

Authentication protocols



Identity attributes

▷ Set of attributes for setting apart individuals

- ♦ Name
- ♦ Numerical identifiers
 - Fixed for life
 - Variable with context
- ♦ Address
- ♦ Photo
- ♦ Identity of relatives
 - Usually parents
- ♦ ...



Authentication: Definition

- ▷ Proof that an entity has a claimed identity attribute
 - Hi, I'm Joe
 - Prove it!
 - Here are my Joe's credentials
 - Credentials accepted/not accepted

- Hi, I'm over 18
- Prove it!
- Here is the proof
- Proof accepted/not accepted



Authentication: proof types

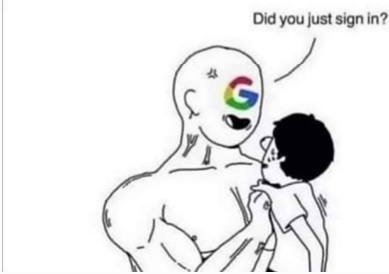
- ▷ Something we know
 - ♦ A secret memorized (or written down...) by Joe
- ▷ Something we have
 - ♦ An object/token solely held by Joe
- ▷ Something we are
 - ♦ Joe's Biometry
- ▷ Multi-factor authentication
 - ♦ Join or consecutive use of different proof types



Multi-factor verification jokes

me: *enters password correctly on new device*

google:



© André Zúquete

Identification, Authentication and Authorization

5

Authentication: goals

- ▷ Authenticate interactors
 - ♦ People, services, servers, hosts, networks, etc.
- ▷ Enable the enforcement of authorization policies and mechanisms
 - ♦ Authorization \Rightarrow authentication
- ▷ Facilitate the exploitation of other security-related protocols
 - ♦ e.g. key distribution for secure communication



© André Zúquete

Identification, Authentication and Authorization

6

Authentication: requirements

▷ Trustworthiness

- ♦ How good is it in proving the identity of an entity?
- ♦ How difficult is it to be deceived?
- ♦ Level of Assurance (LoA) (NIST, eIDAS, ISO 29115)
 - LoA 1 - Little or no confidence in the asserted identity
 - LoA 2 - Some confidence in the asserted identity
 - LoA 3 - High confidence in the asserted identity
 - LoA 4 - Very high confidence in the asserted identity

▷ Secrecy

- ♦ No disclosure of secrets used by legitimate entities



Authentication: requirements

▷ Robustness

- ♦ Prevent attacks to the protocol data exchanges
- ♦ Prevent on-line DoS attack scenarios
- ♦ Prevent off-line dictionary attacks

▷ Simplicity

- ♦ It should be as simple as possible to prevent entities from choosing dangerous shortcuts

▷ Deal with vulnerabilities introduced by people

- ♦ They have a natural tendency to facilitate or to take shortcuts



Authentication: Entities and deployment model

▷ Entities

- ♦ People
- ♦ Hosts
- ♦ Networks
- ♦ Services / servers

▷ Deployment model

- ♦ Along the time
 - Only when interaction starts
 - Continuously along the interaction
- ♦ Directionality
 - Unidirectional
 - Bidirectional (or mutual)



Authentication interactions: Basic approaches

▷ Direct approach

- ♦ Provide **credentials**
- ♦ Wait for verdict
- ♦ Authenticator checks credentials against what it knows

▷ Challenge-response approach

- ♦ Get **challenge**
- ♦ Provide a **response** computed from the **challenge** and the **credentials**
- ♦ Wait for verdict
- ♦ Authenticator checks response for the challenge provided and the credentials it knows



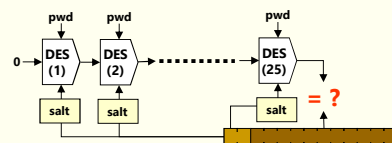
Authentication of people: Direct approach w/ known password

- ▷ A password is matched with a stored value
 - ♦ For a claimed identity (username)

▷ Personal stored value:

- ♦ Transformed by a unidirectional function
 - Key Derivation Function (KDF)
 - Preferably slow!
 - Bcrypt, scrypt, Argon2, PBKDF2
- ♦ UNIX: DES hash + salt
- ♦ Linux: KDF + salt
- ♦ Windows: digest function

DES hash = $\text{DES}_{\text{pwd}}^{25}(0)$
 $\text{DES}_k^n(x) = \text{DES}_k(\text{DES}_k^{n-1}(x))$
 Permutation of 12 subkeys ~ bit pairs with salt (12 bits)



© André Zúquete

Identification, Authentication and Authorization

Authentication of people: Direct approach w/ known password

▷ Advantage

- ♦ Simplicity!

▷ Problems

- ♦ Usage of predictable passwords
 - They enable dictionary attacks
- ♦ Different passwords for different systems
 - To prevent impersonation by malicious admins
 - But our memory has limits!
- ♦ Exchange along insecure communication channels
 - Eavesdroppers can easily learn the password
 - e.g. Unix remote services, PAP



Top 15 2019 by Splashdata

- 1 - 123456
- 2 - 123456789
- 3 - qwerty
- 4 - password
- 5 - 1234567
- 6 - 12345678
- 7 - 12345
- 8 - iloveyou
- 9 - 111111
- 10 - 123123
- 11 - abc123
- 12 - qwerty123
- 13 - 1q2w3e4r
- 14 - admin
- 15 - qwertyuiop

source: <https://www.teampasspassword.com/blog/top-50-worst-passwords-of-2019>
 Image <https://www.pinterest.com/networkboxusa/it-humor>



© André Zúquete

Identification, Authentication and Authorization

12

Password selection jokes

Someone figured out my PASSWORD
Now I have to rename my dog.

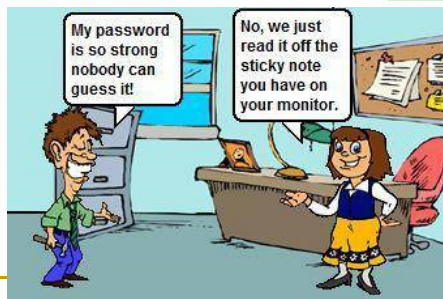
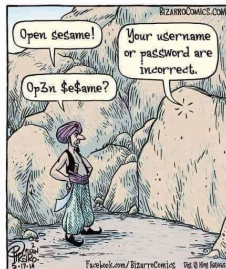


Dear IT,
the more "secure" you try to make our passwords by making them impossible to remember, the more likely I am to save them all in a big word doc named "Passwords"

Signed,
Everyone



Sorry, but your password must contain
an uppercase letter,
a number,
a haiku,
a gang sign,
a hieroglyph,
and the blood of a virgin.



© André Zúquete

Identification, Authentication and Authorization

13

Password bloopers



© André Zúquete

Identification, Authentication and Authorization

14

Authentication of people: Direct approach with biometrics

- ▷ People get authenticated using body measurements
 - ♦ Biometric samples or features
 - ♦ Common modalities
 - Fingerprint
 - Facial recognition
 - Palm print
 - Iris scan
 - Voice recognition
 - DNA
- ▷ Measures are compared with personal records
 - ♦ Biometric references (or template)
 - ♦ Registered in the system with a previous enrolment procedure



Biometrics: advantages

- ▷ Convenient: people do not need to use memory
 - ♦ Just be their self
- ▷ People cannot chose weak passwords
 - ♦ In fact, they don't chose anything
- ▷ Credentials cannot be transferred to others
 - ♦ One cannot delegate their own authentication
- ▷ Stealth identification
 - ♦ Interesting for security surveillance



Biometrics: problems



- ▷ Usability
 - ♦ Comfort of people, ergonomic
 - ♦ Exploitation scenario
- ▷ Biometrics are still being improved
 - ♦ In many cases they can be easily cheated
 - ♦ Liveness detection
- ▷ People cannot change their credentials
 - ♦ Upon their robbery
- ▷ It can be risky for people
 - ♦ Removal of body parts for impersonation of the victim



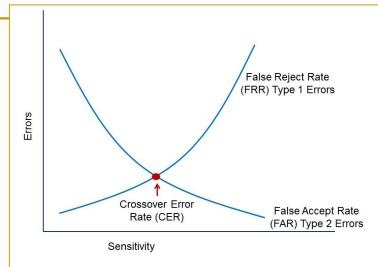
© André Zúquete

Identification, Authentication and Authorization

Image source: <https://biometrics.manguet.org/types/tongue.htm>

17

Biometrics: problems



- ▷ Sensitivity tuning
 - ♦ Reduction of FRR (annoying)
 - ♦ Reduction of FAR (dangerous)
 - ♦ Tuning is mainly performed with the target population
 - Not with attackers!
- ▷ Not easy to deploy remotely
 - ♦ Requires trusting the remote sample acquisition system
- ▷ Can reveal personal sensitive information
 - ♦ Diseases
- ▷ Credentials cannot be (easily) copied to others
 - ♦ In case of need in exceptional circumstances



© André Zúquete

Identification, Authentication and Authorization

Image source: <http://www.pearsonitcertification.com/articles/article.aspx?p=1718488>

18

Authentication of people: Direct approach with OTPs

- ▷ One-time password (OTP)
 - ♦ Credential that can be used only once
- ▷ Advantage
 - ♦ OTPs can be eavesdropped
 - ♦ Eavesdroppers cannot impersonate the OTP owner
 - True for passive eavesdroppers
 - False for active attackers!



Authentication of people: Direct approach with OTPs

- ▷ Problems
 - ♦ Interactors need to know which password they should use at different occasions
 - Requires some form of synchronization
 - ♦ People may need to use extra resources to maintain or generate one-time passwords
 - Paper sheets
 - Computer programs
 - Special devices, etc.



Authentication of people: OTPs and secondary channels

- ▷ OTPs are codes sent through secondary channels
 - ♦ A secondary channel is a channel that is not the one where the code is going to be used
 - SMS, email, Twitter, Firebase, QR codes, NFC, etc.
 - ♦ The secondary channel provides the synchronization
 - Just-in-time provision of OTP
- ▷ Two authentications are possible
 - ♦ Confirm a secondary channel provided by a profile owner
 - In order to trust that that channel belongs to the profile owner
 - ♦ Authenticate the owner of a profile
 - Which is bound to a secondary channel

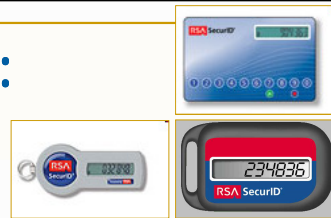


Authentication of people: OTPs produced from a shared key

- ▷ HOTP (Hash-based One Time Password, RFC 4226)
 - ♦ OTP generated from a counter and a shared key
 - ♦ Counters are updated independently
- ▷ TOTP (Time-based One Time Password, RFC 6238)
 - ♦ OTP generated from a timestamp and a shared password
 - ♦ TOTP is HOTP with timestamps instead of counters
 - ♦ Clocks need a rough synchronization



Token-based OTP generators: RSA SecurID



- ▷ Personal authentication token
 - ♦ Or software modules for handhelds (PDAs, smartphones, etc.)
- ▷ It generates a unique number at a fixed rate
 - ♦ Usually one per minute (or 30 seconds)
 - ♦ Bound to a person (User ID)
 - ♦ Unique number computed with:
 - A 64-bit key stored in the token
 - The actual timestamp
 - A proprietary digest algorithm (SecurID hash)
 - An extra PIN (only for some tokens)



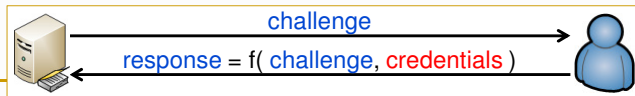
RSA SecurID

- ▷ OTP-based authentication
 - ♦ A user combines their User ID with the current token number
$$\text{OTP} = \text{User ID}, \text{Token Number}$$
- ▷ An RSA ACE Server does the same and checks for match
 - ♦ It also knows the person's key stored in the token
 - ♦ There must be a synchronization to tackle clock drifts
 - RSA Security Time Synchronization
- ▷ Robust against dictionary attacks
 - ♦ Keys are not selected by people



Challenge-response approach: Generic description

- ▷ The authenticator provides a challenge
- ▷ The entity being authenticated transforms the challenge
 - ♦ With its authentication credentials
- ▷ The result (response) is sent to the authenticator
- ▷ The authenticator checks the response
 - ♦ Produces a similar result and checks if they match
 - ♦ Transforms the result and checks if it matches the challenge or a related value



© André Zúquete

Identification, Authentication and Authorization

25

Challenge-response approach: Generic description

- ▷ Advantage
 - ♦ Authentication credentials are not exposed
- ▷ Problems
 - ♦ People may require means to compute responses
 - Hardware or software
 - ♦ The authenticator may have to have access to shared secrets
 - How can we prevent them from using the secrets elsewhere?
 - ♦ Offline dictionary attacks
 - Against recorded challenge-response dialogs
 - Can reveal secret credentials (passwords, keys)



© André Zúquete

Identification, Authentication and Authorization

26

Challenge-response protocols: selection of challenges

- ▷ Challenges cannot be repeated for the same entity
 - ♦ Same challenge → same response
 - ♦ An active attacker can impersonate a user using a previously recorded protocol run
- ▷ Challenges should be nonces
 - ♦ Nonce: number used only once
 - ♦ Stateful services can use counters
 - ♦ Stateless services can use (large) random numbers
 - ♦ Time can be used, but with caution
 - Because one cannot repeat a timestamp

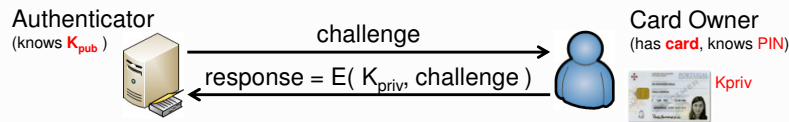


Authentication of people: Challenge-response with smartcards

- ▷ Authentication credentials
 - ♦ The smartcard
 - e.g. Citizen Card
 - ♦ The private key stored in the smartcard
 - ♦ The PIN to unlock the private key
- ▷ The authenticator knows
 - ♦ The corresponding public key
 - ♦ Or some personal identifier
 - Which can be related with a public key through a (verifiable) certificate



Authentication of people: Challenge-response with smartcards



▷ Signature-based protocol

- ♦ The authenticator generates a random challenge
 - Or a value not used before
- ♦ The card owner ciphers the challenge with their private key
 - PIN-protected
- ♦ The authenticator decrypts the result with the public key
 - If the output matches the challenge, the authentication succeeds

▷ Encryption-based protocol

- ♦ Possible when private key decryption is available



Authentication of people: Challenge-response with memorized password

▷ Authentication credentials

- ♦ Passwords selected by people

▷ The authenticator knows

- ♦ All the registered passwords; or
- ♦ A transformation of each password
 - Preferable option
 - Preferably combined with some local value (salt)
 - Preferable using a tunable function (e.g. iterations)



Authentication of people:

Challenge-response with memorized password

- ▷ The authenticator generates a random challenge
- ▷ The person computes a function of the challenge and password
 - e.g. a joint digest: $\text{response} = \text{digest}(\text{challenge}, \text{password})$
 - e.g. an encryption $\text{response} = E_{\text{password}}(\text{challenge})$
- ▷ The authenticator does the same (or the inverse)
 - If the output matches the response (or the challenge), the authentication succeeds
- ▷ Examples
 - CHAP, MS-CHAP v1/v2, S/Key



PAP & CHAP

(RFC 1334, 1992, RFC 1994, 1996)

- ▷ Protocols used in PPP (Point-to-Point Protocol)
 - Unidirectional authentication
 - Authenticator is not authenticated
- ▷ PPP developed in 1992
 - Mostly used for dial-up connections
- ▷ PPP protocols are used by PPTP VPNs
 - e.g. vpn.ua.pt



PAP & CHAP

(RFC 1334, 1992, RFC 1994, 1996)

▷ PAP (PPP Authentication Protocol)

- ♦ Simple UID/password presentation
- ♦ Insecure cleartext password transmission

▷ CHAP (CHallenge-response Authentication Protocol)

Aut → U: authID, challenge

U → Aut: authID, MD5(authID, pwd, challenge), identity

Aut → U: authID, OK/not OK

- ♦ The authenticator may require a reauthentication anytime



MS-CHAP (Microsoft CHAP)

(RFC 2433, 1998, RFC 2759, 2000)

▷ Version 1

A → U: authID, C

U → A: R1, R2

A → U: OK/not OK

$R1 = DES_{LMPH}(C)$

$R2 = DES_{NTPH}(C)$

$LMPH = DEShash(pwd')$

$NTPH = MD4(pwd)$

$pwd' = capitalized(pwd)$

▷ Version 2

A → U: authID, C_A

U → A: C_U, R1

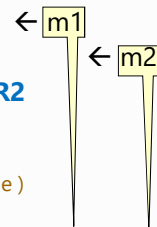
A → U: OK/not OK, R2

$R1 = DES_{PH}(C)$

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

$PH = MD4(password)$

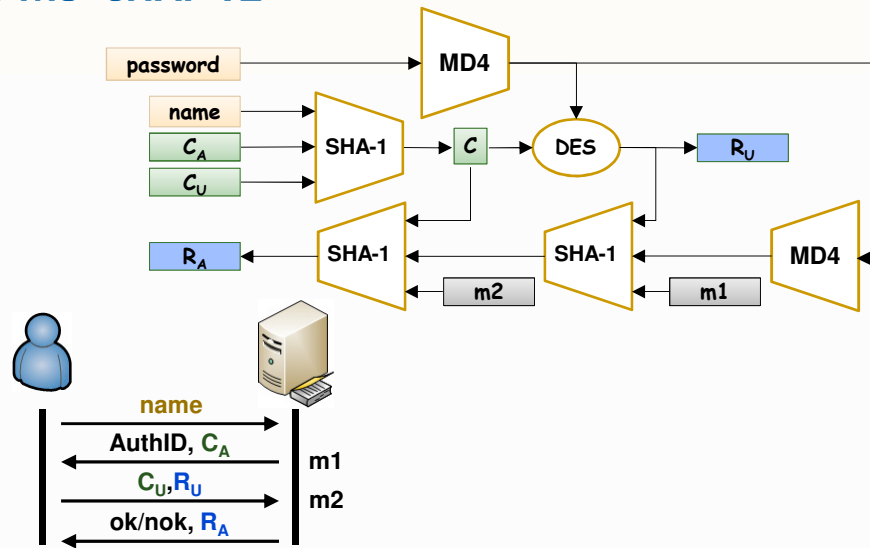
$R2 = SHA(SHA(MD4(PH), R1, m1), C, m2)$



- Mutual authentication
- Passwords can be updated



MS-CHAP v2



© André Zúquete

Identification, Authentication and Authorization

35

Authentication of people: Generation of OTPs with challenges

- ▷ OTPs can be produced from a challenge received
 - ♦ The fundamental protocol is password-based
 - But passwords are OTPs
 - ♦ OTPs are produced from a challenge
 - ♦ One can use several algorithms to handle OTPs



© André Zúquete

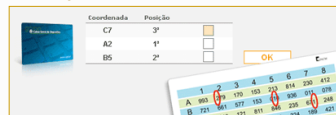
Identification, Authentication and Authorization

36

Authentication of people: OTPs selected from shared data

- ▷ Advantage:
 - ♦ Shared data can be random
 - ♦ No long-term short secrets to protect

- ▷ OTPs build from printed data
 - ♦ Example: online bank codes



- ▷ Selection of an OTP from a printed / saved list

TPM list generated 2020-12-01 14:10 on ubuntu

```

000 3070 3a7e 054 0f8e e9ed 112 7e93 8888 169 h9p8 2a7e 224 M80n w8d1
001 e47e 8791 037 808e 679e 113 d0e6 85a3 169 M07Y eazM 225 08a5 8a7e
002 837a 8a82 038 a99e 979e 114 9f0e 2381 170 M07 /a8d 226 47e2 1393
003 P00Y 8e74 039 a00f 158e 115 1f9e 85a3 171 8a7e 2a7e 227 7e9e w8p0
004 a90f 79e0 040 809e 78e0 116 8280 8083 172 1a7e 2a88 228 470e a7a8
005 8884 889e 041 1a9e 2e18 117 988e 879e 173 48a7 8881 229 04a5 od M
006 8e7e 079e 042 979e 78e0 118 w0e0 8882 174 a7e 2a88 230 54a7 97a8
007 888e 889e 043 989e e819 119 e1a5 8891 175 8a7e 0a14 231 2e0e a7a7
008 889e 889e 044 989e 78e0 120 88e0 8882 176 8e 0a14 232 8a7e
009 888e 888e 045 1e9e e8e0 121 8e 9 97e5 177 a7e 8a7e 233 e830 88e0
010 888e 8e7e 046 8e7e 979e 122 888e 8882 178 888e 2a7e 234 8e7e 88p0
011 988e 88e0 047 888e 888e 123 888e 888e 179 8a7e 88e0 235 0a7e 88e0
012 988e 88e0 048 888e 888e 124 888e 888e 180 888e 888e 236 888e 888e
013 988e 88e0 049 888e 888e 125 888e 888e 181 888e 888e 237 888e 888e
014 988e 88e0 050 888e 888e 126 888e 888e 182 888e 888e 238 888e 888e
015 988e 88e0 051 888e 888e 127 888e 888e 183 888e 888e 239 888e 888e
016 988e 88e0 052 888e 888e 128 888e 888e 184 888e 888e 240 888e 888e
017 888e 88e0 053 888e 888e 129 888e 888e 185 888e 888e 241 888e 888e
018 888e 88e0 054 888e 888e 130 888e 888e 186 888e 888e 242 888e 888e
019 888e 88e0 055 888e 888e 131 888e 888e 187 888e 888e 243 888e 888e
020 888e 88e0 056 888e 888e 132 888e 888e 188 888e 888e 244 888e 888e
021 888e 88e0 057 888e 888e 133 888e 888e 189 888e 888e 245 888e 888e
022 888e 88e0 058 888e 888e 134 888e 888e 190 888e 888e 246 888e 888e
023 888e 88e0 059 888e 888e 135 888e 888e 191 888e 888e 247 888e 888e
024 888e 88e0 060 888e 888e 136 888e 888e 192 888e 888e 248 888e 888e
025 888e 88e0 061 888e 888e 137 888e 888e 193 888e 888e 249 888e 888e
026 888e 88e0 062 888e 888e 138 888e 888e 194 888e 888e 250 888e 888e
027 888e 88e0 063 888e 888e 139 888e 888e 195 888e 888e 251 888e 888e
028 888e 88e0 064 888e 888e 140 888e 888e 196 888e 888e 252 888e 888e
029 888e 88e0 065 888e 888e 141 888e 888e 197 888e 888e 253 888e 888e
030 888e 88e0 066 888e 888e 142 888e 888e 198 888e 888e 254 888e 888e
031 888e 88e0 067 888e 888e 143 888e 888e 199 888e 888e 255 888e 888e
032 888e 88e0 068 888e 888e 144 888e 888e 200 888e 888e 256 888e 888e
033 888e 88e0 069 888e 888e 145 888e 888e 201 888e 888e 257 888e 888e
034 888e 88e0 070 888e 888e 146 888e 888e 202 888e 888e 258 888e 888e
035 888e 88e0 071 888e 888e 147 888e 888e 203 888e 888e 259 888e 888e
036 888e 88e0 072 888e 888e 148 888e 888e 204 888e 888e 260 888e 888e
037 888e 88e0 073 888e 888e 149 888e 888e 205 888e 888e 261 888e 888e
038 888e 88e0 074 888e 888e 150 888e 888e 206 888e 888e 262 888e 888e
039 888e 88e0 075 888e 888e 151 888e 888e 207 888e 888e 263 888e 888e
040 888e 88e0 076 888e 888e 152 888e 888e 208 888e 888e 264 888e 888e
041 888e 88e0 077 888e 888e 153 888e 888e 209 888e 888e 265 888e 888e
042 888e 88e0 078 888e 888e 154 888e 888e 210 888e 888e 266 888e 888e
043 888e 88e0 079 888e 888e 155 888e 888e 211 888e 888e 267 888e 888e
044 888e 88e0 080 888e 888e 156 888e 888e 212 888e 888e 268 888e 888e
045 888e 88e0 081 888e 888e 157 888e 888e 213 888e 888e 269 888e 888e
046 888e 88e0 082 888e 888e 158 888e 888e 214 888e 888e 270 888e 888e
047 888e 88e0 083 888e 888e 159 888e 888e 215 888e 888e 271 888e 888e
048 888e 88e0 084 888e 888e 160 888e 888e 216 888e 888e 272 888e 888e
049 888e 88e0 085 888e 888e 161 888e 888e 217 888e 888e 273 888e 888e
050 888e 88e0 086 888e 888e 162 888e 888e 218 888e 888e 274 888e 888e
051 888e 88e0 087 888e 888e 163 888e 888e 219 888e 888e 275 888e 888e
052 888e 88e0 088 888e 888e 164 888e 888e 220 888e 888e 276 888e 888e
053 888e 88e0 089 888e 888e 165 888e 888e 221 888e 888e 277 888e 888e
054 888e 88e0 090 888e 888e 166 888e 888e 222 888e 888e 278 888e 888e
055 888e 88e0 091 888e 888e 167 888e 888e 223 888e 888e 279 888e 888e

```



S/Key (RFC 2289, 1998)

- ▷ Authentication credentials
 - ♦ A password (pwd)
- ▷ The authenticator knows
 - ♦ The last used one-time password (OTP)
 - ♦ The last used OTP index
 - Defines an order among consecutive OTPs
 - ♦ An seed value for the each person's OTPs
 - The seed is similar to a UNIX salt



S/Key setup

- ▷ The authenticator defines a random seed
- ▷ The person generates an initial OTP as:
$$OTP_n = h^n(\text{seed}, \text{pwd}), \text{ where } h = \text{MD4}$$
 - ♦ Some S/Key versions also use MD5 or SHA-1
- ▷ The authenticator stores seed, n and OTP_n as authentication credentials



S/Key authentication protocol

- ▷ Authenticator sends seed & index of the person
 - ♦ They act as a challenge
- ▷ The person generates index-1 OTPs in a row
 - ♦ And selects the last one as result
 - ♦ $\text{result} = OPT_{\text{index}-1}$
- ▷ The authenticator computes h(result) and compares the result with the stored OPT_{index}
 - ♦ If they match, the authentication succeeds
 - ♦ Upon success, stores the recently used index & OTP
 - index-1 and $OPT_{\text{index}-1}$



S/Key

▷ Advantages

- ♦ Users passwords are unknown to authenticators
- ♦ OTPs can be used as ordinary passwords

▷ Disadvantages

- ♦ People need an application to compute OTPs
- ♦ Passwords can be derived using dictionary attacks
 - From data stored in authenticators
 - From captured protocol runs



Authentication of people: Challenge-response with shared key

▷ Uses a shared key instead of a password

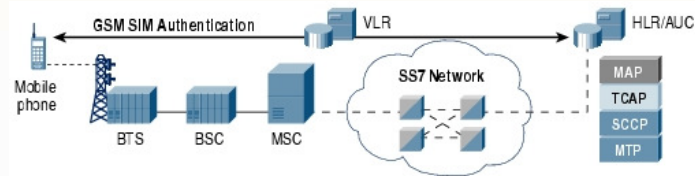
- ♦ Robust against dictionary attacks
- ♦ Requires some token to store the key

▷ Example:

- ♦ GSM



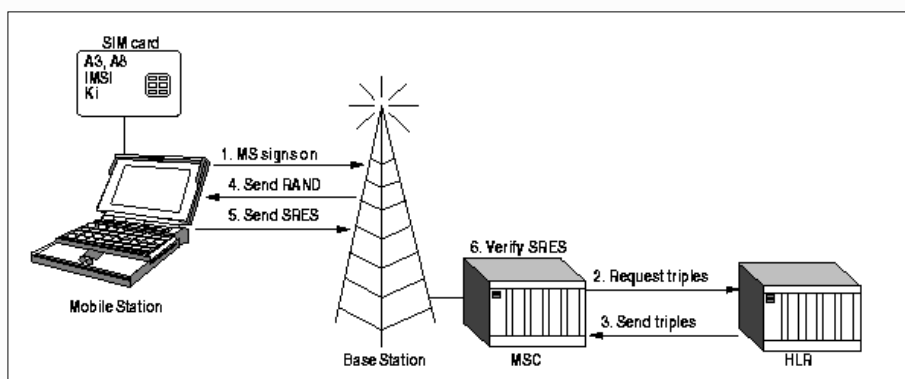
GSM: authentication architecture



- ▷ Based on a secret key shared between the HLR and the station
 - 128 Ki, stored in the station's SIM card
 - Can only be used after entering a PIN
- ▷ Algorithms (initially not public):
 - A3 for authentication
 - A8 for generating a session key
 - A5 for encrypting the communication
- ▷ A3 and A8 implemented by SIM card
 - Can be freely selected by the operator



GSM: mobile station authentication



GSM: mobile station authentication

- ▷ MSC fetches trio from HLR
 - ♦ **RAND, SRES, Kc**
 - ♦ In fact more than one are requested
- ▷ HLR generates RAND and corresponding trio using subscriber's Ki
 - ♦ **RAND**, random value (128 bits)
 - ♦ **SRES = A3 (Ki, RAND)** (32 bits)
 - ♦ **Kc = A8 (Ki, RAND)** (64 bits)
- ▷ Usually operators use COMP128 for A3/A8
 - ♦ Recommended by the GSM Consortium
 - ♦ **[SRES, Kc] = COMP128 (Ki, RAND)**



Host authentication

- ▷ By name or address
 - ♦ DNS name, IP address, MAC address, other
 - ♦ Extremely weak, no cryptographic proofs
 - Nevertheless, used by many services
 - e.g. NFS, TCP wrappers
- ▷ With cryptographic keys
 - ♦ Keys shared among peers
 - With an history of usual interaction
 - ♦ Per-host asymmetric key pair
 - Pre-shared public keys with usual peers
 - Certified public keys with any peer



Service / server authentication

▷ Host authentication

- ♦ All co-located services/servers are indirectly authenticated

▷ Per-service/server credentials

- ♦ Shared keys
 - When related with the authentication of people
 - The key shared with each person can be used to authenticate the service to that person
- ♦ Per-service/server asymmetric key pair
 - Certified or not



TLS (Transport Layer Security, RFC 8446)

▷ Secure communication protocol over TCP/IP

- ♦ Created upon SSL V3 (Secure Sockets Layer)
- ♦ Manages per-application secure sessions over TCP/IP
 - Initially conceived for HTTP traffic
 - Actually used for other traffic types

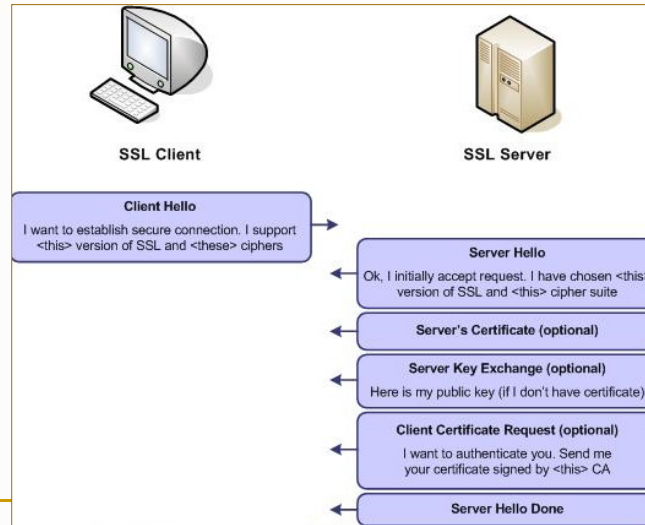
▷ There is a similar version for UDP (DTLS, RFC 6347)

▷ Security mechanisms

- ♦ Communication confidentiality and integrity
 - Key distribution
- ♦ Authentication of communication endpoints
 - Servers (or, more frequently, services)
 - Client users
 - Both with asymmetric key pairs and certified public keys



SSL/TLS interaction diagrams (1st part)

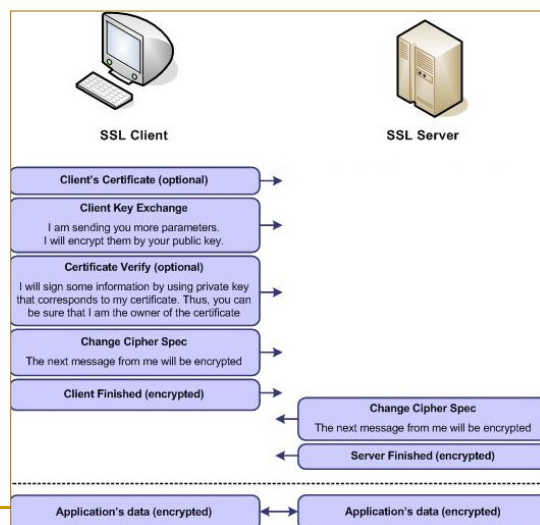


© André Zúquete

Identification, Authentication and Authorization

49

SSL/TLS interaction diagrams (2nd part)



© André Zúquete

Identification, Authentication and Authorization

50

SSH (Secure Shell, RFC 4251)

▷ Alternative to telnet/rlogin protocols/applications

- Manages secure consoles over TCP/IP
- Initially conceived to replace telnet
- Actually used for other applications
 - Secure execution of remote commands (rsh/rexec)
 - Secure copy of contents between machines (rcp)
 - Secure FTP (sftp)
 - Creation of arbitrary secure tunnels (inbound/outbound/dynamic)

▷ Security mechanisms

- Communication confidentiality and integrity
 - Key distribution
- Authentication of communication endpoints
 - Servers / machines
 - Client users
 - Both with different techniques



SSH authentication mechanisms

▷ Server: with asymmetric keys pair

- Inline public key distribution
 - Not certified!
- Clients cache previously used public keys
 - Caching should occur in a trustworthy environment
 - Update of a server's key raises a problem to its usual clients

▷ Client users: configurable

- Username + password
 - By default
- Username + private key
 - Upload of public key in advance to the server



Single Sign-On (SSO)

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



Authentication metaprotocols

- ▷ Generic authentication protocols that encapsulate other authentication protocols
- ▷ Examples
 - ♦ EAP (Extensible Authentication Protocol)
 - Used in 802.1X (Wi-Fi, enterprise mode)
 - e.g. PEAP (Protected EAP) and EAP-TLS run over EAP
 - ♦ ISAKMP (Internet Security Association and Key Management Protocol)
 - Used in IPSec
 - e.g. IKE (Internet Key Exchange) runs over ISAKMP



Authentication services

- ▷ Trusted third parties (TTP) used for authentication
 - ♦ But often combined with other related functionalities
- ▷ AAA services
 - ♦ Authentication, Authorization and Accounting
 - ♦ e.g. RADIUS



Key distribution services

- ▷ Services that distribute a shared key for authenticated entities
 - ♦ That key can then be used by those entities to protect their communication and ensure source authentication
- ▷ Examples
 - ♦ 802.1X (Wi-Fi, enterprise mode)
 - ♦ Kerberos

