J ∫ Skripte Mo 18. März

Kryptologie ICS.KRYPTO

Folien zur Präsenz 5

«Kryptoanalyse mit Schwerpunkt symmetrische Kryptographie»

FS 24, V2.2

Das Kap. 1.3 in [CP-D] ist vorbereitend durchzulesen!

Graphik aus https://www.kryptowissen.de/

©Josef Schuler, dipl. math., dipl. Ing. NDS ETHZ, MSc Applied IT-Security, Feldhof 25, 6300 Zug, j.schuler@bluewin.ch resp. josef.schuler@hslu.ch

Inhaltsübersicht und Ausblick auf Präsenz 6

- Eine Einführung in die klassische Kryptoanalyse mit Schwerpunkt symmetrische Verfahren.
  - **Bitte lesen das Kap. 1.3 in [CP-D] vorbereitend durch.**
  - Wir besprechen das Kap. 10 und verweisen zusätzlich auf das Kap. 25.4.5 „Die affine Chiffre“ im JS Skript „Einführung in die Kryptologie“.
- Eine Zwischenbetrachtung zum Teil 1 «Symmetrische Kryptographie».
  - Wir besprechen, was Sie bis dato erreicht haben sollten, und wie Sie das anhand den Aufgaben in [CP-D] resp. im JS Skript „Aufgaben und Lösungen zum Modul KRYPT an der HSLU-I“ und der Standortbestimmung 1 überprüfen können.
- Ausblick auf Präsenz 6 «Math. Grundlagen der asym. Krypto.».
  - **Wichtig: In den Folien zu Präsenz 6 sind die Kapitel 1 – 5 Repetitionen aus D-MATH. Die Kapitel 1 – 4 sind vorgängig zu repetieren und werden in der Präsenz *nicht mehr besprochen!!* Der bekannte Stoff in Kap. 5 und der neue Stoff in Kap. 6 werden wir in der Präsenz behandeln! Als Nachschlagewerk ist das Skript aus D-MATH in Ilias auch hochgeladen.**

Verweise zur Literatur

- ***Für die Theorie:***
  - JS Skript „Einführung in die Kryptologie“, Kap. 10 & 25.4.5.
  - JS Skript, Präsenzskript in D-MATH SW 9 – 11, „Zahlentheorie I – III“.
  - In [CP-D] die Kap. 1.3, 1.4.4, 5.2 & 5.3.
- ***Aufgaben***
  - Aufgaben in den Folien & JS Skript „Einf. in die Kryptologie“, Kap. 10 & 25.4.5.
  - JS Skript „Aufgaben und Lösungen zum Modul KRYPT an der HSLU-I“, Kap. 2.1, „Aufgaben zu den Präsenzen 0 – 5“.
- ***Bemerkungen:***
  - Die Kapitelnummerierung (Kap. 10.x) in den folgenden Folien entspricht derjenigen im oben erwähnten JS Skript „Einführung in die Kryptologie“. D.h. die Details zu den Folien können im Skript nachgelesen werden. Zudem hat es im Skript weitere Übungen und Beispiele.
  - ***Die Aufgabennummerierung im JS Skript «Einführung in die Kryptologie» und in den vorliegenden Folien stimmen nicht überein!***
  - ***Wichtig:*** Bitte beachten Sie, dass nur das Bearbeiten und Lernen der Folien nicht genügt. Das Durcharbeiten des oben erwähnten Kapitels in JS Skripte „Einführung in die Kryptologie“ ist absolut zentral zum Bestehen der MEP.

Lernziele

- Ich erhalte einen ersten Einblick in die Kryptoanalyse.
- Ich kenne die affine Chiffre und kann die Grösse des Schlüsselraumes berechnen.
- Ich kenne die Definitionen der Attacken.
- Ich kann die verschiedenen Grundtypen der Kryptoanalyse unterscheiden.
- Ich kann den Nutzen einer Brute-Force Attacke abschätzen.
- Ich kann den Aufwand einer Brute-Force Attacke berechnen.
- Ich kann das Prinzip eines Time-Memory Tradeoff Angriffs erklären.
- Ich kann den 2-DES analysieren und kenne die wahre Schlüsselstärke.
- Ich kann einen Angriff auf eine Stromchiffre analysieren, resp. erklären, wieso die Stromchiffre keine Integrität gewährt.

Der Slogan zu dieser Präsenz

Eine Chiffre ist sicher $\overset{\Rightarrow}{\not\Leftarrow}$ ***Brute Force ist unmöglich, d.h. unmögliche Brute Force ist notwendig, aber nicht hinreichend!***

Kap. 10

KLASSISCHE ANALYSE VON BLOCKCHIFFREN

Wir haben schon Kryptoanalyse betrachtet!

- Wir haben schon an einigen Stellen in diesem Skript kryptoanalytische Betrachtungen durchgeführt, so z.B.
  - … Kap. 2.5.3, Beispiel 2.5, eine Side Channel Attacke.
  - … bei der Definition der Attacken und Angreifer (cf. Kap. 4 im Skript).
  - … bei der Definition der Entropie (cf. Kap. 5.1 im Skript).
  - … bei der Stärke eines Passwortes (cf. Kap. 5.2 im Skript).
  - … mit der Bemerkung, dass Blockchiffren deutlich resistenter gegen Angriffe von Quantencomputer sind, als asymmetrische Verfahren (cf. Kap. 7.1.1 im Skript).
  - … beim Feststellen, dass Stromchiffren keine Integrität gewähren (cf. Kap. 7.2.3 im Skript).
  - … beim Lösen des Stromchiffre-Puzzles.
  - … bei der Betrachtung der Eigenschaften der Modi der Blockchiffren.
  - … bei der Betrachtung der Modi der Blockchiffren; was passiert beim Entschlüsseln, wenn man im Chiffretext ein Bit verändert. (cf. Kap. 8.1.6 & 8.3.5 im Skript).
  - … beim Lösen des Blockchiffre-Puzzles.
  - … bei der Betrachtung der zwei verschiedenen Kollisionsresistenzen der Hashfunktionen (cf. Kap. 14.2 im Skript).
  - … usw.

Einführung in die Kryptoanalyse in [CP-D]

In [CP-D] wird die Kryptoanalyse mit dem folgenden Diagramm eingeführt. Dabei wird auch das (nicht kryptographische) Social Engineering erwähnt.

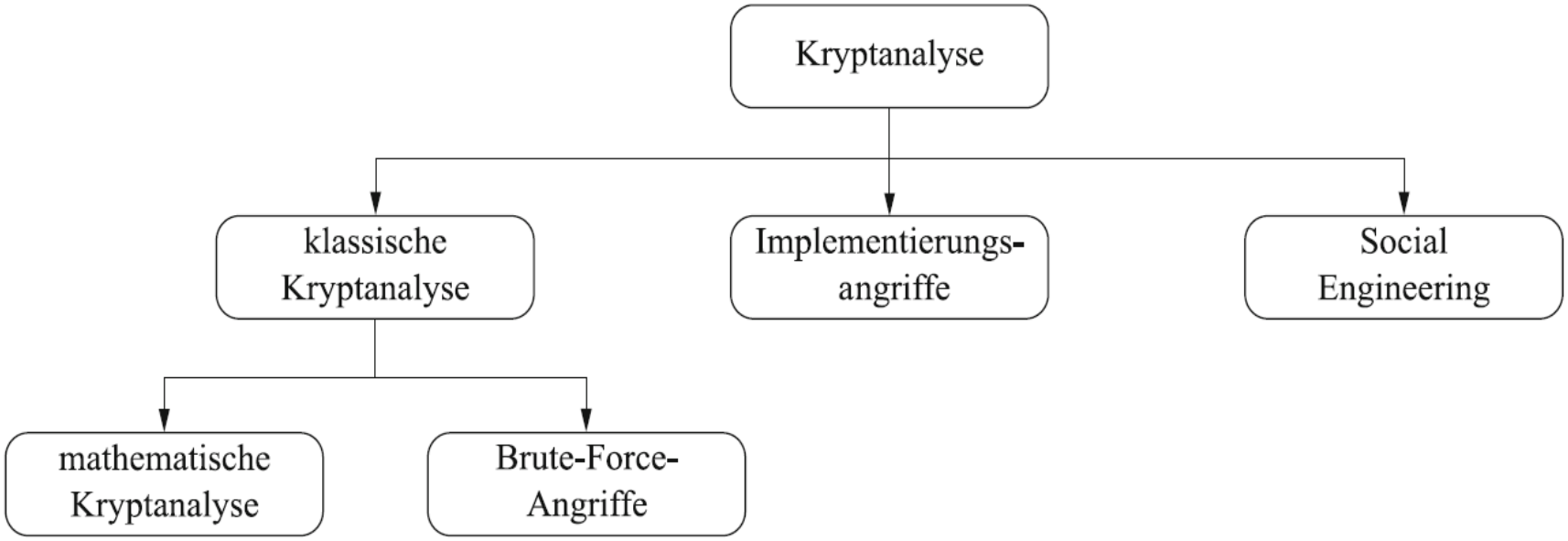


Abb. 1.6 Verschiedene Methoden der Kryptanalyse

und 1.2

Aufgabe

Sollten Sie in der Vorbereitung noch nicht dazugekommen sein, das Kap. 1.3 in [CP-D] zu lesen, dann machen Sie das bitte in der Nachbearbeitung.

Kap. 10.1

Einleitung oder was heisst Sicherheit?

Was heisst Sicherheit? → die 2 Fragen

1) Was und wie viel weiss ein Gegner, d.h. welche **Informationen** stehen ihm zur Verfügung?
2) Wie viele **Computerressourcen** stehen ihm zur Verfügung?

<u>Antwort zur Frage 1)</u>

- Es gilt das ***<u>Prinzip von Kerckhoff:</u>*** „Von einem Kryptosystem ist alles ausser der Schlüssel bekannt“, formuliert A.K. 1883.
- Punktuell geben wir dem Angreifer noch mehr in die Hand, siehe dazu die nachfolgenden Definitionen der Attacken.

- Auguste Kerckhoff, 1835 – 1903

Was heisst Sicherheit? → die Antwort zu 2)

Akademische Antwort zu Frage 2): Die Computerressourcen sind:

- **unbeschränkt:** Es wird keine Annahme gemacht, dass der Gegner rechenmässig eingeschränkt ist (d.h. der Gegner hat unbeschränkte Ressourcen zur Verfügung). Ein System, das unter dieser Annahme sicher ist, heisst ***informationstheoretisch*** sicher.
  - ***Beispiel:*** Cäsar Verschlüsselung eines einzigen Buchstabens. Warum inf.theo.si.?

- **beschränkt:** Es wird eine Annahme über die verfügbare Rechenleistung gemacht. (d.h. der Gegner hat ev. sehr hohe, aber doch beschränkte Ressourcen zur Verfügung). Ein System, welches unter einer solchen Annahme sicher ist, heisst ***berechenmässig*** sicher.

Unakademische Antwort zu Frage 2):

- Wir geben dem Gegner „unverschämte“ Computerressourcen zur Verfügung, wie …
  - … es steht die ganze Sonnenenergie, die zur Erde geschickt wird zur Verfügung.
  - … die ganze Erde kann als Speichermedium (Harddisk) verwendet werden.

Kap. 10.2

Definitionen und Attacken

Die 3 wichtigsten Attacken der Verschlüsselung

I) Ciphertext-only Attacke:

- Der Kryptoanalytiker hat nur die Möglichkeit, verschlüsselten Text für seine Kryptoanalyse zu verwenden, d.h. der Analytiker hat Zugriff auf den unsicheren Übertragungskanal.
- ***Bekannt:*** Nur der Output (= Ciphertext).
- ***Ziel:*** Kenntnis des Input (= Klartext) und/oder des Schlüssels.

II) Known-plaintext Attacke:

- Dem Kryptoanalytiker stehen Klartext/Chiffretext-Paare (M_i, C_i) zur Verfügung, hierbei wird z.B. ausgenutzt, dass bestimmte Anfangs- oder Endphrasen verschlüsselt werden.
  - Einfache Beispiele hierfür sind:
    - ... Sehr geehrte Damen und Herren, ...
    - ... Freundliche Grüsse ...
- ***Bekannt:*** Teile des Inputs und des ganzen Outputs.
- ***Ziel:*** Kenntnis des (restlichen) Klartextes und/oder des Keys.

Die 3 wichtigsten Attacken

III) Chosen-plaintext Attacke:

- Der Kryptoanalytiker ist in der Lage beliebigen Klartext zu verschlüsseln, d.h. er hat Zugriff auf das verwendete Verschlüsselungsgerät.
- ***Zugriff:*** Der Angreifer hat Zugriff auf ein Kryptomodul mit dem geheimen Schlüssel.
- ***Bekannt:*** Ein frei wählbarer Input und den entsprech. Output.
- ***Ziel:*** Kenntnis des Schlüssels.

Bemerkung: Es gibt viele weitere Attacken, z.B.:

- Adaptive-chosen-plaintext Attacke
- Ciphertext-ciphertext-Attacke
- Chosen Ciphertext Attacke (für Public Key Algorithmen)
- Chosen-key Attacke
- Known & Chosen Preimage Attacken gegen MAC Algorithmen

Die 3 wichtigsten Attacken, Aufgaben

Aufgabe 1 Welcher Angriff wird hier beschrieben?

- **Angriff 1:** Der Angriff beruht darauf, dass ausgenutzt wird, dass gewisse Teile der unverschlüsselten Dokumente immer gleich lauten und dass der immer gleich lautende Teil bekannt ist.

  _____

- **Angriff 2:** Der Gegner besitzt ein Verschlüsselungsgerät und will den darin enthaltenen unbekannten Schlüssel eruieren. Er versucht das, indem er selber bestimmte Klartexte hineinschickt.

  _____

- **Angriff 3:** Der Gegner hat nur verschlüsselte Daten zur Verfügung. Diese analysiert er mit verschiedenen Methoden.

  _____

bis 16h05

Aufgabe 2

Erklären Sie kurz, warum eine Chosen-plaintext Attacke bei einer Public Key Verschlüsselung immer möglich ist.

Kap. 10.3 im JS Skript

Die 3 Attacken an der Stromchiffre

Hausaufgabe/Selbststudium

Repetieren Sie dazu insbesondere auch das Stromchiffre Puzzle in Aufgabe 7.5

Kap. 10.4

Die zwei Brute-Force Attacken

$y = ax + b$ → auf x auflösen

$ax = y - b$

$x = \frac{y-b}{a} = a^{-1}(y-b)$

(wenn $ggT(m, a) = 1$

Affine Chiffre

Bevor wir die Brute Force Attacken beschreiben, führen wir zuerst die affine Chiffre (cf. [CP-D], Kap. 1.4.4, resp. 25.4.5 im JS Skript) ein.

- Es sei $x, y, a, b \in \mathbb{Z}_{26}$ mit $\text{ggt}(a, 26) = 1$, d.h. $a \in \mathbb{Z}_{26}^*$
- **Verschlüsselung**: $y = e_k(x) \equiv (a \cdot x + b)\ mod\ 26$
- **Entschlüsselung**: $x = d_k(y) \equiv a^{-1} \cdot (y - b)\ mod\ 26$
- mit dem Schlüssel $k = (a, b)$, mit $\text{ggt}(a, 26) = 1$, es gibt $\varphi(26) = \varphi(2 \cdot 13) = \varphi(2) \cdot \varphi(13) = 1 \cdot 12 = 12$ Elemente, nämlich $a \in \{1; 3; 5; 7; 9; 11; 15; 17; 19; 21; 23; 25\}$
- Die Anzahl möglicher Keys $k = (a, b)$ beträgt $12 \cdot 26 = 312$
- Falls man a = 1 und b = 0 ausschliesst $\Rightarrow 11 \cdot 25 = 275$ Keys

Beispiel: $k = (a, b) = (5, 20), x = \text{ATTACK} \overset{Codierung}{\equiv} 0,19,19,0,2,10$

$y = (5 \cdot x + 20)\ mod\ 26 = 20,11,11,20,4,18 \overset{Decodierung}{=} ULLUES$

$x = \underbrace{5^{-1}}_{=21} \cdot (y - 20)\ mod\ 26 = 0,19,19,0,2,10 = ATTACK$

Check: $5 \cdot 21 \equiv 105 \equiv 104 + 1 \equiv 4 \cdot 26 + 1 \equiv 1\ mod\ 26$

Affine Chiffre, Aufgaben

Aufgabe 3

Wir betrachten wiederum die affine Chiffre von vorhin. Nun aber wollen wir anstatt mit mod 26 neu mit mod 23 rechnen.

1) Berechnen Sie nun die Grösse des Schlüsselraumes k.
2) Wäre das eine gute Idee?

bis 16.4.25

Aufgabe 4

1) Wie sähe es aus, wenn es genau 31 Zeichen im Alphabet gäbe? Wie würden Sie dann die affine Chiffre definieren?
2) Nun betrachten wir wiederum unserer Alphabet mit (nur) 26 Zeichen. Wäre es – wegen des grösseren Schlüsselraumes – nun jetzt eine gute Idee mit mod 31 zu rechnen?
3) Was könnten Sie tun, damit mit mod 31 zu rechnen, eine gute Idee wäre?

Beachten Sie auch die Aufgaben im JS Skript, Kap. 25.4.5.

Die voll. Schlüsselsuche = Exhaustive Key Search

- Ist eine Known-plaintext Attacke.
- D.h. mind. ein Klartext-Chiffretextpaar (M, C) ist bekannt. Damit kann der vermutete Schlüssel getestet werden.
- D.h. entweder $C \overset{?}{=} E(M, K_j)$ oder $M \overset{?}{=} D(C, K_j)$ überprüfen.
- Hat der Schlüsselraum k Schlüssel, so braucht ein vollständiges Durchsuchen k Verschlüsselungen. Im statistischen Mittel ist man mit k/2 Operationen fertig.
- Also Unterschied beachten:
  - Vollständige Schlüsselsuche = k Operationen.
  - Im statischen Mittel ist man mit k/2 Operationen erfolgreich.

Beispiel:

- Von einer Meldung weiss man, dass sie mit ATTACK beginnt.
- Die Verschlüsselung von ATTACK, also ULLUES wird abgehört.
- D.h. Das Paar $(\text{ATTACK}; ULLUES)$ ist bekannt.
- Der Algorithmus (affine Chiffre) ist auch bekannt.
- Mit max. k = 312 (resp. mit den Einschränkungen für a und b, k = 275) Verschlüsselungen hat man den Schlüssel $k = (a, b)$. Resp. im stat. Mittel k/2 = 156 (resp. k/2 = 137,5) Operationen hat man den Schlüssel $k = (a, b)$.

Die voll. Schlüsselsuche = Exhaustive Key Search

Vollständige Schlüsselsuche mit 1, 1000 resp. 10‘000‘000 Chips mit einer Leistung von 1’000’000 Verschlüsselungen pro Sek.

Schlüssel	$k = 2^{\#Bits}$	vollständige Suche mit 1 Chip = 10^6 Op. pro Sek.	Voll. Suche mit 1’000 Chips = 10^9 OP. pro Sek.	Cluster mit durchsch. 10^{11} Op. pro Sek.	Voll. Suche mit 10’000’000 Chips = 10^{13} pro Sec
32 Bit	$4{,}3 \cdot 10^9$	1 h 12 min	4,3 Sek.		
40 Bit	$1{,}1 \cdot 10^{12}$	12,7 Tage	20 min		
45 Bit	$3{,}0 \cdot 10^{13}$	1 Jahr	8 Std		
48 Bit	$2{,}8 \cdot 10^{14}$	9 Jahre	78 Std		
56 Bit	$7{,}2 \cdot 10^{16}$	2304 Jahre	2,3 Jahre	10 Tage	2 Std
64 Bit	$1{,}8 \cdot 10^{19}$	$6 \cdot 10^5$ Jahre	581 Jahre	6 Jahre	22 Tage
128 Bit	$3{,}4 \cdot 10^{38}$	$1{,}1 \cdot 10^{25}$ Jahre	$1{,}1 \cdot 10^{22}$ Jahre	$1{,}1 \cdot 10^{20}$ Jahre	$1{,}1 \cdot 10^{18}$ Jahre

AES im Minimum 128 Bit

Bemerkung: Ein Bit mehr hat eine Verdoppelung des Aufwandes zur Folge. Die Verdoppelung jeder Schlüsselbit bedingt einen quadratischen Aufwand. $2^{\#Bit}$

„Unverschämte Ressourcen“:

Könnte man die gesamte Energie der Sonne, die 24 Std x 365 Tage auf die Erde scheint zur Schlüsselsuche brauchen, so könnte man in einem Jahr den Schlüsselraum eines 157 Bit Schlüssel durchsuchen.

Vollst. vorab berechn. Tabelle = Table look up

- Ist eine Chosen-plaintext Attacke.
- Hat der Schlüsselraum k Schlüssel, so hat man…
  - … einen Vorbereitungsaufwand von k Verschlüsselungen.
  - Und es braucht k Einträge in einer Datenbank.
- Vorbereitung kann massiv parallel durchgeführt werden.
- Für den Angriff – nicht aber für die Vorbereitung – muss man Zugriff auf das Chiffriergerät haben.

Beispiel:

- Es wird (frei wählbar) ATTACK gewählt.
- Ich führe alle k Verschlüsselungen durch.
- Dadurch braucht es 312 (resp. 275) Einträge in der Datenbank. Pro Eintrag braucht es 2 Felder für den Schlüssel $k = (a, b)$ und 6 Felder für das Chiffrat. Total also 312*8 = 2494 Felder, (resp. 275*8 = 2200).
- Den Klartext ATTACK ins Chiffriergerät eingeben, es kommt ULLUES raus. Ein Blick in die Datenbank zeigt $k = (5, 20)$

Table look up

Platzbedarf in # WORMs und # Harddisks für eine vorab berechnete Schlüsseltabelle.
Dabei nehmen wir an, dass die **Blockgrösse** identisch zur **Schlüsselgrösse** ist.
WORM = WriteOnceReadMany (z.B. eine CD)

Blockgrösse = **Schlüsselgrösse**	**Platzbedarf = #Bits** $\cdot 2^{\#Bits}$.	**# WORMs mit 3 GByte =** $2,4 \cdot 10^{10}$ **Bit**	**# HD's mit 10 Terabyte =** $8 \cdot 10^{13}$ **Bit**
32 Bit	$1{,}4 \cdot 10^{11}$	6	--
40 Bit	$4{,}3 \cdot 10^{13}$	1700	0,5
45 Bit	$1{,}6 \cdot 10^{15}$	61'500	20
48 Bit	$1{,}35 \cdot 10^{16}$	$5{,}3 \cdot 10^{5}$	200
56 Bit	$4{,}0 \cdot 10^{18}$	$1{,}6 \cdot 10^{8}$	50'000
64 Bit	$1{,}2 \cdot 10^{21}$	$4{,}9 \cdot 10^{10}$	$1{,}5 \cdot 10^{7}$
128 Bit	$128 \cdot 2^{128} = 2^{7} \cdot 2^{128} = 2^{135} = 4{,}4 \cdot 10^{40}$	$\frac{4{,}4 \cdot 10^{40}}{2{,}4 \cdot 10^{10}} = 1{,}8 \cdot 10^{10}$	$5{,}4 \cdot 10^{26}$

„Unverschämte Ressourcen":

Würde die ganze Erde aus Silizium bestehen, und hätte man diese „Siliziumerde" als Speichermedium zur Verfügung, so hätte man Platz für eine vorab berechnete Tabelle eines 157 Bit Schlüssels.

bis 16445

Kap. 10.5

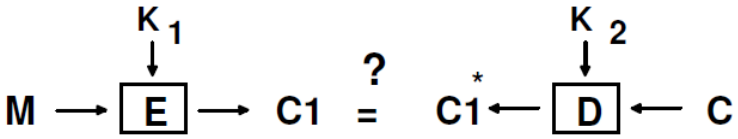
Time-Memory Tradeoffs TMTO

z.B. «Meet in the middle»

Nicht zu Verwechseln mit Man in the middle

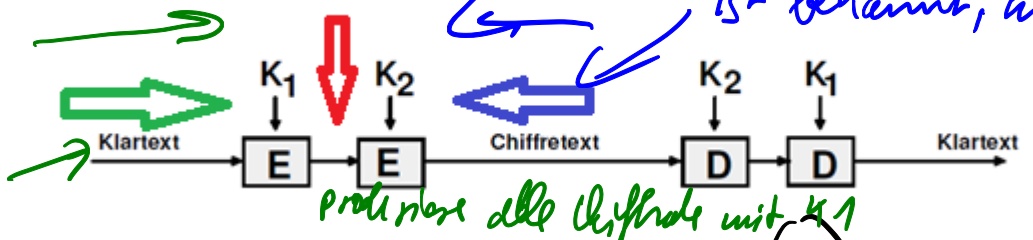
Time-Memory Tradeoff, TMTO

- Ist eine Kombination der beiden Brute-Force Attacken.
- Der Typ wurde gefunden, weil man den Doppel- resp. Tripple-DES für die Vergrösserung der Sicherheit kreiert hat.
- Bei Mehrfach-Verschlüsselungen sind es oft «meet in the middle»-Attacken:



- → nicht zu verwechseln mit «man in the middle».
- Es gibt aber auch «reine» Time-Memory Tradeoff Attacken, die auf 1-fach Blockverschlüsselungen wie AES anwendbar sind. → Nicht Inhalt des Moduls, mathematisch anspruchsvoll.
- ***Also:*** „Meet in the middle Attacke“ $\overset{\Rightarrow}{\not\Leftarrow}$ Ist eine Time-Memory Tradeoff Attacke (TMTO).

Eine 2-fach Verschlüsselung (z.B. Doppel-DES)



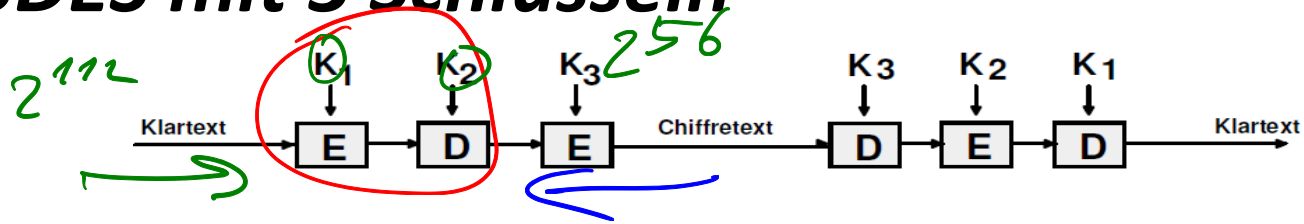
- Der Brute-Force Schlüsselraum beträgt $2 \cdot 56 = 112$ Bit. D.h. $k = 2^{112} \approx 5 \cdot 10^{33}$.
- Wir führen eine Known-plaintext Attacke durch, d.h. wir haben ein bekanntes Klartext-Chiffretext Paar.
- Die erste Verschlüsselung greift man mit dem Klartext an. Nach 2^{56} Verschlüsselungen und Abspeichern aller Werte, kann man von hinten mit dem Chiffretext (max.) alle 2^{56} Entschlüsselungen durchführen. Sobald man in der Mitte das gleiche Resultat hat man ***einen*** Schlüssel gefunden.
- Es braucht also im Maximum $2 \cdot 2^{56} = 2^{57}$ Verschlüsselungen und 2^{56} Einträge in der Datenbank.
- Die Sicherheit des Doppel-DES erhöht sich also nur um 1 Bit!!
- Ev. braucht es mehrere Paare, der Angriff muss aber nur noch auf einen kleinen Teil fortgesetzt werden. D.h. der Aufwand vergrössert sich nur marginal, auch wenn man z.B. 2^{47} «passende» Schlüssel im ersten Schritt gefunden hat!

3DES mit 2 Schlüsseln

- Geglaubte Sicherheit 112 Bit, also Brute-Force Aufwand zum Brechen ca. 2^{112}.
- Meet-in-the-middle wie bei Doppel-DES von vorhin wäre tatsächlich auch in etwa in der Grössenordnung von ca. 2^{112}, also etwa wie Brute-Force.
- Es gibt aber einen Chosen plaintext Angriff TMTO, der dann schlussendlich einen Aufwand von ca. 2^{60} in Anzahl Berechnungen und ca. 2^{60} Anzahl Platz hat.
- Details siehe JS Skript, Kap. 10.5.3 (Nicht Prüfungsstoff).
- Zusammenfassend kann man sagen:

Ein Sicherheitsgewinn von einer 2-fach Verschlüsselung mit 2 Schlüsseln zur 3-fach mit 2 Schlüsseln ist gegeben, aber er ist nicht substanziell.
Selbst der Aufwandzuwachs zur 1-fach Verschlüsselung ist (natürlich) gegeben, aber auch nicht berauschend.

3DES mit 3 Schlüsseln



- Geglaubte Sicherheit 168 Bit, also Brute-Force Aufwand zum Brechen ca. 2^{168}.
- Hier funktioniert wiederum die Meet-in-the-middle Attacke wie bei Doppel-DES von vorhin.
- Der Aufwand ist ca. 2^{112} für die Schlüsselberechnung und 2^{56} Blöcke in einer Tabelle gespeichert.
- Es ginge auch mit 2^{56} für die Schlüsselberechnung und 2^{112} Blöcke in einer Tabelle gespeichert. Nachteil: Vorbereitungsaufwand ist 2^{112}.
- für die
- Auch in JS Skript, Kap. 10.5.4 sind nicht alle Details nicht aufgeführt (so oder so nicht Prüfungsstoff).
- Mit AES sind Mehrfach – Verschlüsselungen nicht mehr nötig.
- Zusammenfassend kann man sagen:

Eine 3-fach Verschlüsselung mit 3 Schlüsseln ist zwar viel stärker als eine 3-fach Verschlüsselung mit 2 Schlüsseln. Die Sicherheit ist aber mit 112 Bit deutlich kleiner als es der grosse Schlüsselraum mit 168 Bit vermuten lässt. Z.Z. sind 112 Bit bis 2030 noch genügend und daher ist 3DES mit 3 Schlüsseln auch bis 2030 noch erlaubt.

Eine 1-fach Blockchiffre (z.B. DES; AES u.a.)

- Auch hier gibt es solche kombinierten Angriffe, die wir aber im Detail nicht betrachten.
- Prognosen und Annahmen bei einer Schlüsselraumgrösse k:
  - Aktuell der Aufwand von je $k^{\frac{2}{3}}$ der beiden Brute-Force Attacken. Das bei einer Erfolgswahrscheinlichkeit von ca. 70%.
  - Pessimistisch: Aufwand von je $k^{\frac{1}{3}}$ der beiden Brute-Force Attacken.

Beispiel

Time-Memory Tradeoff (TMTO) für einen **64** Bit Schlüssel und **64** Bit Blockgrösse.

Schlüssel = Block = 64 Bit	**Platzbedarf = #Bits · $2^{\#Bits}$**	**Zeit = k = $2^{\#Bits}$**
Table look up	$64 \cdot 2^{64} = 2^{70} = 1{,}2 \cdot 10^{21}$	1
Key Search	1	$2^{64} = 1{,}8 \cdot 10^{19}$
TMTO je $k^{\frac{2}{3}}$	$(2^{70})^{2/3} \approx 2^{47} \approx 1{,}1 \cdot 10^{14}$	$(2^{64})^{2/3} \approx 2^{43} \approx 7 \cdot 10^{12}$
Theoret. TMTO je $k^{\frac{1}{3}}$	$\sqrt[3]{2^{70}} \approx 2^{23} \approx 10^{7}$	$\sqrt[3]{2^{64}} \approx 2^{21} \approx 2 \cdot 10^{6}$

Resumée:

Es bräuchte also etwa den Aufwand von einem Key Search und einem Table look up eines **43** Bit Schlüssels, resp. im theoretischen Fall eines **21** Bit Schlüssels.

Bemerkung:

Eigentlich sind die obigen Zahlenwerte viel zu genau. In der Tabelle sind nur Abschätzungen, die konkreten Zahlen ergeben sich aus dem konkreten Angriff und können bis Faktor 10 und mehr von den Werten in der Tabelle abweichen!!

Aufgabe 5

Füllen Sie die Tabelle für einen Time-Memory Tradeoff (TMTO) eines 128 Bit Schlüssels mit 128 Bit Blockgrösse.

Schlüssel = Block = 128 Bit	Platzbedarf = #Bits · $2^{\#Bits}$	Zeit = k = $2^{\#Bits}$
Table look up		
Key Search		
TMTO je $k^{\frac{2}{3}}$		
Theoret. TMTO je $k^{\frac{1}{3}}$		

Resumée:

Es bräuchte also etwa den Aufwand von einem Key Search und einem Table look up eines _____ Bit Schlüssels, resp. im theoretischen Fall eines _____ Bit Schlüssels.

TMTO bei «unverschämten Ressourcen»

Mit «unverschämten Ressourcen» könnte man ja je die Brute-Force Attacken mit 157 Bit durchführen.

Frage:

Wie sähe das aus, wenn man beide «unverschämten Ressourcen» zur Verfügung hätte und mit je $\boldsymbol{k^{2/3}}$ rechnet? D.h. wie gross müsste der Schlüssel gewählt werden, damit je mit $\boldsymbol{k^{2/3}}$ gerechnet es die «unverschämten Ressourcen» bräuchte um zu knacken?

Antwort:

Nehmen wir Einfachheitshalber 160 Bit an. So müssen wir die Gleichung

$$(2^x)^{2/3} = 2^{160} \Rightarrow 2^x = \left(\underbrace{(2^x)^{2/3}}_{=2^{160}}\right)^{3/2} = (2^{160})^{3/2} = 2^{240} \Rightarrow x = 240.$$

Genauer: Wegen den 157 Bit und mal 1,5 gerechnet gibt es ca. 235 Bit.

Aufgabe 6

Wie sähe aus, wenn je mit $\boldsymbol{k^{1/3}}$ gerechnet werden müsste?

Bemerkung:

Beachten und bearbeiten Sie auch die Beispiele und Aufgaben im JS Skript.

TMTO im Vergleich mit Quantencomputer QC

Auf das Thema QC wird in Präsenz 13 (cf. Kap. 16.8) kurz eingegangen. In Bezug auf Blockchiffren kann man zusammengefasst in etwa Folgendes erwähnen:

- Blockchiffren sind viel resistenter als asymmetrische Verfahren, dennoch muss man beachten, dass …
- … die Angriffe mit QC auf Blockchiffren ca. im Quadrat besser sind als Brute-Force.

Beispiel (siehe auch Tabelle zur Aufgabe 5):

Bei AES mit einem 128 Bit Schlüssel hätte man bei…

… Brute-Force einen Aufwand von $2^{128} \approx 3{,}4 \cdot 10^{38}$

… mit Merkle Hellman $k^{\frac{2}{3}}$ sind es $(2^{128})^{\frac{2}{3}} = 2^{128 \cdot \frac{2}{3}} = 2^{\frac{256}{3}} \approx 5 \cdot 10^{25}$

… mit QC wären es noch $\sqrt{2^{128}} = 2^{\frac{128}{2}} = 2^{64} \approx 1{,}8 \cdot 10^{19}$

… mit Merkle Hellman worst case $k^{\frac{1}{3}}$ sind es $(2^{128})^{\frac{1}{3}} = \sqrt[3]{2^{128}} \approx 7 \cdot 10^{12}$

Mit einer Tabelle zusammengefasst:

Brute Force	**Merkle-Hellman mit $\mathbf{k^{2/3}}$**	**Mit Quantencomputer mit $\mathbf{\sqrt{k} = k^{1/2}}$**	**Merkle-Hellman vermutet mit $\mathbf{\sqrt[3]{k} = k^{1/3}}$**
$\mathbf{2^{128}}$	$\mathbf{(2^{128})^{2/3} \approx 2^{85}}$	$\mathbf{(2^{128})^{1/2} = 2^{64}}$	$\mathbf{(2^{128})^{1/3} \approx 2^{43}}$

Fazit 1: D.h. ein 128 Bit AES würde nicht mehr genügen, da 2^{64} zu wenig sicher wäre.

Fazit 2: Angriffe auf Blockchiffren mit Quantencomputer liegen damit zwischen dem $k^{\frac{2}{3}}$ Aufwand von Merkle Hellman und dem $k^{\frac{1}{3}}$ Aufwand des vermuteten worst case von Merkle Hellman.

bis 17.4.20

Kap. 10.6 im JS Skript

Drei vertiefte kryptoanalytische Betrachtungen

Tw. Hausaufgabe/Selbststudium

Betrachtung 1: Grösse eines Codebooks
ISO-1 PIN-Block Definition und Variante HEX F

Wir fokussieren uns im Folgenden i.d.R. nur auf 4-stellige PIN's und die Blockgrösse 64 Bit.

	PIN-Länge	PIN	Padding
1	4	PPPP	Diverse Typen, siehe unten

- An der ersten Stelle steht fix eine 1.
- An der zweiten Stelle steht die Länge der PIN, gemäss Standard eine Zahl von 4, …, C (= 12).
- In der Schweiz (Bankkarte usw.) ist nur 4, 5 oder 6 möglich.
- Danach kommen so viele PIN-Zahlen, wie vorher die Länge definiert ist.

Die erste und einfachste Variante des Paddings ist, dass der Block mit HEX F aufgefüllt wird. D.h. von diesem PIN-Block alles bekannt, ausser die 4-stellige PIN. Somit gibt es insgesamt über alle Benutzer hinweg nur 10'000 verschiedene solcher PIN-Blöcke. Und damit ist selbsterklärend, dass ein mögliches Codebook nur 10'000 Einträge hat.

	PIN-Länge	PIN	Padding
1	4	PPPP	0xFFFFFFFFF

ISO-1 PIN-Block weitere Padding Varianten

Verbesserung 1: Das Padding ist ein Teil der individuellen Kartennummer.

	PIN-Länge	PIN	Padding
1	4	PPPP	Teil der Kartennummer (10 Stellen)

- Es gibt nun pro Kartennummer 10'000 Möglichkeiten
- Theoretisch gibt es 10^{10} Kartennummern.
- Somit gibt es $10^{10} \cdot 10'000 = 10^{14}$ Einträge in einem Codebook.
- Bei 8 Byte pro Eintrag wird das Codebook ca. 1000 Terabyte gross.

Verbesserung 2: Das Padding wird pro Meldung jeweils mit Zufallsbits gestaltet.

	PIN-Länge	PIN	Padding
1	4	PPPP	40 Zufallsbits

- Pro Zufallsbit gibt 10'000 Möglichkeiten
- Theoretisch gibt es 10^{10} Kartennummern.
- Somit gibt es $2^{40} \cdot 10'000 \approx 10^{12} \cdot 10'000 = 10^{16}$ Einträge in einem Codebook, also eine Verbesserung um den Faktor 100 gegenüber der Verbesserung 1.
- Bei 8 Byte pro Eintrag wird das Codebook ca. 100'000 Terabyte gross.

Verbesserung 3: Die Blockgrösse betrage nun 128 Bit (z.B. AES).

- Das Codebook vergrössert sich nochmals um den Faktor $2^{64} \approx 2 \cdot 10^{19}$.
- Weitere (Detail-)Berechnungen im JS Skript, Kap. 10.6.1 → Hausaufgabe.

Betrachtung 2: Was ist der Unterschied?

	PIN-Länge	PIN	Padding
1	4	PPPP	0xFFFFFFFFF

Ein ISO-1 PIN-Block für eine 4-stellige PIN mit fixem Padding (siehe oben).
Diesen PIN-Block verschlüsseln wir nun mit einer Blockchiffren auf 2 Arten:

1) Der PIN-Block wird mit einem fixen Schlüssel verschlüsselt.
2) Der PIN-Block wird mit einem Zufallsschlüssel verschlüsselt, der Zufallsschlüssel wird mit einem fixen Masterschlüssel verschlüsselt und mitgeschickt.

Frage: Was ist der Unterschied zwischen 1) & 2) aus Sicht der Kryptoanalyse?

Kurzantwort (Details bitte im JS Skript in Kap. 10.6.2 als HA durcharbeiten):
Analyse anhand einer vollständigen Schlüsselsuche (known plaintext Attacke!):

1) Der fixe Schlüssel kann mit einer vollständigen Schlüsselsuche stark eingeschränkt werden. Die übrig gebliebenen Schlüssel können mit weiteren Paaren mit viel weniger Aufwand getestet werden. Mit etwa 4 – max. 10 Chiffraten und ca. 1,1 mal einer vollständigen Schlüsselsuche, wird dieser fixe Schlüssel erhalten.
2) Beim Verschlüsseln des Zufallsschlüssel funktioniert das nicht. Der Zufallsschlüssel hat keine «Kontur», daher kann ich keine Verschlüsselungsschlüssel aussondieren.

Betrachtung 3: Strom- vs. Blockchiffre

	PIN-Länge	PIN	Padding
1	4	PPPP	0xFFFFFFFFF

Ein ISO-1 PIN-Block für eine 4-stellige PIN mit fixem Padding (siehe oben).
Diesen PIN-Block verschlüsseln wiederum auf 2 Arten:

1) Mit einer Blockchiffre (Schlüsselgrösse 64 Bit, Blockgrösse 64 Bit)
2) Mit einer Stromchiffre

Nun betrachten wir für diese zwei Verschlüsselungen eine vollständige Schlüsselsuche.

Fazit:
Die vollständige Schlüsselsuche bei der Stromchiffre (das gilt auch für den One-Time-Pad) ist viel, viel weniger aufwändig!

Folgerung und Widerspruch?
Ist das nicht ein Widerspruch, dass wir gesagt haben, dass der OTP perfekt sicher sei?

Details bitte im JS Skript in Kap. 10.6.3 als HA durcharbeiten

Der Slogan zu dieser Präsenz, ein Widerspruch?

Eine Chiffre ist sicher $\overset{\Rightarrow}{\not\Leftarrow}$ ***Brute Force ist unmöglich, d.h.***

unmögliche Brute Force ist notwendig, aber nicht hinreichend!

Wir wollen die Aussage ***Eine Chiffre ist sicher ⇒ Brute Force ist unmöglich*** ein wenig unter die Lupe nehmen.
Wir nehmen wiederum unseren PIN-Block und verschlüsseln ihn mit einem OTP (also perfekt sicher!). Im konkreten Fall ist der Schlüssel 64 Bit gross.

	PIN-Länge	PIN	Padding
1	4	PPPP	40 Zufallsbits

Aufgabe 7

a) Wie gross ist die Brute-Force Attacke auf die PIN?
b) Was haben Sie nach dieser (offensichtlich nicht unmöglichen) Brute-Force Attacke erhalten?
c) Korrigieren Sie nun ggf. die Aussage von links nach rechts des Slogans.

Bemerkung:
Diese Aufgabe 7 ist eng verknüpft mit der vorherigen Folie, resp. mit der Verschlüsselung des PIN-Blocks mit einer Stromchiffre, resp. dem Kap. 10.6.3 im Skript.

Basis-Test Präsenz 5

Bis M430

Aussage	Richtig o. falsch?	Begründung
Eine Chiffre ist genau dann sicher, wenn Brute Force unmöglich ist.	☐ richtig ☐ falsch	
Die Kryptoanalyse ist eine rein mathematische Analyse.	☐ richtig ☐ falsch	
Das Prinzip von Kerckhoff besagt, dass Kryptoalgorithmen unbedingt geheim gehalten werden müssen.	☐ richtig ☐ falsch	
Ein System ist informationstheoretisch sicher, wenn es sicher ist, auch wenn der Gegner unbeschränkte Ressourcen zur Verfügung hat.	☐ richtig ☐ falsch	
Neben den 3 Attacken, die wir besprochen haben, gibt es noch weitere Attacken.	☐ richtig ☐ falsch	
Die Annahme, dass das Verschlüsselungsgerät dem Gegner zur Verfügung steht, ist eine völlig sinnlose Annahme.	☐ richtig ☐ falsch	
Eine Public Key Verschlüsselung muss notwendigerweise gegen eine Chosen-plaintext Attacke sicher sein.	☐ richtig ☐ falsch	

Basis-Test Präsenz 5, Fortsetzung

Aussage	Richtig o. falsch?	Begründung
Braucht eine vollständige Schlüsselsuche k Operationen, ist man im stat. Mittel nach k/2 Operationen erfolgreich.	☐ richtig ☐ falsch	
Grundsätzlich gibt es nur eine Brute Force Attacke.	☐ richtig ☐ falsch	
Eine Meet-in-the-middle Attacke ist identisch zu einer Man in the Middle Attacke.	☐ richtig ☐ falsch	
Eine Meet-in-the-middle Attacke ist eine Time-Memory Tradeoff (TMTO) Attacke.	☐ richtig ☐ falsch	
Ein Doppel-DES ist gegenüber einer TMTO Attacke nur wenig sicherer als ein Single-DES.	☐ richtig ☐ falsch	
Blockchiffren sind resistenter bez. Quantencomputer als es asymmetrische Verfahren sind.	☐ richtig ☐ falsch	
Man vermutet, dass der Einbezug von Quantencomputer beim Brechen von Blockchiffren der stärkst mögliche Angriff ist, resp. sein wird.	☐ richtig ☐ falsch	

Kap. 10.7 im JS Skript

Weitere Zahlenbeispiele

Hausaufgabe/Selbststudium

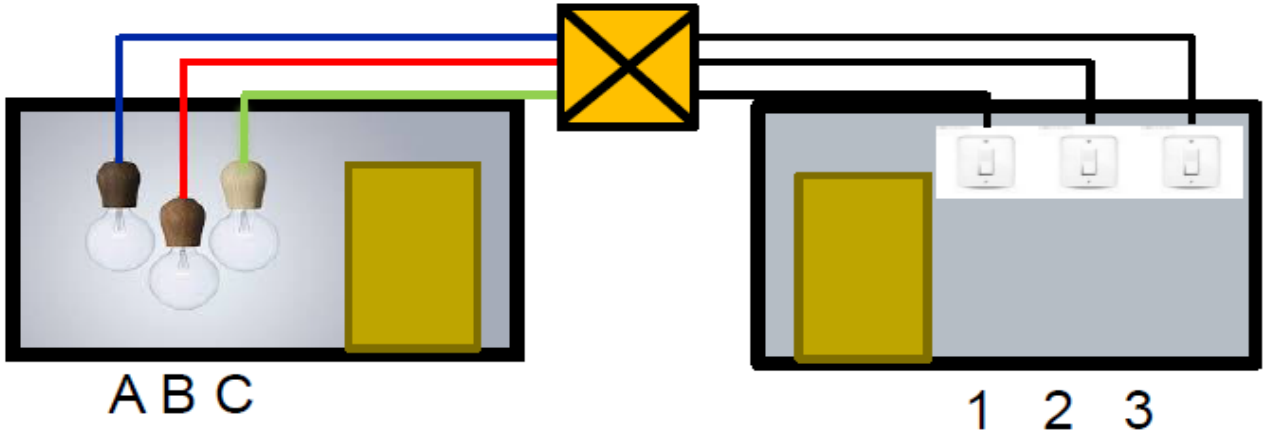
Kap. 10.8

Nachträge inkl. Übersicht der (kryptographischen) Attacken

Nicht Prüfungsstoff

Überlegen Sie mal…

- Selbst wenn es beweisbar ist, dass es ***keine mathematische Lösung*** geben kann, heisst das noch lange nicht, dass keine praktische Lösung existiert.
- **„An implementation of a theoretically secure cryptosystem need not be secure in practice“**
- Prof. Dr. Rolf Oppliger zeigt das in seiner Vorlesung mit dem folgenden Modell:
  - Two rooms – one with 3 light switches and the other with 3 light bulbs
  - The wiring of the light switches and bulbs is unknown
  - The adversary has to find out the wiring, but he or she can enter each room only once



- **Theorist (e.g., mathematician):** (Provably) impossible to solve.
- **Prcatitioner (e.g., physician): … → Schreiben Sie Vorschläge ins Forum.**

Fazit: Beware of side channels and new ways of solving problems and breaking systems.

Z.B. Side Channel Attacken

Beispiel 2.5: Side Channel Attacke, hier eine SPA (Simple Power Analysis)

- Beim RSA werden im Wesentlichen Multiplikationen und Quadraturen gemacht.
- Sei $(1001101111)_2$ ein Schlüssel, dann muss pro «1» eine Multiplikation und eine Quadratur und pro «0» eine Quadratur gemacht werden.
- Diese Operationen brauchen aber unterschiedliche elektrische Leistung. Mit einer SPA (Simple Power Analysis) – also einem bestimmten Typ von Side Channal Attacke – kann der Schlüssel bestimmt werden.
- Dies wird erst mit entsprechenden Gegenmassnahmen verhindert.

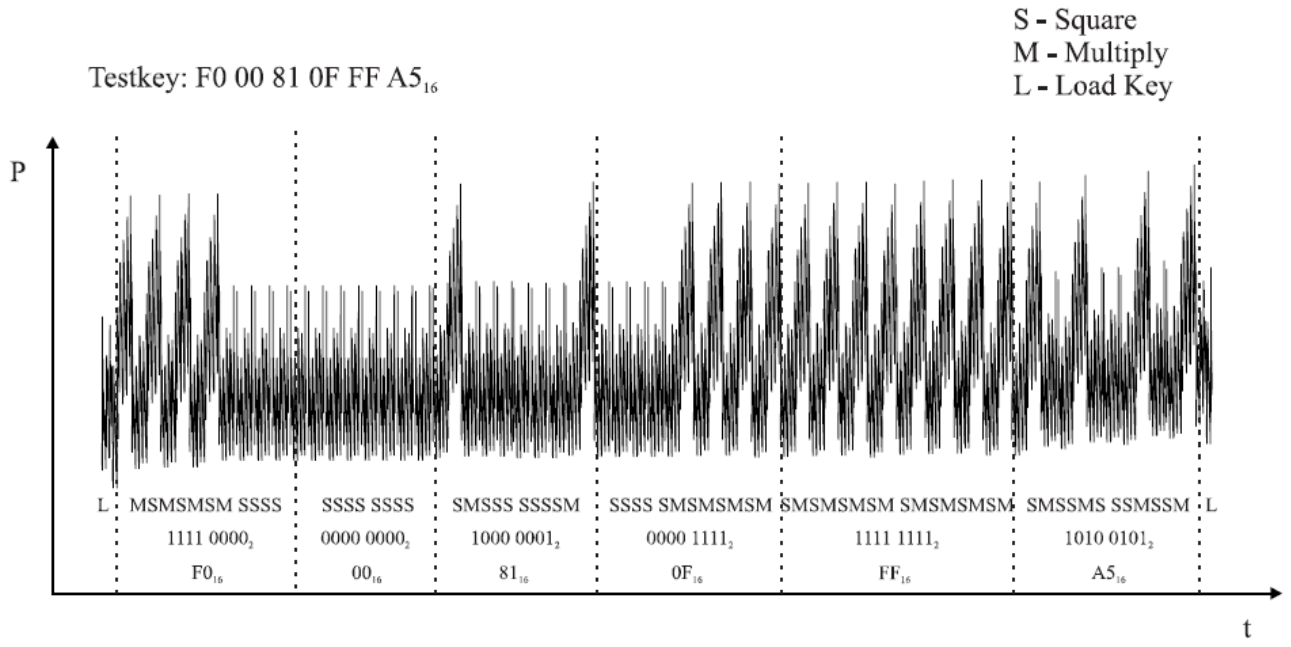


Figure 5.18: SPA Attack Against the RSA Algorithm

Side Channel Attacke? → Nichts Neues!!

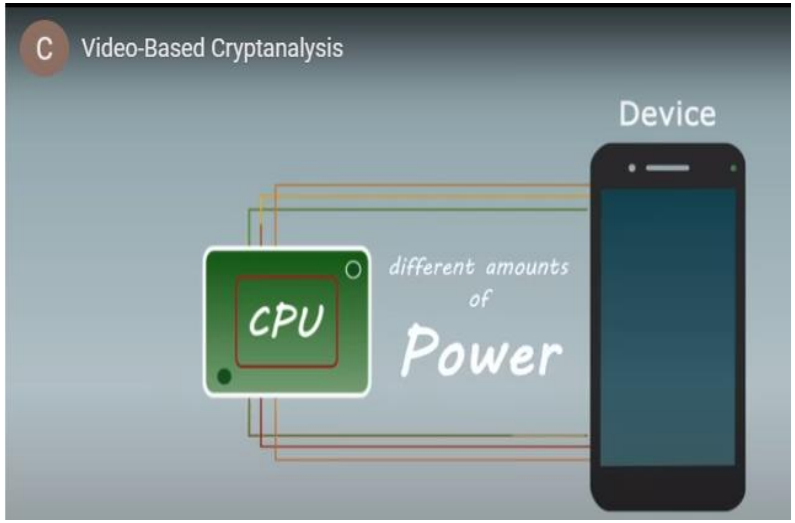
Side Channel Attacke & doch immer wieder neu!

Video-Based Cryptanalysis (*)

Geheime Keys von nicht kompromittierten Geräten bestimmen, indem Videomaterial der Power-LED verwendet wurde, das von kommerziellen Videokameras erhalten wurde.

Beschreibung und viele Videos dazu in: https://www.nassiben.com/video-based-crypta

(*) Details sind ***nicht*** Prüfungsstoff!!



Die erste Implementations Attacke

Die Implementationsattacken existieren seit ca. 1996. Es begann mit einem harmlosen Trick um die auf der Chipkarte gespeicherte PIN (oft nur 4-stellig) zu eruieren. Bei einem falschen PIN wurde der Retryzähler um Eins nach unten geschrieben. Dies hatte zur Folge, dass ein Schreibbefehl initiiert wurde. Schreiben braucht aber mehr Strom und damit konnte man das mit einer entsprechenden Schaltung merken und den ein Power Off einleiten, damit der Schreibbefehl nicht ausgeführt werden konnte. Im statistischen Mittel war man nach n/2 – also bei einer 4-stelligen PIN nach 5000 Versuchen – erfolgreich. Mit Hilfe der richtigen Umgebung war das Finden der PIN somit eine Frage von wenigen Stunden.

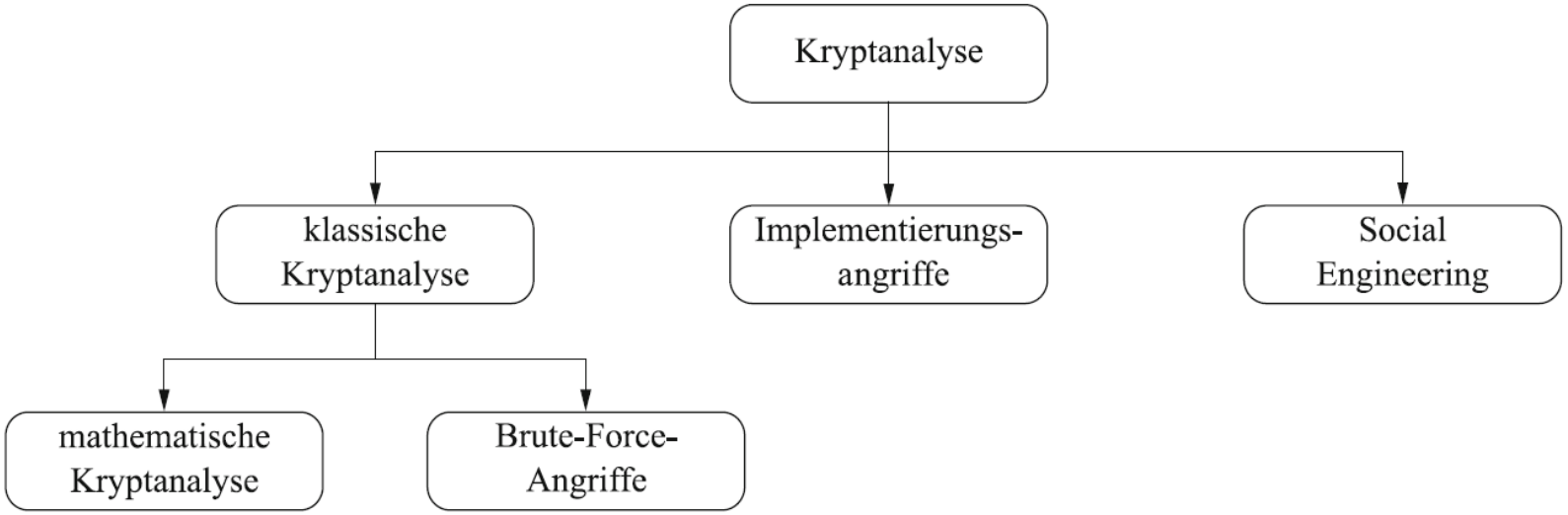
Aufgabe 8 Was war wohl die (erste) Gegenmassnahme?

Bemerkung:

Heute sind sowohl die Attacken wie die Gegenmassnahmen viel komplexer.

Die 3 wichtigen Grundtypen

1. Klassische Kryptoanalysis
   - Time-Memory Tradeoff
   - Differential Cryptoanalysis
   - Linear Cryptoanalysis
   - (Weitere) statistische Methoden, wobei viele davon nur noch historischen Wert haben (Kasiski-Test, Methoden um Enigma zu knacken usw.)
2. Social Engineering wie Erpressung und Drohung
3. Implementations Attacken, viele, viele Typen
   - Z.B. Side Channel Attacken, wie in JS Skript „Einführung in die Kryptologie“, 2.5.3, Beispiel 2.5, resp. in den vorherigen Folien angetönt wurde.



LÖSUNGEN DER AUFGABEN

Aufgabe 1

- Angriff 1: Known Plaintext
- Angriff 2: Chosen Plaintext
- Angriff 3: Ciphertext only

Aufgabe 2 Da mit dem Public Key des Empfängers verschlüsselt wird und dieser allseits bekannt ist.

Aufgabe 3

1) Da 23 eine Primzahl ist, gibt es für a neu 22 Möglichkeiten, d.h. die Anzahl möglicher Keys $k = (a, b)$ beträgt $22 \cdot 23 = 506$
2) Trotz des grösseren Schlüsselraumes, wäre das eine ganz schlechte Idee. Denn die Buchstaben X, Y, Z (23, 24, 25) hätten die gleiche Codierung wie A, B C, also 0, 1, 2.

Aufgabe 4

1) Da 31 eine Primzahl ist, gibt es für a nun 30 Möglichkeiten, d.h. die Anzahl möglicher Keys $k = (a, b)$ beträgt $30 \cdot 31 = 930$
2) Nein es wäre nach wie vor keine gute Idee. Denn ein Chiffrat kann ja in eine Zahl > 26 abgebildet werden. Aber dazu fehlen uns die Zeichen.
3) 5 weitere Zeichen (z.B. «Leerschlag», «?», «!», «,», «.») dazu nehmen.

Aufgabe 5

Schlüssel = Block = 128 Bit	Platzbedarf = #Bits · $2^{\#Bits}$	Zeit = k = $2^{\#Bits}$
Table look up	$128 \cdot 2^{128} = 2^{135} = 4{,}4 \cdot 10^{40}$	1
Key Search	1	$2^{128} = 3{,}4 \cdot 10^{38}$
TMTO je $k^{\frac{2}{3}}$	$(2^{135})^{2/3} \approx 2^{90} \approx 1{,}2 \cdot 10^{27}$	$(2^{128})^{\frac{2}{3}} \approx 2^{85} \approx 5\ 10^{25}$
Theoret. TMTO je $k^{\frac{1}{3}}$	$\sqrt[3]{2^{135}} \approx 2^{45} \approx 3{,}5 \cdot 10^{13}$	$\sqrt[3]{2^{128}} \approx 2^{43} \approx 7 \cdot 10^{12}$

Resumée:

Es bräuchte also etwa den Aufwand von einem Key Search und einem Table look up eines **85** Bit Schlüssels, resp. im theoretischen Fall eines **43** Bit Schlüssels.

Bemerkung:

Eigentlich sind die obigen Zahlenwerte viel zu genau. In der Tabelle sind nur Abschätzungen, die konkreten Zahlen ergeben sich aus dem konkreten Angriff und können bis Faktor 10 und mehr von den Werten in der Tabelle abweichen!!

Aufgabe 6 $(2^x)^{1/3} = 2^{160} \Rightarrow 2^x = \left(\underbrace{(2^x)^{1/3}}_{=2^{160}}\right)^3 = (2^{160})^3 = 2^{480} \Rightarrow x = 480.$

Resp. Wir können die Bitzahl mal 3 rechnen, also 480 Bit.

Genauer: resp. wegen den 157 Bit gibt es mal 3 gerechnet 471 Bit.

Aufgabe 7

a) Der Angriff muss nur auf die 4-stellige PIN gemacht werden. Mit 65'536 Möglichkeiten, hat man alle Möglichkeiten durchgerechnet. ***Fazit:*** Die Brute-Force Attacke ist sehr schnell und keineswegs unmöglich, eben im Gegenteil.

b) Nach der Brute-Force Attacke haben wir einfach alle Möglichkeiten von 0000, …, FFFF erhalten. Also auch alle möglichen 10'000 PIN's von 0000, …, 9999. D.h. nach der Attacke wissen wir genau gleich viel wie vorher: die PIN ist ein Wert von 0000, …, 9999. Und bei einem OTP würden uns dazu auch unendlich viele Ressourcen nichts nützen, wir wüssten nicht mehr. Bei einer Stromchiffre könnte es natürlich sein, dass wir mit unendlich vielen Ressourcen den Algorithmus zur Schlüsselstromerzeugung angreifen könnten. Dieser Algorithmus zur Schlüsselstromerzeugung muss eben so gut sein, dass es dazu sehr, sehr viele Ressourcen (Rechenpower und grosse Mengen von schon verbrauchtem Schlüsselmaterial) vorhanden ist.

c) Die Aussage können wir nun auf 2 Arten präzisieren:

- ***Eine Chiffre ist sicher ⇒ eine erfolgreiche Brute Force ist unmöglich***
- ***Eine Blockchiffre ist sicher ⇒ Brute Force ist unmöglich***

Anmerkung:

Die erste Präzisierung ist mir sympathischer, resp. unter „Brute Force ist unmöglich" ist selbstsprechend „***eine erfolgreiche*** Brute Force ist unmöglich" gemeint und damit behält der ursprüngliche Slogan seine Korrektheit.

Aufgabe 8

Die (erste) Gegenmassnahme war dann, dass der Retryzähler zu Beginn der PIN-Verifikation schon um Eins runter gezählt und gespeichert wurde. So entfiel nach einem PIN NOK das Schreiben. Bei PIN OK wurde der Zähler dann auf 3 gesetzt. Das wollte dann natürlich niemand mehr verhindern.

Basis-Test Präsenz 5

Aussage	Richtig o. falsch?	Begründung
Eine Chiffre ist genau dann sicher, wenn Brute Force unmöglich ist.	☐ richtig ☑ falsch	Sicher gegen Brute Force ist eine notwendige, aber nicht hinreichende Bedinung.
Die Kryptoanalyse ist eine rein mathematische Analyse.	☐ richtig ☑ falsch	Z.B. gehört Social Engeneering auch zur Kryptoanalyse.
Das Prinzip von Kerckhoff besagt, dass Kryptoalgorithmen unbedingt geheim gehalten werden müssen.	☐ richtig ☑ falsch	Im Gegenteil, nur die geheimen Schlüssel müssen geheim gehalten werden.
Ein System ist informationstheoretisch sicher, wenn es sicher ist, auch wenn der Gegner unbeschränkte Ressourcen zur Verfügung hat.	☑ richtig ☐ falsch	
Neben den 3 Attacken, die wir besprochen haben, gibt es noch weitere Attacken.	☑ richtig ☐ falsch	
Die Annahme, dass das Verschlüsselungsgerät dem Gegner zur Verfügung steht, ist eine völlig sinnlose Annahme.	☐ richtig ☑ falsch	Die Chosen-plaintext Attacke setzt voraus, dass man Zugriff auf das Verschlüsselungsgerät hat.
Eine Public Key Verschlüsselung muss notwendigerweise gegen eine Chosen-plaintext Attacke sicher sein.	☐ richtig ☑ falsch	Da man mit dem Public Key des Empfängers verschlüsselt, ist die asym. Verschlüsselung nicht schützbar gegen Chosen-plaintext.

Basis-Test Präsenz 5, Fortsetzung

Aussage	Richtig o. falsch?	Begründung
Braucht eine vollständige Schlüsselsuche k Operationen, ist man im stat. Mittel nach k/2 Operationen erfolgreich.	☑ richtig ☐ falsch	
Grundsätzlich gibt es nur eine Brute Force Attacke.	☐ richtig ☑ falsch	Es gibt zwei Brute Force Attacken, die vollständige Schlüsselsuche und die Vorab Berechnung einer Tabelle.
Eine Meet-in-the-middle Attacke ist identisch zu einer Man in the Middle Attacke.	☐ richtig ☑ falsch	
Eine Meet-in-the-middle Attacke ist eine Time-Memory Tradeoff (TMTO) Attacke.	☑ richtig ☐ falsch	
Ein Doppel-DES ist gegenüber einer TMTO Attacke nur wenig sicherer als ein Single-DES.	☑ richtig ☐ falsch	
Blockchiffren sind resistenter bez. Quantencomputer als es asymmetrische Verfahren sind.	☑ richtig ☐ falsch	
Man vermutet, dass der Einbezug von Quantencomputer beim Brechen von Blockchiffren der stärkst mögliche Angriff ist, resp. sein wird.	☐ richtig ☑ falsch	Der vermutete worst case der Merkle Hellman Attacke ist stärker.

Fazit/Zwischenziele

Bearbeiten der Aufgaben

- In Ilias ist das JS Skript „Aufgaben und Lösungen zum Modul KRYPT an der HSLU-I“ hochgeladen. Die aufgeführten Aufgaben beziehen sich auf die Aufgaben in [CP-D].
- Bitte beachten Sie, dass diese Aufgaben zum Prüfungsstoff gehören und in dieser Art an der Prüfung vorkommen können.

Durchführen der Standortbestimmung 1

- In Ilias ist die Standortbestimmung 1 hochgeladen. Lösen Sie nun diese.

Zwischenziele für den Teil 1 «sym. Krypto»

- Ich habe die Theorie & Aufgaben im JS Skript „Einführung in die Kryptologie“, durchgearbeitet und verstanden.
- Ich habe die Aufgaben im JS Skript „Aufgaben und Lösungen zum Modul KRYPT an der HSLU-I“, Kap. 2.1 „Aufgaben zu den Präsenzen Präsenz 0 – 5“ durchgearbeitet.
- Ich habe die Standortbestimmung 1 erfolgreich durchgeführt.