A Practical Attack on KeeLoq by Indesteege et al. [2008]

Damaris Schindler

Sommerakademie La Colle sur Loup

29. Sep 2009

Inhalt

1. Einleitung
2. Der KeeLoq Algorithmus
3. Die neue Attacke auf KeLoq
4. Komplexität und experimentelle Ergebnisse
5. Fazit

1. Einleitung

1.1 Was ist KeeLoq?

- KeeLoq ist eine Blockchiffre
- Blockbreite 32bit, Schlüssellänge 64 bit, Anzahl der Runden 528
- Rundenfunktion: Non Linear Feedback Shift Register (NLFSR)
- ist ein Zugangsberechtigungssystem
- derzeit hergestellt von Microchip Technology Inc.
- Verwendung: Funkschlüssel für Garagentore und Autos
- Autohersteller, die angeblich KeeLoq verwendet haben: Chrysler, Daewoo, Fiat, GM, Honda, Jaguar, Toyota, Volvo, Volkswagen, etc.
- Toyota Lexus verwendet nach eigener Aussage KeeLoq

1.2 Geschichte der Chiffre

- in den 1980ern von Willem Smit in Südafrika entwickelt
- 1995 an Microchip Technology Inc. für über 10 Millionen US Dollar verkauft
- beliebt, da relativ preiswert und energiesparend
- proprietäre Chiffre, d.h. Design wurde geheimgehalten
- November 2006: Spezifikation taucht auf russischer Hacker-Webseite auf

1.3 Geschichte der Angriffe

- Februar 2007: Bogdanov veröffentlicht Angriff, basierend auf einer slide attack und einer linearen Approximation der verwendeten Boolschen Funktion (Zeitkomplexität: 2^52 KeeLoq Verschlüsselungen, verwendet 2^32 Klartexte,16 GB Speicher)
- Mai 2007, 2008: Courtois et al.: verschiedene Angriffe, basierend auf algebraischen Methoden (Aufwand: 2^27 Verschlüsselungen, Erfolgswahrscheinlichkeit 44%)
- Rump Session Crypto 2007, Eurocrypt 2008: S. Indesteege et al.: A Practical Attack on KeeLoq

2. Der KeeLoq Algorithmus
2.1 Kurzcharakterisierung der Chiffre

- KeeLoq arbeitet mit einem 32-bit Block
- verwendet einen 64-bit Schlüssel
- Verschlüsselung geschieht durch 528 exakt gleiche Runden
- in jeder Runde wird ein bit verändert
- in jeder Runde wird nur ein Schlüsselbit verwendet

2.2 Der Angriff

- beteiligte Kryptographen: Sebastiaan Indesteege, Nathan Keller, Orr Dunkelman, Eli Biham, Bart Preneel
- verwendet 2^16 known plaintexts
- hat eine Zeitkomplexität von 2^44.5 KeeLoq Verschlüsselungen
- basiert auf einer slide attack und einer meet-in-the-middle attack
- 65 Minuten zum Sammeln der plaintext/ciphertext Paare benötigt
- 7.8 Tage Berechungen auf 64 CPU cores
- Attacke kann parallelisiert werden
- Variante mit 2^16 chosen plaintexts benötigt nur 3.4 Tage mit der selben Rechnerleistung
- 10 000 Euro würden genügen, um den Schlüssel in etwa 2 Tagen herauszufinden

2.3 Die KeeLoq Blockchiffre

- Eingabe in der i-ten Runde (0 <= i < 528):

$$Y^{(i)} = (y_{31}^{(i)}, \ldots, y_0^{(i)}) \in \{0,1\}^{32}$$

- Ausgabe in der i-ten Runde (0 < i <= 528):

$$Y^{(i+1)} = (\varphi^{(i)}, y_{31}^{(i)}, \ldots, y_1^{(i)})$$

- 64-bit Schlüssel:

$$K = (k_{63}, \ldots, k_0) \in \{0,1\}^{64}$$

- Eingabe in der Runde 0/ Ausgabe in Runde 528:

$$Y^{(0)} = P \qquad C = Y^{(528)}$$

2.4 Eine Runde KeeLoq Verschlüsselung

- Eine Runde kann wie folgt beschrieben werden:

$$\varphi^{(i)} = \mathrm{NLF}\left(y_{31}^{(i)}, y_{26}^{(i)}, y_{20}^{(i)}, y_{9}^{(i)}, y_{1}^{(i)}\right) \oplus y_{16}^{(i)} \oplus y_{0}^{(i)} \oplus k_{i \bmod 64} \ ,$$
$$Y^{(i+1)} = (\varphi^{(i)}, y_{31}^{(i)}, \ldots, y_{1}^{(i)}) \ .$$

mit der Boolschen Funktion

$$\begin{aligned}\mathrm{NLF}(x_4, x_3, x_2, x_1, x_0) = {} & x_4x_3x_2 \oplus x_4x_3x_1 \oplus x_4x_2x_0 \oplus x_4x_1x_0 \oplus \\ & x_4x_2 \oplus x_4x_0 \oplus x_3x_2 \oplus x_3x_0 \oplus x_2x_1 \oplus x_1x_0 \oplus \\ & x_1 \oplus x_0 \ .\end{aligned}$$

2.5 Non-linear feedback shift register (NLFSR)

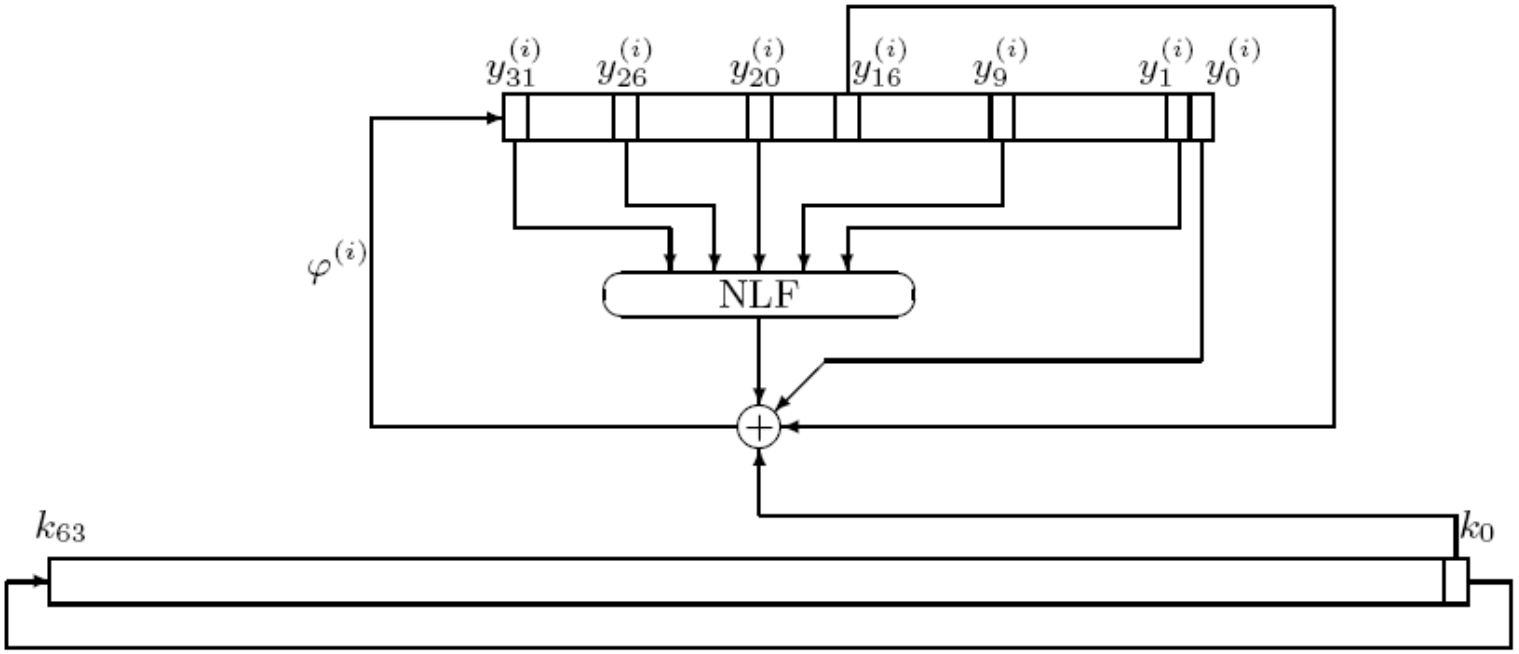

Fig. 1. The i-th KeeLoq encryption cycle

2.6 Entschlüsselung

- besteht aus ebenfalls 528 Runden mit inverser Funktion (i=528 bis i=1):

$$\theta^{(i)} = \mathrm{NLF}\left(y_{30}^{(i)}, y_{25}^{(i)}, y_{19}^{(i)}, y_{8}^{(i)}, y_{0}^{(i)}\right) \oplus y_{15}^{(i)} \oplus y_{31}^{(i)} \oplus k_{i-1 \bmod 64} \ ,$$
$$Y^{(i-1)} = (y_{30}^{(i)}, \ldots, y_{0}^{(i)}, \theta^{(i)}) \ .$$

3. Die neue Attacke auf KeeLoq
3.1 Die „Slide Property“

- eingeführt von Biryukov und Wagner 1999
- verwendet bei Blockchiffren, die aus einer großen Anzahl von Iterationen der selben Funktion bestehen; d.h. die Chiffre kann wie folgt geschrieben werden:

$$C = \underbrace{F\left(F\left(\ldots F(P)\right)\right)}_{r} = F^r(P) \quad .$$

- Eine „Slide Attacke“ nützt sie Selbstähnlichkeit aus
- Def: ein „slide pair“ ist ein Paar von Klartexten mit der Eigenschaft:

$$P_2 = F(P_1)$$

3.2 Slide Attacken

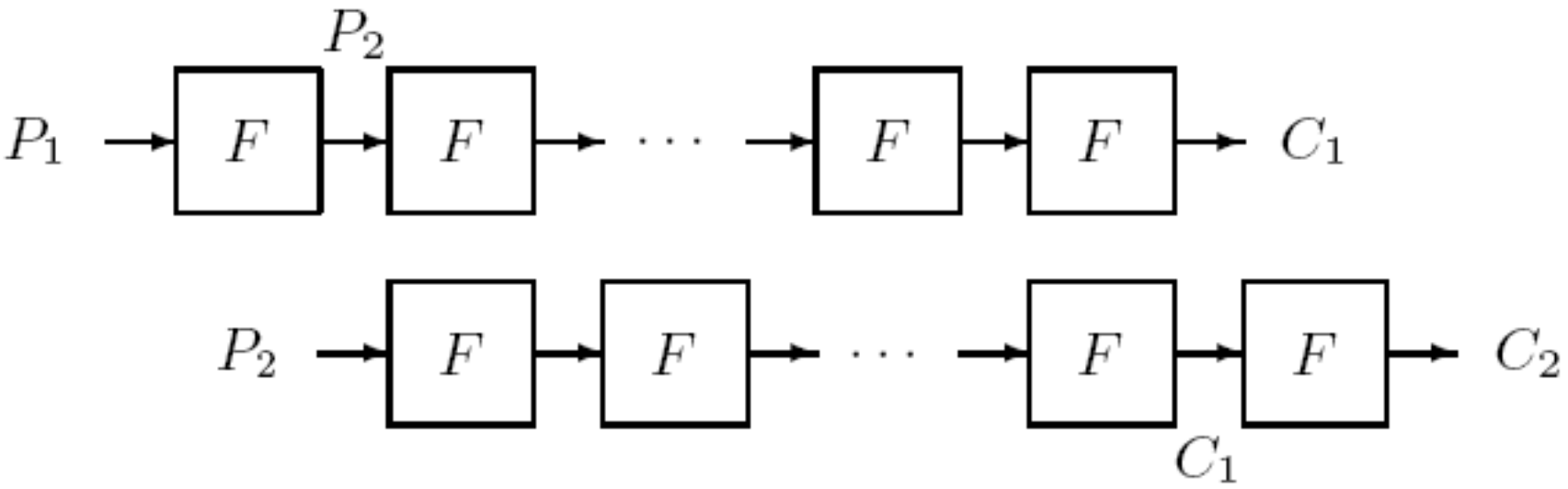

Fig. 2. A typical slide attack

Eigenschaft eines slid-pairs: $C_2 = F(C_1)$

3.2 Slide Attacken

- Idee: Verwende „slide pair“ um F zu entschlüsseln und damit den gesamten Code zu brechen

- Mit einem „slide pair“ können durch Verschlüsseln weitere erzeugt werden
- im Fall von KeeLoq: jeweils 64 Runden ergeben F
- Problem: 528 = 64*8 + 16 nicht durch 64 teilbar
- Lösungsidee: rate die 16 niederwertigsten bits des Schlüssels
- die Wahrscheinlichkeit, dass 2^16 Klartexte ein „slide pair“ enthalten ist in etwa 0.63
- Attacke wird mit jedem Paar ausgeführt, führt bei „slide pair“ zu Erfolg

3.3 Einfache Bestimmung von Schlüssel bits

- von Bogdanov als „linear step“ eingeführt
- Bestimmung von Schlüssel bits, wenn zwei Schritte der KeeLoq Verschlüsselung, die weniger als 32 Runden auseinander liegen bekannt sind
- Seien $Y^{(i)} = (y_{31}^{(i)}, \ldots, y_0^{(i)})$ und $Y^{(i+t)} = (y_{31}^{(i+t)}, \ldots, y_0^{(i+t)})$ zwei bekannte Zustände (t<=32)

3.3 Einfache Bestimmung von Schlüssel bits

- Idee: bei der Verschlüsselung von Y^(i) entsteht das neue Bit

$$\varphi^{(i)} = \mathrm{NLF}\left(y_{31}^{(i)}, y_{26}^{(i)}, y_{20}^{(i)}, y_{9}^{(i)}, y_{1}^{(i)}\right) \oplus y_{16}^{(i)} \oplus y_{0}^{(i)} \oplus k_{i \bmod 64} \,.$$

- phi^(i) ist aber in Y^(i+t) enthalten und damit bekannt
- folglich kann ein Schlüsselbit bestimmt werden:

$$k_{i \bmod 64} = \mathrm{NLF}\left(y_{31}^{(i)}, y_{26}^{(i)}, y_{20}^{(i)}, y_{9}^{(i)}, y_{1}^{(i)}\right) \oplus y_{16}^{(i)} \oplus y_{0}^{(i)} \oplus y_{32-t}^{(i+t)} \,.$$

3.4 Die Attacke - schematische Darstellung

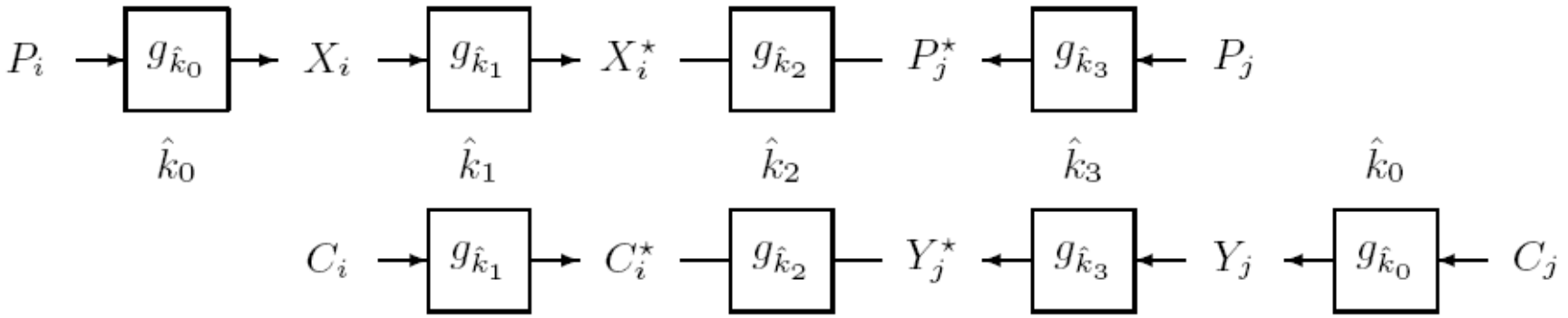

Fig. 3. The notation used in the attack

$\overline{X_i^\star}$, 16 most significant bits

$\underline{P_j^\star}$. 16 least significant bits

3.5 Die Attacke – verwendeter Algorithmus

for all $\hat{k}_0 \in \{0,1\}^{16}$ **do**
  for all plaintexts $P_i, 0 \leq i < 2^{16}$ **do**
    Partially encrypt P_i to X_i.
    Partially decrypt C_i to Y_i.
  for all $P_j^\star \in \{0,1\}^{16}$ **do**
    for all plaintexts $P_j, 0 \leq j < 2^{16}$ **do**
      Determine the key bits $\hat{k}_3$.
      Partially decrypt Y_j to $Y_j^\star$.
      Save the tuple $\left\langle P_j^\star, Y_i^\star, \hat{k}_3 \right\rangle$ in a table.

3.5 Die Attacke – verwendeter Algorithmus

for all plaintexts $P_i, 0 \leq i < 2^{16}$ **do**
  Determine the key bits $\hat{k}_1$.
  Partially encrypt C_i to $C_i^\star$.
  for all collisions $\overline{C_i^\star} = \underline{Y_j^\star}$ in the table **do**
    Determine the key bits $\hat{k}_2$ from $X_i^\star$ and $P_j^\star$.
    Determine the key bits $\hat{k}_2'$ from $C_i^\star$ and $Y_j^\star$.
    if $\hat{k}_2 = \hat{k}_2'$ **then**
      Encrypt 2 known plaintexts with the key $k = (\hat{k}_3, \hat{k}_2, \hat{k}_1, \hat{k}_0)$.
      if the correct ciphertexts are found **then**
        return success (the key is k)
return failure (i.e., there was no slid pair)

4. Komplexität und experimentelle Ergebnisse

- Komplexität (eine KeeLoq Verschlüsselung = eine Einheit)

$$2^{16}\left(32\cdot 2^{16}+2^{16}\left(32\cdot 2^{16}+2^{16}\left(32+N_{\mathrm{coll}}\cdot V\right)\right)\right)$$

- Attacke wurde implementiert und getestet, Resultat bei 16 bekannten key bits: known plaintext attack erfolgreich in 10.97 min, chosen plaintext attack in nur 4.79 min

5. Fazit

- Die vorgestellte Methode kann für die Attacke realer Systeme verwendet werden

Quellen

- A Practical Alltack on KeeLoq. S.Indesteege, N.Keller, O.Dunkelmann, E.Bilham, B.Preneel. 2008
- Algebraic an Slide Attacks on KeeLoq. N.T.Courtois, G.V.Bard, D.Wagnerm. Eurocrypt 2008
- Geknackter Code. Spiegel Online, April 2008
- www.heise.de/newsticker/meldung/, 31.03.2008