Angriffe auf DES

von: Thomas Kawollek
Betreuer: Stefan Kiefer

Hauptseminar Kryptographische Protokolle SS2009

1 Die Kryptanalyse

Die Kryptanalyse, auch Kryptoanalyse genannt, ist ein Teil der Kryptologie. Sie befasst sich mit dem Problem, eine verschlüsselte Nachricht zu entschlüsseln oder den benötigten Schlüssel herauszufinden. Anhand der Entschlüsselungsdauer kann man dann bestimmen, wie sicher ein Verschlüsselungsverfahren ist. Es gibt mehrere Methoden und Verfahren, die unter dem Begriff Kryptanalyse zusammengefasst werden. So kann man zum Beispiel einfach alle möglichen Schlüssel austesten (Brute-Force) oder bekannte häufige Schlüssel austesten (Wörterbuchangriff). 1990 wurde von Eli Biham und Adi Shamir die differenzielle Kryptanalyse veröffentlicht, welche der NSA und IBM anscheinend schon 1974 bekannt war. Ihre Vorgehensweise besteht darin 2 Klartexte, die sich nur minimal unterscheiden, zu verschlüsseln und in den zugehörigen verschlüsselten Texten nach Unterschieden zu suchen. Diese Unterschiede können dann auf den Schlüssel hindeuten.
Eine Weiterentwicklung dessen ist die lineare Kryptanalyse, welche 1993 von Mitsuru Matsui veröffentlicht wurde. Sie basiert auf der Idee sich mithilfe vieler Klartext-Geheimtext-Paare und gewissen Wahrscheinlichkeiten an den Schlüssel anzunähern.

2 Angriffe auf DES

Aufgrund des kurzen Schlüssels beim DES-Verschlüsselungsverfahren ist Brute-Force immer eine Alternative. Trotzdem wurden bei der DES-Challange 3 im Jahr 1999 mit dem Rechner Deep Crack und einem Netzwerk von 100.000 weiteren Rechnern noch 22 Stunden und 15 Minuten gebraucht, um den richtigen Schlüssel zu bestimmen.
Die differenzielle Kryptanalyse sollte DES schneller entschlüsseln als die Brute Force Methode, jedoch stellte sich heraus, dass DES gegen die differenzielle Kryptanalyse optimiert wurde.
Die lineare Kryptanalyse kann jedoch gut gegen DES eingesetzt werden.

2.1 Die lineare Kryptanalyse

Die Idee der linearen Kryptanalyse ist es, durch eine Gleichung, die aus Klartextbits, Geheimtextbits und Schlüsselbits besteht, eine Annäherung an die wahrscheinlichsten Schlüssel zu finden. Dies kann folgendermasen geschehen: Das Ziel der linearen Kryptanalyse ist es eine Formel der Form

$$P[i_1,i_2,...,i_a] \oplus C[j_1,j_2,...,j_b] = K[k_1,k_2,...,k_c]$$

aufzustellen, welche mit einer Wahrscheinlichkeit p, die nicht 50% ist, gelten soll. Dabei steht P für Plaintext(Klartext), C für Ciphertext(Geheimtext), K für Key(Schlüssel), $\oplus$ für XOR
und i_1, i_2, j_1, j_2, k_1, k_2,..., k_c für Bitpositionen.
Die Effektivität dieser Gleichung wird durch die Größe von $|p - 1/2|$ gegeben. Das bedeutet, dass die Gleichung dann am effektivsten ist, wenn die Wahrscheinlichkeit, dass diese stimmt, weit abweichend von 50% ist. Um dies zu überprüfen, kann man folgenden Algorithmus verwenden:

Schritt 1:
N soll für die Anzahl aller möglichen Klartexte stehen
T soll die Anzahl aller Klartexte sein, für welche die linke Seite der Gleichung 0 ergibt

Damit kann man überprüfen, für wie viele Klartext-Geheimtext-Paare der Schlüssel 0 ist, und für wie viele 1. Falls es ausgeglichen ist, also T = N/2, bedeutet dies, dass die Gleichung keine lineare Abhängigkeit zeigt.

Schritt 2:
Falls $T > N/2$ und $p > 1/2$: Man vermutet, dass $K[k_1,k_2,...,k_c] = 0$,
falls $T > N/2$ und $p < 1/2$: Man vermutet, dass $K[k_1,k_2,...,k_c] = 1$
falls $T < N/2$ und $p > 1/2$: Man vermutet, dass $K[k_1,k_2,...,k_c] = 1$
falls $T < N/2$ und $p < 1/2$: Man vermutet, dass $K[k_1,k_2,...,k_c] = 0$

Wenn also die Wahrscheinlichkeit und das tatsächliche Verhältnis, ausgedrückt durch T und N/2, übereinstimmen, kann angenommen werden, dass $K[k_1,k_2,...,k_c] = 0$.
Der Erfolg dieses Algorithmuses steigt mit größerem N und einem größeren Wert für $|p – 1/2|$.

Für die praktische Anwendung wird nun jedoch davon ausgegangen, dass man die vorletzte Runde betrachtet. Damit kommt man auf die Formel
$P[i_1,i_2,...,i_a] \oplus C[j_1,j_2,...,j_b] \oplus F_n(C_L,K_n)[l_1,l_2,...,l_d] = K[k_1,k_2,...,k_c]$
bei der der Ausdruck $F_n(C_L,K_n)[l_1,l_2,...,l_d]$ hinzugekommen ist, welcher für die Funktion in der letzten Runde stehen soll. C_L gibt dabei die rechte Hälfte des Geheimtextes an, und K_n den Schlüssel, welcher in der letzten Runde verwendet wird.
Um nun K_n zu bestimmen, wird ein 2. Algorithmus verwendet.

Schritt 1:
Es können für K_n beliebige Schlüssel eingesetzt werden.
T_i soll die Anzahl aller Klartexte stehen, für welche die linke Seite der Gleichung für den i. Schlüssel 0 ergibt.

Schritt 2:
T_{Max} soll für das größte T_i stehen, T_{Min} für das kleinste.
Falls $|T_{Max} – N/2| > |T_{Min} - N/2|$ und $p > 1/2$: Man benutzt den zu T_{Max} gehörenden Schlüssel und vermutet, dass $K[k_1,k_2,...,k_c] = 0$,
falls $|T_{Max} – N/2| > |T_{Min} - N/2|$ und $p < 1/2$: Man benutzt den zu T_{Max} gehörenden Schlüssel und vermutet, dass $K[k_1,k_2,...,k_c] = 1$,
falls $|T_{Max} – N/2| < |T_{Min} - N/2|$ und $p > 1/2$: Man benutzt den zu T_{Min} gehörenden Schlüssel und vermutet, dass $K[k_1,k_2,...,k_c] = 1$,
falls $|T_{Max} – N/2| < |T_{Min} - N/2|$ und $p < 1/2$: Man benutzt den zu T_{Min} gehörenden Schlüssel und vermutet, dass $K[k_1,k_2,...,k_c] = 0$.

Somit wird immer der Wert für K_n verwendet, welcher am weitesten von 50% entfernt ist, und wenn dies mit der Wahrscheinlichkeit übereinstimmt, wird $K[k_1,k_2,...,k_c]$ gleich 0 gesetzt. Dies findet später Anwendung.

2.2 Lineare Annäherung an die S-Boxen

Wenn es bei der DES-Verschlüsselung keine S-Boxen geben würde, wären alle Verschlüsselungsoperationen lineare Abbildungen, und somit für die lineare Kryptanalyse einfach zu entschlüsseln. Das Problem liegt also in den S-Boxen. Aus diesem Grund schauen wir dieses Problem nun genauer an. Wenn man ein Eingabebit verändert, kann man nicht sofort auf ein Ausgabebit schließen. Deswegen muss man nun versuchen aus mehreren Eingabebits auf mehrere Ausgabebits zu schließen. Die Eingabebits werden als α bezeichnet, und die Ausgabebits als β. Das bedeutet, dass α einen Wert zwischen 1 und 63, und β einen Wert zwischen 1 und 15 annehmen kann. $NS(\alpha,\beta)$ wird nun definiert als die Anzahl der Fälle, dass der XOR-Wert der Eingabebits mit dem der Ausgabebits übereinstimmt.

In nebenstehendem Tabellenausschnitt ist für α und β der jeweilige NS(α,β) - 32 Wert gegeben.

Am effektivsten ist eine Gleichung, wenn |NS(α,β) – 32| maximal ist. Das bedeutet, dass NS(16,15) = 12 der optimale Wert für die Annäherung ist.

Aus der vollständigen Tabelle kann man entnehmen, dass NS(α,β) immer gerade ist, und falls α = 1,32 oder 33 ist NS(α,β) =32.

Für ein zufällig gegebenes X kann man dann mit einer Wahrscheinlichkeit von 19% die Gleichung
X[15] ⊕ F(X,K)[7,18,24,29] = K[22]
aufstellen.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	4	-2	2	-2	2	-4	0	4	0	2	-2	2	-2	0	-4
3	0	-2	6	-2	-2	4	-4	0	0	-2	6	-2	-2	4	-4
4	2	-2	0	0	2	-2	0	0	2	2	4	-4	-2	-2	0
5	2	2	-4	0	10	-6	-4	0	2	-10	0	4	-2	2	4
6	-2	-4	-6	-2	-4	2	0	0	-2	0	-2	-6	-8	2	0
7	2	0	2	-2	8	6	0	-4	6	0	-6	-2	0	-6	-4
8	0	2	6	0	0	-2	-6	-2	2	4	-12	2	6	-4	4
9	-4	6	-2	0	-4	-6	-6	6	-2	0	-4	2	-6	-8	-4
10	4	0	0	-2	-6	2	2	2	2	-2	2	4	-4	-4	0
11	4	4	4	6	2	-2	-2	-2	-2	-2	2	0	-8	-4	0
12	2	0	-2	0	2	4	10	-2	4	-2	-8	-2	4	-6	-4
13	6	0	2	0	-2	4	-10	-2	0	-2	4	-2	8	-6	0
14	-2	-2	0	-2	4	0	2	-2	0	4	2	-4	6	-2	-4
15	-2	-2	8	6	4	0	2	2	4	8	-2	8	-6	2	0
16	2	-2	0	0	-2	-6	-8	0	-2	-2	-4	0	2	10	-20
17	2	-2	0	4	2	-2	-4	4	2	2	0	-8	-6	2	4
18	-2	0	-2	2	-4	-2	-8	4	6	4	6	-2	4	-6	0
19	-6	0	2	-2	4	2	0	4	-6	4	2	-6	4	-2	0
20	4	-4	0	0	0	0	0	-4	-4	4	4	0	4	-4	0
21	4	0	-4	-4	4	-8	-8	0	0	-4	4	8	4	0	4
22	0	6	6	2	-2	4	0	4	0	6	2	2	2	0	0
23	4	-6	-2	6	-2	-4	4	4	-4	-6	2	-2	2	0	4
24	6	0	2	4	-10	-4	2	2	0	-2	0	2	4	-2	-4
25	2	4	-6	0	-2	4	-2	6	8	6	4	10	0	2	-4
26	2	2	-8	-2	4	0	2	-2	0	4	2	0	-2	-2	0
27	2	6	-4	-6	0	0	2	6	8	0	-2	-4	-6	-2	0
28	0	-2	2	4	0	-6	2	-2	6	-4	0	2	-2	0	0
29	4	-2	6	-8	0	-2	2	10	-2	-8	-8	2	2	0	4
30	-4	-8	0	-2	-2	-2	2	-2	2	-2	6	4	4	4	0
31	-4	8	-8	2	-6	-6	-2	-2	2	-2	-2	-8	0	0	-4
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

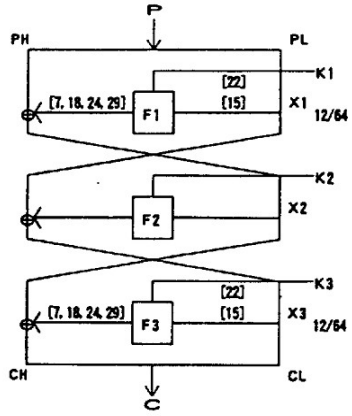
Table 1. A distribution table of S5 (part).

Quelle: [1]

2.3 Lineare Annäherung an DES

Rechts sieht man ein Beispiel von einem 3-Runden DES.
Hier kann man 2 Gleichungen aufstellen. Eine nach der 1. Runde:
$X_2[7,18,24,29] \oplus P_H[7,18,24,29] \oplus P_L[15] = K_1[22]$
und eine vor der letzten Runde:
$X_2[7,18,24,29] \oplus C_H[7,18,24,29] \oplus C_L[15] = K_3[22]$
Wenn man nun den gemeinsamen Teil entfernt und die Gleichungen zusammenfasst, erhält man:
$P_H[7,18,24,29] \oplus P_L[15] \oplus C_H[7,18,24,29] \oplus C_L[15] = K_1[22] \oplus K_3[22]$
Die Wahrscheinlichkeit, dass diese Gleichung nun für einen zufälligen Klartext und den zugehörigen Geheimtext gilt, ist:

$(12/64)^2 + (1 - 12/64)^2 = 0,70$



[Fig. 2] 3-round DES cipher

Quelle: [1]

2.4 Known-Plaintext Angriff

Über weitere Gleichungen, welche man dann zusammenfasst, kann man auch auf mehr Runden DES schließen. Dabei wird die Gleichung länger, und die Wahrscheinlichkeit geringer.
Bei einem Known-Plaintext Angriff benötigt man nun weiterhin Klartext-Geheimtext-Paare.
Bei 8-Runden-DES gilt mit einer Wahrscheinlichkeit von $0,5 + 1,95 \times 2^{-10}$:

$$P_H[7,18,24] \oplus P_L[12,16] \oplus C_H[15] \oplus C_L[7,18,24,29] \oplus F_8(C_L,K_8)[15] = K_1[19,23] \oplus K_3[22] \oplus K_4[44] \oplus K_5[22] \oplus K_7[22]$$

Das Problem an dieser Gleichung ist das K_8 auf der linken Seite. Da wir es noch nicht kennen, muss es eine Annäherung daran geben, und Matsui stellte fest, dass $F_8(C_L,K_8)[15]$ nur von den 6 Bits $K_8[42]$ bis $K_8[47]$ beeinflusst wird. Das bedeutet, dass wir 2^6=64 Versuche benötigen um den 2. Algorithmus anwenden zu können.

Deswegen geht man davon aus, dass der Erfolg der Gleichung nur von $l_1,l_2,...,l_d$ und $|p-1/2| \times \sqrt{N}$ abhängt. Deswegen kann man dann auch davon ausgehen, dass für einen Unterschlüssel $K_n^{(i)}$ und eine zufällige Variable X gilt:
$F_n(X,K_n)[l_1,l_2,...,l_d] = F_n(X,K_n^{(i)})[l_1,l_2,...,l_d]$
Mit dieser Gleichung und einer Anzahl an Klartext-Geheimtext-Paaren von $16 \times |p - 1/2|^{-2}$ oder mehr bekommt man eine Erfolgswahrscheinlichkeit von 99,9%
Wenn man dies nun nicht nur für die letzte Runde, sondern auch für die erste Runde anwendet, kann man relativ schnell 14 Teilschlüsselbits herausfinden. Die restlichen Bits des Schlüssels können dann über weitere Gleichungen recht leicht berechnet werden.
Analog dazu geht dies natürlich auch mit mehr-Runden-DES.
Dafür sind allerdings mehr Klartext-Geheimtext-Paare nötig, und es benötigt ebenfalls mehr Zeit.

2.5 Only-Ciphertext Angriff

Auf diesen Ergebnissen baut man nun beim Only-Ciphertext Angriff auf. Beim 8-Runden-DES gelten folgende 2 Gleichungen mit einer Wahrscheinlichkeit von nur knapp unter 50%.
(1) $P_L[27] \oplus C_H[27] \oplus C_L[0] \oplus F_8(C_L,K_8)[27] = K_2[1] \oplus K_3[8] \oplus K_4[1] \oplus K_6[1] \oplus K_7[8]$
(2) $P_L[7,18,24] \oplus C_H[7,18,24,29,30] \oplus C_L[15] \oplus F_8(C_L,K_8)[30] =$
$K_2[22] \oplus K_3[44] \oplus K_4[22] \oplus K_6[22] \oplus K_7[45] \oplus K_8[22]$
Wenn man nun davon ausgeht, dass es sich bei dem verschlüsselten Text um natürliches Englisch im ASCII-Code formatiert handelt, weiß man, dass $P_L[27]$ immer 0, und $P_L[7,18,24]$ in maximal 35% der Fälle 0 ist. Wenn man nun also diese Teile der Gleichungen eliminiert, erhält man Gleichungen, in denen kein Klartext vorkommt. Somit kann man dann nur aus dem Geheimtext den Schlüssel ermitteln.
Man benötigt hierfür jedoch min. 2^{29} Geheimtexte.
Bei 16-Runden-DES werden dann min. 2^{53} Geheimtexte benötigt.

2.6 Zusammenfassung

Die lineare Kryptanalyse ist eine sehr gute und schnelle Methode um den benutzten Schlüssel zur Verschlüsselung mit DES herauszufinden. Aber aufgrund dessen, dass sehr viele Geheimtexte und Klartexte benötigt werden um mit einer hohen Wahrscheinlichkeit zu dem richtigen Schlüssel zu kommen, ist nicht die Berechnung das Problem, sondern die Anzahl der Klar- und Geheimtexte. Um diese zu sammeln benötigt man wohl sehr viel Zeit. Deshalb ist es meist wohl einfacher DES mit Brute-Force anzugreifen, oder andere Implementierungsfehler von DES in der Software zu suchen.

3. Literatur

[1] http://homes.esat.kuleuven.be/~abiryuko/Cryptan/matsui_des.PDF
[2] http://de.wikipedia.org/wiki/Data_Encryption_Standard
[3] http://de.wikipedia.org/wiki/Kryptanalyse
[4] CrypTool