Computer Security & Access Protection

roadmap:

Overview and Terminology

Security requirements

Threats, adversaries and intruders

Attacks from outside the system

Attacks from inside the system

Security holes

Protection mechanisms

Trusted systems

Trust

Security

Protection

?

translation of terms:

Authenticity:	Authentizität
Availability:	Verfügbarkeit
Confidentiality:	Vertraulichkeit
Denial of Service:	Dienstverweigerung
Integrity:	Datenintegrität, Schutz gegen unautorisierte Veränderung
Intruder, Adversary:	Eindringling, Angreifer, Gegner
Privacy:	Datenschutz
Protection:	Schutz
Security:	(Informations-) Sicherheit (Betriebssicherheit= safety)
Security threat:	Bedrohung
Trust:	Vertrauenswürdigkeit

Definitions:

Trust is a property within a social organization with respect to handling information. Trust defines the requirements and the resulting policies defined by an application area concerning the proper usage of information in the temporal and functional domain. It reflects the flow of information in an organization and is specified in terms of rules between authorization of subjects and clearance of information.

Security is the property of an information processing system. Security defines the requirements useful for an owner and user of information to protect it against security threats. Basic requirements which have to be assured in spite of intentional and malicious attacks are the confidentiality, integrity, availability and authenticity of information.

Protection is the set of hardware and software mechanisms to enforce security in a system.

Access Control

Trusted System:

Mandatory access control.

Rules defined by organization policy.

Secure System:

Discretionary, user defined access control.

Rules defined by individual user.

Goal: Flexibility, Expressiveness, Least Privilege.

Protection System:

Mechanisms in the hardware and the

operating system to enforce access specifications.

Security vs. Privacy

Security protects data against misuse by individuals.

Privacy protects individuals against the misuse of data.

Security is a necessary but not a sufficient condition for trust and privacy !

requirements for security

Confidentiality: data should not be read by unauthorized parties.

Integrity: data should not be changed by unauthorized parties.

Availability: data should be accessible when they are needed.

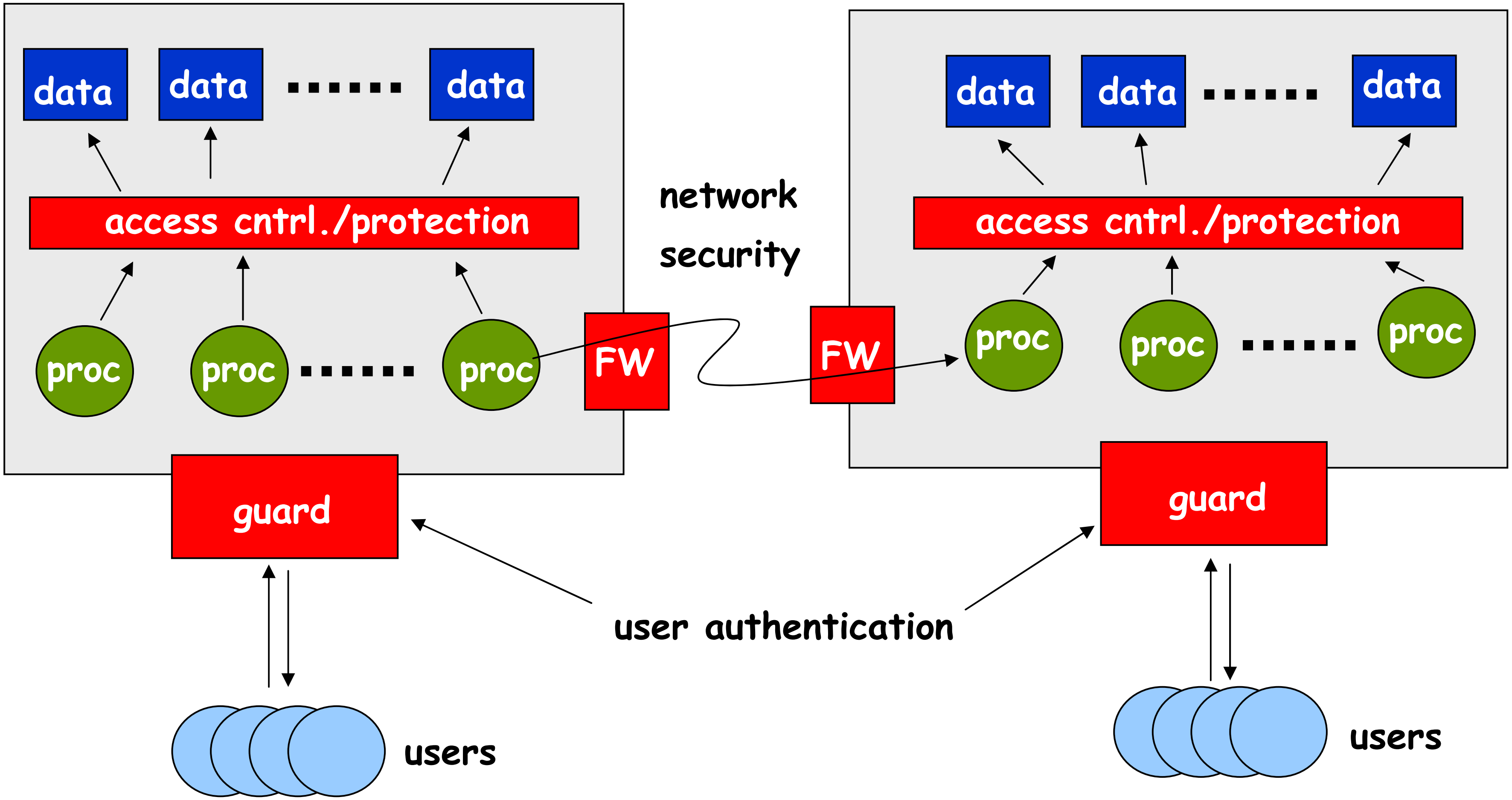
Authenticity: the identity of subjects may not be forged

structuring requirements

acc. DoD Orange Book

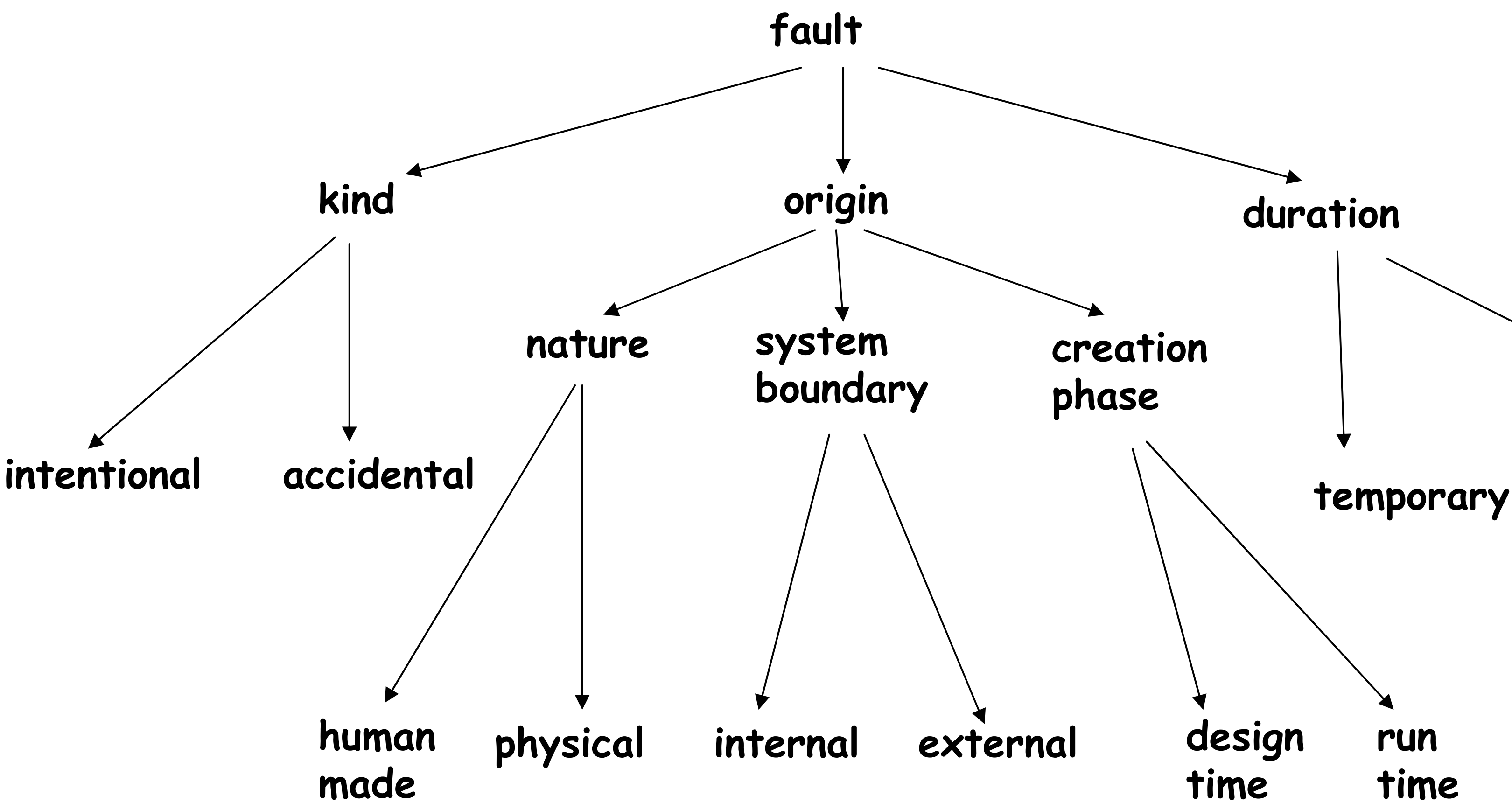
- trusted system
  - policy
  - accountability
  - assurance
    - operational assurance
      - system architecture
      - system integrity
      - hidden channel analysis
      - trusted system administration
      - trusted recovery
    - life cycle assurance
    - documentation

system vulnerabilities



classification of threats

a threat emerges from a fault in some system component or a fault by some user of the system



acc. J.C. Laprie: Dependability: Basic Concepts and associated teminology, 1990

classification of threats

example 1: threats caused by intentional (malicious), human-made faults

system boundary		creation phase		duration		threat
internal	external	desing time	run time	perm.	temp.	
	x		x	x		Intrusion
	x		x		x	Intrusion
x			x	x		Virus
x		x		x		Trojan Horse
x		x		x		malicious logic

classification of threats

example 2: threats caused by accidental faults

	system boundary		creation phase		duration		threat
	internal	external	desing time	run time	perm.	temp.	
physical	x			x	x	x	denial of service
	x			x	x	x	loss of integrity
	x			x	x	x	loss of confidentiality
human made	x		x		x		loss of integrity
	x		x		x		loss of confidentiality

by software or

hardware design faults

classification of adversaries

- occasional non-expert intruders
- expert insiders, unauthorized experienced hackers hacking the system
- expert insiders which have authorized access to the system
- espionage (military and company systems)
- higher forces: Fire, flood, earthquakes
- faults and bugs in the computer and the network
- just humans: e.g. disk with highly confidential data on the garbage etc.

what cryptography can do for security

Confidentiality	encryption of data
Integrity	encryption, digital signatures
Authenticity	encryption of authentication information

Mechanisms:

- one-way functions
- cryptographic hash funtions
- symmetric cryptosystems with a secret key (DES)
- asymmetric cryptosystems with a combination of public/secret key

Def. One-Way-Function

(http://mathworld.wolfram.com/One-WayFunction.html)

Definition: One-Way Function

Informally, a function *f* is a one-way function if

1. The description of *f* is publicly known and does not require any secret information for its operation.
2. Given *x*, it is easy to compute *f*(*x*).
3. Given *y*, in the range of *f*, it is hard to find an *x* such that $f(x) = y$

   More precisely, any efficient algorithm solving a P-problem succeeds in inverting *f* with negligible probability.

The existence of one-way functions is not proven. If true, it would imply $P \neq NP$. Therefore, it would answer the complexity theory NP-problem question of whether all apparently NP-problems are actually P-problems. Yet a number of conjectured one-way functions are routinely used in commerce and industry. For example, it is conjectured, but not proved, that the following are one-way functions:

1. Factoring problem for randomly chosen primes *p*, *q*.
2. Discrete logarithm problem.
3. Discrete root extraction problem. This is the function commonly known as RSA encryption.
4. Subset of sums problem.
5. Quadratic residue problem.

Used e.g. in password encryption, Public Key Cryptography, Digital Signatures, ...

Def. Cryptographic Hash-Function

A ***hash function H*** is a transformation that takes an input m and returns a fixed-size string, which is called the hash value h (that is, $h = H(m)$). Hash functions with just this property have a variety of general computational uses, but when employed in cryptography, the hash functions are usually chosen to have some additional properties.

The basic requirements for a **cryptographic hash function** are as follows.
The input can be of any length.
The output has a fixed length.
$H(x)$ is relatively easy to compute for any given x.
$H(x)$ is one-way.
$H(x)$ is collision-free.

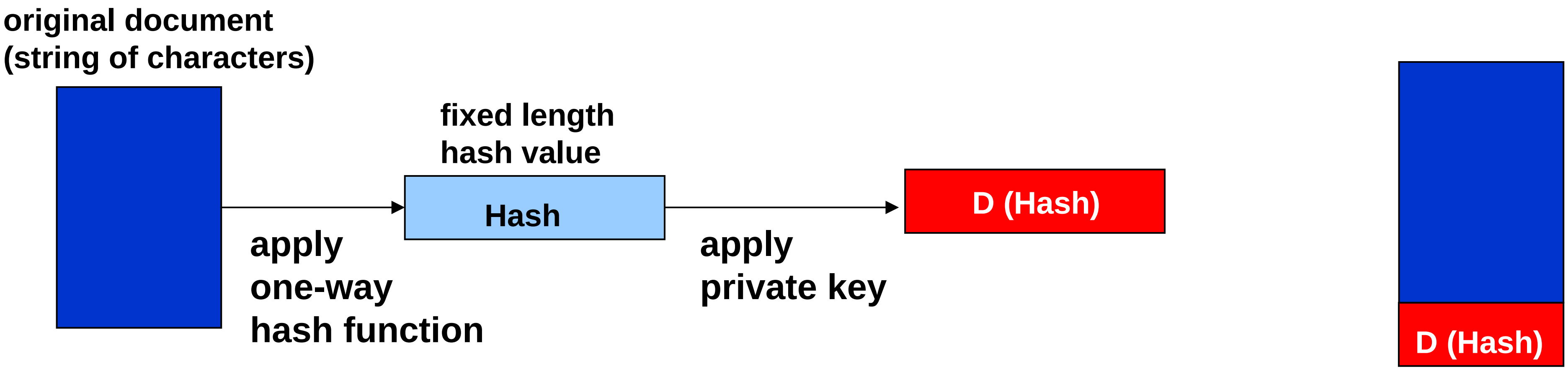
A hash function H is said to be ***one-way*** if it is hard to invert, where ``hard to invert'' means that given a hash value h, it is computationally infeasible to find some input x such that $H(x) = h$.

If, given a string x, it is computationally infeasible to find a string y not equal to x such that $H(x) = H(y)$, then H is said to be a ***weakly collision-free*** hash function.

A ***strongly collision-free*** hash function H is one for which it is computationally infeasible to find any two strings x and y such that $H(x) = H(y)$.

(http://www.rsasecurity.com/rsalabs/node.asp?id=2176)

Example: Digital Signatures



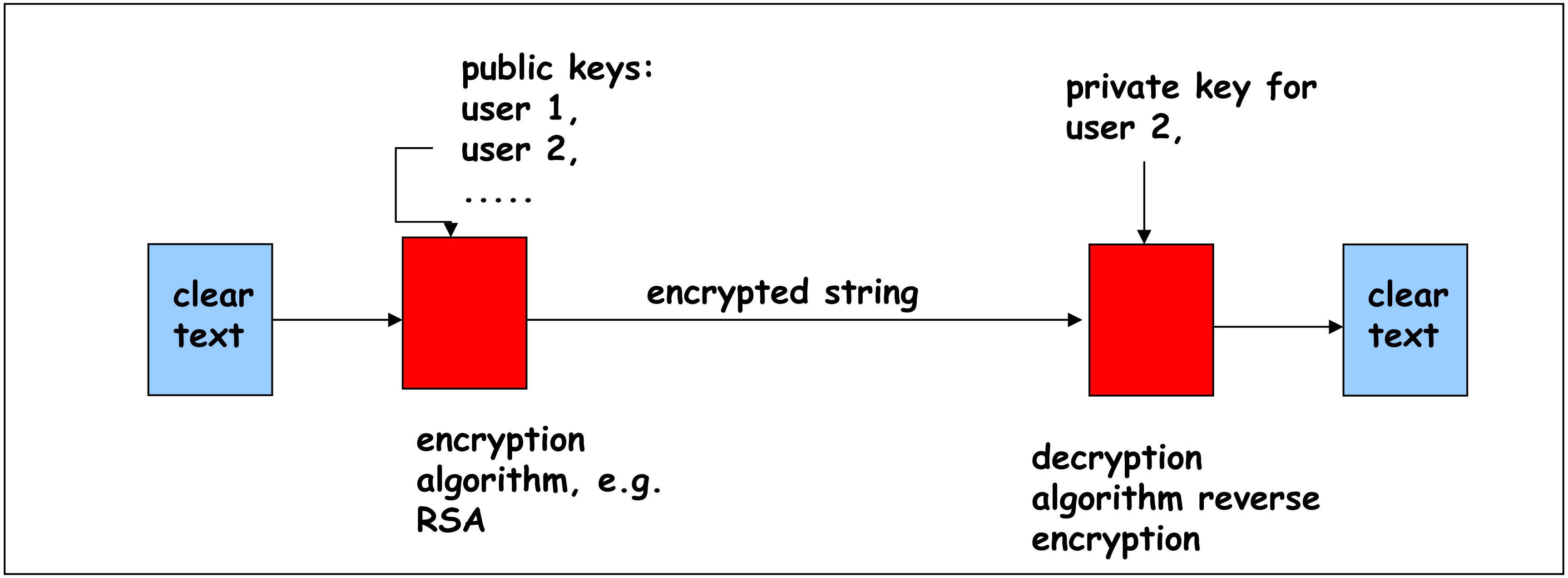
- Receiver calculates the hash value for the document string.
- Receiver applies the public key of the sender E(D(Hash)) to obtain Hash. *
- Then both values are compared and must match.

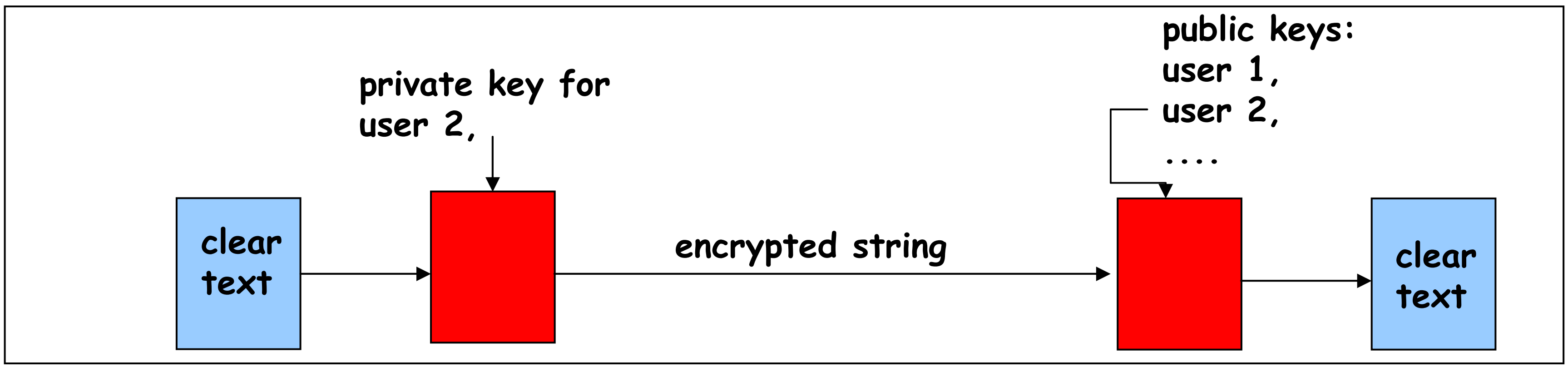
*Note:it is required that E(D(Hash)) = Hash = D(E(Hash)) !!! This is not trie for all encoding functions!

What has to be guaranteed:

1. Integrity of document: this can be checked because the document cannot be changed without changing the hash function ("weakly collision" free property)
2. Authentication of sender: if the document AND the hash value are changed, then applying the public key of the sender to (D(Hash)) will not deliver a correct result.

Public key and Digital Signatures





attacks from outside of the system

The login procedure

```
LBL>telnet elxsi
ELXSI AT LBL
LOGIN: root
PASSWORD:root
INCORRECT PASSWORD, TRY AGAIN
LOGIN: guest
PASSWORD: guest
INCORRECT PASSWORD, TRY AGAIN
LOGIN: uucp
PASSWORD: uucp
WELCOME TO THE ELXSI COMPUTER AT LBL
```

Stoll 89

Tabelle 15.2 Beobachtete Passwortlänge

Länge	Anzahl	Anteil an Gesamtheit
1	55	0,004
2	87	0,006
3	212	0,02
4	449	0,03
5	1.260	0,09
6	3.035	0,22
7	2.917	0,21
8	5.772	0,42
Gesamt	13.787	1,00

aus Stallings: Betriebssysteme, 2003, Pearson Studium

Tabelle 15.3 Passwörter, die aus einer Probe von 13.797 Konten geknackt wurden [KLEI90]

Passwortart	Suchgröße	Anzahl der Treffer	Erratene Passwörter in Prozent
Benutzer-/Kontoname	130	368	2,7%
Zeichenfolge	866	22	0,2%
Zahlen	427	9	0,1%
Chinesisch	392	56	0,4%
Ortsnamen	628	82	0,6%
Gebräuchliche Namen	2,239	548	4,0%
Frauennamen	4,280	161	1,2%
Männernamen	2,866	140	1,0%
Ungewöhnliche Namen	4,955	130	0,9%
Mythen und Legenden	1,246	66	0,5%
Shakespearesch	473	11	0,1%
Sportbegriffe	238	32	0,2%
Science-Fiction	691	59	0,4%
Filme und Schauspieler	99	12	0,1%
Comics	92	9	0,1%
Berühmte Menschen	290	55	0,4%
Redewendungen und Muster	933	253	1,8%
Nachnamen	33	9	0,1%
Biologie	58	1	0,0%
Wörterbuch des Systems	19.683	1,027	7,4%
Rechnernamen	9.018	132	1,0%
Mnemonik	14	2	0,0%
King James-Bibel	7.525	83	0,6%
Verschiedene Wörter	3.212	54	0,4%
Jiddische Wörter	56	0	0,0%
Asteroide	2.407	19	0,1%
GESAMT	62.727	3,340	24,2%

Tabelle 15.3 Passwörter, die aus einer Probe von 13.797 Konten geknackt wurden [KLEI90]

aus Stallings: Betriebssysteme, 2003, Pearson Studium

passwd security

/etc/passwd holds a list of <name, encoded passwd>

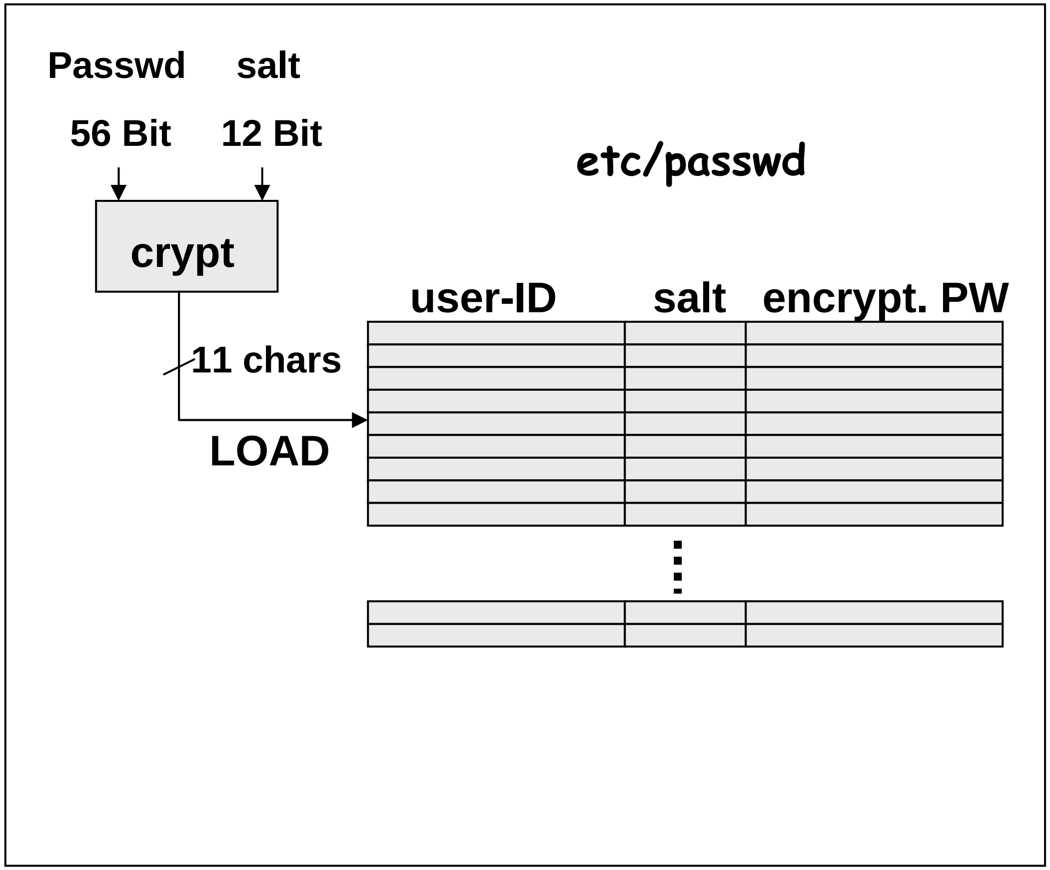
passwd guessing: prepare a list of common passwd, encoded passwd
read the /etc/passwd from some computer
compare encoded passwd
on match > store <name, passwd>

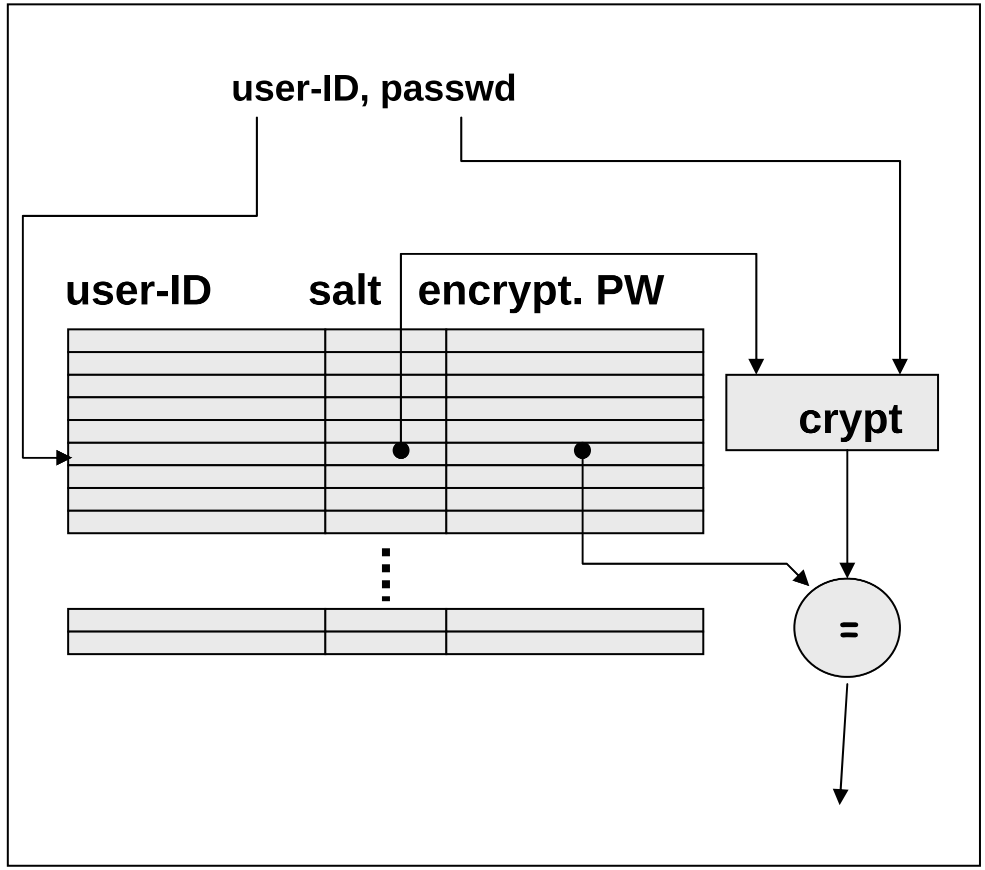
salt: create entries: <name, random number, encoded passwd>
to obtain a match, the cracker has to generate b^n (b=base n=exponent) versions of each passwd.

better passwd: longer names, not in a dictionary, numbers, special characters

one-tme passwd: only used once. (Lamports algrithm to generate the list)

Password mechanisms in Unix





more authentication

challenge-response

chip card + PIN

- magnetic (~ 140 Bytes, costs 0,1 -0,5 €)
- memory cards (~1 KB, ~1 €)
- smart cards (8bit CPU, 16 KB ROM, 4 KB EEPROM, 512 Bytes RAM, 9600 bps communication channel)

biometric authentication

attacks from inside of the system