

**THE CENTRAL BANK OF THE RUSSIAN FEDERATION
(BANK OF RUSSIA)**

**Unified Formats
of Electronic Banking Messages**

**SECURITY OF ELECTRONIC MESSAGES
(EM packets)**

Version 2017.4.0

Moscow

2017

Table of Contents

1. SECURITY OF ELECTRONIC MESSAGES (EM PACKETS)	3
1.1. Scope	3
1.2. Requirements for the security of electronic messages (EM packet)	3
2. PROTECTION OF ELECTRONIC MESSAGES (EM PACKETS) AT THE APPLICATION LEVEL	5
2.1. Scope	5
2.2. Requirements for the protection of electronic messages (an EM packet) using SCs	6
2.3. Rules for generating and verifying an SC	10
3. PROTECTION OF ELECTRONIC MESSAGES (EM PACKETS) WITH AN AC	11
3.1. Scope	11
3.2. Requirements for the protection of an EM (EM packet) with an AC	11
3.3. Rules for generating and verifying an AC	15
3.4. Encryption	16
3.5. Compression	16

1. Security of Electronic Messages (EM packets)

1.1. Scope

This section describes the rules of generating, and verifying the AC and the security code for unified formats of electronic banking messages for the exchange of electronic messages between the business units of the Bank of Russia and credit organisations and other customers of the Bank of Russia located in the territory of the Russian Federation when making cashless payments in the currency of the Russian Federation.

The discrepancy resolution procedure during the exchange of electronic messages proves that there have been no changes in a sent message during a delivery to a recipient based on the use of means of controlling the integrity and confirming the authorship of messages provided by the sending party and the receiving party pursuant to the established order. In connection with this, a necessary requirement when using the UFEBM is the transmission of a message to the recipient in the form in which it was signed by the sender. To protect an electronic message in the light of this requirement, an AC is used.

An EM (EM packet) can be additionally protected at the technological level, for which a security code can be included among the EM (EM packet) elements. As protecting an EM packet or particular electronic messages forming part of a packet with a security code is an element of technological security, the requirement that the EM (EM packet) is to be sent to a recipient in the form in which it was protected with the SC by the sender is not necessary. However, the security code computation algorithm accepts binary data as input; consequently, the EM (EM packet) to be signed must be converted to a common form having an identical binary representation in any case and on any platform. Electronic messages are XML documents; therefore, to convert the EM (EM packet) to be signed and verified to a single form, it is necessary to use algorithms designated for processing XML documents. To convert an XML document to a common form having an identical binary representation in any case and on any platform, the W3C consortium recommends using a canonicalisation algorithm. The additional transformation (normalisation) allowing the removal of excessive information from an XML document enables the formation of an SC only for meaningful data, which makes it possible to protect information regardless of special features of markup.

1.2. Requirements for the security of electronic messages (EM packet)

The need to protect EMs (EM packets) in the payment system of the Bank of Russia using ACs is determined by regulations of the Bank of Russia.

The protection of EMs (EM packets) created in a business unit of the Bank of Russia using a security code (SC) is also determined by regulations of the Bank of Russia.

Note: In respect of document "Temporary Requirements for Ensuring the Security of Technologies for Processing Electronic Payment Documents in the system of the Central Bank of the Russian Federation" No. 60, dated 3 April 1997, the security code (SC) herein shall mean the processing AC.

In BoR RD, work on options for the protection of EMs (EM packets) using ACs and SCs may be arranged (see table 1). The first option can be used only for protecting EMs that are transmitted from a CO/BoR Customer to a business unit of the Bank of Russia. Under all security options, the use of an AC is required for an EM packet and for an unpacked EM.

The number of ACs and SCs for an EM (EM packet) that are **transmitted** during an exchange with a CO/BoR Customer depends on the security option (see table 1). When describing options for EM (EM packet) security using ACs and SCs, conventional designations were used (see table 2).

Table 1. Number of ACs and SCs for EMs during exchange with a CO/BoR Customer, depending on the security option

Option	EM	Data to be signed	Number of ACs	Number of SCs
1 ¹⁾	EM packet	EM packet	[1]	—
		each EM forming part of an EM packet	—	—
	EM ²⁾	EM	[1]	—

Option	EM	Data to be signed	Number of ACs	Number of SCs
2	EM packet	EM packet	[1]	[1]
		each EM forming part of an EM packet	—	—
	EM ²⁾	EM	[1]	[1]
3	EM packet	EM packet	[1]	—
		each EM forming part of an EM packet	—	n×[1]
	EM ²⁾	EM	[1]	[1]
¹⁾ Option 1 is admissible only for protection of EMs generated in CO/BoR Customers and FMOD ²⁾ Unpacketed EM				

T a b l e 2. Conditional designations used when describing options for EM (EM packet) protection using ACs and SCs

Designation	Description
[1]	One and only one instance of an AC or SC is required.
—	AC or SC does not apply
n×[1]	One SC per each EM in a packet (n shall mean the number of EMs forming part of a packet)

The **CDAS 'Signatura'** is used as a cryptographic information security tool.

2. Protection of electronic messages (EM packets) at the application level

A security code can be included among EM elements. The protection of an EM packet or particular electronic messages forming part of an EM packet with a security code is an element of technological protection.

This section contains the rules for executing SCs in an XML document and defines the parts of an XML document to be protected with SCs.

2.1. Scope



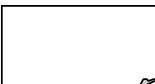

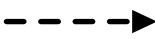
This section contains a description of the rules for executing, generating, and verifying the security code used within UFEBM for the exchange of electronic messages between the business units of the Bank of Russia and credit organisations and other customers of the Bank of Russia located in the territory of the Russian Federation when making cashless settlements in the currency of the Russian Federation.

Electronic messages are XML documents; a requirement that an EM (EM packet) be sent to a recipient in the form in which it was protected with an SC by the sender is not necessary. However, the security code computation algorithm accepts binary data as input; consequently, the EM (EM packet) to be signed must be converted to a common form having an identical binary representation in any case and on any platform. To convert an XML document to a common form having an identical binary representation in any case and on any platform, the W3C consortium recommends using a **canonicalisation** algorithm. The canonicalisation algorithm of XML documents [XML-c14n] converts an XML document to a form that makes it possible to determine the logical equivalence of this XML document with another XML document in canonical form. To determine whether two XML documents are logically equivalent, it is necessary to canonicalise each XML document pursuant to the canonicalisation rules determined by W3C and compare their canonical forms, byte by byte. If both canonical forms contain an identical sequence of bytes, the respective XML documents are logically equivalent.

The canonicalisation rules make it possible to obtain a binary representation of an XML document that does not depend on the parser and the operational platform, but they do not account for the specifics of UFEBM. For example, canonicalisation of an XML document does not involve the removal of nodes with ancillary information, whereas during electronic exchange in the cashless settlement system nodes with ancillary information (processing commands; comments) are not used—that is, information contained therein is ignored. Thus, when processing information from an XML document, only a part of the nodes is used, and all other nodes are ignored. The additional transformation (**normalisation**) allowing the removal of excessive information from an XML document enables the formation of an SC only for meaningful data, which makes it possible to protect information regardless of special features of markup. Generating SCs only for meaningful data regardless of the special features of markup of a particular initial instance of an XML document makes it possible to verify the signed EM (EM packet) irrespective of its storage format (an initial XML document or a document restored from relational data).

SC processing charts are provided in the figures below (see fig. 1, fig. 2). When drawing up the charts, conditional designations were used (see table 3).

Table 3. Conditional designations used when drawing up SC processing charts

Designation	Description	Designation	Description
	Process		Object
	XML document meeting [XML] requirements		Movement between object conditions
			Transfer of an object to a process

2.2. Requirements for the protection of electronic messages (an EM packet) using SCs

2.2.1. Namespaces

For this document version, the following namespaces are used:

“urn:cbr-ru:dsig:v1.1” (the prefix dsig).

N o t e : A namespace prefix does not have any meaning and is used only for tying the names of components and attributes to the designation of a namespace.

2.2.2. Structure and syntax of the security code

An element with an SC value can be added to the elements of any EM (EM packet). The element with an SC value is represented by an element from the namespace "urn:cbr-ru:dsig:v1.1" that can be added before the first child element of an EM (EM packet). A description of an element with an SC value is provided in the table below (see **t a b l e 4**). The element number '0' shows that the element shall precede an element with the number '1' from among the EM elements.

T a b l e 4. EM element with an SC value

Element description	Element type	Multipli city
0.SC value (any namespace="urn:cbr-ru:dsig:v1.1")	Component containing an SC value	[0..n]

N o t e : This notation does not describe the structure of an element with an SC value.

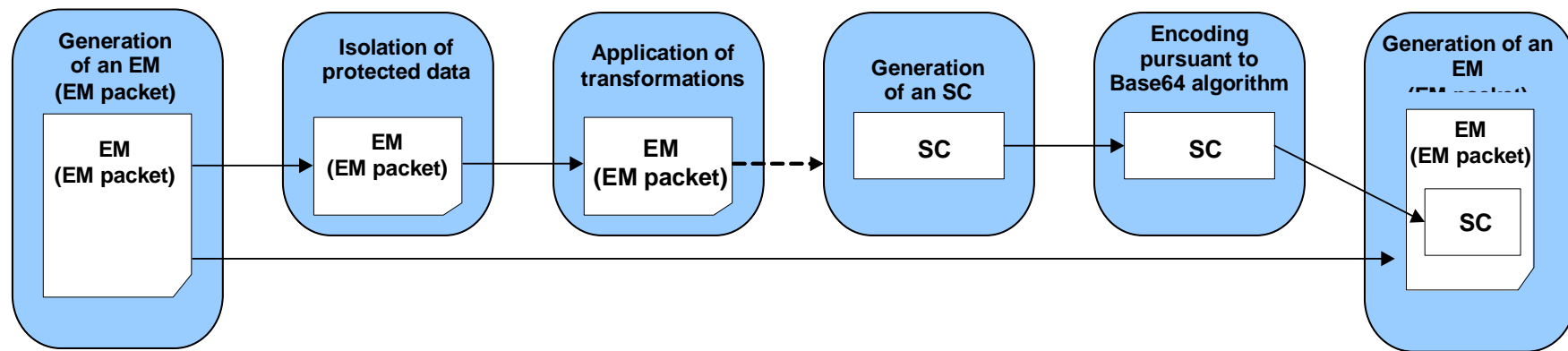


Fig. 1. SC generation chart

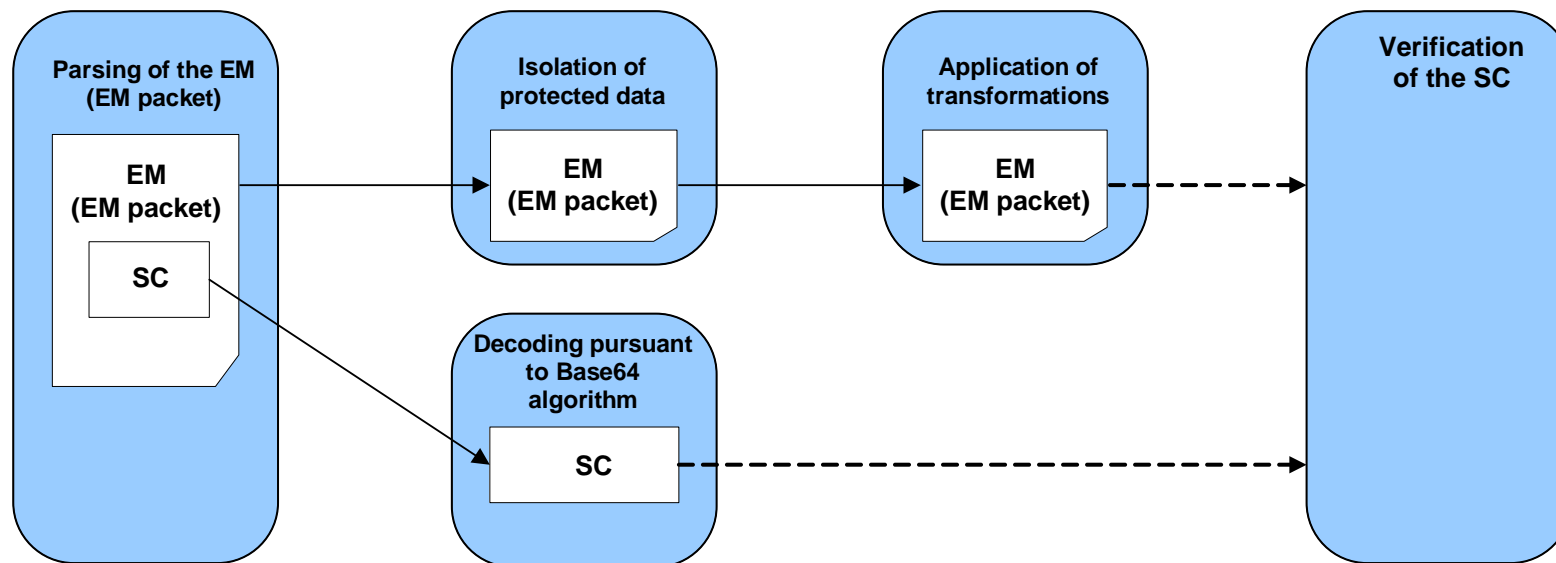


Fig. 2. SC verification chart

Structurally, the element with an SC value is represented by the component **dsig:SigValue**, in which the SC value calculated pursuant to the algorithm specified in the profile of parameters for protecting an EM (EM packet) with an SC is placed (see table 6). The SC value is provided in the format with which the CISS being used operates. Before placement in the component dsig:SigValue, the SC value is to be encoded pursuant to the algorithm [base64]. The structure of the component with the SC value is provided in the table below (see table 5).

The namespaces

“urn:cbr-ru:dsig:v1.1” (the prefix dsig)

“http://www.w3.org/2001/XMLSchema” (the prefix xsd)

Table 5. Component elements with the SC value

Element description	Element type	Multiplicity
0.SC value (dsig:SigValue)	xsd:base64Binary	[0..n]

Example: execution of the SC value:

```
<dsig:SigValue xmlns:dsig="urn:cbr-ru:dsig:v1.1">
RpxoZ6vnUXn9/nTSC9rkqeWt1NYTc+RxWZ5JbdFW6Vlg+ULhx7uDJFPRIqXJnIugF2xz1pgjCtmh
4hz9tLAg==</dsig:SigValue>
```

2.2.3. Profile of the parameters of protecting an EM (EM packet) with an SC

The document describing the exchange of electronic messages between the parties when making settlements through the settlement system of the Bank of Russia (in the Exchange Contract) stipulates the procedure for using protection of EMs (EM packets) with SCs. The protection of EMs (EM packets) with SCs applies pursuant to the profile of parameters of protection of an EM (EM packet) with SCs (see table 6). The profile of protection of an EM (EM packet) with SCs contains the list of specifications and algorithms that apply for converting an EM (EM packet) to a form ensuring its protection with the cryptographic information security system through the placement and verification of SCs. The template also determines the list and the procedure for transformations of an EM (EM packet) before the operation of generating and verifying the SC.

Table 6. Profile of parameters of protection of an EM (EM packet) with an SC

Algorithm	Identifier
EM (EM packet) transformations	
Transformation of an EM (EM packet) for conversion to a normalised form	urn:cbr-ru:dsig:v1.1#normalization
XML canonicalisation without comments [XML-c14n]	https://www.w3.org/TR/xml-c14n
EM (EM packet) encoding	
Encoding algorithm Base64	not used
SC value encoding	
Encoding algorithm Base64	http://www.ietf.org/rfc/rfc2045#base64

The cryptographic security of files with EMs shall be ensured by using a CISS having a certificate or a temporary permit of the Federal Security Service or a temporary permit of the Bank of Russia.

2.2.4. Reference to data to be signed

The location of the component from the namespace "urn:cbr-ru:dsig:v1.1" inside an EM (EM packet) unambiguously determines the part of EM (EM packet) that shall be protected: the **SC** always **protects a parent component** (including all its child components and attributes) in respect of the component with the SC value (**except for all child components** of the first level in respect of the root of the EM (EM packet) **containing the SC values**):

- In an EM that is not packeted, the SC protects the entire EM, except for all child components of the first level in respect of the root of EMs containing SC values.
- In an EM forming part of a packet, the SC protects the entire EM, except for all child components of the first level in respect of the root of EMs containing SC values.
- In an EM packet, the SC protects the entire EM packet, except for all child components of the first level in respect of the root of a packet of EMs containing SC values.

An illustration showing the data to be signed when generating an SC (for an EM forming part of a packet or an EM packet) is provided below (see fig. 3).

2.2.5. Transformations for isolation of data protected with SCs

When **generating** SCs for the isolation of data protected with SCs, the following actions are performed:

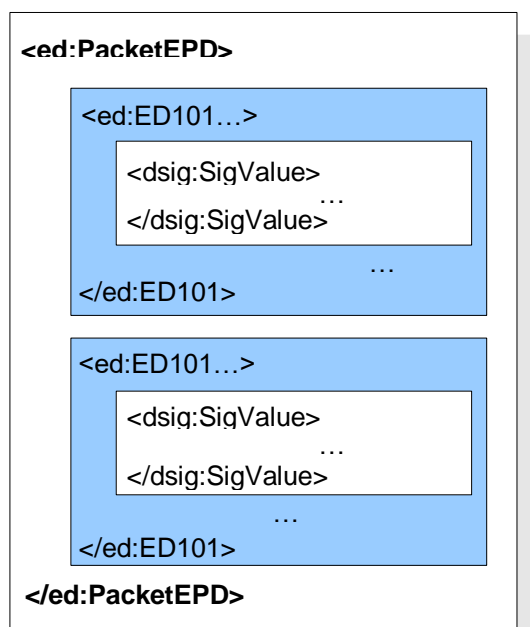
- An XML document whose root component is a component containing the EM (EM packet) to be signed (with all its child components and attributes) is generated.
- All dsig:SigValue components that are child components of the first level, if any, are removed from the root component.

Note: dsig:SigValue components that are components of the first level in respect of the root of the EM (EM packet) to be signed can exist in the part of the EM (EM packet) to be signed if the part of the EM (EM packet) to be signed is already protected with an SC.

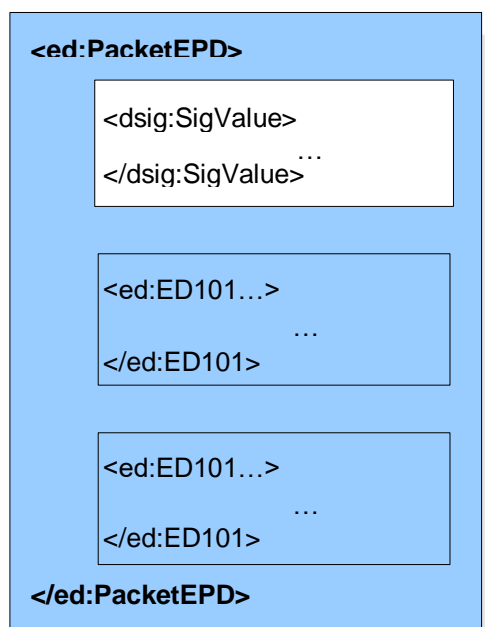
When **verifying** SCs for the isolation of data protected with SCs, the following actions are performed:

- An XML document whose root component is a parent component in respect of the component dsig:SigValue with the value of the SC to be verified (with all its child components and attributes) is to be formed.
- All dsig:SigValue components that are child components of the first level are to be removed from the root component.

Protection of the EM ED101 forming part of an EPM packet using an SC



Protection of the EM PacketEPD with an SC



Data not to be signed



Data to be signed

2.3. Rules for generating and verifying an SC

2.3.1. Rules for generating an SC

The process of generating an SC consists of the following phases:

- a) Generating an XML document containing the protected data of an EM (EM packet) to be protected with an SC.
- b) Isolation of data to be protected with an SC pursuant to 2.2.5 from the generated XML document.
- c) Applying the transformations provided in the profile of parameters for protecting an EM (EM packet) with an SC to an XML document containing only protected data obtained at the previous phase: transformation of the XML document to a normalised form and canonicalisation. As the result of canonicalisation, a byte array will be obtained.
- d) Generating the SC (calculating its value): Calling the CISS function for generating an SC with transfer of the byte array obtained at the previous phase.
- e) Encoding of the SC value obtained at the previous phase (in the format of an SC library, without isolation of the SC value) pursuant to the algorithm [base64].
- f) Placing the SC value encoded at the previous phase in the component sig:SigValue.
- g) Adding the component sid:SigValue to an XML component containing the protected part of the EM (EM packet) before the first child component of the first level in respect of the root of the protected part of the EM (EM packet).

2.3.2. Rules for verifying an SC

The process of verifying SCs on the protected part of an EM (EM packet) consists of the following phases:

- a) Receiving an XML component containing the part of an EM (EM packet) protected with an SC.
- b) Isolating the SC value from the component sig:SigValue.
- c) Decoding the SC value isolated at the previous phase pursuant to the algorithm [base64].
- d) Isolating data protected with an SC pursuant to 2.2.5 from an XML component obtained at the phase described in list a).
- e) Applying the transformations provided in the profile of parameters of protection of an EM (EM packet) with an SC to the XML document obtained at the previous phase: transformation of the XML document to a normalised form and canonicalisation. As the result of canonicalisation, a byte array will be obtained.
- f) Verifying the SC: Calling the CISS function for verifying an SC, transferring the byte array obtained at the phases described in lists e) and c) to it.

3. Protection of electronic messages (EM packets) with an AC

An EM shall be protected with an AC. A necessary requirement is the transmission of a message to a recipient in the form in which it was signed by the sender.

This section contains the rules for executing ACs in EMs and defines the parts of an XML document to be signed.

3.1. Scope

This section contains a description of the rules for executing, generating, and verifying the AC used within UFEBM for the exchange of electronic messages between the business units of the Bank of Russia and credit organisations and other customers of the Bank of Russia located in the territory of the Russian Federation when making cashless settlements in the currency of the Russian Federation.







The exchange of documents in XML format gives rise to the possibility of unauthorised access to information. To prevent unauthorised access to information, support of data encryption is implemented at the application level. To save costs for data transfer and storage, support of data compression is implemented at the application level.

The discrepancy resolution procedure during the exchange of electronic messages consists of evidence that there have been no changes in a sent message during a delivery to a recipient based on the use of means of controlling the integrity and confirming the authorship of messages provided by the sending party and the receiving party pursuant to the established procedure. In connection with this, a necessary requirement when using the UFEBM is the transmission of a message to the recipient in the form in which it was signed by the sender. That means that the EM (EM packet) to be verified shall coincide precisely (up to a byte) with the signed EM (EM packet). Thus, the signed EM (EM packet) must be transferred in the AC envelope in its binary representation. To transfer binary data in an XML document, the specification [XML-schema] recommends using the encoding algorithm [base64].

Data transformed pursuant to the algorithm [deflate] can be unambiguously restored from the compressed sequence, and the chosen encryption algorithm must also unambiguously restore data at decryption. The use of the algorithms [base64] and [deflate] guarantees the identity of the binary representation of the signed EM (EM packet) to be verified.

AC processing charts are provided below (see fig. 4, fig. 5). When drawing up the charts, conditional designations were used (see table 7).

Table 7. Conditional designations used when drawing up AC processing charts

Designation	Description	Designation	Description
	Compulsory process		Object
	Optional process		Movement between object conditions
	XML document meeting [XML] requirements		Transfer of an object to a process

3.2. Requirements for the protection of an EM (EM packet) with an AC

3.2.1. Namespaces

For this document version, the following namespaces are used:

"urn:cbr-ru:dsig:v1.1" (the prefix dsig)

"urn:cbr-ru:dsig:env:v1.1" (the prefix sen)

N o t e : A namespace prefix does not have any meaning and is used only for tying the names of components and attributes to the designation of a namespace.

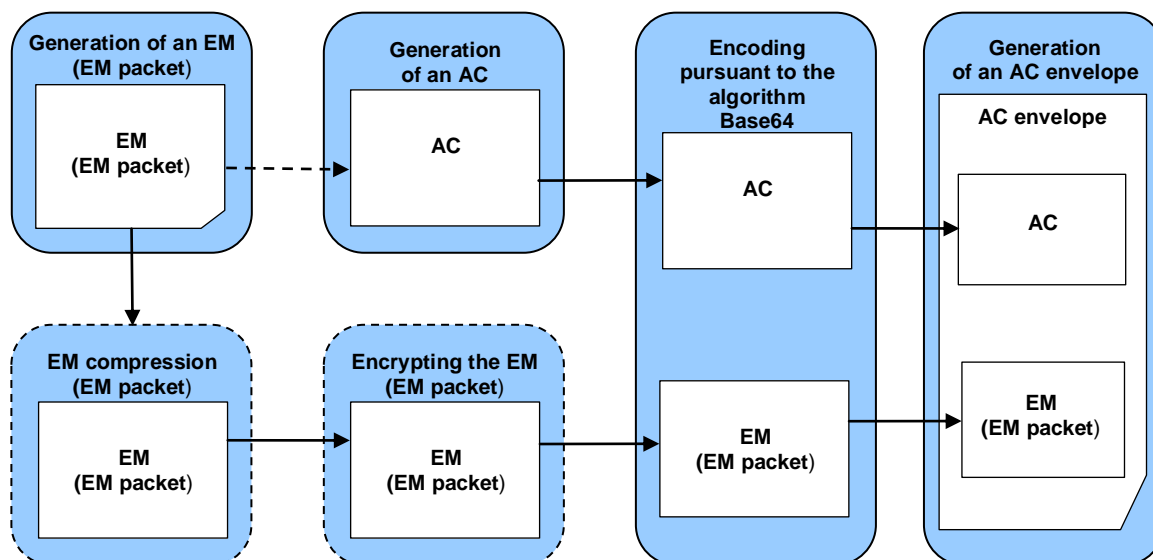


Fig. 4. AC generation chart

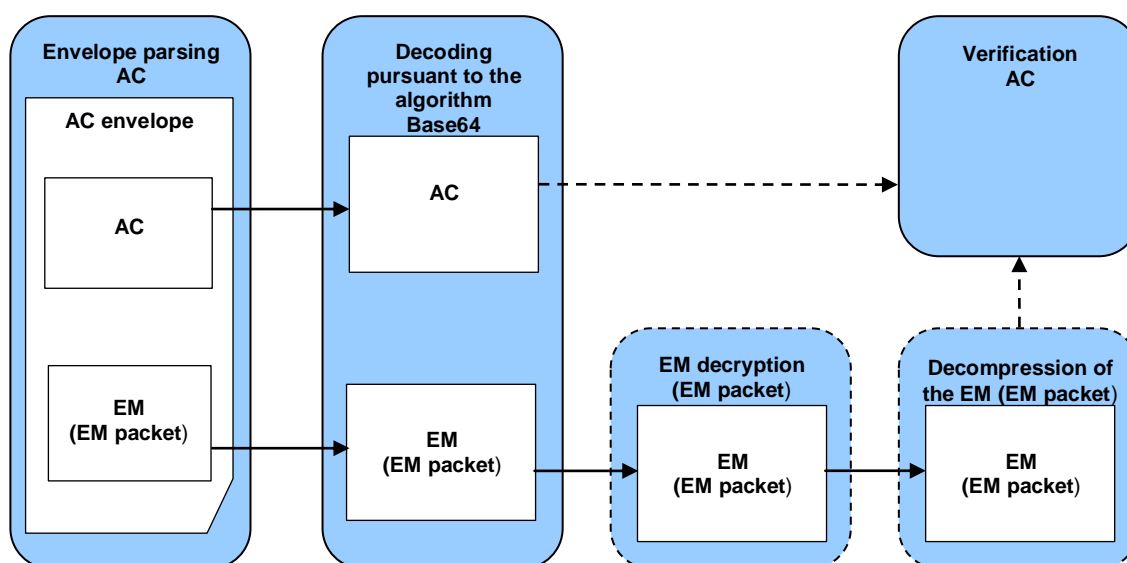


Fig. 5. AC verification chart

3.2.2. Structure and syntax of the AC envelope

The AC envelope contains the AC value and the signed EM (EM packet). The AC envelope is presented in the form of the component **sen:SigEnvelope**.

The AC envelope consists of:

- A container for the AC value that is presented in the form of the component **sen:SigContainer**. The container (the component:SigContainer) contains a component from the namespace "urn:cbr-ru:dsig:v1.1" with the AC value.
- The component **sen:Object** that contains the signed EM (EM packet) encoded pursuant to the algorithm [base64]. Prior to encoding pursuant to the algorithm [base64], EM (EM packet) can be compressed and/or encrypted.

A description of the AC envelope elements (the component **sen:SigEnvelope**) is provided in the table below (see table 8).

Namespace

"urn:cbr-ru:dsig:env:v1.1" (the prefix **sen**)

Table 8. AC envelope elements

Element description	Element type	Multiplicity
1 Container for the AC (sen:SigContainer)		[1]
1.1 AC value (any namespace="urn:cbr-ru:dsig:v1.1")	Component containing the AC value	[1]
2 Container for the object to be signed. (sen:Object)	Component containing the signed EM (EM packet) encoded pursuant to the algorithm [base64]	[1]

Note: This notation does not describe the structure of an element with the AC value.

Structurally, the element with the AC value is represented by the component **dsig:MACValue**, in which the AC value calculated pursuant to the algorithm specified in the profile of parameters of protection of an EM (EM packet) with an AC is placed (see table 10). The AC value shall be presented in a format with which the CISS in use operates; the AC value is not to be isolated. Before placement in the component **dsig:MacValue**, the AC value is to be encoded pursuant to the algorithm [base64]. The structure of the component with the AC value is provided in the table below (see table 9).

The namespaces

"urn:cbr-ru:dsig:v1.1" (the prefix **dsig**)

"http://www.w3.org/2001/XMLSchema" (the prefix **xsd**)

Table 9. Component requisites with the AC value

Element description	Element type	Multiplicity
1 AC value (dsig:MACValue)	xsd:base64Binary	[1]

Example: execution of the AC value:

```
<dsig:MACValue xmlns:dsig="urn:cbr-ru:dsig:v1.1">
RpxoZ6vnUXn9/nTSC9rkqeWt1NYTc+RxWZ5JbdFW6Vlg+ULhx7uDJFPRIIdqxXJnIugF2xz1pgjCtmh
4hz9tLAg==</dsig:MACValue>
```

3.2.3. Profile of the parameters of protection of an EM (EM packet) with an AC

The protection of an EM (EM packet) with an AC applies pursuant to the profile of parameters of protection of an EM (EM packet) with an AC (see table 10). The profile of protection of an EM (EM packet) with ACs contains the list of specifications and algorithms that apply for converting an EM (EM packet) to a form ensuring its protection with the cryptographic information security system through the placement and verification of ACs.

T a b l e 10. Profile of the parameters of protecting an EM (EM packet) with an AC

Algorithm	Identifier
EM (EM packet) transformations	
Transformation of an EM (EM packet) for conversion to a normalised form	not used
XML canonicalisation without comments [XML-c14n]	not used
EM (EM packet) encoding	
Compression algorithm (optional)	http://www.ietf.org/rfc/rfc1951
Encryption algorithm (optional)	to be determined by the Exchange Contract
Encoding algorithm Base64	http://www.ietf.org/rfc/rfc2045#base64
AC value encoding	
Encoding algorithm Base64	http://www.ietf.org/rfc/rfc2045#base64

The cryptographic security of files with EMs shall be ensured by using a CISS having a certificate or a temporary permit of the Federal Security Service or a temporary permit of the Bank of Russia.

3.2.4. Reference to data to be signed

An AC always protects the contents of the element **sen:Object**. The contents of the component **sen:Object** are an XML document containing the EM (EM packet) to be signed, encoded pursuant to the algorithm [base64]. The XML document containing the EM (EM packet) to be signed shall be generated with account for the requirements for the execution of XML documents (see Chapter 10).

An illustration showing the data to be signed when generating an AC is provided below (see fig. 6).

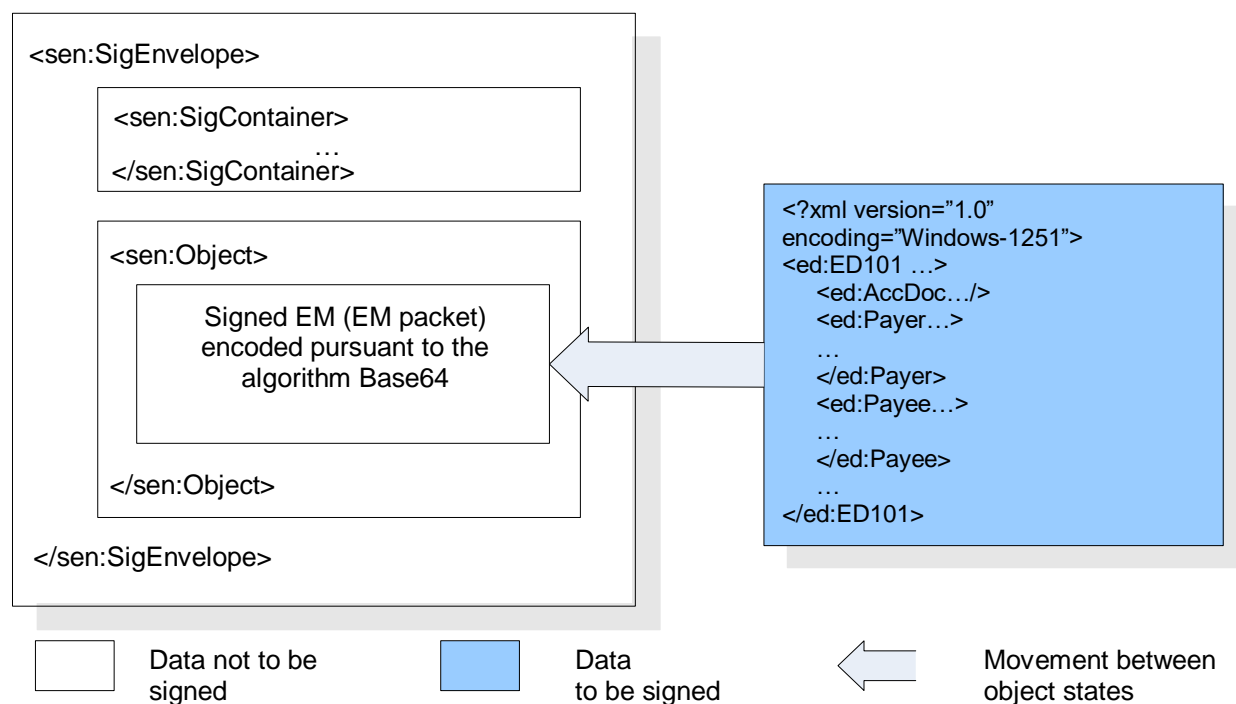


Fig. 6. AC execution illustration showing data to be signed

The EM (EM packet) to be signed can contain SCs as elements. In this case, the AC still protects the total XML document containing the EM (EM packet) along with all elements (including all SCs). An illustration showing the data to be signed when generating an AC is provided below (see fig. 7).

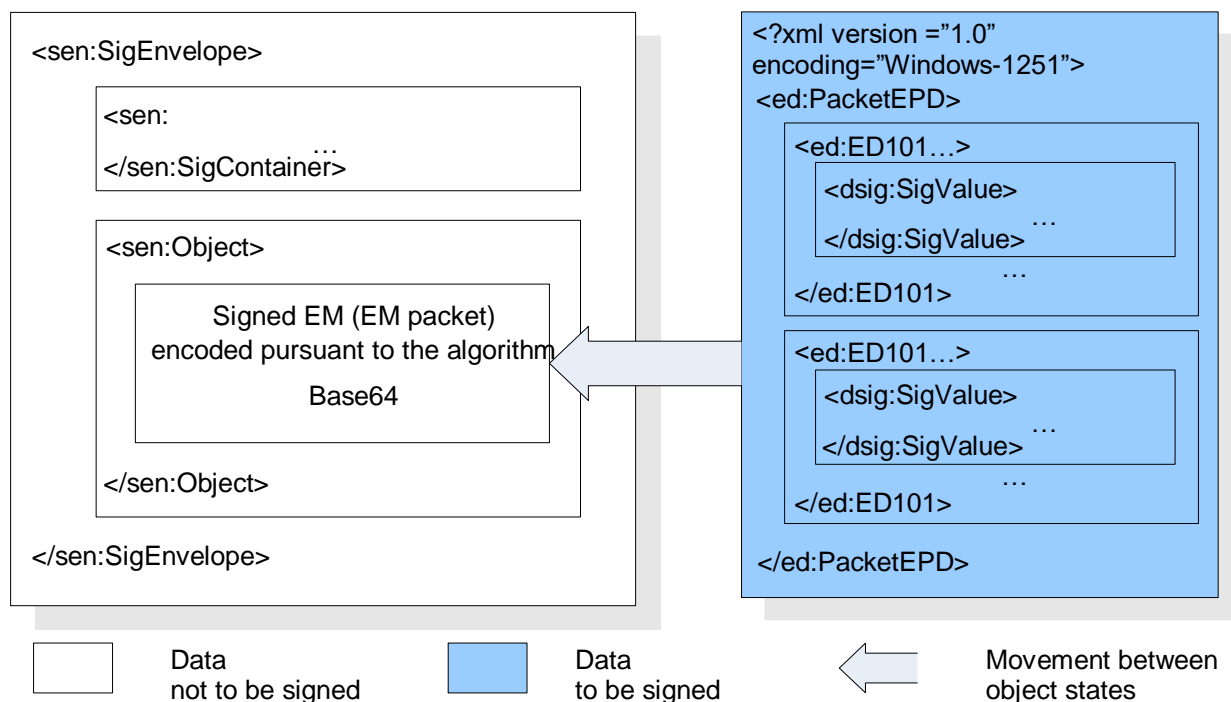


Fig. 7. Illustration of the execution of an AC for an EM packet with protection of each EM with its own SC

3.3. Rules for generating and verifying an AC

This section describes the procedure for transformations of an EM (EM packet) necessary for generating and verifying AC values.

The rules for executing AC values and defining the set of data to be signed inside an EM (EM packet) are provided in the description of EM formats.

3.3.1. Rules for generating an AC

The process of generating an AC envelope consists of the following phases:

- a) Generating an XML document containing the EM (EM packet) to be protected with an AC. The XML document must be generated with account for the requirements for the execution of XML documents pursuant to Subsection 10.1 of the Album of Formats.
- b) Serialising (pursuant to [XML]) the XML document formed at the previous phase in the byte array for which AC will be calculated.
- c) Generating the AC (calculating its value): Calling the CISS function for generating an AC with transfer of the byte array obtained at the previous phase.
- d) Compressing the data array obtained at phase b), if stipulated by the Exchange Contract.
- e) Encrypting the data array obtained at phase b), with account for possible compression at phase d), if stipulated by the Exchange Contract.
- f) Encoding the SC value obtained at phase c) (in the format of an AC library, without isolation of the AC value) pursuant to the algorithm [base64].
- g) Placing the AC value encoded at the previous phase in the component sig:MACValue.
- h) Coding the byte array obtained at phase b), with account for possible compression at phase d) and/or possible encryption at phase e) pursuant to the algorithm [base64].
- i) Placing the byte array encoded at the previous phase in the component sen:Object.
- j) Executing the AC envelope pursuant to Clause 3.2.2.

3.3.2. Rules for verifying an AC

The process for verifying an AC on a XML document consists of the following phases:

- a) Receiving an XML document containing an EM (EM packet) protected with an AC.
- b) Isolating the AC value from the component sig:MACValue.
- c) Decoding the AC value isolated at the previous phase pursuant to the algorithm [base64].
- d) Isolating the EM (EM packet) protected with the AC from the component sen:Object.
- e) Decoding the EM (EM packet) isolated at the previous phase pursuant to the algorithm [base64].
- f) Decrypting the byte array obtained at the previous phase, if stipulated by the Exchange Contract.
- g) Decompressing the data array obtained at the previous phase, if stipulated by the Exchange Contract.
- h) Verifying the AC: Calling the CISS function for verifying an AC with transfer of the byte array obtained at phases c) and e), with account for the possible decryption at phase f) and/or the possible decompression at phase g).

3.4. Encryption

Messages are encrypted and decrypted with CISS tools having a certificate or a temporary permit of the Federal Security Service or a temporary permit of the Bank of Russia. The exchange of and access to information necessary for encryption and decryption is determined by the Exchange Contract.

3.5. Compression

Data is compressed and decompressed using the algorithm [deflate]. The use of data compression is determined by the Exchange Contract.

The party accepting a compressed message shall ensure the complete support of formats [deflate] and [zlib].

3.5.1. Structure of the format of compressed data forming part of the component sen:Object

The compressed data format is a sequence of two blocks, the first of which consists of four bytes and contains the data length before compression, and the second block in the format [zlib] contains the data compressed pursuant to the algorithm [deflate] (see table 11).

Table 11. Structure of the format of compressed data

Block name	Block description	Block length (bytes)
Data size before compression	4-byte modular integer (32 bytes) in little-endian format (the least significant byte is first).	4
Compressed data block	Data block in the format [zlib] compressed pursuant to the algorithm [deflate].	9-n