EBU

OPERATING EUROVISION AND EURORADIO

TECH 3292-s1

BISS – CA
BASIC INTEROPERABLE SCRAMBLING SYSTEM

Supplement 1:
CONDITIONAL ACCESS MODE

Version 1.0

Geneva
March 2018

There are blank pages throughout this document. This document is paginated for two sided printing

Abstract

This specification describes a conditional access mode for the Basic Interoperable Scrambling System (BISS), based on asymmetric cryptography for use on digital contribution circuits (satellite, DSNG, IP etc.).

It allows a dynamic, real-time and granular management of the stream entitlement whilst remaining Interoperable and secure.

Contents

List of Figures

List of Tables

BISS-CA
Basic Interoperable Scrambling System
Conditional Access Mode

EBU Committee	*First Issued*	*Revised*	*Re-issued*
TC	2018		

Keywords: Security, Scrambling, Satellite, AES-128, DVB-CISSA, Conditional Access, Rights Management, Contribution.

1. Introduction

This document describes the Conditional Access mode of the BISS protocol called Mode CA (BISS-CA). It specifies an open, interoperable conditional access system allowing the operator to revoke or allow, in real-time, the reception of programmes by a particular receiver.

The BISS (Basic Interoperable Scrambling System) protocol is a scrambling protocol based on the DVB-CSA specification in its deprecated version 1 (BISS1) and on DVB-CISSA for the version 2 (BISS2). The latest version, BISS2, was published in March 2018 as EBU Tech 3292v2. It uses fixed scrambling keys called Session Words (SWs) to secure the stream. With the BISS-CA publication, the BISS protocol is extended to 4 operational Modes (Mode 0, Mode 1, Mode E, and Mode CA).

The BISS Mode E, referred to as BISS-E, introduces a symmetric cipher to encrypt the Session Words (ESWs) with a Session Key (SK). While it does secure the Session Word itself, it still relies on unsafe transfer methods for the encrypted session words (ESW). Furthermore, it does not allow for a flexible management of the entitled receiver base.

The BISS-CA mode is built on top of the mode E whereby the session word is encrypted. In addition, it allows the operator to change the session key in-stream periodically in a seamless manner for the entitled receivers, while at the same time revoking receivers that are no longer entitled.

This standard addresses the need of sport federations and any content rights holder who is looking for a secure, transparent and traceable contribution and primary distribution system while being vendor agnostic. BISS-CA is backward compatible with existing multiplexers as far as they comply with the MPEG-2-TS [1] and DVB specifications. Furthermore, the protocol is designed to allow additional customisation by reserving space for private data carriage.

2. Glossary

Throughout this document, the following terms are used:

Scrambler	Overall mechanisms required to meet the DVB-CSA1 or DVB-CISSA specification.
Unit	Relates to a device for which this specification might apply.
Management Centre	Refers to an organization controlling or managing the conditional access system.
AES	Advanced Encryption Standard, fast symmetric encryption standard.
BISS	Basic Interoperable Scrambling System
BISS1	BISS version 1 with CSA1 for TS scrambling and DES for session word encryption.
BISS2	BISS version 2 with DVB-CISSA replacing CSA1 and AES128 replacing DES.
BISS-CA	BISS Conditional Access mode allowing secure key transmission in the MPEG transport stream.
bslbf	Bit string, left bit first
CA	Conditional Access
CAT	Conditional Access Table
CBC	Cipher Block Chaining
CISSA	DVB-CISSA Common IPTV Software-oriented Scrambling Algorithm
CSA	(DVB) Common Scrambling Algorithm
CW	Control Word
DES	Data Encryption Standard
DSNG	Digital Satellite News Gathering
DVB	Digital Video Broadcasting
ECM	Entitlement Control Message
EMM	Entitlement Management Message
ES	Elementary stream
ESID	Entitlement session ID
ESK	Encrypted Session Key
ESW	Encrypted Session Word
IRD	Integrated Receiver Decoder
lsb	Least Significant Bit
LSB	Least Significant Byte
MC	Management Centre
msb	Most Significant Bit
MSB	Most Significant Byte
PAT	Programme Association Table
PID	Programme Identification number
PMT	Programme Map Table.
RSA	Rivest–Shamir–Adleman asymmetric cryptosystem
SK	Session key, key transmitted through the EMM
SW	Session word, scrambling key transmitted through the ECM
Uimsbf	Unsigned integer, most significant bit first.

3. BISS Operational Modes

The Scrambler shall support the following four (4) modes of operation:

- **Mode 0:** No scrambling is applied.
- **Mode 1:** Components are scrambled by a Session Word (SW), and the SW is transmitted out of band in clear to the receivers.
- **Mode E:** Components are scrambled by a Session Word (SW), the SW is encrypted with a fixed Session Key (SK) and the resulting Encrypted Session Word (ESW) is transmitted out of band to the receivers.
- **Mode CA:** Components are scrambled with a Session Word (SW), the SW is encrypted with a Session Key (SK), and the resulting Encrypted Session Word (ESW), along with the key information is transmitted in-stream to receivers. Both SW and SK are dynamically changed during the live event transmission

The scrambling mechanism, as defined in the DVB-CSA for BISS1 and DVB CISSA [2] for BISS2, shall be applied at the Transport Stream level only. A Conditional Access Table (CAT) shall be present in the multiplex for BISS Mode 1 and Mode E, although the table shall be empty as no Entitlement Management Message (EMM) stream will be present in these modes. The CAT table is used in conjunction with EMM and ECM messages in Mode CA.

A scrambler that only supports a subset of the defined modes of operation, for BISS1 or BISS2, must do so according to an imposed hierarchy (see Figure 1). As an example, a Scrambler providing support for BISS2 Mode CA must also support BISS2 Modes 0, 1 and E.

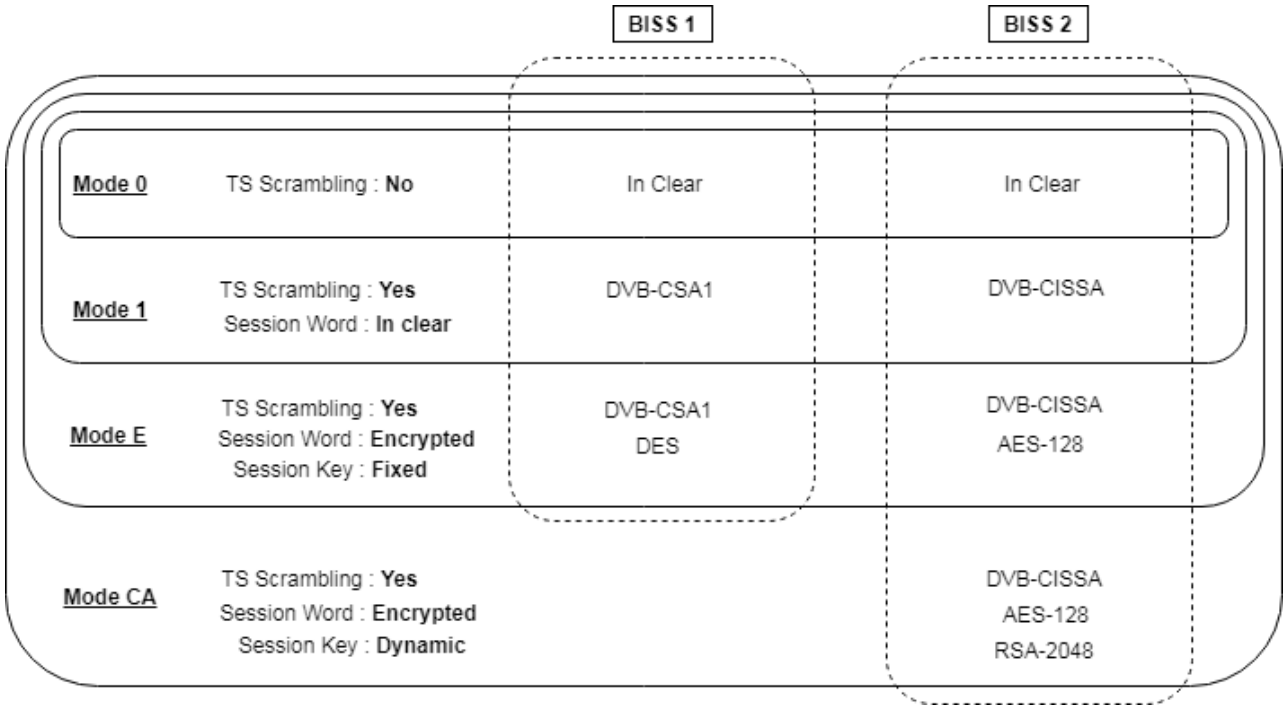

Figure 1: BISS1 and BISS2 Standards & Mode Overview.

4. BISS-CA

4.1 Overview

BISS-CA is a conditional access system based on open cryptographic standards. It uses a combination of symmetric and asymmetric ciphers (see figure 2) to protect the transmitted content and entitle or revoke, in real-time any targeted receivers in an interoperable manner. It is registered as a DVB service owned by the EBU, with the CA_SYSTEM_ID 0x2610.

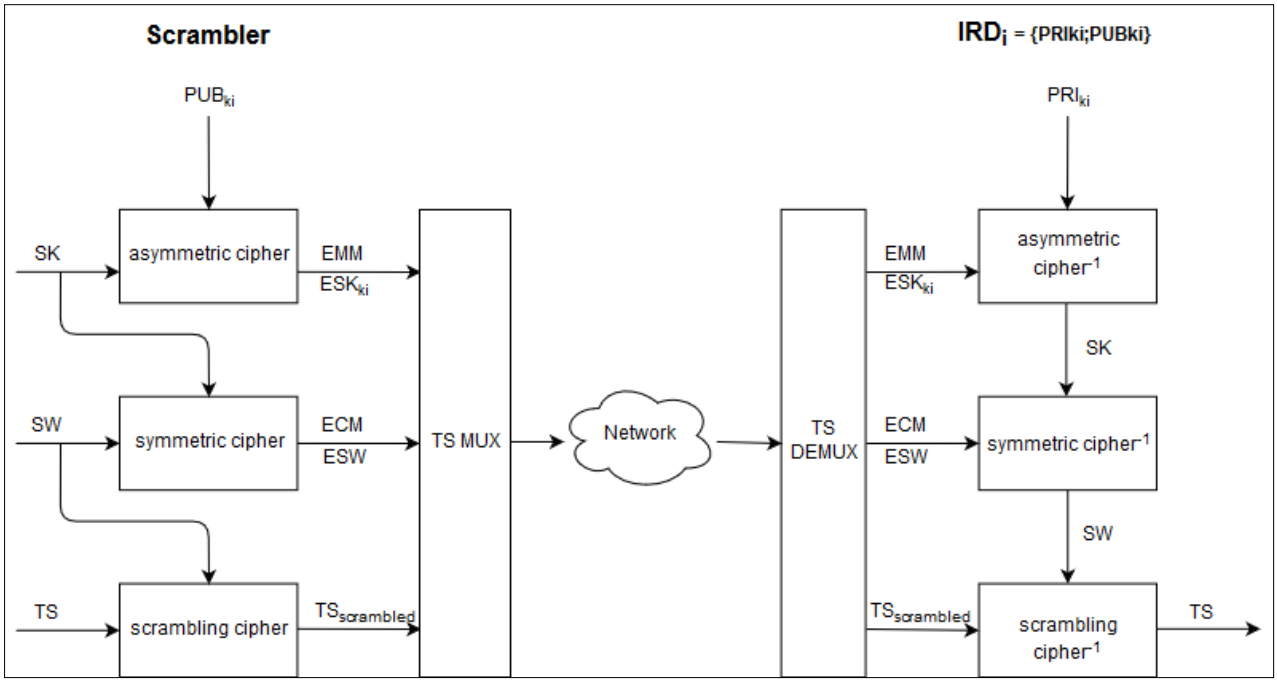

Figure 2: BISS-CA overview

In BISS-CA, each receiver (IRD_i) eligible to descramble the stream has an asymmetric key pair: a public key, and a private key { PRI_{ki} , PUB_{ki} }. The public keys of entitled receivers are transported to the scrambler out of band. The method of transporting the public keys are not defined in this document, but example transport methods are described in the Annex A according to the relevant use cases. An accurate entitlement list is maintained, consisting of a collection of entitled receivers with their corresponding public keys. The list is used by the scrambler to generate the Entitlement Management Messages (EMMs).

A Session Word (SW) is used as an input to the Transport Stream (TS) scrambling algorithm to scramble individual service components in the TS. The Session Word is then encrypted with a symmetric cipher (AES-128) using a Session Key (SK). The resulting Encrypted Session Word (ESW) is transmitted to the entitled receivers' in-band in the TS via Entitlement Control Messages (ECMs) (see § 4.2.2.5.5).

The Session Key (SK), which is required to decrypt the ESW, is encrypted individually with an asymmetric cipher (RSA-2048) using the public key of each entitled receiver (PUB_{ki} for IRD_i). Only the receiver having the corresponding Private Key (PRI_{ki}) will be able to decrypt that Encrypted Session Key (ESK_{ki}). The set of individual ESK_{ki} are transmitted to receivers' in-band in the TS via Entitlement Management Messages (EMMs) (see § 4.2.2.5.3).

The scrambled TS, and the ECM and EMM tables are multiplexed in the same TS. The EMM and ECM table structures are not scrambled, and shall be transmitted on separate PIDs.

To maintain security of the BISS-CA session, Session Words and Session Keys shall be automatically generated using a cryptographically secure random number generator by the sender/scrambler. Neither the SW nor SK shall be available in clear text via control APIs or other management interfaces of the scrambler. . In a redundant scrambler configuration, the scramblers are allowed to share the SK and SW through a secure protocol established between the units.

The exact method of creating random numbers is outside the scope of this document, but an example method using a Deterministic Random Bit Generator (DRBG) and a random or secret seed is described in NIST Special Publication 800-90A [3].

4.2 Protocol Component Description

4.2.1 Public and Private Key pair

A set of public and private key pair is generated for the asymmetric cipher (RSA). The public key is used by the scrambler to uniquely identify an individual receiver or a group of receivers belonging to the same Rights Holder. The private part of the key needs to be known by the receiver (or group of receivers), but shall not be retrievable from the receiver. The public part of the key need to be known by the scrambler and shall be retrievable by an operator or control API.

The format of the keys should be unencrypted PEM files defined in PKCS#8 (see Annex B for more information)

The method of transporting public/private key pairs, and in particular transporting the public key from receivers to the scrambler, is outside the scope of this document.

- The receivers shall have the ability to store several sets of key pairs generated externally (**injected key pairs** – see Annex A).
- The receivers may implement a mechanism to automatically generate a set of key pairs (**self-generated key pairs**). In this case, the manufacturer shall implement a mechanism to certify the origins of the key pair (e.g. embedded certificate).
- Each receiver shall have a **buried key pair**, i.e. a public/private key pair buried in the receiver by the manufacturer that uniquely identifies the receiver. Manufacturers shall maintain an accurate register of these key pairs for verification purpose.

To facilitate common management of a group of receivers, the group of receivers can share the same public/private key pair. Instead of sending an ESK for each individual receiver, the scrambler will transmit only one ESK for the group. Note that operation of the scrambler is the same when entitling an individual receiver or group of receivers. A compromised receiver that is part of a group, and that needs to be revoked, implies revocation of the transmission for all receivers in the group.

Annex A describes different use cases for key pair management schemes**.**

Annex B shows an example of a public key format.

Annex C shows examples of a public and private key pair.

4.2.1.1 Entitlement Key ID (EKID)

The Entitlement Key ID (entitlement_key_id) is a 64-bit identifier that shall be derived from the public key. It is the leftmost truncated 64 bits of the 256-bit Hash of the public key for specific set of key pairs. It is used to uniquely identify a set of key pairs and varies from a set of pairs to another. The public key is processed using a SHA-256 hash function.

To generate the Entitlement Key ID from the public key, the following procedure shall be used:

1. Start with the binary public key DER structure according to PKCS #8, as defined in Annex B
2. Calculate the SHA-256 digest of the binary DER structure
3. The Entitlement Key ID shall be defined as the leftmost 64 bits of the SHA-256 digest string

Annex C shows an example of entitlement key ID generation.

4.2.1.2 Session Word (SW)

Session words are used for scrambling of service components in the Transport Stream. The session word (SW) format depends on the BISS scrambling algorithm in force. When the TS scrambling algorithm used is DVB-CISSA, the session word contains the AES-128 [4] control word.

In BISS-CA, the Session Word is encrypted and the Encrypted Session Word (ESW) along with its metadata is transmitted in the ECM tables.

Annex C shows an example of encrypted session word.

4.2.1.3 Session Key (SK)

Session keys are used for encryption of data in ECM tables. For each individual receiver to be entitled, the Session Key (SK) is encrypted, and individually Encrypted Session Keys, ESK_{ki}, are distributed to the individual receivers in EMM tables.

The Session Key (SK) is required by receivers to decrypt and retrieve the Session Word (SW) from the ECM table.

The Session Key (SK) may change over time, and the two versions that can co-exist at any point in time in a session, are referred to as the odd and even Session Key. To revoke a currently entitled receiver, the Session Key must be changed, and the individually encrypted ESK_{ki} for the receiver to be revoked, must be removed from the EMM.

4.2.1.4 Entitlement Session Id (ESID)

The entitlement_session_id shall be an administratively configured ID that uniquely identifies a content protection session that is using a certain set of session keys (odd/even, and changing over time).

The entitlement_session_id shall be administratively set for every scrambler generating a BISS-CA stream, making the generated stream unique in that administrative scope.

The entitlement_session_id shall be referenced in EMM and ECM tables that belong to the entitlement session in question.

The entitlement_session_id shall also be referenced in any CA_descriptor referencing those EMM or ECM tables. The latter is achieved by adding a bissca_entitlement_session_id_descriptor to the private data part of the CA_descriptors. This mechanism makes it possible to multiplex BISS-CA streams while keeping the session data separated.

4.2.1.5 Original Network ID

Fields have been allocated in the EMM and ECM private data sections that make it possible to trace a stream back to the original_network_id values inserted at the stream origin – independent of any changes to the transport_stream_id or original_network_id in current SI tables, or multiplexing that might have taken place later in the transmission chain.

It is out of the scope of this standard to mandate specific use of these fields, except that their values shall be set to the same values as the corresponding values inserted in PSI/SI tables at the output of an encoder.

The use of the original_network_id also adds an administrative scope in addition to the entitlement_session_id, meaning that the entitlement_session_id can be independently managed within the scope of each original_network_id.

4.2.2 Table Definitions

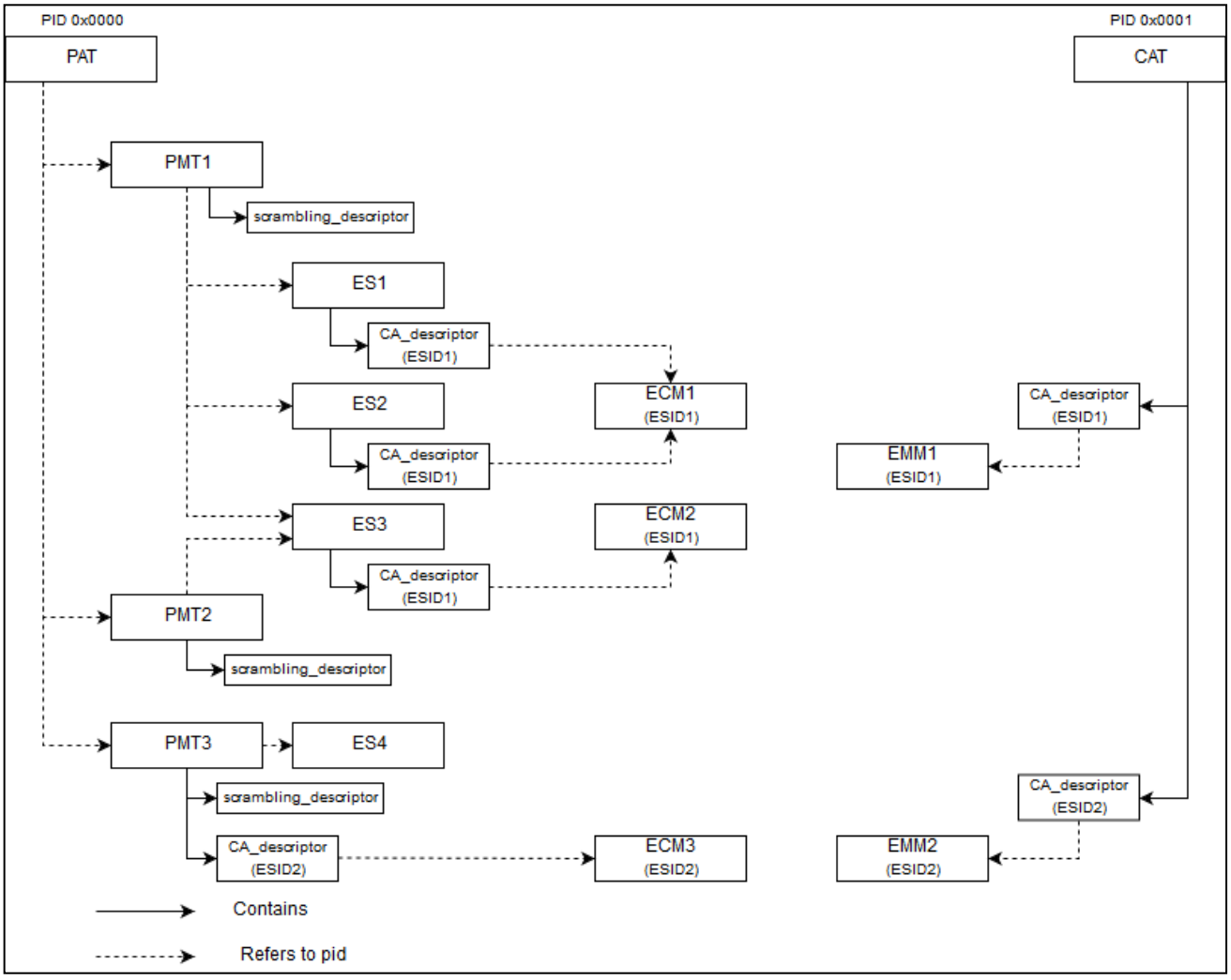

Figure 3: Tables relational diagram.

The CAT and the conditional access descriptor are as defined in ISO/IEC 13818-1 [1].

The scrambling descriptor is as defined in ETSI EN 300 468 [5].

4.2.2.1 CAT

The CAT is present when one or more ES are scrambled. The CA_descriptor in the CAT defines which CA system is used.

Table 1: Conditional Access Section

Syntax	No. of bits	Mnemonic
ca_section() {		
table_id	8	uimsbf
section syntax_indicator	1	bslbf
'0'	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
reserved	18	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
Last_section_number	8	uimsbf
for (i = 0; i < N; i++){		
descriptor()	8	uimsbf
}		
CRC_32	32	rpchof
}		

4.2.2.2 Conditional Access Descriptor

If any elementary stream is scrambled, a CA descriptor shall be present in the PMT for the programme containing that elementary stream. If any system-wide conditional access management information exists within a Transport Stream, a CA descriptor shall be present in the conditional access table (CAT).

Table 2: Conditional Access Descriptor.

Syntax	No. of bits	Mnemonic
ca_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
CA_system_ID	16	uimsbf
reserved	3	bslbf
CA_PID	13	uimsbf
for (i = 0; i < N; i++){		
private_data_byte	8	uimsbf
}		
}		

The meaning of CA_PID in the CA descriptor is context dependent. If the CA descriptor is in:

CAT: CA_PID refers to the EMM PID

PMT: CA_PID refers to the ECM PID
ES: CA_PID refers to the ECM PID

The CA descriptor defines the CA_system_ID, registered by DVB. For BISS-CA, the CA_system_ID shall be 0x2610.

In BISS-CA, the private_data_byte field shall contain one or more descriptors on the form bissca_entitlement_session_id_descriptor.

Table 3: BISS-CA entitlement session descriptor.

Syntax	No. of bits	Mnemonic
bissca_entitlement_session_id_descriptor {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
entitlement_session_id	16	uimsbf
original_network_id	16	uimsbf
}		
}		

The list with pairs of an entitlement_session_id and an original_network_id signals for which such ID pairs data can be found in the CA_PID referenced in this CA_descriptor.

The descriptor_tag for the bissca_entitlement_session_id_descriptor shall be 0x80.

4.2.2.3 Scrambling descriptor

The scrambling descriptor, located in the PMT, defines the scrambling algorithm used.

Table 4: Scrambling descriptor.

Syntax	No. of bits	Mnemonic
scrambling_descriptor {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
scrambling_mode	8	uimsbf
}		

For BISS-CA, the scrambling mode used shall be DVB-CISSA as specified in [4] section 6.

4.2.2.4 Generic private section syntax

The generic PSI private long section syntax will be used for the following table definitions. The generic syntax is defined in ISO/IEC 13818-1:2018 [1] section 2.4.4.10, illustrated in ISO/IEC 13818-1:2018, Table 2-30.

Table 5: Generic private section structure.

Syntax	No. of bits	Mnemonic
private_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
private_indicator	1	bslbf
reserved	2	bslbf
private_section_length	12	uimsbf
If(section_syntax_indicator =='0'){		
for(i=0;i<N;i++){		
private_data_byte	8	bslbf
}		
}		
else {		
table_id_extension	16	uimsbf
reserved	2	uimsbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for(i=0;i< private_section_length-9;i++){		
private_data_byte	8	bslbf
}		
}		
CRC_32	32	rpchof
}		

4.2.2.5 EMM and ECM tables

The EMM and ECM tables shall follow the generic private section syntax, with private data. The following chapters only detail non-generic syntax elements.

A table is comprised of multiple sections with the same table_id.

4.2.2.5.1 Table_id

Tables are identified by their table_id as defined in ETSI EN 300 468 [5]; we use the user defined range to add table_id for the EMM and ECM tables.

Table 6: Table Id Values.

table_id value	Description
0x80	ECM
0x81 to 0x8F	EMMs

4.2.2.5.3 EMM section

The EMM section provides information on the session keys, how they are encrypted and by which receiver, or group of receivers, they can be decrypted.

Table 7: EMM Table structure.

Syntax	No. of bits	Mnemonic
entitlement_management_message_section {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
private_indicator	1	bslbf
reserved	2	bslbf
private_section_length	12	uimsbf
entitlement_session_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
original_network_id	16	uimsbf
last_table_id	8	uimsbf
emm_cipher_type	3	uimsbf
entitlement_priv_data_loop	1	bslbf
reserved	8	uimsbf
descriptor_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
if(emm_cipher_type==RSA_2048_OAEP){		
for(i=0;i<N;i++){		
entitlement_key_id	64	bslbf
encrypted_session_data()	2048	bslbf
if(entitlement_priv_data_loop){		
reserved	4	
descriptor_length	12	
for(i=0;i<N;i++){		
descriptor()		
}		
}		
}		
}		
CRC_32	32	rpchof
}		

- **section_syntax_indicator:** a 1-bit field which shall be set to "1". It indicates that the section uses the generic long table section syntax.
- **private_indicator:** a 1-bit field which shall be set to "1". It indicates the private_section syntax.
- **entitlement_session_id:** an administratively designated ID field that has a 1-to-1 relation to a set of session keys (odd, even, changing over time). Shall be referenced in EMM, ECM and CA_descriptors relating to that entitlement_session_id.
- **original_network_id:** this field shall be set to the same value as the original_network_id field in PSI/SI tables in the TS being scrambled.
- **last_table_id:** this 8-bit field identifies the last table_id used for EMM data.
- **emm_cipher_type:** indicates which algorithm that is used for EMM payload encryption. In the BISS–CA protocol the EMM messages shall use the RSA 2048 bits OAEP. RSA OAEP refers to the implementation of the RSA protocol as defined in NIST special publication 800-56B [6] and PKCS#1 [7] .

Table 8: EMM Cipher Type Table.

emm_cipher_type	Algorithm used for EMM payload encryption
0b000	RSA 2048bits OAEP
0b010 to 0b111	Reserved for future use

- **entitlement_priv_data_loop:** indicates an entitlement_priv_data descriptor loop is present in all the entitlement_key_id loop elements.
- **descriptor_length:** the length in bytes of the following descriptor loop.
- **entitlement_key_id:** This 64-bit field is a unique identifier of a public key pair used by a receiver. It is based on EUI-64. It allows receivers to locate the right ESK in the EMM table – encrypted with the public key given by the **entitlement_key_id**, for which they also have the corresponding private key.

A key pair given by an **entitlement_key_id** may either be used by a single receiver, or it can be shared by several receivers belonging to the same media rights holder.

- **encrypted_session_data:** The session data structure as shown in Table 9, encrypted with RSA, using the public key given by the **entitlement_key_id**, and using the OAEP padding algorithm as defined in PKCS #1 [7].

- For BISS-CA the hash function shall be SHA-256 and mask generation function shall be MGF1-SHA-256.

4.2.2.5.4 Session data

The session data contains the Session Key (SK) and the output stream control commands. The data is formatted as a descriptor loop.

The stream control commands describe which actions a BISS-CA certified receiver must comply with. For example, a receiver may not be allowed to forward a descrambled transport stream or it may have to insert a digital watermark in the output video.

For the session data to be protected against a third-party modification, the session data is encrypted with an asymmetric cipher.

Descriptor tags in the range 0x80 to 0xBF are reserved for EBU usage. Tags between 0xC0 and 0xFE may be used to carry vendor- or operator-specific (i.e. proprietary) information to receivers.

Table 9: Session data descriptor.

Syntax	No. of bits	Mnemonic
session_data {		
reserved	4	bslbf
descriptor_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
}		

Annex C shows an example of session data in EMM.

4.2.2.5.4.1 Session key descriptor

The session_key_descriptor shall contain information about the session key, as well as the session key itself.

Table 10: Encrypted session key descriptor.

Syntax	No. of bits	Mnemonic
session_key_descriptor {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
session_key_type	7	uimsbf
session_key_parity	1	bslbf
for(i=0;i<N;i++) {		
session_key_data()		
}		
}		

- **descriptor_tag:** the tag shall be 0x81
- **session_key_type:** the type of the session key, key length is given by type.
- **session_key_parity:** indicates if the transmitted session key is even (0) or odd (1).

Table 11: Session key type values.

session_key_type	Key type
0b000	AES-128
0b001 to 0b111	Reserved for future use

- **session_key_data:** The session key used to encrypt session words.

The maximum number of session_key_descriptors in the session_data structure shall be two, in which case the session_key_parity shall be different.

4.2.2.5.4.2 Entitlement flags descriptor

The entitlement_flags_descriptor shall always be present and shall contain flags indicating parameters for how the entitled receiver shall behave.

Table 12: Entitlement flags descriptor.

Syntax	No. of bits	Mnemonic
entitlement_flags_descriptor {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
prevent_descrambled_forward	1	bslbf
prevent_decoded_forward	1	bslbf
insert_watermark	1	bslbf
Reserved	5	bslbf
}		

- **descriptor_tag:** the tag shall be 0x82
- **prevent_descrambled_forward:** this is a 1-bit field which when set to "1" indicates that the descrambled stream shall not be forwarded unscrambled. In the situation of a transcoder, the transcoder shall not forward its output stream unscrambled.
- **prevent_decoded_forward:** this is a 1-bit field which when set to "1" indicates that the descrambled and decoded stream shall not be forwarded unscrambled.
- **insert_watermark:** this is a 1-bit field which when set to "1" indicates that the descrambled and decoded service shall be watermarked. The stream shall not be descrambled, decoded, transcoded, if no watermark technology is available.
- **reserved:** must be set to '0'

4.2.2.5.5 ECM section

The ECM section provides information about the session words, how they are encrypted, and how to retrieve the session key in the EMM table to decrypt them.

Table 13: ECM Table

Syntax	No. of bits	Mnemonic
entitlement_control_message_section {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
private_indicator	1	bslbf
reserved	2	bslbf
private_section_length	12	uimsbf
entitlement_session_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
original_network_id	16	uimsbf
ecm_cipher_type	3	uimsbf

reserved	1	bslbf
descriptor_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
if(ecm_cipher_type==AES_128_CBC){		
session_key_parity	1	bslbf
reserved	7	bslbf
AES_128_CBC_enc_session_word_iv	128	bslbf
AES_128_CBC_enc_session_word_0	128	bslbf
AES_128_CBC_enc_session_word_1	128	bslbf
}		
CRC_32	32	rpchof
}		

- **section_syntax_indicator:** the section_syntax_indicator is a 1-bit field which shall be set to "1". It indicates that the section uses the generic long table section syntax.
- **private_indicator:** the private_indicator is a 1-bit field which shall be set to "1". It indicates the private_section syntax.
- **entitlement_session_id:** an administratively designated ID field that has a 1-to-1 relation to a set of session keys (odd, even, changing over time). Shall be referenced in EMM, ECM and CA_descriptors relating to that group.
- **original_network_id:** this field should be set to the same value as the original_network_id field in PSI/SI tables inserted in the TS being scrambled.
- **ecm_cipher_type:** indicates which algorithm is used to generate the encrypted_session_word. In the BISS-CA Protocol the ECM message shall use the AES-128 cipher in CBC mode. The corresponding ecm_cipher_type value is described in the ECM cipher table (Table 14).

Table 14: ECM cipher types values.

ecm_cipher_type	Algorithm used for encrypting SW
0b000	AES 128 CBC
0b001 to 0b111	Reserved for future use

- **descriptor_length:** the length in bytes of the following descriptor loop.
- **session_key_parity:** indicates if the session key used to encrypt the encrypted_session_word is even (0) or odd (1).
- **AES_128_CBC_enc_scrambling_word_iv:** the Initialization Vector used when encrypting the following session word. This value shall be randomly generated by the sender every time the ECM payload is updated, and the value shall not be equal for two entries with the same session_key_parity (unique IVs when the same session key is used for encryption of two session words).
- **AES_128_CBC_enc_scrambling_word_0:** a 128-bit field. The ESW containing the SW of parity even (0).
- **AES_128_CBC_enc_scrambling_word_1:** a 128-bit field. The ESW containing the SW of parity odd (1).

Table 15: list of TS scrambling modes in BISS-CA.

TS scrambling mode	AES_128_CBC_enc_scrambling_word interpretation	No. of bits	Mnemonic
DVB-CISSA	DVB-CISSA CW	128	bslbf

5. Normative Implementation Considerations

At the start of an entitlement session, or when the list of entitled receivers change, the scrambler generates a new set of random session keys and session words, and start transmitting the updated EMMs and then the ECMs. When the EMM with the new individual ESKs have been transmitted to all entitled receivers, specifically for the duration of time given below, SWs in the ECM are encrypted with the new SK.

This ensures that receivers that were entitled for the previous entitlement session cannot descramble the new feed. It also ensures that revoked receivers are no longer able to descramble the TS.

- Session Words (SW) should be changed regularly. In BISS-CA mode, the Session Word shall be automatically generated by the sender and conveyed to the entitled receivers according to the protocol even in the DSNG use case.

ECM and EMM tables are repeated regularly with a period T_{EMM} and T_{ECM} respectively with $T_{EMM} > T_{ECM}$.

- The minimum ECM repetition interval T_{ECM_min} = 100 ms
- The minimum period for ECM change $T_{ECM\_change\_min} = 10 * T_{ECM\_min}$ = 1 second

- The minimum EMM repetition interval $T_{EMM\_min} = 2 * T_{ECM\_min}$ = 200 ms
- The minimum period for EMM change $T_{EMM\_change\_min} = 10 * T_{EMM\_min}$ = 2 seconds

ESK shall to be transmitted at least twice before SWs are encrypted with the new SK. This is done to increase robustness against packet and table corruption. In addition, a minimum period of a second (i.e. half the minimum EMM change period) shall be observed to give the receiver enough time to manage the new keys. Then the maximum time before a receiver can acquire and use an EMM is $T_{EMM\_acq\_max} = 2 * T_{EMM} + T_{EMM\_change\_min}/2$

Similarly, the ESW shall be transmitted at least twice in addition to a minimum period of half the minimum ECM change period, before TS is scrambled with the new SW. Then the maximum time before a receiver can acquire and use an ECM is $T_{ECM\_acq\_max} = 2 * T_{ECM} + T_{ECM\_change\_min}/2$.

The periods of SK change (T_{EMM_change}) and SW change (T_{ECM_change}) may vary during a transmission but must be longer than $T_{EMM\_acq\_max}$ and $T_{ECM\_acq\_max}$ respectively.

$T_{EMM\_change} > T_{EMM\_acq\_max} > T_{ECM}$

- The EMM bitrate is limited to a maximum (B_{EMM_max}) of 1 Mbit/s.
- The ECM bitrate is limited to a maximum that can be inferred from the minimum ECM repetition interval (T_{ECM_min})

A BISS-CA compliant transmitter shall comply with the following rules:

- The maximum number of SW encrypted and transmitted in an ECM table is two (in which case the SW parities must be different).

- The number of SK encrypted and transmitted in an EMM table is two.
- Only one SK shall be used at a time to encrypt a SW.
- The SK must be transmitted $T_{EMM\_acq\_max}$ before being used for encrypting the SW.
- The SK used for encrypting the SW must be currently transmitted in the EMM.
- The SW must be transmitted $T_{ECM\_acq\_max}$ before being used for scrambling the TS.
- The SW used for scrambling the TS must be currently transmitted in the ECM.
- The TSC field in the TS header of the scrambled TS alongside EMMs and ECMs tables shall indicate which key is currently used.
- When a SK change is executed, both the currently used SK (by the ECM) and the next one shall be transmitted in the EMM for a minimum duration of $T_{EMM\_acq\_max.}$ Until this duration of time has passed, the next SK cannot be used by the ECM. After this duration, the new SK can be used by the ECM, and when not used in the ECM, the unused SK should not be transmitted in the EMM anymore.

EMM and ECM message timing is illustrated in Figure 4, overleaf.

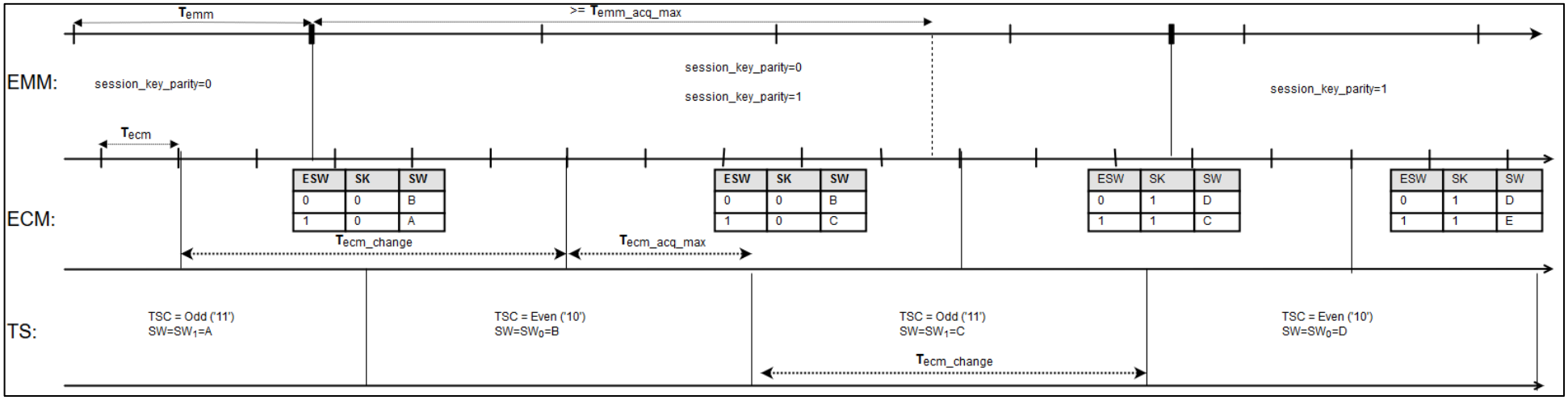

Figure 4: EMM & ECM Messages timing.

6. References

[1] ISO/IEC 13818-1:2018 Generic coding of moving pictures and associated audio information – Part1:Systems; https://www.iso.org/standard/74427.html

[2] ETSI TS 103 127 V1.1.1 (2013-05) - Digital Video Broadcasting (DVB); Content Scrambling Algorithms for DVB-IPTV Services using MPEG-2 Transport Streams : http://www.etsi.org/deliver/etsi_ts/103100_103199/103127/01.01.01_60/ts_103127v010101p.pdf

[3] Recommendation for random Number Generation Using Deterministic Random Bit Generators; https://csrc.nist.gov/publications/detail/sp/800-90a/rev-1/final

[4] Federal Information Processing Standards, Publication 197 - ADVANCED ENCRYPTION STANDARD (AES) https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf

[5] ETSI EN 300 468 v1.13.1 ; DVB document A038 ; Specification for Service information (SI in DVB Systems); https://www.dvb.org/resources/public/standards/a38_dvb-si_specification.pdf

[6] NIST 800-56b; Recommendation for pairwise Key-Establishment schemes Using Integer factorization cryptography; https://csrc.nist.gov/publications/detail/sp/800-56b/rev-1/final

[7] RFC2437 : RSA cryptography specifications Version 2.0; https://tools.ietf.org/html/rfc2437

[8] RFC 5280 : PKIX Certificate and CRL Profile; X.509; https://tools.ietf.org/html/rfc5280#appendix-A.1

[9] PKCS#8 https://tools.ietf.org/html/rfc5208#section-2

[10] RSA OID algorithm, RFC 5698 https://tools.ietf.org/html/rfc5698#section-8

[11] OID algorithm registration at IANA; https://www.iana.org/assignments/dssc/dssc.xhtml

[12] PKCS#1 RFC8017, https://tools.ietf.org/html/rfc8017#appendix-A.1

Annex A: Use Cases & Credential Management Description.

The BISS-CA Mode is a conditional access mode of the BISS protocol that enables real-time addition or revocation of a receiver. Depending on the use case, the receivers may be submitted to different registration and credential exchange processes. This section describes, for information only, how the BISS-CA mode can be implemented in two generic use case scenarios: a managed and unmanaged network of IRDs.

- A managed network of IRDs refers to a network where all receivers are administratively controlled by a central authority i.e. a management centre. In this particular case, the management centre shall have a method to control the receivers. This can be a physical access after delivery from the manufacturer before dispatch on the network and/or it can be secured remote access to the IRDs for upgrades.
- An unmanaged network of IRDs is a network where receivers are managed independently by different authorities. A third party SNG falls into this category.

This section will provide information on the keys and session credentials management process.

A1 Entitlement Credentials Management

There are 3 unique identifiers that are used during a transmission.

1. The **Injected ID or Buried ID** (mainly used in BISS-E modes) which is a **128 bit** unique identifier generated by the manufacturer (buried ID) or inserted by the management centre (Injected ID) on premise before dispatching the receivers. This ID remains unchanged and inaccessible to the operators once inserted.

2. The **Entitlement Key ID** (entitlement_key_id) is a **64 bit** identifier derived from the public key. It is the leftmost truncated 64 bits of the 256 bit Hash of the public key for specific set of Key pairs. It is used to uniquely identify a set of key pairs and varies from a set of pairs to another. The public key in processed using a SHA-256 hash function.

3. A set of Public/Private key pairs that will be used during the transmission to encrypt and decrypt the messages containing the transmission session key. The public key is retrievable by any operator while the private key is not. A receiver can host 3 types of key pairs:

    - **Injected Key pairs** are keys generated by the management centre and injected in the receivers before dispatch. The keys database is managed by a management centre. A receiver can host several injected key pairs.
    - **A buried Key pair** is a unique set of public/private key injected by the manufacturer. It can be used together with a serial number, the buried ID or any other unique identifier (example a license number in the case of a software implementation to uniquely identify a receiver.
    - **Self-generated key pairs** are key pairs generated by the receivers for its own use. The private key remains buried and inaccessible to the operator. In this particular case, it is necessary for security purpose, to implement a mechanism that certifies the origins of the self-generated key (e.g. certificates managed by a trusted central authority).

These identifiers are managed differently depending on the use case. While the injected ID and Injected Key pairs are managed and maintained by the management centre, the buried ID and Key pairs are managed by the manufacturers. The manufacturers shall maintain an accurate database

of the buried key pairs. A session is characterised by:

- The **entitlement_session_id**
- A list of entitled IRD characterised by the **entitlement key ID** and the corresponding **public key** of each entitled receiver.

In the case of a managed network of receivers, the entitlement list is provided and updated by the management centre. The list will be composed of injected key pairs corresponding to receivers or group of receivers managed by the centre. The list can be provided to the transmission operator by any means (email, USB, etc...) deemed secure by the management centre.

In the case of an unmanaged set of receivers, the receiver for which a stream entitlement is requested, should be identified and added to the entitlement list. The receiver operator has the following options:

- **Option 1:** The operator of the receiver communicates (via email or other communication means) the receivers' buried public key and serial number or any unique identifier such as a licence number (in case of a software receiver), to the operator of the scrambler/transmitter. The operator of the scrambler should verify the origin of the public key, i.e. that it were genuinely created and is still endorsed (not revoked) by the receiver manufacturer.
- **Option 2:** If available as a receiver functionality, the receiver self-generates a key pair and generates a self-signed certificate from the public key. The resulting certificate, which includes the public key, is communicated to the operator of the scrambler. The operator of the scrambler should verify that the key originates from a genuine manufacturer and not from a malicious 3rd party receiver. The operator should verify the certificate issued before entitling the receiver.

Annex B: Public Key Format Description (informative)

The RSA Public Private Key pair used in this specification is stored in the receiver and the Public Key is communicated to the BISS-CA scrambler. While the method for communicating the public key is not defined herein, to foster interoperability between various implementations of BISS-CA, it is useful to provide some information about the format of the keys and how they could be distributed.

The file format used for storage and distribution of Public Keys are typically unencrypted PEM files as defined in PKCS#8 [9]. PEM files are text files containing a header, some binary data, and a footer. When representing a Key, the binary data is the ASN.1 DER-encoded Public key.

The contents of a PKCS #8 PEM file are formatted as follows:

```
-----BEGIN PUBLIC KEY-----
BASE64 ENCODED DER STRUCTURE
-----END PUBLIC KEY-----
```

This format is well suited for representing Public Keys, and allows simple transport of Public Keys between various software systems, either as file transfer, e-mail or other messaging system, or a more sophisticated protocol.

A single PEM file can hold more than one Public Key. PEM files according to PKCS#8 are not limited to RSA keys, but can contain keys for any kind of cypher.

B1 Binary DER structure

The Base64 encoded top level DER structure is defined as the data structure `SubjectPublicKeyInfo` from X.509 [8]:

```
SubjectPublicKeyInfo ::= SEQUENCE {
algorithm       AlgorithmIdentifier,
PublicKey       BIT STRING
}
```

`AlgorithmIdentifier` is referenced by PKCS#8 [9] and is defined in X.509 [8]:

```
AlgorithmIdentifier ::= SEQUENCE {
algorithm       OBJECT IDENTIFIER,
parameters      ANY DEFINED BY algorithm OPTIONAL
}
```

The algorithm OID for RSA is defined in RFC 5480 [https://tools.ietf.org/html/rfc5480#appendix-A] and RFC 5698 [10], and registered with IANA at [11]. The RSA public key algorithm OID is [1.2.840.113549.1.1.1]. There are no optional algorithm parameters for RSA.

For a public RSA key, the `PublicKey` field in the `SubjectPublicKeyInfo` structure will consist of the `RSAPublicKey` data structure, defined in PKCS#1 [12]:

```
RSAPublicKey ::= SEQUENCE {
modulus           INTEGER,  -- n
publicExponent    INTEGER   -- e
}
```

Annex C: Component Examples

C1 Example of Public & Private Key pair

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAvkIDCEWUjaiPeNuROvTiGOAhtodNQAyBeIW66DiscEOddR1C
ikj4NuW7KjW5Oic9/GFt0nCr2rYGVsHpWOdPvkSLybmtX5ndxni9CxomdcUQNNL9
948cXBuNMThc6RPQlH1O7kL6B/ETBXrNJLTVQUABVnIVjr04PEMfi4cztGT964ul
OQ5e/dQJUpxbG3aL5BxFSnX89JwSHgdn1dwtpOC9FGrEX3I41wxdR0OwKzI5klKv
wgdsipofi/3JUI9rOuC+uhgET40uQxr27KL5Y+2NjEKoooqUZTD0Q+5CSWYXm6hq
tHeBJB92/xD3Do0ZEDArQ5iViysH71q0ObjMowIDAQABAoIBAQCXUPMOciB7JfOt
wJtaE4d7F09Y13VWGlwimeGUbfafdvdVPMc+KljXeJEKOh4uJSXEg0yQETJtSVXz
TFglcBrZDbVL5BQCs+JRxpc7rDmocum3yZNZgAWjL/p0igpDCZJbdun+z2ACTva8
5fUgW348XgZyVVvV4aJHM290TjyOHF5hOod4l0KjOlec5bAdz28CaEqftqytypQm
IFXQ0+06k9Dthm8HCwl6NLWwDPe7s+vNMJlq+tR4Nfu7beEDT0ydgI30FB7rDMkZ
I/Y2TL6HsiUBh83JIkj6g2SqkdzHjTI52fWDBZ1lCEzIKh+br5anaSun3YUFkEFU
tXdWWClhAoGBAPtA3fk5rQuSq05hl4rMrl/FFRiEcuuMHF/0wojBHyac7plaQW6S
eni9z5NMnfvNQVdjsUHKSYVRypMfodWmim4jtk3/FpwEjXniJFNPs2Yv+FFNBSPI
ptIVIq6ext7NA7syeCy4INRF/Spj3hBIH9IsOx6IsQbY69/jwZcVzkTzAoGBAMHa
JyxLTETxLuAQ5wuXbfWk4iq5so6zwDzbe97GlCTqL+YP6KsNKkbkc/1JDjYu8KKu
bL8LE7bKMU+4dE97olZhvZSdi9O69P80iLlM/gK4a+6adjcc5DcacgIX1Xa41uk3
oEGJaiqzdvgDjthYntJx1d3V1o3zBwniW4nkCQWRAoGAcJUSYbh8V7Ey3X5RXzpz
8CnpWAERVYaEuGJ+QLT4dl7fcwvEQf2Ur0GuH3y3VbsVSkk7hhVUeE68DMyhwZBM
eym5aJ2izeokUrcIO+R8qI9aH2P5p50jUwNxdPlkdzU6NMlam/8thrCNzk7NlFId
IBn9q6LoX/8XQk1V05NLyA0CgYBizwyakkLs/TaUdWkfSm98/y9c8cxm2o6JNqLb
+cI3UrtZfBBvZ8V93yKUHzHEQobblSbO9hl1WXhrFy0J+o3Tk/xrDSbhpHEOyDtM
oEb1IgW52DebffmBcNRd5sIiwrKgq37fCOj5nQJuBnpAImPKBsYpTb8QGakjy6I3
FenXUQKBgAvaPmlMVdpU5eAl23p7bn77FvvCgDklgMg11fl3hWZN6QGX+DMWS03+
EkllfhoCuiSoA8lGCipLiAZZ5rfFBLUkQkrKTmKhTlgkXsB7akT6k1O1QVkn/2jb
lsk0R8JgfaKjOvt3Ba57qvSAtLa8Misn5lmv9/kDrGILCYGRNw8a
-----END RSA PRIVATE KEY-----


-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvkIDCEWUjaiPeNuROvTi
GOAhtodNQAyBeIW66DiscEOddR1Cikj4NuW7KjW5Oic9/GFt0nCr2rYGVsHpWOdP
vkSLybmtX5ndxni9CxomdcUQNNL9948cXBuNMThc6RPQlH1O7kL6B/ETBXrNJLTV
QUABVnIVjr04PEMfi4cztGT964ulOQ5e/dQJUpxbG3aL5BxFSnX89JwSHgdn1dwt
pOC9FGrEX3I41wxdR0OwKzI5klKvwgdsipofi/3JUI9rOuC+uhgET40uQxr27KL5
Y+2NjEKoooqUZTD0Q+5CSWYXm6hqtHeBJB92/xD3Do0ZEDArQ5iViysH71q0ObjM
owIDAQAB
-----END PUBLIC KEY-----
```

C2 Example of Entitlement Key ID generation

Base64 decoded / DER of the Public key in the section above

```
30 82 01 22 30 0d 06 09 2a 86 48 86 f7 0d 01 01 01 05 00 03 82 01 0f 00 30
82 01 0a 02 82 01 01 00 be 42 03 08 45 94 8d a8 8f 78 db 91 3a f4 e2 18 e0
21 b6 87 4d 40 0c 81 78 85 ba e8 38 ac 70 43 9d 75 1d 42 8a 48 f8 36 e5 bb
2a 35 b9 3a 27 3d fc 61 6d d2 70 ab da b6 06 56 c1 e9 58 e7 4f be 44 8b c9
b9 ad 5f 99 dd c6 78 bd 0b 1a 26 75 c5 10 34 d2 fd f7 8f 1c 5c 1b 8d 31 38
5c e9 13 d0 94 7d 4e ee 42 fa 07 f1 13 05 7a cd 24 b4 d5 41 40 01 56 72 15
8e bd 38 3c 43 1f 8b 87 33 b4 64 fd eb 8b a5 39 0e 5e fd d4 09 52 9c 5b 1b
76 8b e4 1c 45 4a 75 fc f4 9c 12 1e 07 67 d5 dc 2d a4 e0 bd 14 6a c4 5f 72
38 d7 0c 5d 47 43 b0 2b 32 39 92 52 af c2 07 6c 8a 9a 1f 8b fd c9 50 8f 6b
3a e0 be ba 18 04 4f 8d 2e 43 1a f6 ec a2 f9 63 ed 8d 8c 42 a8 a2 8a 94 65
30 f4 43 ee 42 49 66 17 9b a8 6a b4 77 81 24 1f 76 ff 10 f7 0e 8d 19 10 30
2b 43 98 95 8b 2b 07 ef 5a b4 39 b8 cc a3 02 03 01 00 01
```

SHA-256 of DER:

```
1d 68 e8 a4 52 15 55 23 05 60 c4 6f 2b 69 0e 18 fe 6b 62 46 1a 96 e7 7d 51
50 7a 86 94 82 72 71
```

Resulting entitlement key id:

```
0x1d68e8a452155523
```

C3 Example of session data in EMM

SK parity=0

SK: `29 82 38 be 84 ae 1d 6c d6 2a e9 52 90 64 9d f1`

Entitlement flags: 0

Session data:

```
0000  00 16 81 11 00 29 82 38 be 84 ae 1d 6c d6 2a e9
0010  52 90 64 9d f1 82 01 00
```

Encrypted session data:

```
4f df a9 8a 0a b5 5f 68 1f 6f ec 2b 38 f4 69 7f 46 0f c9 20 8e a2 bb c5 1b
16 84 f5 01 18 7f 6e 09 d5 24 30 07 54 9f 22 43 71 87 ff a8 2d 2b b5 3e d5
ba ed 02 d6 bd 1d 0b 58 07 41 1a e0 60 23
cb 92 3b 4f 89 5f db 5f 61 1f 39 70 99 40 8a d1 75 66 21 a0 56 90 9b a8 0a
6f c5 9e 68 a7 1d e5 6a ac 60 eb dc 24 e9 3b 1a 0b c8 7e 16 00 fe 75 ea cf
e0 6a 2d 66 73 9f 16 cb c9 e7 4b 7b cb 08
88 df 17 77 ca c5 8e 0d 14 44 e5 5f 4c 80 2b 39 d8 f4 16 37 20 e6 dd 50 5c
6d ca 7c a2 d3 95 6d 45 7e 82 e6 8e c3 98 0a 6e ad 3c fe a8 88 d3 c2 5a 4e
c8 8f 60 73 7d e8 f3 a7 b3 d4 07 e2 9a 5c
39 14 06 3f cb 14 04 bf 33 37 16 2e dd 04 b7 0c d7 30 2a 07 cd d1 0e e1 84
5d af ea a8 4c c6 92 43 66 ad b2 59 3c 58 43 cd e5 1c 37 58 52 10 1f 02 e4
d1 65 a6 27 28 e6 1b 99 21 8f 12 3a 8e 24
```

C4 Example of ESW

IV = 6d fd bf 58 b0 39 4b 4a aa a4 ef 86 5f 63 bf 86

SW0 = 9d 42 c2 ec d2 de 5b 15 f2 27 ae a7 db a5 f8 f0

SW1 = 1c 27 3f c4 bd 66 4b 40 fd 8c d3 b0 5b 26 d3 42

Using SK from last section:

ESW0: 21 8b f6 fa c3 9f ac e8 25 cd 1e de b7 bf 6a 17

ESW1: 10 f9 3b 6e 5d 94 f5 cc 38 52 05 74 b1 4b 19 40

Parity+IV+ESW0+ESW1:

00 6d fd bf 58 b0 39 4b 4a aa a4 ef 86 5f 63 bf 86 21 8b f6 fa c3 9f ac e8
25 cd 1e de b7 bf 6a 17 10 f9 3b 6e 5d 94 f5 cc 38 52 05 74 b1 4b 19 40