BISS-E

— Basic Interoperable Scrambling System with Encrypted keys

August 2002

European Broadcasting Union
Case postale 45
Ancienne Route 17A
CH-1218 Grand-Saconnex
Geneva, Switzerland
techreview@ebu.ch

EBU·UER

Summary

This specification describes a Basic Interoperable Scrambling System (BISS-E) for use on digital contribution circuits (satellite, DSNG, etc.) which use MPEG-2 compression, the DVB-S modulation scheme and the DVB Common Scrambling Algorithm with fixed keys.

BISS-E uses encrypted Session Keys and allows centrally-managed Conditional Access.

Contents

Chapter 1: Introduction 1
1.1. Overview 1
1.2. Nomenclature 1
1.3. Notations 1
1.4. Security requirements 2
1.5. Modes of Operation 2
1.6. Mode 0 2
1.7. Mode 1 and Mode E. 2

Chapter 2: BISS Mode 1 — functional requirements 3
2.1. Overview 3
2.2. CA_descriptor 3

Chapter 3: BISS Mode E — functional requirements 4
3.1. Clear Session Word 4
3.2. Encrypted Session Word 4
3.3. Decryption scheme 4
3.3.1. Overview. 4
3.3.2. Unit identifiers 5
a) Injected ID – mandatory 5
b) Buried ID – optional for the supplier 6
3.3.3. Decryption function 6
a) Mapping. 6
b) Interoperable function 6
3.3.4. Post-processing function 8

Bibliography 9

Glossary of Terms. 10

Appendix A: DES usage in BISS-E 11

Chapter 1
Introduction

1.1. Overview

The rapid increase in the use of Digital Satellite News Gathering (DSNG) technology has resulted in the availability of digital codec equipment from a number of vendors. At the same time, the absence of standard methods for the securing and scrambling of DSNG broadcasts has spawned the development of several different proprietary security mechanisms.

The widespread acceptance of DVB standards now makes is possible to propose and provide a security mechanism that offers interoperability between the equipment of different DSNG vendors. This would enable broadcasters to combine equipment from several vendors, while making systems more future-proof.

The **Basic Interoperable Scrambling System** (BISS) is based on the DVB-CSA specification [1], and the use of fixed clear keys called **Session Words** (SWs). BISS specification Mode 1 is used for DSNG.

BISS specification Mode E (BISS with Encrypted keys – referred to as BISS-E) introduces an additional mechanism to accept the insertion of **Encrypted Session Words** (ESWs) while, at the same time, conserving interoperability. This mechanism is backward compatible with BISS specification Mode 1.

1.2. Nomenclature

Throughout this document, the following definitions are used:

- **Scrambler** relates to the overall mechanisms required to meet the DVB-CSA specification.
- **Session Word** relates to the word assigned during a transmission by the Management Centre.
- **Unit** relates to a device for which this specification might apply.
- **Management Centre** refers to an organization controlling or managing the conditional access system.
- **Decryption function** refers to a logical function used to decrypt the Encrypted Session Words, with the help of a key.
- **Interoperable function** refers to a decryption function that shall be embedded in all units.

1.3. Notations

The bits in binary numbers or sequences are numbered from the left, according to engineering notation. Bit 0 is on the right and is the least significant one; the bit on the left is the most significant one.

Here is an example of engineering notation for an n-bit number:

$$b_{n-1}b_{n-2.} \ldots\ b_1b_0$$

1.4. Security requirements

The DSNG model requires the direct entry of a Session Word at the transmitter and receiver, to control access to the transmission. The sender and receiver(s) of the transmission share the SW, such that only the intended parties will receive the transmission, outlined as follows:

1) The Session Word is entered at the DSNG unit in the field, or at the transmitting earthstation.
2) The Session Word is entered at the receiving IRDs.
3) If the Session Words are the same, then the IRDs are able to decrypt the broadcast.
4) If the Session Words are different, then the broadcast is not received.

The security requirements for fixed contribution systems are somewhat different to the DSNG model. The secure exchange of SWs is fundamental to such systems and is achievable by encrypting them.

1.5. Modes of Operation

The Scrambler must be capable of supporting the following three modes of operation:

» **Mode 0**: No scrambling.
» **Mode 1**: All components are scrambled by a fixed **Control Word** (CW), derived from a clear SW.
» **Mode E**: All components are scrambled by a fixed CW, derived from an **Encrypted Session Word** (ESW).

The scrambling mechanism, as defined in the DVB-CSA specification, shall be applied at the Transport level only.

A **Conditional Access Table** (CAT) shall be present in the multiplex for BISS Mode 1 and BISS-E, although the table shall be empty as no **Entitlement Management Message** (EMM) stream will be present.

Note: A Scrambler that only supports a subset of the defined modes of operation must do so according to an imposed hierarchy. A Scrambler providing support for Mode E must also support Modes 0 and 1.

1.6. Mode 0

The Scrambler must be capable of disabling the scrambling operation. In this mode, there will be no *CA_descriptor* in the **Programme Map Table** (PMT) and no **Entitlement Control Message** (ECM) stream. The *Transport_Scrambling_Control* bits of the Transport Packets will be set to “`00`”.

1.7. Mode 1 and Mode E

The complete specifications for Mode 1 and Mode E are given in *Chapter 2* and *Chapter 3* respectively.

Chapter 2
BISS Mode 1 — functional requirements

2.1. Overview

This mode has been designed specifically for DSNG applications, fly-away operations, emergency situations, etc. It may also be used as a fall-back solution while using the complete BISS-E system. In Mode 1, a fixed 12-character SW is inserted in the scrambler. The 64-bit CW is derived from the SW according to the DVB-CSA specification.

Manual entry of the SW shall be in hexadecimal notation, with the digits entered most-significant-nibble first, i.e. from left to right as viewed in hexadecimal notation.

For example, `0xA13DBC42908F` would be entered in the following sequence: `A,1,3,D,B,C,4,2,9,0,8,F`.

Remote entry of the SW shall also be provided, although the specification of that interface is beyond the scope of this document.

The Scrambler shall ensure that the SW cannot be changed more than ten times in a 5-minute period and that there is a minimum of 10 seconds between changes.

In this mode there will be a *CA_descriptor* in the PMT, present at programme level, but no ECM stream. A single unique *CA_System_ID* is assigned to identify BISS.

The *Transport_Scrambling_Control* bits of the Transport Packets shall be set to "`10`".

2.2. CA_descriptor

The *CA_descriptor* which must be present in the PMT to support BISS is defined in *Table 1*.

Semantics:

CA_system_ID: this is a 16-bit field indicating the type of CA system applicable for the associated ECM streams. The value of this field for BISS is `0x2600`. See [2].

CA_PID: this is a 13-bit field indicating the Packet Identification Number (PID) of the Transport Stream packets that shall contain the ECM information. For BISS, no ECM information is required, so this field shall contain the value `0x1FFF`.

Table 1
Conditional access descriptor – Mode 1.

Syntax	No. of bits	Identifier
CA_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
CA_system_ID	16	uimsbf
reserved	3	bslbf
CA_PID	13	uimsbf
}		

Chapter 3
BISS Mode E — functional requirements

3.1. Clear Session Word

The unit shall be compliant with BISS Mode 1. It shall support the insertion of a 12-character clear SW through the front panel and through a remote control interface. It shall use the SW as specified in *Chapter 2* (BISS Mode 1).

The clear SW, once entered via the user interface or remote control port, shall not be readable through any unit interface.

3.2. Encrypted Session Word

The unit shall support the insertion of ESWs through the front panel and through a remote control interface. The definition of the remote control port is outside the scope of this document.

The ESW is a 16-character number that is transformed by the unit into a 12-character clear SW. The clear SW is then used by the unit to decrypt the broadcast according to *Chapter 2* (BISS Mode 1).

Once the ESW has been entered via the front panel or via the remote control interface, it shall be impossible to read it back through any unit interface.

The manual entry of the ESW shall be in hexadecimal form; the 16 digits are entered with the most-significant nibble first (i.e. the left-most nibble).

For example, if the ESW is `0xF76EE249BE01A286`, it shall be entered in the following sequence:

`F,7,6,E,E,2,4,9,B,E,0,1,A,2,8` and `6`.

3.3. Decryption scheme

3.3.1. Overview

The equipment shall include the following features:

- An **identifier**, denoted ***ID***, comprising a 14-character hexadecimal word which shall be *injected* by the user and shall be used as the default. The injected ID is mandatory. Optionally, in addition, the supplier may *bury* an ID. In this case, the user shall actively select the buried ID.
- A **DES decryption function**, denoted ***f( )***, as described in *Section 3.3.3*. Additional functions may be supplied but are beyond the scope of this document.
- A **simple post-processing function**, denoted ***P( )***, as described in *Section 3.3.4*.

The processing of the ESW in the unit to provide the clear SW is illustrated in *Fig. 1* and further examples are described in *Appendix A*. The role of the Management Centre in generating the ESW, in accordance with the

Data Encryption Standard (DES), is indicated in *Appendix A* but a detailed specification is outside the scope of this document.

The mapping of the ID is a simple expansion from 56 to 64 bits, by adding an odd parity bit after every 7 bits. The reduction of the decrypted SW from 64 to 48 bits is obtained by deleting the first and last bit of each byte (see the examples given in *Appendix A*).

After the application of the post-processing function, ***P( )***, the clear SW is obtained to feed the BISS equipment as in Mode 1.

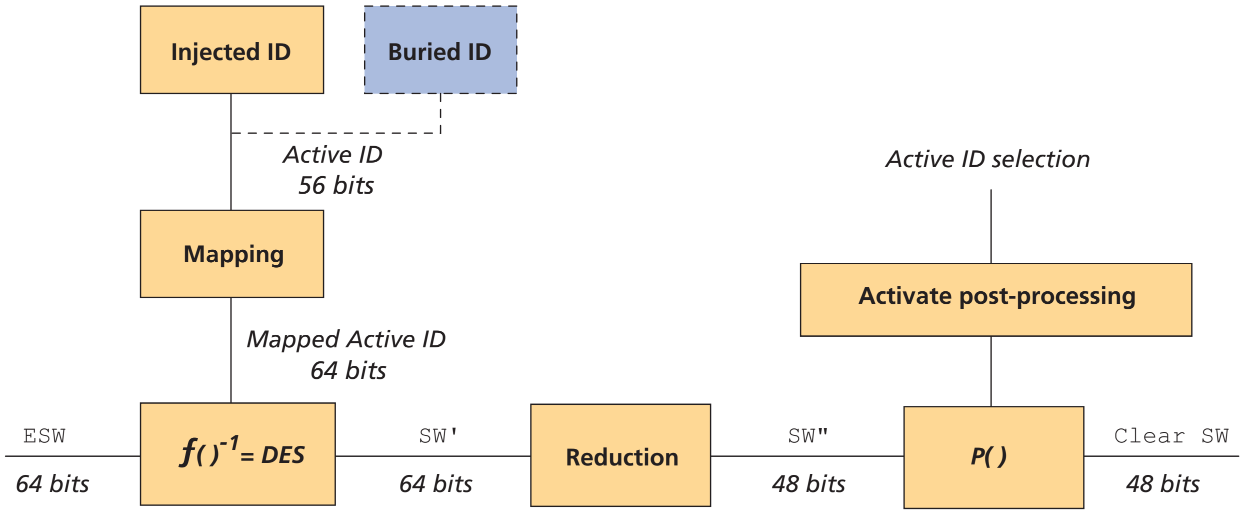

Figure 1
The signal processing required to produce a clear Session Word.

3.3.2. Unit identifiers

This document specifies two types of identifiers for each unit.

1) An **injected identifier** (ID_i) which is a secret key embedded in the unit. This is ***mandatory***.
2) Additionally, the manufacturer may provide a **buried identifier** (ID_b), defined by the manufacturer and linked uniquely to the device itself. This is not mandatory, but if implemented it shall comply with this document.

A user shall be able to select the identifier of his choice via the front panel and the remote control interface. The selected identifier is used as the active ID to decrypt the ESW.

a) Injected ID – mandatory

The injected ID is a 14-character identifier that can be entered in the BISS unit by the user at any time.

Units shall support the insertion of the injected ID through its front panel and through its remote control interface. There shall be no mechanism for reading back part or all of the injected ID via any unit interface.

The same ID can be injected in more than one piece of equipment, e.g. for redundancy management.

The manual or remote entry of the injected ID shall be in hexadecimal form; the 14 digits are entered with the most-significant nibble first (i.e. the left-most nibble).

For example, if the injected ID is `0xF09A423F56738A`, it shall be entered in the following sequence:

`F,0,9,A,4,2,3,F,5,6,7,3,8` and A.

b) Buried ID – optional for the supplier

The buried ID is a 14-character identifier that uniquely identifies a particular unit. Buried IDs are optional.

Two different units shall have a different buried ID, at least for the equipment produced by the same manufacturer. Units from different manufacturers could have the same buried ID, but that would be a fortuitous case.

The manufacturer shall ensure that, without his agreement, nobody can modify the buried ID.

3.3.3. Decryption function

Units shall implement the interoperable function specified in *Section b)* below. Additional decryption functions may optionally be implemented. In this case, it shall be possible to select the decryption function via the front panel and the remote control interface. The definition of these additional functions is outside the scope of this document.

The interoperable function is mandatory on all units. It uses the 64-bit ESW and the 64-bit Mapped Active ID to compute a 12-character word called `SW"`.

a) Mapping

The Mapped Active ID is derived from the 56-bit Active ID by mapping the 56 Active ID bits into the upper seven bits of a sequence of 8 bytes, then the LSB of each byte is set such that the resultant byte has odd parity.

Table 1
Mapping of Active ID to Mapped Active ID.

Active Key Bit	Mapped Byte / Bit	Mapped Key Bit
55	0 / 7	63
54	0 / 6	62
53	0 / 5	61
52	0 / 4	60
51	0 / 3	59
50	0 / 2	58
49	0 / 1	57
Odd_parity	0 / 0	56
48	1 / 7	55
Etc.	Etc.	Etc.

[Note that the above uses engineering notation: msb = bit 63, lsb = bit 0]

b) Interoperable function

The interoperable function is the simple DES algorithm used in the Electronic Codebook (ECB) mode and the decrypt state. This function is described in FIPS PUB 46-3 [3] and FIPS PUB 81 [4]. The algorithm key is the 64-bit Mapped Active ID [1]. The data bloc to decrypt is the ESW. Note that this document uses *engineering* notation (msb = bit 63, lsb = bit 0), while DES uses *FIPS* notation (msb = bit 1, lsb = bit 64). Hence the bit mapping is:

» DES key (1 ... 64) → Mapped Active ID (63 ... 0)

» DES data bloc (1 ... 64) → ESW (63 ... 0)

1. The DES algorithm does not use bits 8, 16, 24, 32, 40, 48, 56 and 64 of the key, numbered with the FIPS notation. Hence the useful key length is actually 56 bits, which is compatible with the exportation restrictions.

The result of the decryption algorithm is defined in 64 bits and is called `SW'`.

The mapping between `SW'` and `SW"` is given in *Table 2*. This mapping simply removes the most significant and the least-significant bit of each byte. The third column of the table is the DES result, where the bits are numbered using FIPS notation.

Table 2
Mapping between `SW'` and `SW"`

SW"(47)	SW'(62)	D(2)
SW"(46)	SW'(61)	D(3)
SW"(45)	SW'(60)	D(4)
SW"(44)	SW'(59)	D(5)
SW"(43)	SW'(58)	D(6)
SW"(42)	SW'(57)	D(7)
SW"(41)	SW'(54)	D(10)
SW"(40)	SW'(53)	D(11)
SW"(39)	SW'(52)	D(12)
SW"(38)	SW'(51)	D(13)
SW"(37)	SW'(50)	D(14)
SW"(36)	SW'(49)	D(15)
SW"(35)	SW'(46)	D(18)
SW"(34)	SW'(45)	D(19)
SW"(33)	SW'(44)	D(20)
SW"(32)	SW'(43)	D(21)
SW"(31)	SW'(42)	D(22)
SW"(30)	SW'(41)	D(23)
SW"(29)	SW'(38)	D(26)
SW"(28)	SW'(37)	D(27)
SW"(27)	SW'(36)	D(28)
SW"(26)	SW'(35)	D(29)
SW"(25)	SW'(34)	D(30)
SW"(24)	SW'(33)	D(31)
SW"(23)	SW'(30)	D(34)
SW"(22)	SW'(29)	D(35)
SW"(21)	SW'(28)	D(36)
SW"(20)	SW'(27)	D(37)
SW"(19)	SW'(26)	D(38)
SW"(18)	SW'(25)	D(39)
SW"(17)	SW'(22)	D(42)
SW"(16)	SW'(21)	D(43)
SW"(15)	SW'(20)	D(44)

Table 2
Mapping between SW′ and SW″

SW″(14)	SW′(19)	D(45)
SW″(13)	SW′(18)	D(46)
SW″(12)	SW′(17)	D(47)
SW″(11)	SW′(14)	D(50)
SW″(10)	SW′(13)	D(51)
SW″(9)	SW′(12)	D(52)
SW″(8)	SW′(11)	D(53)
SW″(7)	SW′(10)	D(54)
SW″(6)	SW′(9)	D(55)
SW″(5)	SW′(6)	D(58)
SW″(4)	SW′(5)	D(59)
SW″(3)	SW′(4)	D(60)
SW″(2)	SW′(3)	D(61)
SW″(1)	SW′(2)	D(62)
SW″(0)	SW′(1)	D(63)

3.3.4. *Post-processing function*

The post-processing function ***P( )*** converts SW″ into the clear SW, as shown in the example below. The conversion gives different results, depending on the type of ID used to decrypt the ESW. When the active key is the **injected ID**, function ***P( )*** is the **identity function** (i.e. SW″ = SW). If the optional **buried ID** is used, function ***P( )*** consists of **rotating** the SW″ by **one bit to the right**.

If SW" = $b_{47}\, b_{46} \ldots\ b_1\, b_0$

SW = ***P(***SW″***)*** =		
	$b_{47}\, b_{46} \ldots\ b_1\, b_0$	If the active key is an **injected ID**
	$b_0\, b_{47}\, b_{46} \ldots\ b_2 b_1$	If the active key is a **buried ID**
	Undefined	**Other** cases

In **other** cases, the definition of ***P( )*** is outside the scope of this document. It shall however have a different mathematical behaviour (i.e. it shall produce different results) than when the active key is either the *injected* ID or the *buried* ID.

Bibliography

[1] **DVB Common Scrambling Algorithm, Version 2.0. (July 2002)**
http://portal.etsi.org/dvbandca/DVB/DVBINTRO.asp

[2] ETR 289: **Digital broadcasting systems for television, sound and data services; Support for use of scrambling and conditional access (CA) within digital broadcasting systems**.
http://webapp.etsi.org/workprogram/SimpleSearch/QueryForm.asp

[3] FIPS PUB 46-3: **Data Encryption Standard**.
http://www.itl.nist.gov/fipspubs/by-num.htm

[4] FIPS PUB 81: **DES Modes of Operation**.
http://www.itl.nist.gov/fipspubs/by-num.htm

Glossary of Terms

BISS	Basic Interoperable Scrambling System
bslbf	Bit string, left bit first
CA	Conditional Access
CAT	Conditional Access Table
CSA	(DVB) Common Scrambling Algorithm
CW	Control Word
DES	Data Encryption Standard
DSNG	Digital Satellite News Gathering
DVB	Digital Video Broadcasting
ECB	Electronic Codebook
ECM	Entitlement Control Message
EMM	Entitlement Management Message
ESW	Encrypted Session Word
lsb	Least Significant Bit
LSB	Least Significant Byte
MC	Management Centre
msb	Most Significant Bit
MSB	Most Significant Byte
PID	Packet Identification number
PMT	Programme Map Table
SW	Session Word
uimsbf	Unsigned integer, most significant bit first

Appendix A
DES usage in BISS-E

The following tables show six examples to the processes described on *page 13*, for two different cases:

» The "Active ID" is an "injected ID" (Table A.1)

» The "Active ID" is a "buried ID" (Table A.2)

Table A.1 — Results while using "injected ID".

Encrypted Session Word (64 bits) *(Note 1)*	Active ID (56 bits) *(Note 2)*	Expanded Session Word (64 bits) *(Note 4)*
	Mapped Active ID (64 bits) *(Note 3)*	Session Word (after reduction) (48 bits) *(Note 5)*

Example 1:

E81816B87E5CF9C4	FB5F9C585DD359	B2CB8F3948770EF9
	FBAEE68A85EF4CB3	6651DC93B1FC

Example 2:

FE0E810E96648C43	DC91106F13C87B	AA8130F0B47D381B
	DC49450DF19E20F7	5406386BE70D

Example 3:

51B276C10BA89683	89F27FB3ACA107	066283A18293C5A6
	89F89EF73B64850E	0F1050049893

Example 4:

1D599C25C30C2C11	85513F83A5FF36	0A74BCCBCA80AAFE
	85A84FF13B2FFD6D	17A7A594057F

Example 5:

ED89E0D818B43B98	CD23BF7D8F6BC3	840D91C0F6444C27
	CD91EFEFD97AAE86	086220EE2993

Example 6:

DA457923D39BDB81	E0A48CAA093AC7	7EE18611180D64A2
	E0522394A149EA8F	FF00C8306C91

Table A.2 — Results while using "buried ID".

Encrypted Session Word (64 bits) (Note 1)	Active ID (56 bits) (Note 2)	Expanded Session Word (64 bits) (Note 4)
		Session Word (after reduction) (48 bits) (Note 5)
	Mapped Active ID (64 bits) (Note 3)	Session Word (after post processing P) (48 bits) (Note 6)

Example 1:

389C30A2AB29F3F2	FB5F9C585DD359	E6149C72936D9F71
		CCA3B92763F8
	FBAEE68A85EF4CB3	6651DC93B1FC

Example 2:

288871F187278E6A	DC91106F13C87B	D500E2E1EBF8F135
		A80C70D7CE1A
	DC49450DF19E20F7	5406386BE70D

Example 3:

53C0C292CC2A88B3	89F27FB3ACA107	0E4405410427894C
		1E20A0093126
	89F89EF73B64850E	0F1050049893

Example 4:

6EC275343C161742	85513F83A5FF36	6EC275343C161742
		2F4F4B280AFE
	85A84FF13B2FFD6D	17A7A594057F

Example 5:

0E027A2DBDB5622C	CD23BF7D8F6BC3	8899A203EE8A194D
		10C441DC5326
	CD91EFEFD97AAE86	086220EE2993

Example 6:

C5E9F325179F97A2	E0A48CAA093AC7	7FC10CA0311A4846
		FE019060D923
	E0522394A149EA8F	FF00C8306C91

Notes:

1) **ESW** is received from the **Management Centre** (64 bits) and is entered manually or via the remote control port.

2) **Active ID** is used for the DES^{-1} decryption process (56 bits).

3) The **Active ID** is expanded to 64 bits by taking each group of 7 bits, adding an odd parity bit and forming a byte (the parity bit is the least significant bit of the byte). This **Mapped Active ID** is used as the key of the DES^{-1} decryption process (in ECB mode).

4) **Expanded Session Word** is the result of the $\mathbf{DES^{-1}}$ process (64 bits).

5) **Session Word after reduction** (`SW″`) is calculated by taking each byte of **Expanded Session Word**, removing both its lsb and msb, and concatenating 8 times the 6-bit words to 48 bits.

6) According to the **Active ID Selection** (Buried or Injected), a post-processing function ***P( )*** is activated on the `SW″`. The function ***P( )*** is defined to be either **Identity** function (for Injected ID), or **rotate right by 1 bit** `SW″` (for the buried ID). The output of ***P( )*** is the **Session Word** (48 bits).

<u>The ***Equipment*** processes</u>:

Getting the **Encrypted Session Word** (ESW) into the equipment:

1) Map the **Active ID** to get the **Mapped Active ID**.

2) Do the decryption using DES^{-1} (at ECB mode) with the **Mapped Active ID** as <u>key</u>, and **ESW** as <u>data</u>.

3) The result is the **Expanded Session word** (`SW′`).

4) Reduce the result to a 48-bit word as in *Note 5* above.

5) Do the post-processing function as described in *Note 6*. The output of ***P( )*** is denoted as the **Session Word** (`SW`).

<u>The **Management Centre** processes</u>:

Choosing a **Session Word**:

1) Do the preprocessing function (inverse of the post-processing function, described in *Note 6*), and get `SW″`.

2) Do the Expansion (expand each 6 bits to one byte, filling both the least and the most significant bits of each byte by 2 random bits) and get **Expanded Session Word** (`SW′`).

3) Map the **Active ID** to get the **Mapped Active ID**.

4) Do the encryption using DES (at ECB mode) with the **Mapped Active ID** as <u>key</u>, and the **Expanded** `SW` as <u>data</u>.

5) Get the results of the encryption (ESW), and send it to the equipment.