key

New Counter > Server Value

YubiKey OTP Validated

User ID | User AES Key | User Counter

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

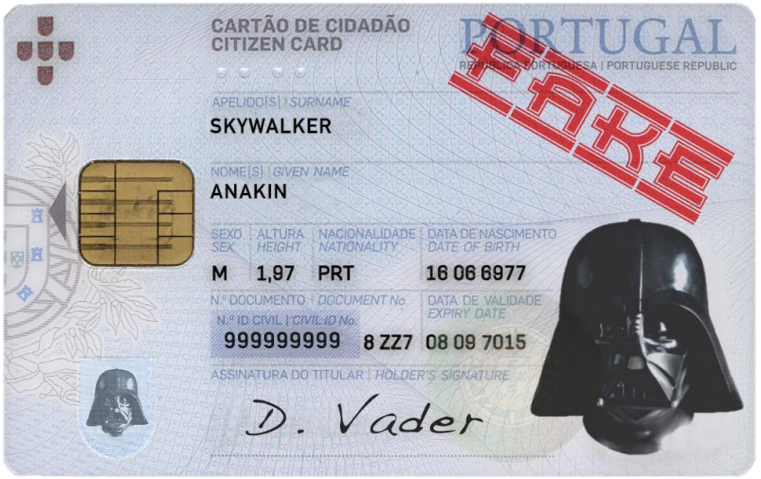
- **Credenciais não são expostas**
  - Nunca circulam no canal de comunicação
  - Circula uma transformação da credencial
- **Robustas contra ataques de MITM**
  - Atacante captura desafio e resultado mas não consegue replicar a transformação
- **Compatíveis com outras aproximações**
  - Dispositivos 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 noutros 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(desafio || password)
   - ou... resultado = E_k(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
```

MS-CHAP (Microsoft CHAP) (RFC 2433, 1998, RFC 2759, 2000)

Versão 1

A → U: authID, C

U → A: R

A → U: OK/not OK

$R = DES_{PH}(C)$

$PH = LM_{PH}$ or NT_{PH}

LM_{PH} = DEShash(password)

NT_{PH} = password

Versão 2

A → U: authID, C_A ← m1

U → A: C_U , Ru ← m2

A → U: OK/not OK, R_A

$RU = DES_{PH}(C)$

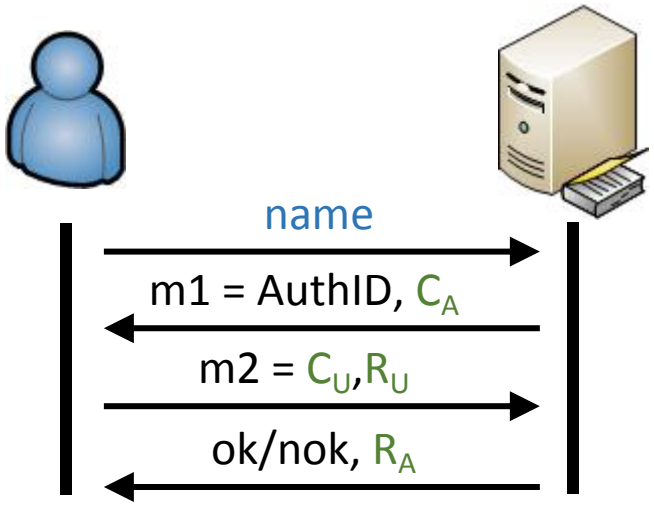
$C = SHA(C_U, C_A, username)$

$PH = MD4(password)$

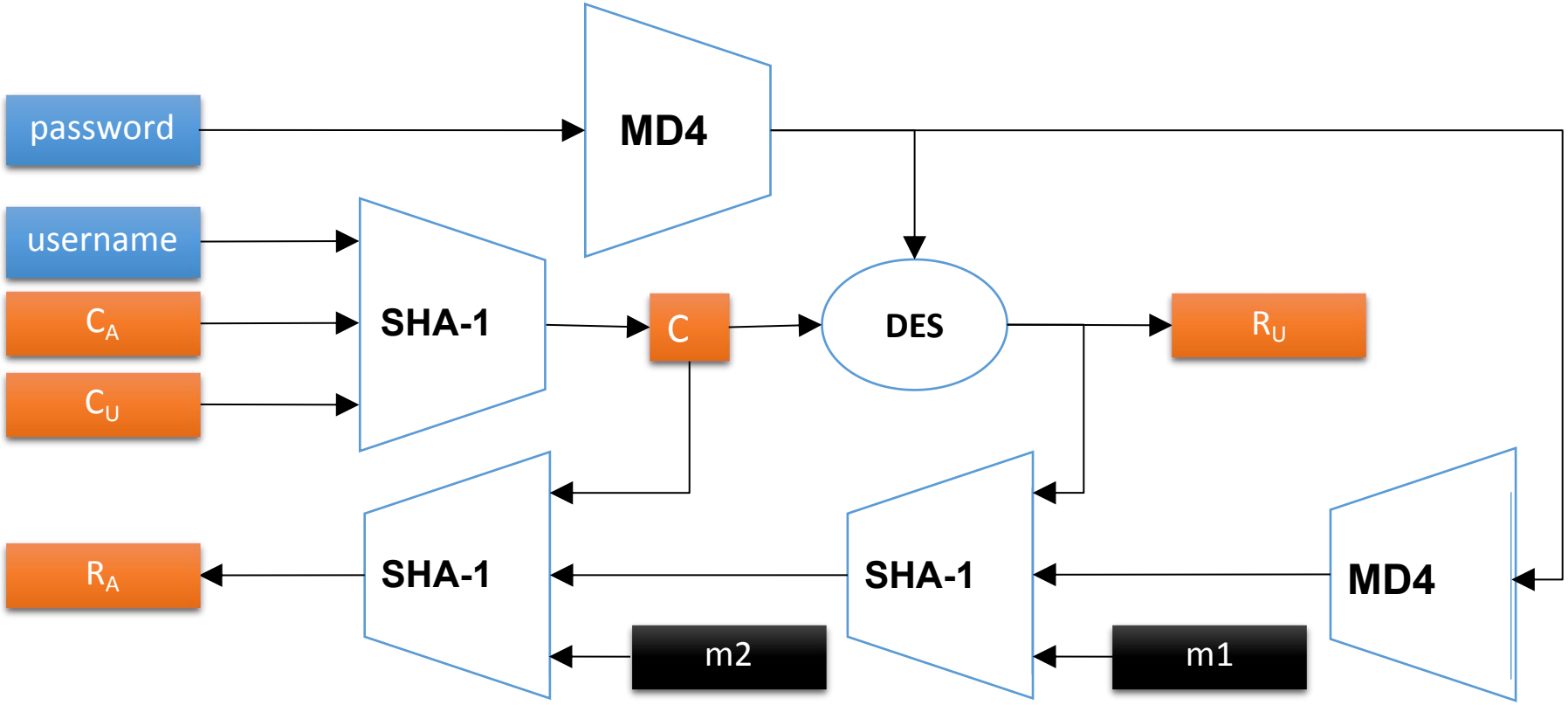
$R_A = SHA(SHA(MD4(PH), Ru, m1), C, m2)$

- Autenticação Mútua.
- Senhas podem ser alteradas

MS-CHAP (Microsoft CHAP) (RFC 2433, 1998, RFC 2759, 2000)



MS-CHAP (Microsoft CHAP) (RFC 2433, 1998, RFC 2759, 2000)



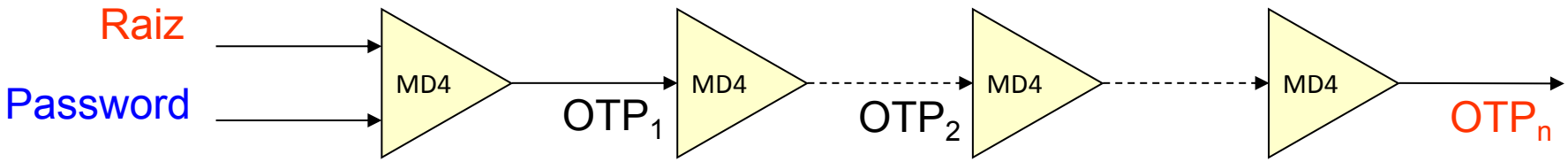
S/Key (RFC 2289, 1998)

- **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:
     - $OTP_n = H_n(raiz, password)$, 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 = $OTP_{indice-1}$

- **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

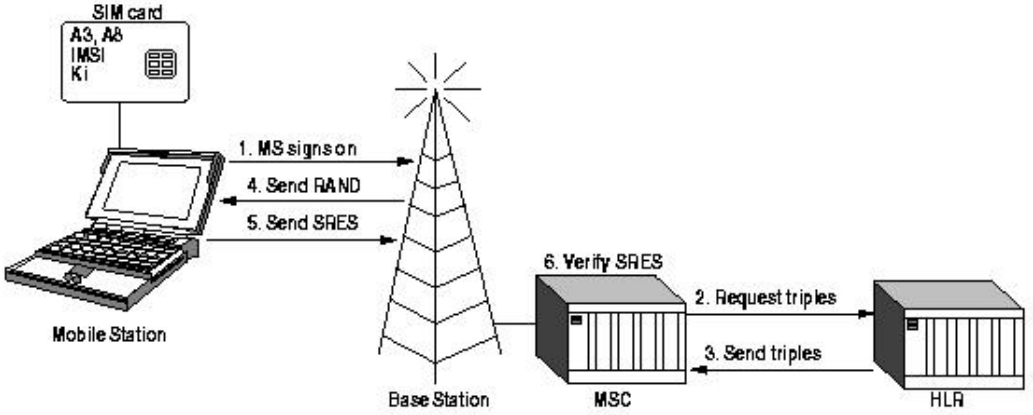
- **Baseado num segredo partilhado entre o HLR e o subscritor**
  - Utiliza uma chave simétrica de 128 bits, denominada de Ki
  - Ki encontra-se no Subscriber Identification Module (SIM)
  - Smartcard fornece respostas baseadas na Ki

- **Algoritmos (inicialmente desconhecidos):**
  - Autenticação: A3
  - Geração da chave de sessão: A8
  - Comunicação: A5 (cifra contínua)

- **A3 e A(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

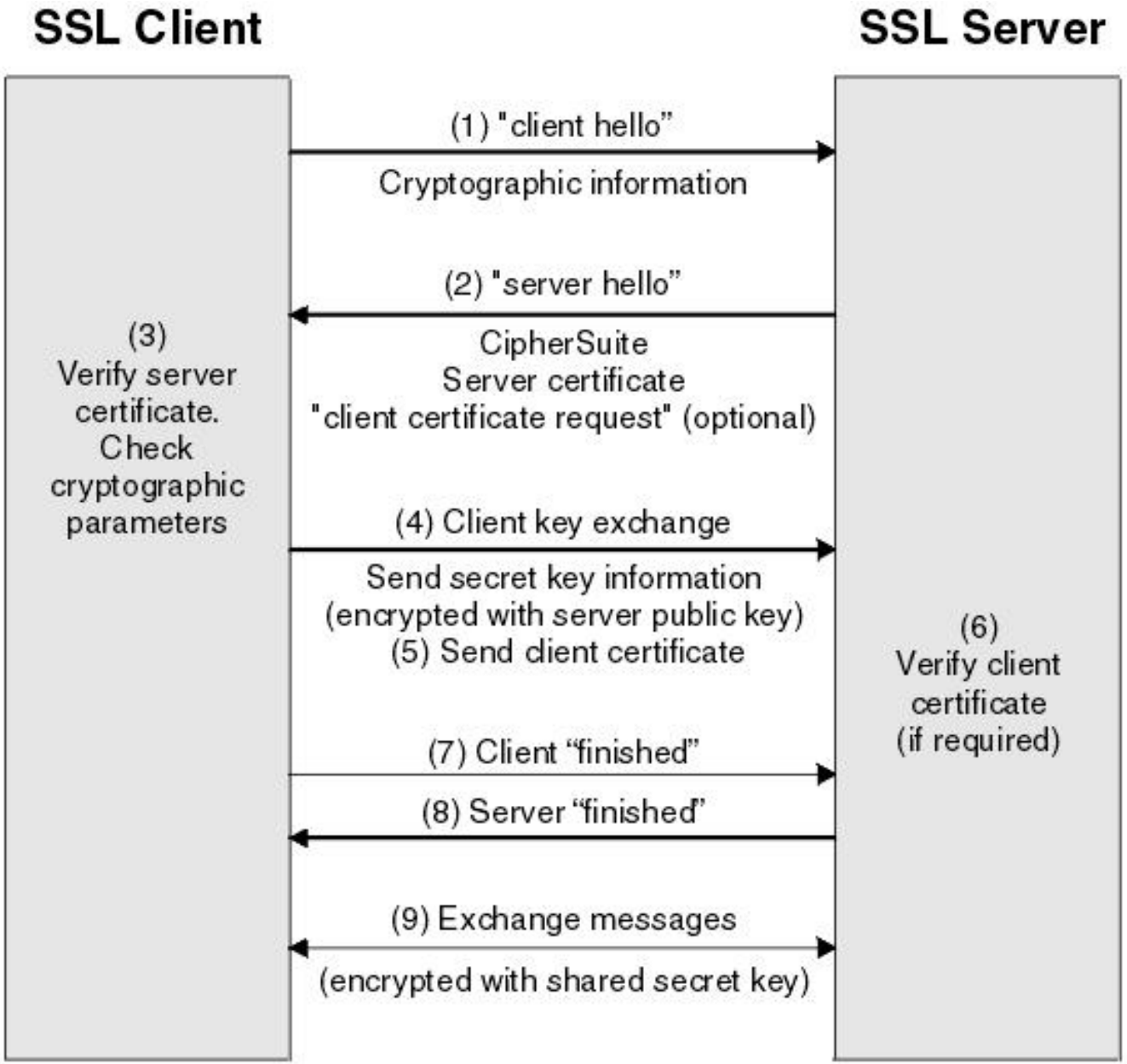
- **Autenticação do Sistema**
  - Todos os serviços localizados no mesmo sistema são automaticamente autenticados

- **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
(1) "client hello"
Cryptographic information
(2) "server hello"
CipherSuite
Server certificate
"client certificate request" (optional)
(3)
Verify server
certificate.
Check
cryptographic
parameters
(4) Client key exchange
Send secret key information
(encrypted with server public key)
(5) Send client certificate
(6)
Verify client
certificate
(if required)
(7) Client "finished"
(8) Server "finished"
(9) Exchange messages
(encrypted with shared secret key)

Fonte: IBM

TLS Ciphersuites

- **A noção de ciphersuites é o que permite a negociação de mecanismos entre clientes e servidores**
  - Ambos enviam as suas ciphersuites, usando uma partilhada por ambos
  - 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 Mechs

- **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/vYxuqqgrrZE45Gh5GqJeRPg6nFwrc+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

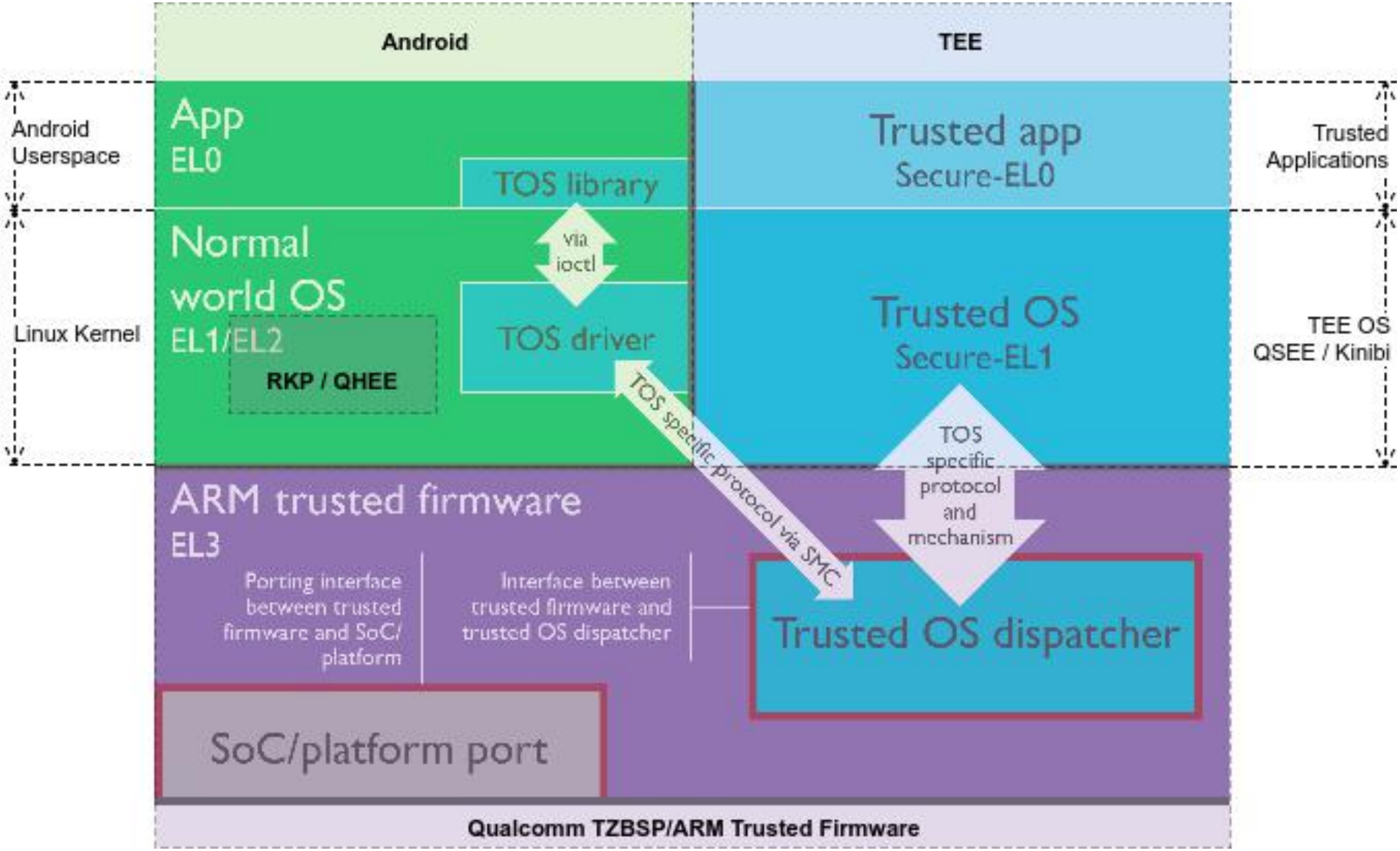
Dispositivos comuns

- **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

Dispositivos móveis: Smartphones



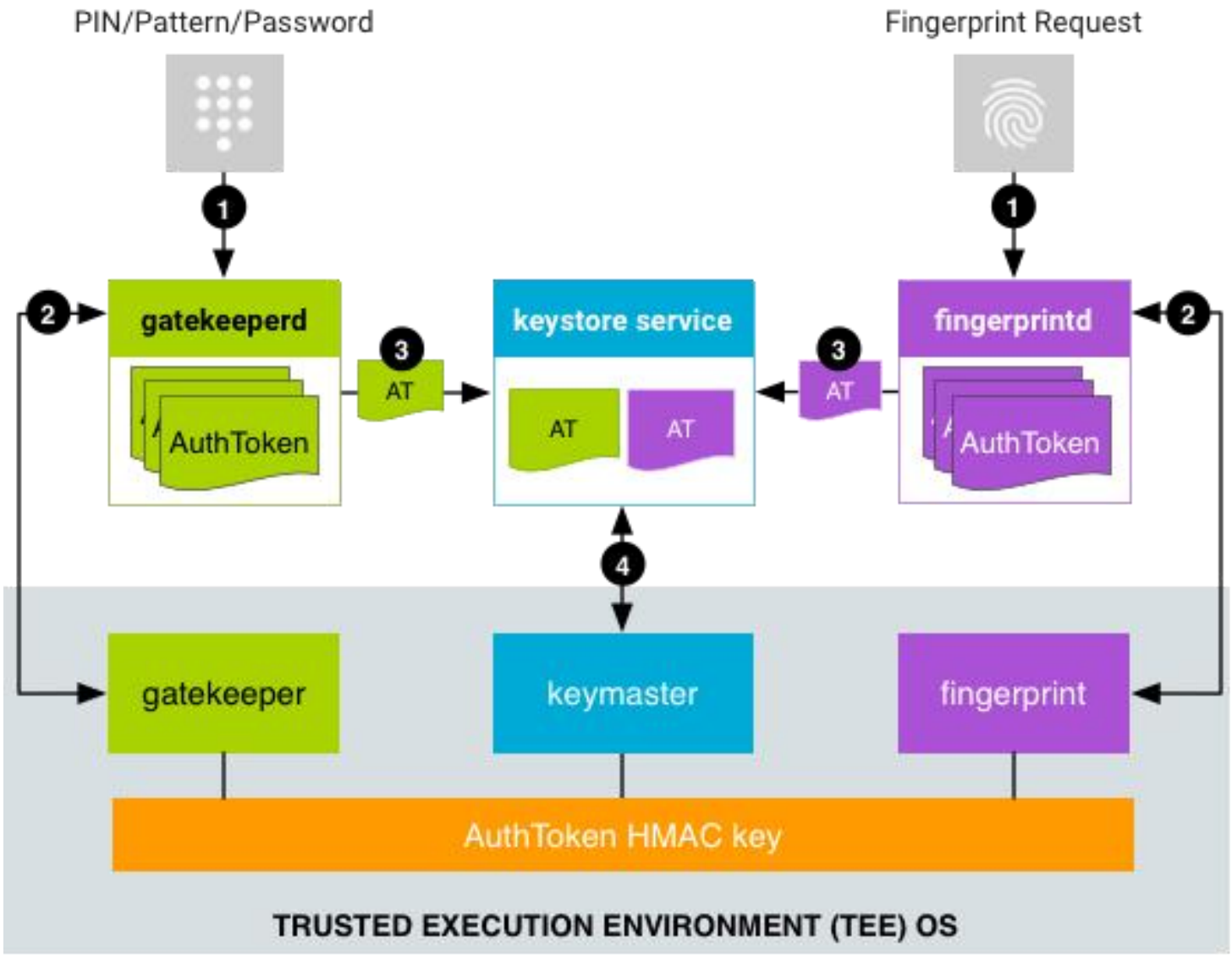
Smartphones: Android

- **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

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=HMAC_SHA256(instante temporal || AppID || R, HBK)
  - 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
- **Imagem é 2D**
  - Fácil forjar credenciais
  - Modelos, impressões
- **Apenas usado em versões agora obsoletas**
- **Usado em autenticação de edifícios**

An optical sensor.

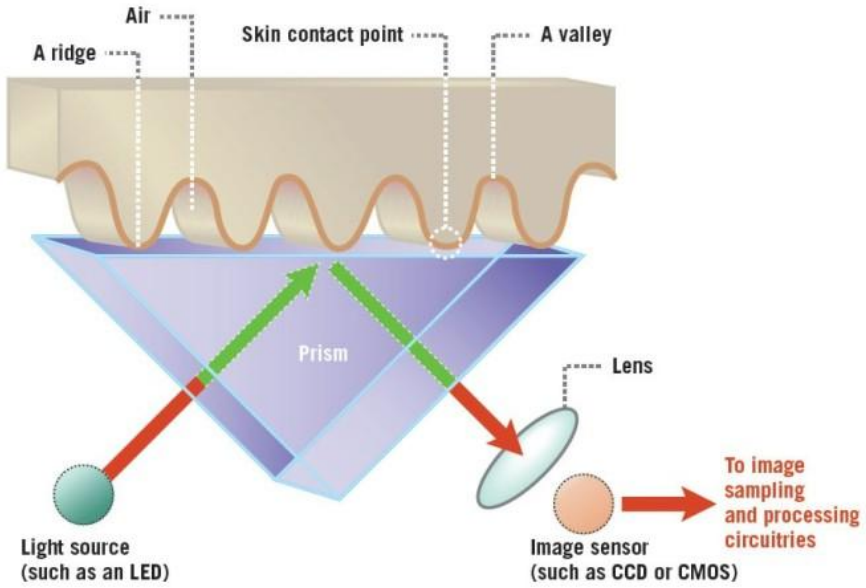
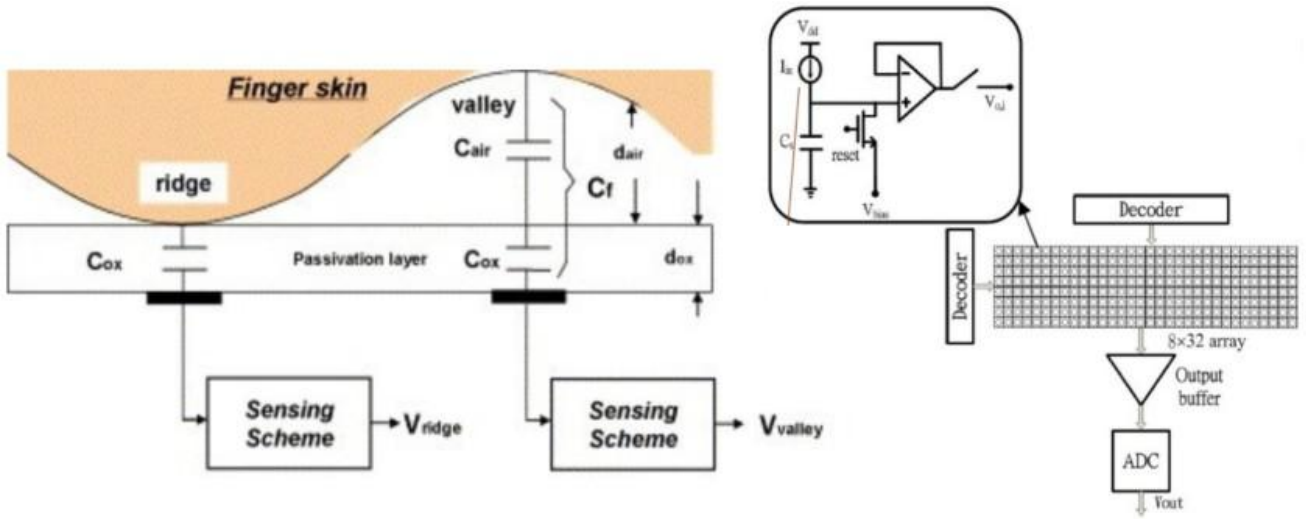


Figure 2

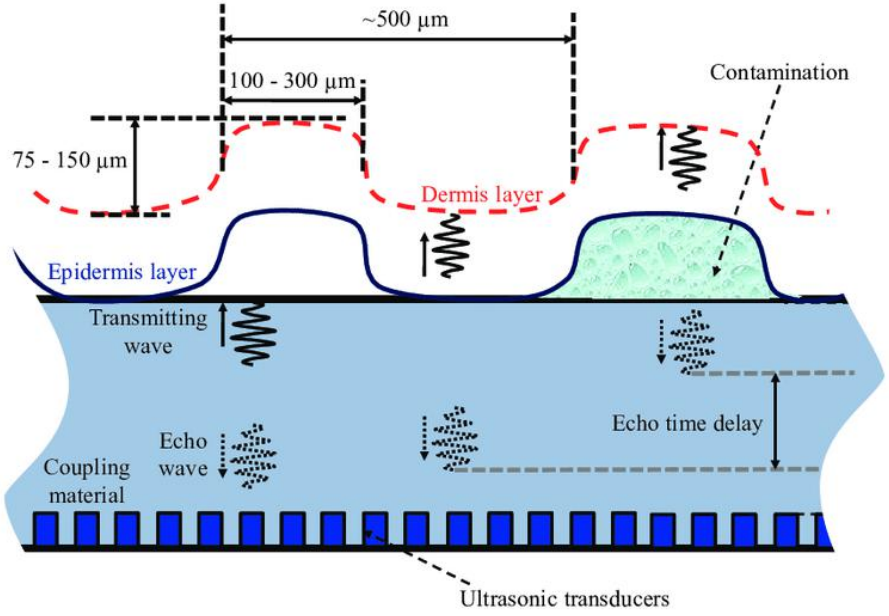
Impressões Digitais: Leitores Capacitivos

- **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



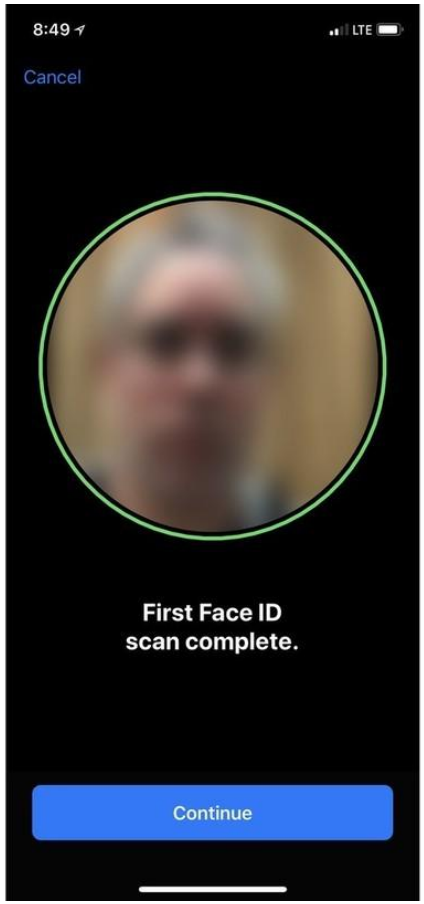
Smartphones: Reconhecimento Facial

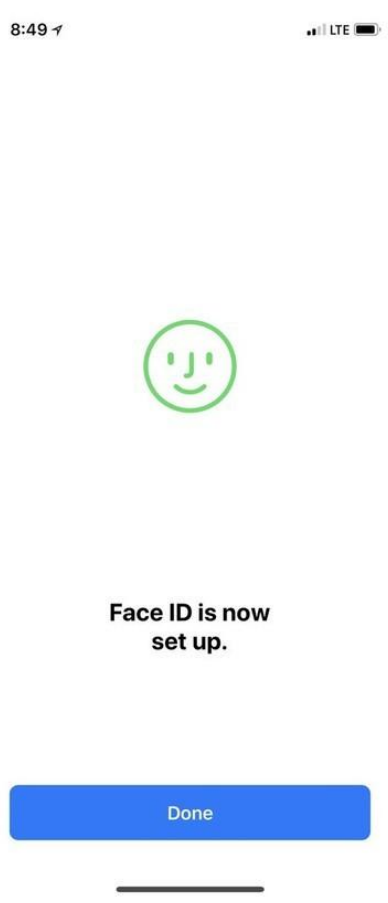
- **Objetivo: Verificar a correspondência entre uma imagem e um modelo treinado**

- **Requer um aprovisionamento inicial para treinar o modelo**
  - Autenticações corretas sucessivas podem melhorar o modelo

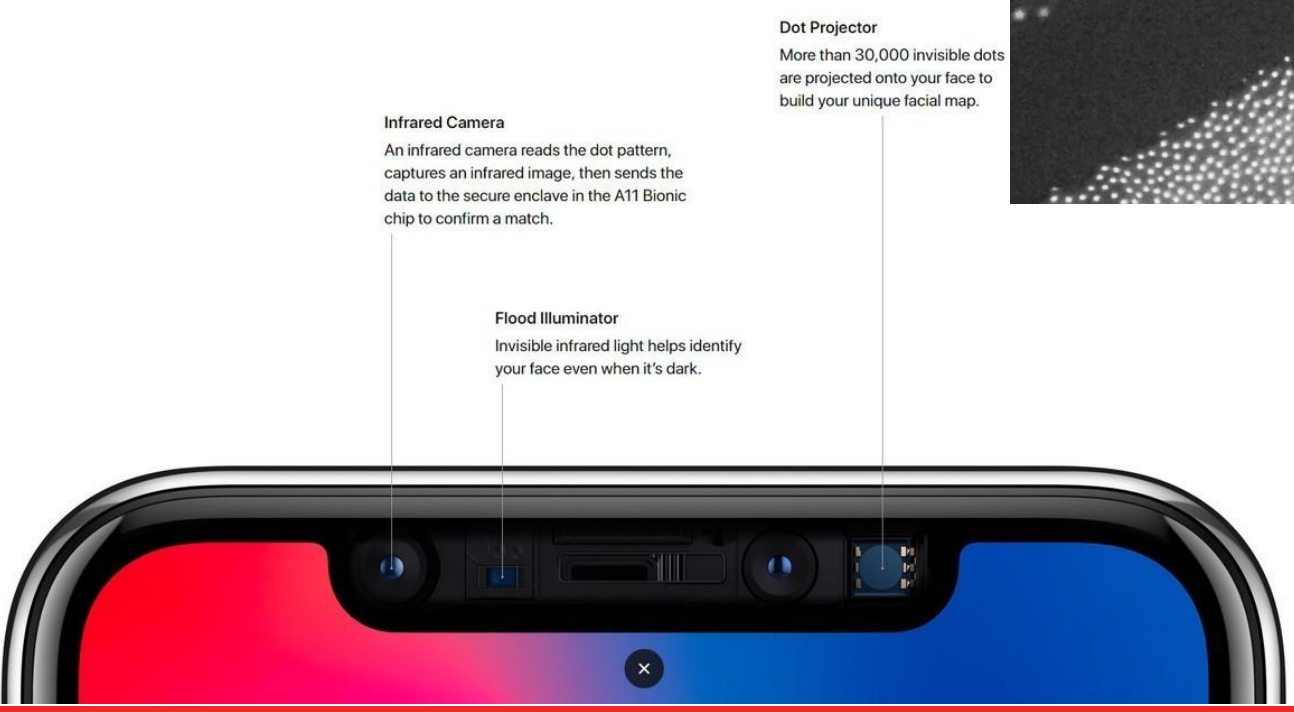
- **Problemas:**
  - Imagens simples podem ser falsificadas: Gémeos, fotografias, filmes
    - Solução: Requerer uma ação (ex, piscar o olho)
  - Nem sempre robusto a alterações de luminosidade
    - Solução: Imagens de Infravermelho
  - Não robusto a alterações do sujeito (barba, óculos)
  - Não robusto a alterações da direção

Smartphones: Face ID





Smartphones: Face ID



Computadores Portáteis

- **São considerados 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

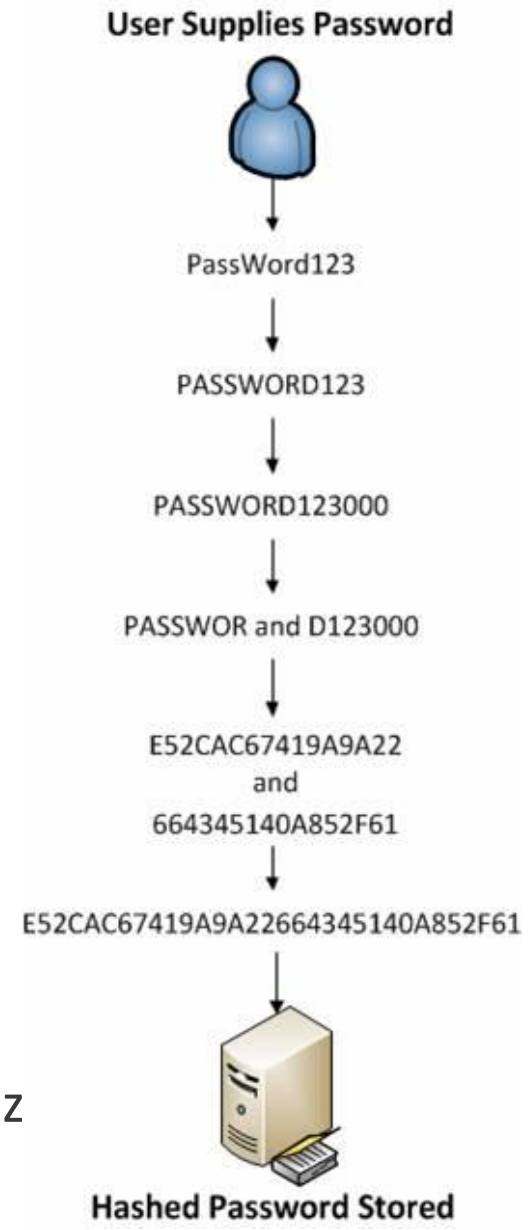
- **Suporta variados métodos de autenticação**
  - PIN, Senhas, Biometria, Smartcards, Tokens
  - Suporta autenticação remota (MS, Active Directory)

- **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/C8MxFlN1$YWNjZDczOWVmNWNmNjBiY
mRlNjBmYWUxZTc4YTJmM2FjZDVmNGU3MmM3MjI2YzZkYzI
2YjRlMDU4:17716:0:99999:7:::
```

- **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

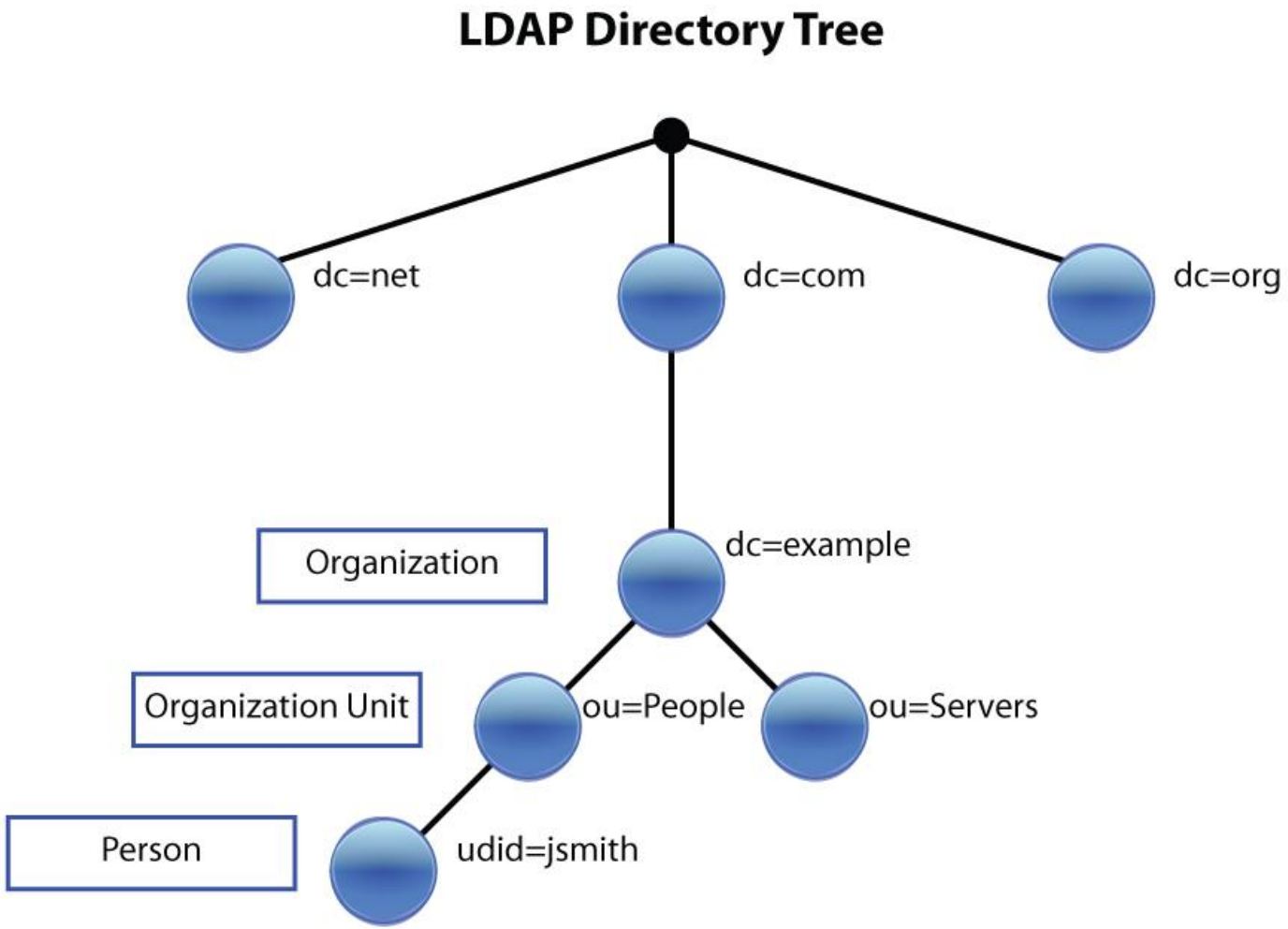
SSO: Single Sign On

- **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



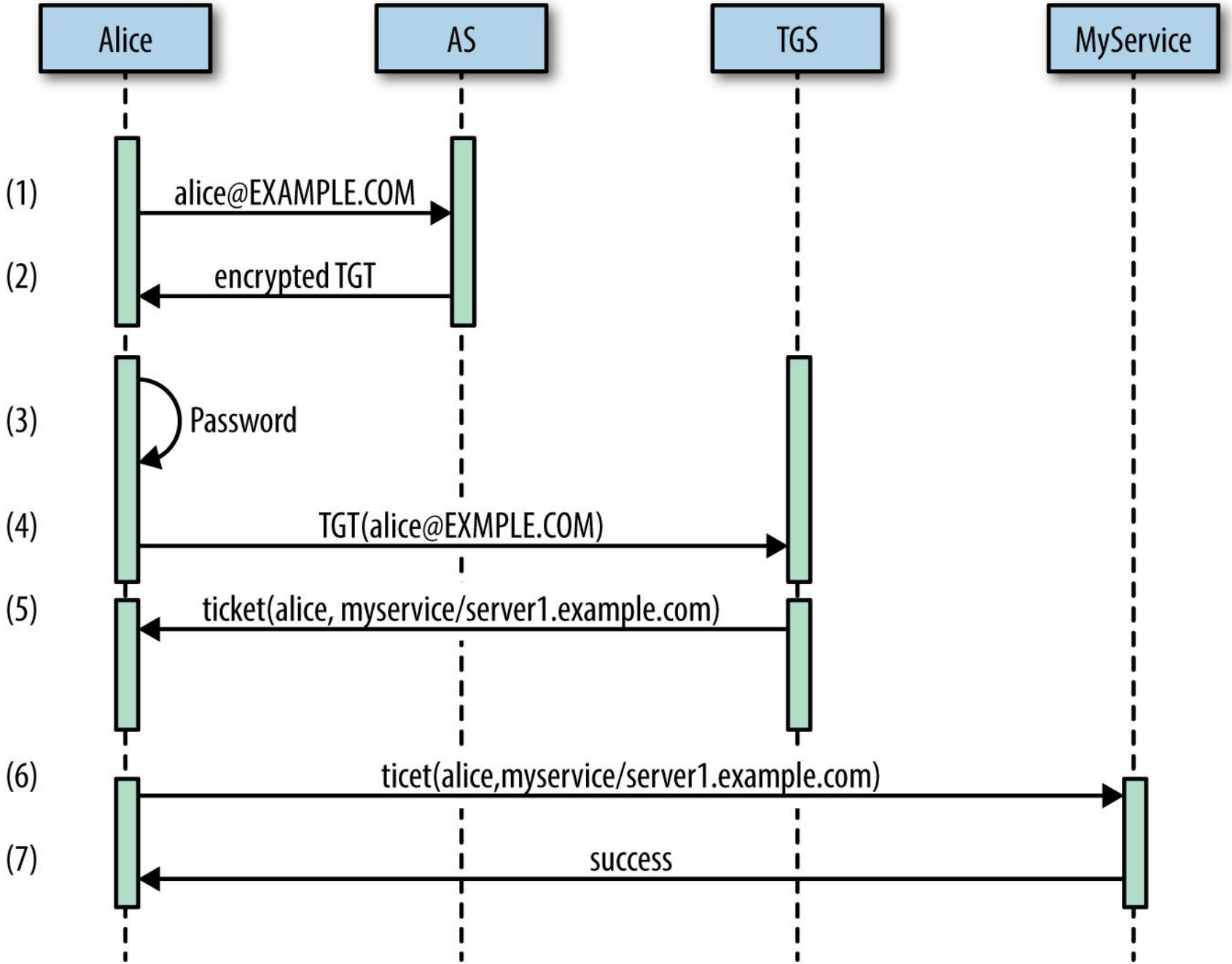
SSO: Kerberos

- **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)
- **Quarto 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: Enc_{user_key}(Client/TGS Session Key)
  - B: Enc_{tgs_key}(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=$Enc_{client/TGS\ SessionKey}$(ClientID, Timestamp)
- **TGS responde com 2 mensagens:**
  - E=$Enc_{service\_key}$(ClientID, client address, validity, Client/Server Session Key)
  - F=$Enc_{client/TGS\ Session}$ Key(Client/Server Session Key)

USE CASE


Alice
AS
TGS
MyService
(1)
alice@EXAMPLE.COM
(2)
encrypted TGT
(3)
Password
(4)
TGT(alice@EXMPLE.COM)
(5)
ticket(alice, myservice/server1.example.com)
(6)
ticet(alice,myservice/server1.example.com)
(7)
success