Bedingungslose Sicherheit von Verschlüsselungssystemen

Nikolas Vogt

Im Rahmen des Kryptografie-Seminars im WS 05/06
bei Herrn Prof. Kreuzer
an der Universität Dortmund

1 Einleitung

Die Sicherheit von vielen aktuell benutzten Verschlüsselungssystemen, insbesondere *„public-key“*, basiert auf der Schwierigkeit einer rechnerischen Aufgabe, wie das Faktorisieren von ganzen Zahlen oder das Berechnen von diskreten Logarithmen. Die Beweise für die Sicherheit dieser Systeme zeigen, dass die Schwierigkeit der rechnerischen Aufgabe entscheidend für die Fähigkeit des Gegner ist, einen erfolgreichen Angriff durchzuführen. Man kann beweisen, dass öffentlich zugängliche *„one-time pads“*, welche durch Einweg-Permutationen mit einem *„hard-core“* Prädikat erzeugt werden, rechnerisch sicher sind. Das *„hard-core“* Prädikat einer Einwegfunktion $f(x)$ ist eine Eigenschaft $b(x)$, welche man sehr leicht aus x, aber nur sehr schwer aus $f(x)$ berechnen kann.
Die Sicherheit der Verschlüsselung ist größtenteils auf die Familien der Einwegfunktionen beschränkt, wie z.B. RSA oder modulares Wurzelziehen - die Einwegeigenschaft dieser Familien basiert auf der angenommenen Schwere des Ziehens der e-ten Wurzel oder des Faktorisierens großer Zahlen. Die Sicherheit dieser Algorithmen ist nur bedingt, denn es besteht ein Risiko, dass in Zukunft diese mathematischen Probleme gelöst werden.
Auf der anderen Seite bietet Shannons informationstheoretische Modell der Sicherheit bedingungslose Sicherheit. Die perfekte Sicherheit von *Vernam's „one-time pad“* (siehe letzter Vortrag) hängt nicht von der Schwere eines mathematischen Problems oder der Beschränkung der Leistungsfähigkeit des Rechners des Angreifers ab.
Obgleich perfekte Sicherheit in den meisten praktischen Situationen nicht zu erreichen ist, existieren verschiedene Versuche praktisch umsetzbare Verschlüsselungssysteme zu kreieren, deren Sicherheit nicht auf Vermutungen basieren und welche beweisbar nah an perfekte Informationsssicherheit kommen.
Ein Beispiel dafür ist die von Bennett und Brassard eingeführte Quantenkryptologie. Zwei Parteien verständigen sich auf einen Sicherheitsschlüssel per übertragene polarisierte Photonen über einen Fiberglaskanal. Die Sicherheit dieses Verfahrens basiert auf der Unschärferelation der Quantenmeschanik.
Ein anderes Beispiel für die bedingungslose Sicherheit von Verschlüsselungssystemen beruht auf der Grundlage, dass Kommunikationskanäle verrauscht sind, und daher ein Angreifer nie alle Informationen bekommen kann (vgl. Abschnitt 3), oder auf der begrenzten Speicherkapazität des Angreifers (vgl. Abschnitt 2).

2 Das Modell des beschränkten Speichers

Wir werden eine kurze Einführung in das von U.M.Maurer entworfene Verschlüsselungsschema geben, welches eine bedingungslose Sicherheit aufgrund der begrenzt verfügbaren Speicherkapazität des Angreifers garantiert.

Das Modell:
Alice will eine Nachricht $m \in M := \{0,1\}^n$ zu Bob schicken. Sie benutzt einen *„one-time pad“* zur Verschlüsselung, z.B. durch bitweise XOR-Verknüpfung. Gewöhnlich haben wir eine Wahrscheinlichkeitsverteilung auf M welche, zusammen mit der Wahrscheinlichkeitsauswahl des Schlüssels, eine Wahrscheinlichkeitsverteilung auf dem Raum der Geheimtexte C liefert. Ohne Beschränkung der Allgemeinheit nehmen wir an, dass $prob\,(m) > 0 \,\forall\, m \in M$ und $prob\,(c) > 0 \,\forall\, c \in C$. Die Lauscherin Eve ist eine passive Angreiferin, d.h. sie beobachtet nur den Geheimtext $c \in C \subseteq \{0,1\}^n$ und versucht dadurch Informationen über den Klartext zu bekommen.

Das Satelliten-Szenario:
Alice und Bob extrahieren den Schlüssel k für die Verschlüsselung der Nachricht m von öffentlich zugänglichen zufälligen Bits. Ein mögliches Szenario ist das Satelliten-Szenario, dabei wird die zufällige Bitfolge von einer Radio-Quelle (z.B. ein Satellit oder eine natürliche Strahlungsquelle) mit einer sehr hohen Datenrate (siehe Beispiel unten) ausgesandt. Eve, die Angreiferin, kann nur einen Teil dieser Bits speichern.
Also nehmen wir an, dass die öffentlich zugängliche Quelle zufällige (gleichverteilte) Bits an Alice, Bob und Eve mit einer sehr hohen Datenrate sendet. Zusätzlich nehmen wir an, dass die Kommunikationswege fehlerfrei sind.
Alice und Bob wählen ihre Schlüsselbits von dem Datenstrom über die gleiche Zeitperiode T, entsprechend einer privaten Strategie, welche Eve nicht bekannt ist. Der Geheimtext wird später übertragen, nach Ende der Zeit T. Aufgrund ihrer beschränkten Speicherkapazität kann Eve nur einen kleinen Teil der gesendeten Bits während der Zeit T speichern.
Um den Einmalschlüssel k für den Nachricht $m \in \{0,1\}^n$ zu extrahieren, müssen Alice und Bob zeitgleich über dieselbe Zeitperiode T von der Quelle zufällige Bits speichern. Von nun an werden wir mit R (für *„randomizer“*, Zufallsgenerator) die während der Zeitperiode T übertragenen zufälligen Bits bezeichnen. Die übertragenen Bits werden wir als Elemente einer Matrix $(r_{i,j} | 1 \leq i \leq l, 0 \leq j \leq t-1)$ mit l Zeilen und t Spalten auffassen. Also besteht R aus $|R| = lt$ Bits. Meistens ist l sehr klein (z.B. 50) und t sehr groß, wenn man es mit der Länge n der Nachricht m vergleicht.
Alice und Bob müssen sich im Vorhinein auf einen (kurzen) privaten Schlüssel $(s_1, ..., s_l)$ einigen, wobei die s_i zufällig aus $\{0, ..., t-1\}$ (unter Berücksichtigung der Gleichverteilung) ausgewählt und benutzt werden für den Schlüssel $k := k_0 ... k_{n-1}$ mit

$$k_j := r_{1,(s_1+j) mod\, t} \oplus r_{2,(s_2+j) mod\, t} \oplus ... \oplus r_{l,(s_l+j) mod\, t}.$$

Mit anderen Worten, Alice und Bob wählen aus jeder Reihe i die Bitfolge b_i der Länge n aus, angefangen bei einer zufällig ausgewählten „Saat“ s_i (wenn das Ende der Reihe erreicht ist, springen sie zurück zum Anfang) und bekommen ihren Schlüssel k, indem sie diese b_i mit XOR verknüpfen.

Der Angriff:
Die Angreiferin Eve empfängt die zufälligen Bits ebenfalls während der Zeit T und speichert einige von ihnen, in der Hoffnung, dass dies ihr helfen werden Informationen aus der übertragenen Nachricht am Ende der Zeitperiode T zu erhalten. Aufgrund ihrer begrenzten Speichermöglichkeit kann sie nur q der Bits $r_{i,j}$ speichern. Die Bits werden mit Hilfe eines Wahrscheinlichkeitsalgorithmus ausgewählt. Bei jedem dieser Bits weiß Eve dessen Wert und die wahrscheinliche Position in der Bitfolge R. Eves Wissen über die zufälligen Bits ist zusammengefasst in der Zufallsvariable S. Man kann S ebenso als Wahrscheinlichkeits-Algorithmus auffassen, welcher q Positionen und Werte zurückgibt. Wie üblich

beim Studieren des *„one-time pad“* Verschlüsselungssystemes, kennt Eve die Verteilung von M. Wir nehmen an, dass sie kein vorheriges Wissen über die Nachricht $m \in M$, welche Alice an Bob schickt, hat.

2.1 Theorem
Es existiert ein Ereignis E, so dass für alle Wahrscheinlichkeitsstrategien S zum Speichern der q Bits des Zufallsgenerators R gilt:

$$I(M; CS|E) = 0 \text{ und } prob(E) \geq 1 - n\delta^l.$$

Wobei $\delta := \frac{q}{|R|}$ der Teil der gespeicherten Bits von Eve ist.

Zum Beweis wird folgendes Theorem aus dem letzten Vortrag benötigt:

2.2 Theorem
Es sei E ein zufälliger Verschlüsselungsalgorithmus mit deterministischer Erweiterung $E_D : M \times K \rightarrow C$. Jedes Mal, wenn eine Nachricht $m \in M$ verschlüsselt wird, wird ein neuer Einmalschlüssel k nach irgendeiner Wahrscheinlichkeitsverteilung auf K unabhängig von m gewählt. Wir nehmen an, dass der Klartext m aus dem Geheimtext c und dem Einmalschlüssel k zurückgewonnen werden kann (dass man also keine weiteren Informationen dafür benötigt). Dann kann die Entropie des Schlüssels nicht kleiner sein, als die der Nachrichten:

$$H(K) \geq H(M).$$

Beweisskizze (Theorem 2.1).
Sei $(s_1, ..., s_l)$ der private Schlüssel und $k = k_0...k_{n-1}$ der von Alice und Bob aus R extrahierte Schlüssel. k_j wird aus R gewonnen durch bitweises XORen der Bits $r_{1,(s_1+j) mod t}, r_{2,(s_2+j) mod t}, ...,$ $r_{l,(s_l+j) mod t}$. Wenn Eves Speicherstrategie nur ein Bit von diesen nicht speichert, dann wirkt das resultierende Bit k_j auf sie wirklich zufällig, trotz ihres Wissens S über R. Die Wahrscheinlichkeit des Ereignisses F, dass Eve mit ihrer Speicherstrategie alle Bits für mindestens ein j speichert, ist sehr klein $\left(\leq n \cdot \delta^l\right)$. E ist das Komplement von F. Tritt E ein, so ist aus Eves Sicht der Schlüssel wirklich zufällig, d.h. wir haben die Situation eines eines *„Vernam's one-time pad“*, wodurch die gemeinsame Information gleich Null ist (siehe Theorem 2.2). □

2.3 Bemerkungen

1. Die gemeinsame Information in Theorem 2.1 ist bedingt durch das Ereignis E. Das bedeutet, dass alle beteiligten Entropien (Unsicherheiten) mit der bedingten Wahrscheinlichkeit unter E errechnet werden.

2. U.M. Maurer beweist in seinem Paper *„Conditionally-perfect secrecy and a provably-secure randomized cipher“* eine stärkere Version. Sie beinhaltet den Fall, dass Eve vorher schon ein Wissen V über den Klartext hat, wobei V gleichverteilt ist auf M. Dann ist die gemeinsame Information $I(M; CS|V, E)$ zwischen M und CS, unter den Bedingungen V und E, gleich 0. Bedingt von V bedeutet, dass die gemeinsame Information nicht die Menge von Information über M beinhaltet resultierend von dem Wissen über V.

3. Die Angreiferin Eve kann kein Vorteil durch das Erfahren des Sicherheitsschlüssels $(s_1, ..., s_l)$ nach dem Aussenden des Zufallsgenerators R beziehen.

4. Das Modell des Angriffs, welches hier vorgestellt wird, ist etwas eingeschränkt. In der ersten Phase, während des Empfangs der Zufallsquelle, benutzt die Lauscherin Eve nicht ihre volle Rechenleistung, sie speichert nur ein paar der übertragenden Bits und benutzt nicht den Bit-Stream als Input für die Berechnung zu diesem Zeitpunkt. Das übliche Modell des Angriffs, bei dem Eve beliebige Bits an Information errechnet und speichert, wird weiter unten betrachtet.

2.4 Beispiel
Ein mit einer Datenrate von 16 Gbit/s sendender Satellit, wird für einen Tag benutzt, um eine zufällige Bitfolge R von ungefähr $1,5 \cdot 10^{15}$ Bits zu erstellen. Sei R angeordnet in $l := 100$ Zeilen und $t := 1,5 \cdot 10^{13}$ Spalten. Der Klartext soll 6 MB groß sein, also ungefähr $n = 5 \cdot 10^7$ Bits. Alice und Bob müssen sich auf einen privaten Schlüssel $(s_1, ..., s_l)$ der Länge $100 \cdot log_2(1,5 \cdot 10^{13})) \approx 4380$ Bits einigen. Die Speicherkapazität der Angreiferin Eve wird mit 100 TB angenommen, was ungefähr $8,8 \cdot 10^{14}$ Bits entspricht. Dann ist $\delta \approx 0,587$ und

$$prob(notE) \leq 5 \cdot 10^7 \cdot 0,587^{100} \approx 3,7 \cdot 10^{-16} < 10^{-15} \ .$$

Also: Die Wahrscheinlichkeit, dass Eve irgendwelche zusätzlichen Informationen über den Klartext durch Beobachtung den Geheimtextes erfährt unter Verwendung einer optimalen Speicherstrategie ist kleiner als 10^{-15}.

2.5 Lemma
Seien A, B, E Teilmengen des Wahrscheinlichkeitsraums X, mit $prob\,(E) > 0$. Angenommen A und B haben die gleiche bedingte Wahrscheinlichkeit unter E, d.h. $prob\,(A|E) = prob\,(B|E)$. Dann gilt:

$$|prob\,(A) - prob\,(B)\,| \leq prob\,(X \setminus E).$$

Beweis.
Nach der Formel der totalen Wahrscheinlichkeit gilt:
$|prob\,(A) - prob\,(B)\,| = |prob\,(E) \cdot prob\,(A|E) + prob\,(X \setminus E) \cdot prob\,(A|X \setminus E) - prob\,(E) \cdot prob\,(B|E) - prob\,(X \setminus E) \cdot prob\,(B|X \setminus E)\,| = |prob\,(X \setminus E) \cdot (prob\,(A|X \setminus E) - prob\,(B|X \setminus E))\,| \leq prob\,(X \setminus E)$
□

2.6 Satz
Seien X und Y endliche Wahrscheinlichkeitsräume mit gemeinsamer Verteilung XY. Dann gilt:

1. $H(X|Y) \geq 0$.
2. $H(XY) = H(X) + H(Y|X)$.
3. $H(XY) \leq H(X) + H(Y)$.
4. $H(Y) \geq H(Y|X)$.
5. $I(X;Y) = I(Y;X) = H(X) + H(Y) - H(XY)$.
6. $I(X;Y) \geq 0$.

Beweis:
(letzter Vortrag)

2.7 Satz
E ist genau dann perfekt sicher, wenn für jeden wahrscheinlichkeitstheoretischen Entscheidungsalgorithmus A und alle $m_0, m_1 \in M$ mit $m_0 \neq m_1$ gilt:

$$prob\left(A\left(m_0, m_1, c\right) = m : m \overset{u}{\leftarrow} \left\{m_0, m_1\right\}, c \leftarrow E\left(m\right)\right) = \tfrac{1}{2}.$$

Beweis:
(letzter Vortrag)

2.8 Theorem
Für jede wahrscheinlichkeitstheoretische Strategie S zum Speichern eines Teils δ von allen zufälligen Bits, und jedem wahrscheinlichkeitstheoretischen Entscheidungsalgorithmus $A\left(m_0, m_1, c, s\right)$ und alle $m_0, m_1 \in M$ mit $m_0 \neq m_1$, gilt:

$$prob(A(m_0, m_1, c, s) = m : m \overset{u}{\leftarrow} \left\{m_0, m_1\right\}, c \leftarrow E(m), s \leftarrow S) \leq \tfrac{1}{2} + n\delta^l.$$

Bemerkung:
Die oben genannte Formel ist äquivalent zu:
$|prob(A(m_0, m_1, c, s) = m_0 : c \leftarrow E(m_0), s \leftarrow S) -$
$prob(A(m_0, m_1, c, s) = m_0 : c \leftarrow E(m_1), s \leftarrow S)| \leq n\delta^l$.

Beweisskizze (Theorem 2.8).
Aus Theorem 2.1 und Satz 2.6 folgt:

$$prob\left(c, s | m_0, E\right) = prob\left(c, s | m_1, E\right)$$

für alle $m_0, m_1 \in M$, $c \in C$ und $s \in S$.
Errechnet mit der bedingten Wahrscheinlichkeit unter E erhält man:
$prob\left(A(m_0, m_1, c, s) = m_0 | E : c \leftarrow E(m_0), s \leftarrow S\right) =$
$= \sum_{c,s} prob(c, s | m_0, E) \cdot prob(A(m_0, m_1, c, s) = m_0 | E)$
$= \sum_{c,s} prob(c, s | m_1, E) \cdot prob(A(m_0, m_1, c, s) = m_0 | E)$
$= prob\left(A(m_0, m_1, c, s) = m_0 | E : c \leftarrow E(m_1), s \leftarrow S\right)$
Mit Hilfe von Lemma 2.5 folgt:
$|prob(A(m_0, m_1, c, s) = m_0 : c \leftarrow E(m_0), s \leftarrow S) -$
$prob(A(m_0, m_1, c, s) = m_0 : c \leftarrow E(m_1), s \leftarrow S)| \leq prob(notE) \leq n\delta^l$ □

2.9 Bemerkung
Wie wir oben beobachtet haben, ist das Modell des Angriffs etwas eingeschränkt, da die Angreiferin Eve den Bitstream als Input für die Berechnung in der ersten Phase, während des Empfangs von der zufälligen Quelle, nicht benutzt. Der erste Beweis der Sicherheit für das Modell des generellen Angriffs, wobei Eve ihre unbegrenzte Rechenleistung zu jedem Zeitpunkt ohne Einschränkung nutzt, steht in einem Paper von C.Cachin und U.M. Maurer *„Unconditional security against memorybounded adversaries“*. Um die dort verwendeten Techniken zu veranschaulichen, werden wir einen kurzen Überblick über Teile von dem Paper geben (einige dieser Techniken werden im Modell des verrauschten Kanals (Abschnitt 3) angewandt).
Wie zuvor gibt es eine Quelle wirklich zufälliger Bits. Alice, Bob und Eve empfangen diese Bits über perfekte, fehlerfreie Kanäle. Wir durchsuchen den Bitstream nach Sequenzen der Länge n zur

Nutzung als *„one-time pad“* zur Verschlüsselung einer Nachricht $m \in \{0,1\}^n$. Die zufällige Bitquelle generiert N Bits, welche wir mit $R := (r_1, ..., r_N)$ bezeichnen. Die Speicherkapazität q von Eve ist kleiner als N, so dass sie nicht in der Lage ist, die komplette zufällige Bitfolge zu speichern. Im Gegensatz zu dem vorhergegangenen Modell, kann sie nicht nur q Bits von dem Zufallsgenerator speichern, sondern auch einige wahrscheinlichkeitstheoretische Algorithmen U ausführen, während sie die zufällige Quelle empfängt, um die q Bits an Information von R zu berechnen und zu speichern. Wie zuvor bezeichnen wir mit $\delta := \frac{q}{N}$ den Anteil von Eves Speicherkapazität unter Berücksichtigung der kompletten Anzahl der zufälligen Bits. Die Schlüsselgenerierung erfolgt in zwei Phasen:

1. *„advantage destillation“*:
In der ersten Phase, genannt *„advantage distillation“*, extrahieren Alice und Bob genügend viele, z.B. l Bits $S := (s_1, ..., s_l)$ von R an zufällig ausgewählten Positionen $P := (p_1, ..., p_l)$:

$$s_1 := r_{p_1}, s_2 := r_{p_2}, ..., s_l := r_{p_l}.$$

Sie haben sich vorher auf diese Positionen geeinigt und halten diese geheim. Da Eve nur q Bits speichern kann, ist ihre Information über S unvollständig. Zum Beispiel kann man beweisen, dass Eve höchstens den Anteil δ von den l Bits in S (im informationstheoretischen Sinn) wissen kann. Sei e der Integerteil von Eves „Unwissen“ H über S. Dann fehlen Eve ungefähr e Bits an Information über S.

2. *„privacy amplification“*:
In der zweiten Phase wenden Alice und Bob eine Technik an, genannt *„privacy amplification“* oder *„entropy smoothing“*, um e Bits von S zu extrahieren, so dass Eve so gut wie keine Information über die resultierende Zeichenkette $\tilde{S}$ hat. Eves Unwissen über $\tilde{S}$ ist nah an e, so dass von Eves Sicht $\tilde{S}$ wirklich zufällig erscheint. Dadurch kann es Alice und Bob als beweisbar sicherer Schlüssel k in Form eines *„one-time pads“* dienen.
„Privacy amplification“ wird durchgeführt durch die zufällige Auswahl eines Vertreters der sogenannten universellen Klasse von Hash-Funktionen (siehe unten). Alice wählt zufällig ein Element h von so einer universellen Klasse H (unter Berücksichtigung der Gleichverteilung) aus und sendet h über einen öffentlichen Kanal zu Bob. Dadurch kann Eve sogar H und h wissen. Alice und Bob wenden beide h auf S an um ihren Schlüssel $k := h(S)$ als *„one-time pad“* zu gewinnen. In Abschnitt 3 werden wir näher auf diese Phase eingehen.
Seien H und K die Zufallsvariablen, welche die wahrscheinlichkeitstheoretische Auswahl der Funktion h und der Schlüssels k beschreiben. Das folgende Theorem wird in dem oben genannten Paper von Cachin und Maurer bewiesen.

2.10 Theorem
Gegeben ist eine feste Speicherkapazität von Eve und ϵ_1,$\epsilon_2 > 0$, dann existiert ein Sicherheitsereignis E, so dass:

$$prob\,(E) \geq 1 - \epsilon_1 \text{ und } I\,(K; H|U = u, P = p, E) \leq \epsilon_2,$$

und daher insbesondere

$$I\,(K; UHP|E) \leq \epsilon_1 \text{ und } (I(K; UH|E) \leq \epsilon_2,$$

vorausgesetzt die Größe $|R|$ des Zufallsgenerators und die Anzahl l der ausgewählten Elemente aus R von S sind genügend groß (und dadurch der Anteil von Eves Wissen über R genügend klein).

Beweis:
(siehe Paper von Cachin und Maurer)

2.11 Bemerkungen

1. Explizite Formeln verbunden mit den Schranken ϵ, $|R|$ und l werden in dem Paper von Cachin und Maurer hergeleitet.

2. $I\left(K; H|U=u, P=p, E\right) \leq \epsilon_2$ bedeutet das Folgende: Vermutlich hat Eve das spezielle Wissen $U=u$ über den Zufallsgenerator R und hat die Positionen P der von Alice und Bob nach der Übertragung des Zufallsgenerators R ausgewählten Bits gelernt. Dann ist der durchschnittliche Inhalt an Information (gemessen in Bits in dem Informationstheoritschen Sinn) dass Eve den Schlüssel k vom Studieren der Hash-Funktion h ableiten kann, geringer als ϵ_2, vorausgesetzt das Sicherheitsereignis E tritt auf.

Wie wir schon vorher festgestellt haben, ist ein Schlüsselschritt in dem Beweis von Theorem 2.10 die *„privacy amplification“* um fast die ganze Entropie eines Bitstrings in einem zufälligen Bitstring zu transformieren. Für diesen Zweck reicht es nicht aus, mit der klassischen Shannon Entropie zu arbeiten. Stattdessen ist es nötig, mit einem allgemeineren Informationsmaßstab: die *„Rényi entropy“* des Grades α, $0 \leq \alpha \leq \infty$ zu arbeiten. Hier wird insbesondere die *„Rényi entropy“* des 2. Grades, auch *„collision entropy“* genannt, benötigt.

2.12 Definition
Sei X ein endlicher Wahrscheinlichkeitsraum. Die **Shannon-Entropie** oder **Unsicherheit** von X ist definiert als:

$$H(X) := \sum_{x \in X,\ prob(x) \neq 0} prob(x) \cdot log_2(\frac{1}{prob(x)}) = - \sum_{x \in X,\ prob(x) \neq 0} prob(x) \cdot log_2(prob(x)).$$

2.13 Definition
Die **Rényi Entropie**, eine Erweiterung der Shannon Entropie, ist ein Mittel zur Quantifizierung der Entropie eines Systems. Die Entropie charakterisiert, wie viel Information wir im Durchschnitt erhalten, wenn wir den Wert der Zufallsvariablen kennen. Alternativ: Die Entropie charakterisiert die Unsicherheit über einen Wert einer Zufallsvariablen bevor man ihn kennt. Die Rényi Entropie ist definiert als:

$$H_\alpha\left(X\right) = \frac{1}{1-\alpha} log_2(\sum_{x \in X,\ prob(x) \neq 0} prob(x)^\alpha).$$

wobei $\alpha > 0, \neq 1$. Im Limes für α gegen 1, konvergiert H_α gegen die Shannon Entropie. Für alle α und α' mit $\alpha \leq \alpha'$, gilt:

$$H_\alpha \leq H_{\alpha'}.$$

Insbesondere gilt also: $H(X) \leq H_2(X)$.

2.14 Definition
Sei S eine Zufallsvariable mit den Werten in der endlichen Menge $\widetilde{S}$. Die **„collision probability“** $prob_c\left(S\right)$ von S ist definiert als:

$$prob_c\left(S\right) := \sum_{s \in \widetilde{S}} prob(S=s)^2.$$

Die *collision entropy* oder *Rényi entropy* (der Ordnung 2) von S ist:

$$H_2(S) := -log_2(prob_c(S)) = -log_2\left(\sum_{s\in\widetilde{S}} prob(S=s)^2\right).$$

$prob_c(S)$ ist die Wahrscheinlichkeit, dass zwei unabhängige Ausführungen von S das gleiche Resultat erzeugen.
Die mathematische Basis der *„privacy amplification“* ist das *„Smoothing Entropy Theorem“*. Es besagt, dass fast die ganze *„collision entropy“* einer Zufallsvariable S in einheitliche zufällige Bits konvertiert werden kann, durch Auswahl einer Funktion h von einer universellen Klasse von Hash-Funktionen und Anwendung von h auf S.

2.15 Definition
Eine Menge H von Hash-Funltionen $h : X \to Y$ wird **„universelle Klasse von Hash-Funktionen“** genannt, wenn für alle verschiedenen $x_1, x_2 \in X$ gilt:

$$prob\left(h(x_1) = h(x_2) : h \overset{u}{\leftarrow} H\right) = \tfrac{1}{|Y|}.$$

H heißt **„starke universelle Klasse von Hash-Funktionen“**, wenn für alle verschiedenen $x_1, x_2 \in X$“ und für alle (nicht notwendigerweise verschiedenen) $y_1, y_2 \in Y$ gilt:

$$prob\left(h(x_1) = y_1, h(x_2) = y_2 : h \overset{u}{\leftarrow} H\right) = \tfrac{1}{|Y|^2}.$$

Insbesondere sind starke universelle Klassen auch universelle Klassen. (Starke) universelle Klassen von Hash-Funktionen verhalten sich wie komplette Zufallsfunktionen unter Berücksichtigung der Kollision oder Wertepaare.

2.16 Beispiel
Eine direkte Berechnung zeigt, dass die Menge der linearen Abbildungen $\{0,1\}^l \to \{0,1\}^e$ eine starke universelle Klasse von Hash-Funktionen ist. Es existieren kleinere Klassen, z.B. die Menge:

$$H := \{h : \mathbb{F}_{2^l} \to \mathbb{F}_{2^e}, x \mapsto msb_e(a_0 x + a_1) \,|a_0, a_1 \in \mathbb{F}_{2^l}\}$$

ist stark universell. Hier betrachten wir $\{0,1\}^m$ ausgestattet mit der Struktur von $\mathbb{F}_{2^n}$ des Galois Körpers mit 2^m Elementen, und msb_e bezeichnet die e signifikantesten Bits.

2.17 Bemerkung
Während des Schlüsselgenerierungs-Schemas müssen Alice und Bob l Bits von den N übertragenden Bits der zufälligen Quelle auswählen. Sie müssen sich vorher auf zufällige Positionen $P = (p_1, ...p_l)$ einigen und diese geheim halten bis die Übertragung zu Ende ist. Auf den ersten Blick sind $l \cdot log_2(N)$ Bits nötig, um diese Positionen zu beschreiben. Weil l groß ist, muss eine große Anzahl an Bits zufällig ausgewählt werden und sicher von Alice nach Bob übertragen werden. Starke universelle Klassen von Hash-Funktionen liefern ebenso eine Lösung für dieses Problems.
Angenommen $N = 2^m$, und wir betrachten $\{0,1\}^m$ als $\mathbb{F}_{2^m}$. Alice und Bob arbeiten mit einer starken universellen Klasse

$$H := \{h : \mathbb{F}_{2^m} \to \mathbb{F}_{2^m}, x \mapsto a_0 x + a_1 | a_0, a_1 \in \mathbb{F}_{2^m}\}$$

von Hash-Funktionen. Sie fixieren vorher l paarweise verschiedene Elemente $x_1, ..., x_l \in \mathbb{F}_{2^m}$. H und die x_i können Eve bekannt sein. Nun werden zufällig einige h von H (unter Berücksichtigung der Gleichverteilung) ausgewählt und angewandt auf die x_i was zu l gleichverteilten und paarweise unabhängigen Positionen

$$p_1 = h(x_1), p_2 = h(x_2), ..., p_l = h(x_l)$$

führt. Dadurch wird die zufällige Auswahl der l Positionen auf die Auswahl des Elements h aus H reduziert, und dies erfordert die zufällige Auswahl von $2m = 2log_2(N)$ Bits. Für den Beweis von Theorem 2.10 wurden paarweise unabhängige Positionen benötigt: es ist nicht notwendig, dass die Sequenz der Positionen wirklich zufällig ist.

2.18 Beispiel (Cachin-Maurer-Schema)
Ein Satellit überträgt zufällige Bits mit einer Datenrate von 16 Gbits/s und wird einen Tag lang benutzt, um einen Zufallsgenerator R mit ungefähr $N = 1,7 \cdot 10^{15}$ Bits zu erstellen. Die Speicherkapazität der Angreiferin Eve wird mit 100 TB angenommen, was ungefähr $8,8 \cdot 10^{14}$ Bits entspricht. Alice und Bob wählen $l = 1,7 \cdot 10^9$ Bits aus R aus. Mittels *„privacy amplification“* erhalten sie einen Schlüssel k von ungefähr 6 MB und Eve kennt nicht mehr als 10^{-20} Bits von k, vorausgesetzt das Sicherheitsereignis E tritt ein. Die Wahrscheinlichkeit von E ist $\geq 1 - 10^{-4}$. Alice und Bob einigen sich auf l zufällige Positionen durch Wahl einer zufälligen starken universellen Hash-Funktion, welche $2log_2(N) \approx 102$ Bits benötigt. Ungefähr 10 GB an Speicher sind nötig zur Speicherung der Positionen und der Bits an den Positionen. Alice und Bob wählen zur *privacy amplification* eine universelle Hash-Funktion und teilen sie mit, was ungefähr l Bits $\approx$ 200 MB benötigt. Die Größe der Funktionen kann durch Benutzung von einer Funktion aus der Klasse der „fast überall universellen“ Hash-Funktionen stark reduziert werden.

2.19 Bemerkung
Das Ergebnis des Modells des begrenzten Speichers, welches wir hier betrachtet haben, ist nicht völlig zufriedenstellend. In Theorem 2.7 wird angenommen, dass der Angreifer nur in der Lage ist feste eher als beliebige Bits an Information über den Zufallsgenerator zu speichern. In dem *Cachin-Maurer Schema* müssen Alice und Bob, wie das Beispiel zeigt, über eine sehr große Speicherkapazität zum Speichern der Positionen und der Werte der zufälligen Bits verfügen, und die Größe des benötigten Speichers wächst stark an, wenn die Wahrscheinlichkeit des Sicherheitsereignisses E näher an 1 als an $1 - 10^{-4}$ liegen soll (was nicht zu vernachlässigen ist).
Kürzlich ist ein Fortschritt im Modell des begrenzten Speichers durch Anwendung des generellen Modells von Angriffen und beim Beweis der Sicherheit von Schemata, welche ähnlich sind zum Schema von U.M. Maurer erzielt worden.

3 Das Modell des verrauschten Kanals

Wie zuvor nutzen wir das Satelliten-Szenario. Zufällige Bits werden von einer Radio-Quelle ausgesandt. Alice und Bob empfangen diese Bits und generieren einen Schlüssel von diesen mittels öffentlicher Diskussion. Bei diesem Modell sind die Kommunikationskanäle von der Radio-Quelle nicht fehlerfrei. Alice, Bob und Eve empfangen die zufälligen Bits mit den Fehlerwahrscheinlichkeiten α, β und ϵ. Zwischen Alice und Bob besteht zusätzlich ein Kommunikationskanal, welcher fehlerfrei ist. Diesen Kanal kann die Angreiferin Eve belauschen. Wir nehmen an, dass Eve eine passive Angrieferin ist. Es gibt andere Modelle, welche einen aktiven Angriff berücksichtigen, wie z.B. das Modell von U.M.Maurer. Die Sicherheit des Schlüssels basiert darauf, dass kein Kommunikationskanal fehlerfrei ist und somit die Angreiferin nie alle Information über den Schlüssel erhalten kann. Erstaunlicherweise funktioniert das System auch, wenn Eve die zufälligen Bits über einen „besseren" Kanal empfängt als Alice und Bob.

Das Modell:

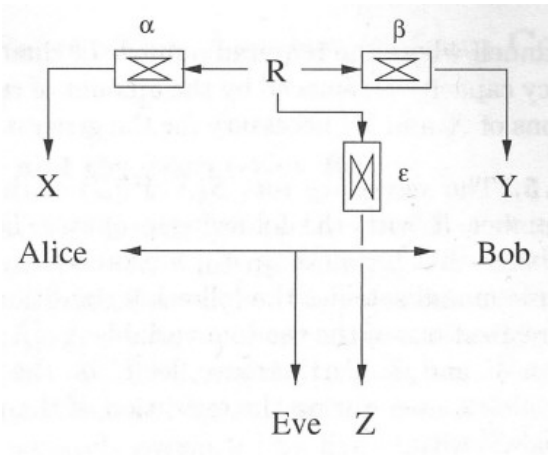

Abbildung 1: Das Satelliten-Modell

Die Schlüsselfestlegung funktioniert in drei Phasen. Wie in Abschnitt 2, startet es mit *„advantage distillation"* und endet mit *„privacy amplification"*. Es existiert eine zusätzliche Phase dazwischen, welche *„information reconciliation"* genannt wird.

1. *„advantage destillation“*:
Die erste Phase dient dazu, den Vorteil, den Eve durch ihre bessere Antenne hat, auszugleichen. Alice und Bob gleichen über ihren Kommunikationskanal ihre beiden Strings ab und filtern diejenigen Bits heraus, welche mit einer höheren Wahrscheinlichkeit gleich sind. Dazu existieren mehrere Protokolle. Wir werden hier das *„Parity-Check Protocol“* vorstellen.

Parity-Check Protocol:

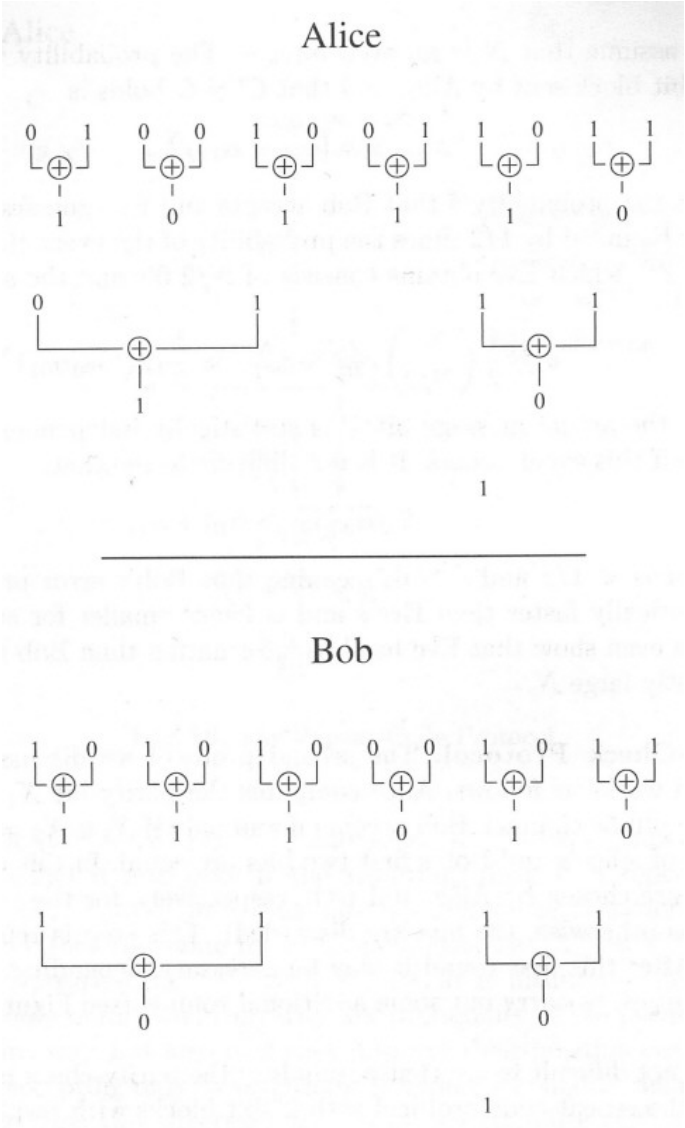

Abbildung 2: Parity-Check-Protokoll

Alice berechnet das Prüfbit $X_1 \oplus X_2$ und sendet dies über den Kommunikationskanal zu Bob. Bob akzeptiert dies nur, wenn $X_1 \oplus X_2 = Y_1 \oplus Y_2$, z.B. bei Gleichheit der ersten beiden Bits von Alice und Bob. In diesem Fall werden diese Bits für die nächste Runde genommen, ansonsten werden sie gestrichen. Dieser Schritt wird mehrere Male wiederholt, je nachdem wie hoch die Fehlerwahrscheinlichkeiten sind. Wodurch die Länge des Strings verkleinert wird.

2. *„information reconciliation“*:

Nach der ersten Phase haben Alice und Bob die Zeichenketten S_A und S_B errechnet. Bob hat mehr Information über Alices String als Eve. Nach Abschluss dieser Phase, welche als interaktive Fehlerkorrektur dient, müssen die beiden Zeichenketten gleich sein. Bei diesem Vorgang sollte möglichst wenig Information an Eve gehen. Für diese Phase gibt es ebenfalls mehrere Protokolle, wir werden hier das *Binary-Search Protokoll* vorstellen.

Binary-Search Protocol:

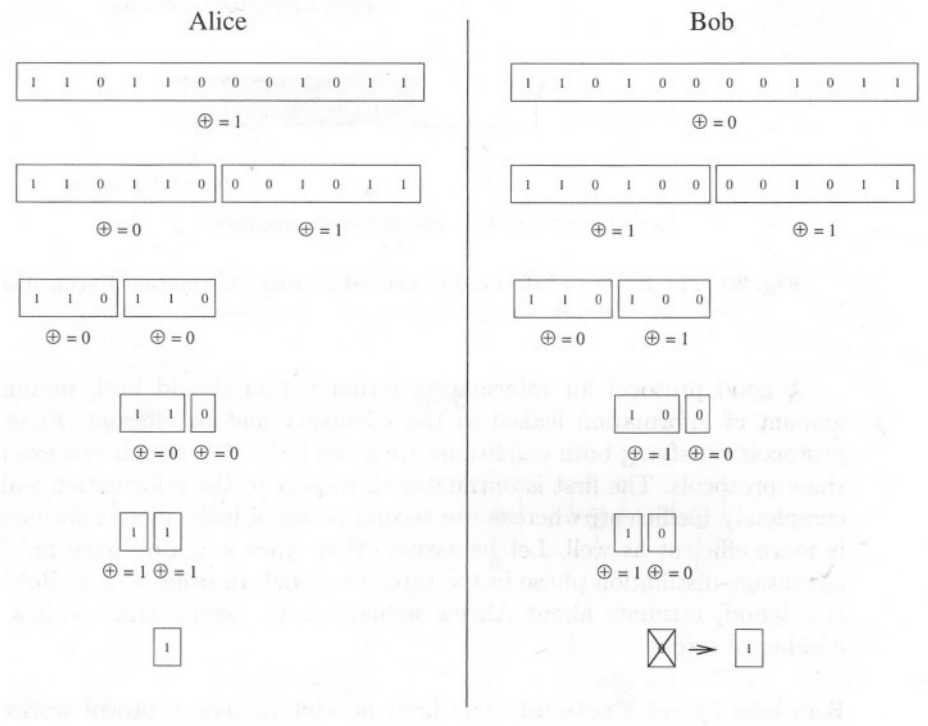

Abbildung 3: Binary-Search-Protokoll

Die Idee dieses Protokolls ist es, die fehlerhaften Stellen zu finden und zu berichtigen. Dabei bilden Alice und Bob die XOR-Summe von zufällig ausgewählten (aber gleichen) Substrings. Sollten diese Zeichenketten unterschiedlich sein, so ist mit einer Wahrscheinlichkeit von $\frac{1}{2}$ das Prüfbit verschieden. Ist das Prüfbit unterschiedlich, so teilt man den String in zwei gleichgroße Hälften, und wiederholt diesen Vorgang für beide Hälften. Da der Algorithmus endlich ist, wird so am Ende der Fehler gefunden und kann berichtigt sind. Alice und Bob wiederholen den Vorgang so lange, bis alle Fehler berichtigt wurden. Wenn n die Länge von den Zeichenketten S_A und S_B ist, so benötigt man $\lceil log_2 n \rceil$ Bits pro korrigierten Fehler.

3. *„privacy amplification“*:
Nach Phase 2 ist mit einer sehr hohen Wahrscheinlichkeit der gleiche zufällige String S der Größe l für Alice und Bob nutzbar und sie haben einen Vorteil gegenüber Eve. Eves Information über S, berechnet mit der *„Rényi Entropie“*, ist unvollständig. Durch Benutzung der *„privacy amplification“* Techniken, ähnlich wie sie in Abschnitt 2 beschrieben wurden, können Alice und Bob nun ihren gewünschten Schlüssel generieren. In dieser Phase wird, wie in Abschnitt 2 beschrieben, eine zufällige Funktion aus der Klasse der universellen Hash-Funktionen dafür benutzt, den zum Teil sicheren String S in einen hochsicheren String $\tilde{S}$ zu transformieren.

3.1 Bemerkung:
Als Erstes wählt Alice zufällig einen Vertreter aus einer festen universellen Klasse H von Hash-Funktionen aus, welche Bitstrings der Länge n auf Bitstrings der Länge r abbildet. Anschließend sendet sie diese über einen öffentlichen Kanal zu Bob. Alice und Bob berechnen nun $\tilde{S} := h(S)$. Wie lang dieser String $\tilde{S}$ sein kann, hängt von Eves Wissen über S ab. Die Tatsache, dass Eve einige Information über den String S hat, ist eine Beschreibung dafür, dass Eve die vollständige Information $U = u$ über S hat. Die Zufallsvariable S ist nicht gleichverteilt, z.B.:

$$H(S|U = u) < n.$$

In diesem Fall sagt man, dass Eve $n - H(S|U = u)$ Bits der (Shannon-)Information über S besitzt. Da der resultierende String $\tilde{S}$ der Bedingung

$$H(\tilde{S}|C, U = u) \approx r$$

genügen muss (wobei r die Länge von $\tilde{S}$ und C die Kommunikation über den öffentlichen Kanal ist). Intuitiv könnte man meinen, dass Eve t Bits an Information über S hat, so dass die Länge des resultierenden Strings $\tilde{S}$ ungefähr $n - t$ sein könnte. Diese Tatsache wäre richtig, wenn Eve deterministische Information über S hätte.
Wenn Eves Information nicht deterministisch ist, so ist die Aussage im Allgemeinen falsch, dass $n-t$ sichere Bits extrahiert werden können, wenn Eve t Bits an Shannon-Information über S hat, wie das folgende Beispiel zeigen wird:

3.2 Beispiel
Sei $P_{S|U=u}(s_0) = \frac{1}{2}$ für ein $s_0 \in \{0,1\}^n$ und $P_{S|U=u}(s) = 1/(2 \cdot (2^n - 1))$ für alle $s \neq s_0$. Dann ist $H(S|U = u) \approx \frac{n}{2}$, aber kein sicherer String $\tilde{S}$ (von jeder Länge, und vor allem nicht von $\frac{n}{2}$) kann extrahiert werden, weil Eve S kennt, deshalb auch $\tilde{S} = h(S)$, mit einer Wahrscheinlichkeit von $\frac{1}{2}$ (wobei h eine zufällig ausgewählte Hash-Funktion ist). Das bedeutet, dass $\tilde{S}$ nicht hochsicher sein kann.

3.3 Bemerkung
Im Folgenden benötigen wir die *„collision probability“* und die *Rényi Entropie* aus Abschnitt 2. Die *„collision probability“* ist die Wahrscheinlichkeit, dass zwei unabhängige Realisierungen der Zufallsvariablen X den gleichen Wert ergeben. Äquivalent zu dieser Aussage: Dies ist die Wahrscheinlichkeit zum Raten einer richtigen Realisierung von X, mit einer optimalen Strategie auf der Basis einer unabhängigen Realisierung von X, wobei die Verteilung von X unbekannt ist. Aus der *Jensenschen Ungleichung* folgt:

$$H_2(X) = -log_2(E[P_X]) \leq E[-log_2 P_X] = H(X).]$$

Es zeigt sich, dass die *Rényi Entropie* ein guter Informationsmaßstab in dem Kontext von *“privacy amplification“* darstellt.

3.4 Theorem
Sei S ein Bitstring der Länge n mit bedingter Wahrscheinlichkeit $P_{S|U=u}$, wobei $U = u$ das gegebene Wissen von Eve über S ist, sei G die Zufallsvariable, welche die wahrscheinlichkeitstheoretische Auswahl (unter Berücksichtigung der Gleichverteilung) eines Vertreters g einer universellen Klasse H von Hash-Funktionen darstellt. Die Funktion bildet einen Bitstring der Länge n auf einen der Länge r ab. Sei $\tilde{S} = G(S)$, dann gilt:

$$r \geq H(\tilde{S}|G, U = u) \geq H_2(\tilde{S}|G, U = u) \geq r - \frac{2^{r-H_2(S|U=u)}}{ln2}.$$

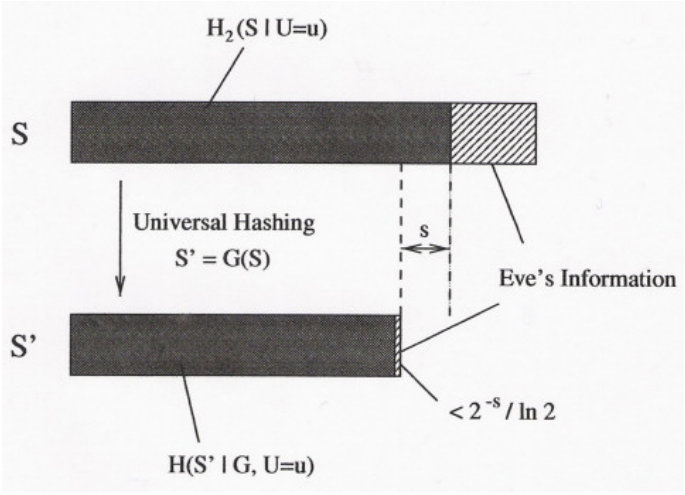

Abbildung 4: Rényi Entropie kann extrahiert werden mittels universellem Hashen

Intuitiv sagt Theorem 3.4 aus, dass wenn die Länge r von $\tilde{S}$ gewählt wird als

$$r := H_2(S|U = u) - s,$$

wobei s ein Sicherheitsparameter ist, dann ist der resultierende String $\tilde{S}$ hochsicher, wobei die Sicherheit exponetiell in s wächst.

3.5 Bemerkung
Eine wichtige Frage bleibt noch, und zwar welchen Einfluss der Informationsaustausch während der *"information reconciliation"* Phase auf die *Rényi Entropie* von S aus Eves Sicht hat, und deshalb auch auf den Schlüssel, welcher letztendlich generiert wird. Es wird in dem Paper von Cachin bewiesen, dass das Lernen von r physikalischen Bits die *Rényi Entropie* bis auf eine vernachlässigbare Wahrscheinlichkeit nicht signifikant um mehr als r Bits reduzieren kann.

Quellenangaben

1. C.Cachin und U.M. Maurer, *„Unconditional Security Against Memory-Bounded Adversaries“*

2. U.M. Maurer, *„Conditionally-perfect secrecy and a probably-secure randomized cipher“*

3. C.Cachin, *„Smooth Entropy and Rényi Entropy“*

4. Delfs und Knebel, *„Introduction to Cryptography“*

5. S. Wolf, *„Unconditional Security in Cryptography“*

6. Internet (Wikipedia, etc.)