Verfahren zur Verschlüsselung von Textmeldungen

BOS-Krypt

als Ergänzung zur TR BOS

„Geräte für die digitale Funkalarmierung“

Teil 1 - Spezifikation

INHALT

Vorwort

Die Gesamtdokumentation besteht aus den drei Teilen:

1. BOSKRYPT – Spezifikation

2. BOSKRYPT - Schlüsselaustauschverfahren

3. BOSKRYPT - IOP Prozess

Weitere Informationen

Als Ergänzung gibt es noch BOSKRYPT - Historie und Hinweise zur Anwendung und Implementierung

A1. Verwendung innerhalb der BOS

Die Anwendung des Verfahrens kann von allen BOS Anwendern, z.B. im Rahmen von Beschaffungen, in eigener Zuständigkeit gefordert werden, wenn rechtliche Vorgaben, z.B. des Datenschutzes, in der Zukunft eine Verschlüsselung der Texte zwingend erfordern. Es ermöglicht dann erstmals den gemischten Betrieb von Komponenten unterschiedlicher Hersteller. Die Verwendung dieser Beschreibung und die Nutzung der darin aufgeführten Verfahren ist allen Herstellern kostenfrei möglich. Die Vertragsgestaltung zwischen Hersteller und Anwender bzw. anderen Herstellern für Produkte oder Dienstleistungen, z.B. die Nachrüstung von DME, Lieferung einer DLL etc. wird im Innenverhältnis zwischen Hersteller / Hersteller und Anwender geregelt. Die Aufwendungen für den in Teil 3 beschriebenen IOP handeln die Hersteller bei Bedarf untereinander aus.

A2. Bezug zur Prüfung nach TR BOS

Es besteht noch keine Pflicht der Hersteller von DA Komponenten das Verfahren in Ihre Produkte zu integrieren, folglich ist es auch kein Bestandteil der Prüfung nach TR BOS.

A3. Wahlfreiheit anderer Kryptoverfahren

Ebenfalls ist es möglich, dass Hersteller eigene Standards der Verschlüsselung, parallel oder eigenständig, zur Anwendung bringen, wenn die Anwender dies zulassen und wichtige Gründe dafür sprechen.

A4. Ort der Verschlüsselung

Bei der Verschlüsselung der Meldetexte kommen grundsätzlich zwei Stellen in Frage. Der digitale Alarmgeber nach TR BOS (DAG) oder das in den meisten Leitstellen eingesetzte Einsatzleitsystem (ELS). Grundsätzlich können auch beide Quellen parallel betrieben werden wenn sichergestellt ist, dass ein durch das ELS verschlüsselter Text nicht noch einmal durch den DAG verschlüsselt wird. Aufgrund dieser Möglichkeiten stellen ELS eine eigene Gattung Sender dar, da sie Texte unabhängig von der verwendeten Infrastruktur verschlüsseln können.

A5. IOP (Inter-Operabilitäts-Prozess)

Da eine Überprüfung der Funktionalitäten durch die Zentralprüfstelle an der LFS-BW nicht vorgesehen ist, wird zur Erreichung der Kompatibilität der einzelnen Produkte mit dem so genannten IOP gearbeitet. Dabei testen die Hersteller ihre Produkte gegenseitig auf Funktionalität. Die Ergebnisse werden von den Herstellern an die Prüfstelle schriftlich gemeldet. Diese trägt eine festgestellte Kompatibilität in eine Tabellenmatrix ein, links senkrecht die Sender (DAU, DAG, ELS, Prüfsender), oben quer Empfänger (DME, DSE und andere Empfangskomponenten). Der Kreuzungspunkt enthält mindestens das Datum der Meldung(en) und ergänzende Informationen (Berichtsnummer). Der Umfang der Prüfung wird separat definiert, ebenso der Inhalt der Meldung an die Prüfstelle.

Die Hersteller können Meldungen auch einseitig abgeben, z.B. ein ELS Hersteller A der im Rahmen eines Kundenauftrages das Verfahren über die Alarmierungsinfrastruktur des Herstellers X ohne dessen Mitwirkung zum DME B positiv evaluiert hat.

A6. Praxishinweise

Für die Nutzung des Verfahrens gibt es einige Hinweise die im Teil „Historie und Hinweise zur Anwendung und Implementierung“ aufgeführt sind.

B. Beschreibung des Verfahrens

B1. Einleitung

Dieses Dokument beschreibt die Nachrichtenverschlüsselung für das RPC1 Verfahren. Der beschriebene Algorithmus gewährleistet Vertraulichkeit, Integrität und Authentizität, einer Nachricht, die über ein RPC1 Alarmierungsnetzwerk übertragen wird. Er stellt sicher, dass ein unbefugter Empfänger nicht in der Lage ist eine Nachricht zu decodieren oder zu verändern, ohne dass der berechtigte Empfänger dies bemerkt. Die Verschlüsselung erfolgt in der Regel beim Absender (DAG oder ELS), die Dekodierung im Endgerät und garantiert so eine Ende-zu-Ende-Verschlüsselung. Zur Härtung des Verfahrens nutzt der Algorithmus, außer für die schnelle Textalarmierung, keine einheitlichen Netzwerkschlüssel sondern individuelle Schlüssel für jeden RIC / Unteradresse.

Dies hat folgende Vorteile:

- Sollte einer der Schlüssel kompromittiert werden sind die anderen Aussendungen immer noch sicher.
- Der Aufwand für die Neuprogrammierung der kompromittierten Empfänger ist deutlich geringer. Bei bekannt werden eines Netzschlüssels müssten alle DME eines Landkreises neu programmiert werden
- DME Schlüssel können auf Gemeindeebene oder Abteilungsebene verwaltet werden. Die Zahl der Personen die Zugriff auf alle Schlüssel haben kann klein gehalten werden.

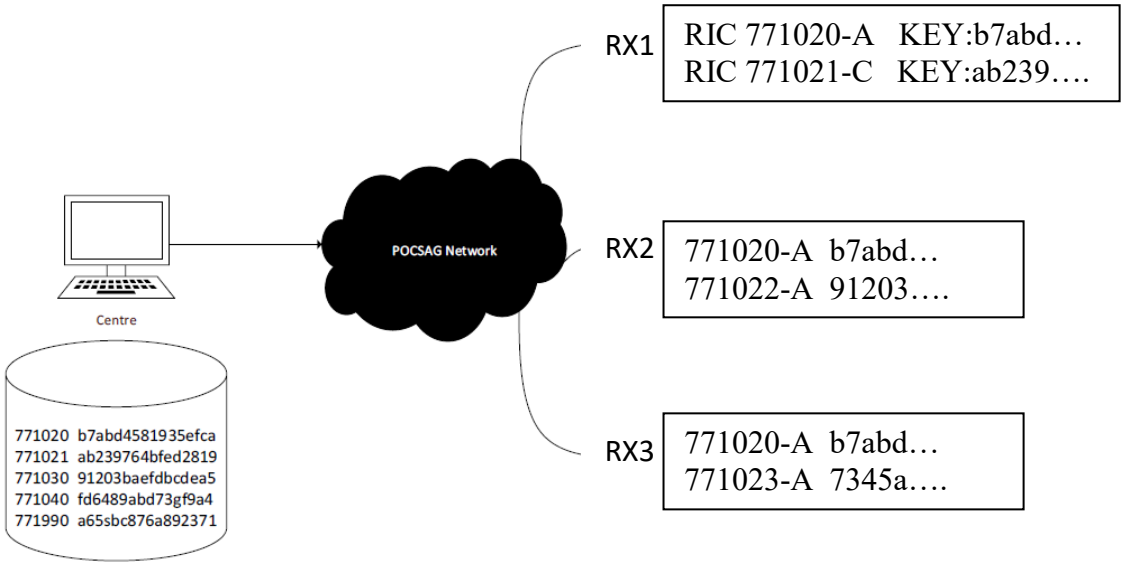
Sobald die Verschlüsselung, wie in Abschnitt B7 beschrieben durchgeführt wurde, kann die Nachricht über das Netz übertragen werden. Alle mit diesem RIC programmierte DME (im Beispiel alle DME) erhalten die Meldung und entschlüsseln sie unter Verwendung der intern gespeicherten Schlüssel.

Ein unbefugter Empfänger kann zwar feststellen welcher RIC alarmiert wurde und vermutlich auch irgendwann den richtigen Empfänger zuordnen, der Textinhalt bleibt ihm aber verborgen. Wenn die empfangene verschlüsselte Nachricht einige wenige Fehler enthält, wird sie trotzdem angezeigt. Da der Algorithmus den CTR und nicht den CBC-Modus von AES nutzt, wird nur das Zeichen welches Bitfehler enthält beschädigt. Daher können auch Nachrichten mit Übertragungsfehlern aus dem Kontext heraus von den Benutzern verstanden werden. Der Empfänger muss den Benutzer aber darauf hinweisen dass die Nachricht Übertragungsfehler enthält. Insofern unterscheidet sich das Verhalten und die Anzeige nicht vom bisherigen unverschlüsselten Empfang.

In Fällen, in denen aufgrund technischer Probleme keine Verschlüsselung möglich ist, kann der Empfänger im Rückfallmodus auch unverschlüsselte Meldungen verarbeiten. Die Texte können dabei auf den gleichen RIC wie die verschlüsselte Übertragung gesendet werden, da der Algorithmus in der Lage ist, verschlüsselte und unverschlüsselte Nachrichten zu unterscheiden.

Verschlüsselte Texte sind prinzipbedingt länger als Klartextmeldungen, weil sie weitere Informationen enthalten. Um die Länge der verschlüsselten Texte möglichst kurz zu halten wird der in Abschnitt B10 beschriebene Komprimierungsalgorithmus verwendet.

B2. Grundsätzlicher Aufbau des Gesamtsystems



Wie aus vorstehender Zeichnung ersichtlich, enthält der Alarmgeber eine Datenbank in der zu jedem RIC ein entsprechender Chiffrierschlüssel gespeichert ist. Wenn ein Alarm an den RIC 771020-A gesendet werden soll, verschlüsselt der Alarmgeber den Text mit dem zugeordneten Schlüssel „b7abd45819…..". Alle Empfänger die mit diesem RIC /UA programmiert sind müssen dabei über den gleichen Schlüssel verfügen.

B3. Schlüssellänge

Die Schlüssellänge je RIC / Unteradresse beträgt 256 Bit.

B4. Schlüsselzuordnung beim Alarmgeber

Alarmgeber müssen in der Lage sein für jeden RIC und Unteradresse einen Schlüssel zu speichern. Für Alarmierungen mit schneller Textalarmierung ist ein getrennter Speicher mit min. 256 individuellen Schlüsseln vorzusehen. Der gemischte Betrieb muss möglich sein.

B5. Schlüsselzuordnung beim Empfänger

Durch bereitgestellte Hilfsmittel, in der Regel eine PC Software, muss der Anwender die Möglichkeit haben jedem RIC / UA einen Schlüssel zuzuordnen. Die Zuordnung sollte möglichst automatisiert erfolgen. In Endgeräte dürfen keine Schlüssel programmiert werden deren RIC nicht angelegt sind bzw. die nicht benötigt werden.

B6. Algorithmus

Nachfolgend wird der Ver- und Entschlüsselungsalgorithmus beschrieben.

B6.1 Verschlüsselung

• C ist der Klartext ggf. ergänzt um Füllzeichen (siehe B.19),

• IV ist der Startvektor (IV = Initialization vector)

• K ist der Schlüssel (K=Key) für einen bestimmten RIC / Unteradresse,

• E ist der verschlüsselte Text (encrypted text)

• H ist eine berechnete Prüfsumme unter Verwendung von SHA1 (H=Hash)

• R ist die Zusammensetzung der Zeichenfolge “ENCR” und des Klartextes C in der komprimierten Form (siehe weiter unten)

• T ist die gesendete Textmeldung,

• SYMenc ist die symmetrische Verschlüsselungsfunktion und “SYM $_{dec}$“ ist die symmetrische Entschlüsselungsfunktion im CTR Modus unter Verwendung von AES256,

• “BINARY2TEXT $_{enc}$” ist die Funktion zur Wandlung im Sendefall, “BINARY2TEXT $_{dec}$” ” ist die Funktion zur Wandlung im Empfangsfall,

• und SHA1 die Funktion zur Berechnung der kryptografischen Prüfsumme (Hash).

• und CRC-8 die Funktion zur Berechnung der Prüfsumme der Systemwerte im IV.

Die Verschlüsselung arbeitet dann wie folgt:

1) Alle Zeichen werden gemäß POCSAG Codetabelle gewandelt, dies betrifft insbesondere die deutschen Umlaute, falls genutzt, werden zufällige Füllzeichen ergänzt (siehe B19).

2) Die Prüfsumme H = SHA1(C) wird berechnet (C ohne die nach B.19 ergänzten Füllzeichen)

3) Der Textstring “ENCR” und der Klartext C werden verkettet

4) Die Kombination aus 3) wird komprimiert (siehe auch B.8)

5) Die Prüfsumme CRC = CRC-8 (aus den ersten 56 Bit des IV) wird berechnet

6) Der IV wird aus Zeitstempel, Schlüsselindex, Zufallsbits und CRC-8 zusammengesetzt

Zeitstempel	Schlüsselindex	Zufallsbits	CRC-8	SHA-1	„ENCR“	Alarmtext*
32 Bit	8 Bit	16 Bit	8 Bit	40 Bit	28 Bit	0 .. n
IV				Prüfsumme	Kennung	komprimiert

* Die Anzahl der Bits ist vom konkreten Alarmtext abhängig. Die Geräte müssen so ausgelegt sein, dass sie eine (Mindestbitzahl=Mindesttextlänge) von 840 Bit verarbeiten können. Die (Höchstbitzahl=Maximaltextlänge) wird auf 1260 Bit festgesetzt.

7) Berechnung von: $E = SYM_{enc}(K, IV, H \,||\, R)$.

8) Wandlung von : $T = BINARY2TEXT_{enc}(IV \,||\, E)$.

9) Aussendung von T über das Funknetz.

B6.2 Entschlüsselung

D ist die komplette entschlüsselte Übertragung einschließlich der Verschlüsselungskennung "ENCR" und der Hash-Summe (D = H || R). HD ist die neu berechnete Hash-Summe. Die anderen Symbole sind identisch wie im Abschnitt Verschlüsselung beschrieben.

1) Wenn T ein Zeichen enthält das nicht vom BINARY2TEXT Algorithmus unterstützt wird, ist zu prüfen ob die POCSAG-Prüfsumme (also der CRC im 32 Bit RPC1 Codewort, nicht die SHA1 Prüfsumme) für dieses Zeichen ungültig ist. Falls ja wird das fehlerhafte Zeichen durch ein gültiges ersetzt (z.B. ein „+“ Zeichen). Ansonsten, bei richtiger RPC1-Prüfsumme, aber nach wie vor nicht vorhandenem Zeichen im BINARY2TEXT Algorithmus ist die Übertragung möglicherweise nicht verschlüsselt und könnte angezeigt wie empfangen werden.

2) Rückwandlung von: (IV, E) = BINARY2TEXTdec (T).

3) Prüfung des CRC-8

4) Berechnen von : D = SYMdec (K, IV, E)

5) Dekomprimieren von R aus D (alles in D außer H, siehe auch Abschnitt B8).

6) Prüfen ob der Text in R mit "ENCR" beginnt. Wenn dies nicht der Fall ist wurde die Nachricht wahrscheinlich nicht verschlüsselt und es kann die ursprüngliche empfangene Nachricht angezeigt werden.

7) Ableiten von C aus R.

8) Entfernen aller Füllzeichen am Ende von C.

9) Berechnen von HD = SHA1 (C).

10) Für die weitere Behandlung sind mehrere Fälle zu unterscheiden, die zur Verbesserung der Übersicht im Nachgang unter 5.2.1 näher erläutert werden.

11) Rückwandlung aller Zeichen in den POCSAG-Zeichensatz.

12) Anzeigen von C.

13) Wenn der aus dem IV abgeleitete Zeitstempel eine zu große Differenz aufweist, wird die Meldung (im Empfänger konfigurierbar) entweder

- mit einer Warnung versehen und normal signalisiert
- nur optisch angezeigt
- nur eine Warnung angezeigt
- oder komplett verworfen

Was als zu große Abweichung gilt, ist ein Systemparameter der durch die Anwender individuell festgelegt werden kann (siehe B6).

B6.2.1 Behandlung von Prüfsummenfehlern

Für den Anwender wählbar sollte zwischen dem ausschließlich verschlüsselten und dem Mischbetrieb (verschlüsselt / unverschlüsselt) unterschieden werden können. Die Auswahl kann für jeden RIC individuell wählbar sein. Ohne Auswahlmöglichkeit ist Mischbetrieb vorzusehen.
Es ist auch denkbar die einzelnen Kombinationen der nachfolgenden Tabelle im Empfänger konfigurierbar zu gestalten. Dadurch können über eine Einstellmöglichkeit alle Kombinationen abgebildet werden.

Fall	SHA1 gültig?	CRC-8 gültig?	ENCR erkannt	Modus nur Krypto	Mischbetrieb
1	J	J	J	entschlüsselt anzeigen	entschlüsselt anzeigen (1)
2	J	J	N	nicht anzeigen	direkt anzeigen
3	J	N	J	nicht anzeigen	direkt anzeigen
4	J	N	N	nicht anzeigen	direkt anzeigen
5	N	J	J	nicht anzeigen	entschlüsselt anzeigen (2)
6	N	J	N	nicht anzeigen	direkt anzeigen
7	N	N	J	nicht anzeigen	direkt anzeigen
8	N	N	N	nicht anzeigen	direkt anzeigen

(1) Dies ist der Idealfall der fehlerfreien Übertragung
(2) Dieser Fall kann auftreten wenn ein Empfang fehlerfrei beginnt und dann vor dem Ende abbricht. Die Wahl „entschlüsselt anzeigen“ sichert dann die Darstellung des Textes soweit er empfangen werden konnte.

B7. Zeitsynchronisation zwischen Sender und Empfänger

Einige Funktionen des Verfahrens erfordern eine zeitliche Synchronität zwischen Sender und Empfänger. Die aktuelle Ortszeit ist dabei nicht erforderlich solange Sender und Empfänger über die gleiche Zeit verfügen. Aus praktischen Gründen bietet es sich an UTC zu nutzen. So sind bei der Sommer / Winterzeit keine Zeitsprünge zu beachten. Die richtige Anzeige von Datum/Uhrzeit in der Ortszeit bleibt dann dem Endgerät vorbehalten.

Die Zeit, nach der ein Empfänger eine Alarmierung als zeitlich verfristet ansieht, kann im Endgerät konfigurierbar gestaltet werden. Die Programmiersoftware soll dann 30 Minuten als Standardwert vorgeben. Sie darf nicht unter 10 Minuten einstellbar sein um Uhrenfehler abzufangen. Die Funktion muss abschaltbar sein. Die Empfänger prüfen das Zeitintervall nach B11 in beide Richtungen, d.h im Falle des Standardwertes auf +- 30 Minuten.

B8. Zeichenkodierung

Alle Werte größer acht Bit werden in der Little-Endian-Notation gespeichert. Dies bedeutet, dass das höchstwertige Byte in der äußersten rechten Position (bei horizontaler Schreibweise) gespeichert wird. Übertragen werden die Bytes dann von links nach rechts mit dem MSB zuerst. Abweichend davon wird für die SHA1-Prüfsumme in die Big-Endian-Notation verwendet, so wie in FIPS 180-4 vorgeschlagen.

B9. Komprimierung und Dekomprimierung der Texte

POCSAG unterstützt nur einen 7-Bit-Zeichensatz. Zeichen, die mit acht Bit kodiert sind, haben also ein redundantes Bit. Zur Redundanzreduktion wird deshalb der folgende Algorithmus angewendet:

1. Überprüfung ob alle Zeichen einen ASCII Wert unter 128 dez. haben. Zeichen über 128 dez. werden in ein Ersatzzeichen unter 128 dez. gewandelt (siehe Tabelle Anhang A).

2. Das 8. Bit der einzelnen Zeichen wird entfernt. Dieses muss immer 0 sein, da nur ASCII-Werte unter 128 dez. zulässig sind bzw. vorher nach 1. gewandelt wurde.

3. Die verbleibenden Bits werden gemäß Beispiel unten zusammen geschoben / verkettet.

4. Aus den verketten Bits werden neue Bytes (8 Bit Werte) gebildet.

5. Falls das letzte Byte keine vollen acht Bit mehr enthält werden 0 – Füllbits ergänzt.

Beispiel
Der Text "This is a `Test.`" soll komprimiert werden.
Hinweis: Der Punkt am Ende hat normal einen ASCII Wert von 46 dez. , im hier verwendeten UK POCSAG Zeichensatz aber 94 dez.

Char	ACSII dec	ASCII bin (big-endian)	ASCII bin (little-endian)
T	84	01010100	00101010
h	104	01101000	00010110
i	105	01101001	10010110
s	115	01110011	11001110
	32	00100000	00000100
i	105	01101001	10010110
s	115	01110011	11001110
	32	00100000	00000100
a	97	01100001	10000110
	32	00100000	00000100
t	116	01110100	00101110
e	101	01100101	10100110
s	115	01110011	11001110
t	116	01110100	00101110
.	94	01011110	01111010

ASCII bin	step 2 (7 bit)	final result bin	final result dec
00101010	0010101	00101010	84
00010110	0001011	00101110	116
10010110	1001011	01011110	122
11001110	1100111	01110000	14
00000100	0000010	01010010	74
10010110	1001011	11110011	207
11001110	1100111	10000010	65
00000100	0000010	10000110	97
10000110	1000011	00001000	16
00000100	0000010	10111101	189
00101110	0010111	00111100	60
10100110	1010011	11100101	167
11001110	1100111	11011110	123
00101110	0010111	10000000	1
01111010	0111101	*saved space*	*saved space*

Wie in der Tabelle oben gezeigt werden die Bits eines nachfolgenden Bytes „links“ heraus geschoben, um beim vorhergehenden rechts angefügt zu werden. Beim ersten Byte wird nur ein Bit angefügt, beim zweiten zwei usw. (abwechselnd rot und grün markiert). Nach sieben Schritten ist ein (7 Bit) Wert komplett in den vorhergehenden Bytes aufgegangen. Das letzte Byte wird mit 0 Bits aufgefüllt, um eine Länge von 8 Bit zu erhalten.

B10. Binär zu Text Wandlung / Base 64

BOSKRYPT nutzt BASE64 für die Wandlung von binär zu Text. Die Wandlung ist erforderlich, da manche Alarmierungssysteme nur druckbare Zeichen übertragen bzw. Steuerzeichen, insbesondere 04=ETX zu unerwünschten Effekten führen können.
BASE64 wird in RFC 4648 beschrieben Er verwendet einen sehr begrenzten Zeichensatz. Daher sind verschlüsselte Aussendungen in der Regel länger, weil es vier Zeichen erfordert um drei Bytes darzustellen.

B11. Vermeidung von Manipulationen durch wiederholte Aussendung

Durch Aufzeichnung einer Aussendung und spätere Wiederaussendung könnte eine ungewollte Alarmierung durchgeführt werden, auch wenn die Texte verschlüsselt sind und der Nachrichteninhalt nicht bekannt ist. Um das zu vermeiden wird der Initialisierungsvektor (IV) durch einen Zeitstempel

ergänzt. Ein unbefugter Empfänger kann selbst dann keine Nachrichteninhalte gewinnen, wenn er in der Lage wäre den Zeitstempel zu lesen. Eine Manipulation des Zeitstempels und erneute Übertragung scheitert daran, dass der Zeitstempel ein Teil des IV ist und eine Änderung dazu führt das der Text nicht entschlüsselt werden kann.

Der Algorithmus nutzt 32 Bits des IV für den Zeitstempel. Er ist in Sekunden ab dem 1. Januar 2014 00.00.00 in UTC codiert. Der Zeitstempel hat damit eine Laufzeit bis zum 7. Februar 2150 06:28:15 Uhr.

Ein Empfänger darf einen (erneuten) Empfang einer Alarmierung mit falscher Zeitinformation nicht grundsätzlich ignorieren. Durch Konfiguration am Endgerät soll festgelegt werden können, ob er unterdrückt oder dem Anwender signalisiert wird. Der Anwender ist über eine geeignete Darstellung am Endgerät auf die Zeitabweichung hinzuweisen. Es ist dann Aufgabe des Nutzers anhand des Textes über die Relevanz zu entscheiden.

Als Anzeige sollen bevorzugt der Text „ZEIT“, ggf. durch Attribute wie Blinken oder Inversdarstellung unterstützt, oder ein leicht verständliches Symbol wie eine Uhr oder Sanduhr eingesetzt werden.

B12. Initialisierungsvektor

Der Initialisierungsvektor ist 64 Bit lang und besteht aus einem 32-Bit-Zeitstempel und einem weiteren 32-Bit-Wert der sich aus den drei Teilen Schlüsselindex, Zufallsbits und CRC-8 zusammensetzt. Der Zeitstempel wird in Sekundenschritten ab 1. Januar 2014 00.00.00 in UTC codiert. Beispiel: Der 1. Dezember 2014 14:29:11 wird als B720B901 codiert. Kombiniert mit einem Zufallswert könnte ein typischer IV wie folgt aussehen: B720B90100FE1AF8, wobei B720B901 der Zeitstempel und 00FE1AF8 ein aus Schlüsselindex =00, Zufallswert =FE1A und CRC-8 =F8 zusammengesetzter Wert ist. Hauptsächlich soll der Initialisierungsvektor vermeiden, dass zwei gleiche Klartexte C, die mit dem gleichen Schlüssel K verschlüsselt sind, zum gleichen verschlüsselten Text E führen.

$$SYM_{enc}(K, IV_1, C) \neq SYM_{enc}(K, IV_2, C), \qquad if\ IV_1 \neq IV_2$$

Da die Größe des Initialisierungsvektors begrenzt ist, besteht eine gewisse Wahrscheinlichkeit, dass bei der Generierung von Zufallswerten nach einer Weile zu einer Kollision kommt und derselbe

Initialisierungsvektor wieder verwendet wird. Die Wahrscheinlichkeit einer solchen Kollision kann berechnet werden:

$$P(m,n) = 1 - e^{-\left(\frac{n^2}{2m}\right)} \qquad (2)$$

Quelle: [2] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone: "Handbook of Applied Cryptography"; 1997 CRC Press; page 53

wobei m der maximale Wert des Initialisierungsvektor und n die Anzahl der Übertragungen pro Sekunde ist.

Da der zweite Teil des Initialisierungsvektor aus dem Zeitstempel besteht, ist die Wahrscheinlichkeit einer Kollision gering, vor allem wenn man betrachtet dass POCSAG-Übertragungen langsam sind und nicht mehr als eine (Text)Nachricht pro Sekunde möglich ist.

Der gleiche IV mit einem anderen RIC ist meist unproblematisch, weil unterschiedliche RIC in der Regel unterschiedliche Schlüssel haben.

$$SYM_{enc}(K_1, IV, C) \neq SYM_{enc}(K_2, IV, C), \qquad if\ K_1 \neq K_2$$

Bei der Wahl der Größe der Initialisierungsvektor ist zu beachten dass es unerlässlich ist, dass dieser fehlerfrei übertragen wird. Bereits ein einzelner, durch den RPC1 Code nicht korrigierbarer Bitfehler ,verhindert eine erfolgreiche Entschlüsselung. Daher sollte die Größe des IV so kurz, wie im Hinblick auf die Sicherheit annehmbar, gehalten werden. Er wird durch den CRC-8 gesichert.

B13. Schlüsselindex

Bei der Anwendung von Verfahren zur zeitlichen Optimierung von gleichen Textmeldungen an verschiedene, dynamisch gebildete Empfängergruppen wird die Textinformation über einen speziell definierten, immer gleichen Text RIC übertragen. Der Text könnte wie jede andere Alarmierung über das hier beschriebene Verfahren gesichert werden. Dafür würden dann nur die vier Schlüssel der Unteradressen A..D zur Verfügung stehen. Um den Schlüsselraum zu erweitern wird im IV ein 8 Bit Index übertragen über den der zugehörige Schlüssel zur Dekodierung ausgewählt wird. Da dieser Index bei jedem Alarm individuell übertragen wird, entfällt die Notwendigkeit einer manuellen oder automatisch getrennt durchgeführten Schlüsselumschaltung. Damit besteht auch nicht die Gefahr, dass Empfänger die ausgeschaltet oder außerhalb des Empfangsbereiches waren, Kommandos zur Schlüsselumschaltung verpassen. Empfänger wenden diesen Index nur auf den Text RIC der schnellen Textalarmierung an. Sender setzen ihn außerhalb der schnellen Textalarmierung auf 00. Der Schlüsselindex ist 8 Bit lang, das MSB wird zuerst übertragen.

B14. Sicherheitshinweis zum IV, Belegung der Zufallsbits

Durch konstruktive Maßnahmen innerhalb der Verschlüsselungssoftware ist sicherzustellen dass ein IV nur einmal genutzt wird. Da sich der IV mit dem Zeitstempel automatisch jede Sekunde ändert und z.Z 16 Zufallsbits möglich sind, ist dies einfach zu erreichen. Eine Verletzung dieser Vorgabe behindert nicht die Alarmierung, eröffnet aber eine Sicherheitslücke.

Die, zur Zeit 16 Bits die innerhalb des IV zufällig belegt werden sollen, können durch die Verschlüsselungssoftware nach einem beliebigen Verfahren ermittelt werden. Das Verfahren muss nicht offen gelegt werden da es für die Dekodierung nicht erforderlich ist.

B15. CRC-8-Prüfsumme

Die CRC-8-Prüfsumme soll die Integrität der Verfahrensvariablen des IV gewährleisten. Ein Empfänger erkennt an einem falschen CRC einen Fehler, bei den zur erfolgreichen Entschlüsselung erforderlichen Werten. Er hat keine wesentliche kryptografische Relevanz, hilft aber fehlerhafte Darstellungen („Zeichensalat“) zu vermeiden. Eingesetzt wird das international für Telekommunikationsanwendungen bekannte CRC-8. Es ist für die hier vorgesehene Anwendung

ausreichend sicher und entsprechend kurz. Außerdem lässt es sich auch mit leistungsschwachen Mikrocontrollern einfach und schnell berechnen.

Das Generatorpolynom lautet: $(x+1)\ (x^7+x^6+x^5+x^4+x^3+x^2+1)$

Entsprechend modulo 2 ausmultipliziert: x^8+x^2+x+1

Der Startwert vor jeder Berechnung ist 00000000. Der berechnete CRC-8 bildet die letzten 8 Bit des IV. Das MSB wird zuerst übertragen.

B16. SHA1-Prüfsumme

Der SHA1 Hashwert ist gemäß Definition 160 Bit lang. Übertragen werden jedoch nur die ersten 40 Bits (MSB).

B17. Meldungsvergleich

Die Eigenschaft innerhalb einer (programmierbaren) Frist eingegangene identische Textmeldungen nur einmal zu signalisieren muss auch bei Anwendung der Verschlüsselung erhalten bleiben. Der Vergleich erfolgt grundsätzlich mit den entschlüsselten Texten (Klartextvergleich).

B18. Länge verschlüsselter Texte

Die Länge einer Übertragung wird durch die folgenden Faktoren gegenüber dem Klartext erhöht:

1) Ergänzung des ASCII-String "ENCR" (28 Bit)
2) den Initialisierungsvektor (64 Bit)
3) die Hash-Summe (40 Bit)
4) durch Füllzeichen
4) durch die (Um)codierung und Übertragung mit BASE64

Die Klartextlänge wird auf 180 Zeichen begrenzt. Geräte für das BOSKRYPT Verfahren müssen mindestens 120 Klartextzeichen verarbeiten können.

B19. Füllzeichen

BOSKRYPT ermöglicht es durch mengenmäßig zufällig gewählte Füllzeichen die Textlänge einer Alarmierung zu variieren obwohl immer die gleiche Nutzdatenlänge übertragen wird. Bei Nutzung dieser Funktion wird durch den Algorithmus eine zufällige Zahl von (binären, 8 Bit) Nullbyte Werten (hex 00) an das Ende des Klartextes angehängt.

Da sich durch diese Option die Aussendezeiten verlängern, sollten durch den Sender folgende Optimierungen möglich sein:

- Abschalten der Funktion durch Konfiguration des Anwenders
- Einsatz von Füllzeichen nur bei Texten unter 30 Zeichen
- Begrenzung der (Zufalls)Zahl auf einen Wert zwischen 0 bis 30

C. Steuerfunktionen

C1. Zeit / Datum

Die Empfänger sollen eine automatische Einstellung des Datums/Zeit über Funk durch spezielle Textinhalte ermöglichen. Dafür können die bereits eingeführten Verfahren der einzelnen Hersteller oder dass nachfolgend beschriebene Verfahren zum Einsatz kommen.

Auf einen, netzweit einheitlichen RIC, wird ein Text gesendet wie er nachfolgend beschrieben ist:

#ZEIT=HHmmddMMyy#ZEIT=HHmmddMMyy

Beispiel für 17:23 Uhr am 03.11.2015 #ZEIT=1723031115#ZEIT=1723031115

Hinweis:
Nachfolgend wird nur noch von Zeit gesprochen, es ist aber immer die Kombination aus Datum und Uhrzeit gemeint.

Ein einheitlicher RIC ist erforderlich um die Netzlast bei der Verteilung möglichst gering zu halten. Die Netzinfrastruktur muss diesen RIC regelmäßig aussenden, empfohlen werden einmal pro Stunde. Im Fall einer unverschlüsselten Aussendung wird die Unteradresse A genutzt, für eine verschlüsselte Aussendung die Unteradresse D.

Verhalten der Empfänger

1. Ein Empfänger prüft den RIC auf die obige Zeichenkombination. Zur Sicherung wird die Information gedoppelt, beide Zeitstempel müssen gleich sein, sonst muss der Empfänger die Meldung verwerfen.

2. Eine nach 1. korrekte, über Unteradresse= D verschlüsselt übertragene Zeitinformation wird dazu genutzt die interne Uhr zu stellen.

3. Beträgt die Abweichung mehr als 30 Minuten soll der DME Träger die Umstellung bestätigen. Auf die erforderliche Bestätigung ist der Anwender entsprechend der Einstellungen akustisch und optisch hinzuweisen. Die Bestätigung entfällt bei Sirenensteuerempfängern (DSE).

Die Übertragung mit UA=A (unverschlüsselt) sollte nur in Ausnahmefällen erfolgen oder wenn die Zeitprüfung nicht genutzt wird.

C2. Empfänger sperren

C2.1 - Teil Empfänger

Durch Empfang einer vordefinierten Zeichenfolge (Sperrwort) soll ein Empfänger dahingehend deaktiviert werden, dass er keine weiteren Alarme signalisiert. Der Empfang muss weiterhin möglich sein um ggf. ein Freigabewort zu empfangen. Der Empfang dieses Sperrwort muss auf allen programmierten RIC möglich sein. Das Sperrwort darf nur aus druckbaren Zeichen des POCSAG Zeichensatzes bestehen und darf nur nach fehlerfreier verschlüsselter Übertragung genutzt werden. Das Sperrwort muss möglichst für einen Empfänger einmalig sein und 16 Zeichen haben. Es kann vereinfacht aus der erweiterten bzw. gekürzten Seriennummer oder einen herstellerspezifischen Algorithmus durch die Programmiersoftware ermittelt und vorgeschlagen werden. Eine manuelle Änderung muss möglich sein. Die Programmiersoftware sollte eine Möglichkeit haben einen Empfänger auch direkt zu sperren ohne den Umweg über den HF Empfang zu gehen.
Dem Sperrwort wird das Kommando „SPERR= “vorangestellt. Das Sperrwort darf über die Programmiersoftware nur in das Endgerät geschrieben werden wenn die Programmiersoftware keinen weiteren Auslesschutz, z.B. über Passwort, enthält.

Beispiel: SPERR=17dEor#3/do1amdS

Die Entsperrung kann über eine Neuprogrammierung oder das Entsperrungsverfahren nach C3.1 erfolgen. Die Implementierung dieser Funktion ist den Herstellern freigestellt und im Datenblatt anzugeben.

C2.2 - Teil Sender

Für die Aussendung sind keine besonderen Maßnahmen erforderlich. Jeder DAG, der in der Lage ist verschlüsselte Aussendungen nach dieser Spezifikation durchführen, kann für die Steuerkommandos nach Teil C dieser Beschreibung eingesetzt werden.

C3. Empfänger freigeben

C3.1 - Teil Empfänger

Durch Empfang einer vordefinierten Zeichenfolge (Entsperrwort) soll ein Empfänger wieder aktiviert werden können. Der Empfang dieses Entsperrwort muss auf allen programmierten RIC möglich sein. Das Sperrwort und der verschlüsselte Empfang ist zu dem unter C2.1 identisch. Dem Sperrwort wird das Kommando „ ENTSPERR= “vorangestellt.

Beispiel: ENTSPERR=17dEor#3/do1amdS

Die Implementierung dieser Funktion ist den Herstellern freigestellt und im Datenblatt anzugeben.

C3.2 - Teil Sender

Für die Aussendung gilt das unter C2.2 angeführte.

Anhang A - Codetabelle

DEC	HEX	French	German	Italian	Swiss	UK	US
32	20	<space>	<space>	<space>	<space>	<space>	<space>
33	21	!	!	!	!	!	!
34	22	"	"	"	"	"	"
35	23	£	#	£	ù	£	#
36	24	$	$	$	$	$	$
37	25	%	%	%	%	%	%
38	26	&	&	&	&	&	&
39	27	'	'	'	'	'	'
40	28	(	(	(	(	(	(
41	29	)	)	)	)	)	)
42	2A	*	*	*	*	*	*
43	2B	+	+	+	+	+	+
44	2C	,	,	,	,	,	,
45	2D	-	-	-	-	-	-
46	2E	.	.	.	.	.	.
47	2F	/	/	/	/	/	/
48	30	0	0	0	0	0	0
49	31	1	1	1	1	1	1
50	32	2	2	2	2	2	2
51	33	3	3	3	3	3	3
52	34	4	4	4	4	4	4
53	35	5	5	5	5	5	5
54	36	6	6	6	6	6	6
55	37	7	7	7	7	7	7
56	38	8	8	8	8	8	8
57	39	9	9	9	9	9	9
58	3A	:	:	:	:	:	:
59	3B	;	;	;	;	;	;
60	3C	<	<	<	<	<	<
61	3D	=	=	=	=	=	=
62	3E	>	>	>	>	>	>
63	3F	?	?	?	?	?	?
64	40	À	§	§	à	@	@
65	41	A	A	A	A	A	A
66	42	B	B	B	B	B	B
67	43	C	C	C	C	C	C
68	44	D	D	D	D	D	D
69	45	E	E	E	E	E	E
70	46	F	F	F	F	F	F
71	47	G	G	G	G	G	G
72	48	H	H	H	H	H	H
73	49	I	I	I	I	I	I
74	4A	J	J	J	J	J	J
75	4B	K	K	K	K	K	K
76	4C	L	L	L	L	L	L

DEC	HEX	French	German	Italian	Swiss	UK	US
77	4D	M	M	M	M	M	M
78	4E	N	N	N	N	N	N
79	4F	O	O	O	O	O	O
80	50	P	P	P	P	P	P
81	51	Q	Q	Q	Q	Q	Q
82	52	R	R	R	R	R	R
83	53	S	S	S	S	S	S
84	54	T	T	T	T	T	T
85	55	U	U	U	U	U	U
86	56	V	V	V	V	V	V
87	57	W	W	W	W	W	W
88	58	X	X	X	X	X	X
89	59	Y	Y	Y	Y	Y	Y
90	5A	Z	Z	Z	Z	Z	Z
91	5B	°	Ä	°	é	[	[
92	5C	Ç	Ö	ç	ç	\	\
93	5D	§	Ü	é	ê	]	]
94	5E	^	^	^	^	^	^
95	5F	_	_	_	è	_	_
96	60	´	´	´		´	´
97	61	A	a	a	a	a	a
98	62	B	b	b	b	b	b
99	63	C	c	c	c	c	c
100	64	D	d	d	d	d	d
101	65	E	e	e	e	e	e
102	66	F	f	f	f	f	f
103	67	G	g	g	g	g	g
104	68	H	h	h	h	h	h
105	69	I	i	i	i	i	i
106	6A	J	j	j	j	j	j
107	6B	K	k	k	k	k	k
108	6C	L	l	l	l	l	l
109	6D	M	m	m	m	m	m
110	6E	N	n	n	n	n	n
111	6F	O	o	o	o	o	o
112	70	P	p	p	p	p	p
113	71	Q	q	q	q	q	q
114	72	R	r	r	r	r	r
115	73	S	s	s	s	s	s
116	74	T	t	t	t	t	t
117	75	U	u	u	u	u	u
118	76	V	v	v	v	v	v
119	77	W	w	w	w	w	w
120	78	X	x	x	x	x	x
121	79	Y	y	y	y	y	y
122	7A	Z	z	z	z	z	z
123	7B	É	ä	à	ä	{	{
124	7C	Ù	ö	ò	ö	¦	¦
125	7D	È	ü	è	ü	}	}
126	7E		ß	ì	û	~	~

Anhang B - Algorithmen und Parameter

Verschlüsselungsalgorithmus	AES (FIPS 197)
Schlüssellänge	256 Bit
Standard	Blockmode (NIST 800-38A, Kap. 5.11
Prüfung der Textintegrität	40 von 160 Bit der SHA1 Prüfsumme
Binär zu ASCII Wandlung	BASE 64 nach RFC 4648
Größe des Initialisierungsvektors	64 Bit, davon 32 Bit als Zeitstempel

Anhang C – Übersichtstabelle Verlängerung der Aussendung durch Kryptierung

1. Schritt: Über die Zeichenzahl Lc den Wert von Le AES bestimmen

		CTR mode	
L_C	L_{CC}	Saved	L_E AES
40	39	5	57
45	43	6	61
50	48	6	66
55	52	7	70
60	56	8	74
65	61	8	79
70	65	9	83
75	70	9	88
80	74	10	92
85	78	11	96
90	83	11	101
95	87	12	105
100	91	13	109
105	96	13	114
110	100	14	118
115	105	14	123
120	109	15	127
125	113	16	131
130	118	16	136
135	122	17	140
140	126	18	144
145	131	18	149
150	135	19	153
155	140	19	158
160	144	20	162
165	148	21	166
170	153	21	171
175	157	22	175
180	161	23	179
185	166	23	184
190	170	24	188
195	175	24	193
200	179	25	197
205	183	26	201
210	188	26	206

2. Schritt: Über den Wert von Le (AES), unten nur als Le bezeichnet, den Wert entsprechend dem Codierungsverfahren bestimmen

Beispiel: 100 Zeichen Klartext führen zu 109 Zeichen Le (AES) und daraus folgend bei Nutzung von BASE64 zu 148 zu übertragenden Zeichen.

L_E	BASE64	POCSAGASCII85	L_E	BASE64	POCSAGASCII85
31	44	39	123	164	154
32	44	40	127	172	159
35	48	44	128	172	160
39	52	49	131	176	164
40	56	50	136	184	170
44	60	55	140	188	175
48	64	60	144	192	180
53	72	67	149	200	187
56	76	70	152	204	190
57	76	72	153	204	192
61	84	77	158	212	198
64	88	80	160	216	200
66	88	83	162	216	203
70	96	88	166	224	208
72	96	90	168	224	210
74	100	93	171	228	214
79	108	99	175	236	219
80	108	100	176	236	220
83	112	104	179	240	224
88	120	110	184	248	230
92	124	115	188	252	235
96	128	120	192	256	240
101	136	127	193	260	242
104	140	130	197	264	247
105	140	132	200	268	250
109	148	137	201	268	252
112	152	140	206	276	258
114	152	143	208	280	260
118	160	148	216	288	270
120	160	150			

Anhang D - Änderungshistorie

01.04.16 - V1.01
Im Kapitel B.12 war das IV Beispiel nicht zum Text passend bzw. mit falschem CRC angegeben. Beispiel wurde auf den Text angepasst.

29.06.16 – V1.02
Im Kapitel C2 / C3 war bei den Beispielen nach „SPERR=“ bzw. „ENTSPERR=“ versehentlich ein Leerzeichen. Zur Klarstellung und Vereinheitlichung mit „ZEIT=“ wurde dieses entfernt.

11.10.16 – V1.03
Auf Seite 9 wurde der Verweis „(siehe B.21)“ in „(siehe B.19)“ geändert