Integritätssicherung durch Signatur im IEC-Protokoll (IEC1107, EN62056-21)

Ein Vorschlag von Wolfgang Unsöld, Ing.-Büro W.Unsöld IEC_Signatur.pdf 23.02.2015

Dieses Dokument beschreibt ein Verfahren, das höchste Integritätssicherung garantiert, einfach zu handhaben ist, aber keine Verschlüsselung ausführt.
Dieses Verfahren ist voll kompatibel im bestehenden IEC-Protokoll implementierbar, sämtliche Abruf- und Übertragungsabläufe können erhalten bleiben, da der Telegrammrahmen und somit die Transportschicht nicht verändert werden.
Es wird innerhalb des Telegrammrahmens eine Signatur eingebaut, die über die Nutzdaten erstellt wurde.
Die Signierung erfolgt direkt in der Datenquelle. Die Verifizierung des IEC-Telegramms kann beim Datenempfänger sowohl beim Empfang, als auch jederzeit später, auch nach Archivierung des IEC-Telegramms, erfolgen.

Als Sicherungsverfahren wurde exakt das selbe Verfahren ausgewählt, das vom DSfG-Arbeitskreis für das DSfG-Protokoll beschlossen wurde.

Die Datensignatur erfolgt durch Hashbildung nach RIPEMD160 und anschließender ECDSA-Signatur mit asymmetrischem Schüssel nach ANSI X9.62.
Das Ergebnis der Signatur wird als Datenblock 99. an den Nutzdaten angehängt übertragen.

Das asymmetrische Schlüsselpaar wird während der Geräteinbetriebnahme im Gerät erzeugt.
Der private Schlüssel ist nur im Gerät (Sicherheitschip) gespeichert. Es gibt keine Möglichkeit von außen an diesen ranzukommen.
Der öffentliche Schlüssel kann von berechtigten Datennutzern über den Parametrierzugang ausgelesen werden. Diese speichern eine Kopie des öffentlichen Schlüssels in ihrer Datenzentrale (Abruf+Auswertesoftware).
Der öffentliche Schlüssel besteht aus einem Kurvenpunkt, also zwei Koordinaten (Px, Py), wobei Px und Py (inklusive gegebenenfalls führender Nullen) als Langzahlen jeweils eine Länge von genau 24 Byte aufweisen.

Algorithmus:

Vor dem Telegrammende ‚!' wird als letzter Datenblock der Block 99. eingefügt.
Die zu signierenden Daten erstrecken sich von dem Zeichen, das auf <STX> folgt bis einschließlich dem <LF>, das vor dem Block 99. steht.

Der Hashwert über diese zu signierenden Daten wird mit dem Algorithmus RIPEMD160 berechnet. Das Resultat (auch Message Digest genannt) ist ein Datenfeld von 20 Byte (160 Bits).

Nach der Berechnung des Message Digest über die zu signierenden Daten wird die eigentliche digitale Signatur mit dem Algorithmus ECDSA gemäß ANSI X9.62, 2005 berechnet.
Zur Berechnung sind die Domain-Parameter der elliptischen Kurve ANSI p192r1 einzusetzen.

Das Resultat der Berechnung der ECDSA-Signatur sind zwei Langzahlen R und S von jeweils 24 Byte Länge, gegebenenfalls inklusive führender Nullen, die beide im Datenblock 99. übertragen werden.

99.(V;R;S) V = Verfahrenkennzahl, ist immer ‘0’ bei diesem Verfahren

Zur Abbildung des Signaturergebnisses wird jeweils ein Byte in zwei Hex-Zeichen dargestellt. Damit werden die 24 Byte langen Zahlen R und S jeweils mit 48 Hexzeichen, in der Folge MSB bis LSB dargestellt.

Beispiel:

```
/FTLGTDS-V02.20<CR><LF>
<CR><LF>
<STX>F(000)<CR><LF>
0.0.(Laufenburg)<CR><LF>
0.1.(12345678901234567890)<CR><LF>
0.2.(123456;2008;02.20;0003)<CR><LF>
0.4.(12345678901234567890123456789012345678901234567890123)<CR><LF>
0.5.(12345678901234567890)<CR><LF>
7.1.(01)<CR><LF>
7.2.1(12345678901234567890;2;1;m3;15;5)<CR><LF>
7.2.2(12345678901234567890;2;1;m3;60;15)<CR><LF>
7.2.3(12345678901234567890;1;0;m3;60;15)<CR><LF>
7.2.4(12345678901234567890;4;0;m3;60;15)<CR><LF>
7.6.1(596;m3/h;1;080206033000+01)<CR><LF>
7.6.2(0;m3/h;0;000000000000+00)<CR><LF>
7.6.3(0;m3/h;0;000000000000+00)<CR><LF>
7.6.4(0;m3/h;0;000000000000+00)<CR><LF>
7.6.1*01(2000;m3/h;1;080102004500+01)<CR><LF>
7.6.2*01(998;m3/h;0;080102030000+01)<CR><LF>
7.6.3*01(15;m3/h;1;080930140000+01)<CR><LF>
7.6.4*01(0;m3/h;0;080102000000+01)<CR><LF>
7.8.1(9000651.3;m3;1;080206033533+01)<CR><LF>
7.8.2(9000651.3;m3;1;080206033533+01)<CR><LF>
7.8.3(00000006;m3;1;080206033533+01)<CR><LF>
7.8.4(00000001;m3;1;080206033533+01)<CR><LF>
7.8.1*01(9000628.1;m3;5;080201050000+01)<CR><LF>
7.8.2*01(9000628.1;m3;5;080201050000+01)<CR><LF>
7.8.3*01(00000006;m3;5;080201050000+01)<CR><LF>
7.8.4*01(00000001;m3;5;080201050000+01)<CR><LF>
7.8.1*00(00000000;m3;0;080930134331+01)<CR><LF>
7.8.2*00(00000000;m3;0;080930134331+01)<CR><LF>
7.8.3*00(00000000;m3;0;080930134331+01)<CR><LF>
7.8.4*00(00000000;m3;0;080930134331+01)<CR><LF>
7.24.1(232;m3;5)<CR><LF>
7.24.2(232;m3;5)<CR><LF>
7.24.3(0;m3;5)<CR><LF>
7.24.4(0;m3;5)<CR><LF>
7.24.1*01(90006281;m3;5)<CR><LF>
7.24.2*01(90006281;m3;5)<CR><LF>
7.24.3*01(6;m3;5)<CR><LF>
7.24.4*01(1;m3;5)<CR><LF>
7.71.(4;288;06;7;<CR><LF>
080206033500+01A901@00068;00019000651.3a;00019000651.3a;000100000006a;0001000000001a<CR><LF>
080206033000+01A901@00067;00019000643.0a;00019000643.0a;000100000006a;0001000000001a<CR><LF>
080206032500+01AB01@00066;00019000634.6a;00019000634.6a;000100000006a;0001000000001a<CR><LF>
080206032228+010009@00065;00019000628.1a;00019000628.1a;000100000006a;0001000000001a<CR><LF>
080201050000+01FF01@00064;02019000628.1a;02019000628.1a;020100000006a;0201000000001a<CR><LF>
080102032622+010008@00063;00019000628.1a;00019000628.1a;000100000006a;0001000000001a<CR><LF>
080102032500+01A901@00062;00019000628.1a;00019000628.1a;000100000006a;0001000000001a<CR><LF>
)<CR><LF>
30.(080206033534+01)<CR><LF>
34.(99%;190104;080101)<CR><LF>
94.(06)<CR><LF>
99.(0;067E245A879102B273784F53EC73644826253E23A2536EF4;037546586D42AA4263B73C90230354A3105437D5C2F7
4E65)<CR><LF>
!<CR><LF>
<ETX><BCC>
```

Die Signatur wurde über die Nutzdaten von dem Zeichen ‚F' beginnend, das auf <STX> folgt, bis einschließlich dem <CR><LF>, das vor dem 99.(...) Block steht, erzeugt.

Aufbau des 99.-Datenblocks:

99.(V;R;S) V = Verfahren-Kennzahl, ‘0’ = RIPEMD160 + ECDSA-p192r1
,1’ = SHA256 + ECDSA-p192r1
R = Signaturergebniszahl R, 48 Hex-Zeichen
S = Signaturergebniszahl S, 48 Hex-Zeichen

Es können innerhalb der 99.-Blocks nach einem ;-Separator weitere firmenspezifische Datenelemente folgen, zB ein BCC-Modifikator B: 99.(V;R;S;B)