Übung 6

Abgabe: 06.12.2018 **bis** 12:15 Uhr

1. Reading Assignment & Moore's Law 40 Punkte

Lesen Sie den Artikel "Breaking Ciphers with COPACOBANA - A Cost-Optimized Parallel Code Breaker", der vom Lehrstuhl für Kommunikationssicherheit auf dem internationalen Workshop *CHES 2006* vorgestellt wurde. Es handelt sich hierbei um einen (leicht verständlichen) wissenschaftlichen Artikel, der Ihnen einen ersten Einblick in die Arbeit der Kryptogemeinde eröffnet.
Beantworten Sie in jeweils maximal **2-3 Sätzen** die folgenden Fragen:

a) Wie viele DES-Schlüssel konnten pro Sekunde auf einem einzelnen FPGA durchprobiert werden? (5 Pkt.)

b) Wie viele Tage dauerte eine gesamte DES-Schlüsselsuche im worst-case, wenn alle 120 FPGAs der COPACOBANA zur Verfügung standen? (5 Pkt.)

c) Die COPACOBANA wurde 2006 entwickelt. Wie viele Tage dauert die DES-Schlüsselsuche im Jahre 2019, wenn Sie Moore's Law (Verdopplung der Rechengeschwindigkeit alle 18 Monate) zugrunde legen? (5 Pkt.)

d) Worin bestehen die Vorteile, Algorithmen zur Kryptanalyse auf FPGAs anstatt in Software oder auf ASICs zu implementieren? (10 Pkt.)

e) Gegeben sei nun eine Blockchiffre mit einer Schlüssellänge von 128 Bits, wie z.B. AES. Angenommen wir schalten 15.000 COPACOBANAs parallel – wie lange dauert eine *durchschnittliche Suche* im Jahre 2019? Nehmen Sie an, dass Sie die COPACABANA mit aktuellen FPGAs, deren Leistung sich entsprechend zu Moore's Law entwickelt hat, bauen.
Hinweis: Nehmen Sie an, dass die COPACABANA AES-Schlüssel genauso schnell durchprobieren kann wie DES-Schlüssel. (10 Pkt.)

f) Wie viele Jahre müssen wir warten bis wir einen COPACABANA-Cluster bauen können, der einen symmetrischen Algorithmus mit 128 Bit in einer durchschnittlichen Suchzeit von 24 Stunden berechnen kann? Wir nehmen an, dass wir wieder 15.000 COPACABANAs verwenden, die in unserem Cluster parallel geschaltet sind. (5 Punkte)

2. Fehlerinjektionsangriff auf die DES-Implementierung einer Smartcard 60 Pkt

Durch einen erfolgreichen Fehlerinjektionsangriff auf die DES-Implementierung einer Smartcard können Sie den Algorithmus derart stören, dass er statt der vollen 16 Runden nur noch **eine einzige Runde** des DES durchläuft und das Ergebnis als Chiffrat ausgibt. Desweiteren manipulieren sie die DES-Implementierung so, dass die **initiale Permutation** IP und die **finale Permutation** IP^{-1} **nicht mehr durchlaufen** werden.

Sie schließen nun ein Smartcard-Lesegerät an ihrem Laptop an und führen ein Challenge-Response-Protokoll mit der auf diese Art gestörten Smartcard durch. Bei diesem Protokoll senden Sie einen 64-Bit-Klartext X (die sogenannte *Challenge*) an die Smartcard, und die Smartcard antwortet mit einem 64-Bit-Chiffrat Y (der sogenannten *Response*).

Ihr Ziel ist es, den geheimen Schlüssel k, welcher sicher auf der Smartcard abgespeichert ist, herauszufinden, um zukünftig auch ohne die Smartcard beliebige Challenge-Response-Paare selbst erzeugen zu können.

Zunächst erzeugen Sie ein einzelnes Klartext-Chiffrat-Paar (X_1, Y_1).

a) Beschreiben Sie Schritt für Schritt, wie sie vorgehen müssen, um Rückschlüsse auf den Rundenschlüssel k_1 zu ziehen. (15 Pkt.)

b) Auf wie viele Kandidaten (Möglichkeiten) können Sie den Rundenschlüssel k_1 mit nur einem Klartext-Chiffrat-Paar einschränken? (10 Pkt.)

Sie erzeugen nun ein weiteres Klartext-Chiffrat-Paar (X_2, Y_2). Die hexadezimalen Werte beider Paare lauten folgendermaßen:

$$X_1 = \texttt{0xF57DDDB12EAA799D}$$
$$Y_1 = \texttt{0x2EAA799D332924D8}$$
$$X_2 = \texttt{0x15837977D5202437}$$
$$Y_2 = \texttt{0xD5202437B5D8FC31}$$

c) Berechnen Sie den Rundenschlüssel k_1. (25 Pkt.)

d) Auf wie viele Kandidaten können Sie den Hauptschlüssel k durch die Kenntnis des Rundenschlüssels k_1 einschränken? Nennen Sie eine geeignete Methode um k zu bestimmen. (10 Pkt.)