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

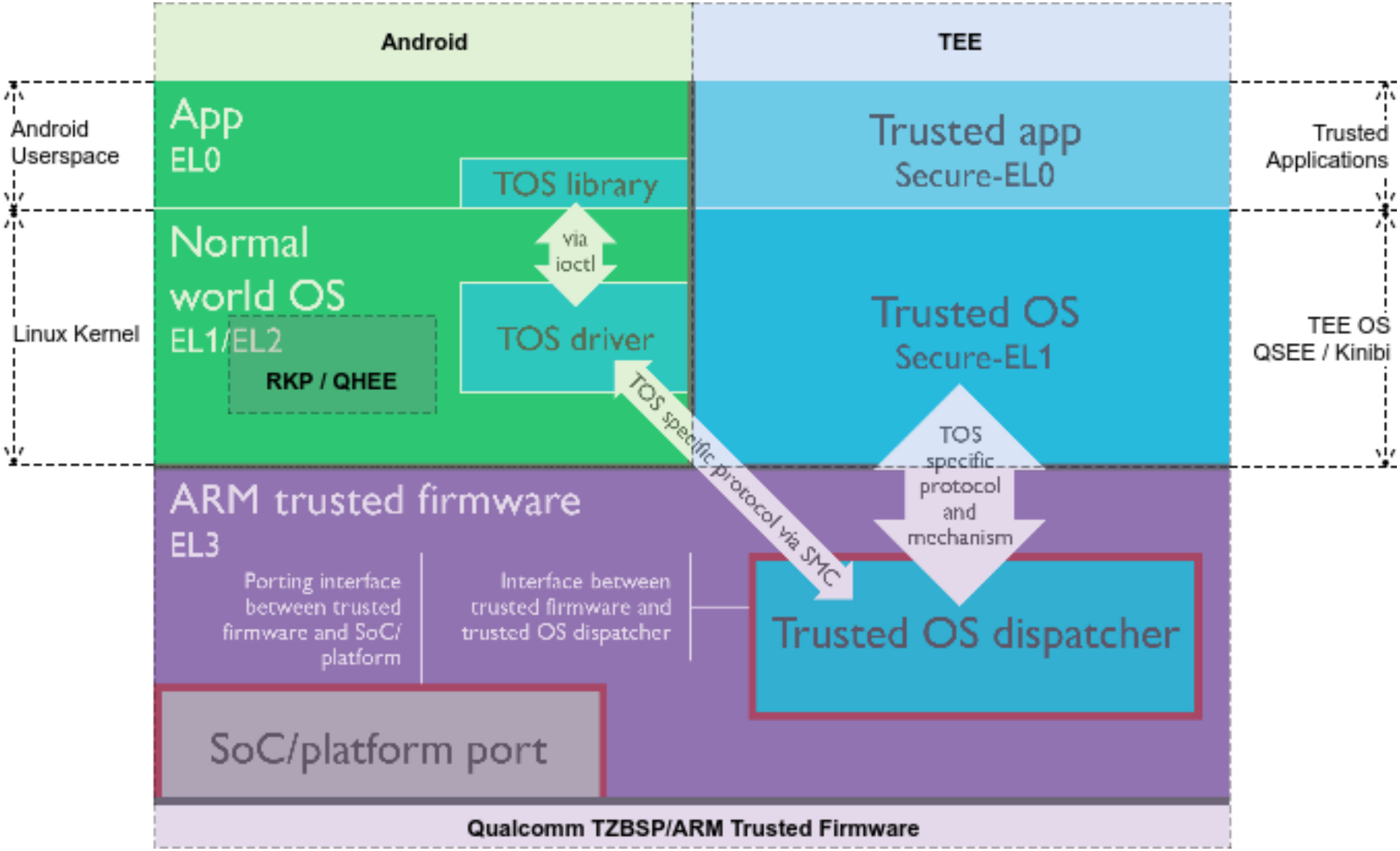
Smartphones: Android

- **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)

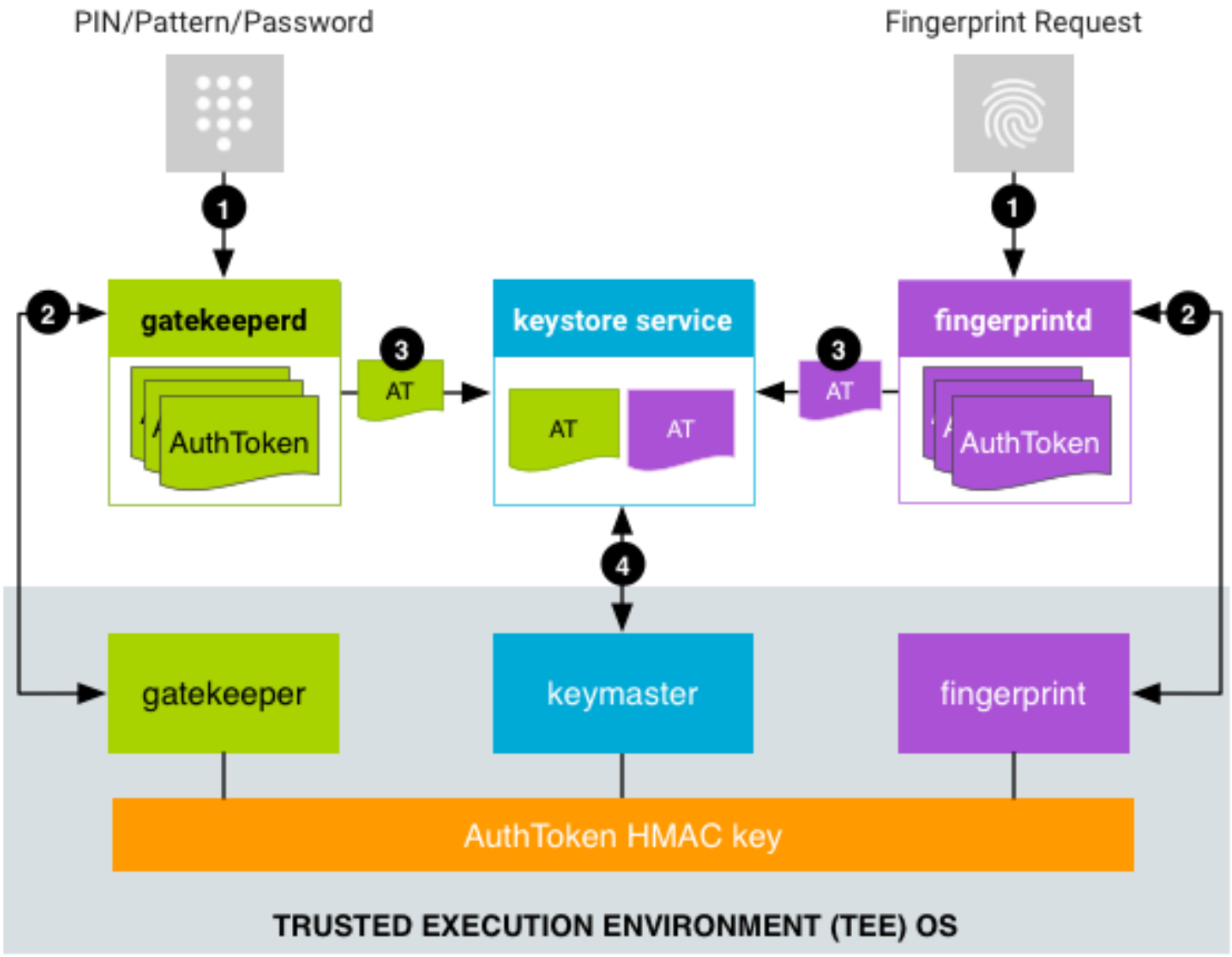
- **Gateways de Segurança**
  - Gatekeeper: para PINs/Passwords e Padrões
  - Fingerprint: para impressões digitais

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

An optical sensor.

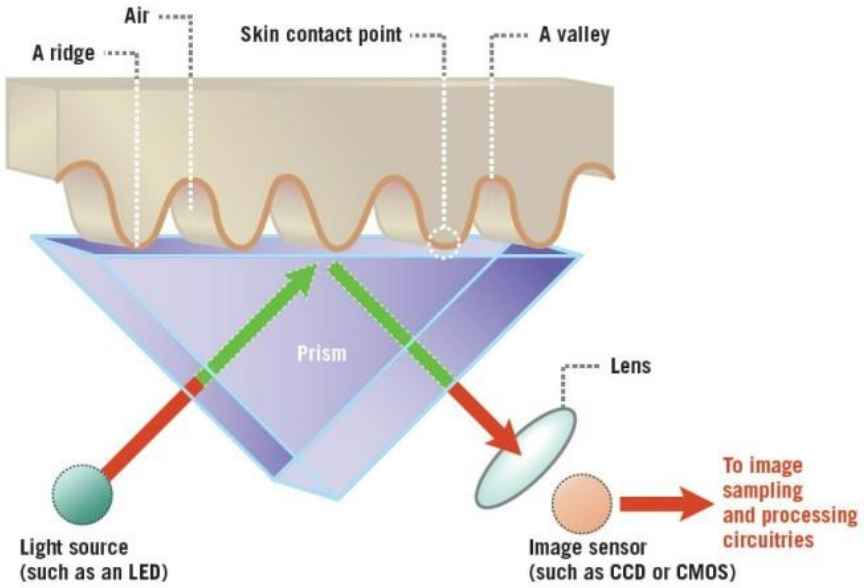
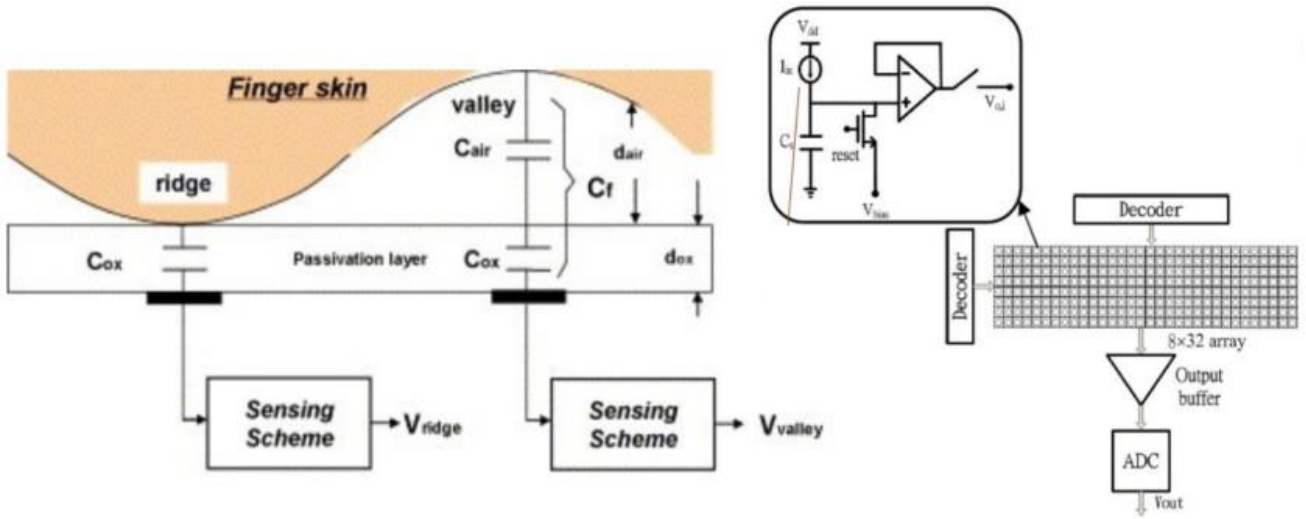


Figure 2

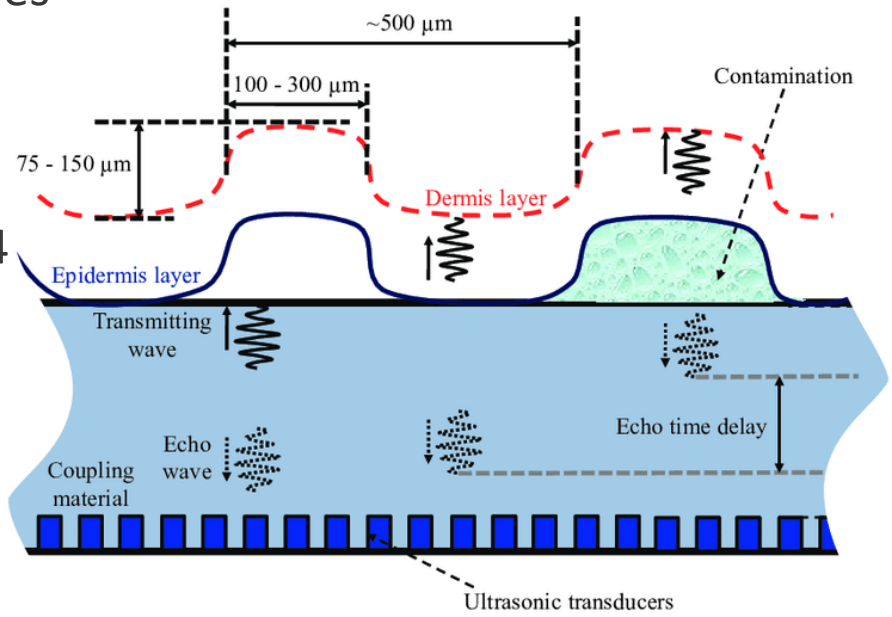
Impressões Digitais: Leitores Capacitivos

- **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
- **Mesmo assim com falhas**
  - youtube/watch?v=hJ35ApLKpN4



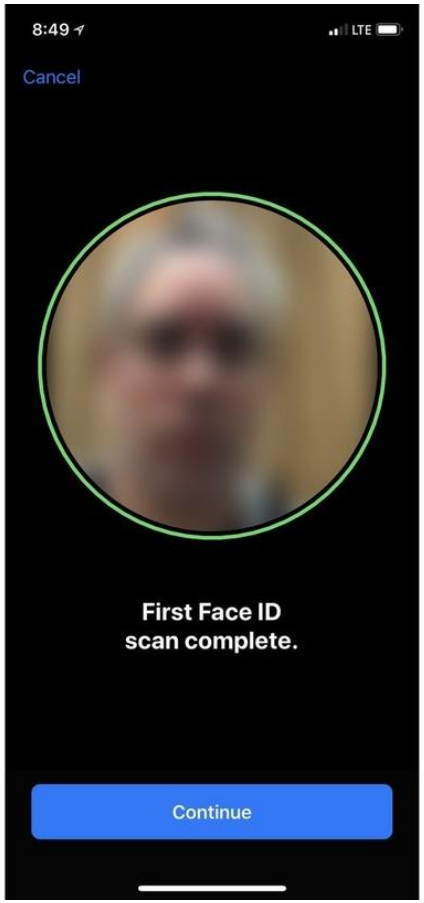
Smartphones: Reconhecimento Facial

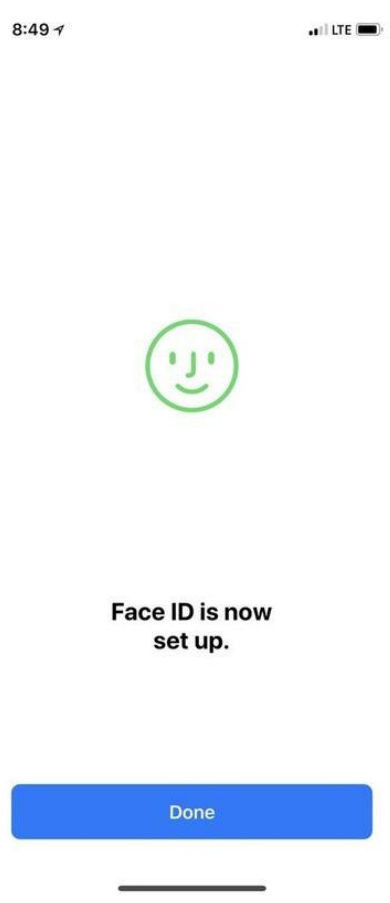
- **Objetivo: Verificar a correspondência entre uma imagem e um modelo treinado**

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

- **Problemas:**
  - Imagens simples podem ser falsificadas: 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

Dot Projector
More than 30,000 invisible dots are projected onto your face to build your unique facial map.

Infrared Camera
An infrared camera reads the dot pattern, captures an infrared image, then sends the data to the secure enclave in the A11 Bionic chip to confirm a match.

Flood Illuminator
Invisible infrared light helps identify your face even when it's dark.

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

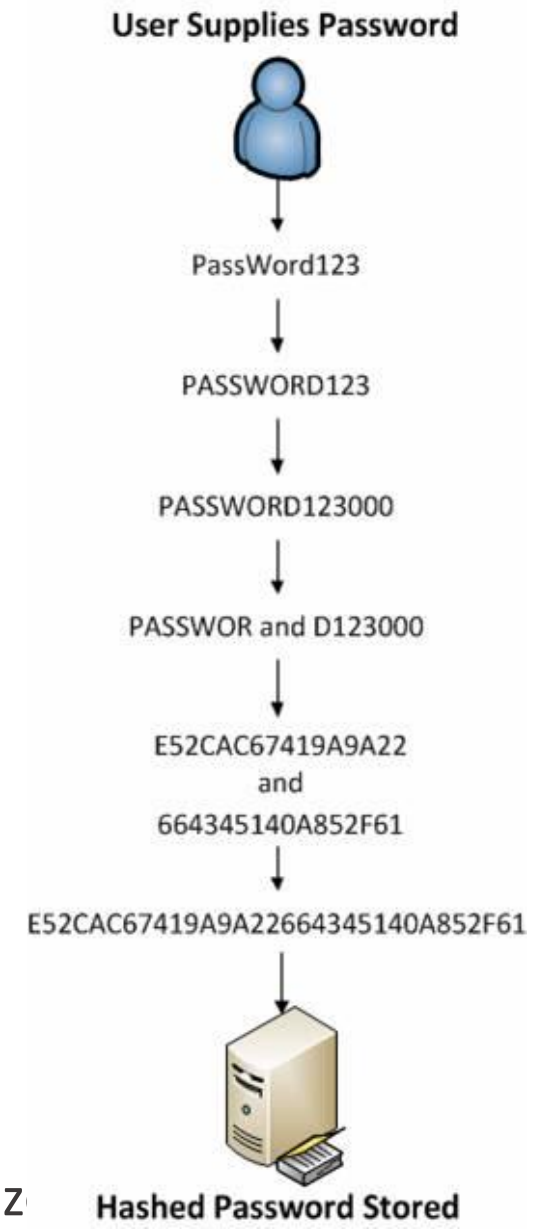
- **Suporta variados métodos de autenticação**
  - PIN, Senhas, Biometria, Smartcards, Tokens
  - Suporta autenticação remota (MS, Active Directory)

- **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/C8MxFlN1$YWNjZDczOWVmNWNmNjBiYmRlNjBmYWUxZTc4YTJmM2FjZDVmNGU3MmM3MjI2YzZkYzI2YjRlMDU4:17716:0:99999:7:::
```

- **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

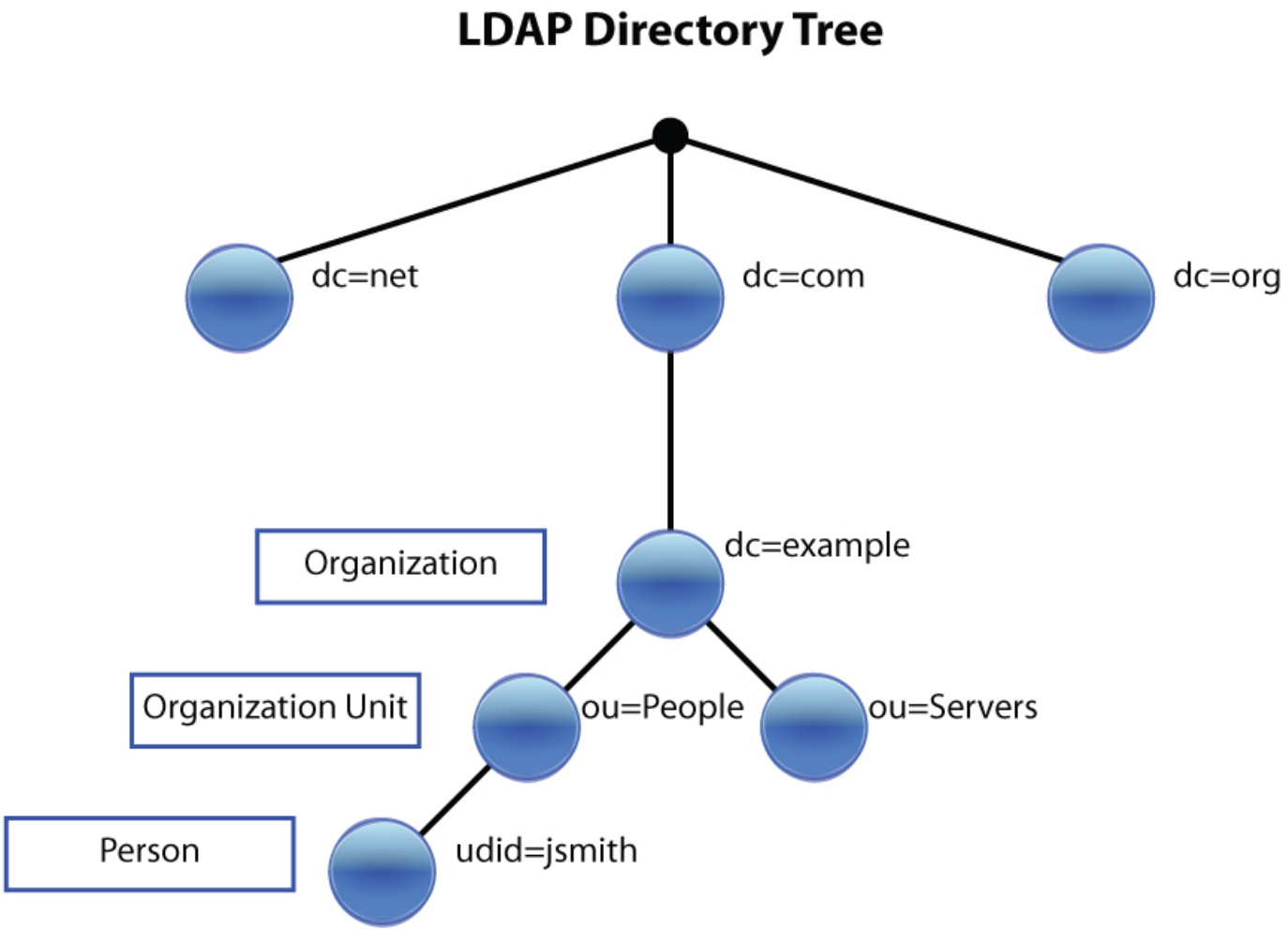
SSO: Single Sign On

- **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



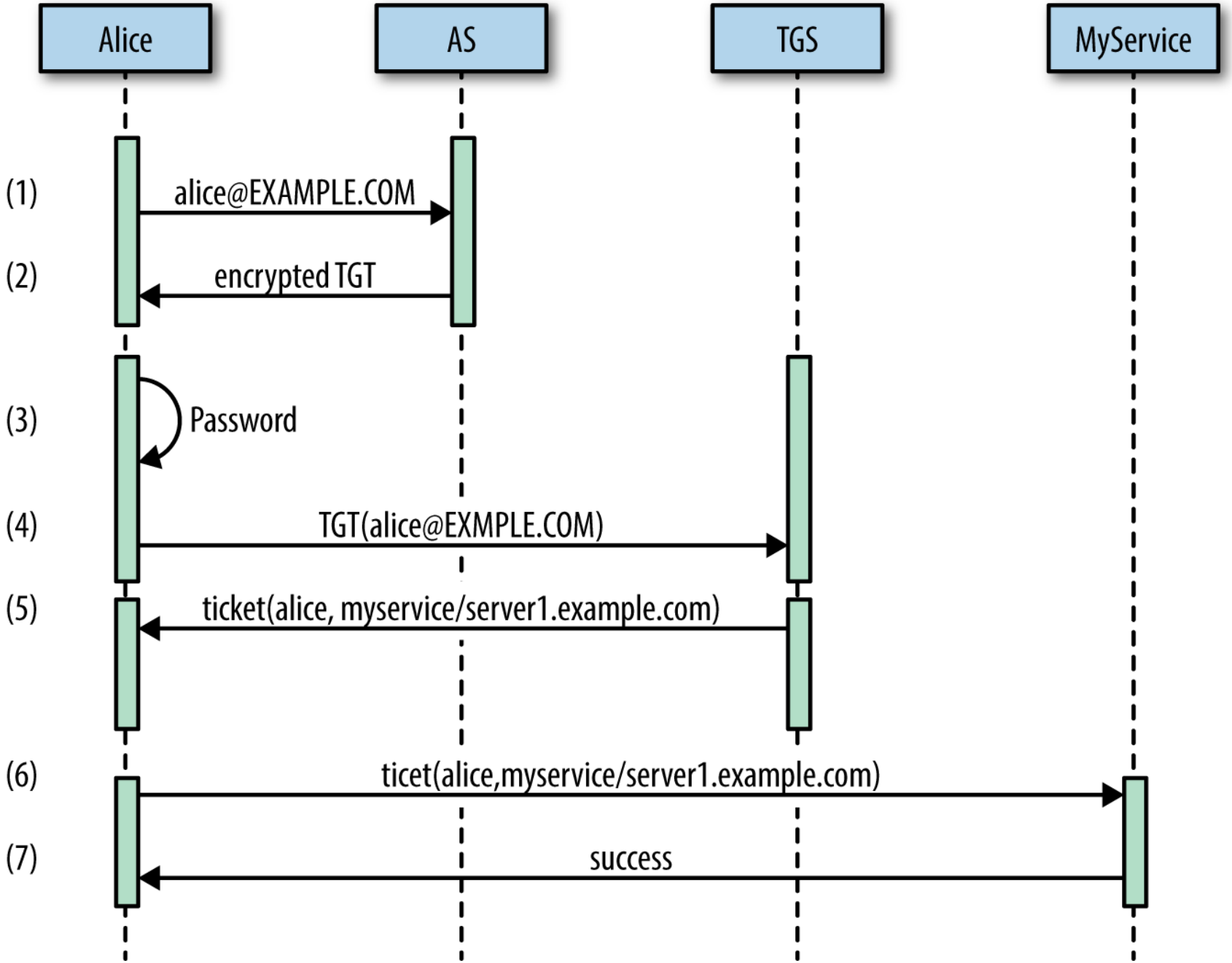
SSO: Kerberos

- **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