

9 Rückruf von Capabilities (4)

- ▲ Problem: Rückruf während der Bearbeitung eines Objekts
 - ◆ inkonsistente Zustände möglich
- ★ Lösung in Hydra
 - ◆ Parameter-Capabilities, die durch eine rechteverstärkende Parameterschablone angenommen werden, zeigen auf das Originalobjekt
- ▲ Nachteil
 - ◆ nicht vertrauenswürdige Prozeduren können rückruffreie Capability erlangen
 - ◆ Problem fällt in die selbe Kategorie wie rechteverstärkende Parameterschablonen an sich

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

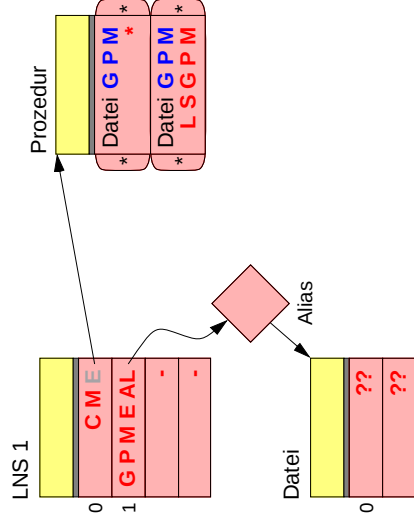
I-Security.doc 1998-02-11 10:55

L.68

Reproduzieren oder Abdruck ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jegliche Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg.

9 Rückruf von Capabilities (5)

Beispiel:



Call(0, 2, 1, (-E), 1, (-E))

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

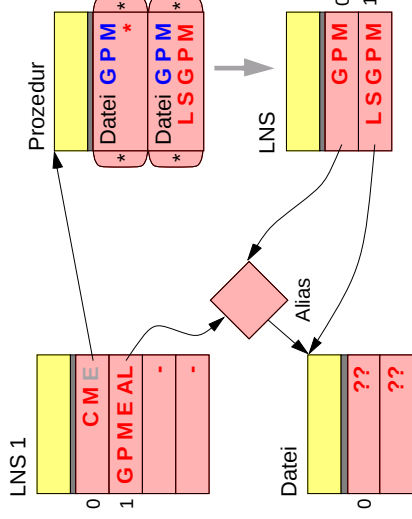
I-Security.doc 1998-02-11 10:55

L.69

Reproduzieren oder Abdruck ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jegliche Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg.

9 Rückruf von Capabilities (6)

Beispiel:



Call(0, 2, 1, (-E), 1, (-E))

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

L.70

Reproduzieren oder Abdruck ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jegliche Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg.

10 Garantierter Zugriff

Schutz vor Rückruf

- ◆ Modifizierende Benutzer eines Objekts
 - kooperierende Benutzer: Rückruf keine Gefahr
 - nicht kooperierende Benutzer: Rückruf nötig
 - ◆ Benutzer eines Objekts ohne modifizierende Zugriffe
 - Aufruf von Prozeduren ist unkritisch, da Aufrufe nicht rückrufbar sind
 - lesende Zugriffe auf Objekte sind kritisch:
 - Verhindern des Rückrufs ist jedoch nicht ausreichend
 - Daten im Objekt könnten gelöscht oder verfälscht werden
 - Capabilities im Objekt oder deren Rechte könnten entfernt werden
- ★ Hydra führt das Einfrierrecht (*Freeze right FRZRTS*) ein
- ◆ Einfrieren nimmt Modifikationsrecht weg
 - ◆ ein Objekt kann nur gefroren werden, wenn alle Capabilities in der C-List bereits das Einfrierrecht haben

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

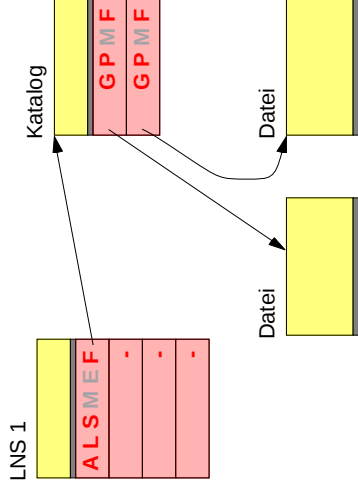
I-Security.doc 1998-02-11 10:55

L.71

Reproduzieren oder Abdruck ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jegliche Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg.

10 Garantierter Zugriff (2)

■ Beispiel:



SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

L72

Reproduktion oder andere Verwendung dieser Vorlesungs- oder Lehrunterlagen ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

11 Bewertung von Hydra

- Hydra demonstrierte die Beherrschbarkeit einer ganzen Reihen von Sicherheitsproblemen
 - ◆ Ergebnisse flossen in eine ganze Reihe von Systemen
 - ◆ reine Capability-basierte Systeme haben sich jedoch nie durchgesetzt
- Hydras Probleme
 - ◆ lagen im wesentlichen nicht am Capability-Mechanismus
 - ◆ es gabe keine vernünftigen Editoren und Compiler
 - ◆ Hardware besaß keine Hardware zur Unterstützung von Paging

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

L73

Reproduktion oder andere Verwendung dieser Vorlesungs- oder Lehrunterlagen ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

I.6 Kryptographische Maßnahmen

- Verschlüsseln und Entschlüsseln vertraulicher Daten
 - ◆ aus den verschlüsselten Daten soll die Originalinformation nur mit Hilfe eines Schlüssels restauriert werden können
 - ◆ Schlüssel bleibt geheim
- Authentisierung
 - ◆ Empfänger kann verifizieren, wer der Absender ist
- Sicherer Kanal
 - ◆ gesendete Informationen können nicht gefälscht und verfälscht werden
 - ◆ nur der adressierte Empfänger kann die Informationen lesen
 - ◆ Empfänger kann den Absender authentisieren

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

L74

Reproduktion oder andere Verwendung dieser Vorlesungs- oder Lehrunterlagen ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

I.6 Kryptographische Maßnahmen (2)

- Funktionen
 - ◆ Verschlüsselungsfunktion E (encrypt): $E(K, T)$
 - ◆ Entschlüsselungsfunktion D (decrypt): $D(K, T)$
 - ◆ K = Schlüssel, T = zu verschlüsselnder Text/Daten
- Verwandte Schlüssel
 - ◆ es gilt: K_1 und K_2 sind verwandt, wenn gilt: $D(K_2, E(K_1, T)) = T$
- Symmetrisches Verschlüsselungsverfahren
 - ◆ es gilt: $K_1 = K_2$

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

L75

Reproduktion oder andere Verwendung dieser Vorlesungs- oder Lehrunterlagen ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

I.6 Kryptographische Maßnahmen (3)

- Forderungen an ein Verschlüsselungsverfahren
 - ◆ Wenn K_2 unbekannt ist, soll es sehr aufwendig sein aus $E(K_1, T)$ das T zu ermitteln (Entschlüsselungsangriff)
 - ◆ Es soll sehr aufwendig sein aus T und $E(K_1, T)$ den Schlüssel K_1 zu ermitteln (Klartextangriff)
 - ◆ Bei asymmetrischen Verfahren soll es sehr aufwendig sein, aus K_1 den Schlüssel K_2 zu ermitteln und umgekehrt.

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

Reproduktion oder andere Verwendung dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

L.76

1 Monoalphabetische Verfahren

- Verfahren nach Caesar

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E

- ◆ Verschlüsselungsfunktion: $E: M \rightarrow (M + k) \bmod 26$
- ◆ k ist variierbar (26 Möglichkeiten)

- Zufällige Substitution

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
F	Q	H	A	J	U	L	G	N	S	P	W	R	O	T	C	V	Y	X	M	Z	K	B	I	D	E

- ◆ 26! Möglichkeiten

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

Reproduktion oder andere Verwendung dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

L.77

1 Monoalphabetische Verfahren (2)

★

Nachteil

- ◆ vollständiges Ausprobieren möglich bei Caesar
- ◆ Häufigkeitsanalyse der Buchstaben
 - für eine Sprache gibt es häufigere Buchstaben, z.B. [e](#) im Deutschen
 - durch die Häufigkeitsanalyse können die Möglichkeiten stark eingeschränkt werden; vollständiges Probieren wird ermöglicht

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

Reproduktion oder andere Verwendung dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

L.78

2 Polyalphabetische Verschlüsselung

- Einsatz von vielen Abbildungen, die durch einen Schlüssel ausgewählt werden

- ◆ Beispiel: Vigenère (Caesar-Verschlüsselung mit zyklisch wiederholten Folgen von Verschiebungswerten)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
...																									
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

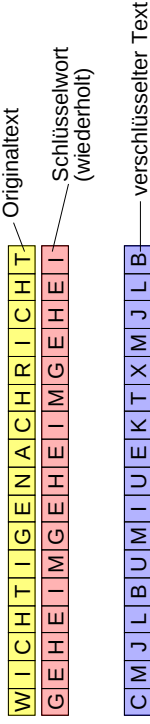
I-Security.doc 1998-02-11 10:55

Reproduktion oder andere Verwendung dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

L.79

2 Polyalphabetische Verschlüsselung (2)

- ◆ Auswahl der Zeile durch den entsprechenden Buchstaben des Schlüsselwortes



- ▲ Gilt als nicht sicher
- ◆ Koinzidenzanalyse
- ◆ Häufigkeitsanalysen und Brute force Attacke

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc: 1998-02-11 10:55

1.80

Reproduzieren oder Abdruck ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Kopieren ist strafbar.

2 Polyalphabetische Verfahren (3)

- Koinzidenz

- ◆ Wahrscheinlichkeit für zwei gleiche Buchstaben untereinander bei umbrechendem Text
 - zufällige Buchstabenwahl: 3,8%
 - englischer Text: 6,6%
- ◆ Brechen polyalphabetischer Verfahren
 - Bestimmen der Koinzidenz für verschiedene Textlängen
 - Textlänge mit höchster Koinzidenz ist wahrscheinlich Schlüsseltextlänge
 - danach Häufigkeitsanalyse pro Buchstabe des Schlüsseltexts

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc: 1998-02-11 10:55

1.81

Reproduzieren oder Abdruck ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Kopieren ist strafbar.

3 One-Time Pad Verfahren

- Theoretisch sicheres Verfahren

- ◆ Liste von Zufallszahlen (so viele wie Zeichen in der Nachricht): $r[i]$
- ◆ Zeichen $z[i]$ der Nachricht wird verschlüsselt mit $c[i] = (z[i] + r[i]) \bmod 26$
- ◆ Empfänger braucht die gleiche Liste

- ◆ theoretisch sicher, da aus dem $c[i]$ nicht auf $z[i]$ geschlossen werden kann

- ▲ Praktisch unbrauchbar

- ◆ echte Zufallszahlen nötig
- ◆ lange Liste nötig
 - jede Liste kann nur einmal verwendet werden
 - Liste muß so lang wie die Nachricht sein

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

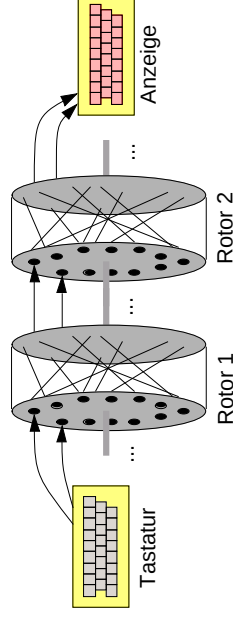
I-Security.doc: 1998-02-11 10:55

1.82

Reproduzieren oder Abdruck ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Kopieren ist strafbar.

4 Rotormaschinen

- Drehende Scheiben verändern ständig die Permutation



- ◆ Einstellen einer Anfangsposition für die Rotoren
- ◆ bei jedem Zeichen wird erster Rotor um eine Position weitergedreht
- ◆ zweiter Rotor rotiert mit niedrigerer Geschwindigkeit
- ◆ zum Entschlüsseln sind entsprechende Gegenstücke nötig

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc: 1998-02-11 10:55

1.83

Reproduzieren oder Abdruck ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Kopieren ist strafbar.

4 Rotormaschinen (2)

- Enigma
 - ◆ deutsche Chiffriermaschine aus dem zweiten Weltkrieg
 - ◆ drei Rotore und Reflektor
 - Reflektor leitet Strom wieder bei einer anderen Position durch die Rotoren zurück: Verfahren wird symmetrisch
 - Entschlüsselern mit den gleichen Rotoren möglich
- Verfahren gilt als nicht sicher
- ◆ Brute force Attacke: *Collosus* Computer
- Schlüsseldemo
- ◆ <http://www.cs.uoregon.edu/~inoble/enigma/apple/>

SP I

Systemprogrammierung I

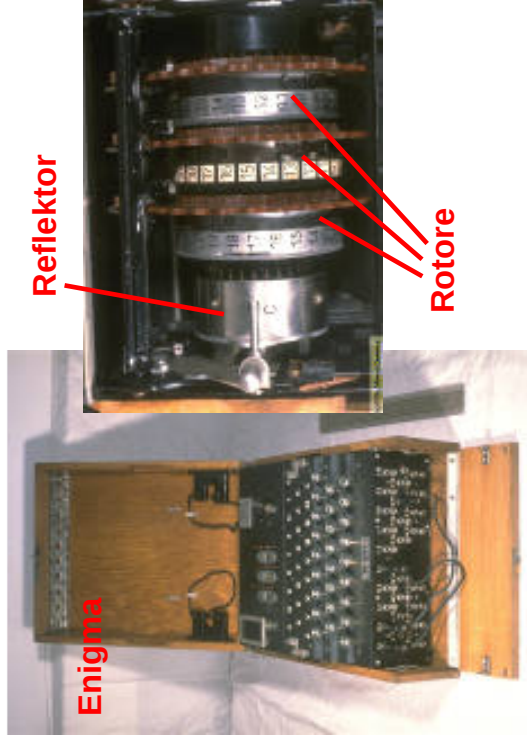
© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

1.84

Reproduzieren oder Abdruck ohne Genehmigung ist strafbar. Dieses Dokument ist urheberrechtlich geschützt und darf nicht weiterverbreitet werden.

4 Rotormaschinen



SP I

Systemprogrammierung I

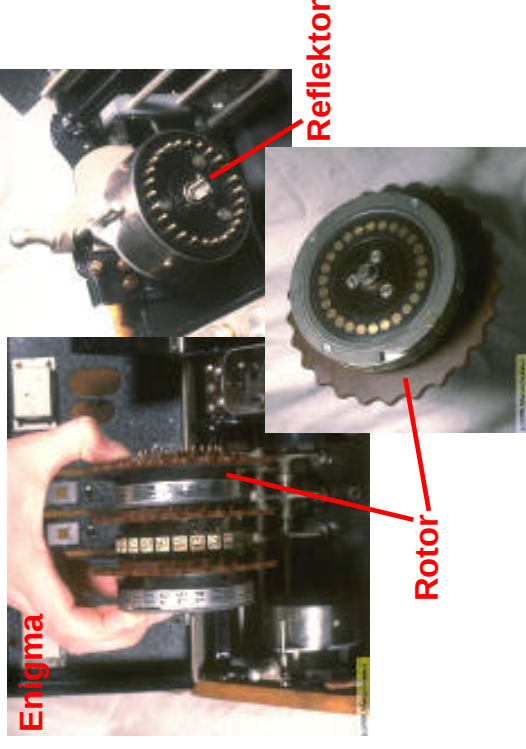
© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

1.85

Reproduzieren oder Abdruck ohne Genehmigung ist strafbar. Dieses Dokument ist urheberrechtlich geschützt und darf nicht weiterverbreitet werden.

4 Rotormaschinen



SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

1.86

Reproduzieren oder Abdruck ohne Genehmigung ist strafbar. Dieses Dokument ist urheberrechtlich geschützt und darf nicht weiterverbreitet werden.

5 Heutige symmetrische Verfahren

- DES (*Data Encryption Standard*, 1977)
 - ◆ entwickelt von IBM
 - ◆ amerikanischer Standard (Kriegswaffe)
 - ◆ blockorientiertes Verfahren (64 Bit Block, 56 Bit Schlüssel)
 - ◆ 16 Runden
 - ◆ trotz vieler Angriffe und trotz des Alters: gilt praktisch als sicher

SP I

Systemprogrammierung I

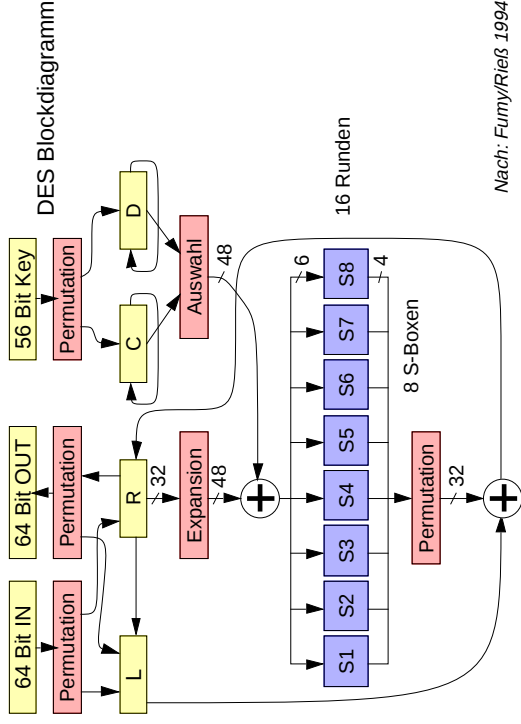
© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

1.87

Reproduzieren oder Abdruck ohne Genehmigung ist strafbar. Dieses Dokument ist urheberrechtlich geschützt und darf nicht weiterverbreitet werden.

5 Heutige symmetrische Verfahren (2)



SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

Reproduktion oder andere Verwendung dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

1.88

5 Heutige symmetrische Verfahren (3)

- Triple DES
- ◆ dreifache Verschlüsselung mit DES
- ◆ Nutzung von drei oder mindestens zwei verschiedenen Schlüsseln
- IDEA (*International Data Encryption Algorithm*)
- ◆ Alternative zu DES
- ◆ 64 Bit Blockgröße
- ◆ 128 Bit Schlüssel
- ◆ keine Permutationen und S-Boxen
- ◆ stattdessen: Addition, Multiplikation und XOR
- ◆ 8 Runden und Output-Transformation
- ◆ Einsatz: z.B. PGP (*Pretty Good Privacy*)

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

Reproduktion oder andere Verwendung dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

1.89

6 Beispiel: UNIX Paßwörter

- Paßwörter wurden zunächst im Klartext gespeichert
- ◆ Paßwortdatei muß streng geschützt werden
- ◆ strenger Schutz oft nicht möglich (z.B. Backup der Platte)
- ◆ Superuser kann die Paßwörter von Benutzern einsehen
- Verschlüsseln der Paßwörter
- ◆ nur die verschlüsselte Version wird gespeichert
- ◆ verschlüsselte Paßwörter dürfen nicht leicht entschlüsselt werden können
- ▲ Ausprobieren von Paßwörtern
- ◆ Benutzer wählen Namen und Gegenstände als Paßwort
- ◆ Verschlüsseln von gängigen Begriffen und Vergleich mit verschlüsselt gespeicherten Paßwörtern
- ◆ Verschlüsselungszeit fließt mit ein in die Sicherheitsbetrachtung

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

Reproduktion oder andere Verwendung dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

1.90

6 Beispiel: UNIX Paßwörter (2)

- Heutiges Verfahren
- ◆ zufällige Auswahl eines von 4096 Werten (*Salt*)
- ◆ der Salt fließt mit in die Verschlüsselung ein, so daß ein und dasselbe Paßwort in 4096 Varianten vorkommen kann
- ◆ Verschlüsselung mit DES
- ◆ Zugriff auf verschlüsselte Paßwörter wird weitestgehend verhindert (Shadow-Paßwortdatei)
- ★ Vorteil
- ◆ Ausprobieren von Paßwörtern benötigt mehr Zeit
- ◆ Vergleich zweier Paßwörter weitestgehend unmöglich

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc 1998-02-11 10:55

Reproduktion oder andere Verwendung dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

1.91

6 Beispiel: UNIX Paßwörter (3)

- Politik am IMMD
 - ◆ Mindestlänge 8 Zeichen
 - ◆ mindestens 5 verschiedene Zeichen
 - ◆ mindestens 3 Zeichenklassen (Groß-, Kleinbuchstaben, Ziffern, Sonderzeichen)
 - ◆ keine Wiederholungen von Zeichenfolgen erlaubt
 - ◆ keine aufeinanderfolgenden Zeichen erlaubt, z.B. "123"
 - ◆ ...
 - ◆ Begriffe, Namen, etc. werden ausgeschlossen und müssen hinreichend verfremdet sein

★ Angriff durch Ausprobieren wird weitestmöglich erschwert

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc: 1998-02-11 10:55

Reproduzieren oder Verwenden dieses Dokuments ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jegliche Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg.

L.92

7 Heutige asymmetrische Verfahren

- RSA (Rivest, Shamir und Adleman)
 - ◆ Öffentlicher Schlüssel (zum Verschlüsseln) besteht aus (e, N)
 - ◆ Ein Block M wird verschlüsselt durch: $C = E((e, N), M) = M^e \bmod N$
 - ◆ C wird entschlüsselt durch: $M = D((d, N), C) = C^d \bmod N$
 - ◆ Wahl der Schlüssel:
 - Es muß gelten $\forall M : (M^e)^d = M \bmod N$
 - Aus Kenntnis von e und N darf d nur mit hohem Aufwand ermittelbar sein
 - ◆ Lösung:
 - $N = pq$ mit p und q zwei hinreichend große Primzahlen
 - zufällige Wahl von d , teilerfremd zu $(p-1)(q-1)$
 - Berechnung von e aus der Bedingung: $ed = 1 \bmod ((p-1)(q-1))$
- ◆ Es ist aufwendig die Primfaktoren von N zu berechnen (mit diesen wäre es möglich d zu ermitteln)

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc: 1998-02-11 10:55

Reproduzieren oder Verwenden dieses Dokuments ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jegliche Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg.

L.93

7 Heutige asymmetrische Verfahren (2)

- Vorteil asymmetrischer Verfahren (*Public key*-Verfahren)
 - ◆ nur ein Schlüsselpaar pro Teilnehmer nötig (sonst ein Schlüsselpaar pro Kommunikationskanal)
 - ◆ Schlüsselverwaltung erheblich vereinfacht
 - jeder Teilnehmer erzeugt sein Schlüsselpaar und
 - veröffentlicht seinen öffentlichen Schlüssel
 - ◆ Authentisierung durch digitale Unterschriften möglich
 - ◆ gilt als sicher bei hinreichend großer Schlüssellänge (1024 Bit)
- Nachteil
 - ◆ relativ langsam berechenbar
 - ◆ gemischter Betrieb von asymmetrischen und symmetrischen Verfahren zur Geschwindigkeitssteigerung

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc: 1998-02-11 10:55

Reproduzieren oder Verwenden dieses Dokuments ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jegliche Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg.

L.94

7 Heutige asymmetrische Verfahren (3)

- Beispiel: PGP Verschlüsselung

Originaltext

IDEA

öffentlicher RSA Schlüssel des Empfängers

indiv. IDEA Schlüssel

verschlüsselte Nachricht
- ◆ Daten werden mit einem individuellen Schlüssel IDEA-verschlüsselt
- ◆ IDEA-Schlüssel wird RSA-verschlüsselt der Nachricht angehängt
- ★ Nachricht an mehrere Adressaten verschickbar
- ◆ lediglich der IDEA-Schlüssel muß in mehreren Varianten verschickt werden (je eine Version verschlüsselt mit dem öffentl. Schlüssel des Empfängers)

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

I-Security.doc: 1998-02-11 10:55

Reproduzieren oder Verwenden dieses Dokuments ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jegliche Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg.

L.95