Prof. Dr. Ernst-Günter Giessmann
Institut für Informatik
Humboldt-Universität zu Berlin[1]

Berlin, 2023-04-13

Gutachten zur Äquivalenz der Nachrichten der Kommunikationsprotokolle nach *ep2* zu den *Key Blocks* entsprechend den Anforderungen der PCI PIN Security (TR-31 und TR-34)

Zusammenfassung:

Im Rahmen des ep2 Systems werden symmetrische Schlüssel mit asymmetrischen Methoden übertragen. Wenn dabei keine durch [X9 TR-34] definierten interoperablen Formate und Protokolle verwendet werden, muss nachgewiesen werden, dass die Sicherheitsanforderungen der Payment Card Industry (PCI) [PCI PIN-Security] eingehalten werden:

> **Requirement 18:** Procedures must exist to prevent or detect the unauthorized substitution (unauthorized key replacement and key misuse) of one key for another or the operation of any cryptographic device without legitimate keys.

Im Ergebnis der vorliegenden Analyse wird bestätigt, dass zur Erzeugung und Übertragung symmetrischer Schlüssel mit den asymmetrischen Verfahren von ep2 diese Anforderungen eingehalten werden.

Sicherheitsziele der PCI PIN Security

Das zu erfüllende Sicherheitsziel lautet (Control Objective 5):

> „Keys are used in a manner that prevents or detects their unauthorized usage."

Die detaillierten Anforderungen der PCI an *Key Blocks* dazu lauten (Requirement 18-3 [PCI PIN-Security, S. 58]):

> "Encrypted symmetric keys must be managed in structures called key blocks. The key usage must be cryptographically bound to the key using accepted methods.
>
> The phased implementation dates are as follows:
>
> - Phase 1 – Implement Key Blocks for internal connections and key storage within Service Provider Environments – this would include all applications and databases connected to hardware security modules (HSM). Effective date: 1 June 2019.
> - Phase 2 – Implement Key Blocks for external connections to Associations and Networks. Effective date: 1 January 2023.
> - Phase 3 – Implement Key Block to extend to all merchant hosts, point-of-sale (POS) devices and ATMs. Effective date: 1 January 2025.
>
> Acceptable methods of implementing the integrity requirements include, but are not limited to:
>
> - A MAC computed over the concatenation of the clear-text attributes and the enciphered portion of the key block, which includes the key itself - e.g. TR 31;
> - A digital signature computed over that same data e.g., TR-34;
> - An integrity check that is an implicit part of the key-encryption process such as that which is used in the AES key-wrap process specified in *ANSI X9.102*."

Auf die Verwendung asymmetrischer Verfahren wird in den FAQ eingegangen:

[1] E-Mail-Adresse: giessmann@informatik.hu-berlin.de
Postanschrift: Prof. Dr. Ernst-Günter Giessmann, Am Tonberg 28, 16727 Velten

Q 19 September 2020: HSMs are required to support key blocks using the ASC X9 TR-31 key-derivation methodology for TDES keys, and for AES keys must support either the TR-31 methodology and/or the ISO 20038 methodology. TR-31 and ISO 20038 are methods to package keys (the key blocks) for conveyance or storage, but they use symmetric mechanisms for that and for key conveyance require a symmetric key exchange key that is pre-shared for use as the key block protection key. Where a symmetric key is not previously established with a POI device for remote key distribution, and asymmetric methods will be used, is it required to support a key block methodology?

A: Yes. A method such as ASC X9 TR 34: Interoperable Method for Distribution of Symmetric Keys using Asymmetric Techniques: Part 1 – Using Factoring-Based Public Key Cryptography Unilateral Key Transport must be used. Under TR-34, similar to TR-31 and ISO 20038, the Key Block consists of three parts:

- The Key Block Header (KBH) which contains attribute information about the Key and the Key Block
- The confidential data that is being exchanged/stored
- The Key Block Binding Method

However, TR-34 uses asymmetric methods for the Key Block Binding Method, instead of the symmetric methods used in TR-31 or ISO 20038 which require that a symmetric key was previously exchanged between the POI device and the KDH.

Allerdings ist es ausdrücklich zulässig, äquivalente Methoden zu verwenden ([Supp. 18-3, S.9].

Q July 2019: PIN Security Requirement 18-3 requires the implementation of key blocks. Interoperable methods include those defined in ANSI X9.143 and ISO 20038. The requirement also allows for any equivalent method whereby the equivalent method includes the cryptographic binding of the key-usage information to the key value using accepted methods. How are equivalent methods determined?

A: Equivalent methods must be subject to an independent expert review and said review is publicly available:

- The review by the independent expert must include proof that in the equivalent method the encrypted key and its attributes in the Key Block have integrity protection such that it is computationally infeasible for the key to be used if the key or its attributes have been modified. Modification includes, but is not limited to:
  - Changing or replacing any bit(s) in the attributes or encrypted key
  - Interchanging any bits of the protected Key Block with bits from another part of the block
- The PTS laboratory will validate that any device vendors implementing this methodology have done so following all guidelines of said evaluation and peer review, including any recommendations for associated key management.

Im Folgenden werden an Hand der vorgelegten Unterlagen die zu schützenden geheimen Schlüssel identifiziert und ihre Erzeugung, Übertragung und Verwendung untersucht. Dabei sind die Kriterien des ANSI Standards X9 TR-34-2012 [X9 TR-34] zugrunde gelegt worden.

Key Blocks nach TR-34

Die Datenstruktur der *key blocks* nach TR-34 [X9 TR-34X9 TR-31] verwendet die gleichen Elemente wie die TR-31/X9.143 [X9.143]:

- der *key block header* (KBH)*,* der Attributinformationen über die gesamte Struktur und den Verwendungszweck des auszutauschenden oder zu speichernden Schlüssel enthält,
- die vertraulichen Daten, die ausgetauscht oder gespeichert werden, sowie
- der kryptographische Prüfwert (*Signature*), der den *key block header* und die Schlüsseldaten aneinander bindet.

<table>
<tr><td colspan="2">← TR-31 KBH →</td><td colspan="3"></td><td colspan="2">← TR-31 KBH →</td></tr>
<tr><td>Header</td><td>Header (optional)</td><td>Version</td><td>ID_{KDH} (IssuerAndSerial Number</td><td>Key (k_n)</td><td>Header</td><td>Header (optional)</td></tr>
<tr><td colspan="2">← Plaintext (Data) →</td><td colspan="5">← Encrypted (EnvelopedData) →</td></tr>
<tr><td colspan="7">← Signed (SignedData) →</td></tr>
</table>

Für die Verschlüsselung (*EnvelopedData*) und Signaturen (*SignedData*) werden in der TR-34 die Datenstrukturen nach Cryptographic Message Syntax [X9.73] verwendet ([X9 TR-34, Kap. 5, S.23 ff.]).

Sowohl der *Key Block Header* als auch die verschlüsselten Daten haben bis ins Detail standardisierte Formate und verwenden festgelegte Bezeichner. Da der Übergang zu neuen Formaten im Feld eine gewisse Übergangszeit erfordert, sind gegenwärtig auch äquivalente Formate zugelassen.

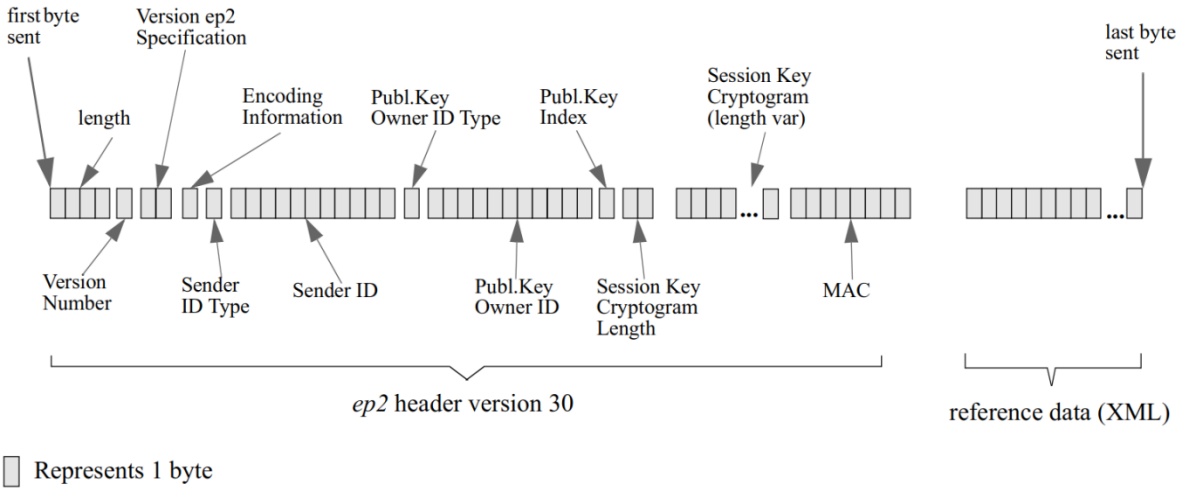
Für das Folgende sei darauf hingewiesen, dass sich die *IssuerAndSerialNumber* ID_{KDH} des (Key Distribution Host) nicht nur im unverschlüsselten Teil des *Headers*, sondern auch im verschlüsselten Signaturblock befindet. In der TR-34 heißt es dazu ([X9 TR-34, S. 25]):

> *Additionally, the ID_{KDH} is included in the encrypted data to prevent a signature stripping attack.*

Es handelt sich hier folglich um eine Sicherheitsanforderung, mit der unsignierte Schlüsselblöcke verhindert werden sollen. Das spielt im Folgenden jedoch keine Rolle, da die Integrität der verschlüsselten Blöcke nicht durch Signaturen geschützt werden. In den verschlüsselten Block ist die SenderID integriert ([SecSpec, Table 5, S. 9-35]), so dass auch bei ep2 eine Absender-Substitution ausgeschlossen wird.

Schlüsselübertragung

Wenn symmetrische Schlüssel in den verschiedenen Kommunikationskanälen übertragen werden ([SecSpec, Table 2, S. 9-15f]), dann werden sie verschlüsselt. Das Kryptogramm wird dabei in einem Block der Form



übertragen ([InterfaceSpec, Fig. 8, S. 10-31f]).

Schlüsselableitung

Die übertragenen Schlüssel werden nicht direkt verwendet, sondern es wird aus ihnen je nach vorgesehenem Einsatz mit einem Salt-Wert und dem zugehörigen Initialwert ein entsprechender Schlüssel

abgeleitet. Das Verfahren ist in [SecSpec, S. 33 und 63f] beschrieben. Zugrunde liegt dabei das allgemein anerkannte HKDF-Verfahren nach RFC5869 mit der Hash-Funktion SHA-256.

Dieses Verfahren erfüllt die gleichen Sicherheitsanforderungen an die Schlüsselableitung wie die in X9.143 vorgeschlagene CMAC-Berechnung.

Die nach den PCI-Anforderungen erforderliche Angabe des Schlüsseltyps im *Key Block Header* ist daher nicht erforderlich, da im Rahmen des ep2-Protokolls für den jeweiligen Verwendungszweck nur der entsprechende abgeleitete Schlüssel verwendet wird. Jeder durch eine eigene „Typ"-Konstante (64 Bit) und einen zufälligen geheimen Salt-Wert (256 Bit) erzeugte Schlüssel kann nur für den durch die Konstante bestimmte Zweck eingesetzt werden. Das auf der Hash-Funktion SHA-256 basierende Verfahren HKDF garantiert darüber hinaus, dass aus einem abgeleiteten Schlüssel der initiale Sitzungsschlüssel und der einzusetzende Salt-Wert nicht bestimmt werden können.

Im *Key Block Header* nach TR-31 sind verschiedene Schlüsselattribute anzugeben. Neben dem Schlüsseltyp (*type of the key* [X9 TR-31, Table 6]/[X9.143, Table 2]) gehören dazu noch der zugehörige Algorithmus ([X9 TR-31, Table 7]/]/[X9.143, Table 3]) und der Schlüsselverwendungszweck (*mode of use* [X9 TR-31, Table 8]/]/[X9.143, Table 4]). Die TR-31 lässt dabei die Schlüsselableitung als Schlüsselverwendungszweck 'X' (0x58) ausdrücklich zu:

> Key used to derive other key(s) (X) – The key is used only in a key derivation process that produces one or more derived keys ([X9.143, S. 20]).

Zusätzlich gibt es im ep2-Protokoll noch den Schlüsselverwendungszweck 'G' (0x47), der allerdings nur bei der Übertragung der Schlüssel zur Erzeugung der Surrogate PAN zum Terminal im *Key Block Header* nach X9.143 verwendet wird:

> Generate only (G) – The key can be used to generate a verification value (e.g. MAC or PIN Offset), but it cannot be used to verify the value ([X9.143, S. 20]).

Da es im ep2-Protokoll für die Sitzungsschlüssel nur einen Schlüsselverwendungszweck gibt, ist im Rahmen der Äquivalenzbetrachtung für diese eine solche Angabe nicht erforderlich.

Verschlüsselung der Sitzungsschlüssel

Für die Verschlüsselung der Sitzungsschlüssel im Block wird bei ep2 das asymmetrische Verschlüsselungsverfahren RSAES-OAEP [RFC8017] eingesetzt. Dabei wird ein Schlüsselpaar verwendet, das ausschließlich zur Verschlüsselung verwendet wird. Die entsprechenden öffentlichen Schlüssel zur Verschlüsselung werden im Terminal gespeichert.

RSAES-OAEP ist mit der als Maskengenerierungsfunktion eingesetzten Hash-Funktion SHA-256 beweisbar sicher ([RFC8017], 8.2, p. 66]). Die verwendeten Schlüssellängen von 2048 werden für den Zeitraum bis 2030 als ausreichend sicher angesehen und ihre Unterstützung von der TR-34 [X9 TR-34, A.2.6] gefordert. Die ep2 Spezifikation sieht darüber hinaus auch Schlüssellängen von 3072 und 4096 Bit vor. Ein entsprechendes Migrationskonzept garantiert dann die Sicherheit über das Jahr 2030 hinaus.

Integritätsschutz der Sitzungsschlüssel

Der Block ([InterfaceSpec, Fig. 8, S. 10-31f]), in dem die Sitzungsschlüssel übertragen werden, hat keinen zusätzlichen Integritätsschutz wie eine Signatur nach TR-34 oder einen MAC-Wert nach TR-31. Der MAC, der im Schlüsselblock ([InterfaceSpec, Fig. 8, S. 10-31f]) übertragen wird, sichert die Übertragung der XML-Nachrichten in den jeweiligen Kommunikationskanälen Terminal ↔ Acquirer, POS management system/Terminal ↔ Acquirer, Service Center ↔ Acquirer, jeweils beim Senden und Empfangen.

Der Integritätsschutz der übertragenen Schlüsseldaten wird im ep2-System indirekt gesichert. Dabei muss jedoch gewährleistet sein, dass die Schlüsselbits gegen Veränderung geschützt bleiben.

Neben digitalen Signaturen und MAC-Sicherung sind auch andere Schutzmechanismen zulässig:

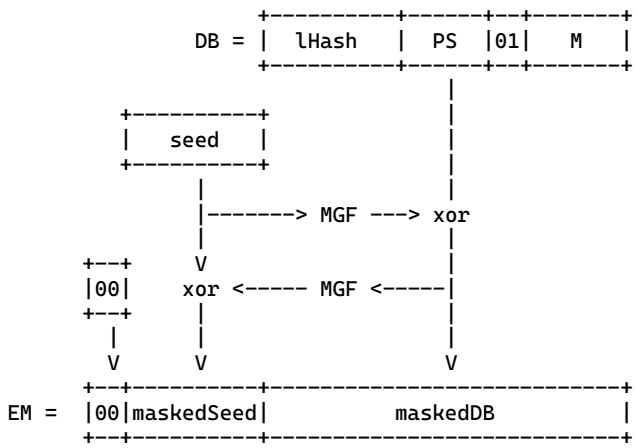
> Acceptable methods of implementing the integrity requirements include, but are not limited to: MAC…, digital signature … integrity check that is an implicit part of the key-encryption process … ([PCI PIN-Security, S. 58])

Da bei einem asymmetrischen Verfahren potentiell jeder in der Lage ist, Daten mit dem öffentlichen Schlüssel des Empfängers zu verschlüsseln, muss die Authentizität der Daten geprüft werden.

Die zu verschlüsselnde Nachricht *M* (97 Byte) ist dabei die Konkatenation von Header Byte (0x7F), der SenderID, dem Session Key, dem Component Secret, dem bei der Schlüsselableitung zu verwendendem Salt-Wert, und den für die Verschlüsselung mit AES im CTR-Modus zufälligen Nonce-Werten beim Senden und Empfangen ([SecSpec, Table 5, S. 9-35]):

Field Name	Length	Description	Fmt
Data Header	1	Value '7F' (hex)	b
Sender Identifier	8	Alphanumeric number identifying the sender of the message. Following rules shall be used for the truncation • <Acquirer Identifier>, <PMS Identifier> and <Service Center Identifier> n11: • left truncation of 3 digits and port to an8<Terminal Identification> uan8: copy to an8	an
<Session Key>	16	Current session key	b
<Component Secret>	16	<Component Secret> of the sender	b
<Salt>	32	<Salt> value used during the extraction phase of the key derivation function (KDF)	b
<Nonce Variant Encryption Send>	12	Nonce used in encipherment with <Session Key Variant Encryption>	b
<Nonce Variant Encryption Receive>	12	Nonce used in encipherment with <Session Key Variant Encryption>	b

RSAES-OAEP kodiert dann diese Nachricht *M* nach dem EME-OAEP-Schema ([RFC8017, Fig. 1, S. 24]).



Bei der RSAES-OAEP-eigenen Integritätsprüfung werden nach der RSA-Entschlüsselung das erste Byte (0x00) von EM, der `lhash`-Wert (32 Byte), der Padding-String aus Null-Bytes (93 Byte) sowie das Trennbyte (0x01) aus dem demaskierten Datenblock DB geprüft. Darüber hinaus wird eine Prüfung der entschlüsselten Nachricht vorgenommen. Dabei müssen SenderID (8 Byte) und Component Secret (16 Byte) den erwarteten Werten entsprechen.

Diese zusätzlichen Prüfungen bilden einen zuverlässigen Integritätsschutz für die Schlüsseldaten. Die Erstellung einer gültigen Nachricht mit einem modifizierten Sitzungsschlüssel wird durch das Component Secret verhindert, die EME-OAEP-Kodierung garantiert, dass jede Veränderung eines Bits im verschlüsselten Key Block zu einem Fehler bei der Prüfung der entschlüsselten Nachricht führt.

MAC-Prüfung der XML-Nachricht

Die MAC-Berechnung wird nach der ep2 Spezifikation nicht auf die Daten selbst, sondern auf den Hash-Wert der Daten angewendet ([SecSpec, S. 9-36f]). Bei der Hash-Funktion SHA-256 und der Länge der ausgetauschten Nachrichten bedeutet das keine Einschränkung der Sicherheit.

Die MAC-Berechnung basiert wieder auf der HMAC-Funktion nach RFC2104, dabei werden die speziell dafür abgeleiteten Schlüssel für Sender und Empfänger genutzt.

Die X9.143 verwendet für die *Key Blocks* einen auf 8 Byte reduzierten CMAC, lässt aber HMAC-Schlüssel und ihre Verwendung zu. Es wird für diese Schlüssel sogar ein gesonderter optionaler Block definiert, mit dem die verwendete Hash-Funktion spezifiziert wird und die SHA-256 einschließt. Da sich HMAC und CMAC in ihren Sicherheitseigenschaften nicht unterscheiden, ist die Verwendung des auf 8 Byte reduzierten HMAC-Wertes ein äquivalentes Verfahren.

Key Wrapping für 'PAN Receipt' und 'PAN Surrogate' Verwendung

Im Abschnitt 8.7 der ep2 Security Specification ([SecSpec]) werden für die Übertragung der symmetrischen Schlüssel (Key PAN Receipt und Key PAN Surrogate) Schlüsselblöcke nach dem Vorbild der X9.143 verwendet.

Die einzelnen Datenfelder für den Block entsprechen dabei genau den Vorgaben der X9.143, Kapitel 6.3.

Um diese Schlüssel zu unterscheiden, wird im Key Block Header der Schlüsselverwendungszweck 'X' (Key used to derive other keys) für den Schlüssel PAN Receipt und der Verwendungszweck 'G' (Generate only) für die Schlüssel PAN Surrogate (primary und secondary).

Für die Schlüsselableitung wird wieder das auf der Hash-Funktion SHA-256 basierende HKDF-Verfahren nach RFC5869 eingesetzt. Auf die Spezifikation der Hash-Funktion in einem optionalen Feld wird dabei verzichtet, da ohnehin nur SHA-256 eingesetzt wird.

Sicherheitsbewertung

Die bei ep2 eingesetzten Verfahren werden durch in der X9 TR-34 und X9.143 als geeignete eingestufte Algorithmen unterstützt. Dazu zählen SHA-256, RSAES-OAEP, HMAC. Sie machen jegliche Veränderung einzelner Bits erkennbar. Eine Entfernung oder ein Tausch einzelner Bits oder Datengruppen untereinander wird für das Session Key Cryptogram nach der Entschlüsselung durch EME-OAEP-Prüfung und der Prüfung von SenderID und Component Secret erkannt. Beim Key Wrapping werden die Anforderungen der X9 TR-31/X9.143 unmittelbar erfüllt.

Die verwendeten Parameter, wie SHA-256 und RSA-Schlüssellängen von 2048, 3072 und 4096 Bit, sind auf hohem Sicherheitsniveau.

Die Sicherheitsanforderungen an die Struktur der *Key Blocks* nach TR-34 und TR-31 werden durch äquivalente Verfahren erfüllt. Das ep2-Protokoll garantiert durch eine für den Verwendungszweck spezifische Schlüsselableitung die Bindung der abgeleiteten Schlüssel an ihren Verwendungszweck. Eine Verfälschung der Schlüsseldaten ist durch die Integritätsprüfung bei der RSAES-OAEP-Entschlüsselung und der EME-OAEP-Dekodierung ausgeschlossen.

Literaturverzeichnis

[InterfaceSpec] eft/pos 2000 Interface Specification, Version 8.1.0, 2022-12-31

[PCI PIN-Security] Payment Card Industry (PCI) PIN Security Requirements and Testing Procedures, Version 3.1, March 2021

[PIN-FAQ] PTS PIN Security Requirements, Technical FAQs for use with Version 3, Payment Card Industry (PCI), September 2021

[HSM-FAQ] PTS HSM Security Requirements, Technical FAQs for use with Version 3, Payment Card Industry (PCI), August 2022

[RFC2104] H. Krawczyk, M. Bellare, R. Canetti:, HMAC: Keyed-Hashing for Message Authentication,IETF, Informational RFC 2104, 1997-02

[RFC5869] H. Krawczyk, P. Eronen: HMAC-based Extract-and-Expand Key Derivation Function (HKDF), IETF, Informational RFC 5689, 2010-05

[RFC8017] K. Moriarty, B. Kaliski, J. Jonsson, A. Rusch: PKCS #1: RSA Cryptography Specifications Version 2.2, IETF, Informational RFC 8017, 2016-11

[SecSpec] eft/pos 2000 Security Specification, Version 8.1.0, 2022-12-31

[X9.73] Cryptographic Message Syntax — ASN.1 and XML, Accredited Standards Committee X9, American National Standards Institute, 2017

[X9.143] Retail Financial Services Interoperable Secure Key Block Specification (ANSI X9.143-2022), Accredited Standards Committee X9, American National Standards Institute, 2022-05

[X9 TR-34] Interoperable Method for Distribution of Symmetric Keys using Asymmetric Techniques: Part 1 – Using Factoring-Based Public Key Cryptography (Unilateral Key Transport), American National Standards Institute, 2012-08

[X9 TR-31] Interoperable Secure Key Exchange Key Block Specification (ASC X9 TR 31-2018), Accredited Standards Committee X9, American National Standards Institute, 2018

[Supp. 18-3] Information Supplement: PIN Security Requirement 18-3 − Key Blocks, PIN Assessment Working Group PCI Security Standards Council, July 2022