Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: February 5, 2015

S. Leontiev, Ed.
D. Pichulin, Ed.
CRYPTO-PRO
A. Fedchenko, Ed.
S-Terra
August 4, 2014

Using GOST 28147-89 with IPsec Encapsulating Security Payload (ESP)
draft-fedchenko-ipsecme-cpesp-gost-04

Abstract

This document defines the usage of GOST 28147-89 algorithm when providing data integrity and confidentiality in ESP protocol.

The contents of this document is technically equivalent to its TC26 ROSSTANDART specification.

This specification is maintained by TC26 ROSSTANDART and further updates are available at: http://www.tc26.ru/.

Table of Contents

1. Introduction

This document contains a technical specification approved by the Technical Committee #26 ("Cryptography and security mechanisms") of Federal Agency on Technical Regulating and Metrology of the Russian Federation (ROSSTANDART) [TC26ESP].

This memo describes implementation features and additional identification types of ESP protocol [RFC4303] when used with GOST 28147-89 encryption algorithm. This document defines the following payload transforms in ESP protocol:

- combined mode transform ESP_GOST-4M-IMIT;

- combined mode transform ESP_GOST-1K-IMIT.

This memo does not define GOST 28147-89 cryptographic algorithm and formats of a cryptographic data representation. The algorithm itself is defined in GOST 28147-89 national standard [GOST28147] [RFC5830], the data and parameters representation corresponds [RFC4357], [RFC4491], [RFC4490] and [TC26IKE].

The development objective of this document is to provide interoperability of IPsec protocol implementations, produced by Russian vendors.

2. Terms and Definitions

This document operates with terms and definitions from IPsec [RFC4301] and ESP [RFC4303] standards, only additional definitions are described below.

2.1. Requirements Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.2. Notation

encryptCNT(IV, K, D): GOST 28147-89 encryption of data D in CNT mode with key K and initialization vector IV;

decryptCNT(IV, K, D): GOST 28147-89 decryption of data D in CNT mode with key K and initialization vector IV;

Divers(K, D): diversification of key K by diversification data D (Section 7 of [RFC4357]), S-box identifiers are defined in Section 5.1, in this document diversification data is a sequence of 8 bytes, interpreted as a 64-bit unsigned integer in network byte order;

gost28147IMIT(IV, K, D): generation of GOST 28147-89 MAC on data D with key K, initialization vector IV and inner zero padding to 8-byte boundary;

A: associated data (AEAD in Section 2.1 of [RFC5116], according to GOST MAY contain address, timestamp, IV et al.);

Seq#: 64-bit packet number (if ESN [RFC4304] is not negotiated, Seq# value always belongs to the range 1..2^32-1);

Seq#h: upper portion of Seq#;

Seq#l: lower portion of Seq#;

IV(Seq#): initialization vector of Seq#-th packet;

Kc_e(Seq#): combined mode encryption algorithm key of Seq#-th packet;

Kc_i(Seq#): combined mode MAC algorithm key of Seq#-th packet;

Kc_i2(Seq#): preliminary packet check key of Seq#-th packet and SA (only for ESP_GOST-1K-IMIT);

Kr_e: base encryption key of SA;

Kr_i: base MAC key of SA;

KeyMeshing: key meshing algorithm as described in [RFC4357];

SPI-Auth-Code: authentication code computed in the ISAKMP SA context (of IKE protocol or another key negotiation protocol) for given IPsec SA and intended for auditing and preliminary check of packet;

substr(s..f, bytes): bytes from byte s to byte f, coherently chosen from the sequence of "bytes" in network byte order.

3. Establishing of ESP Security Association (SA)

ISAKMP protocol provides mechanisms of security attributes negotiation. The basic specification of ISAKMP protocol is defined in [RFC2408].

In the framework of ISAKMP SA (IKE or another key negotiation protocol) for the given IPsec SA at least the following components are negotiated:

- 32-bit SPI authentication code (non-confidential);
- 256-bit symmetric key Kr_e;
- 256-bit symmetric key Kr_i (only for ESP_GOST-1K-IMIT);
- GOST 28147-89 parameters (S-box set);
- SA lifetime in kilobytes (SA Lifetime, Kbytes);
- SA lifetime in seconds (SA Lifetime, seconds);
- maximum number of invalid packets.

Depending on the transform type, the amount of the negotiated keying material (KEYMAT) is:

For ESP_GOST-4M-IMIT: 36 bytes (Kr_e and SPI-Auth-Code);

For ESP_GOST-1K-IMIT: 68 bytes (Kr_e, Kr_i and SPI-Auth-Code).

4. Encapsulating Security Payloads

ESP packet payload MUST meet the requirements to combined mode payload transform defined in Section 2 of [RFC4303]. Encapsulating security payloads with GOST 28147-89 encryption algorithm MUST comply with the following requirements:

- Initialization Vector (IV) of 8 bytes is sent in the packet;
- ESP payload is padded to align on 8-byte boundary;
- in case of ESN usage, Seq#h value is not included in the transmitted packet;
- ICV is not explicitly padded;

- ICV of 4 bytes (for ESP_GOST-4M-IMIT transform) or 8 bytes (for ESP_GOST-1K-IMIT transform) is transmitted in the packet.

For being negotiated IPsec SAs it is RECOMMENDED to:

- turn on an anti-replay service;

- limit the total number of ESP packets and the total size of the ESP payloads with same Seq# by applying additional organizational and technical measures in case of an anti-replay service is not used or partially applied.

4.1. Using GOST 28147-89 for ESP Payloads

Associated data involved in the ESP MAC generation MUST contain the following ancillary information:

A = SPI | Seq#l | IV

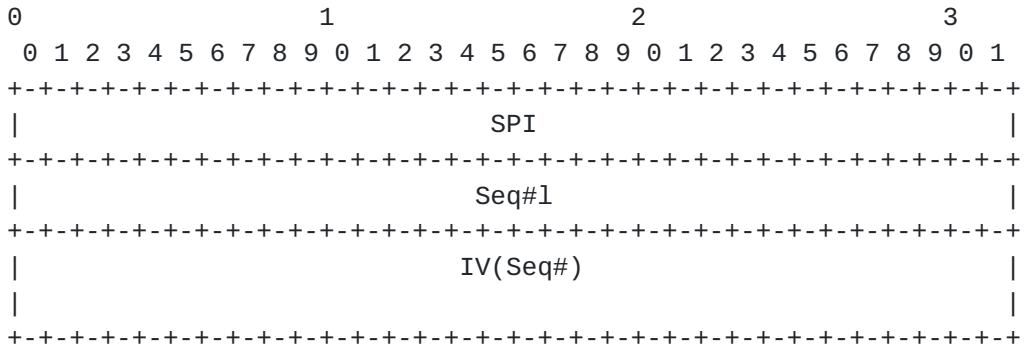

Figure 1: Associated data (A) for ESP

The transforms use vector IV(Seq#), the format of this vector is shown below:

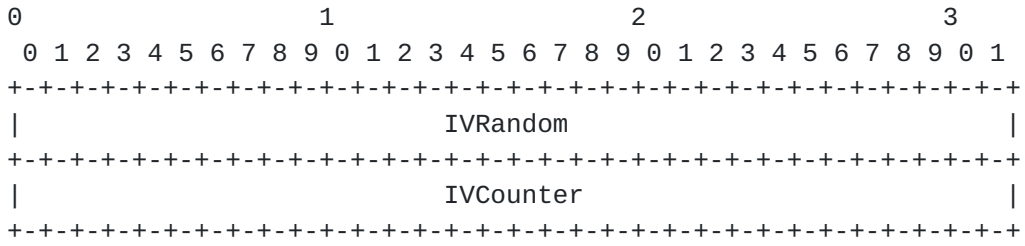

Figure 2: IV for ESP_GOST-4M-IMIT and ESP_GOST-1K-IMIT

In this case:

IVRandom - 4 random bytes

IVCounter = (SPI-Auth-Code + SPI + Seq#l + IVRandom) mod 2^32

A receiver MUST control IVCounter value on the Seq# preliminary check phase and MUST NOT perform any cryptographic operations in case of failure in preliminary check.

If an ESP implementation supports audit logging, this failure MAY be classified as an error in Seq# checking. Particularly, packets with an incorrect IVCounter MUST NOT cause changes in the counter of invalid packets and the counter of SA lifetime in kilobytes.

4.2. Outbound Packet Processing

Outbound packet processing MUST comply with the requirements specified in Section 3.3 of [RFC4303] with the following modifications:

- in addition to the checks specified in Section 3.3.1 of [RFC4303], it is RECOMMENDED to check the ESP payload length in accordance with SA parameters;
- MAC in the transforms is calculated as follows:

  substr(0..3, ICV) = gost28147IMIT(0, Kc_i(Seq#), A | Payload Data | Padding | Pad Length | Next Header [| Seq#h])

- encryption in ESP_GOST-4M-IMIT transform is performed without a key meshing;
- encryption in ESP_GOST-1K-IMIT transform is performed with the key meshing mode id-Gost28147-89-CryptoPro-KeyMeshing;
- encryption in the transforms is performed by the formula:

  encryptCNT(IV(Seq#), Kc_e(Seq#), Payload Data | Padding | Pad Length | Next Header)

- additional MAC in ESP_GOST-1K-IMIT transform is calculated as follows:

  substr(4..7, ICV) = gost28147IMIT(0, Kc_i2(Seq#), A | Encrypted Payload [| Seq#h] | substr(0..3, ICV))

- the sender is RECOMMENDED to increase the counter of SA lifetime in kilobytes and compare this value with the maximum of SA lifetime (in kilobytes) for this SA. In case of exceeding the maximum it is RECOMMENDED to disable this SA.

4.3. Inbound Packet Processing

Inbound packet processing MUST comply with the requirements specified in Section 3.4 of [RFC4303] with the following modifications:

- in addition to the checks specified in Section 3.4.2 of [RFC4303], it is RECOMMENDED to check the ESP payload length in accordance with SA parameters;

- during the preliminary check phase for ESP_GOST-4M-IMIT and ESP_GOST-1K-IMIT transforms it is RECOMMENDED to validate IV(Seq#) of the packet (Section 4.1 of this document);

- during the preliminary check phase for ESP_GOST-1K-IMIT transform ICVchk2 MUST be generated, if substr(4..7, ICV) does not match with ICVchk2, a receiver MUST abort the packet processing, and MUST NOT change the state of the corresponding SA, and MAY not audit such events, and it is NOT RECOMMENDED to audit such events without specific requirements. ICVchk2 is calculated as follows:

  ICVchk2 = gost28147IMIT(0, Kc_i2(Seq#), A | Encrypted Payload [| Seq#h] | substr(0..3, ICV))

- the receiver is RECOMMENDED to increase the counter of SA lifetime in kilobytes and compare them with the maximum value of SA lifetime (in kilobytes). In case of exceeding the maximum it is RECOMMENDED to disable this SA;

- encryption in ESP_GOST-1K-IMIT transform performed with the key meshing mode id-Gost28147-89-CryptoPro-KeyMeshing;

- decryption in ESP_GOST-4M-IMIT transform performed without a key meshing;

- decryption in ESP_GOST-1K-IMIT transform performed with the key meshing mode id-Gost28147-89-CryptoPro-KeyMeshing;

- decryption in the transforms performed by the formula:

  decryptCNT(IV(Seq#), Kc_e(Seq#), Encrypted Payload)

- during the MAC check phase for the transforms ICVchk MUST be generated, if substr(0..3, ICV) does not match with ICVchk, the receiver is RECOMMENDED to increase the counter of invalid packets of the SA and compare this value with the maximum number of invalid packets for this SA. In case of exceeding the maximum it is RECOMMENDED to disable this SA. ICVchk is calculated as follows:

```
ICVchk = gost28147IMIT(0, Kc_i(Seq#), A | Payload Data |
Padding | Pad Length | Next Header [ | Seq#h ])
```

4.4. MTU Calculation

In determining the MTU in the case of using ESP_GOST-4M-IMIT or ESP_GOST-1K-IMIT transform should be guided by the rules specified in Section 2 of [RFC4303], with the addition of 12 bytes for ESP GOST-4M-IMIT (8 bytes IV plus 4 bytes ICV) or 16 bytes for ESP_GOST-1K-IMIT (8 bytes IV and 8 bytes ICV), rounding the result to the higher multiple of 8.

4.5. ESP_GOST-4M-IMIT Transform

Parameter values for ESP_GOST-4M-IMIT transform are shown below:

```
KeyMeshing = id-Gost28147-89-None-KeyMeshing

Kc_e(Seq#) = Divers(Divers(Divers(Kr_e, Seq#&0xffffffff00000000),
                 Seq#&0xffffffffffff0000),
                 Seq#&0xfffffffffffffc0)

Kc_i(Seq#) = Kc_e(Seq#)
```

4.6. ESP_GOST-1K-IMIT Transform

Parameter values for ESP_GOST-1K-IMIT transform are shown below:

```
KeyMeshing = id-Gost28147-89-CryptoPro-KeyMeshing;

Kc_e(Seq#) = Divers(Divers(Divers(Kr_e, Seq#&0xffffffff00000000),
                 Seq#&0xffffffffffff0000),
                 Seq#)

Kc_i(Seq#) = Kc_e(Seq#)

Kc_i2(Seq#) = Divers(Divers(Divers(Kr_i, Seq#&0xffffffff00000000),
                 Seq#&0xffffffffffff0000),
                 Seq#)
```

5. Additional ESP SA Parameters and Attributes

To match the attributes of transform using the protocol IKE [RFC2409] both sides MUST negotiate the application ID, ESP_GOST_Vendor_ID. The format of this ID is shown below:

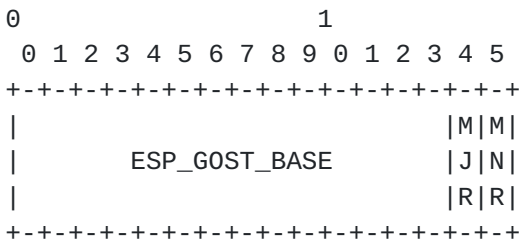

Figure 3: ESP_GOST_Vendor_ID

In this case ESP_GOST_BASE = { '\x03', '\x10', '\x17', '\xE0', '\x7F', '\x7A', '\x82', '\xE3', '\xAA', '\x69', '\x50', '\xC9', '\x99', '\x99' } (first 14 bytes of GOST R 34.11-94 hash function for "IKE/GOST" string). MJR and MNR bytes represent the values of the major and minor parts of version number for ESP_GOST transform, respectively (i.e. MJR = 1, MNR = 1).

Parameter	Attribute Value	Format	Default Value
GOST 28147-89 Parameters	32401	B	-
Maximum packet size	32403	V	65536
Maximum number of invalid packets (SA Life Type)	64402	-	10^5

Table 1: ESP_GOST SA Parameters

5.1. GOST 28147-89 Parameters

GOST-28147-89-SBOX	Attribute Value
id-Gost28147-89-CryptoPro-A-ParamSet	65403
id-Gost28147-89-CryptoPro-B-ParamSet	65404
id-Gost28147-89-CryptoPro-C-ParamSet	65405
id-Gost28147-89-CryptoPro-D-ParamSet	65406
id-tc26-gost-28147-param-Z	65407

Table 2: GOST 28147-89 Parameters

The id-tc26-gost-28147-param-Z parameter set is RECOMMENDED for Internet usage by TC26 ROSSTANDART ([TC26UZ] and [rus-popov-esp-sbox-00-rb]).

5.2. Maximum Value of Invalid Packets Counter

In the case of SA-Life-Type = Max-Integrity-Fails when counter of invalid packets is reached SA-Life-Duration, it is RECOMMENDED to block packet processing for this SA and RECOMMENDED to start deletion procedure for this SA.

5.3. Maximum Packet Length

Attribute class: Max-Packet-Len (32403), attribute format: variable length (V)

In case of using ESP protocol with IPv6 Jumbograms [RFC2675] (sizes from 64 KB to 4 GB), it is RECOMMENDED to negotiate the maximum packet length parameter.

6. IKEv2 Encrypted Payloads

Integrity verification as a whole meets the requirements described in Section 5 of [RFC5282] and [RFC5116] which extends the basic integrity control method [RFC5996] for case of combined mode algorithms.

Key material derivation for ESP transforms performed in accordance with the Section 2.14 of [RFC5996], but integrity control keys of IKEv2 (SK_ai and SK_ar) are not used and their size MUST be treated as 0 octets.

The size of the key material for SK_ei and SK_er keys complies with

the size defined in Section 2.14 of [RFC5996], specifically, 36 or 68 bytes for each key when using ESP_GOST-4M-IMIT or ESP_GOST-1K-IMIT respectively. The required key material is derived iteratively by applying PRF function without alignment to the size of the hash function value.

6.1. Using GOST 28147-89 for IKEv2 Payloads

Associated data involved in the IKEv2 MAC generation MUST contain the following ancillary information:

A = IKEv2-Header | Unencrypted IKE Payloads | Payload Header | IV(Seq#)

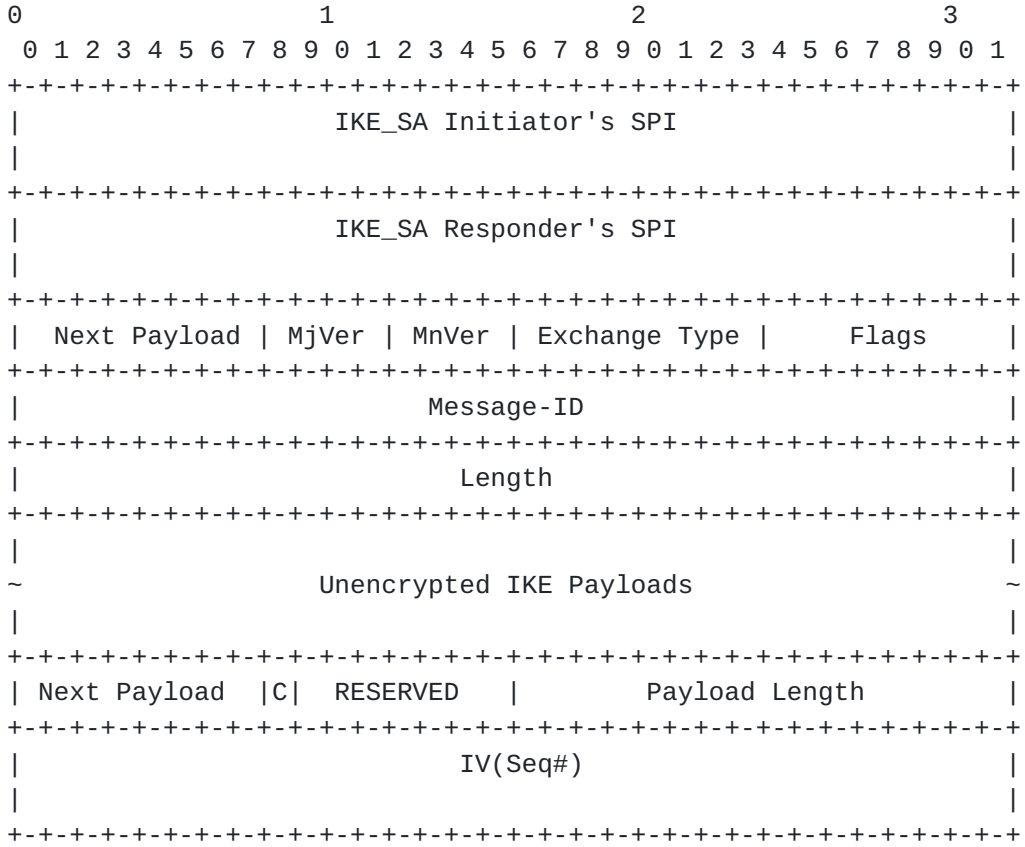

Figure 4: Associated data (A) for IKEv2 payloads

The transforms use vector IV(Seq#), the format and recommendations for which are described in Section 4.1 of this document with the following clarifications:

Seq# = Message-ID*2 + R-flag

SPI = SPIi#h + SPIi#l (for packets from Initiator)

SPI = SPIr#h + SPIr#l (for packets from Responder)

In the case of applying the requirements of [RFC6311] it is RECOMMENDED either to avoid IKEV2_MESSAGE_ID_SYNC_SUPPORTED negotiation, or to limit the number of packets with Message-ID = 0 (Section 12 of [RFC6311]) and total size of that packets, starting from second packet.

7. Acknowledgements

The authors express their gratitude to Alex S. Kuzmin, a chairman of Rosstandart subcommittee on cryptography (TC26), who initiated a creation of IPsec Working Group, and not only infuse the initial impetus to the development of national standardized IPsec solution based on GOST 28147-89 algorithm but also constantly supported this work methodologically and practically.

The authors would like to thank Russian representatives of CISCO and CheckPoint, as well as Gazprom, company, which initiates a process to ensure compatibility of IPsec products from different vendors.

The authors express special thanks to Dmitry N. Zakharov (LLC "Factor-TS") for protocol implementation verification, Valery A. Smyslov (JSC "ELVIS-PLUS") and Andrew L. Chmora (JSC "InfoTeCS") for number of valuable suggestions and improvements in the protocol itself and its description, Viktor M. Timakov (JSC "Signal-COM") for productive discussions on the cryptographically strong GOST 28147-89 MAC generation mode.

TC26 Author's Addresses

Vladimir O. Popov
"CRYPTO-PRO", LLC
18, Suschevsky Val str.
Moscow, 127018, Russian Federation
Phone: +74957804820
EMail: vpopov@cryptopro.ru

Mikhail M. Gruntovich
"OKB SAPR", PJSC
8, 2nd Kozhevnichesky lane
Moscow, 115114, Russian Federation

Phone: +74952356265
EMail: gmm@accord.ru

Dmitry M. Avramenko
"R-alfa", LLC
4/1, Raspletina str.
Moscow, 123060, Russian Federation
EMail: info@alpha.ru

Mark Y. Koshelev
"ELVIS-PLUS", PJSC
Zelenograd, 5/23, driveway 4806
Moscow, 124498, Russian Federation
Phone: +74952760211
EMail: info@elvis.ru

George O. Martanov
Federal State Unital Enterprise "Scientific and Technical Centre Atlas"
38, Obraztsova str.
Moscow, 127018, Russian Federation
Phone: +74956892352
EMail: atlas@stcnet.ru

Yury Y. Sidorin
Federal State Unital Enterprise "Scientific and Technical Centre Atlas"
38, Obraztsova str.
Moscow, 127018, Russian Federation
Phone: +74956892352
EMail: atlas@stcnet.ru

Valery A. Smyslov
"ELVIS-PLUS", PJSC
Zelenograd, 5/23, driveway 4806
Moscow, 124498, Russian Federation
Phone: +74952760211
EMail: info@elvis.ru

Stanislav V. Smyshlyaev
"CRYPTO-PRO", LLC
18, Suschevsky Val str.
Moscow, 127018, Russian Federation
Phone: +74957804820

EMail: svs@cryptopro.ru

Viktor M. Timakov
"Signal-COM", PJSC
19, Usievicha str.
Moscow, 125315, Russian Federation
Phone: +74956632365
EMail: v.timakov@signal.ru

Andrey V. Fedotov
"Factor-TS", LLC
11A, 1st Magistralny lane
Moscow, 123290, Russian Federation
Phone: +74956443130
EMail: fedotov@factor-ts.ru

Artem V. Chuprina
"Nryptokom", LLC
1, Pionerskaya str., office 001
Vladivostok, 690001, Russian Federation
Phone: +74232605201
EMail: kript225@gmail.com

8. IANA Consideration

IANA has assigned two numbers for ESP transforms:

<TBD-3> for ESP_GOST-4M-IMIT;

<TBD-4> for ESP_GOST-1K-IMIT.

8.1. Remove After IANA Consideration

Currently, preliminary implementations are using the following private numbers for transforms:

253 for ESP_GOST-4M-IMIT;

252 for ESP_GOST-1K-IMIT.

8.2. Not Subject for IANA Consideration

Private "magic numbers" used in this document:

Class	Value	Type	Reference
GOST-28147-89-SBOX	32401	B	Section 5.1
Max-Packet-Len	32403	B	Section 5.3

Table 3: ESP_GOST "magic numbers"

Private values used in this document:

Name	Value	Attribute
Max-Integrity-Fails	64402	SA-Life-Type [RFC2407]
id-Gost28147-89-CryptoPro-A-ParamSet	65403	GOST-28147-89-SBOX Section 5.1
id-Gost28147-89-CryptoPro-B-ParamSet	65404	GOST-28147-89-SBOX Section 5.1
id-Gost28147-89-CryptoPro-C-ParamSet	65405	GOST-28147-89-SBOX Section 5.1
id-Gost28147-89-CryptoPro-D-ParamSet	65406	GOST-28147-89-SBOX Section 5.1
id-tc26-gost-28147-param-Z	65407	GOST-28147-89-SBOX Section 5.1

Table 4: ESP_GOST private values

9. Security Considerations

Implementations are RECOMMENDED to examine for compliance with requirements specified by the Decree of the Government of the Russian Federation (#957). The parameters of cryptographic algorithm affect the strength of the encryption. The use of parameters not described in [RFC4357] are NOT RECOMMENDED for implementation, unless tested according to Section 9 of [RFC4357].

GOST 28147-89 block length is 64-bit, so to ensure the confidentiality and integrity of data IPsec implementations are RECOMMENDED to use the following limitations:

- For ESP_GOST-4M-IMIT transform it is RECOMMENDED to process packets that do not exceed the size of 64 Kbytes. For IPv6 packets it is NOT RECOMMENDED to negotiate Max-Packet-Len value more than 64 Kbytes and it is NOT RECOMMENDED to use IPv6 Jumbograms ([RFC2675]) without appropriate research. It is RECOMMENDED to renegotiate keys for a new ESP SA after the transfer of maximum amount of data for SA (SA Lifetime, Kbytes) - 2^80 bytes;

- For ESP_GOST-1K-IMIT transform it is RECOMMENDED to renegotiate keys for a new ESP SA after the transfer of maximum amount of data for SA (SA Lifetime, Kbytes) - 2^80 bytes;

- For implementations it is RECOMMENDED to negotiate SA lifetime in kilobytes and seconds, Section 4.4.2.1 of [RFC4301];

- It is NOT RECOMMENDED to negotiate SA lifetime in seconds more than 86400 seconds (24 hours);

- It is NOT RECOMMENDED to negotiate Max-Integrity-Fails value more than 10^5 without appropriate research.

10. Examples

The examples below use default settings. Encryption with id-Gost28147-89-CryptoPro-B-ParamSet.

10.1. ESP_GOST-4M-IMIT Example

ESP_GOST-4M-IMIT plaintext (length - 53. bytes):

```
4500001d 04050607 08090a0b 0c0d0e0f 10111213 14151617 18191a1b 1c1d1e1f
20212223 24252627 28292a2b 2c2d2e2f 30313233 34
```

ESP_GOST-4M-IMIT parameters:

```
SPI
 31323334
Seq#l
 0000007d
ESN
 not negotiated
Padding | Pad Length | Next Header
 000104

SPI-Auth-Code
 cb4e1a7f
Kr_e
 b63d156f 7aac0dc7 cd915c35 63f61b9d 5c730a74 e331bc8c 3fc24a36 06463893
Kr_e2 = Divers(Kr_e, Seq# & 0xffffffff00000000)
 3a742e54 a9e8a3a1 1f4af5f7 c92fdac1 55142bee 86766e8c 03fad68d 35baf3d7
Kr_e1 = Divers(Kr_e2, Seq# & 0xffffffffffff0000)
 752bdd27 99dcde7b 92c04591 40fac2cb 974f39dc cf589d45 00a67c75 99ff9fcc
Kc_e = Divers(Kr_e1, Seq# & 0xfffffffffffffc0)
 0772fe26 c770590f 22902ad2 1a919eee ccab5396 baf2f1b5 54366c30 27a38614
```

ESP_GOST-4M-IMIT encrypted data (length - 76. bytes (== 8.+8.+53.+3.+4.)):

```
 31323334 0000007d 05060708 01865538 fa104495 3cd50f2b cca22b90 f4f36257
 8f4c2435 f04ada62 d0abc8b3 099e0473 d1ccd142 1586d564 a5e3d1c2 34e529ff
 fe652e24 caad891c 0bd8ba08
```

10.2. ESP_GOST-1K-IMIT Example

ESP_GOST-1K-IMIT plaintext (length - 1049. bytes):

```
4500001d 04050607 08090a0b 0c0d0e0f 10111213 14151617 18191a1b 1c1d1e1f
20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637 38393a3b 3c3d3e3f
40414243 44454647 48494a4b 4c4d4e4f 50515253 54555657 58595a5b 5c5d5e5f
60616263 64656667 68696a6b 6c6d6e6f 70717273 74757677 78797a7b 7c7d7e7f
80818283 84858687 88898a8b 8c8d8e8f 90919293 94959697 98999a9b 9c9d9e9f
a0a1a2a3 a4a5a6a7 a8a9aaab acadaeaf b0b1b2b3 b4b5b6b7 b8b9babb bcbdbebf
c0c1c2c3 c4c5c6c7 c8c9cacb cccdcecf d0d1d2d3 d4d5d6d7 d8d9dadb dcdddedf
e0e1e2e3 e4e5e6e7 e8e9eaeb ecedeeef f0f1f2f3 f4f5f6f7 f8f9fafb fcfdfeff
00010203 04050607 08090a0b 0c0d0e0f 10111213 14151617 18191a1b 1c1d1e1f
20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637 38393a3b 3c3d3e3f
40414243 44454647 48494a4b 4c4d4e4f 50515253 54555657 58595a5b 5c5d5e5f
60616263 64656667 68696a6b 6c6d6e6f 70717273 74757677 78797a7b 7c7d7e7f
80818283 84858687 88898a8b 8c8d8e8f 90919293 94959697 98999a9b 9c9d9e9f
a0a1a2a3 a4a5a6a7 a8a9aaab acadaeaf b0b1b2b3 b4b5b6b7 b8b9babb bcbdbebf
c0c1c2c3 c4c5c6c7 c8c9cacb cccdcecf d0d1d2d3 d4d5d6d7 d8d9dadb dcdddedf
e0e1e2e3 e4e5e6e7 e8e9eaeb ecedeeef f0f1f2f3 f4f5f6f7 f8f9fafb fcfdfeff
00010203 04050607 08090a0b 0c0d0e0f 10111213 14151617 18191a1b 1c1d1e1f
20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637 38393a3b 3c3d3e3f
40414243 44454647 48494a4b 4c4d4e4f 50515253 54555657 58595a5b 5c5d5e5f
60616263 64656667 68696a6b 6c6d6e6f 70717273 74757677 78797a7b 7c7d7e7f
80818283 84858687 88898a8b 8c8d8e8f 90919293 94959697 98999a9b 9c9d9e9f
a0a1a2a3 a4a5a6a7 a8a9aaab acadaeaf b0b1b2b3 b4b5b6b7 b8b9babb bcbdbebf
c0c1c2c3 c4c5c6c7 c8c9cacb cccdcecf d0d1d2d3 d4d5d6d7 d8d9dadb dcdddedf
e0e1e2e3 e4e5e6e7 e8e9eaeb ecedeeef f0f1f2f3 f4f5f6f7 f8f9fafb fcfdfeff
00010203 04050607 08090a0b 0c0d0e0f 10111213 14151617 18191a1b 1c1d1e1f
20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637 38393a3b 3c3d3e3f
40414243 44454647 48494a4b 4c4d4e4f 50515253 54555657 58595a5b 5c5d5e5f
60616263 64656667 68696a6b 6c6d6e6f 70717273 74757677 78797a7b 7c7d7e7f
80818283 84858687 88898a8b 8c8d8e8f 90919293 94959697 98999a9b 9c9d9e9f
a0a1a2a3 a4a5a6a7 a8a9aaab acadaeaf b0b1b2b3 b4b5b6b7 b8b9babb bcbdbebf
c0c1c2c3 c4c5c6c7 c8c9cacb cccdcecf d0d1d2d3 d4d5d6d7 d8d9dadb dcdddedf
e0e1e2e3 e4e5e6e7 e8e9eaeb ecedeeef f0f1f2f3 f4f5f6f7 f8f9fafb fcfdfeff
00010203 04050607 08090a0b 0c0d0e0f 10111213 14151617 18
```

ESP_GOST-1K-IMIT parameters:

```
SPI
 31323334
Seq#l
 0000007d
ESN
 negotiated
Seq#h
 0000000b
Padding | Pad Length | Next Header
 00000000 000504

SPI-Auth-Code
 c4c08a66
Kr_e
 b63d156f 7aac0dc7 cd915c35 63f61b9d 5c730a74 e331bc8c 3fc24a36 06463893
Kr_e2 = Divers(Kr_e, Seq# & 0xffffffff00000000)
 59f1548e a0b38639 c546fb94 2164d780 f075460a cb72e4cf f3068a03 a184d544
Kr_e1 = Divers(Kr_e2, Seq# & 0xffffffffffff0000)
 c210c1fc 6988bb00 6ed69111 9d63a619 6c4cee21 799786db 2af3bda9 c77f9e20
Kc_i = Kc_e = Divers(Kr_e1, Seq#)
 fdcd7812 74018a14 5aee2da7 f21a1581 148378b9 272e3d88 0585ac2e 94b4bbe2
Kr_i
 cb4e1a7f 2d61710d f264423c ad4384de ce01d676 90556865 f1cb7f7f ab4103c0
Kr_i2 = Divers(Kr_i, Seq# & 0xffffffff00000000)
 52aa9784 2ee74f7a 5c3c7436 9fe4f415 5a4bc218 05fc6263 2a1ef408 4d8de3a2
Kr_i1 = Divers(Kr_i2, Seq# & 0xffffffffffff0000)
 7ea6c8f3 209ac480 f181aa61 5c38d07b fd680717 16581ff9 c2963646 6094cc3a
Kc_i2 = Divers(Kr_i1, Seq#)
 138309a0 5813c2bf d3bfb2a9 aff9b511 555c2088 ababcac7 f21f0871 1036aff8
```

ESP_GOST-1K-IMIT encrypted data (length - 1080. bytes (== 8.+8.+ 1049.+7.+4.+4.)):

```
31323334 0000007d 05060708 faf8c51f 85bc7a31 676f1453 c167d042 1e87a0d1
a23cbb97 cefbd7fe e95c3d1a b9f3fa9e 144dc92a 97e6de75 5b1fda97 8436c90b
289c222f 80de286b bdf190b3 be7f6abb f627c56d 25ecc471 9bc9a5f3 403f1852
67b43b70 a860b606 625ab17b 839078dd a55c9e76 78388625 27f17ef8 da3c964f
0e320c68 6e71c9bb e17381d0 321fdfb5 8092f205 2df6d199 90ec758d 31eb6eeb
3e152d43 89d4d402 9ab77fb4 09fcf757 b4086948 cf9d8843 5a2b21f2 6c41aa5f
6b881623 be08b64e 4aeaba95 706598f3 5d56ee18 0feafff4 32b35a54 b37a7c77
6e408d09 65aa2aa4 41f69040 03cec18a e1529416 88afe127 1c0fd90b c9699020
9a98527f e8d4a285 de2f1d09 3b0f6779 a2391f71 d12d1219 c98b74ce 30b35b04
381896c5 205ca00b ed4df571 d24ecfd3 7134b910 7c8eeb2d 6e3dedf3 ffab4af1
1e69b296 fb4a9d0d 3dc364e0 4f0c6ab9 925322e4 0a8ff71e 4281d099 87260500
bdfbdaf2 7669db9e 5b6021c7 91e77aa9 e7e4fe4a c6cafefc 80ff180b e3ec8d33
6fb8172f 460daaf0 1f407230 bc282c25 9f14bc46 2d8d972e d27bf894 29696abb
03d37ff0 f4ff8c0b c1f62112 eba15368 218a94ca 16847d57 55522e27 60913848
50e84cbb 75438c26 9d216dee 67f82392 4f90a745 1b62b561 badadf9e d9bd481f
6c8d1152 307e5b3b ead13bea 6b3d0b8b 20c0e17e 84109d55 3c431bb3 20ce8ea1
c69e7cfb d720e186 dd4bbdb0 00008b23 f7271bcd ab1f10f2 009d98d1 66af8d49
bf41d101 7aec62b3 1c4ad813 f3508887 d626f23d 387e5398 0ac70ddb 2a5688b4
97f16918 b75f52ca d70751ce 3df694ff 7a07f6ba c1de8abd 6ff88df4 c48299b5
c3e056ec 28ab6d6b 7cd16a59 4f41617b 8cb3bdff a5819619 348cb287 6bf4dcdd
4985141b 2dd6f25a 49fce6cb b43a3dc3 8ca7dfcc fb3b4eca 4b1750eb 0d9da228
80bedd9a 56d1768e 5e883cb1 282c6746 792a9fe9 2a2eabfd 9e48f25c 25ee1f6e
f75c0d05 8ea213a4 f355cac0 608625f1 d78bc810 7d382ef3 0e87240a 4a4c1b87
0c0714a5 7ddd9d09 e12eaf2d 032a1264 cdaf6feb 52b4f3ea 3abb0304 518a11ec
3086d016 cc81461a 34fe9c5a 112a213d ffee305a 7133184c e05df5aa eabd1d9f
341c2951 a126eabf aad3fd0c 6f81faa4 1ccea61e 9a654dec 266147e4 cf14fed7
17656776 781ac09c 6b1e5650 e0a37e3d 98c7a384 3f3c001f e8e5db0f d9c03490
9775bf74 25cf5f63 5befb502 af14595a 56517525 5c3752d9 2f6c7de7 7e80fb37
b6c7d772 d117e4aa e6b1f3a3 1215889f 02111e63 93bd59cb b263a914 275efa37
8069f230 994564af 9f340363 293286ac 6a97d66b 8045fba9 41f41575 6f52d018
162d445c 2f8e6d47 99d00bf7 f419207f bfc51477 7c217b7d bfe9f660 acdd2c43
b6fcb8fc 37db0f6d 78e29e58 88f35340 18217600 cbae06d1 1f24f66c e3c876b6
5157a1e1 356aeed2 e5f4f5c9 3e4784c2 22678beb a7113bc2 e2de9439 382395c6
9c4d0e84 b900f9e1 88f6ec28 651d9462 bb3465d8 eb50af47
```

11. References

11.1. Normative References

[GOST28147]
Government Committee of the USSR for Standards,

"Cryptographic Protection for Data Processing System, Gosudarstvennyi Standard of USSR, GOST 28147-89: Encryption, Decryption, and Message Authentication Code (MAC) Algorithms (In Russian)", 1989.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4304] Kent, S., "Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP)", RFC 4304, December 2005.

[RFC4357] Popov, V., Kurepkin, I., and S. Leontiev, "Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms", RFC 4357, January 2006.

[TC26ESP] Technical committee #26 of Federal Agency on Technical Regulating and Metrology of the Russian Federation, "Cryptographic Protection for Data Processing System, Technical specification Use GOST 28147-89 for Encapsulating Security Payload (ESP) IPsec protocol. (In Russian)", 2013.

[TC26UZ] Technical committee #26 of Federal Agency on Technical Regulating and Metrology of the Russian Federation, "Methodical recommendations for S-box parameters assignment in GOST 28147-89. (In Russian)", 2014.

[rus-popov-esp-sbox-00-rb] Technical committee #26 of Federal Agency on Technical Regulating and Metrology of the Russian Federation, "Using additional S-box parameters in GOST 28147-89 for payload encryption in IPsec ESP. (Work in progress)", 2014.

11.2. Informative References

[RFC2407] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[RFC2408] Maughan, D., Schneider, M., and M. Schertler, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", RFC 2675, August 1999.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4490] Leontiev, S. and G. Chudov, "Using the GOST 28147-89, GOST R 34.11-94, GOST R 34.10-94, and GOST R 34.10-2001 Algorithms with Cryptographic Message Syntax (CMS)", RFC 4490, May 2006.

[RFC4491] Leontiev, S. and D. Shefanovski, "Using the GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms with the Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 4491, May 2006.

[RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", RFC 5116, January 2008.

[RFC5282] Black, D. and D. McGrew, "Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol", RFC 5282, August 2008.

[RFC5830] Dolmatov, V., "GOST 28147-89: Encryption, Decryption, and Message Authentication Code (MAC) Algorithms", RFC 5830, March 2010.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

[RFC6071] Frankel, S. and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap", RFC 6071, February 2011.

[RFC6311] Singh, R., Kalyani, G., Nir, Y., Sheffer, Y., and D. Zhang, "Protocol Support for High Availability of IKEv2/ IPsec", RFC 6311, July 2011.

[TC26IKE] Technical committee #26 of Federal Agency on Technical Regulating and Metrology of the Russian Federation, "Cryptographic Protection for Data Processing System, Technical specification Use GOST 28147-89, GOST R

34.11-94, GOST R 34.10-2001, for IKE and ISAKMP. (In Russian)", 2013.

Appendix A. Compatibility

Requirements for a transforms implementation:

- ESP_GOST-4M-IMIT - REQUIRED;
- ESP_GOST-1K-IMIT - OPTIONAL, it is required for applications that must be strictly robust to attacks based on timing and EMI analysis, and in case of usage very big IPv6 packets (more than 64 kilobytes);
- id-Gost28147-89-CryptoPro-B-ParamSet - REQUIRED;
- id-tc26-gost-28147-param-Z - REQUIRED.

Appendix B. Compatibility with Older IKEv1 Implementations

Some IKEv1 implementations are incompatible with the recommendations [RFC4301], [RFC4303], [RFC6071], and do not support ICV implementation and ICV check in case of negotiation "NULL" integrity algorithm, i.e. a proposal doesn't have an "Authentication Algorithm" (5) attribute.

Such IKEv1 protocol implementations MAY negotiate this attribute value for "Authentication Algorithm" (5):

GOST-NULL-INTEGRITY-ALGORITHM - 65411.

ESP SA behavior in such case MUST be the same as there is an absence of "Authentication Algorithm" (5) attribute.

Authors' Addresses

Serguei E. Leontiev (editor)
"CRYPTO-PRO", LLC
18, Suschevsky Val str.
Moscow 127018
Russian Federation

Phone: +7 (916) 686 10 81
Fax: +74957804820
Email: lse@cryptopro.ru
URI: http://www.cryptopro.ru

Dmitry N. Pichulin (editor)
"CRYPTO-PRO", LLC
18, Suschevsky Val str.
Moscow 127018
Russian Federation

Phone: +7 (905) 540 88 60
Fax: +74957804820
Email: pdn@cryptopro.ru
URI: http://www.cryptopro.ru

Andrey A. Fedchenko (editor)
"S-Terra" LLC
Zelenograd, 6, driveway 4806
Moscow 124460
Russian Federation

Fax: +74999409061
Email: hell@s-terra.com
URI: http://www.s-terra.com/