Authentication In Specific Systems

Authentication in Systems

Devices and systems operate based on an identity

- With personal data is restricted to its owner
- Each system implements specific authentication processes

Validation against credentials/template

- Credentials/biometric template can be local
  - Frequently it is only local
- Can make use of secure execution mechanisms

Should provide offline authentication mechanisms

- Can support online mechanisms

Smartphones

Considered to be personal devices

- Frequently used to personally identify a person

Can exploit the existence of a SIM card or other HW

- Sold to an existing entity, Registered to an entity, Protected by a PIN code

Can use multiple authentication sources

- Passwords, PINs, Patterns, Biometrics

Supported by Trusted Environment

- Android: Trusty OS

Smartphones: Android

Uses a user-authenticated-gated keys

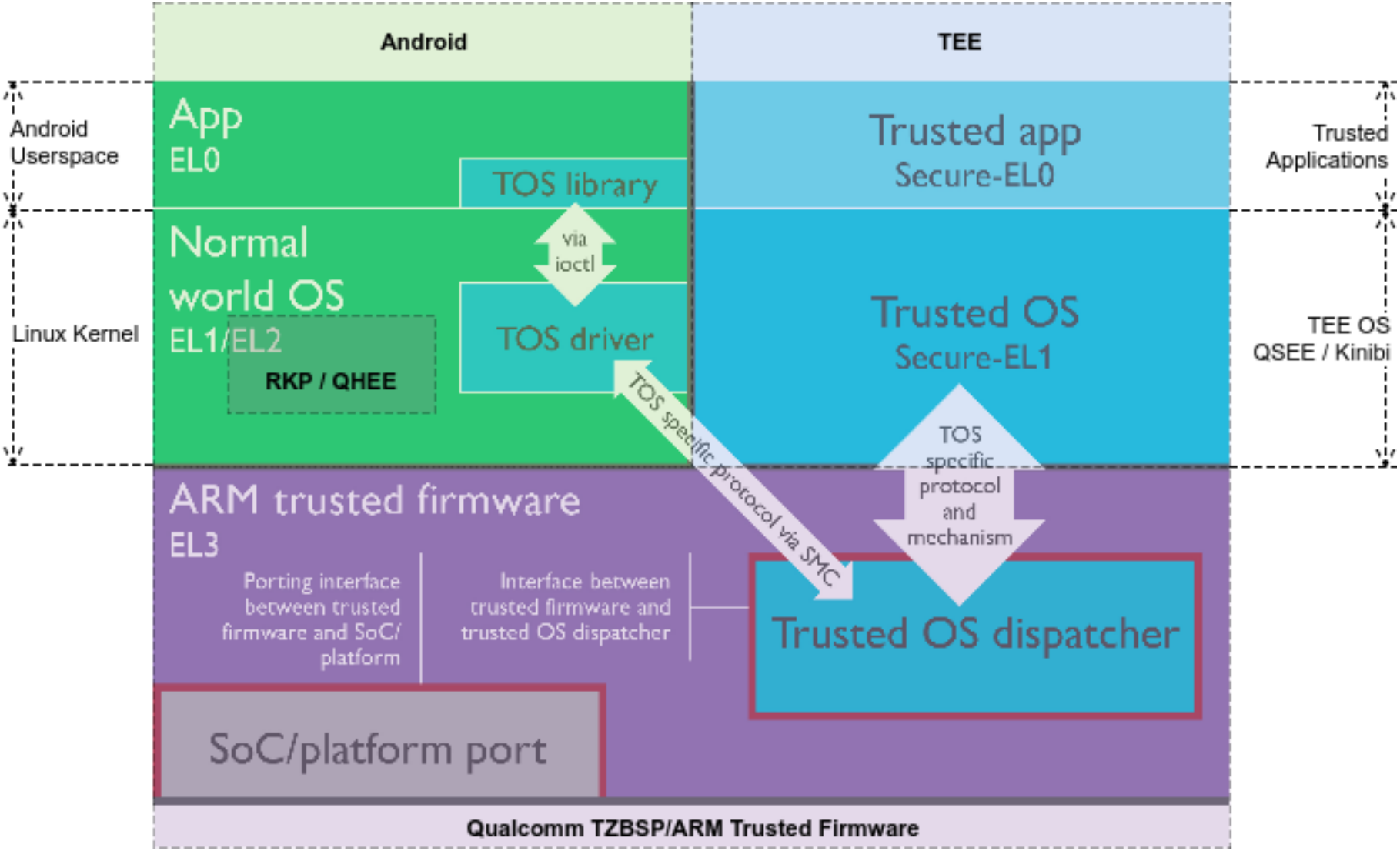
- Gate authenticates users to unlock keys
- Keystore stores keys in a protected environment

Security gates

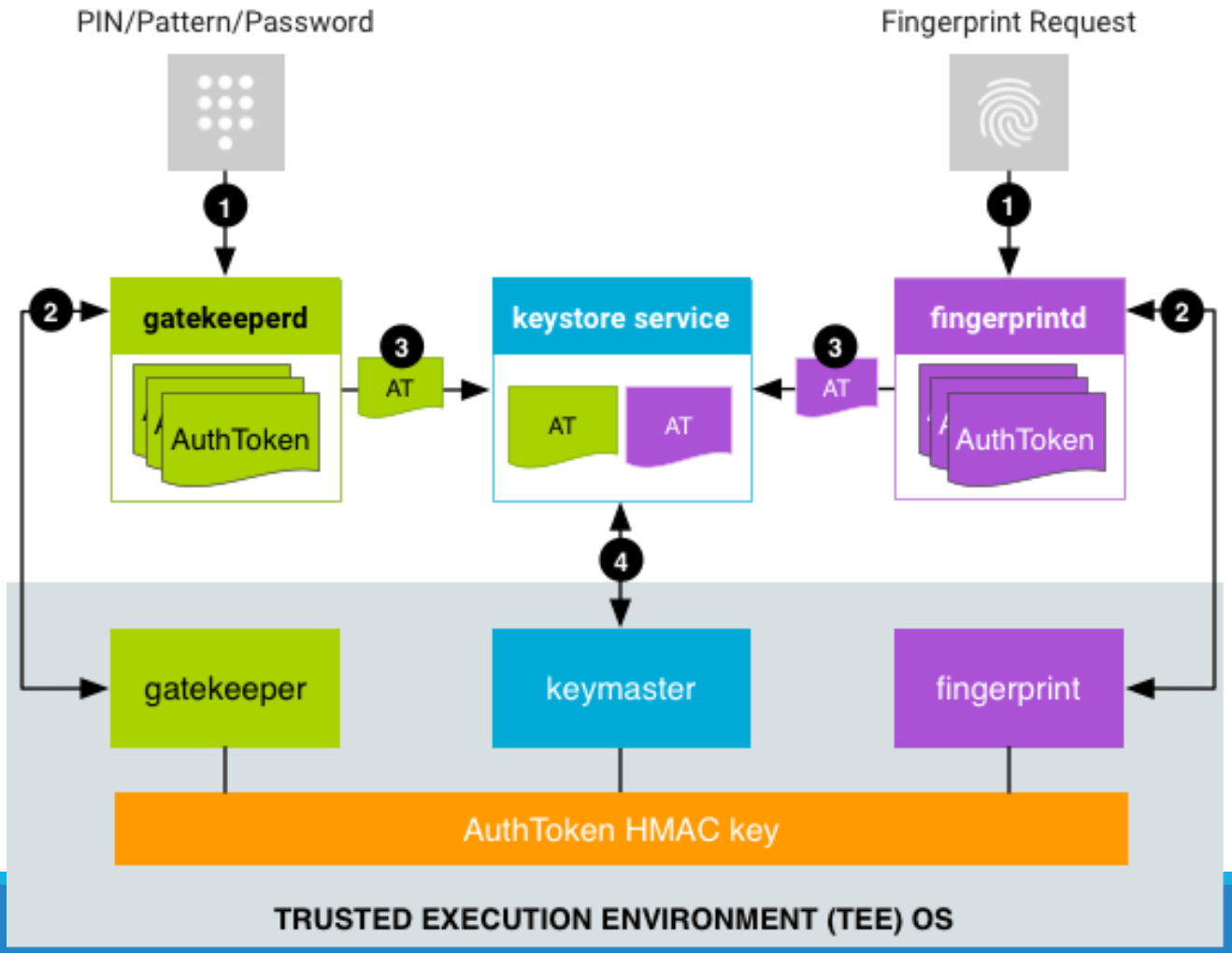
- Gatekeeper: for PINs/Passwords/Patterns
- Fingerprint: for fingerprints

PINs/Passwords/Patterns tied to an identity

- providing a pin unlocks its keys
- Secret keys tied to a user


Android
TEE
Android Userspace
App
EL0
TOS library
Trusted app
Secure-EL0
Trusted Applications
Normal world OS
EL1/EL2
via ioctl
TOS driver
Linux Kernel
RKP / QHEE
Trusted OS
Secure-EL1
TEE OS
QSEE / Kinibi
TOS specific protocol via SMC
TOS specific protocol and mechanism
ARM trusted firmware
EL3
Porting interface between trusted firmware and SoC/ platform
Interface between trusted firmware and trusted OS dispatcher
Trusted OS dispatcher
SoC/platform port
Qualcomm TZBSP/ARM Trusted Firmware

Smartphones: Android



Smartphones: Android Gatekeeper

Initial enrollment required

- Identity plus shared secret (PIN, Password, Pattern)
- 64bit random User Secure ID is generated and stored

Gatekeeper in the App Environment

- Sends SID + credentials to TEE
- Receives signed AuthToken
- Contacts keystore to obtain keys

Trusted Environment

- Validates credentials for SID
- Generates with valid AuthToken

Smartphones: 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: Keymaster

Provides access to the keystore

- API based, not full RW access
- Replies to requests from authorized services (shared secret), having a valid (recent) AuthToken

Keymaster 1: Android 6

- Signing API (sign, verify, import keys)

Keymaster 2: Android 7

- Support for AES and HMAC
- Key Attestation: Certifies keys (origin, property, usages)
- Version Binding: ties keys to OS and TEE version, preventing downgrades

Keymaster 3: Android 8

- ID Attestation: Key device identifiers are stored as HMAC(HWKEY, IDn)

Keymaster 4: Android 9

- Embedded Secure Elements: allowing embedded “smartcards”

Android: Keymaster Key Attestation

Objective: Ensure keys are originated from the TEE, and are authentic

Other assurances:

- Generated by the current TEE (based on its ID)
  - ID=HMAC_SHA256(instante temporal || AppID || R, HBK)
    - R = a tag::RESET_SINCE_ID_ROTATION, HBK: a secret Hardware Backed Key

Call: attestKey(KeyToAttest, attestParams)

Result: A X.509 certificate

- Signed by a specific root certificate
- With an extension containing the result

Smartphones: Gatekeeper auth

PIN: Direct input of a digit based code

- Usually 4 digits but can be changed up to 16 digits
- Not related to the SIM PIN
- Vulnerable to attacks using sensors (gyro/accell)

Password: Direct input of a stream of characters

- Usually limited to 16 chars
- Less vulnerable to attacks using sensors (gyro/accell)

Pattern: Direct input of a pattern

- Potentially more secure than 4 digit PINS
- Stored as a unsalted SHA-1 digest
- Vulnerable to over-the-shoulder attacks, grease marks

Smartphones: Fingerprint

TEE stores a multi sample profile of a fingerprint

- always encrypted, even inside TEE
- associated to a SID
- Deleted if user is removed from device

Profile is obtained from sensor, validated in TEE

- Cannot be extracted
- Fingerprint is sent to TEE for validation

Security level varies with sensor implementation

- Several implementations

Fingerprint types: Optical

Sensor takes picture of finger

- Can use LEDs for illumination

Only a 2D image

- fooled by pictures, fingerprint models, latent prints

Present in first versions and entry level devices

An optical sensor.

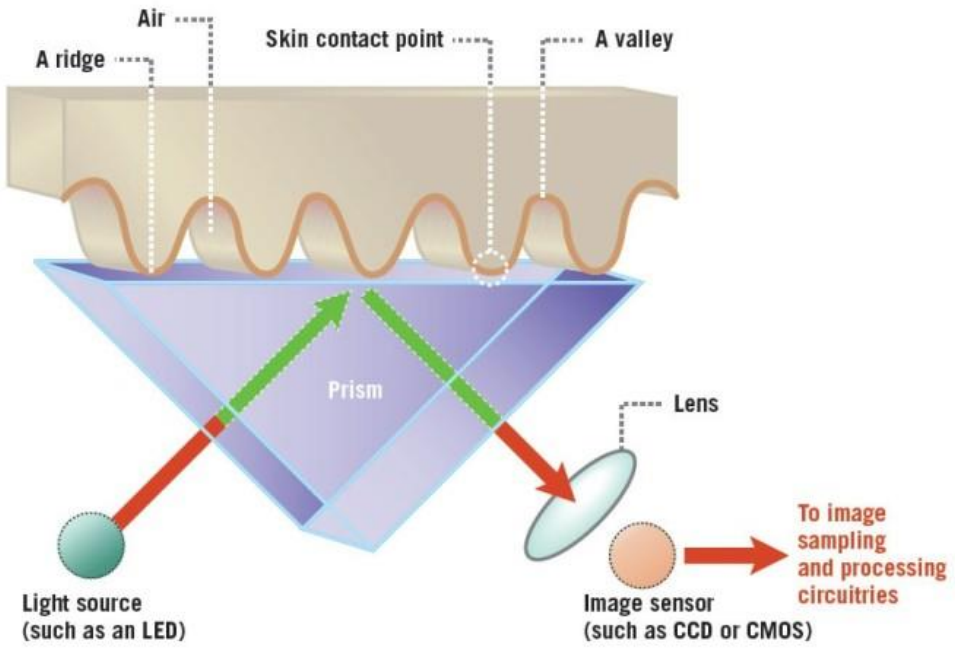


Figure 2

Fingerprint types: Capacitive

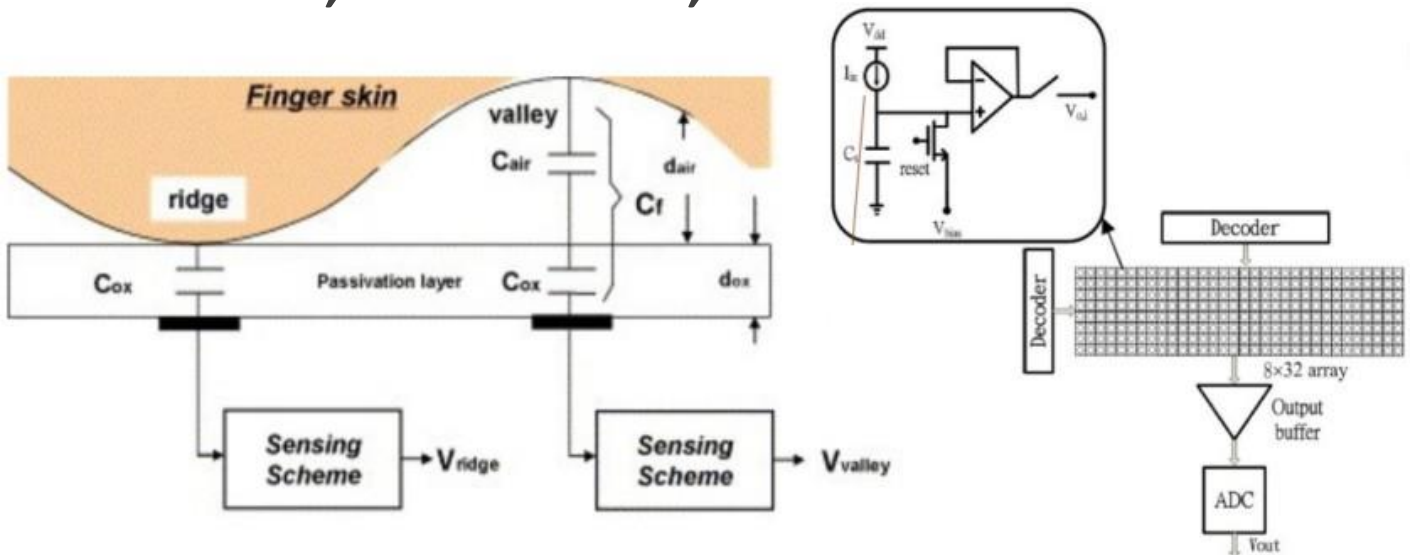
Sensor measures capacitance along the surface

- Ridges and valleys (in sub-epithermal layers)
- Allows for Swipe implementations (cheaper versions)

Vulnerable to prosthetic (silicone) fingers

- With model from authenticated user

Interference from sweat, hand lotions, water



Fingerprint types: Ultrasonic

Ultrasound emitter and receiver

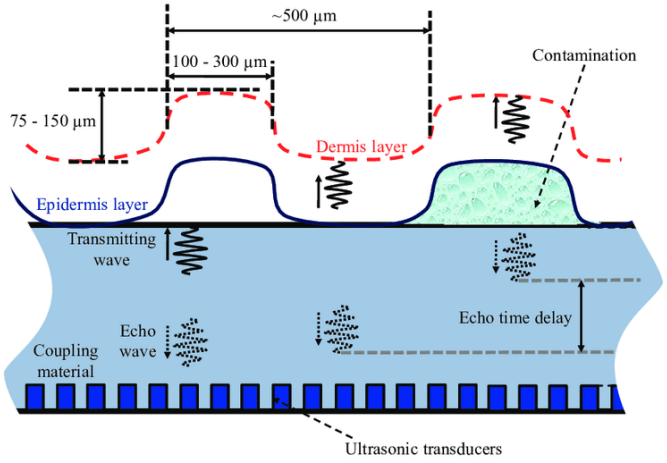
- Emitter: Emits pulse that is transmitted to the finger
- Received: listens for echos as sound encounters features

More difficult to circumvent and more resilient to surface material

- Echos penetrate through water, lotion, and bumps on features

Still possible…

- youtube/watch?v=hJ35ApLKpN4



Smartphones: Face Recognition

Objective: Match face against trained model

- Based on commonly available face recognition software

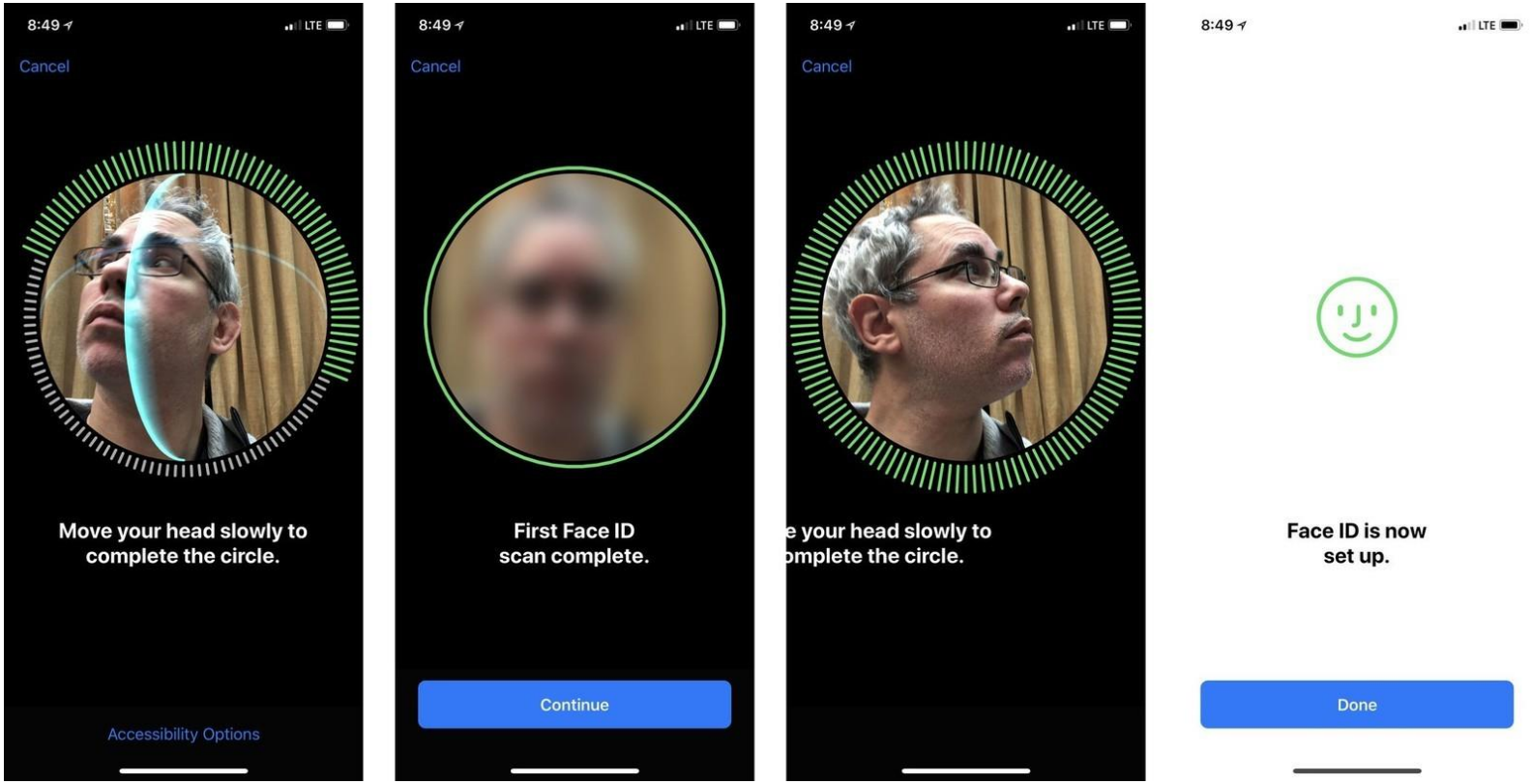
Requires initial enrollment to create train model

- Successful authentication can increase train data

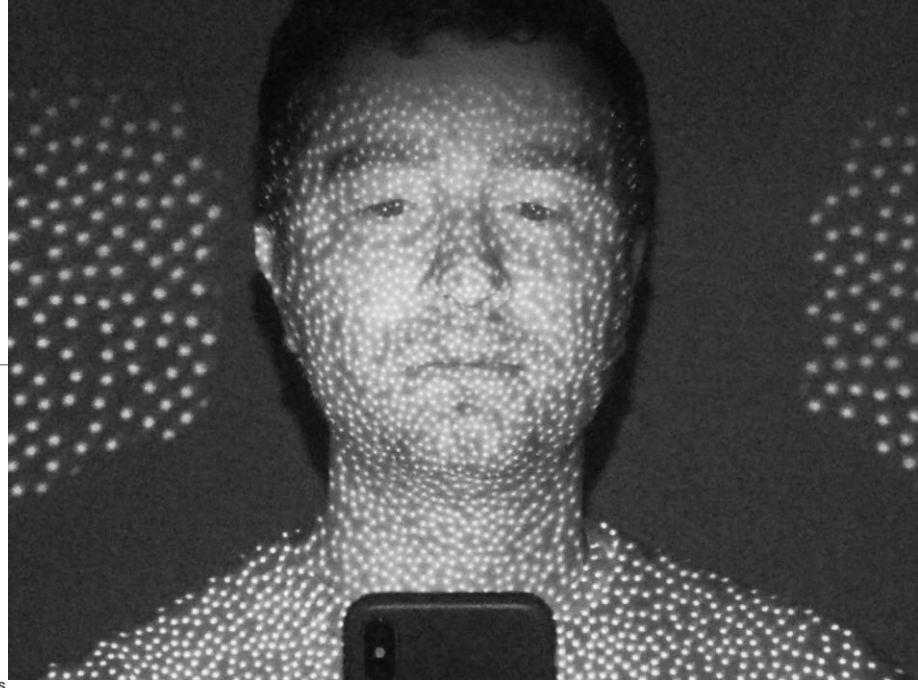
Has some issues:

- Simple image can be fooled by a picture/movie/evil twin
- Not resilient to changes in lighting
- Not resilient to changes of the subject (glasses, beard)
- Not resilient to changes in posture

Smartphones: Face ID



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.

Laptops

Laptops are considered as potentially shared devices

- Not really considered as individual devices
- May have some sensors/readers
- May have Trusted Platform Modules (TPM)

Authentication bound to underlying OS

- Simpler than smartphones
  - No SIM card
  - No TEE
  - Simpler biometric approach

No universal support for hardware backed key store

Laptops: Hardware support

Fingerprint sensors like in smartphones

- Swipe, discrete or in power button

Hardware for face recognition

- standard camera (standard in all laptops)
- infrared camera (more recent implementations)

Smartcard reader

- Allows use of traditional SmartCards (e.g., CC)
- More frequent in laptops for corporate environments

Can interact with other devices

- Smartphone, bracelet, Yubikey

OS: Windows

Supports a wide range of authentication methods

- PIN, Password, Biometrics, SmartCards, Tokens
- supports remote authentication (e.g., Active Directory)

Credentials stored in SAM (Security Account Manager)

- Optional: partially encrypted using SysKey
- trivial to remove a user password (delete SAM entry)
- Mapped to windows registry in HKLM/SAM

Since W Vista UAC enforces Access Control after authentication

- Vista launched in 2006
- UAC can still be disabled!

OS: Windows Passwords

Password: Direct validation against stored value

- Stored in c:\Windows\System32\Config\SAM
- Encrypted with Boot Key (SysKey)
- Complexity imposed by Admin Policy

LM Password Hash Up to W7

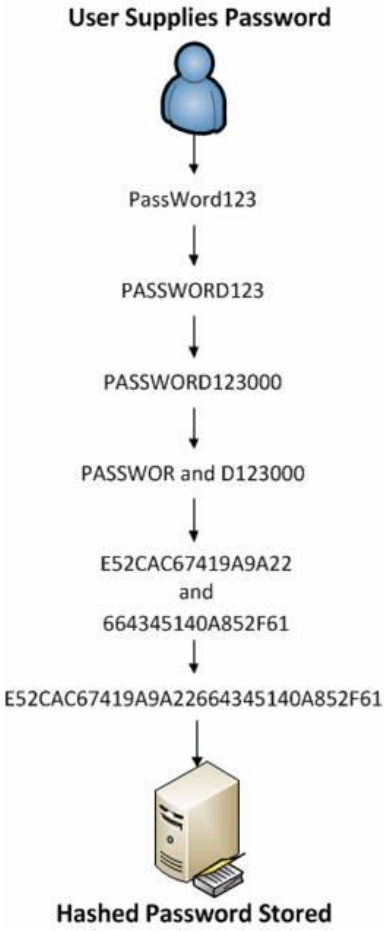
- Encrypts standard value (KGS!@#$%) using DES(password, standard)

NTLM Password Hash

- Non Salted MD4(Password)
- Same password -> same hash

Validation:

- Request username and password
- Calculates hash, compares the result with stored value



OS: Windows PIN

Backed by a Trusted Platform Module (TPM)

- Similar to TEE, provides secure environment with storage
- Can guarantee hardware tamper free state

PIN unlocks TPM which allows access to keys

- repeated incorrect attempts will lock TPM
- cannot be extracted (bound to device)

OS: Windows Hello

Uses Visible Light + IR cameras to obtain 3D image

- Can have LED for flood illumination
- IR camera adds resilience to lighting changes
- Two cameras introduce 3D depth data (from the parallax)
- PIN is mandatory as backup

Vulnerable

- to 3d printed face?
- to IR sensitive print
- to standard print in earlier W10

OS: Linux

Supports a wide range of authentication methods

- PIN, Password, Biometrics, SmartCards, Tokens
- supports remote authentication (e.g., Active Directory)

Pluggable Authentication Modules allows per app authentication policies/mechanisms

- without modification to applications
- e.g: SmartCards, OTP, Kerberos, LDAP, Databases, Network Location, etc..

Standard Credentials stored in /etc/shadow

- not encrypted
- Alternate authentication methods may use other storage (e.g., TPM, SmartCard, Database)

OS: Linux Passwords

User account info in /etc/passwd

- username, user id, shell...

Credentials stored in /etc/shadow

- only readable by root, transformed using a salted digest

Validation:

- obtain credential from user
- access shadow: verify hash used and obtain salt
- calculate hash(salt + password) for N rounds (default is 5000)
- compare result obtained

Entry:
user:6kZ2HbBT/C8MxFlN1$YWNjZDczOWVmNWNmNjBiYmRlNjBmYWUxZTc4YTJmM2FjZDVmNGU3MmM3MjI2YzZkYzI2YjRlMDU4:17716:0:99999:7:::

Meaning: username:$ hash used $ salt $ password hash: ... dates and validity

SSO: Single Sign On

Unique, centralized authentication for a set of federated services

- The identity of a client, upon authentication, is given to all federated services
- The identity attributes given to each service may vary
- The authenticator is called Identity Provider (IdP)

Examples

- SSO authentication at UA
  - Performed by a central IdP (idp.ua.pt)
  - The identity attributes are securely conveyed to the service accessed by the user

SSO: Single Sign On

Trusted third parties (TTP) used for authentication

- But often combined with other related functionality
- E.g. Google, Facebook

AAA services

- Authentication, Authorization and Accounting
- e.g. RADIUS and DIAMETER

SSO: Single Sign On

Advantages:

- Can reuse same credentials over multiple systems/services
- Single secure repository for credentials
  - More difficult to steal credentials when used in many services
- Can implement restrictions to services/systems

Disadvantages:

- Requires additional servers
- Single point of failure: without authentication systems, no one will be authenticated
  - Important to also deploy local credentials for admins
- Introduces delays in the authentication process

SSO: Single Sign On

Requires software that "injects" remote users into local system

- Windows: Remote users not available in SAM
- Linux: Remote users not available in /etc/passwd
- Must cache data to enable large number of validations (e.g., ls)

May provide further information to be used as user profile

- Type of user (student, professor, admin)
- email, home, other preferences

Systems that make use of SSO need to be provisioned

- And sometimes, specifically authorized

SSO: LDAP
Lightweight Directory Access Protocol

Protocol to keep distributed directory information

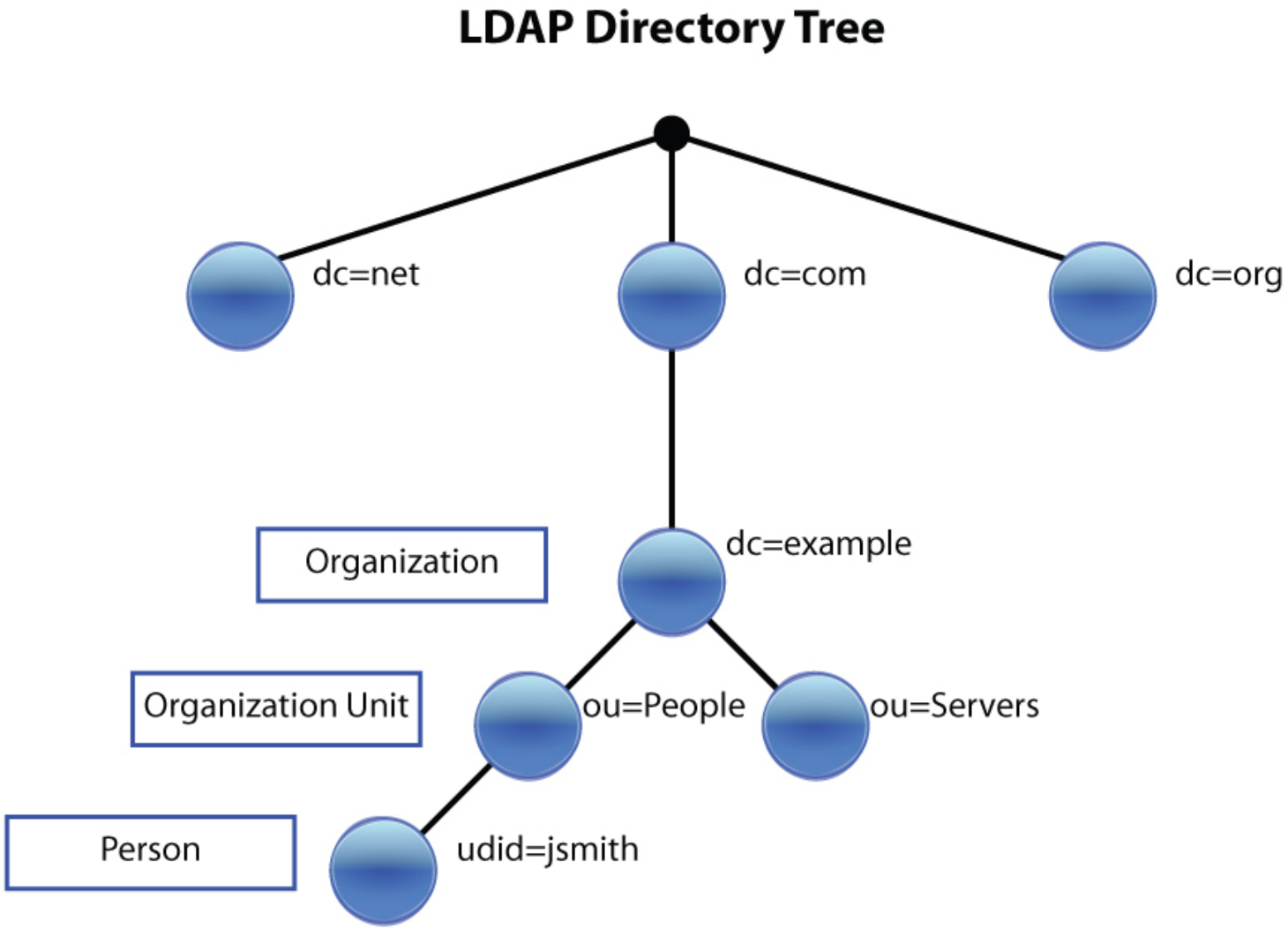
- Directory keeps hierarchical information about users, systems and services
  - E.g., address book, user profile
- Information is organized in a tree: dn=user,ou=deti,dc=ua, dc=pt
  - DC: Domain Component, OU: Organizational Unit, DN: Distinguished Name
- Each record obeys to a specified composition of individual schemas

Access to LDAP can be anonymous or authenticated

- Anonymous information: general contacts and configurations
- Authenticated (Bind): Specific profile info

LDAP Bind: credentials are user path and password

- Support for different authentication methods: PLAIN, SASL, Certificates
- Supports same username in different domains
  - **dn=usera,ou=deti,dc=ua,dc=pt vs dn=userb,ou=deti,dc=ua,dc=pt**


LDAP Directory Tree
dc=net
dc=com
dc=org
Organization
dc=example
Organization Unit
ou=People
ou=Servers
Person
udid=jsmith

SSO: Kerberos

Authentication protocol for usage in networked environments

- Based on the notion of Tickets with limited validity
- Default process for Microsoft Active Directory (and CodeUA)

Supports mutual authentication

- Actually, the authenticator will send the password to the client!

Four Key Entities

- Client: Wishes to access a service
- Service Server (SS): Provides a service the user wants to access
- Ticket Granting Server (TGS): Provides access to services
- Authentication Server (AS): Provides access to the TGS for each user

Key Distribution Center: AS + TGS (+database)

SSO:
Kerberos: Client Authentication

1: Client password is transformed (e.g. hash)

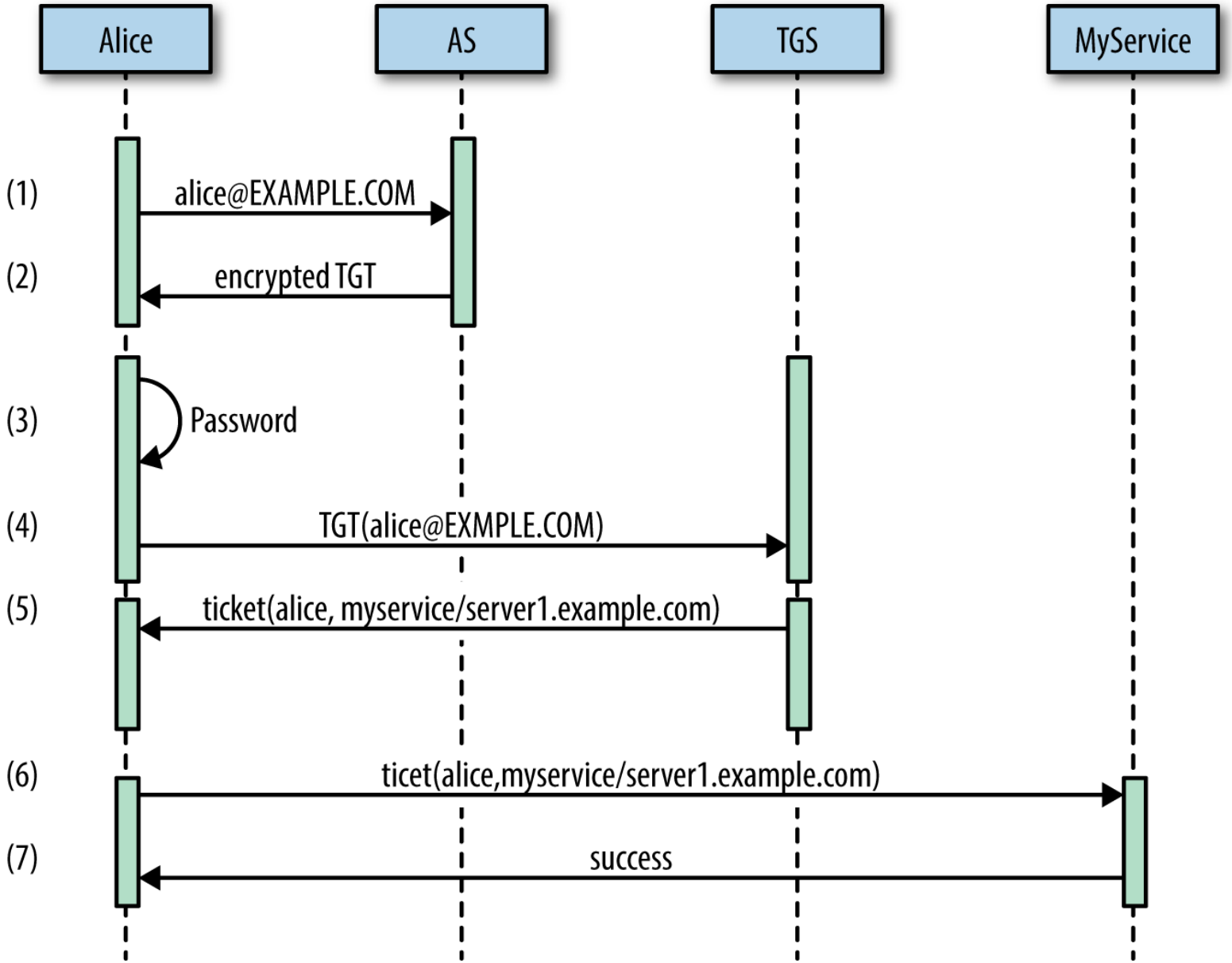
2: Client sends authentication request to AS with ClientID

3: AS replies with 2 messages:

- A: E_{user_key}(Client/TGS Session Key)
- B: E_{tgs_key}(TGT)
  - Ticket Granting Ticket = Client, client network address, validity, Client/TGS Session Key

4: User uses its key to decrypt A

- if password equals the one stored in AS he has access to TGS Session Key
- He can request Authorization to access the Service


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