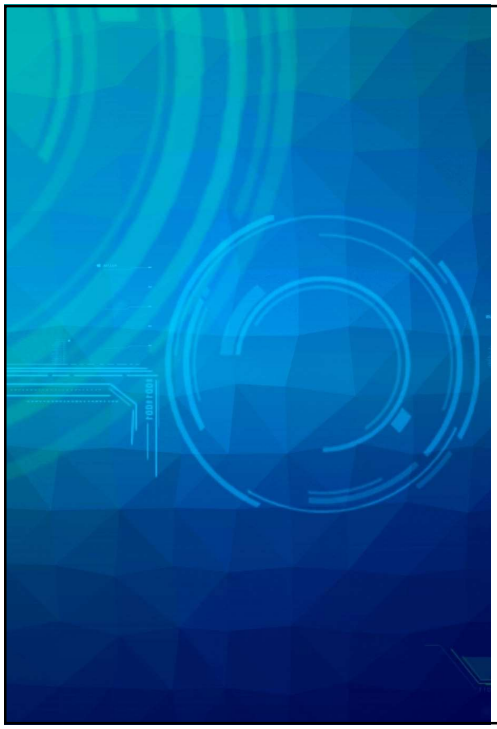
1. Identifies the user and 2. Verifies that the identity declared matches with the one of the user.

1

Learning objectives

- Discuss the four general means of authenticating a user's identity.
- Explain the mechanism by which **hashed passwords** are used for user authentication.
- Present an overview of **token-based user authentication**.
- Introduce the basics of **biometric authentication**.
- Discuss the issues involved and the approaches for **remote user authentication**.
- Summarize some of the key **security issues** for user authentication.

2 We can deal with direct authentication without a medium role. With remote u.a. we introduce the problem of networking that can be compromised.



Preliminary question

WHEN DO YOU THINK YOU HAVE BEEN SUBJECT TO USER AUTHENTICATION IN YOUR EXPERIENCE?

CAN YOU IDENTIFY THE TWO PHASES OF IDENTIFICATION AND VERIFICATION?

3

Two functions for user authentication

1. User identification
   - by means of a credential or an ID provided by the user to the system
2. User verification
   - by the exchange of authentication information
   - establishes the validity of the claim

- Note: user authentication is distinct from message authentication!

4

Digital Authentication Guideline

NIST SP 800-63-3 defines digital user authentication as (October 2016) :

"The process of establishing confidence in user identities that are presented electronically to an information system."

almost like we can make mistakes

5

Identification and authentication security requirements

Basic Security Requirements (NIST SP 800-171):

1. Identify users, processes acting on behalf of users, or devices.
2. Authenticate (or verify) the identities of those users, processes, or devices
   - prerequisite to allowing access to organizational information systems.

there are digital proxies that act on behalf of the user. For example, after a login, you have processes acting on your behalf.

↳ This doc. refers to security issues at any level.

6

if user authenticating is associated to privileged accounts we need multifactor auth. If the privileges are lower just for the network.

This document is about authentication at all levels, give the proper flexibility for that

Identification and authentication security requirements

Derived Security Requirements:

3. Use multifactor authentication for local and network access to privileged accounts and for network access to non-privileged accounts.
4. Employ replay-resistant authentication mechanisms for network access to all accounts. ↳ session steal with packets
5. Prevent reuse of identifiers for a defined period.
6. Disable identifiers after a defined period of inactivity.
7. Enforce a minimum password complexity and change of characters when new passwords are created.
8. Prohibit password reuse for a specified number of generations.
9. Allow temporary password use for system logons with an immediate change to a permanent password.
10. Store and transmit only cryptographically-protected passwords.
11. Obscure feedback of authentication information.

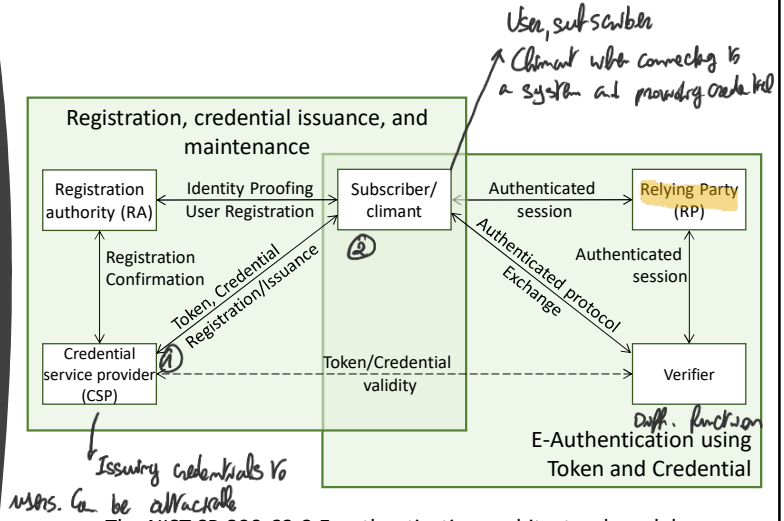
Related to policies

A detailed feedback can be exploited by an attacker.

7

Authentication architectural model (NIST)

Shows how auth processes are organized.



Issuing credentials to users. Can be attackable

The NIST SP 800-63-3 E-authentication architectural model

8

The subscr. has to connect to a Reg. Authority which identifies the identity of the subscriber. Can happen IRL by showing a document for ex. (step 1). Step 2: the Authority requests for credential to ①. Credentials can be PW, biometrics exc. To obs that we need an extra step for issuing credentials (like needing for taking biometrics or just giving credentials). It can be done physically, digitally or mixed.

Once this is done, the subscriber can use services. We connect to the Verifier to provide the credentials (the process may require interaction with @, like getting redirected to SPID website, I provide the credentials to the CSP (1) if they have an agreement)
and then we have access to an authenticated session.

four means of authenticating user identity

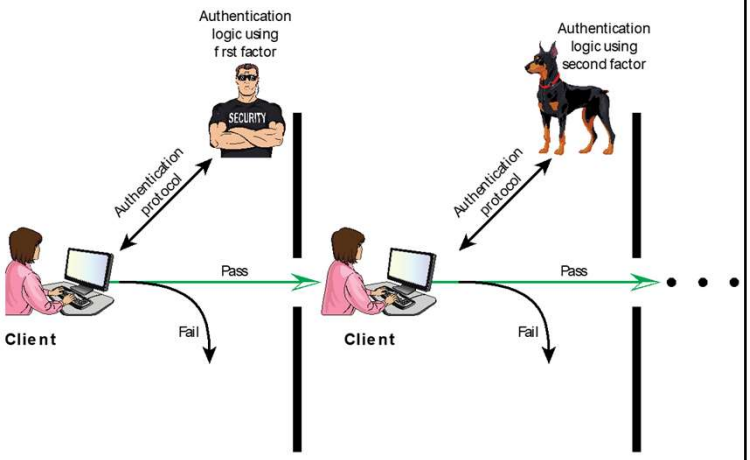
- Something the individual knows — Password, PIN, answers to prearranged questions
- Something the individual possesses (token) — Smartcard, electronic keycard, physical key
- Something the individual is (static biometrics) — Fingerprint, retina, face
- Something the individual does (dynamic biometrics) — Voice pattern, handwriting, typing rhythm

Can be lost or stolen

Problems with false negatives and false positives, depending on the methods.
Problem to store biometrics, you can change password but not biometry

9

Multifactor authentication



10

Risk assessment for user authentication

The inherent risk needs to be assessed. In 3 steps

Three separate concepts:

- **Assurance Level** (Likelihood of a threat to become reality)
- **Potential impact**
- **Areas of risk**

11

Assurance level

↑ The confidence level can be given to an auth system.

Describes an organization's **degree of certainty** that a user has presented a credential that refers to his or her identity

This degree is defined as:

- The degree of confidence in the vetting process used to establish the identity of the individual to whom the credential was issued
- The degree of confidence that the individual who uses the credential is the individual to whom the credential was issued

Four levels of assurance

- Level 1 •Little or no confidence in the asserted identity's validity → Case of supermarket card
- Level 2 •Some confidence in the asserted identity's validity → We don't have very stringent requirements (1 factor to be used)
- Level 3 •High confidence in the asserted identity's validity → We have higher requirements (double factor) (Not HUGE IMPACT BUT HIGH)
- Level 4 •Very high confidence in the asserted identity's validity → We need very high confidence, strong credentials and 2+ factors.

The process of giving credentials to a user

Evaluating is complicated. Not done at NIST way but in a qualitative way with the four levels.

12

Potential impact

- FIPS 199 defines three levels of potential impact on organizations or individuals should there be a breach of security:
  - Low
    - An authentication error could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals
  - Moderate
    - An authentication error could be expected to have a serious adverse effect
  - High
    - An authentication error could be expected to have a severe or catastrophic adverse effect

13

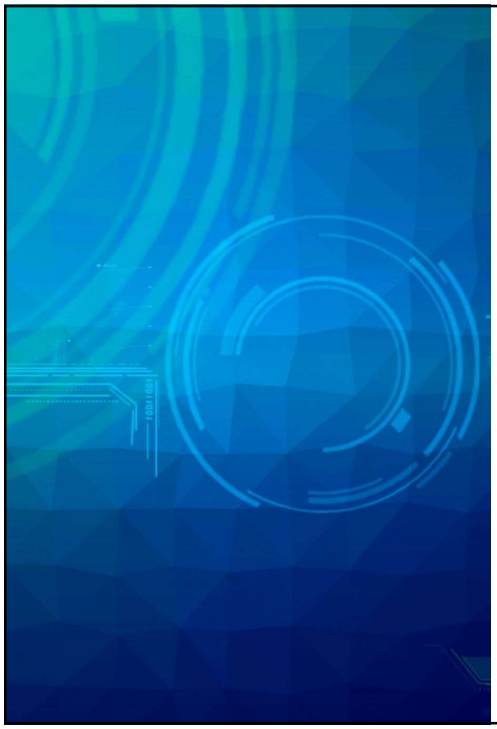
Areas of risk: Maximum potential impact for each assurance level

	Assurance Level Impact Profiles			
Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
Inconvenience, distress, or damage to standing or reputation	Low	Mod	Mod	High
Financial loss or organization liability	Low	Mod	Mod	High
Harm to organization programs or interests	None	Low	Mod	High
Unauthorized release of sensitive information	None	Low	Mod	High
Personal safety	None	None	Low	Mod/High
Civil or criminal violations	None	Low	Mod	High

Categories of potential impact

Based on it we can adopt a certain assurance level.

14



Review question

REFERRING TO THE NIST SP 800-63-3 MODEL, DID YOU EVER EXPERIENCE A CASE OF AUTHENTICATION THAT FOLLOWS THAT MODEL?
DESCRIBE THAT AUTHENTICATION SYSTEM FROM THE USER PERSPECTIVE.

15

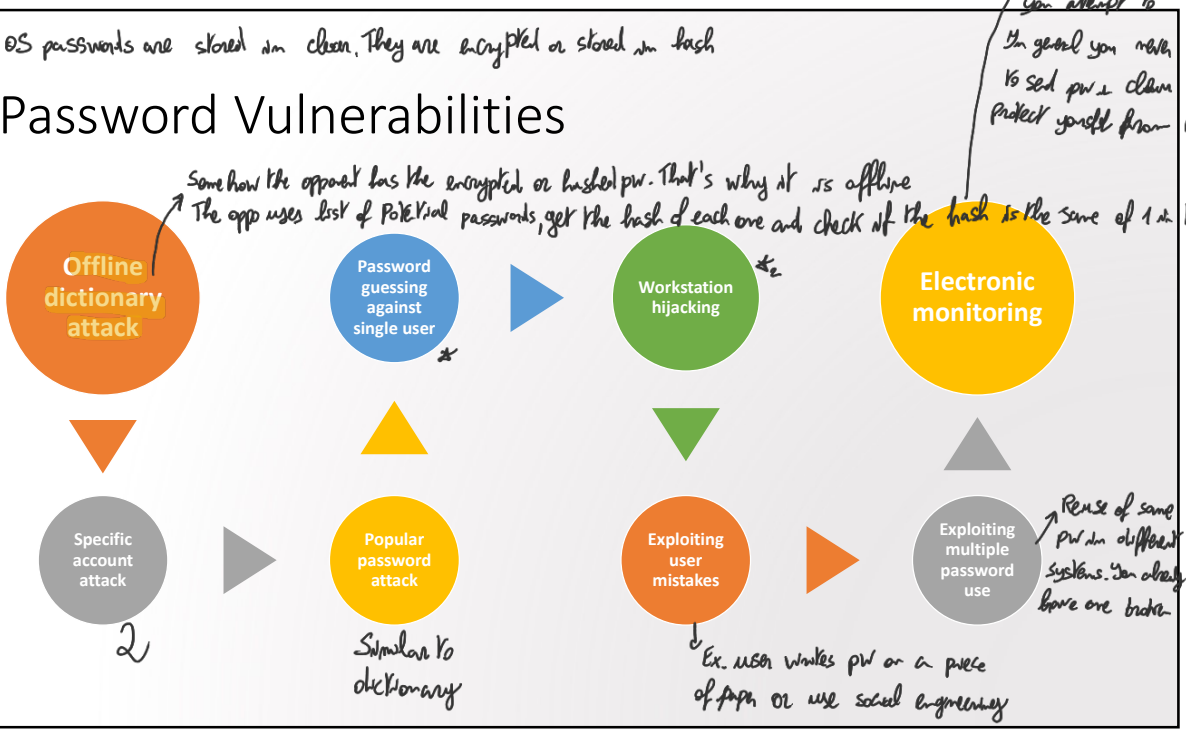
Password-based authentication

Very simple! That's why it was so successful.

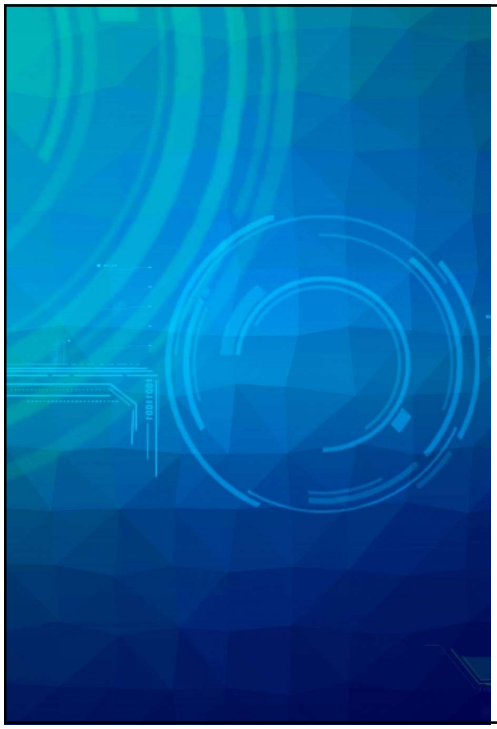
- Widely used line of defense against intruders
  - User provides name/login and password
  - System compares password with the one stored for that specified login
- The user ID: is associated to identity but also the privileges.
  - Determines that the user is authorized to access the system
  - Determines the user's privileges
  - Is used in discretionary access control
    ↳ method of access control in OS.

16

Password Vulnerabilities



17



Question

?

AN ADVERSARY PERFORMS A BRUTE FORCE ATTACK AND IN 1 YEAR TESTS EXHAUSTIVELY ALL THE POTENTIAL PINS OF A USER.

IF THE PINS ARE OF 6 DIGITS, EACH IN AN ALPHABET OF 10 SYMBOLS, HOW LONG IT TAKES TO TEST ONE PIN?

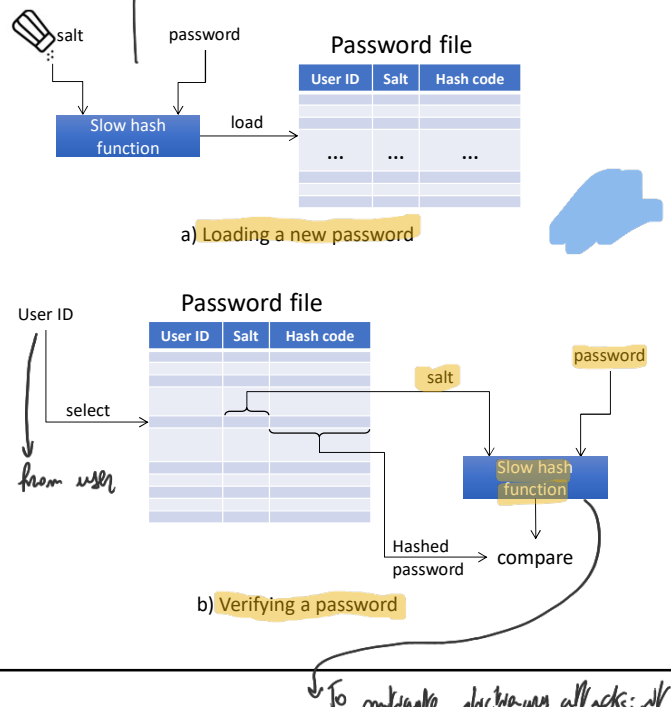
1000 000

1. Countermeasures: keep pw on a protected space and use a good hash function.

2. We could attempt an attack to a casual user or target a specific user. Mitigate online attacks: introduce delays and block.

When PW is created PW is mixed with salt and we feed it into the slow hash function.

Unix password scheme



To mitigate dictionary attacks: it will take a lot to check a lot of passwords.

19

Without salt we could use hashed dictionaries by themselves.

Unix password scheme

The salt serves three purposes:

- It prevents duplicate passwords from being visible in the password file.
- It becomes nearly impossible to find out whether a person with passwords on two or more systems has used the same password on all of them.
- It greatly increases the difficulty of offline dictionary attacks.
  - Say the attacker obtains a copy of the password file and salt is not used
    - … and its goal is to guess a single password of any user.
  - The attacker may submit a large number of likely passwords to the hashing function:
    - If any of the guesses matches one of the hashes in the file, then the attacker has found a password that is in the file.
  - Using salt instead, the attacker must take each guess and submit it to the hash function once **for each** salt value in the dictionary file, multiplying the number of guesses that must be checked.

↳ You are building a rainbow table associated to each salt

20

Unix implementation

Original scheme

- Up to eight printable characters in length
- 12-bit salt used to modify DES encryption into a one-way hash function
- Repeatedly encrypted 25 times
- Output translated to 11 character sequence

Was possible to implement a rainbow attack

Now regarded as inadequate

- Still often required for compatibility with existing account management software or multivendor environments

21

Improved implementations

Much stronger hash/salt schemes available for Unix

Recommended hash function is based on MD5

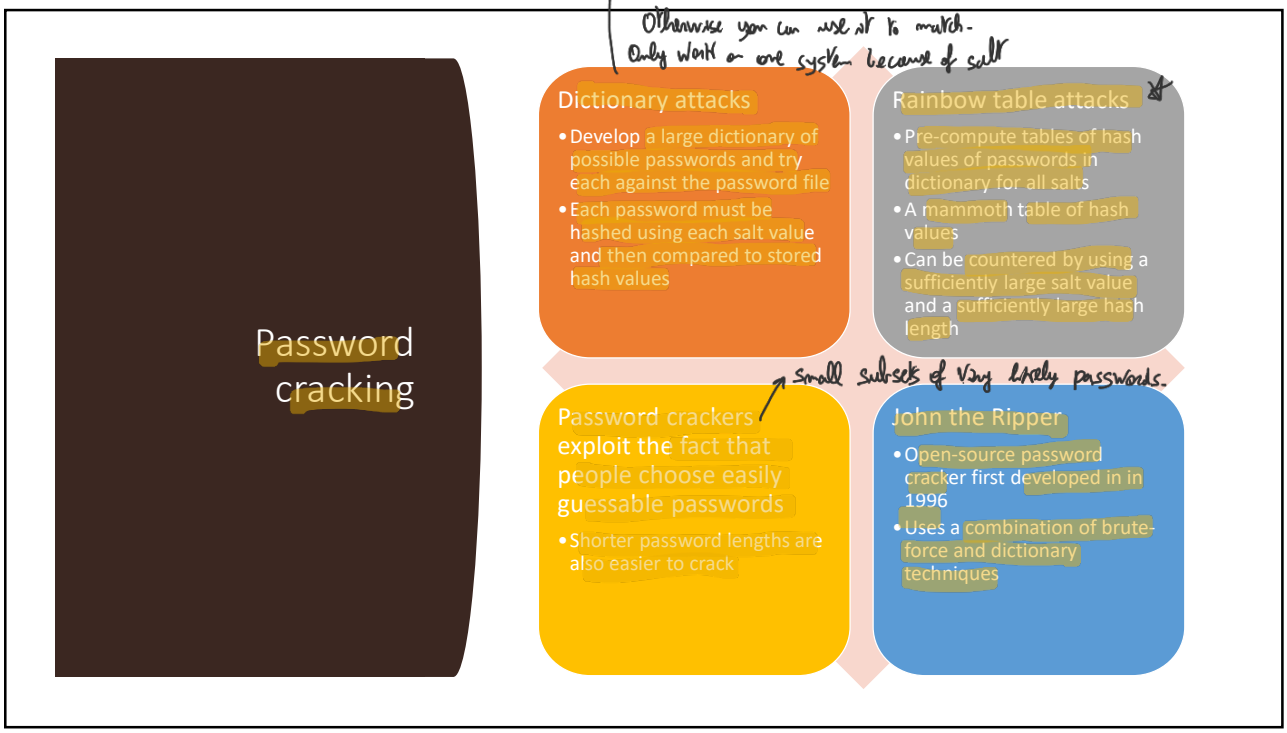
- Salt of up to 48-bits
- Password length is unlimited
- Produces 128-bit hash
- Uses an inner loop with 1000 iterations to achieve slowdown

OpenBSD uses Blowfish block cipher-based hash algorithm called Bcrypt (this kind of parameter you could use to slow it down, the next user had a slower attack.

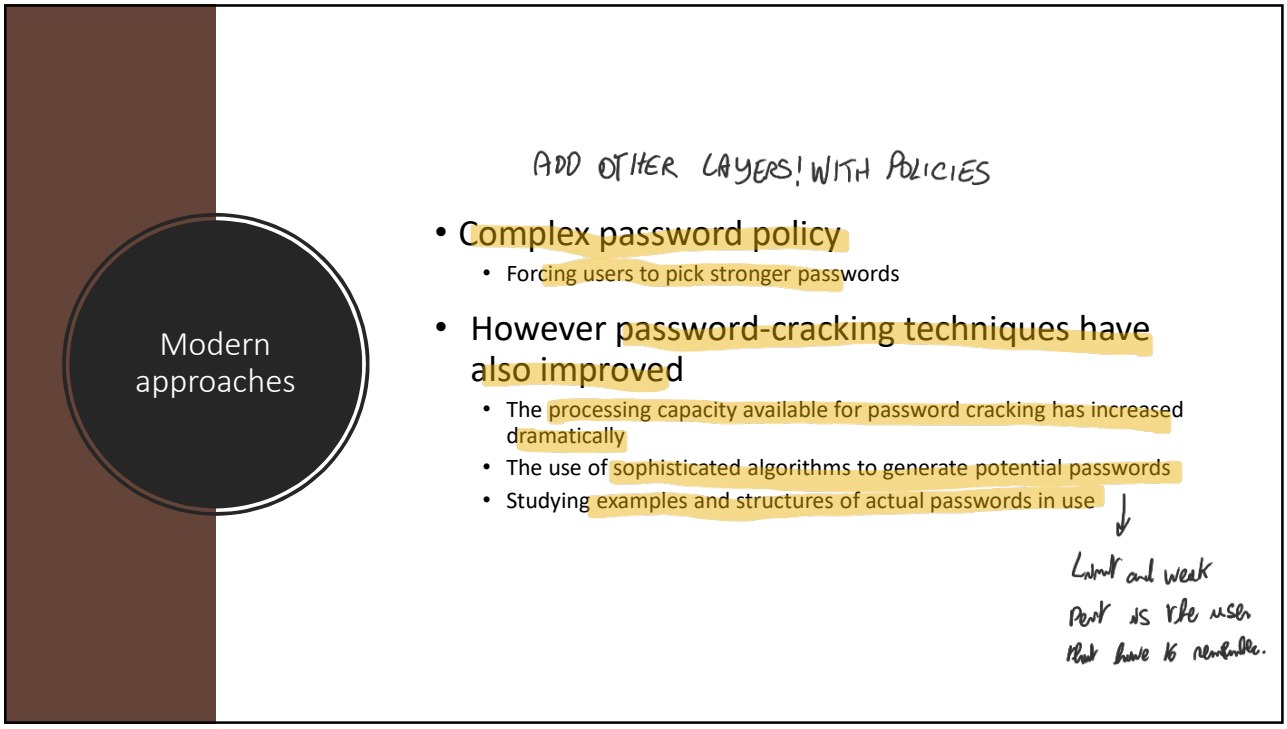
- Most secure version of Unix hash/salt scheme
- Uses 128-bit salt to create 192-bit hash value

... and the story goes on ...

22

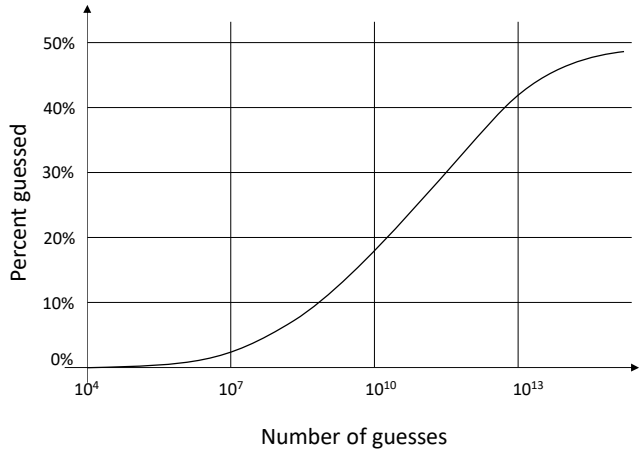


23 * Table can be used on any system!



24

Percentage of passwords guess after a given number of guesses



Experiment: trying PW on a system with PW in a dictionary.
On an average system half of the PW can be guessed

25

in a dictionary.
If user chooses PW in a simple way, it reduces the number of trials needed to guess (the colour of)

Password file access control

for online attacks strategies are delay and preventing to many trials.

Can block offline guessing attacks by denying access to encrypted passwords (Protecting the PW file).

- Make available only to privileged users
- Shadow password file

Vulnerabilities

- Weakness in the OS that allows access to the file
- Accident with permissions making it readable
- Users with same password on other systems
- Access from backup media
- Sniff passwords in network traffic

backups not encrypted

There could be in the way the OS is programmed or an error with the admin with wrong permissions, increasing attack surfaces.

Attempt to use same PW on another system

26 Now split into two: shadow file only stores the hashes, the accessible file has user id, salt etc.

Dual boot can bypass the security schemes.

Because PW file contains many other info required for many other things, so accessed by many pieces of SW that access them.

	qwerty123	password1	password2	pw123	...	
0000	~~~	~~~	~~~	~~	~~	
0001						
0002						
⋮						
1024						

* There is the procedure to change your PW!

Mitigate the problem of vulnerable passwords

Password selection strategies

User education (Training) But it is not enough
Users can be told the importance of using hard to guess passwords and can be provided with guidelines for selecting strong passwords

Computer generated passwords
Users have trouble remembering them (Sometimes happens with banking). This might introduce other vulnerabilities.

Reactive password checking
System periodically runs its own password cracker to find guessable passwords
Takes time. Cannot be done frequently.

Complex password policy
Users is allowed to select his own password, but the system checks if the password is allowable. If not, rejects it
Goal is to eliminate guessable passwords while allowing the user to select a password easy to remember

27

* User may get overwhelmed and user doesn't have an idea of what could be important or not.
Different: before the user has some constraint. Here the check is done immediately.

Proactive password checking

- Rule enforcement
  - Specific rules that passwords must adhere to
- Password checker
  - Compile a large dictionary of "bad" passwords not to use
- But how to make a fast pswd check?
  - Bloom filter, used to implements password checkers
    - Exploits *k* hash functions to build tables of hash values
    - Check desired password against this table
    - It's a probabilistic, efficient way to check if a password is in the dictionary of forbidden passwords

Heuristic, gives very likely answer. If I hash the PW by the user I can't compare the hashes, I will have the same problem.

28

Bloom filter password checker

- Bloom filter constructed over a dictionary D of passwords… say that:
  - $|D| = d$ words
  - $h_1, \dots, h_k$ are hash functions, with $h_i(x) \in [0, n]$
  - Bloom filter B is an array of n bits

Bloom filter constructed as:

```
Let B[i] = 0 for each i ∈ [0,n]
for each x ∈ D:
    for each j ∈ [1,k] : B[h_j(x)] = 1
```

29

Bloom filter password checker

To check a password y with the bloom filter:

```
if B[h_j(y)] = 0 for some j ∈ [1,k] :
    return(valid) // password y is not in D
else
    return(rejected) // password y may be in D
```

The bloom filter does not have false negatives:

- if valid then y not present in D

The bloom filter may have false positives:

- if rejected then y may still not in D

30

Bloom filter password checker

Example: $D = \{Stefano, Paolo\}$

Say that:

$$h_1(Stefano) = 10;\ h_2(Stefano) = 7; h_3(Stefano) = 3$$
$$h_1(Paolo) = 7;\ h_2(Paolo) = 12; h_3(Paolo) = 1$$

Hence $B =$

0	1	2	3	4	5	6	7	8	9	10	11	12
0	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**

Now, consider passwords $Rita$ and $Sofia$, and that:

$$h_1(Rita) = 3;\ h_2(Rita) = 12; h_3(Rita) = 1$$
$$h_1(Sofia) = 2;\ h_2(Sofia) = 9; h_3(Sofia) = 7$$

Then:

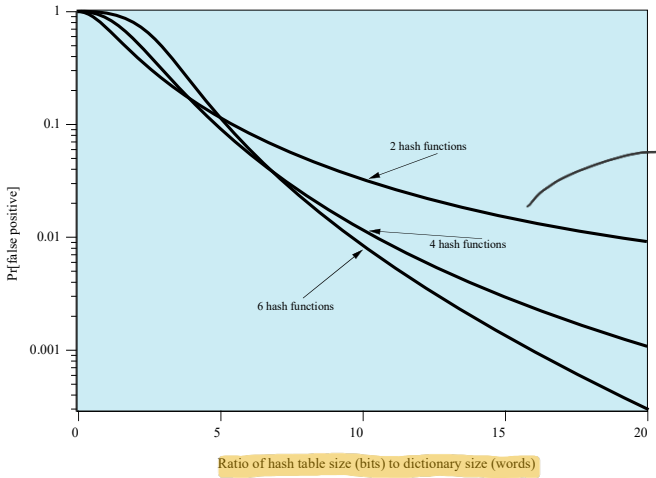
- $Rita$ is rejected
- $Sofia$ is valid

31

Bloom filter password checker

Need to find a proper configuration of the parameters (values of k, m, n), else the mechanism may be unusable

- it may give too many false positives



With less hash functions you have to lower the ratio of hash table size to dictionary size.

Example:
$d = 10^6$
$k = 6$
$n = 10 \cdot 10^6$ (from 0 to 20 · d)
Filter takes about 1.2MBytes

You reduce faster the probability of false positive

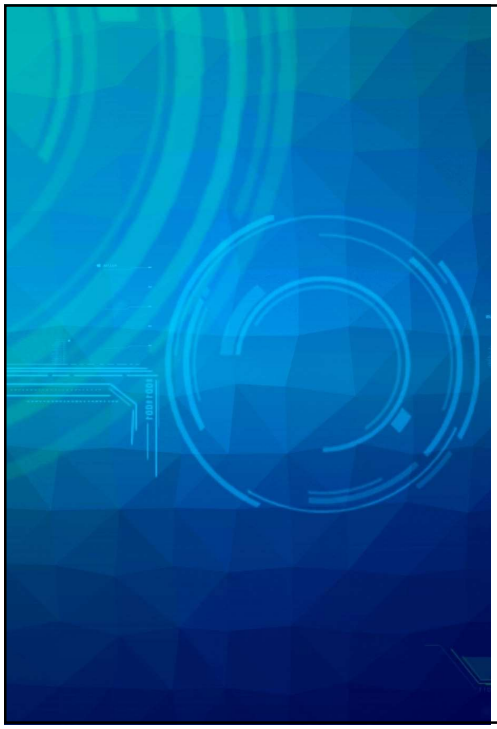
32

curiosity

- Several account (and passwords) breaches, even recently
- Want to know whether your account had been broken?

  https://haveibeenpwned.com/
- contains a DB of many breaches

34

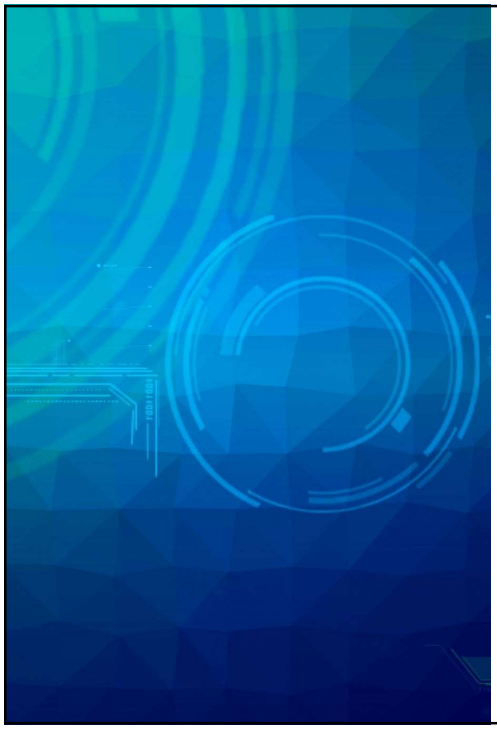


Question

COMMENT ABOUT THE SUITABILITY OF THE PASSWORDS:

a. Back#To#Black#
b. 09876
c. viaFermi3
d. ketchup
e. onafetS
f. S0LI3van7e

35



Question

ASSUME A SYSTEM THAT USE RANDOMLY-GENERATED PASSWORDS. PASSWORDS ARE 8 CHARACTERS LONG IN THE ALPHABET OF THE CAPITAL LETTERS.

WHAT SHOULD BE THE APPROPRIATE RANGE FOR THE PSEUDO-RANDOM NUMBER GENERATOR?

36

Token-based authentication

- Objects that a user possesses for the purpose of user authentication are called tokens:
  - Memory cards
  - Smart tokens/cards
- Applications / Features:
  - Electronic identity cards
  - Eid functions
  - Passwords authenticated connection establishment (PACE)

↳ Must be in posses of the user.

37

Completely analog. Just press and imprint on paper the code of the card and apply signature

Types of cards used as tokens

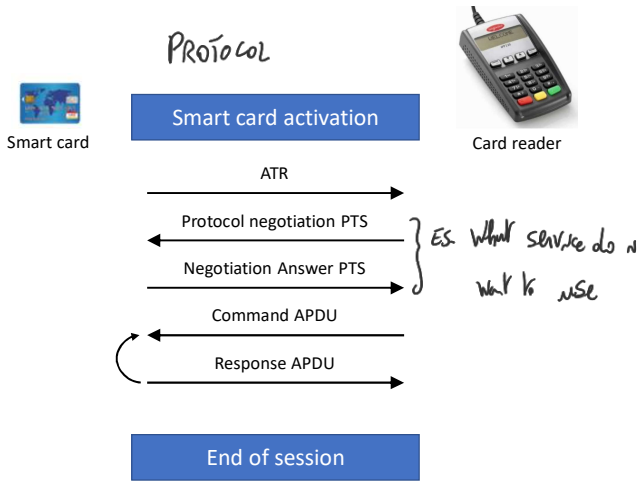
Card Type	Defining Feature	Example
Embossed	Raised characters only, on front	Old credit card
Magnetic stripe	Magnetic bar on back, characters on front	Old bank/telephone card
Memory	Electronic memory inside	Prepaid phone card
Smart tokens Contact Contactless	Electronic memory and processor inside Electrical contacts exposed on surface Radio antenna embedded inside	Biometric ID card

old telephone cards. Have a reader in which you insert a pin (if asked) and you could authenticate

Can implement encryption algorithm

38

Smart cart / reader exchange



ATR: Anwser to reset
PTS: protocol type selection
APDU: Application protocol data unit

42

Electronic identity cards (eID)

Use of a smart card as a national identity card for citizens

Can serve the same purposes as other national ID cards, and similar cards such as a driver's license, for access to government and commercial services

Can provide stronger proof of identity and can be used in a wider variety of applications

In effect, is a smart card that has been verified by the national government as valid and authentic

An example is the German card *neuer Personalausweis*

Has human-readable data printed on its surface

- Personal data
- Document number
- Card access number (**CAN**)
- Machine readable zone (**MRZ**)

43

Basic functionality = user authentication. They can store info and can identificate yourself online or provide the digital signature of the document.

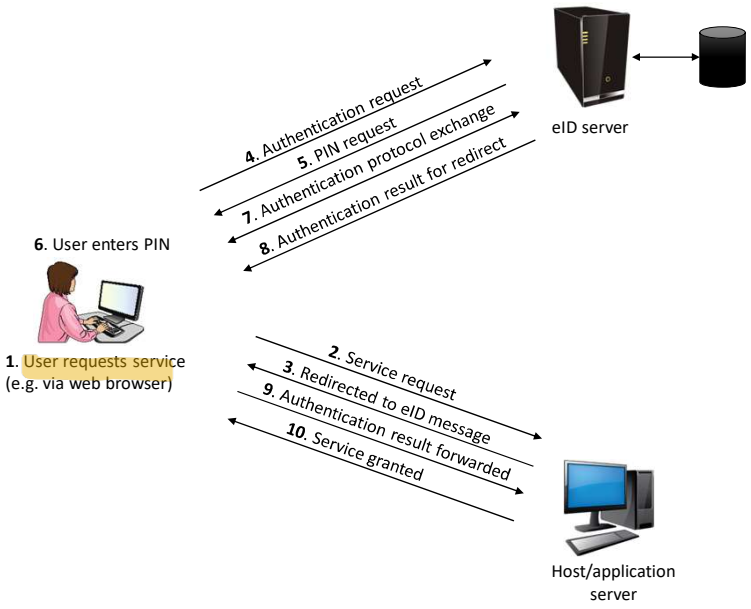
Electronic functions for eID cards

Function	Purpose	PACE Password	Data	Users
ePass (mandatory)	Authorized offline inspection systems read the data.	CAN or MRZ	Face image; two fingerprint images (optional); MRZ data	Offline biometric identity verification reserved for government access
eID (activation optional)	Online applications read the data or access functions as authorized.	eID PIN	Family and given names; artistic name and doctoral degree; date and place of birth; address and community ID; expiration date	Identification; age verification; community ID verification; restricted identification (pseudonym); revocation query Offline inspection systems read the
	Offline inspection systems read the data and update the address and community ID.	CAN or MRZ		
eSign (certificate optional)	A certification authority installs the signature certificate online.	eID PIN	Signature key; X.509 certificate	Electronic signature creation
	Citizens make signature creation electronic signature with eSign PIN.	CAN		

CAN = card access number
MRZ = machine readable zone
PACE = password authenticated connection establishment
PIN = personal identification number

44

User authentication with eID



45

Password Authenticated Connection Establishment (PACE)

- Ensures that the contactless RF chip in the eID card cannot be read without explicit access control
- Online applications: the user enters the 6-digit PIN (which should only be known to the holder of the card)
- For offline applications, either the MRZ printed on the back of the card or the CAN printed on the front is used

46

Biometric authentication

Big difference: can't replace physical characteristics. Passwords, smartcards, you either have it or not. Biometrics are subject to errors.

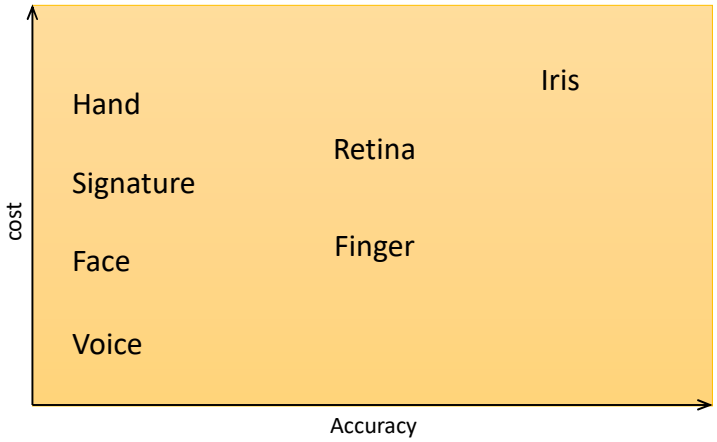
- Attempts to authenticate an individual based on unique physical characteristics
- Based on pattern recognition
- Is technically complex and expensive when compared to passwords and tokens
- Physical characteristics used include:
  - Facial characteristics
  - Fingerprints
  - Hand geometry
  - Retinal pattern
  - Iris
  - Signature
  - Voice

} Can get very expensive.

Problem: we need to store the biometrics. And the comparisons can be imprecise.

47

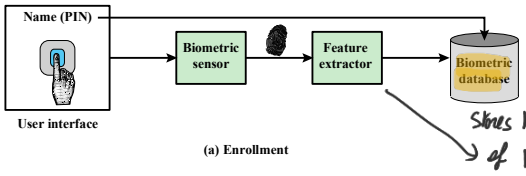
Cost vs accuracy of biometric characteristics in user authentication schemes



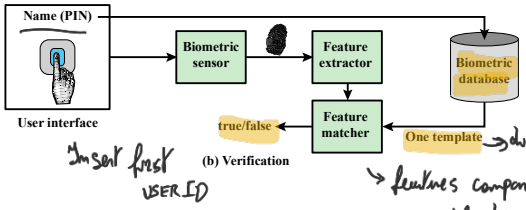
48

A generic biometric system

Associate the credentials to a user



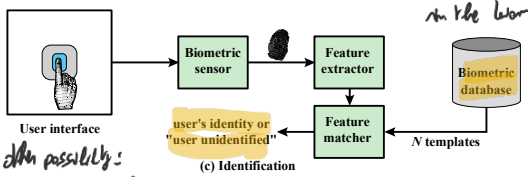
Stores the features of the biometry



different features can be extracted

Insert fresh USER ID

features compared to the template in the biometry



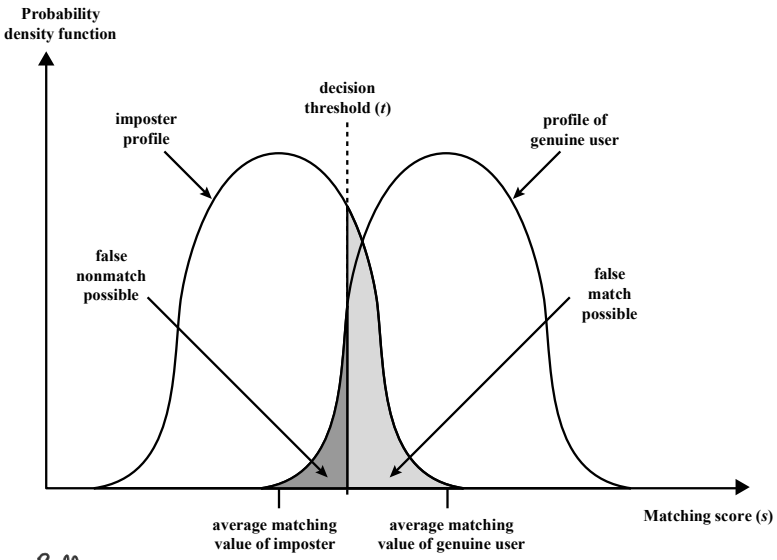
other possibility: use biometry also for identification

=> We have to extract all the templates now.

49

Describe the biometry of the user with a curve

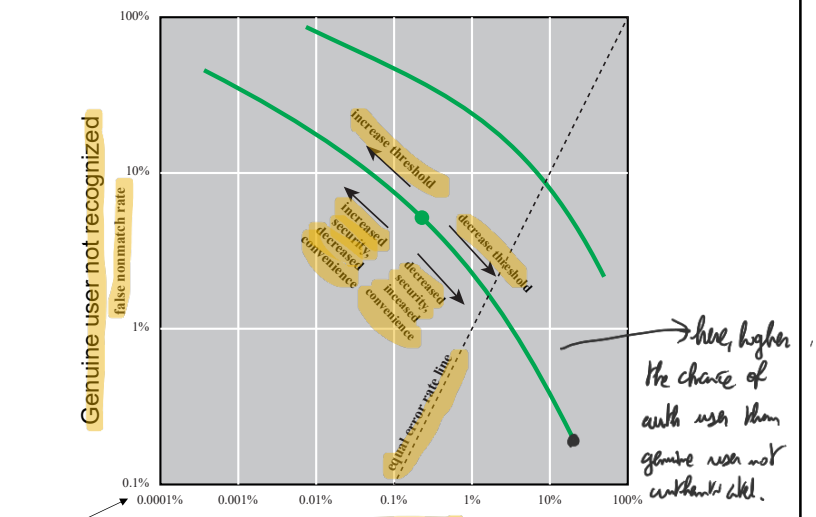
Profiles of biometric characteristics



Problem is when imposter has the same percent of biometry.

50 We need a threshold. If the biometry of another person pass the threshold, we have a false positive.
To reduce the amount of false positives we need to push the threshold further.

Idealized biometric measurement operating characteristic curves



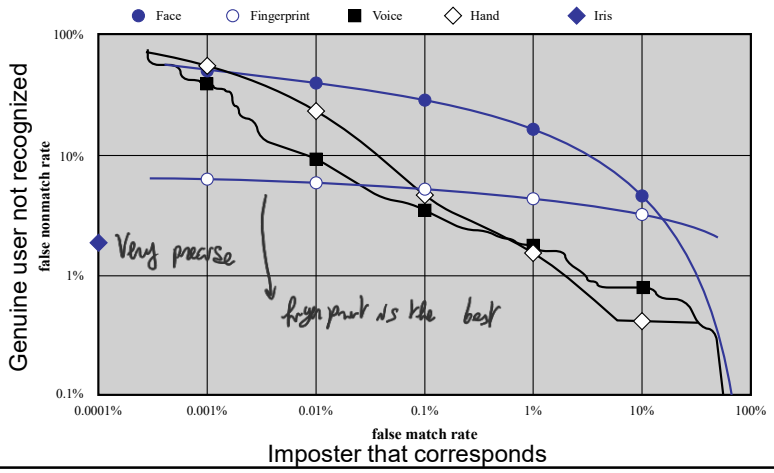
here, higher the chance of auth user than genuine user not authenticated.

We have to find a suitable tradeoff.

51

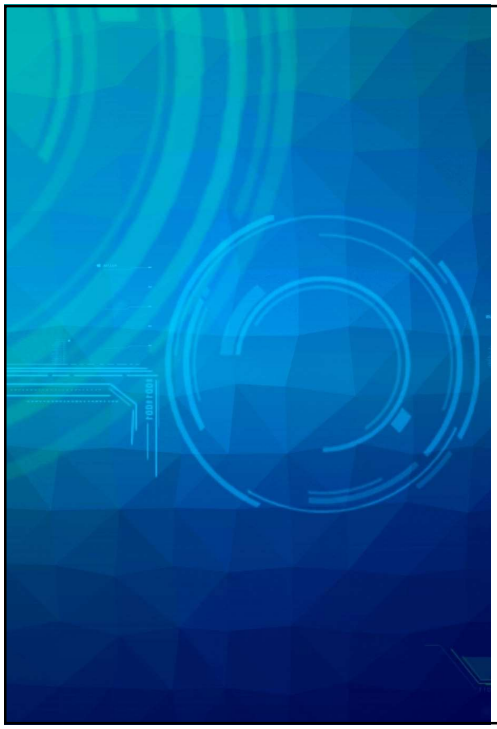
• This one works better than the upper one.

Actual biometric measurement operating characteristic curves (log-scale)



52

If we take fingerprint, we can get a false match in 0.001% with false negatives at 9%.



Review question

"JANE SPLIT A STONE INTO TWO PIECES, KEPT ONE FOR HERSELF, AND GAVE THE OTHER TO JASON. SO SHE SAID - WHEN YOU WILL SEND YOUR EMISSARY GIVE HIM THIS HALF OF THE STONE AND I WILL RECOGNIZE HIM."

WHAT KIND OF AUTHENTICATION IS THIS?

53

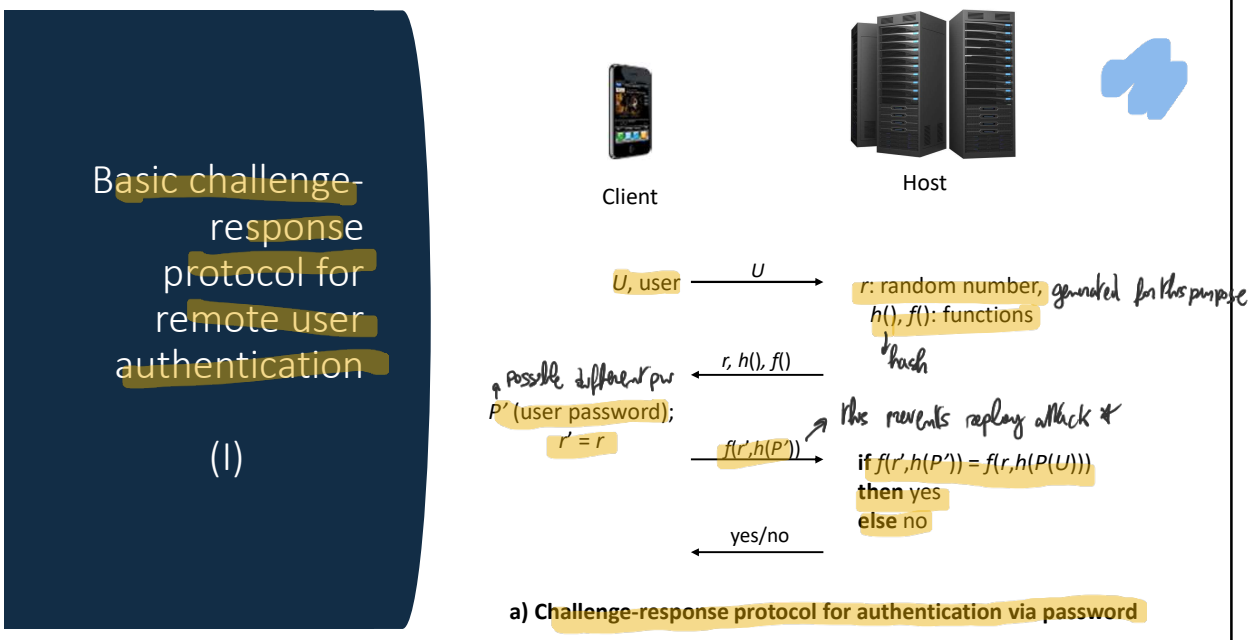
Remote user authentication

- Authentication over a network, the Internet, or a communications link is more complex
- Additional security threats such as:
  - Eavesdropping, capturing a password, replaying an authentication sequence that has been observed
- Generally rely on some form of a challenge-response protocol to counter threats

54

Basic challenge-response protocol for remote user authentication

(I)

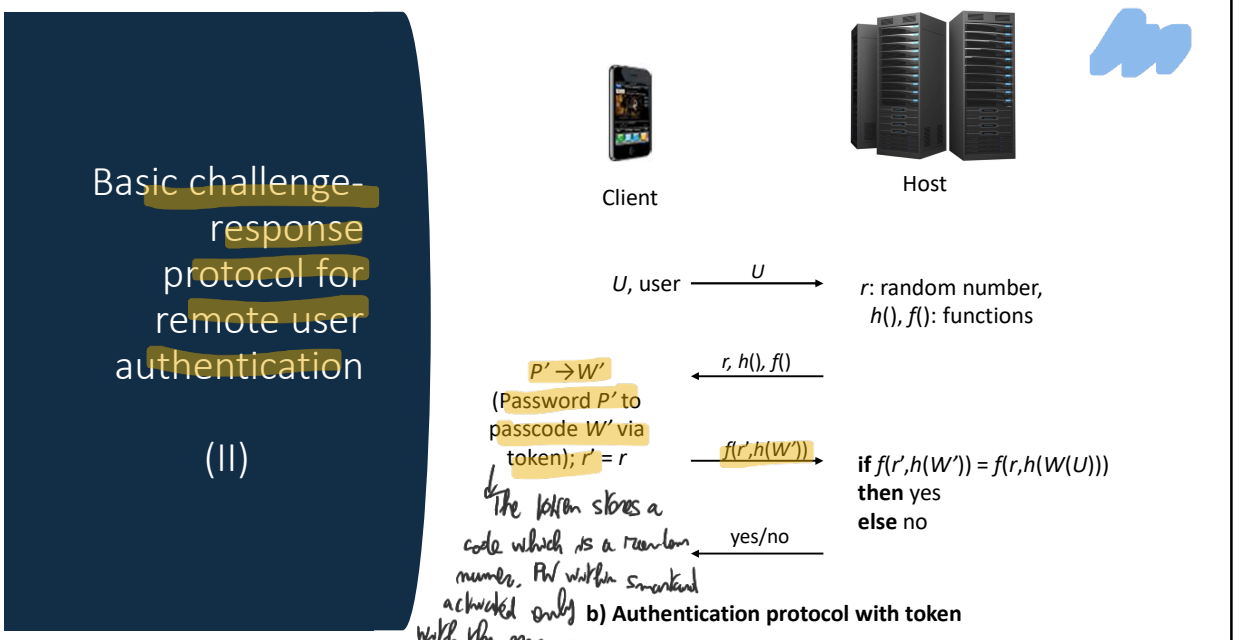


a) Challenge-response protocol for authentication via password

55 *NOTE: f should not be invertible, otherwise attacks can get the hash of the pw. f should be computationally infeasible to invert.

Basic challenge-response protocol for remote user authentication

(II)



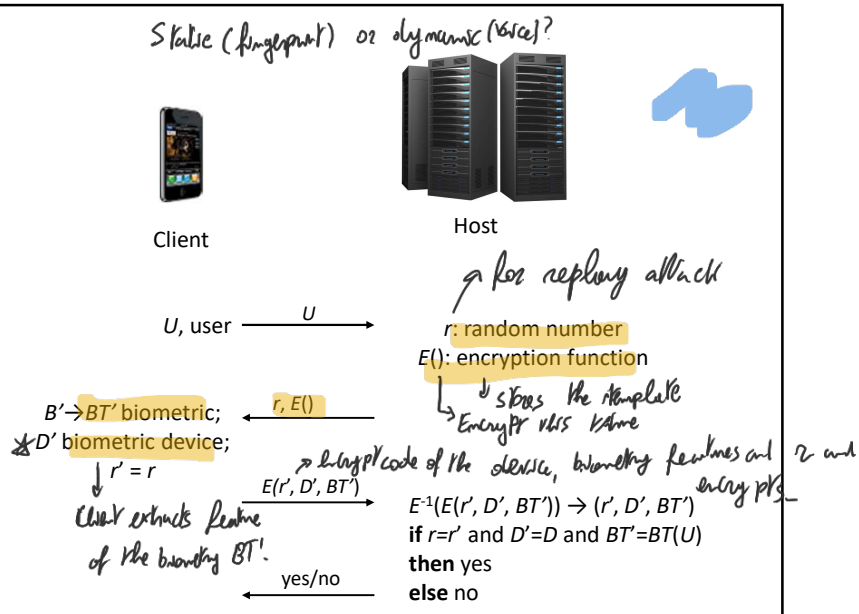
The token stores a code which is a random number. PW within smartcard activated only with the proper pw

b) Authentication protocol with token

56 User uses the pincode to activate the card. If very smart, the card computes f(r',h(w')). Else this computation must be computed by the reader that can be forged. If smart, the opponent might get your pw but still needs your card physically.

Basic challenge-response protocol for remote user authentication

(III)

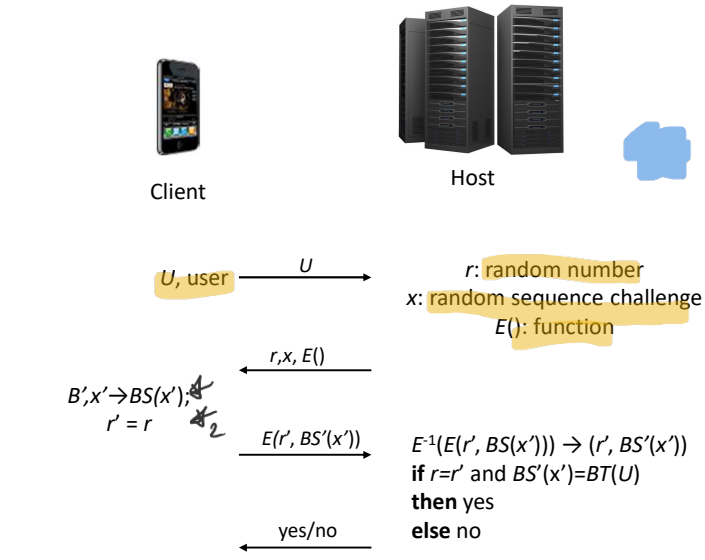


c) Authentication protocol with static biometric

57 We use encryption because the matches might not be exact. If we used hash we will never have perfect matches.
* Yes some cases when might be such noisy different biometrics and you need to know what the device is.

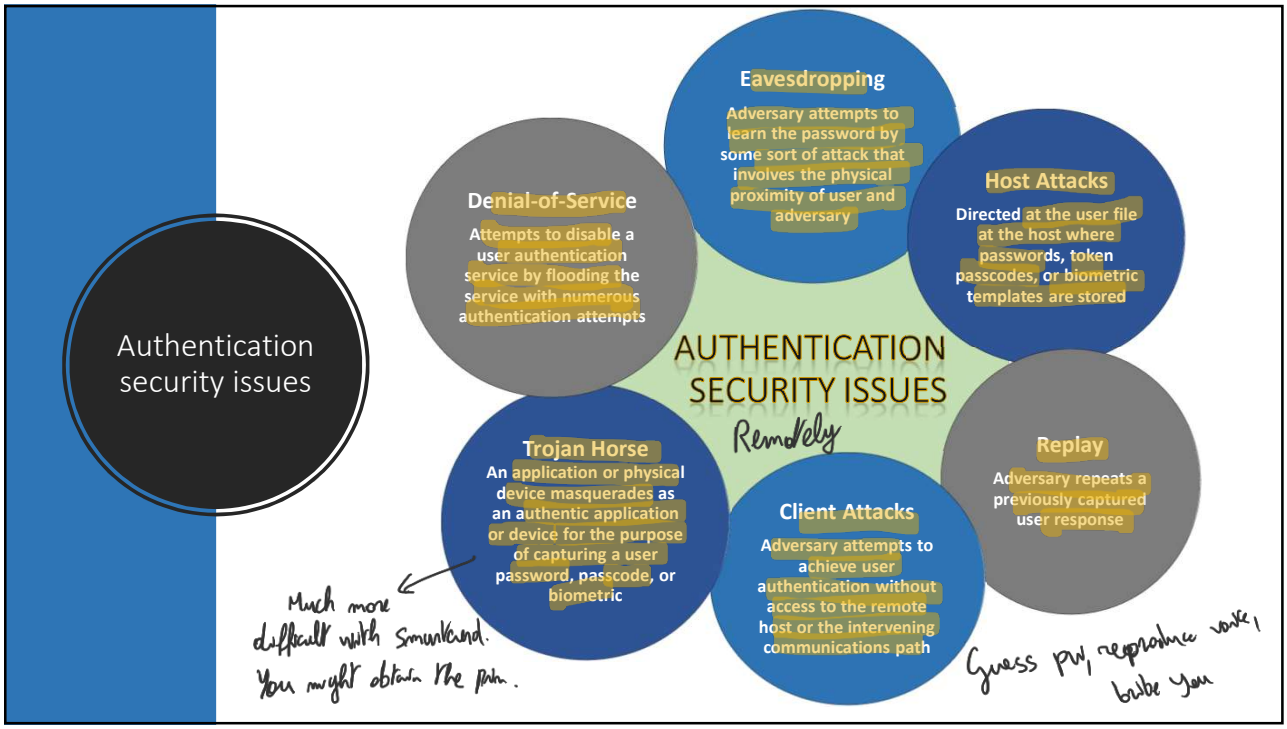
Basic challenge-response protocol for remote user authentication

(IV)



d) Authentication protocol with dynamic biometric

58 * The user is expected to do something: X could be "say the 3 words in sequence". X = the three words.
*2 We produce a biometric signal and sends it to the server. Compares the signal to the template signals stored.



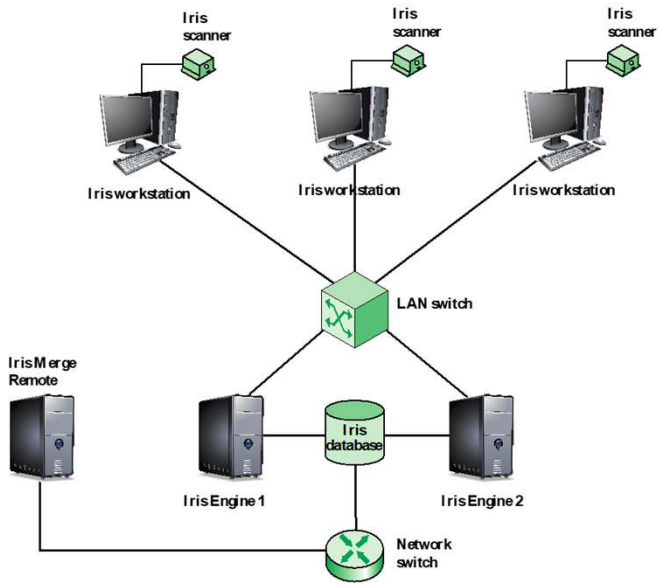
59

Some potential attacks, susceptible authenticators and typical defenses

Attacks	Authenticators	Examples	Typical Defenses
Client attack	Password	Guessing, exhaustive search	Large entropy; limited attempts
"	Token	Exhaustive search	Large entropy; limited attempts; theft of object requires presence
"	Biometric	False match	Large entropy; limited attempts
Host attack	Password	Plaintext theft, dictionary/exhaustive search	Hashing; large entropy; protection of password database
"	Token	Passcode theft	Same as password; 1-time passcode
"	Biometric	Template theft	Capture device authentication; challenge response
Eavesdropping, theft, and copying	Password	"Shoulder surfing"	User diligence to keep secret; administrator diligence to quickly revoke compromised passwords; multifactor authentication
"	Token	Theft, counterfeiting hardware	Multifactor authentication; tamper resistant/evident token
"	Biometric	Copying (spoofing) biometric	Copy detection at capture device and capture device authentication
Replay	Password	Replay stolen password response	Challenge-response protocol
"	Token	Replay stolen passcode response	Challenge-response protocol; 1-time passcode
"	Biometric	Replay stolen biometric template response	Copy detection at capture device and capture device authentication via challenge-response protocol
Trojan horse	Password, token, biometric	Installation of rogue client or capture device	Authentication of client or capture device within trusted security perimeter
Denial of service	Password, token, biometric	Lockout by multiple failed authentications	Multifactor with token

60

General iris scan site architecture for UAE system

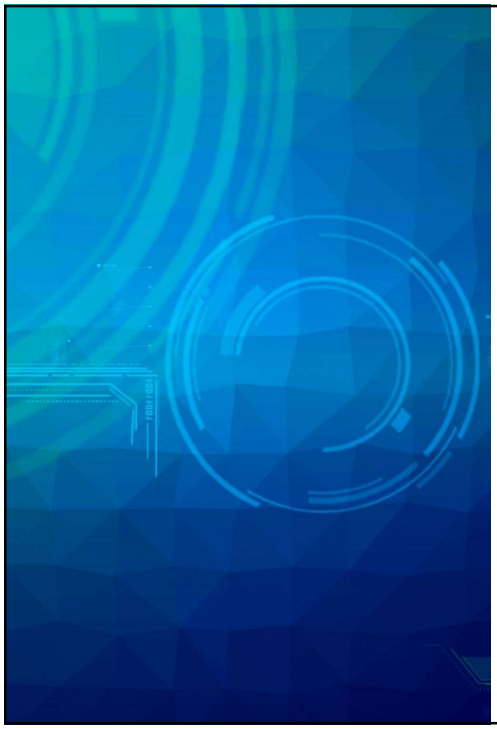


61

Summary

- Digital user authentication principles
  - A model for digital user authentication
  - Means of authentication
  - Risk assessment for user authentication
- Password-based authentication
  - The vulnerability of passwords
  - The use of hashed passwords
  - Password cracking of user-chosen passwords
  - Password file access control
  - Password selection strategies
- Token-based authentication
  - Memory cards
  - Smart cards
  - Electronic identity cards
- Biometric authentication
  - Physical characteristics used in biometric applications
  - Operation of a biometric authentication system
  - Biometric accuracy
- Remote user authentication
  - Password protocol
  - Token protocol
  - Static biometric protocol
  - Dynamic biometric protocol
- Security issues for user authentication

63

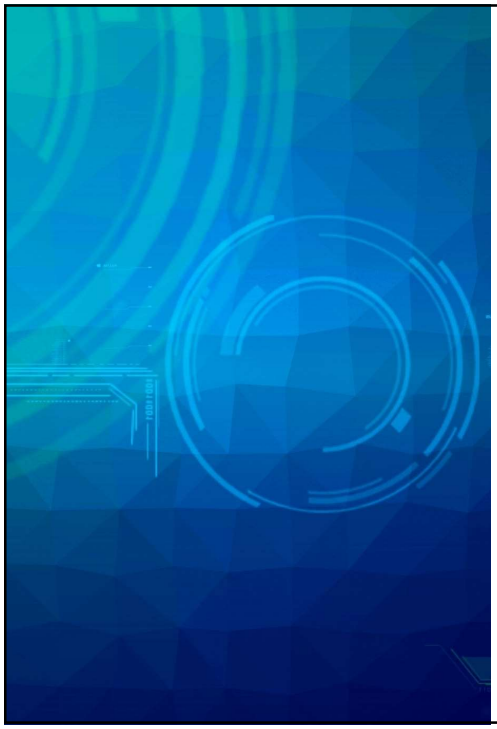


Question

THE SALT IN THE UNIX PASSWORD SCHEME INCREASES THE DIFFICULTY OF GUESSING BY A FACTOR OF 4096.

1. HOW MANY BITS IS THE SALT?
2. THE SALT IS STORED IN PLAINTEXT IN THE SAME ENTRY AS THE CORRESPONDING CIPHERTEXT PASSWORD. THEREFORE, IT IS KNOWN TO THE ATTACKER AND NEED NOT BE GUESSED. WHY IS IT ASSERTED THAT THE SALT INCREASES SECURITY?

64



Question

STILL ABOUT SALT:
WOULDN'T IT BE POSSIBLE TO THWART COMPLETELY ALL PASSWORD CRACKERS BY DRAMATICALLY INCREASING THE SALT SIZE TO, SAY, 24 OR 48 BITS?

65

Exercise 1

A system requests the users to choose passwords at least 8 and at most 10 characters long and that are chosen within an alphabet of 40 symbols. The system combines the passwords with a 10 bits salt to produce a hash code for each password that is stores, along with the salt, in the password file. The hash is 128 bits long.
Assume also that testing a password takes 0.1 milliseconds.

1. An adversary that got access to the password file performs a brute force attack to crack the password of a specific user. How long it will take in the worst case and on average?
2. Assume the system has 10,000 users, and the adversary is interested in breaking the password of one arbitrary user (any user would be OK to get in). How long it will take on average?
   How long it would take if no salt was used?

66

Solution 1.1

- Passwords from 8 to 10 chars
- Alphabet of 40 symbols.
- Salt of 10 bits and hash of 128 bits.
- Testing a password takes 0,1 milliseconds.
- 10,000 users in the system

1) Brute force attack to crack the password of a specific user.

How long it will take in the worst case and on average?

The number of different passwords are: _____

Each password is combined with a salt randomly chosen in _____ combinations

Thus the number of combinations to be generated is: _____

In the worst case it will take: _____

On average it will take: _____

67

Solution 1.1

- Passwords from 8 to 10 chars
- Alphabet of 40 symbols.
- Salt of 10 bits and hash of 128 bits.
- Testing a password takes 0,1 milliseconds.
- 10,000 users in the system

68

Solution 1.2

- Passwords from 8 to 10 chars
- Alphabet of 40 symbols.
- Salt of 10 bits and hash of 128 bits.
- Testing a password takes 0,1 milliseconds.
- 10,000 users in the system

2) breaking the password of an arbitrary user

How long it will take in the worst case and on average?

How long it would take if no salt was used?

The number of different passwords are : _____

Each password is combined with a salt randomly chosen in _____ combinations.

On average _____

Thus it will take on average: _____

If no salt was used it will take on average: _____

69

Solution 1.2

- Passwords from 8 to 10 chars
- Alphabet of 40 symbols.
- Salt of 10 bits and hash of 128 bits.
- Testing a password takes 0,1 milliseconds.
- 10,000 users in the system

70

Exercise 2

A system requests the users to choose passwords at least 8 and at most 10 characters long and that are chosen within an alphabet of 40 symbols. The system combines the passwords with a 10 bits salt to produce a hash code for each password that is stores, along with the salt, in the password file. The hash is **32 bits long**.
Assume also that testing a password takes 0.1 milliseconds.

An adversary that got access to the password file performs a brute force attack to crack the password of a specific user. How long it will take in the worst case and on average?

71

Solution 2

- Passwords from 8 to 10 chars
- Alphabet of 40 symbols.
- Salt of 10 bits and hash of 32 bits.
- Testing a password takes 0,1 milliseconds.

Brute force attack to crack the password of a specific user.

How long it will take on average?

The number of different passwords are: _____

Each password is combined with a salt randomly chosen in _____ combinations

The number of different hashes in which the password is encoded is: _____

On average, the number of combinations to be generated is: _____

On average it will take: _____

72

Solution 2

- Passwords from 8 to 10 chars
- Alphabet of 40 symbols.
- Salt of 10 bits and hash of 32 bits.
- Testing a password takes 0,1 milliseconds.

73