Institut für Theoretische Informatik
Prof. Dr. Dennis Hofheinz
Übungsleiter: Thomas Agrikola

Stammvorlesung Sicherheit im Sommersemester 2017

Übungsblatt 2

Hinweis. Für die Vorlesung und Übung wurde eine Mailingliste eingerichtet. Wir werden diese Liste benutzen, um aktuelle Informationen zur Vorlesung/Übung zu verschicken, außerdem kann die Liste verwendet werden, um Fragen zu diskutieren. Alle weiteren Informationen zur Anmeldung etc. können hier nachgelesen werden:

`https://lists.ira.uni-karlsruhe.de/mailman/listinfo/sicherheit2017`

(Die Anmeldung ist freiwillig, erleichtert aber die Kommunikation)

Aufgabe 1. Aus der Vorlesung ist bekannt, dass eine Blockchiffre im CBC-Modus IND-CPA-sicher ist, wenn $\mathsf{E} : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$, für $k, \ell \in \mathbb{N}$, ununterscheidbar von einer echt zufälligen Funktion ist und der Initialisierungsvektor IV für jeden Verschlüsselungsvorgang neu gleichverteilt zufällig gezogen wird. Wir betrachten nun die folgenden zwei Fälle:

Fall 1: IV wird fest und für jeden Verschlüsselungsvorgang gleich gewählt,

Fall 2: IV wird, ausgehend von einer fixen Wahl, bei jedem Verschlüsselungvorgang um 1 hochgezählt (dabei wird nicht zwischen Verschlüsselungsvorgängen des Orakels und des Experiments unterschieden).

Geben Sie für diese beiden Fälle jeweils einen Angreifer an, der das IND-CPA-Spiel immer gewinnt.

Lösungsvorschlag zu Aufgabe 1. CBC-Modus (Erinnerung): Für einen Schlüssel K und eine Nachricht M aus n Blöcken $M_1, \ldots, M_n$ von jeweils ℓ Bit (dass die Länge von M ein Vielfaches der Blocklänge ℓ ist erreichen wir ggf. durch ein deterministisches Padding) gilt

$$Enc(K, M) = (C_0 := IV, C_1 := E(K, M_1 \oplus C_0), \ldots, C_n := E(K, M_n \oplus C_{n-1}))$$

Wir betrachten zuerst den **Fall 1**, in dem ein Initialisierungsvektor $IV \in \{0,1\}^\ell$ fest für alle Verschlüsselungen gewählt wird. Sei $\mathcal{A}$ ein Angreifer für das IND-CPA-Spiel. $\mathcal{A}$ agiert wie folgt:

1. $\mathcal{A}$ wählt zwei beliebige Nachrichten $M_0 \neq M_1$ gleicher Länge.
2. $\mathcal{A}$ schickt M_0, M_1 an das Experiment und erhält das Challenge-Chiffrat C^* (eine Verschlüsselung von M_0 oder M_1).
3. $\mathcal{A}$ schickt M_0 an das Verschlüsselungsorakel, dass ihm vom Experiment zur Verfügung gestellt wird, und erhält ein Chiffrat C.
4. Falls $C = C^*$ schickt rät $\mathcal{A}$ $b' := 0$ für das geheime Bit b des Experiments; ansonsten $b' := 1$.

Dieser Angreifer $\mathcal{A}$ gewinnt das IND-CPA-Spiel immer. Die Verschlüsselung ist deterministisch (da IV gleich für jeden Verschlüsselungsvorgang ist) so dass $\mathcal{A}$ mit Hilfe des Verschlüsselungsorakels testen kann, zu welcher Nachricht das Challenge-Chiffrat C^* gehört. *Ein deterministisches Verschlüsselungsverfahren kann also niemals IND-CPA-sicher sein.*

Betrachten wir nun **Fall 2**, in dem ein Initialisierungsvektor $IV \in \{0,1\}^\ell$ bei jeder neuen Verschlüsselung um 1 hochgezählt wird. Im Folgenden bezeichnen wir mit $IV_i = IV + i$ den Initialisierungsvektor im i-ten Verschlüsselungsvorgang.

1. $\mathcal{A}$ schickt$M_0 := 0^\ell$ an das Verschlüsselungsorakel und erhält $C = Enc(K, M_0) = (IV_1, E(K, M_0 \oplus IV_1$. Der Angreifer kennt nun also IV_1 und $E(K, IV_1)$ (da $M_0 \oplus IV_1 = 0^\ell \oplus IV_1 = IV_1$).
2. $\mathcal{A}$ berechnet $IV_2 = IV_1 + 1$, setzt $M_1 := IV_1 \oplus IV_2$, schickt M_0, M_1 an das Experiment und erhält C^*.
3. Falls $E(K, IV_1)$ die zweite Komponente von C^* ist, rät $\mathcal{A}$ $b' := 1$ für das geheime Bit b des Experiments; ansonsten $b' := 0$.

$\mathcal{A}$ gewinnt das IND-CPA-Spiel im Fall 2 mit Wahrscheinlichkeit 1. *Eine Blockchiffre im CBC-Modus mit einem durch Angreifer vorhersagbaren Initialisierungsvektor kann also nicht IND-CPA-sicher sein.*

Aufgabe 2. Es sei SKE ein IND-CPA-sicheres, symmetrisches Verschlüsselungsverfahren mit einem Verschlüsselungsalgorithmus Enc und einem Entschlüsselungsalgorithmus Dec. Wir verwenden die Algorithmen von SKE um zwei weitere Verschlüsselungsverfahren zu konstruieren:

(1.) Betrachten Sie das symmetrische Verschlüsselungsverfahren SKE' mit den folgenden Algorithmen:

- $\mathsf{Enc}'(K, M) := \mathsf{Enc}(K, \mathsf{Enc}(K, M))$,
- $\mathsf{Dec}'(K, C) := \mathsf{Dec}(K, \mathsf{Dec}(K, C))$.

(a) Zeigen Sie die Korrektheit von SKE', d.h. zeigen Sie, dass Dec' Chiffrate die mit Enc' erstellt wurden, korrekt entschlüsselt.

(b) Zeigen Sie, dass SKE' ebenfalls IND-CPA-sicher ist (Hinweis: Reduzieren Sie die Sicherheit von SKE' auf die Sicherheit von SKE. Nehmen Sie dazu zum Widerspruch an, dass ein Angreifer auf SKE' existiert und konstruieren Sie daraus einen Angreifer auf SKE).

(2.) Betrachten Sie das symmetrische Verschlüsselungsverfahren SKE^* mit den folgenden Algorithmen:

- $\mathsf{Enc}^*(K, M) := (M_{(0)}, \mathsf{Enc}(K, M))$ $(= (C_1, C_2))$,
- $\mathsf{Dec}^*(K, C) = \mathsf{Dec}^*(K, (C_1, C_2)) := \mathsf{Dec}(K, C_2)$.

Dabei bezeichne $M_{(0)}$ das niederwertigste Bit des Klartextes M.

(a) Zeigen Sie die Korrektheit von SKE^*.

(b) Zeigen Sie, dass SKE^* nicht IND-CPA-sicher ist.

Lösungsvorschlag zu Aufgabe 2.

(1.) (a) Sei M ein beliebiger Klartext. Verschlüsseln wir diesen mit dem Algorithmus Enc' von SKE', so erhalten wir als Chiffrat $C = \mathsf{Enc}'(K, M) = \mathsf{Enc}(K, \mathsf{Enc}(K, M))$. Entschlüsseln wir C mithilfe von Dec', so ergibt sich:

$$\begin{aligned}\mathsf{Dec}'(K, C) &= \mathsf{Dec}'(K, \mathsf{Enc}(K, \mathsf{Enc}(K, M))) = \mathsf{Dec}(K, \mathsf{Dec}(K, \mathsf{Enc}(K, \mathsf{Enc}(K, M))))\\ &= \mathsf{Dec}(K, \mathsf{Enc}(K, M)) = M.\end{aligned}$$

(b) Wir nehmen zum Widerspruch an, dass SKE' nicht IND-CPA-sicher ist. D.h. es existiert ein effizienter Angreifer $\mathcal{A}$, sodass

$$\Pr[\mathcal{A} \text{ gewinnt das IND-CPA-Spiel gegen } \mathsf{SKE}'] - 1/2$$

nicht vernachlässigbar ist. Wir konstruieren aus $\mathcal{A}$ einen Angreifer $\mathcal{B}$ auf die IND-CPA Sicherheit von SKE. $\mathcal{B}$ verwendet $\mathcal{A}$ als Subroutine, indem er für $\mathcal{A}$ das IND-CPA Spiel mit SKE' simuliert.

$\mathcal{B}$ verhält sich in seinem IND-CPA Spiel mit SKE folgendermaßen:

- $\mathcal{B}$ erhält vom Spiel Zugriff auf ein $\mathsf{Enc}(K, \cdot)$-Orakel und startet $\mathcal{A}$.

- $\mathcal{A}$ stellt im Laufe des Spiels Verschlüsselungsanfragen und erwartet Chiffrate, die von Enc' erstellt wurden. $\mathcal{B}$ muss diese Anfragen mithilfe seines $\mathsf{Enc}(K,\cdot)$-Orakels beantworten. Schickt $\mathcal{A}$ eine Anfrage für die Nachricht M, so stellt $\mathcal{B}$ eine Anfrage an sein Orakel mit der Nachricht M, erhält ein Chiffrat $C = \mathsf{Enc}(K, M)$, gibt C an sein Orakel und erhält

  $$C' = \mathsf{Enc}(K, C) = \mathsf{Enc}(K, \mathsf{Enc}(K, M)) = \mathsf{Enc}'(K, M).$$

  $\mathcal{B}$ gibt C' an $\mathcal{A}$ weiter und hat die Verschlüsselungsanfrage damit korrekt beantwortet.
- $\mathcal{A}$ gibt irgendwann zwei Nachrichten M_1, M_2 aus. $\mathcal{B}$ schickt nun zuerst jeweils M_1 und M_2 an sein $\mathsf{Enc}(K,\cdot)$-Orakel und erhält $C_1 = \mathsf{Enc}(K, M_1)$ und $C_2 = \mathsf{Enc}(K, M_2)$. $\mathcal{B}$ gibt nun C_1, C_2 als seine Challenge-Nachrichten aus.
- Das Spiel zieht ein Bit b zufällig, berechnet

  $$C^* = \mathsf{Enc}(K, C_b) = \mathsf{Enc}(K, \mathsf{Enc}(K, M_b)) = \mathsf{Enc}'(K, M_b)$$

  und gibt C^* an $\mathcal{B}$ aus.
- B gibt C^* an $\mathcal{A}$ weiter.
- Stellt $\mathcal{A}$ weitere Verschlüsselungsanfragen, so beantwortet $\mathcal{B}$ diese wie oben.
- $\mathcal{A}$ gibt irgendwann ein Bit b' aus. $\mathcal{B}$ gibt nun ebenfalls b' aus.

Da $\mathcal{B}$ alle Verschlüsselungsanfragen von $\mathcal{A}$ korrekt beantwortet und auch C^* ein für $\mathcal{A}$ korrekt erstelltes Challenge-Chiffrat ist, simuliert $\mathcal{B}$ das IND-CPA-Spiel mit SKE' für $\mathcal{A}$ perfekt. Da $\mathcal{B}$ einfach das Bit b' von $\mathcal{A}$ ausgibt, gilt:

$$\begin{aligned}\Pr[\mathcal{B} \text{ gewinnt im IND-CPA-Spiel gegen } \mathsf{SKE}] - 1/2 \\ = \Pr[\mathcal{A} \text{ gewinnt im IND-CPA-Spiel gegen } \mathsf{SKE}'] - 1/2,\end{aligned}$$

was nach Voraussetzung nicht vernachlässigbar ist. Da $\mathcal{B}$ ebenfalls effizient ist (die Laufzeit entspricht ungefähr der von $\mathcal{A}$, mit einem kleinen Overhead), haben wir einen effizienten und erfolgreichen IND-CPA Angreifer für SKE konstruiert. Dies steht im Widerspruch zur IND-CPA-Sicherheit von SKE. Somit kann $\mathcal{A}$ nicht existieren und wir haben die IND-CPA-Sicherheit von SKE' gezeigt.

Anmerkung: Eigentlich müsste man noch zeigen, dass im obigen Algorithmus auch $|C_1| = |C_2|$ gilt, da $\mathcal{B}$ diese als seine Challenge-Nachrichten ausgibt. Da bereits $|M_1| = |M_2|$, gilt dies aber sowieso mit überwältigender Wahrscheinlichkeit (Warum?).

(2.) (a) Sei M ein beliebiger Klartext. Verschlüsseln wir diesen mit dem Algorithmus Enc^* von SKE^*, so erhalten wir als Chiffrat $C = (C_1, C_2) = \mathsf{Enc}^*(K, M) = (M_{(0)}, \mathsf{Enc}(K, M))$. Entschlüsseln wir C mithilfe von Dec^*, so ergibt sich:

$$\mathsf{Dec}^*(K, C) = \mathsf{Dec}(K, C_2) = \mathsf{Dec}(K, \mathsf{Enc}(K, M)) = M.$$

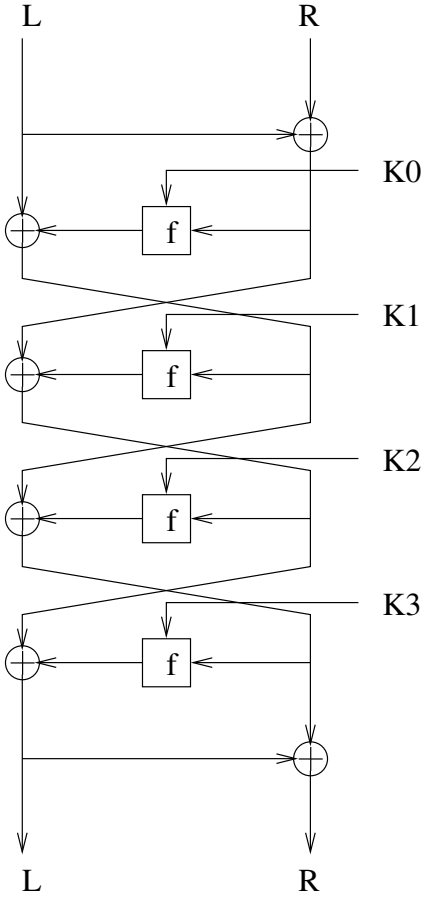
(b) Sei $\mathcal{A}$ ein Angreifer für das IND-CPA-Spiel. $\mathcal{A}$ agiert wie folgt:

- $\mathcal{A}$ wählt zwei Nachrichten ($M_0 \neq M_1$ gleicher Länge, sodass $M_{0,(0)} = 0$ und $M_{1,(0)} = 1$.
- $\mathcal{A}$ schickt M_0, M_1 an das Experiment und erhält das Challenge-Chiffrat $C^* = (C_1^*, C_2^*)$ (eine Verschlüsselung von M_0 oder M_1).
- $\mathcal{A}$ betrachtet C_1^* und gibt $b' = 0$ aus, wenn $C_1^* = 0$. Ansonsten gibt er $b' = 1$ aus.

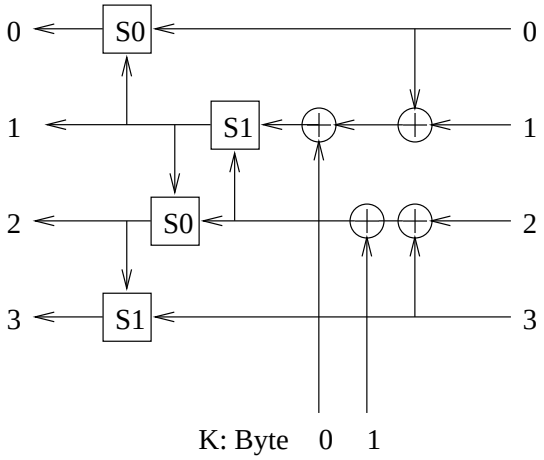
Dieser Angreifer $\mathcal{A}$ gewinnt das IND-CPA-Spiel immer, da C_1^* dem niederwertigsten Bit des verschlüsselten Klartextes entspricht. Somit kann $\mathcal{A}$ den verschlüsselten Klartext hier exakt bestimmen, da er seine beiden Challenge-Nachrichten so gewählt hat, dass sie sich im niederwertigsten Bit unterscheiden. Problematisch ist hier also, dass der Verschlüsselungsalgorithmus einen Teil der Nachricht im Klartext preisgibt, was für einen erfolgreichen Angriff ausreicht, selbst wenn es sich dabei nur um ein Bit handelt. Fazit: Damit ein Verschlüsselungsverfahren IND-CPA-sicher sein kann, darf es für Angreifer nicht möglich sein, aus einem Chiffrat irgendwelche Informationen über den verschlüsselten Klartext zu berechnen.

Aufgabe 3. Der ebenso brilliante wie einflussreiche Wissenschaftler und Superbösewicht Doktor Meta hat nach Jahren der Planung endlich **den** idiotensicheren Plan zur Übernahme der Weltherrschaft ausgearbeitet. Da er von der Ausarbeitung des Plans sehr erschöpft ist, macht er erstmal Urlaub und überlässt in der Zwischenzeit seinem mittelmäßig intelligenten Handlanger die Aufgabe seinen Plan verschlüsselt an seine Anhänger zu übermitteln. Da die Geheimhaltung des Plans essentiell für dessen Gelingen ist, hat Doktor Meta selbstverständlich bereits dafür gesorgt, dass er, sein Handlanger und alle seine Anhänger in Besitz eines gemeinsamen geheimen Schlüssels sind. Der Handlanger verwendet zur Verschlüsselung des Plans eine leichte Vereinfachung des symmetrischen Verfahrens FEAL-4 (siehe auch `http://de.wikipedia.org/wiki/FEAL`).

Helfen Sie uns die fiesen Weltherrschaftspläne Doktor Metas ans Licht zu bringen indem Sie das verwendete Verschlüsselungsverfahren durch lineare Kryptoanalyse brechen (und damit unseren Geheimdiensten ermöglichen, den Plan zu entschlüsseln und die Welt zu retten)!



Die f-Funktion und die S-Boxen ($S_i(a,b) := rot2((a+b+i) \mod 256)$ sind gegenüber dem Original nicht verändert:



Der Ausgangspunkt für eine lineare Approximationen der f-Funktion ist das jeweils letzte Bit der Addition der 4 S-Boxen, bei dem gegenüber allen anderen Bits kein Übertrag auftreten kann. Die ersten beiden S-Boxen liefern die Gleichungen:

- $F[2] = B[0] + F[8]$ und
- $F[10] = 1 + B[0] + B[8] + K[0] + B[16] + B[24] + K[8]$ (hier geht auch der Schlüssel mit ein!),

wobei $F[i]$ das i-te Ausgabebit von f ist; $B[i]$ und $K[i]$ entsprechend das i-te Eingabe- bzw. Schlüsselbit.

(a) Geben Sie die Gleichungen an, die die beiden anderen S-Boxen liefern.

(b) Geben Sie die 3-Runden Charakteristiken an, die sich durch die Gleichung zur ersten und dritten S-Box ergeben.

(c) Beschreiben Sie das Vorgehen der linearen Kryptoanalyse, das die vier Charakteristiken verwendet.

(d) Bei wievielen Klartext/Chiffrat-Paaren erwarten Sie einen Erfolg der Analyse?
Empirisch funktioniert der Angriff bei etwa 15 Paaren problemlos. Wie erklärt sich der Unterschied zu dem geschätzten Wert (falls er abweicht)?

Lösungsvorschlag zu Aufgabe 3. Bei der linearen Kryptoanalyse werden Teile einer Chiffre mit Hilfe sogenannter linearer Charakteristiken angenähert. Eine lineare Charakteristik gibt an, dass die Parität einer bestimmten Menge von Eingabebits und Ausgabebits mit einer bestimmten Wahrscheinlichkeit p gerade ist.

Wenn wir z.B. die im FEAL benutzte S-Box S0(A,B) betrachten, stellen wir fest, dass die Parität der Eingabebits A[0], B[0] und des Ausgabebits S[2] immer gerade ist. Oder bei der S-Box S1(A,B) ist die Parität der selben Bits (A[0], B[0], S[2]) immer ungerade.

Die Charakteristiken werden für die ersten drei Runden aufgestellt. Dann wird für den Teilschlüssel der letzten Runde vollständig gesucht nach Kandidaten, bei denen die linearen Approximationen korrekt auftreten. Das gleiche macht man mit einer Charakteristik für die letzten drei Runden mit dem Teilschlüssel der ersten Runde. Die verbleibenden Schlüsselbits kann man dann leicht mit vollständiger Suche finden.

(a) Die genannten zwei Beziehungen können verwendet werden, um lineare Approximationen für die gesamte Funktion $F := f(B, k)$ anzugeben:

- Für S_0 oben (Byteposition 0): $F[2] = B[0] + F[8]$ (d.h. gerade Parität zwischen den drei genannten Bits)
- Für S_1 an Byteposition 1: $F[10] = 1 + B[0] + B[8] + K[0] + B[16] + B[24] + K[8]$ (d.h. ungerade Parität zwischen den genannten Bits. Hier geht auch der Schlüssel mit ein!)
- Für S_0 an Byteposition 2: $F[18] = F[8] + B[16] + B[24] + K[8]$
- Für S_1 an Byteposition 3: $F[26] = 1 + F[16] + B[24]$.

(b) Diese 4 Charakteristiken für f können zu Charakteristiken für die ersten 3 Runden erweitert werden. Da alle beteiligten Charakteristiken deterministisch (d.h. Parität gerade entweder mit Wahrscheinlichkeit 0 oder 1) sind, ist die entstehende Charakteristik auch deterministisch.

Für die erste Gleichung ergibt sich, dass die Parität über die Eingabebits $L[0]$, $L[2]$, $L[8]$, $R[0]$, sowie die Ausgabebits nach der dritten Runde $R_3[2]$, $R_3[8]$ und $L_3[0]$ immer gleich 0 ist.

Die dritte Gleichung dehnt sich zu einer deterministischen Charakteristik aus: Die Parität über die Eingabebits $L[8]$, $L[16]$, $L[18]$, $L[24]$, $R[16]$, $R[24]$, sowie die Ausgabebits $R_3[8]$, $R_3[18]$, $L_3[16]$, $L_3[24]$ ist immer konstant (jedoch im Gegensatz zur ersten Drei-Runden-Charakteristik schlüsselabhängig).

(c) Dies kann genutzt werden, um jetzt eine vollständige Suche nach dem Teilschlüssel K_3 durchzuführen:

Für alle Möglichkeiten für K_3 werden die gesammelten Chiffrate teilweise entschlüsselt. Dann wird überprüft, ob für alle Paare (Klartext, teilweise entschlüsseltes Chiffrat) die angegebene Parität

gleich ist. Wenn nein, kann dieser Wert für K_3 ausgeschlossen werden. Wenn ja, ist ein Kandidat für K_3 gefunden (der mit Hilfe der anderen 3 deterministischen Charakteristiken genauer überprüft werden kann).

Wenn nach der Anwendung aller Charakteristiken für die ersten drei Runden einige Kandidaten für K_3 übrig sind, kann eine ähnliche Attacke gegen K_0 erfolgen: Nun werden die letzten drei Runden linear approximiert und für K_0 eine vollständige Suche durchgeführt.

K_1 und K_2 können dann aus den teilweise verschlüsselten Klartexten und den teilweise entschlüsselten Chiffraten berechnet werden.

(d) Angenommen, $K' \in \{0,1\}^{16}$ ist ein falsch geschätzter Rundenschlüssel. Dann ist die (idealisierte) Wahrscheinlichkeit, dass trotzdem eine Charakteristik für n Klartext-/Chiffrat-Paare konstant ist, $(1/2)^{n-1}$ (bzw. $(1/2)^n$, dass die Charakteristik immer den „richtigen“ Wert hat, falls die konkrete Parität nicht schlüsselabhängig und damit bekannt ist).

Damit ergibt sich folgende Abschätzung der Erfolgswahrscheinlichkeit eines wie geschildert durchgeführten Angriffs auf den *ersten* Rundenschlüssel von FEAL-4 mittels der zwei Charakteristiken aus Aufgabenteil (b):

$$\begin{aligned}
\Pr[\text{Angriff auf } K_0 \text{ erfolgreich}] &= \Pr\left[\begin{array}{l}\text{jedes falsche } K_0 \text{ erfüllt mind.}\\ \text{eine Charakteristik nicht}\end{array}\right] \\
&= \textstyle\prod_{K_0 \text{ falsch}} 1 - 2^{-2n-2} \\
&= (\underbrace{1 - 2^{-2n-2}}_{\leq 1})^{2^{16}-1} \\
&\geq (1 \underbrace{-2^{-2n-2}}_{\geq -1})^{2^{16}} \\
&\overset{\text{Bernoulli}}{\geq} 1 - 2^{-2n-2} \cdot 2^{16} = 1 - 2^{14-2n}
\end{aligned}$$

Hierbei wurde die Ungleichung von Bernoulli benutzt. Analog ergibt sich, dass die Erfolgswahrscheinlichkeit für einen Angriff auf den *vierten* Rundenschlüssel mindestens $1 - 2^{14-2n}$ ist. (Sind erster und vierter Rundenschlüssel gefunden, können die restlichen Rundenschlüssel durch zwei vollständige Suchen der Größe jeweils 2^{16} bestimmt werden.) Außerdem gilt:

$$\begin{aligned}
\Pr[\text{Angriff nicht erfolgreich}] &= \Pr[\neg E_1 \vee \neg E_4] \\
&\overset{\text{Union Bound}}{\leq} \Pr[\neg E_1] + \Pr[\neg E_4] \\
&\leq 2 \cdot (2^{14-2n})
\end{aligned}$$

Damit ist die gesamte Erfolgswahrscheinlichkeit für diesen Angriff $\geq 1 - 2^{15-2n}$. Insgesamt ist der Angriff auf beide Rundenschlüssel schon für $n = 12$ mit einer Wahrscheinlichkeit von $> 99\%$ erfolgreich. Dies deckt sich in etwa mit der in der Aufgabenstellung angegeben Größe $n = 15$.

Aufgabe 4. Symmetrische Verschlüsselungsverfahren in der Praxis: Wählen Sie drei (verbreitete) kryptographische Protokolle bzw. drei Programme aus, in denen symmetrische Verschlüsselungsverfahren zum Einsatz kommen. Welche Chiffren werden jeweils verwendet? (Bei Blockchiffren: Welche Modi werden benutzt? Wie wird ggf. der Initialisierungsvektor gewählt?) Woher kommt der verwendete Schlüssel?

Lösungsvorschlag zu Aufgabe 4.

- Wie bei Aufgabe 2 erwähnt findet der XTS-Modus, unter anderem mit AES als “innere” Blockchiffre, im “OS X Mountain Lion“-Betriebssystem (in der ”FileVault 2“-Applikation) und in der Software TrueCrypt Anwendung. Zur Ableitung des Schlüssels wird bei beiden Anwendungen eine *Password-Based Key Derivation Function* (PBKDF) eingesetzt.
- Im TLS-Protokoll werden u.a. AES und 3DES im CBC-Modus verwendet. Der Schlüssel wird aus einer vorangehenden Schlüsselaustauschphase abgeleitet. Bis TLS 1.0 wurde als Initialisierungsvektor im CBC-Modus der letzte Chiffratblock des vorhergegangen Verschlüsselungsvorgangs. Dies führte 2011 zum sogenannten BEAST-Anriff auf TLS 1.0. (Das Vorgehen des Angreifers bei BEAST ist im Prinzip übrigens das des Angreifers im zweiten Teil von Aufgabe 1.). Seit TLS 1.1 wird der IV im ersten Block verschlüsselt.

- GnuPG, eine Implementierung von PGP, verwendet für symmetrische Verschlüsselung z.B. die Blockchiffren AES, 3DES, Twofish, Blowfish und CAST5 im CFB-Modus (wurde in der Vorlesung nicht besprochen). Der Schlüssel wird bei jedem Vorgang frisch zufällig gezogen. Der Initialisierungsvektor wird unter Einsatz des Schlüssels aus einer Art Checksumme abgeleitet.