Proseminar Technische Informatik

AES vs. SkipJack

Eine Übersicht über die beiden Verschlüsselungsverfahren
und ihre mögliche Anwendung in WSNs

Oliver Türpe
18. Juni 2010

1 Was beinhaltet dieser Report?

In dieser Arbeit soll der Unterschied zwischen zwei Verschlüsselungsverfahren dargestellt werden. Am Anfang steht eine allgemeinen Erklärung zum Thema Blockchiffre und danach wird ein genauerer Blick auf das Verschlüsselungsverfahren AES geworfen. Nach diesen Erläuterungen wird das Verfahren SkipJack vorgestellt. Am Ende der Arbeit grenze ich die beiden Verfahren von einander ab und werde die Vor- und Nachteile der beiden Verfahren zeigen.

2 Motivation

Warum brauchen wir Verschlüsselung in Kabellosen Sensornetzen (WSN)? Ist es wichtig, welche Technik wir dazu nutzen? Und wenn wir eine Technik gefunden haben, ist diese auch geeignet?

Diese Fragen sollen diese Arbeit beantworten.

Warum brauchen wir Verschlüsselung? Da WSNs viele verschiedene Daten ermitteln, ist es wichtig, die Daten nicht jedem zugänglich zu machen. Durch eine Grundverschlüsselung ist dies am einfachsten zu ermöglichen, da so nur der, der die Daten erfasst, sie entschlüsseln kann. Doch die Frage nach der Verschlüsselungstechnik ist nicht einfach zu beantworten. So gibt es in der IT Welt lizenzfreie Verfahren, die eine Verschlüsselung erlauben, ohne Mehrkosten zu verursachen. Im Zuge dieser Arbeit werden zwei dieser Verfahren untersucht.

3 Was ist eine Blockchiffre?

Im Prinzip arbeitet eine (symmetrische) Blockchiffre wie folgt:[1]

- Es wird ein geheimer Schlüssel erstellt oder ausgewählt.
- Die zu verschlüsselnden Daten werden noch im Klartext in mehrere gleich große Blöcke unterteilt.
- Danach wird jeder Block mit dem Schlüssel kombiniert und dann transformiert. So kann der neue Block nicht mehr ohne den Schlüssel zum Klartext transformiert werden.

- Aus diesen transformierten Blöcken besteht nun der "Geheimtext". Die Länge des Textes wird vom Chiffrieren nicht geändert, weil sich die Blockgröße nicht ändert.
- Da der Algorithmus invers ist (deshalb symmetrisches Verfahren genannt), kann mit dem gleichen Schlüssel der Geheimtext wieder lesbar gemacht werden.

4 Was ist ein WSN?

4.1 Definition

Als "kabelloses Sensornetzwerk"(WSN) bezeichnet man im Allgemeinen ein Netz von verschiedenen Sensorknoten, die per Funk kommunizieren. Diese können entweder infrastruktur-basiert sein oder sich selbst organisieren. Ziel des Sensornetzes ist es jedoch, immer die Umgebung mit Sensoren zu überwachen, Ereignisse zu entdecken und zu protokollieren.[4]

4.2 Hardware von WSNs

Im Allgemeinen besteht ein Sensorknoten aus einem Prozessor, einem Speichermedium sowie mehreren Sensoren, die mit dem eingebauten Funkmodul kommunizieren. Die Energieversorgung des Knotens wird durch eine Batterie gewährleistet[6].

Meist besteht ein Sensorknoten aus einem oder mehreren Sensoren. Es existieren aber auch Entwürfe mit Knoten ohne Sensoren, diese sind dann nur zur Verwaltung der Kommunikation gedacht.

Man unterscheidet zwei Arten von Sensornetzen. Zum einen die heterogenen Sensornetze[1], die sich besonders für Anwendungen eignen, wo verschiedene Daten aufgezeichnet und erfasst werden. Zum anderen gibt es homogene Sensornetze[2], die sich sehr gut eignen, um bestimmte Ergebnisse zu verifizieren, da hier die gleichen Daten redundant erfasst werden.

4.3 Probleme von WSNs

WSNs bringen auch Probleme mit sich. So liegt die Rechenleistung des Mikroprozessor weit unter der uns gewohnten Leistung. Auch der Speicher ist gering, da nur so die kleine Bauweise - bei moderaten Kosten - ermöglicht werden kann. Das größte Problem bei WSNs ist jedoch die Energieversorgung. Die Batterie eines Sensorknotens besitzt nur eine geringe Leistung, außerdem wird viel Energie bei der Übertragung von Daten per Funk verbraucht.

5 Advanced Encryption Standard - ein lizenzgebührenfreier Standard

Als das NIST[3] 1997 offiziell eine Suche nach einer symmetrischen Blockchiffre, die zur Verschlüsselung von persönlichen Daten genutzt werden sollte, ausschrieb, war ein Kandidat der Rijnadael Algorith-

[1] Hiermit bezeichnet man Sensornetze, die aus einer Vielzahl von verschiedenen Sensoren bestehen.[6]
[2] Ein Sensornetz wird als homogen bezeichnet, wenn alle Sensorknoten über die gleiche Ausstattung verfügen.[6]
[3] U.S. National Institute of Standards and Technology (www.nist.gov)

mus[4].
Dieser Algorithmus basiert auf Bytes und Worten. Jedes Byte wird laut [7] als $a = a_7 \dots a_0$ mit $a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$ dargestellt. Auf diesem Polynom muss nun noch die Addition sowie die Multiplikation festgelegt werden.

- **Addition**: Hier wird die XOR-Verknüpfung[5] genutzt.
- **Multiplikation**: Das Polynom wird ähnlich wie im Körper $\mathbb{R}$ definiert. Nur wird am Ende durch das festes Polynom $t(x) := x^8 + x^4 + x^3 + x + 1$ (über $\mathbb{F}_2$ irreduzibel[6]) modulo gerechnet.

5.1 Spezifikation

Da eine symmetrische Blockchiffre vorliegt, wissen wir, dass mit dem selben Schlüssel dechiffriert und chiffriert werden kann. Bei Rijndael können Blockgröße und Schlüssellänge unabhängig voneinander gewählt werden. Beide können Werte von 128, 192 oder 256 Bit betragen[8]. Daraus ergibt sich beispielsweise bei einer Schlüssellänge von "nur" 128 Bit eine mögliche Anzahl von 2^{128} Schlüsseln.[8]

5.2 Runden bei Rijndael

Teile des Algorithmus werden mehrfach ausgeführt, diese Ausführung wird dann als Runde bezeichnet. Der Ablauf einer Runde ist folgendermaßen[8]:

1. Erst wird ein Rundenschlüssel[7] erstellt.
2. Darauf folgt die Schüsseladdition. Bitweise wird nun eine XOR-Verknüpfung zwischen dem Block, welcher verschlüsselt werden soll und dem Rundenschlüssel ausgeführt.
3. Danach werden die Daten monoalphabetisch verschlüsselt[8].
4. Nun werden die Zeilen der Tabelle, die die monoalphabetische Verschlüsselung vornimmt, vertauscht. Diese Verschiebung ist blockorientiert.
5. Sind alle Zeilen vertauscht, werden die Spalten mit einer konstanten multipliziert und anschließend per XOR verknüpft.

5.3 Struktur von Rijndael

Die Menge an Runden hängt von der Schlüssellänge und der Blockgröße (jeweils in Bit) ab. In Tabelle 1 kann man erkennen, wie sich Blockgröße und Schlüssellänge auf die Anzahl der Runden, also der

[4] Dieser Algorithmus ist nach seiner Entwicklern Joan Daemen und Vincent Rijmen benannt.
[5] Eine XOR Verknüpfung wird in der Aussagenlogik definiert. Die Gesamtausssage ist wahr, wenn genau eine Teilaussage wahr ist. In der Informatik entspricht dies der Addition und folgender Modulo-Rechnung mit 2.
[6] Irreduzible Polynome zerfallen in einem Körper in ihre Linearkombinationen (hiermit werden meist die Nullstellen bezeichnet).
[7] Ein Rundenschlüssel ist ein Teilschlüssel des Originalschlüssels.
[8] In einer Tabelle ist genau vorgeschrieben, welche Werte der Blöcke mit anderen getauscht werden. Vergleiche dazu die Tausch-Tabelle von SkipJack.

wiederholten Durchführung der Chiffrierung, auswirken. Mehr Runden bedeuten hier mehr Abstraktion.

Blockgröße (→) / Schlüssellänge (↓)	128	192	256
128	10	12	14
192	12	12	14
256	14	14	14

Tabelle 1: Anzahl der Runden

5.4 Die Stärken von Rijndael

5.4.1 Sicherheit

Eine der interessantesten Fragen beim Thema Verschlüsselung lautet: Wie leicht ist dieses System zu knacken?

Um die Stärke von Rijndael zu zeigen, nehmen wir an, dass ein Klartext mit 10 Runden verschlüsselt wurde. Man müsste nun $10*2^{32}$ Klartextblöcke sammeln und auf diese Blöcke dann 2^{44} Operationen anwenden, um den Schlüssel zu ermitteln[9]. Außerdem müsste eine Menge von 440 GB Klartext verschlüsselt werden, um ihn anschließend zu analysieren.

Bei nur einer Runde mehr, ist ein Rechenaufwand von 2^{120} notwendig. Dieser immense (Mehr-) Aufwand pro Runde zeigt, wie sicher der Algorithmus ist. Aus Tabelle 1 können wir entnehmen, dass mindestens 10 Runden durchgeführt werden.

Man sollte jedoch bedenken, dass bei praktischen kryptografischen Verfahren die Sicherheit nur rechnerisch bewiesen werden kann, da ein Versuch mit bekannten Methoden zu lange dauern würde.

5.4.2 Bewertung der Sicherheit

Rijndael gilt weltweit als sehr sicher[12]. Kryptografen testen und benutzen den Algorithmus. Außerdem wurde er von einer nicht staatlichen Institution entwickelt. Wichtig ist jedoch, dass es drei Entwicklern 2001 möglich war den kompletten Algorithmus als geschlossene Formel mit 2^{50} Termen darzustellen[14]. Dies muss allerdings kein Nachteil sein. Es zeigt eher die Nachvollziehbarkeit und Einfachheit des Algorithmus. Zu Bedenken ist die Laufzeit des Algorithmus, durch das Vertauschen der Spalten in einer Matrix, welche meist durch ein zwei-dimensionales Array repräsentiert wird, geht viel Zeit verloren, da dies mit Schleifen realisiert werden muss. Versucht man hier die Dauer zu minimieren, ist dies nur auf Kosten des Speichers möglich.

Viele größere Unternehmen setzen außerdem AES in ihren Produkten ein. Die bekanntesten wären IEEE 802.11i, FileVault und Microsoft .NET[11][10][9].

[9] Das wären 17592186044416 Operationen, man bräuchte also mit einem Core i7 ungefähr 148082374 Stunden (ca. 16950 Jahre), da er 33 GigaFLOPS schafft [2].

6 SkipJack - ein Forschungsergebnis der 80er Jahre

Das symmetrische Verschlüsselungsverfahren SkipJack wurde von der National Security Agency (NSA) entwickelt. Nach einer langen Phase der Geheimhaltung wurde am 24. Juni 1998 die Spezifikation des Algorithmus veröffentlicht[18].

Im Folgenden wird auf die technischen Spezifikationen näher eingegangen und an einem Beispiel veranschaulicht, wie der Algorithmus arbeitet und warum er sicher[10] ist.

6.1 Spezifikation

Die Länge des Schlüssels beträgt genau 80 Bit und die Blockgröße 64 Bit[18]. Bei SkipJack werden immer 8 Byte an Datenblöcken verschlüsselt, dadurch werden abwechselnd zwei Regeln eingesetzt. Zu Beginn erläutere ich die verwendete Terminologie und anschließend die zwei Regeln (Regel A und Regel B).

- **Wort:** Ein Wort ist ein 16-bit Wert.
- **Permutation eines Wortes:** Die Permutation ist eine invertierbare Funktion von $V^n \rightarrow V^n$. Hierbei werden die Werte permutiert, nicht jedoch die Bits in den Werten.
- $X \oplus Y$, das exklusive Or (XOR) von zwei Mengen X und Y

6.2 Das Regelwerk

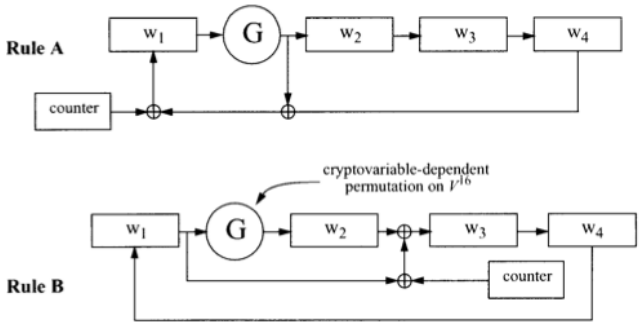

Abbildung 1: Visualisierung der Regeln[18]

Regel A:

1. G permutiert das erste Wort (ab jetzt w_1).
2. Das neue Wort w_1^* ist die XOR[11] Ausgabe von G, der Zähler und w_4.

[10] Als "sicher" bezeichnet man eine Verschlüsselung, wenn der Aufwand und die Kosten, um die Verschlüsselung zu knacken sich mit dem Gewinn durch das Knacken aufwiegen. Als "sehr sicher" bezeichne ich Verfahren, bei denen die Kosten und der Aufwand den Gewinn übersteigen.

[11] Eine XOR Verknüpfung wird in der Aussagenlogik definiert. Die Gesamtausssage ist wahr $\Leftrightarrow$ genau eine Teilaussage wahr ist. In der Informatik entspricht dies der Addition mit der modulo 2 Rechnung.

3. Die Wörter w_2 und w_3 werden ein Register nach rechts geschoben.

4. Das neue Wort w_2^* ist die Ausgabe von G.

5. Der Zähler wird um eins erhöht.

Regel B funktioniert ähnlich.
Die Regeln werden durch folgende Gleichungen für die Ver- und Entschlüsselung repräsentiert[18]. In Tabelle 2 und 3 sind die verschiedenen Gleichungen dargestellt. Die Schrittnummer ist in den Gleichungen immer durch den Exponenten dargestellt.

Regel A	Regel B
$w_1^{k+1} = G^k(w_1^k) \oplus w_4^k \oplus counter^k$	$w_1^{k+1} = w_4^k$
$w_2^{k+1} = G^k(w_1^k)$	$w_2^{k+1} = G^k(w_1^k)$
$w_3^{k+1} = w_2^k$	$w_3^{k+1} = w_1^k \oplus w_2^k \oplus counter^k$
$w_4^{k+1} = w_3^k$	$w_4^{k+1} = w_3^k$

Tabelle 2: Verschlüsselungsgleichungen bei SkipJack[18]

Regel A^{-1}	Regel B^{-1}
$w_1^{k-1} = [G^{k-1}]^{-1}(w_2^k)$	$w_1^{k-1} = [G^{k-1}]^{-1} w_2^k$
$w_2^{k-1} = w_3^k$	$w_2^{k-1} = [G^{k-1}]^{-1}(w_2^k) \oplus counter^{k-1}$
$w_3^{k-1} = w_4^k$	$w_3^{k-1} = w_4^k$
$w_4^{k-1} = w_1^k \oplus w_2^k \oplus counter^{k-1}$	$w_4^{k-1} = w_1^k$

Tabelle 3: Entschlüsselungsgleichungen bei SkipJack[18]

6.3 Die einzelnen Schritte

Der Algorithmus benötigt insgesamt 32 Schritte[12][18].

- **Verschlüsseln:**
  - Die Eingabe ist: w_i^0, $1 \leq i \leq 4$.
  - Starte nun den Zähler (counter) mit 1.

[12] Als Schritt bezeichnet man bei SkipJack das Anwenden der verschiedenen Regeln, wie ich es oben benannt habe.

  - Führe 8 mal Regel A durch, danach 8 mal Regel B.
  - Nun wieder 8 mal Regel A, gefolgt von 8 mal Regel B.
  - Die Verschlüsselung ist nun abgeschlossen.
  - Nach jedem Schritt wird der Zähler dabei um eins erhöht.
  - Die Ausgabe ist dann: w_i^{32}, $1 \leq i \leq 4$.

- **Entschlüsseln:**
  - Die Eingabe ist: w_i^{32}, $1 \leq i \leq 4$.
  - Wir starten mit einem Zähler von 32.
  - Nun wird 8 mal Regel B^{-1} ausgeführt gefolgt von 8 mal Regel A^{-1}.[13]
  - Dieser Schritt wiederholt sich dann.
  - Der Zähler wird hierbei jedes Mal um eins vermindert.
  - Als Ausgabe erhalten wir dann: w_i^0, $1 \leq i \leq 4$.

6.4 G - Permutationen

Jede Runde von G beinhaltet ein Byte voller Kryptovariablen. Diese Kryptovariablen sind eine Vier-Runden-Feistelchiffre[14]. Den schematischen Aufbau von G ist in Abbildung 2 visualisiert.

Die hierbei angewendete Rundenfunktion ist eine Byte-Tausch-Tabelle, im Folgenden mit **F** bezeichnet.

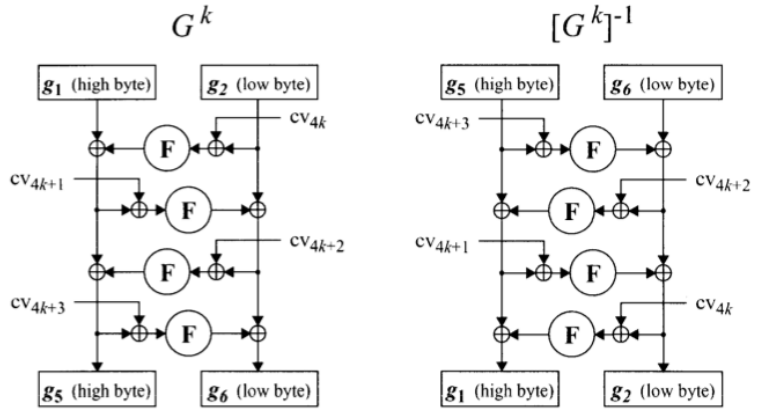

Abbildung 2: Schematische Darstellung der Rundenfunktionen[18]

[13] Die beiden Regeln A^{-1} und Regel B^{-1} wurden von mir im letzten Abschnitt genannt.
[14] Die Feistelchiffre ist nach dem deutschen Erfinder und Mitarbeiter von IBM Horst Feistel benannt. Dies ist ein Synonym für eine Blockchiffre. [3]

6.5 Die Tausch-Tabelle F

Aus der SkipJack Spezifikation wurde die Tabelle **F** angehängt[15]. Diese ist in Hexadezimal Notation. Die erste Zeile gibt hierbei die ersten 4 Bits der Eingabe an, während die erste Spalte die zweiten 4 Bits der Eingabe darstellen. An Abbildung 3 kann man schnell ablesen, wie die verschiedenen Werte

	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
0x	a3	d7	09	83	f8	48	f6	f4	b3	21	15	78	99	b1	af	f9
1x	e7	2d	4d	8a	ce	4c	ca	2e	52	95	d9	1e	4e	38	44	28
2x	0a	df	02	a0	17	f1	60	68	12	b7	7a	c3	e9	fa	3d	53
3x	96	84	6b	ba	f2	63	9a	19	7c	ae	e5	f5	f7	16	6a	a2
4x	39	b6	7b	0f	c1	93	81	1b	ee	b4	1a	ea	d0	91	2f	b8
5x	55	b9	da	85	3f	41	bf	e0	5a	58	80	5f	66	0b	d8	90
6x	35	d5	c0	a7	33	06	65	69	45	00	94	56	6d	98	9b	76
7x	97	fc	b2	c2	b0	fe	db	20	e1	eb	d6	e4	dd	47	4a	1d
8x	42	ed	9e	6e	49	3c	cd	43	27	d2	07	d4	de	c7	67	18
9x	89	cb	30	1f	8d	c6	8f	aa	c8	74	dc	c9	5d	5c	31	a4
Ax	70	88	61	2c	9f	0d	2b	87	50	82	54	64	26	7d	03	40
Bx	34	4b	1c	73	d1	c4	fd	3b	cc	fb	7f	ab	e6	3e	5b	a5
Cx	ad	04	23	9c	14	51	22	f0	29	79	71	7e	ff	8c	0e	e2
Dx	0c	ef	bc	72	75	6f	37	a1	ec	d3	8e	62	8b	86	10	e8
Ex	08	77	11	be	92	4f	24	c5	32	36	9d	cf	f3	a6	bb	ac
Fx	5e	6c	a9	13	57	25	b5	e3	bd	a8	3a	01	05	59	2a	46

Abbildung 3: Die Tausch Tabelle **F**[18]

getauscht werden. Es wird beim Tausch immer ein Block gesucht und mit dem jeweiligen Eintrag der Tabelle getauscht. Als Beispiel nehmen wir hier das Wort Informatik[16]:
I n f o r m a t i k $\rightarrow^{\alpha}$ 49 6E 66 6F 72 6D 61 74 69 $6B_{16}$ $\rightarrow^{F}$ B4 9B 65 76 B2 98 D5 B0 00 56

Die Laufzeit des Algorithmus dürfte sehr kurz sein, da er nur aus XOR Verknüpfungen besteht und diese schnell durch den Chip realisiert werden können. Der Speicherverbrauch dürfte recht gleichmäßig sein, da bestimmte Blöcke nur überschrieben werden.

7 Ein Vergleich der Algorithmen

Im Folgendem werden die Gemeinsamkeiten und Unterschiede der beiden Algorithmen erläutert, um heraus zu finden, in wie weit sich die Algorithmen für ein WSN eignen. Als Vergleichskriterien bieten

[15] Da die Tabelle in der Spezifikation des Algorithmus festgelegt ist, wird sie nicht auswechselbar sein.[18]
[16] α ist hier die Funktion, die jedes Zeichen eines Wortes in seinen hexadezimalen Wert laut der ASCII Tabelle umwandelt.

sich hier der Speicheraufwand und die Laufzeit an. Ein weiteres Kriterium ist der Grad der Abstraktion.

7.1 Gemeinsamkeiten

Wie schon zuvor beschrieben, handelt es sich bei beiden um eine symmetrische Blockchiffrierung. Beide Algorithmen sind des weiteren sicher gegen brute-force Attacken, wie die beiden Beispiele zeigen. Sie enthalten weiterhin Regeln, die öfter ausgeführt werden und somit leicht zu realisieren sind. Des weiteren haben die beiden Algorithmen gemeinsam, dass ihre Spezifikation öffentlich zugänglich ist und somit von jedem implementiert und verifiziert werden können.

7.2 Unterschiede

Neben den Erfindern des Algorithmus ist wohl ihr Ruf der größte Unterschied. SkipJack wurde von der NSA entwickelt. Auf Grund der Meinung vieler Menschen, die NSA hätte eine Hintertür eingebaut, ergibt sich das ein Problem. Diese Meinung wird bestärkt durch die Haltung der NSA in der Vergangenheit, in dem die Spezifikation nicht veröffentlicht werden sollte.

AES hingegen hat den Nachteil, das der Algorithmus als geschlossene Formel darstellbar ist. Es ist nur nicht geklärt, ob dieser Term bestehend aus Millionen von Summanden bei einem Versuch das System zu knacken helfen würde.

Ein weiterer Unterschied ist die Häufigkeit der Verwendung. Bei meiner Recherche habe ich keine Anwendung von SkipJack gefunden, während AES öfter in kryptologischer Software anzutreffen ist. Besonders die feste Schlüssellänge bei SkipJack ist meiner Meinung nach dafür verantwortlich daran, dass mehrere Wissenschaftler einen Angriff entwickeln konnten, mit dem es möglich war, den Schlüssel teilweise zu ermitteln.[19] Dieser funktionierte aber nur, wenn man sowohl Klartext als auch den dazu passenden verschlüsselten Text besaß (dies ist aber in unserem Fall nie realistisch).

Funktional arbeitet der SkipJack Algorithmus schneller, da er mehr Runden benötigt, weil die Block- und Schlüsselgrößen kleiner sind. Dafür sind diese Runden schnell auf einem WSN ausgeführt, da es sich wie oben erwähnt nur um XORs (s.o.) handelt. Der Speicherverbrauch ist zudem gering, da nur Blöcke überschrieben werden. Durch die kleine Schlüssellänge sollte der Schlüssel aber häufig getauscht werden, was zusätzlichen Traffic verursacht und besonders bei der geringen Kapazität der Energieversorgung auf Kosten der Laufzeit geht.

Bei AES sind die Anzahl der Runden kleiner, dafür werden mehr Rechenoperationen ausgeführt. Die Schleifen schlagen sich hier auf die Laufzeit nieder. Wenn man jedoch zwei Tabellen beim Tausch anlegt, braucht man keine Schleifen, sondern speichert die Tabelle in anderer Form zwischen. Dies ist bei großen Tabellen jedoch sehr speicherintensiv. Der Rechenaufwand wäre dann kleiner als bei SkipJack, dafür aber der Speicherverbrauch höher. Da der Schlüssel während einer längeren Anwendung nicht gewechselt werden muss, entsteht hier weniger Traffic während des Betriebs des Knotens, was zu einer längeren Betriebszeit führt.

In Tabelle 4 sind die Vor- und Nachteile der beiden Algorithmen noch einmal zusammen gefasst.

Kriterium	SkipJack	AES
Laufzeit der Berechnung	+	-
Speicherverbrauch	+	-
Betriebszeit	-	+
Energieverbrauch	-	+
Traffic	-	+
"Ruf"	0	+

Tabelle 4: Kriterien im Vergleich

8 Welcher Algorithmus ist der bessere?

Ich denke, dass in Kabellosen Sensornetzwerken der AES besser geeignet ist, da es durch die variable Schlüssel- sowie Blockgröße besser auf die Leistung der Mikroprozessoren und kleinen Flash-Speicher zugeschnitten werden kann. Es dürfte auch möglich sein, bestimmte Teile des Algorithmus hardwareseitig zu implementieren, so dass der Prozessor weniger Arbeit hat und die Energie nicht schneller verbraucht wird.

Ich würde aber entweder nur die Kommunikation unter den einzelnen Sensorknoten verschlüsseln oder die Daten auf dem Speicher. Favorisieren würde ich jedoch die Verschlüsselung der Daten auf dem Festspeicher, da die Daten bei einem Verlust des Knotens nicht ausgelesen werden können. Wenn die Daten drahtlos übermittelt werden, ist es egal, ob sie verschlüsselt sind oder nicht, da nur grundverschlüsselte Daten übermittelt werden. Ein eventueller Lauscher könnte mit diesen Daten nichts anfangen, da die übertragenen Daten zu wenig Informationen enthalten, um die Verschlüsselung zu umgehen.

Literatur

[1] Lai, Xuejia : *On the design and security of block ciphers* / Xuejia Lai. - 1. Aufl.. - Konstanz : Hartung-Gorre, 1992. - XII, 108 S. : graph. Darst. - 3-89191-573-X. - (ETH series in information processing ; 1)

[2] http://www.tecchannel.de/_misc/img/detail.cfm\pk=373936&fk=1775602&id=il-78759939369673440 (15.05.2010)

[3] Beutelspacher, Albrecht : *Kryptografie in Theorie und Praxis : mathematische Grundlagen für elektronisches Geld, Internetsicherheit und Mobilfunk* / Albrecht Beutelspacher ; Heike B. Neumann ; Thomas Schwarzpaul. - Wiesbaden : Vieweg, 2005. - XIV, 319 S. - 3-528-03168-9

[4] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, Erdal Cayirci, *A Survey on Sensor Networks*, August 2002, Georgia Institute of Technology, Auszug aus dem IEEE Communications Magazine

[5] Dr. rer. nat. Bastian Katz, *Algorithmen für Ad-hoc- und Sensornetze*, SS 2009, Universität Karlsruhe, Internetseite: http://i11www.iti.uni-karlsruhe.de/teaching/sommer2009/sensornetze/index (02.06.2010)

[6] T. Hänselmann: *An FDL'ed Textbook on Sensor Networks*. Elektronisches Lehrbuch zur Vorlesung Sensornetze an der Universität Mannheim, verfügbar unter der GNU-Lizenz für freie Dokumentation

[7] Doemen, Joan; Rijmen, Vincent, *AES Proposal: Rijndael*: Rijndael, Proton World Int., Katholische Universität Leuven, 09/1999

[8] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[9] http://msdn.microsoft.com/en-us/library/system.security.cryptography.rijndael.aspx (15.05.2010)

[10] http://docs.info.apple.com/article.html?path=Mac/10.4/de/mh1877.html (10.06.2010)

[11] Edney, J. und W. A. Arbaugh (2003): *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*. 1. Auflage, Addison Wesley, Boston

[12] Eli Biham and Nathan Keller. *Cryptanalysis of reduced variants of RIJNDAEL.* In Proceedings of the Third Advanced Encryption Standard Conference. NIST, April 2000.

[13] Chrstian Wilkin, *Der Algorithmus des "Advanced Encryption Standard"*, Dezember 2004, Verfügbar bei der Fachhochschule Trier, Fachbereich Design und Informatik oder unter der privaten Homepage des Autors: http://www.realtec.de/privat/files/AES_Krypto_Seminar.pdf

[14] Niels Ferguson; Richard Schroeppel; Doug Whiting, *A simple algebraic representation of Rijndael. Counterpane Internet Security*, Sandia National Laboratory, Hi/fn, Inc. 2001.

[15] http://www.repges.net/AES-Kandidaten/aes-kandidaten.html

[16] http://www.formaestudio.com/rijndaelinspector/archivos/inspector.html (18.06.2010)

[17] http://www.codeplanet.eu/tutorials/cpp/51-advanced-encryption-standard.html (18.06.2010)

[18] SkipJack and KEA Algorithm Specifications, Version 2.0, 29 May 1998. Available at the National Institute of Standards and Technology's web page, http://csrc.nist.gov/groups/STM/cavp/documents/skipjack/skipjack.pdf

[19] Eli Biham, Alex Biryukov, Orr Dunkelman, Eran Richardson, Adi Shamir, *Initial Observations on the Skipjack Encryption Algorithm*, 25. Juni 1998, verfügbar unter: http://www.cs.technion.ac.il/ biham/Reports/SkipJack/

[20] https://www.cosic.esat.kuleuven.be/nessie/ U(10.06.2010)