Specification Bulletin No. 144
First Edition June 2014

Terminal Unpredictable Number generation

This specification update introduces an algorithm for generating Terminal Unpredictable Numbers.

Effective Dates

This bulletin is effective immediately.

Applicability

This Specification Bulletin applies to:

- *EMV Integrated Circuit Card Specifications for Payment Systems Version 4.3 Book 2 Security and Key Management*
- *EMV Integrated Circuit Card Specifications for Payment Systems Version 4.3 Book 4 Cardholder, Attendant, and Acquirer Interface Requirements*

Related Documents

- None

Description

This bulletin introduces a specific approved method into Book 2 for generating Terminal Unpredictable Numbers (UNs). This bulletin also clarifies that it is permitted that a terminal use the same UN throughout a transaction.

Background

The Terminal Unpredictable Number (tag 9F37) is sent to the card during an EMV transaction and is included by the card in the data protected by the card cryptogram (whether it be a DDA/CDA signature or an Application Cryptogram). The purpose of the UN is to ensure that card cryptograms are freshly produced and not replays of cryptograms previously obtained from the card. It is for this reason that the UNs generated by a terminal must be unpredictable. It should be infeasible for an attacker to guess or control the values of the UNs that the terminal generates (for them each possible 32-bit value should be equally likely even given knowledge of previous UN values).

The EMV Acquirer and Terminal Security Guidelines includes guidance and best practice for UN generation and a schema for generating UNs. The purpose of the current bulletin is to go beyond this schema and introduce a specific algorithm for an EMV kernel to generate UNs. This algorithm may be considered as a whitening or washing process to be applied to a random number produced outside the kernel (e.g. a PCI approved hardware RNG). However this method is also considered suitable for

generating UNs in the absence of such an external random number source (e.g. in case the hardware RNG fails partially or completely).

Specification Change Notice

Please replace section 6.5.6 of Book 4 as follows:

Original text:

6.5.6 Unpredictable Number

The terminal shall be able to generate an Unpredictable Number, which may be used for input to the application cryptogram algorithm to ensure the unpredictability of data input to this calculation or for random transaction select ion for terminal risk management.

The Unpredictable Number could be generated by a dedicated hardware random number generator or could, for example, be a function of previous Application Cryptograms, the terminal Transaction Sequence Counter and other variable data (e.g. date/time). In the second example the function could be a hash function or more preferably a keyed encipherment function.

New text:

6.5.6 Unpredictable Number

The terminal shall be able to generate an Unpredictable Number (tag 9F37) to be used for input to the card cryptograms (Application Cryptograms and DDA/CDA signatures) so as to ensure the unpredictability of data input to this calculation and thereby the freshness of the cryptogram.

A terminal may use the same Unpredictable Number throughout a transaction. The Unpredictable Number could be generated by a dedicated hardware random number generator or could, for example, be a function of previous Application Cryptograms, the terminal Transaction Sequence Counter and other variable data (e.g. date/time). In the second example the function could be a hash function or a keyed encipherment function.

Section 11.3 of Book 2 provides an example of an approved method for generating the Unpredictable Number using a hash function.

Please insert the following text into Book 2 Section 11 Terminal Security and Key Management Requirements.

11.3 Unpredictable Number Generation

This section defines an approved method for generating the Terminal Unpredictable Number (UN).

The following functions are involved:

- SHA[] is SHA-256.
- LS4B[] outputs the least significant 4 bytes of the input

The following data elements are involved:

TID	Terminal ID	EMV Data element tag '9F1C'	8 bytes alpha numeric (see Table 33 of EMV Book 3)
IFDSN	IFD Serial	EMV Data element tag '9F1E'	8 bytes alpha numeric (see

	Number		Table 33 of EMV Book 3)
TVP	Time varying parameter	Date and time with finer granularity than 0.1 second	Implementation dependent. Should be an internal value not the external clock
RAND	Value from external RNG	Random value sourced outside the kernel (e.g. PCI-approved hardware RNG) to be used if available	8 bytes binary
P	Pre-image of UN	P is an internal register maintained in the terminal in volatile memory and never output.	32 bytes binary
Q	Persistent variant of P	Q shall be initialised to a terminal-unique random number prior to deployment. It is maintained in the terminal in non-volatile memory and never output. It is updated every time the power is cycled.	32 bytes binary

Note that Q is held in non-volatile memory and so persists when the terminal is powered down.

UN generation is as follows

1. Power-up

   Each time the terminal is powered-up it updates Q and generates a value P as follows:

   Set Q = SHA[Q || TVP || IFDSN || TID || RAND (if available)]
   Set P = Q

2. Per transaction
   - Before a transaction, the terminal generates a UN as follows:
     Set UN = LS4B(SHA[P || RAND (if available)])
   - After a transaction (even if it fails) refresh P as follows:
     Set P = SHA[P || TVP || RAND (if available) || AC (if available)]

3. Power-down
   Set Q = P