Stream: Independent Submission
RFC: 9385
Category: Informational
Published: May 2023
ISSN: 2070-1721
Author: V. Smyslov
ELVIS-PLUS

RFC 9385
Using GOST Cryptographic Algorithms in the Internet Key Exchange Protocol Version 2 (IKEv2)

Abstract

This document defines a set of cryptographic transforms for use in the Internet Key Exchange Protocol version 2 (IKEv2). The transforms are based on Russian cryptographic standard algorithms (called "GOST" algorithms). Use of GOST ciphers in IKEv2 is defined in RFC 9227. This document aims to define the use of GOST algorithms for the rest of the cryptographic transforms used in IKEv2.

This specification was developed to facilitate implementations that wish to support the GOST algorithms. This document does not imply IETF endorsement of the cryptographic algorithms used in this document.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9385.

Copyright Notice

Table of Contents

1. Introduction

The Internet Key Exchange Protocol version 2 (IKEv2) defined in [RFC7296] is an important part of the IP Security (IPsec) architecture. It is used for the authenticated key exchange and for the negotiation of various protocol parameters and features.

This document defines a number of transforms for IKEv2, based on Russian cryptographic standard algorithms (often referred to as "GOST" algorithms) for hash function, digital signature, and key exchange method. These definitions are based on the recommendations established by the Standardisation Technical Committee "Cryptographic information protection", which describe how Russian cryptographic standard algorithms are used in IKEv2 [GOST-IKEv2]. Along with the transforms defined in [RFC9227], the transforms defined in this specification allow for the use of GOST cryptographic algorithms in IPsec protocols.

This specification was developed to facilitate implementations that wish to support the GOST algorithms. This document does not imply IETF endorsement of the cryptographic algorithms used in this document.

2. Terminology and Notation

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Overview

Russian cryptographic standard algorithms (GOST algorithms) are a set of cryptographic algorithms of different types -- ciphers, hash functions, digital signatures, etc. In particular, Russian cryptographic standard [GOST3412-2015] defines the "Kuznyechik" and "Magma" block ciphers (also defined in [RFC7801] and [RFC8891], respectively). Cryptographic standard

[GOST3410-2012] defines the elliptic curve digital signature algorithm (also defined in [RFC7091]), while [GOST3411-2012] defines two cryptographic hash functions with different output lengths (also defined in [RFC6986]). These hash functions are often referred to as "Streebog" hash functions, although this is not an official name and is not used in the provided references. The parameters for the elliptic curves used in GOST signature and key exchange algorithms are defined in [RFC7836].

4. IKE SA Protection

IKE Security Association (SA) protection using GOST algorithms is defined in [RFC9227]. In particular, two transforms of Type 1 (Encryption Algorithm Transform IDs) can be used for IKE SA protection: ENCR_KUZNYECHIK_MGM_KTREE (32) based on the "Kuznyechik" block cipher and ENCR_MAGMA_MGM_KTREE (33) based on the "Magma" block cipher, both in Multilinear Galois Mode (MGM).

The information here is provided for convenience. For full details, please see [RFC9227].

5. Pseudorandom Function

This specification defines a new transform of Type 2 (Pseudorandom Function Transform IDs): PRF_HMAC_STREEBOG_512 (9). This transform uses the Pseudorandom Function (PRF) HMAC_GOSTR3411_2012_512 defined in Section 4.1.2 of [RFC7836]. The PRF uses the GOST R 34.11-2012 ("Streebog") hash function with a 512-bit output defined in [RFC6986] and [GOST3411-2012] with HMAC [RFC2104] construction. The PRF has a 512-bit block size and a 512-bit output length.

6. Shared Key Calculation

This specification defines two new transforms of Type 4 (Key Exchange Method Transform IDs): GOST3410_2012_256 (33) and GOST3410_2012_512 (34). These transforms use the Elliptic Curve Diffie-Hellman (ECDH) key exchange algorithm over twisted Edwards curves. The parameters for these curves are defined in Appendix A.2 of [RFC7836]. In particular, transform GOST3410_2012_256 uses the id-tc26-gost-3410-2012-256-paramSetA parameter set and GOST3410_2012_512 uses the id-tc26-gost-3410-2012-512-paramSetC parameter set (both defined in [RFC7836]).

The shared secret is computed as follows. The initiator randomly selects its private key d_i from {1,..,q - 1}, where q is the subgroup order and is a parameter of the selected curve. Then a public key Q_i is computed as a point on the curve:

```
   Q_i = d_i * G
```

where G is the generator for the selected curve. It is then sent to the responder. The responder makes the same calculations to get d_r and Q_r and sends Q_r to the initiator. After peers exchange Q_i and Q_R, both sides can compute a point on the curve:

```
   S = ((m / q) * d_i) * Q_r = ((m / q) * d_r) * Q_i
```

where m is the group order and is a parameter of the selected curve. The shared secret K is an x coordinate of S in a little-endian representation. The size of K is determined by the size of the used curve and is either 256 or 512 bits.

When the GOST public key is transmitted in the Key Exchange payload (Section 3.4 of [RFC7296]), it **MUST** be represented as x coordinate immediately followed by y coordinate, each in a little-endian representation. The size of each coordinate is determined by the size of the used curve and is either 256 or 512 bits, so that the size of the Key Exchange Data field in the Key Exchange payload is either 64 or 128 octets.

6.1. Recipient Tests

Upon receiving a peer's public key, implementations **MUST** check that the key is actually a point on the curve. Otherwise, the exchange fails. Implementations **MUST** check that the calculated public value S is not an identity element of the curve. If S appears to be the identity element of the curve, the exchange fails. The INVALID_SYNTAX notification **MAY** be sent in these cases.

7. Authentication

IKEv2 allows various authentication methods to be used for IKE SA establishment. Some methods are tied to a particular algorithm, while others may be used with different algorithms. This specification makes no restrictions on using the latter ones with the GOST algorithms. In particular, "Shared Key Message Integrity Code" (2), defined in [RFC7296], and "NULL Authentication" (13), defined in [RFC7619], can be used with GOST algorithms with no changes to the process of the AUTH payload content calculation.

When the GOST digital signature algorithm is used in IKEv2 for authentication purposes, the "Digital Signature" (14) authentication method, defined in [RFC7427], **MUST** be specified in the AUTH payload.

The GOST digital signature algorithm GOST R 34.10-2012 is defined in [RFC7091] and [GOST3410-2012]. There are two variants of the GOST digital signature algorithm -- one over a 256-bit elliptic curve and the other over a 512-bit key elliptic curve. The signature value, as defined in [RFC7091] and [GOST3410-2012], consists of two integers: r and s. The size of each integer is either 256 or 512 bits depending on the elliptic curve used. The content of the Signature Value field in the AUTH payload **MUST** consist of s immediately followed by r, each in a big-endian representation, so that the size of the field is either 64 or 128 octets. The AlgorithmIdentifier ASN.1 objects for the GOST digital signature algorithm are defined in Section 7.2.

7.1. Hash Functions

The GOST digital signature algorithm uses the GOST R 34.11-2012 ("Streebog") hash functions defined in [RFC6986] and [GOST3411-2012]. There are two "Streebog" hash functions: one with a 256-bit output length and the other with a 512-bit output length. The former is used with the GOST digital signature algorithm over a 256-bit elliptic curve and the latter over a 512-bit key elliptic curve.

This specification defines two new values for the "IKEv2 Hash Algorithms" registry: STREEBOG_256 (6) for the GOST hash function with a 256-bit output length and STREEBOG_512 (7) for the GOST hash function with a 512-bit output length. These values **MUST** be included in the SIGNATURE_HASH_ALGORITHMS notification if a corresponding GOST digital signature algorithm is supported by the sender and its local policy allows the use of this algorithm (see Section 4 of [RFC7427] for details).

7.2. ASN.1 Objects

This section lists GOST digital signature algorithm ASN.1 AlgorithmIdentifier objects in binary form. With GOST digital signature algorithms, optional parameters in AlgorithmIdentifier objects are always omitted. These objects are defined in [RFC9215] and [USING-GOST-IN-CERTS] and are provided here for convenience.

7.2.1. id-tc26-signwithdigest-gost3410-12-256

```
id-tc26-signwithdigest-gost3410-12-256 OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rosstandart(7) tc26(1)
        algorithms(1) signwithdigest(3) gost3410-12-256(2)}
```

The optional parameters field must be omitted.

```
Name = id-tc26-signwithdigest-gost3410-12-256
OID = 1.2.643.7.1.1.3.2
Length = 12
0000: 300a 0608 2a85 0307 0101 0302
```

7.2.2. id-tc26-signwithdigest-gost3410-12-512

```
id-tc26-signwithdigest-gost3410-12-512 OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rosstandart(7) tc26(1)
        algorithms(1) signwithdigest(3) gost3410-12-512(3)}
```

The optional parameters field must be omitted.

```
Name = id-tc26-signwithdigest-gost3410-12-512
OID = 1.2.643.7.1.1.3.3
Length = 12
0000: 300a 0608 2a85 0307 0101 0303
```

8. Security Considerations

The security considerations of [RFC7296] and [RFC7427] apply.

The security of GOST elliptic curves is discussed in [GOST-EC-SECURITY]. The security of the "Streebog" hash functions is discussed in [STREEBOG-SECURITY]. A second preimage attack on "Streebog" hash functions is described in [STREEBOG-PREIMAGE] if the message size exceeds 2^{259} blocks. This attack is not relevant to how "Streebog" hash functions are used in IKEv2.

9. IANA Considerations

IANA has assigned one Transform ID in the "Transform Type 2 - Pseudorandom Function Transform IDs" registry:

Number	Name	Reference
9	PRF_HMAC_STREEBOG_512	RFC 9385

Table 1: New Pseudorandom Function Transform ID

IANA has assigned two Transform IDs in the "Transform Type 4 - Key Exchange Method Transform IDs" registry:

Number	Name	Recipient Tests	Reference
33	GOST3410_2012_256	RFC 9385, Section 6.1	RFC 9385
34	GOST3410_2012_512	RFC 9385, Section 6.1	RFC 9385

Table 2: New Key Exchange Method Transform IDs

IANA has assigned two values in the "IKEv2 Hash Algorithms" registry:

Number	Hash Algorithm	Reference
6	STREEBOG_256	RFC 9385
7	STREEBOG_512	RFC 9385

Table 3: New IKEv2 Hash Algorithms

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC6986] Dolmatov, V., Ed. and A. Degtyarev, "GOST R 34.11-2012: Hash Function", RFC 6986, DOI 10.17487/RFC6986, August 2013, <https://www.rfc-editor.org/info/rfc6986>.

[RFC7091] Dolmatov, V., Ed. and A. Degtyarev, "GOST R 34.10-2012: Digital Signature Algorithm", RFC 7091, DOI 10.17487/RFC7091, December 2013, <https://www.rfc-editor.org/info/rfc7091>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[RFC7427] Kivinen, T. and J. Snyder, "Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)", RFC 7427, DOI 10.17487/RFC7427, January 2015, <https://www.rfc-editor.org/info/rfc7427>.

[RFC7836] Smyshlyaev, S., Ed., Alekseev, E., Oshkin, I., Popov, V., Leontiev, S., Podobaev, V., and D. Belyavsky, "Guidelines on the Cryptographic Algorithms to Accompany the Usage of Standards GOST R 34.10-2012 and GOST R 34.11-2012", RFC 7836, DOI 10.17487/RFC7836, March 2016, <https://www.rfc-editor.org/info/rfc7836>.

[RFC9215] Baryshkov, D., Ed., Nikolaev, V., and A. Chelpanov, "Using GOST R 34.10-2012 and GOST R 34.11-2012 Algorithms with the Internet X.509 Public Key Infrastructure", RFC 9215, DOI 10.17487/RFC9215, March 2022, <https://www.rfc-editor.org/info/rfc9215>.

[RFC9227] Smyslov, V., "Using GOST Ciphers in the Encapsulating Security Payload (ESP) and Internet Key Exchange Version 2 (IKEv2) Protocols", RFC 9227, DOI 10.17487/RFC9227, March 2022, <https://www.rfc-editor.org/info/rfc9227>.

10.2. Informative References

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[RFC7619] Smyslov, V. and P. Wouters, "The NULL Authentication Method in the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 7619, DOI 10.17487/RFC7619, August 2015, <https://www.rfc-editor.org/info/rfc7619>.

[RFC7801] Dolmatov, V., Ed., "GOST R 34.12-2015: Block Cipher "Kuznyechik"", RFC 7801, DOI 10.17487/RFC7801, March 2016, <https://www.rfc-editor.org/info/rfc7801>.

[RFC8891] Dolmatov, V., Ed. and D. Baryshkov, "GOST R 34.12-2015: Block Cipher "Magma"", RFC 8891, DOI 10.17487/RFC8891, September 2020, <https://www.rfc-editor.org/info/rfc8891>.

[GOST3410-2012] Federal Agency on Technical Regulating and Metrology, "Information technology. Cryptographic data security. Signature and verification processes of [electronic] digital signature", GOST R 34.10-2012, 2012. (In Russian)

[GOST3411-2012] Federal Agency on Technical Regulating and Metrology, "Information technology. Cryptographic data security. Hashing function", GOST R 34.11-2012, 2012. (In Russian)

[GOST3412-2015] Federal Agency on Technical Regulating and Metrology, "Information technology. Cryptographic data security. Block ciphers", GOST R 34.12-2015, 2015. (In Russian)

[GOST-IKEv2] Standardisation Technical Committee "Cryptographic information protection", "Information technology. Cryptographic data security. Using Russian cryptographic algorithms in the Internet Key Exchange protocol version 2 (IKEv2)", MR 26.2.001-22, 2022. (In Russian)

[GOST-IKEv2-TESTVECTORS] Standardisation Technical Committee "Cryptographic information protection", "Information technology. Cryptographic data security. The test vectors for the use of Russian cryptographic algorithms in the IKEv2 key exchange protocol", MR 26.2.002-22, 2022. (In Russian)

[USING-GOST-IN-CERTS] Federal Agency on Technical Regulating and Metrology, "Information technology. Cryptographic data security. Usage of GOST R 34.10-2012 and GOST R 34.11-2012 algorithms in certificate, CRL and PKCS#10 certificate request in X. 509 public key infrastructure", R 1323565.1.023-2018, 2018. (In Russian)

[GOST-EC-SECURITY] Alekseev, E., Nikolaev, V., and S. Smyshlyaev, "On the security properties of Russian standardized elliptic curves", DOI 10.4213/mvk260, 2018, <https://doi.org/10.4213/mvk260>.

[STREEBOG-SECURITY] Wang, Z., Yu, H., and X. Wang, "Cryptanalysis of GOST R hash function", DOI 10.1016/j.ipl.2014.07.007, December 2014, <https://doi.org/10.1016/j.ipl.2014.07.007>.

[STREEBOG-PREIMAGE] Guo, J., Jean, J., Leurent, G., Peyrin, T., and L. Wang, "The Usage of Counter Revisited: Second-Preimage Attack on New Russian Standardized Hash Function", Cryptology ePrint Archive, Paper 2014/675, 2014, <https://eprint.iacr.org/2014/675>.

Appendix A. Test Vectors

This appendix contains test vectors for two scenarios. The test vectors were borrowed from [GOST-IKEv2-TESTVECTORS]. In both scenarios, peers establish, rekey, and delete an IKE SA and ESP SAs. The IP addresses of the peers used in both scenarios are the same:

- initiator's IP address is 10.111.10.171
- responder's IP address is 10.111.10.45

The test vectors also cover IKE message protection for transforms defined in [RFC9227]. The keys SK_ei and SK_er are transform keys (see Section 4.4 of [RFC9227]), and the keys K1i, K2i, K3i, K1r, K2r, and K3r represent nodes in the key tree for the initiator and responder correspondently. The leaf keys K3i and K3r are effectively message protection keys (K_msg in terms of [RFC9227]). MGM nonces (also known as Initial Counter Nonces) are defined in Section 4.3 of [RFC9227]. The Initialization Vector (IV) format is defined in Section 4.2 of [RFC9227], and the Additional Authenticated Data (AAD) format is defined in Section 4.7 of [RFC9227].

All other keys and entities used in the test vectors are defined in [RFC7296].

A.1. Scenario 1

In this scenario, peers establish, rekey, and delete an IKE SA and ESP SAs using the following prerequisites:

- Peers authenticate each other using a Pre-Shared Key (PSK).
- Initiator's ID is "IKE-Initiator" of type ID_FQDN.
- Responder's ID is "IKE-Responder" of type ID_FQDN.
- No NAT is present between the peers.
- IKE fragmentation is not used.
- IKE SA is created with the following transforms:
  - ENCR_KUZNYECHIK_MGM_KTREE
  - PRF_HMAC_STREEBOG_512
  - GOST3410_2012_512
- ESP SAs are created with the following transforms:
  - ENCR_KUZNYECHIK_MGM_KTREE
  - ESN off

The 256-bit PSK used for authentication:

```
00000000: e2 69 24 cf 15 32 93 47 3a 11 a4 97 a8 a4 5c b3
00000010: 4e 28 31 ef 0e 28 bb 77 69 69 c6 3c 68 bf e1 0d
```

This scenario includes four sub-scenarios, which are described below.

A.1.1. Sub-Scenario 1: Establishment of IKE and ESP SAs Using the IKE_SA_INIT and the IKE_AUTH Exchanges

```
Initiator                                 Responder

HDR, SAi1, KEi, Ni [,N+]      --->
                              <---    HDR, SAr1, KEr, Nr [,N+]

HDR, SK {IDi, [IDr,] [N+,]
     AUTH, SAi2, TSi, TSr}    --->
                              <---    HDR, SK {IDr, [N+,]
                                           AUTH, SAr2, TSi, TSr}
```

Initiator's actions:

(1) Generates random SPIi for IKE SA

```
00000000: e9 d3 f3 78 19 1c 38 40
```

(2) Generates random IKE nonce Ni

```
00000000: 48 b6 d3 b3 ab 56 f2 c8 f0 42 d5 16 e7 21 d9 31
00000010: f9 ac 10 f9 7f 80 8c 51 2b d6 f4 59 93 a7 4d 13
```

(3) Generates ephemeral private key

```
00000000: 95 07 3a 04 dc db ce 77 f5 5e 4f fe 97 0c cd 6f
00000010: 0a e0 b5 c6 53 bd a0 da 47 fc 03 b5 8a e1 d5 1d
00000020: 89 e6 c0 db dc b1 ea 74 59 1f 1d 0c 9f 3f 4f dc
00000030: 10 d5 c9 cc a4 34 9c 3d 3e 6b dd 57 c5 d6 c9 01
```

(4) Computes public key

```
00000000: 96 1b 9b 21 4f 7e e9 83 ec 27 a0 64 0c 77 4f be
00000010: 78 31 be fd 1e 63 7d 6e 76 eb 2f 81 23 80 62 87
00000020: ba 2c f7 31 a2 70 b7 3e 8a 1d 91 93 72 cf 61 c8
00000030: d3 18 f6 bc f7 a0 44 c8 11 a7 fe d2 99 ea 8b 4d
00000040: 59 fa a7 38 ae 03 48 d2 aa f7 ff 11 e0 60 29 dd
00000050: 16 59 58 78 8e 3b e2 b5 48 36 3c ca 07 1a 5d be
00000060: a7 42 79 81 74 22 6f 53 15 d2 c2 f6 06 d4 0f ed
00000070: 70 f0 1c cf 89 2e ac 3c fe 01 02 91 85 06 7b d4
```

(5) Creates message

```
IKE SA Init
E9D3F378191C3840.0000000000000000.00000000 IKEv2 R<-I[316]
  SA[52]{
    P[48](#1:IKE::5#){
      Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
                 ENCR_MAGMA_MGM_KTREE,
      PRF=PRF_HMAC_STREEBOG_512,
      KE=GOST3410_2012_512,
         GOST3410_2012_256}},
  KE[136](GOST3410_2012_512){961B9B...067BD4},
  NONCE[36]{48B6D3...A74D13},
  N[28](NAT_DETECTION_SOURCE_IP){92B291...F4E2BF},
  N[28](NAT_DETECTION_DESTINATION_IP){77E199...98A613},
  N[8](IKEV2_FRAGMENTATION_SUPPORTED)
```

(6) Sends message, peer receives message

```
       10.111.10.171:54294->10.111.15.45:500 [316]

00000000: e9 d3 f3 78 19 1c 38 40 00 00 00 00 00 00 00 00
00000010: 21 20 22 08 00 00 00 00 00 00 01 3c 22 00 00 34
00000020: 00 00 00 30 01 01 00 05 03 00 00 08 01 00 00 20
00000030: 03 00 00 08 01 00 00 21 03 00 00 08 02 00 00 09
00000040: 03 00 00 08 04 00 00 22 00 00 00 08 04 00 00 21
00000050: 28 00 00 88 00 22 00 00 96 1b 9b 21 4f 7e e9 83
00000060: ec 27 a0 64 0c 77 4f be 78 31 be fd 1e 63 7d 6e
00000070: 76 eb 2f 81 23 80 62 87 ba 2c f7 31 a2 70 b7 3e
00000080: 8a 1d 91 93 72 cf 61 c8 d3 18 f6 bc f7 a0 44 c8
00000090: 11 a7 fe d2 99 ea 8b 4d 59 fa a7 38 ae 03 48 d2
000000A0: aa f7 ff 11 e0 60 29 dd 16 59 58 78 8e 3b e2 b5
000000B0: 48 36 3c ca 07 1a 5d be a7 42 79 81 74 22 6f 53
000000C0: 15 d2 c2 f6 06 d4 0f ed 70 f0 1c cf 89 2e ac 3c
000000D0: fe 01 02 91 85 06 7b d4 29 00 00 24 48 b6 d3 b3
000000E0: ab 56 f2 c8 f0 42 d5 16 e7 21 d9 31 f9 ac 10 f9
000000F0: 7f 80 8c 51 2b d6 f4 59 93 a7 4d 13 29 00 00 1c
00000100: 00 00 40 04 92 b2 91 d3 9b 53 51 c8 33 c2 1f 2e
00000110: 92 ef 24 88 ef f4 e2 bf 29 00 00 1c 00 00 40 05
00000120: 77 e1 99 fe 3b 7e 33 42 b5 af ad 51 cf 97 91 4b
00000130: 08 98 a6 13 00 00 00 08 00 00 40 2e
```

Responder's actions:

(7) Parses received message

```
IKE SA Init
E9D3F378191C3840.0000000000000000.00000000 IKEv2 I->R[316]
  SA[52]{
    P[48](#1:IKE::5#){
      Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
                 ENCR_MAGMA_MGM_KTREE,
      PRF=PRF_HMAC_STREEBOG_512,
      KE=GOST3410_2012_512,
         GOST3410_2012_256}},
  KE[136](GOST3410_2012_512){961B9B...067BD4},
  NONCE[36]{48B6D3...A74D13},
  N[28](NAT_DETECTION_SOURCE_IP){92B291...F4E2BF},
  N[28](NAT_DETECTION_DESTINATION_IP){77E199...98A613},
  N[8](IKEV2_FRAGMENTATION_SUPPORTED)
```

(8) Generates random SPIr for IKE SA

```
00000000: 8d df f4 01 fb fb 0b 14
```

(9) Generates random IKE nonce Nr

```
00000000: fb 81 c8 80 e5 f0 35 60 99 ef 46 b2 72 44 95 0f
00000010: 03 85 f4 73 92 67 b7 68 43 8f 90 69 16 fe 63 f0
```

(10) Generates ephemeral private key

```
00000000: 7f 49 e3 77 39 db 03 cc fe fe c9 63 17 71 e9 f1
00000010: 50 4b 98 79 b3 df 3b 48 bd f3 89 72 52 07 47 4f
00000020: 70 29 f8 39 63 2c 89 b6 92 39 18 27 9c fb 80 f5
00000030: 43 af 8b 9c 68 bb 93 22 1e 18 7d c2 1b dc e1 22
```

(11) Computes public key

```
00000000: ad b4 e4 db b9 af 28 59 ab 76 4d 30 fd d4 7a f3
00000010: 5f 8c cb 85 8c cc ca 30 5e 4a 9d 20 52 32 48 88
00000020: 69 81 48 5e ae db 1e 8c 0d 8d db 12 3e f5 ef 1d
00000030: 7f e8 83 39 7f e6 5d 6e 51 ca 9e ee f5 b6 ba 02
00000040: db 10 87 47 ba 38 b3 17 95 60 6d a3 81 15 5c 3d
00000050: 6b 86 d3 59 2f 5f 74 14 17 a9 64 20 3d 05 12 08
00000060: 02 75 15 ac ff 08 7c aa 82 1d f6 89 6c f4 33 e0
00000070: 01 4e 11 68 73 7e e3 e9 c6 88 ce 90 9b 39 05 48
```

(12) Creates message

```
IKE SA Init
E9D3F378191C3840.8DDFF401FBFB0B14.00000000 IKEv2 I<=R[300]
  SA[36]{
    P[32](#1:IKE::3#){
      Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
      PRF=PRF_HMAC_STREEBOG_512,
      KE=GOST3410_2012_512}},
  KE[136](GOST3410_2012_512){ADB4E4...390548},
  NONCE[36]{FB81C8...FE63F0},
  N[28](NAT_DETECTION_SOURCE_IP){6D7A48...683D59},
  N[28](NAT_DETECTION_DESTINATION_IP){481A5B...905499},
  N[8](IKEV2_FRAGMENTATION_SUPPORTED)
```

(13) Sends message, peer receives message

```
       10.111.10.171:54294<-10.111.15.45:500 [300]

00000000: e9 d3 f3 78 19 1c 38 40 8d df f4 01 fb fb 0b 14
00000010: 21 20 22 20 00 00 00 00 00 00 01 2c 22 00 00 24
00000020: 00 00 00 20 01 01 00 03 03 00 00 08 01 00 00 20
00000030: 03 00 00 08 02 00 00 09 00 00 00 08 04 00 00 22
00000040: 28 00 00 88 00 22 00 00 ad b4 e4 db b9 af 28 59
00000050: ab 76 4d 30 fd d4 7a f3 5f 8c cb 85 8c cc ca 30
00000060: 5e 4a 9d 20 52 32 48 88 69 81 48 5e ae db 1e 8c
00000070: 0d 8d db 12 3e f5 ef 1d 7f e8 83 39 7f e6 5d 6e
00000080: 51 ca 9e ee f5 b6 ba 02 db 10 87 47 ba 38 b3 17
00000090: 95 60 6d a3 81 15 5c 3d 6b 86 d3 59 2f 5f 74 14
000000A0: 17 a9 64 20 3d 05 12 08 02 75 15 ac ff 08 7c aa
000000B0: 82 1d f6 89 6c f4 33 e0 01 4e 11 68 73 7e e3 e9
000000C0: c6 88 ce 90 9b 39 05 48 29 00 00 24 fb 81 c8 80
000000D0: e5 f0 35 60 99 ef 46 b2 72 44 95 0f 03 85 f4 73
000000E0: 92 67 b7 68 43 8f 90 69 16 fe 63 f0 29 00 00 1c
000000F0: 00 00 40 04 6d 7a 48 7a 9d ce 80 6f b0 09 4b f7
00000100: 8d fd ec eb 2e 68 3d 59 29 00 00 1c 00 00 40 05
00000110: 48 1a 5b 15 12 e4 26 a3 8d 88 8b 65 8e 17 b3 f1
00000120: 38 90 54 99 00 00 00 08 00 00 40 2e
```

Initiator's actions:

(14) Parses received message

```
IKE SA Init
E9D3F378191C3840.8DDFF401FBFB0B14.00000000 IKEv2 R=>I[300]
  SA[36]{
    P[32](#1:IKE::3#){
      Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
      PRF=PRF_HMAC_STREEBOG_512,
      KE=GOST3410_2012_512}},
  KE[136](GOST3410_2012_512){ADB4E4...390548},
  NONCE[36]{FB81C8...FE63F0},
  N[28](NAT_DETECTION_SOURCE_IP){6D7A48...683D59},
  N[28](NAT_DETECTION_DESTINATION_IP){481A5B...905499},
  N[8](IKEV2_FRAGMENTATION_SUPPORTED)
```

(15) Computes shared key

```
00000000: a2 43 6c bd 2d c1 0f 81 0d f7 6f 24 ae 78 70 f2
00000010: 27 5d 1b dc c5 52 0e d8 53 e5 c5 43 98 f7 35 ce
00000020: 32 70 89 2b 8e 89 0b 7d b3 98 77 cd bd 31 5d 18
00000030: 10 5d 8b ac 16 f0 aa fd bc dc 7c 69 75 14 48 a8
```

(16) Computes SKEYSEED

```
00000000: fc 7b d9 80 4b 15 00 60 d2 08 17 3a 08 4b a9 2a
00000010: 0f 01 cb c3 ef e9 b5 aa 15 5b 0e 80 24 68 3c 4c
00000020: 6c fb e9 c8 16 7d 54 2d 48 ee 61 71 01 68 ca 68
00000030: 4f 7c b0 1b 61 29 20 9a 68 88 5b 3f d7 19 0b d0
```

(17) Computes SK_d

```
00000000: 6b 2b 83 d7 a9 10 5f f4 27 e8 05 86 b7 f0 09 31
00000010: 16 43 81 ae 88 7a 3f c9 65 30 73 00 e5 82 81 52
00000020: 68 07 ba e5 39 ef 6e a7 75 db 2c c9 1c d3 4b 70
00000030: e0 be 97 14 81 bb 0c 80 ef b3 6e 12 2a 08 74 36
```

(18) Computes SK_ei

```
00000000: 8c 6d f1 8f 6a ff 9f 1b 3e be 40 ef e2 64 c2 bf
00000010: 8e 6e d7 4c b5 8b 0a 74 a7 30 0c 21 7e 66 c7 d4
00000020: 83 00 37 c3 08 01 7e c3 0a 71 62 01
```

(19) Computes SK_er

```
00000000: df e8 7d 5f 9c da 5e 45 b8 b9 11 02 63 6c 08 47
00000010: f6 4f c5 5d 6a 7b 4b 91 52 32 0a a2 5e c0 31 34
00000020: 65 20 72 e7 0a 1e ff 7d da ba 17 31
```

(20) Computes SK_pi

```
00000000: 93 11 c6 4c d7 12 b5 40 f9 e8 7e 73 c5 28 a7 d8
00000010: 89 48 1c f1 bf a3 ad 67 cf b4 d9 6a 9b fe 3c ea
00000020: 2f cc 2a 5e d4 e4 0b 27 7f be c9 9d c3 8d b7 68
00000030: 03 c1 f3 f8 94 af 47 8b d8 35 b8 6b c2 ca 38 16
```

(21) Computes SK_pr

```
00000000: 7b b0 4b 24 74 9c 73 68 7f 34 a3 b8 17 6b 9e 30
00000010: f2 eb 33 73 23 ff 49 1e e3 07 e7 9f 77 b6 2a ef
00000020: 5a 5e a9 02 8e 90 5c 83 49 ec 1e aa a4 05 bc e1
00000030: fb c4 5b f0 27 d6 9b 41 77 6f e1 48 f3 37 99 e5
```

(22) Computes prf(SK_pi, IDi)

```
00000000: 06 d3 d4 36 ab 5b 4f 41 d4 3d fc 79 1f 13 a3 89
00000010: e9 a6 6e d7 87 7d 72 d1 9d 71 78 2d 05 ee 47 fb
00000020: 82 c8 8f 86 cd b5 05 1d 25 7c 1e 79 18 ef 4e 4e
00000030: 8d ca f4 47 12 c6 7f 6a 32 7d d8 e8 f2 8e f8 33
```

(23) Uses PSK

```
00000000: e2 69 24 cf 15 32 93 47 3a 11 a4 97 a8 a4 5c b3
00000010: 4e 28 31 ef 0e 28 bb 77 69 69 c6 3c 68 bf e1 0d
```

(24) Computes prf(PSK,"Key Pad for IKEv2")

```
00000000: 01 3c a5 24 59 4e bc 78 99 20 61 6c 3f 03 e5 2e
00000010: 7a 75 2a 0b 78 36 bd 0a 89 ce 1d e7 8b 23 32 ae
00000020: 08 9a a0 03 1d da f6 14 8c 38 c6 bd 7c 03 13 24
00000030: bd af c8 ad 88 18 8f 41 d0 12 b9 e1 5a 66 8f 10
```

(25) Computes content of AUTH payload

```
00000000: c9 9b 01 9a 89 ee 56 53 ab 28 25 a1 d7 51 54 ac
00000010: 01 42 fb d6 2e bc 1e f3 65 73 63 5b 16 81 4b 97
00000020: 38 b4 20 5d 09 d9 b4 21 b4 0c f4 55 27 80 e7 4c
00000030: cf 66 d0 14 25 87 7c 20 84 68 d5 79 3a 74 1e e3
```

(26) Computes K1i (i1 = 0)

```
00000000: f2 ac 10 7a 1f 92 d1 b1 1b b1 74 c3 42 76 a3 3f
00000010: fa ea 1b 1e 81 10 c1 01 7a 25 9a 00 8d 76 57 de
```

(27) Computes K2i (i2 = 0)

```
00000000: 77 e0 16 18 ad 76 e8 5a 66 2f 88 c4 c0 92 ec 33
00000010: 6d 23 63 28 28 d5 77 d8 84 e1 01 b1 8d 84 a7 1d
```

(28) Computes K3i (i3 = 0)

```
00000000: 36 ff fa db 84 a9 f1 21 d5 84 16 db eb af 21 a2
00000010: 12 6d 5c 35 95 fe 89 cf 27 47 52 8a b7 36 92 d4
```

(29) Selects SPI for incoming ESP SA

```
00000000: 0a de 5f cd
```

(30) Creates message

```
IKE SA Auth
E9D3F378191C3840.8DDFF401FBFB0B14.00000001 IKEv2 R<-I[334]
  E[306]{
    IDi[21](FQDN){"IKE-Initiator"},
    AUTH[72](Preshared-Key){C99B01...741EE3},
    N[8](INITIAL_CONTACT),
    N[12](SET_WINDOW_SIZE){4},
    CP[16](REQUEST){IP4.Address[0], IP4.DNS[0]},
    SA[56]{
      P[52](#1:ESP:0ADE5FCD:5#){
        Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
                   ENCR_MAGMA_MGM_KTREE,
                   ENCR_KUZNYECHIK_MGM_MAC_KTREE,
                   ENCR_MAGMA_MGM_MAC_KTREE,
        ESN=Off}},
    TSi[40](2#){10.111.10.171:icmp:8.0, 0.0.0.0-255.255.255.255},
    TSr[40](2#){10.0.0.2:icmp:8.0, 10.0.0.0-10.0.0.255},
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(31) Composes MGM nonce

```
00000000: 00 00 00 00 83 00 37 c3 08 01 7e c3 0a 71 62 01
```

(32) Composes AAD

```
00000000: e9 d3 f3 78 19 1c 38 40 8d df f4 01 fb fb 0b 14
00000010: 2e 20 23 08 00 00 00 01 00 00 01 4e 23 00 01 32
```

(33) Composes plaintext

```
00000000: 27 00 00 15 02 00 00 00 49 4b 45 2d 49 6e 69 74
00000010: 69 61 74 6f 72 29 00 00 48 02 00 00 00 c9 9b 01
00000020: 9a 89 ee 56 53 ab 28 25 a1 d7 51 54 ac 01 42 fb
00000030: d6 2e bc 1e f3 65 73 63 5b 16 81 4b 97 38 b4 20
00000040: 5d 09 d9 b4 21 b4 0c f4 55 27 80 e7 4c cf 66 d0
00000050: 14 25 87 7c 20 84 68 d5 79 3a 74 1e e3 29 00 00
00000060: 08 00 00 40 00 2f 00 00 0c 00 00 40 01 00 00 00
00000070: 04 21 00 00 10 01 00 00 00 00 01 00 00 00 03 00
00000080: 00 2c 00 00 38 00 00 00 34 01 03 04 05 0a de 5f
00000090: cd 03 00 00 08 01 00 00 20 03 00 00 08 01 00 00
000000A0: 21 03 00 00 08 01 00 00 22 03 00 00 08 01 00 00
000000B0: 23 00 00 00 08 05 00 00 00 2d 00 00 28 02 00 00
000000C0: 00 07 01 00 10 08 00 08 00 0a 6f 0a ab 0a 6f 0a
000000D0: ab 07 00 00 10 00 00 ff ff 00 00 00 00 ff ff ff
000000E0: ff 29 00 00 28 02 00 00 00 07 01 00 10 08 00 08
000000F0: 00 0a 00 00 02 0a 00 00 02 07 00 00 10 00 00 ff
00000100: ff 0a 00 00 00 0a 00 00 ff 29 00 00 08 00 00 40
00000110: 0a 00 00 00 08 00 00 40 0b 00
```

(34) Encrypts plaintext using K3i as K_msg, resulting in ciphertext

```
00000000: a5 7d 65 70 aa c3 ef f7 df d6 5c 58 f6 2e ea 80
00000010: 82 15 dc 9d ae 42 1c f0 4c e4 cd 2a 45 f0 22 96
00000020: ea d2 06 cc 9b 59 97 9e 45 5d 27 5f b4 fd 55 6a
00000030: 90 bb 14 da df 9f 56 b0 e8 4c 89 a5 d8 f1 f6 55
00000040: a9 f0 82 90 57 28 86 a5 bd 12 85 2f 2e 51 54 29
00000050: fe 04 45 a4 90 f0 f8 0e 8b e9 c7 37 05 8f 6b bb
00000060: 36 b0 24 8a 5f a3 ca f3 7e 7d f9 8e 73 4b b0 14
00000070: ce b0 af 63 4c 4f ea 60 f6 46 4c 61 76 7c 9f 18
00000080: 0c 61 73 fa 30 9f 91 c4 22 c9 ab 61 80 5a de 8e
00000090: 06 40 36 7a 71 59 a5 ad 1c 67 25 03 9b af 2b 04
000000A0: 9f c1 de 51 11 7b f1 16 20 81 78 3f a8 01 d6 c8
000000B0: 79 89 d9 65 3e ea 58 6d ac 48 fc 4a 9a b9 48 02
000000C0: d7 2b 01 5d 6a 2d cb 65 bb ad 99 86 e2 03 08 76
000000D0: 1b dd 7c 56 3c 49 a4 2c da 24 1f ad 54 79 f5 d8
000000E0: 0e 52 8a 49 92 90 66 80 85 00 b7 d8 89 5f b7 f4
000000F0: 92 c1 5b ed 8a 16 00 f3 9a f8 90 4b fa 6a b2 de
00000100: 2a 89 74 9f 99 c7 c3 57 88 5b 88 95 5c ec 46 52
00000110: 04 c4 49 08 05 ab ee 1c 80 f6
```

(35) Computes ICV using K3i as K_msg

```
00000000: 7a 4f 14 38 e6 5f 6b 8c f5 5d 55 f5
```

(36) Composes IV

```
00000000: 00 00 00 00 00 00 00 00
```

(37) Sends message, peer receives message

```
       10.111.10.171:54294->10.111.15.45:500 [334]

00000000: e9 d3 f3 78 19 1c 38 40 8d df f4 01 fb fb 0b 14
00000010: 2e 20 23 08 00 00 00 01 00 00 01 4e 23 00 01 32
00000020: 00 00 00 00 00 00 00 00 a5 7d 65 70 aa c3 ef f7
00000030: df d6 5c 58 f6 2e ea 80 82 15 dc 9d ae 42 1c f0
00000040: 4c e4 cd 2a 45 f0 22 96 ea d2 06 cc 9b 59 97 9e
00000050: 45 5d 27 5f b4 fd 55 6a 90 bb 14 da df 9f 56 b0
00000060: e8 4c 89 a5 d8 f1 f6 55 a9 f0 82 90 57 28 86 a5
00000070: bd 12 85 2f 2e 51 54 29 fe 04 45 a4 90 f0 f8 0e
00000080: 8b e9 c7 37 05 8f 6b bb 36 b0 24 8a 5f a3 ca f3
00000090: 7e 7d f9 8e 73 4b b0 14 ce b0 af 63 4c 4f ea 60
000000A0: f6 46 4c 61 76 7c 9f 18 0c 61 73 fa 30 9f 91 c4
000000B0: 22 c9 ab 61 80 5a de 8e 06 40 36 7a 71 59 a5 ad
000000C0: 1c 67 25 03 9b af 2b 04 9f c1 de 51 11 7b f1 16
000000D0: 20 81 78 3f a8 01 d6 c8 79 89 d9 65 3e ea 58 6d
000000E0: ac 48 fc 4a 9a b9 48 02 d7 2b 01 5d 6a 2d cb 65
000000F0: bb ad 99 86 e2 03 08 76 1b dd 7c 56 3c 49 a4 2c
00000100: da 24 1f ad 54 79 f5 d8 0e 52 8a 49 92 90 66 80
00000110: 85 00 b7 d8 89 5f b7 f4 92 c1 5b ed 8a 16 00 f3
00000120: 9a f8 90 4b fa 6a b2 de 2a 89 74 9f 99 c7 c3 57
00000130: 88 5b 88 95 5c ec 46 52 04 c4 49 08 05 ab ee 1c
00000140: 80 f6 7a 4f 14 38 e6 5f 6b 8c f5 5d 55 f5
```

Responder's actions:

(38) Computes shared key

```
00000000: a2 43 6c bd 2d c1 0f 81 0d f7 6f 24 ae 78 70 f2
00000010: 27 5d 1b dc c5 52 0e d8 53 e5 c5 43 98 f7 35 ce
00000020: 32 70 89 2b 8e 89 0b 7d b3 98 77 cd bd 31 5d 18
00000030: 10 5d 8b ac 16 f0 aa fd bc dc 7c 69 75 14 48 a8
```

(39) Computes SKEYSEED

```
00000000: fc 7b d9 80 4b 15 00 60 d2 08 17 3a 08 4b a9 2a
00000010: 0f 01 cb c3 ef e9 b5 aa 15 5b 0e 80 24 68 3c 4c
00000020: 6c fb e9 c8 16 7d 54 2d 48 ee 61 71 01 68 ca 68
00000030: 4f 7c b0 1b 61 29 20 9a 68 88 5b 3f d7 19 0b d0
```

(40) Computes SK_d

```
00000000: 6b 2b 83 d7 a9 10 5f f4 27 e8 05 86 b7 f0 09 31
00000010: 16 43 81 ae 88 7a 3f c9 65 30 73 00 e5 82 81 52
00000020: 68 07 ba e5 39 ef 6e a7 75 db 2c c9 1c d3 4b 70
00000030: e0 be 97 14 81 bb 0c 80 ef b3 6e 12 2a 08 74 36
```

(41) Computes SK_ei

```
00000000: 8c 6d f1 8f 6a ff 9f 1b 3e be 40 ef e2 64 c2 bf
00000010: 8e 6e d7 4c b5 8b 0a 74 a7 30 0c 21 7e 66 c7 d4
00000020: 83 00 37 c3 08 01 7e c3 0a 71 62 01
```

(42) Computes SK_er

```
00000000: df e8 7d 5f 9c da 5e 45 b8 b9 11 02 63 6c 08 47
00000010: f6 4f c5 5d 6a 7b 4b 91 52 32 0a a2 5e c0 31 34
00000020: 65 20 72 e7 0a 1e ff 7d da ba 17 31
```

(43) Computes SK_pi

```
00000000: 93 11 c6 4c d7 12 b5 40 f9 e8 7e 73 c5 28 a7 d8
00000010: 89 48 1c f1 bf a3 ad 67 cf b4 d9 6a 9b fe 3c ea
00000020: 2f cc 2a 5e d4 e4 0b 27 7f be c9 9d c3 8d b7 68
00000030: 03 c1 f3 f8 94 af 47 8b d8 35 b8 6b c2 ca 38 16
```

(44) Computes SK_pr

```
00000000: 7b b0 4b 24 74 9c 73 68 7f 34 a3 b8 17 6b 9e 30
00000010: f2 eb 33 73 23 ff 49 1e e3 07 e7 9f 77 b6 2a ef
00000020: 5a 5e a9 02 8e 90 5c 83 49 ec 1e aa a4 05 bc e1
00000030: fb c4 5b f0 27 d6 9b 41 77 6f e1 48 f3 37 99 e5
```

(45) Extracts IV from message

```
00000000: 00 00 00 00 00 00 00 00
```

(46) Computes K1i (i1 = 0)

```
00000000: f2 ac 10 7a 1f 92 d1 b1 1b b1 74 c3 42 76 a3 3f
00000010: fa ea 1b 1e 81 10 c1 01 7a 25 9a 00 8d 76 57 de
```

(47) Computes K2i (i2 = 0)

```
00000000: 77 e0 16 18 ad 76 e8 5a 66 2f 88 c4 c0 92 ec 33
00000010: 6d 23 63 28 28 d5 77 d8 84 e1 01 b1 8d 84 a7 1d
```

(48) Computes K3i (i3 = 0)

```
00000000: 36 ff fa db 84 a9 f1 21 d5 84 16 db eb af 21 a2
00000010: 12 6d 5c 35 95 fe 89 cf 27 47 52 8a b7 36 92 d4
```

(49) Composes MGM nonce

```
00000000: 00 00 00 00 83 00 37 c3 08 01 7e c3 0a 71 62 01
```

(50) Extracts ICV from message

```
00000000: 7a 4f 14 38 e6 5f 6b 8c f5 5d 55 f5
```

(51) Extracts AAD from message

```
00000000: e9 d3 f3 78 19 1c 38 40 8d df f4 01 fb fb 0b 14
00000010: 2e 20 23 08 00 00 00 01 00 00 01 4e 23 00 01 32
```

(52) Extracts ciphertext from message

```
00000000: a5 7d 65 70 aa c3 ef f7 df d6 5c 58 f6 2e ea 80
00000010: 82 15 dc 9d ae 42 1c f0 4c e4 cd 2a 45 f0 22 96
00000020: ea d2 06 cc 9b 59 97 9e 45 5d 27 5f b4 fd 55 6a
00000030: 90 bb 14 da df 9f 56 b0 e8 4c 89 a5 d8 f1 f6 55
00000040: a9 f0 82 90 57 28 86 a5 bd 12 85 2f 2e 51 54 29
00000050: fe 04 45 a4 90 f0 f8 0e 8b e9 c7 37 05 8f 6b bb
00000060: 36 b0 24 8a 5f a3 ca f3 7e 7d f9 8e 73 4b b0 14
00000070: ce b0 af 63 4c 4f ea 60 f6 46 4c 61 76 7c 9f 18
00000080: 0c 61 73 fa 30 9f 91 c4 22 c9 ab 61 80 5a de 8e
00000090: 06 40 36 7a 71 59 a5 ad 1c 67 25 03 9b af 2b 04
000000A0: 9f c1 de 51 11 7b f1 16 20 81 78 3f a8 01 d6 c8
000000B0: 79 89 d9 65 3e ea 58 6d ac 48 fc 4a 9a b9 48 02
000000C0: d7 2b 01 5d 6a 2d cb 65 bb ad 99 86 e2 03 08 76
000000D0: 1b dd 7c 56 3c 49 a4 2c da 24 1f ad 54 79 f5 d8
000000E0: 0e 52 8a 49 92 90 66 80 85 00 b7 d8 89 5f b7 f4
000000F0: 92 c1 5b ed 8a 16 00 f3 9a f8 90 4b fa 6a b2 de
00000100: 2a 89 74 9f 99 c7 c3 57 88 5b 88 95 5c ec 46 52
00000110: 04 c4 49 08 05 ab ee 1c 80 f6
```

(53) Decrypts ciphertext and verifies ICV using K3i as K_msg, resulting in plaintext

```
00000000: 27 00 00 15 02 00 00 00 49 4b 45 2d 49 6e 69 74
00000010: 69 61 74 6f 72 29 00 00 48 02 00 00 00 c9 9b 01
00000020: 9a 89 ee 56 53 ab 28 25 a1 d7 51 54 ac 01 42 fb
00000030: d6 2e bc 1e f3 65 73 63 5b 16 81 4b 97 38 b4 20
00000040: 5d 09 d9 b4 21 b4 0c f4 55 27 80 e7 4c cf 66 d0
00000050: 14 25 87 7c 20 84 68 d5 79 3a 74 1e e3 29 00 00
00000060: 08 00 00 40 00 2f 00 00 0c 00 00 40 01 00 00 00
00000070: 04 21 00 00 10 01 00 00 00 00 01 00 00 00 03 00
00000080: 00 2c 00 00 38 00 00 00 34 01 03 04 05 0a de 5f
00000090: cd 03 00 00 08 01 00 00 20 03 00 00 08 01 00 00
000000A0: 21 03 00 00 08 01 00 00 22 03 00 00 08 01 00 00
000000B0: 23 00 00 00 08 05 00 00 00 2d 00 00 28 02 00 00
000000C0: 00 07 01 00 10 08 00 08 00 0a 6f 0a ab 0a 6f 0a
000000D0: ab 07 00 00 10 00 00 ff ff 00 00 00 00 ff ff ff
000000E0: ff 29 00 00 28 02 00 00 00 07 01 00 10 08 00 08
000000F0: 00 0a 00 00 02 0a 00 00 02 07 00 00 10 00 00 ff
00000100: ff 0a 00 00 00 0a 00 00 ff 29 00 00 08 00 00 40
00000110: 0a 00 00 00 08 00 00 40 0b 00
```

(54) Parses received message

```
IKE SA Auth
E9D3F378191C3840.8DDFF401FBFB0B14.00000001 IKEv2 I->R[334]
  E[306]{
    IDi[21](FQDN){"IKE-Initiator"},
    AUTH[72](Preshared-Key){C99B01...741EE3},
    N[8](INITIAL_CONTACT),
    N[12](SET_WINDOW_SIZE){4},
    CP[16](REQUEST){IP4.Address[0], IP4.DNS[0]},
    SA[56]{
      P[52](#1:ESP:0ADE5FCD:5#){
        Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
                   ENCR_MAGMA_MGM_KTREE,
                   ENCR_KUZNYECHIK_MGM_MAC_KTREE,
                   ENCR_MAGMA_MGM_MAC_KTREE,
        ESN=Off}},
    TSi[40](2#){10.111.10.171:icmp:8.0, 0.0.0.0-255.255.255.255},
    TSr[40](2#){10.0.0.2:icmp:8.0, 10.0.0.0-10.0.0.255},
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(55) Computes prf(SK_pi, IDi)

```
00000000: 06 d3 d4 36 ab 5b 4f 41 d4 3d fc 79 1f 13 a3 89
00000010: e9 a6 6e d7 87 7d 72 d1 9d 71 78 2d 05 ee 47 fb
00000020: 82 c8 8f 86 cd b5 05 1d 25 7c 1e 79 18 ef 4e 4e
00000030: 8d ca f4 47 12 c6 7f 6a 32 7d d8 e8 f2 8e f8 33
```

(56) Uses PSK

```
00000000: e2 69 24 cf 15 32 93 47 3a 11 a4 97 a8 a4 5c b3
00000010: 4e 28 31 ef 0e 28 bb 77 69 69 c6 3c 68 bf e1 0d
```

(57) Computes prf(PSK,"Key Pad for IKEv2")

```
00000000: 01 3c a5 24 59 4e bc 78 99 20 61 6c 3f 03 e5 2e
00000010: 7a 75 2a 0b 78 36 bd 0a 89 ce 1d e7 8b 23 32 ae
00000020: 08 9a a0 03 1d da f6 14 8c 38 c6 bd 7c 03 13 24
00000030: bd af c8 ad 88 18 8f 41 d0 12 b9 e1 5a 66 8f 10
```

(58) Computes content of AUTH payload and compares it with the received one

```
00000000: c9 9b 01 9a 89 ee 56 53 ab 28 25 a1 d7 51 54 ac
00000010: 01 42 fb d6 2e bc 1e f3 65 73 63 5b 16 81 4b 97
00000020: 38 b4 20 5d 09 d9 b4 21 b4 0c f4 55 27 80 e7 4c
00000030: cf 66 d0 14 25 87 7c 20 84 68 d5 79 3a 74 1e e3
```

(59) Computes keys for ESP SAs

```
00000000: ff 42 3b a3 78 29 2b 10 52 c8 bf 06 fa ba 6d 5f
00000010: e2 db 51 1b 74 1b 54 ad 35 85 e3 cf 2b 77 52 42
00000020: bc 8c d8 ba dd f4 46 9e 89 41 5c d6
00000000: 8c eb 84 af 18 01 18 36 b7 8d 65 be 03 ca 69 64
00000010: 89 6e a8 91 03 bc 9a dc bd 49 10 ab 20 83 9f 83
00000020: b1 7c 45 9d ab d8 ab 6f de 6a 62 d1
```

(60) Computes prf(SK_pr,IDr)

```
00000000: 32 61 00 71 e8 1a d6 a1 12 8d ef 4e 2a e9 bb c2
00000010: 9f 3d ba 28 1b 2a a5 10 a2 ad c6 b1 73 07 c9 f1
00000020: 50 9e 1c d7 a5 85 8f a8 40 ef dd a7 ae 33 71 74
00000030: c8 8b a9 f4 3a 83 0f c1 c5 3c 9b 21 9f a9 58 25
```

(61) Uses PSK

```
00000000: e2 69 24 cf 15 32 93 47 3a 11 a4 97 a8 a4 5c b3
00000010: 4e 28 31 ef 0e 28 bb 77 69 69 c6 3c 68 bf e1 0d
```

(62) Computes prf(PSK,"Key Pad for IKEv2")

```
00000000: 01 3c a5 24 59 4e bc 78 99 20 61 6c 3f 03 e5 2e
00000010: 7a 75 2a 0b 78 36 bd 0a 89 ce 1d e7 8b 23 32 ae
00000020: 08 9a a0 03 1d da f6 14 8c 38 c6 bd 7c 03 13 24
00000030: bd af c8 ad 88 18 8f 41 d0 12 b9 e1 5a 66 8f 10
```

(63) Computes content of AUTH payload

```
00000000: 35 ce 8a ab dd 3d b1 5f 38 7b 2e c9 a6 24 7a 1f
00000010: a7 bb a0 6f b6 5e d8 81 07 d3 43 c8 a5 db 37 51
00000020: 0e 9d 9a 85 66 18 7a 0f 5c e2 1b fb 27 56 65 ed
00000030: 0e 41 fe ce 5e 95 bf 8a ae 57 f6 d6 26 d2 d1 2d
```

(64) Computes K1r (i1 = 0)

```
00000000: 61 cd ad b1 01 10 71 7c dc 18 81 1d 1f aa e3 13
00000010: 4b 07 f8 f7 49 a7 3d 0a 57 2f e1 61 bc ab 85 c4
```

(65) Computes K2r (i2 = 0)

```
00000000: 5f e7 47 77 da f7 54 d7 a8 e5 eb ed f9 82 c8 a9
00000010: 74 0c 54 77 6f eb b8 70 a4 43 43 3e c2 9e ce a6
```

(66) Computes K3r (i3 = 0)

```
00000000: e8 af 72 c4 c3 55 a2 6a fb ad 37 fd b4 b9 7f d6
00000010: f6 c8 cc 32 3f 50 32 40 06 86 ce 85 1b 02 28 f3
```

(67) Selects SPI for incoming ESP SA

```
00000000: 50 3c 8d af
```

(68) Creates message

```
IKE SA Auth
E9D3F378191C3840.8DDFF401FBFB0B14.00000001 IKEv2 I<=R[286]
  E[258]{
    IDr[21](FQDN){"IKE-Responder"},
    AUTH[72](Preshared-Key){35CE8A...D2D12D},
    N[8](INITIAL_CONTACT),
    N[12](SET_WINDOW_SIZE){64},
    CP[16](REPLY){IP4.Address[4]=10.1.1.2},
    SA[32]{
      P[28](#1:ESP:503C8DAF:2#){
        Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
        ESN=Off}},
    TSi[24](1#){10.1.1.2},
    TSr[24](1#){10.0.0.0-10.0.0.255},
    N[8](ADDITIONAL_TS_POSSIBLE),
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(69) Composes MGM nonce

```
00000000: 00 00 00 00 65 20 72 e7 0a 1e ff 7d da ba 17 31
```

(70) Composes AAD

```
00000000: e9 d3 f3 78 19 1c 38 40 8d df f4 01 fb fb 0b 14
00000010: 2e 20 23 20 00 00 00 01 00 00 01 1e 24 00 01 02
```

(71) Composes plaintext

```
00000000: 27 00 00 15 02 00 00 00 49 4b 45 2d 52 65 73 70
00000010: 6f 6e 64 65 72 29 00 00 48 02 00 00 00 35 ce 8a
00000020: ab dd 3d b1 5f 38 7b 2e c9 a6 24 7a 1f a7 bb a0
00000030: 6f b6 5e d8 81 07 d3 43 c8 a5 db 37 51 0e 9d 9a
00000040: 85 66 18 7a 0f 5c e2 1b fb 27 56 65 ed 0e 41 fe
00000050: ce 5e 95 bf 8a ae 57 f6 d6 26 d2 d1 2d 29 00 00
00000060: 08 00 00 40 00 2f 00 00 0c 00 00 40 01 00 00 00
00000070: 40 21 00 00 10 02 00 00 00 00 01 00 04 0a 01 01
00000080: 02 2c 00 00 20 00 00 00 1c 01 03 04 02 50 3c 8d
00000090: af 03 00 00 08 01 00 00 20 00 00 00 08 05 00 00
000000A0: 00 2d 00 00 18 01 00 00 00 07 00 00 10 00 00 ff
000000B0: ff 0a 01 01 02 0a 01 01 02 29 00 00 18 01 00 00
000000C0: 00 07 00 00 10 00 00 ff ff 0a 00 00 00 0a 00 00
000000D0: ff 29 00 00 08 00 00 40 02 29 00 00 08 00 00 40
000000E0: 0a 00 00 00 08 00 00 40 0b 00
```

(72) Encrypts plaintext using K3r as K_msg, resulting in ciphertext

```
00000000: 9b 5d 58 8a 99 44 11 d6 5b 93 7f 98 57 0d 0f 09
00000010: 0c a3 d9 36 41 b5 9c 91 94 17 3a cb 00 88 24 5e
00000020: 25 b7 0d 75 2f fb 4d d0 ab 2c cc 84 42 e7 f8 1b
00000030: 5a e6 88 13 9a 3e b1 03 79 31 0c 69 f6 17 a2 40
00000040: f8 aa 74 2e 62 29 ee 57 43 3f 10 bf 44 73 51 97
00000050: 2c 93 a4 02 87 3d 37 45 2c f1 3e 16 c3 d9 ec b3
00000060: b8 6f 66 1a f1 73 44 7c db 74 11 e6 07 4a 75 23
00000070: 83 df 00 52 ae 68 60 39 83 4c c3 b1 d5 7a e8 7f
00000080: 61 59 9e 4f 92 3c 2f 04 3b c3 ac e7 23 3f 1c a7
00000090: a5 3f 4d 33 1f 46 25 9f 09 5e f4 75 e0 12 32 5b
000000A0: 29 64 a4 40 1a b5 c9 cd 9e 8f 91 cc 5b 7d 14 15
000000B0: d0 89 70 e0 c6 d8 e4 e0 93 ff 02 4c 69 db ab 84
000000C0: d6 8f b9 f9 ed 07 aa 96 29 2a 50 c2 c4 b6 e5 cb
000000D0: 8e 16 33 7a 20 a4 3b 0e f2 53 9b b1 63 c0 46 4b
000000E0: d9 31 a8 98 f5 17 8a ff 0a c0
```

(73) Computes ICV using K3r as K_msg

```
00000000: 4a db a4 67 7e a1 3c 54 22 1f cf 62
```

(74) Composes IV

```
00000000: 00 00 00 00 00 00 00 00
```

(75) Sends message, peer receives message

```
       10.111.10.171:54294<-10.111.15.45:500 [286]

00000000: e9 d3 f3 78 19 1c 38 40 8d df f4 01 fb fb 0b 14
00000010: 2e 20 23 20 00 00 00 01 00 00 01 1e 24 00 01 02
00000020: 00 00 00 00 00 00 00 00 9b 5d 58 8a 99 44 11 d6
00000030: 5b 93 7f 98 57 0d 0f 09 0c a3 d9 36 41 b5 9c 91
00000040: 94 17 3a cb 00 88 24 5e 25 b7 0d 75 2f fb 4d d0
00000050: ab 2c cc 84 42 e7 f8 1b 5a e6 88 13 9a 3e b1 03
00000060: 79 31 0c 69 f6 17 a2 40 f8 aa 74 2e 62 29 ee 57
00000070: 43 3f 10 bf 44 73 51 97 2c 93 a4 02 87 3d 37 45
00000080: 2c f1 3e 16 c3 d9 ec b3 b8 6f 66 1a f1 73 44 7c
00000090: db 74 11 e6 07 4a 75 23 83 df 00 52 ae 68 60 39
000000A0: 83 4c c3 b1 d5 7a e8 7f 61 59 9e 4f 92 3c 2f 04
000000B0: 3b c3 ac e7 23 3f 1c a7 a5 3f 4d 33 1f 46 25 9f
000000C0: 09 5e f4 75 e0 12 32 5b 29 64 a4 40 1a b5 c9 cd
000000D0: 9e 8f 91 cc 5b 7d 14 15 d0 89 70 e0 c6 d8 e4 e0
000000E0: 93 ff 02 4c 69 db ab 84 d6 8f b9 f9 ed 07 aa 96
000000F0: 29 2a 50 c2 c4 b6 e5 cb 8e 16 33 7a 20 a4 3b 0e
00000100: f2 53 9b b1 63 c0 46 4b d9 31 a8 98 f5 17 8a ff
00000110: 0a c0 4a db a4 67 7e a1 3c 54 22 1f cf 62
```

Initiator's actions:

(76) Extracts IV from message

```
00000000: 00 00 00 00 00 00 00 00
```

(77) Computes K1r (i1 = 0)

```
00000000: 61 cd ad b1 01 10 71 7c dc 18 81 1d 1f aa e3 13
00000010: 4b 07 f8 f7 49 a7 3d 0a 57 2f e1 61 bc ab 85 c4
```

(78) Computes K2r (i2 = 0)

```
00000000: 5f e7 47 77 da f7 54 d7 a8 e5 eb ed f9 82 c8 a9
00000010: 74 0c 54 77 6f eb b8 70 a4 43 43 3e c2 9e ce a6
```

(79) Computes K3r (i3 = 0)

```
00000000: e8 af 72 c4 c3 55 a2 6a fb ad 37 fd b4 b9 7f d6
00000010: f6 c8 cc 32 3f 50 32 40 06 86 ce 85 1b 02 28 f3
```

(80) Composes MGM nonce

```
00000000: 00 00 00 00 65 20 72 e7 0a 1e ff 7d da ba 17 31
```

(81) Extracts ICV from message

```
00000000: 4a db a4 67 7e a1 3c 54 22 1f cf 62
```

(82) Extracts AAD from message

```
00000000: e9 d3 f3 78 19 1c 38 40 8d df f4 01 fb fb 0b 14
00000010: 2e 20 23 20 00 00 00 01 00 00 01 1e 24 00 01 02
```

(83) Extracts ciphertext from message

```
00000000: 9b 5d 58 8a 99 44 11 d6 5b 93 7f 98 57 0d 0f 09
00000010: 0c a3 d9 36 41 b5 9c 91 94 17 3a cb 00 88 24 5e
00000020: 25 b7 0d 75 2f fb 4d d0 ab 2c cc 84 42 e7 f8 1b
00000030: 5a e6 88 13 9a 3e b1 03 79 31 0c 69 f6 17 a2 40
00000040: f8 aa 74 2e 62 29 ee 57 43 3f 10 bf 44 73 51 97
00000050: 2c 93 a4 02 87 3d 37 45 2c f1 3e 16 c3 d9 ec b3
00000060: b8 6f 66 1a f1 73 44 7c db 74 11 e6 07 4a 75 23
00000070: 83 df 00 52 ae 68 60 39 83 4c c3 b1 d5 7a e8 7f
00000080: 61 59 9e 4f 92 3c 2f 04 3b c3 ac e7 23 3f 1c a7
00000090: a5 3f 4d 33 1f 46 25 9f 09 5e f4 75 e0 12 32 5b
000000A0: 29 64 a4 40 1a b5 c9 cd 9e 8f 91 cc 5b 7d 14 15
000000B0: d0 89 70 e0 c6 d8 e4 e0 93 ff 02 4c 69 db ab 84
000000C0: d6 8f b9 f9 ed 07 aa 96 29 2a 50 c2 c4 b6 e5 cb
000000D0: 8e 16 33 7a 20 a4 3b 0e f2 53 9b b1 63 c0 46 4b
000000E0: d9 31 a8 98 f5 17 8a ff 0a c0
```

(84) Decrypts ciphertext and verifies ICV using K3r as K_msg, resulting in plaintext

```
00000000: 27 00 00 15 02 00 00 00 49 4b 45 2d 52 65 73 70
00000010: 6f 6e 64 65 72 29 00 00 48 02 00 00 00 35 ce 8a
00000020: ab dd 3d b1 5f 38 7b 2e c9 a6 24 7a 1f a7 bb a0
00000030: 6f b6 5e d8 81 07 d3 43 c8 a5 db 37 51 0e 9d 9a
00000040: 85 66 18 7a 0f 5c e2 1b fb 27 56 65 ed 0e 41 fe
00000050: ce 5e 95 bf 8a ae 57 f6 d6 26 d2 d1 2d 29 00 00
00000060: 08 00 00 40 00 2f 00 00 0c 00 00 40 01 00 00 00
00000070: 40 21 00 00 10 02 00 00 00 00 01 00 04 0a 01 01
00000080: 02 2c 00 00 20 00 00 00 1c 01 03 04 02 50 3c 8d
00000090: af 03 00 00 08 01 00 00 20 00 00 00 08 05 00 00
000000A0: 00 2d 00 00 18 01 00 00 00 07 00 00 10 00 00 ff
000000B0: ff 0a 01 01 02 0a 01 01 02 29 00 00 18 01 00 00
000000C0: 00 07 00 00 10 00 00 ff ff 0a 00 00 00 0a 00 00
000000D0: ff 29 00 00 08 00 00 40 02 29 00 00 08 00 00 40
000000E0: 0a 00 00 00 08 00 00 40 0b 00
```

(85) Parses received message

```
IKE SA Auth
E9D3F378191C3840.8DDFF401FBFB0B14.00000001 IKEv2 R=>I[286]
  E[258]{
    IDr[21](FQDN){"IKE-Responder"},
    AUTH[72](Preshared-Key){35CE8A...D2D12D},
    N[8](INITIAL_CONTACT),
    N[12](SET_WINDOW_SIZE){64},
    CP[16](REPLY){IP4.Address[4]=10.1.1.2},
    SA[32]{
      P[28](#1:ESP:503C8DAF:2#){
        Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
        ESN=Off}},
    TSi[24](1#){10.1.1.2},
    TSr[24](1#){10.0.0.0-10.0.0.255},
    N[8](ADDITIONAL_TS_POSSIBLE),
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(86) Computes prf(SK_pr, IDr)

```
00000000: 32 61 00 71 e8 1a d6 a1 12 8d ef 4e 2a e9 bb c2
00000010: 9f 3d ba 28 1b 2a a5 10 a2 ad c6 b1 73 07 c9 f1
00000020: 50 9e 1c d7 a5 85 8f a8 40 ef dd a7 ae 33 71 74
00000030: c8 8b a9 f4 3a 83 0f c1 c5 3c 9b 21 9f a9 58 25
```

(87) Uses PSK

```
00000000: e2 69 24 cf 15 32 93 47 3a 11 a4 97 a8 a4 5c b3
00000010: 4e 28 31 ef 0e 28 bb 77 69 69 c6 3c 68 bf e1 0d
```

(88) Computes prf(PSK,"Key Pad for IKEv2")

```
00000000: 01 3c a5 24 59 4e bc 78 99 20 61 6c 3f 03 e5 2e
00000010: 7a 75 2a 0b 78 36 bd 0a 89 ce 1d e7 8b 23 32 ae
00000020: 08 9a a0 03 1d da f6 14 8c 38 c6 bd 7c 03 13 24
00000030: bd af c8 ad 88 18 8f 41 d0 12 b9 e1 5a 66 8f 10
```

(89) Computes content of AUTH payload and compares it with the received one

```
00000000: 35 ce 8a ab dd 3d b1 5f 38 7b 2e c9 a6 24 7a 1f
00000010: a7 bb a0 6f b6 5e d8 81 07 d3 43 c8 a5 db 37 51
00000020: 0e 9d 9a 85 66 18 7a 0f 5c e2 1b fb 27 56 65 ed
00000030: 0e 41 fe ce 5e 95 bf 8a ae 57 f6 d6 26 d2 d1 2d
```

(90) Computes keys for ESP SAs

```
00000000: ff 42 3b a3 78 29 2b 10 52 c8 bf 06 fa ba 6d 5f
00000010: e2 db 51 1b 74 1b 54 ad 35 85 e3 cf 2b 77 52 42
00000020: bc 8c d8 ba dd f4 46 9e 89 41 5c d6
00000000: 8c eb 84 af 18 01 18 36 b7 8d 65 be 03 ca 69 64
00000010: 89 6e a8 91 03 bc 9a dc bd 49 10 ab 20 83 9f 83
00000020: b1 7c 45 9d ab d8 ab 6f de 6a 62 d1
```

A.1.2. Sub-Scenario 2: IKE SA Rekeying Using the CREATE_CHILD_SA Exchange

```
Initiator                                Responder

HDR, SK {SAi, Ni, KEi [,N+]}  --->
                              <---    HDR, SK {SAr, Nr, KEr [,N+]}
```

Initiator's actions:

(1) Generates random SPIi for new IKE SA

```
00000000: 43 87 64 8d 6c 9e 28 ff
```

(2) Generates random IKE nonce Ni

```
00000000: 6c 83 67 41 1b 45 94 1d 79 94 51 2d 3f 7d 1e ce
00000010: 06 76 a6 09 cc a9 3a 8f f8 17 81 ff 28 08 5a 4c
```

(3) Generates ephemeral private key

```
00000000: cf 8f f0 df 04 24 43 b5 7e 15 2c bd 9f cd bd d9
00000010: 20 b5 35 7c e8 8b a6 d7 bd 7f 32 39 3d 5e 9a 3c
00000020: eb 88 4f 7f 6c 5d 03 05 fc bf 08 12 41 76 f4 a6
00000030: 2e 4c f7 ce 55 18 9d 6a 54 1f f7 57 46 23 cd 26
```

(4) Computes public key

```
00000000: 04 db 0b d3 9a ac 83 f3 e9 9d a9 11 c3 12 f6 df
00000010: f6 ae 99 38 55 20 1f 83 c8 28 ed 14 f9 68 88 77
00000020: ac 78 36 41 7a d7 93 a7 ee 4c 6a d7 f2 50 24 f5
00000030: a8 7b 03 28 22 9f a4 66 11 20 57 64 56 7c 36 3c
00000040: 72 c7 91 0a 1c fd 64 54 f1 17 97 6a 35 48 dc 8f
00000050: 85 97 20 12 2f 35 55 58 9b ca 7a 84 f3 01 cf ca
00000060: 78 e7 41 87 d3 3f 0f 2b 6d 78 59 ad f2 f2 c2 97
00000070: db 0b 75 6e 00 38 a2 72 8d 17 6b 44 f9 8b 95 66
```

(5) Creates message

```
Create Child SA
E9D3F378191C3840.8DDFF401FBFB0B14.00000002 IKEv2 R<-I [281]
  E[253]{
    SA[44]{
      P[40](#1:IKE:4387648D6C9E28FF:3#){
        Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
        PRF=PRF_HMAC_STREEBOG_512,
        KE=GOST3410_2012_512}},
    NONCE[36]{6C8367...085A4C},
    KE[136](GOST3410_2012_512){04DB0B...8B9566},
    N[12](SET_WINDOW_SIZE){4}}
```

(6) Uses previously computed key K3i

```
00000000: 36 ff fa db 84 a9 f1 21 d5 84 16 db eb af 21 a2
00000010: 12 6d 5c 35 95 fe 89 cf 27 47 52 8a b7 36 92 d4
```

(7) Composes MGM nonce

```
00000000: 00 00 00 01 83 00 37 c3 08 01 7e c3 0a 71 62 01
```

(8) Composes AAD

```
00000000: e9 d3 f3 78 19 1c 38 40 8d df f4 01 fb fb 0b 14
00000010: 2e 20 24 08 00 00 00 02 00 00 01 19 21 00 00 fd
```

(9) Composes plaintext

```
00000000: 28 00 00 2c 00 00 00 28 01 01 08 03 43 87 64 8d
00000010: 6c 9e 28 ff 03 00 00 08 01 00 00 20 03 00 00 08
00000020: 02 00 00 09 00 00 00 08 04 00 00 22 22 00 00 24
00000030: 6c 83 67 41 1b 45 94 1d 79 94 51 2d 3f 7d 1e ce
00000040: 06 76 a6 09 cc a9 3a 8f f8 17 81 ff 28 08 5a 4c
00000050: 29 00 00 88 00 22 00 00 04 db 0b d3 9a ac 83 f3
00000060: e9 9d a9 11 c3 12 f6 df f6 ae 99 38 55 20 1f 83
00000070: c8 28 ed 14 f9 68 88 77 ac 78 36 41 7a d7 93 a7
00000080: ee 4c 6a d7 f2 50 24 f5 a8 7b 03 28 22 9f a4 66
00000090: 11 20 57 64 56 7c 36 3c 72 c7 91 0a 1c fd 64 54
000000A0: f1 17 97 6a 35 48 dc 8f 85 97 20 12 2f 35 55 58
000000B0: 9b ca 7a 84 f3 01 cf ca 78 e7 41 87 d3 3f 0f 2b
000000C0: 6d 78 59 ad f2 f2 c2 97 db 0b 75 6e 00 38 a2 72
000000D0: 8d 17 6b 44 f9 8b 95 66 00 00 00 0c 00 00 40 01
000000E0: 00 00 00 04 00
```

(10) Encrypts plaintext using K3i as K_msg, resulting in ciphertext

```
00000000: 00 16 cf 92 8a 87 4c 02 79 31 04 22 c3 d9 5f fd
00000010: 5a 19 23 62 25 d1 99 c2 af 75 4d f1 3c ac c0 c1
00000020: c7 db d0 fd 93 ac 6d 25 b4 19 01 e6 df e8 51 c2
00000030: 88 a9 8a 26 92 98 ec ce c1 2f cf ca ce 9b 5a 6d
00000040: 4c 8b cf 97 63 5a a3 e6 46 49 0f 1f 05 54 00 49
00000050: 6b d8 14 f4 e2 ee b3 66 2a 13 9b dd 63 53 7a 82
00000060: 2a d8 bf 48 aa db 79 21 d3 d8 ac b1 ac 8f 9b 41
00000070: a7 49 81 95 d7 54 46 e2 00 9b 17 3a ab 9a 4c 8f
00000080: 19 9e ac 61 cc f6 02 47 a1 7e f4 48 5b e7 3c a7
00000090: 53 dc 03 9e ea 5f c4 99 60 6e db 6a 21 fe 7c 7b
000000A0: 11 ed bf 44 59 73 fa 65 01 98 e4 e6 10 63 87 27
000000B0: 8b f0 8c bb 94 52 dd 97 ee dc ce 88 c4 45 b4 16
000000C0: f2 8b d4 74 cb 46 38 57 f4 44 88 23 44 06 d9 91
000000D0: 00 ea 81 2c e7 f6 66 0f a8 45 0f 1d 8c 2d f1 02
000000E0: a2 06 78 c7 e0
```

(11) Computes ICV using K3i as K_msg

```
00000000: b1 2f da a5 96 fa 27 ee 67 de 9e 95
```

(12) Composes IV

```
00000000: 00 00 00 00 00 00 00 01
```

(13) Sends message, peer receives message

```
       10.111.10.171:54294->10.111.15.45:500 [281]

00000000: e9 d3 f3 78 19 1c 38 40 8d df f4 01 fb fb 0b 14
00000010: 2e 20 24 08 00 00 00 02 00 00 01 19 21 00 00 fd
00000020: 00 00 00 00 00 00 00 01 00 16 cf 92 8a 87 4c 02
00000030: 79 31 04 22 c3 d9 5f fd 5a 19 23 62 25 d1 99 c2
00000040: af 75 4d f1 3c ac c0 c1 c7 db d0 fd 93 ac 6d 25
00000050: b4 19 01 e6 df e8 51 c2 88 a9 8a 26 92 98 ec ce
00000060: c1 2f cf ca ce 9b 5a 6d 4c 8b cf 97 63 5a a3 e6
00000070: 46 49 0f 1f 05 54 00 49 6b d8 14 f4 e2 ee b3 66
00000080: 2a 13 9b dd 63 53 7a 82 2a d8 bf 48 aa db 79 21
00000090: d3 d8 ac b1 ac 8f 9b 41 a7 49 81 95 d7 54 46 e2
000000A0: 00 9b 17 3a ab 9a 4c 8f 19 9e ac 61 cc f6 02 47
000000B0: a1 7e f4 48 5b e7 3c a7 53 dc 03 9e ea 5f c4 99
000000C0: 60 6e db 6a 21 fe 7c 7b 11 ed bf 44 59 73 fa 65
000000D0: 01 98 e4 e6 10 63 87 27 8b f0 8c bb 94 52 dd 97
000000E0: ee dc ce 88 c4 45 b4 16 f2 8b d4 74 cb 46 38 57
000000F0: f4 44 88 23 44 06 d9 91 00 ea 81 2c e7 f6 66 0f
00000100: a8 45 0f 1d 8c 2d f1 02 a2 06 78 c7 e0 b1 2f da
00000110: a5 96 fa 27 ee 67 de 9e 95
```

Responder's actions:

(14) Extracts IV from message

```
00000000: 00 00 00 00 00 00 00 01
```

(15) Uses previously computed key K3i

```
00000000: 36 ff fa db 84 a9 f1 21 d5 84 16 db eb af 21 a2
00000010: 12 6d 5c 35 95 fe 89 cf 27 47 52 8a b7 36 92 d4
```

(16) Composes MGM nonce

```
00000000: 00 00 00 01 83 00 37 c3 08 01 7e c3 0a 71 62 01
```

(17) Extracts ICV from message

```
00000000: b1 2f da a5 96 fa 27 ee 67 de 9e 95
```

(18) Extracts AAD from message

```
00000000: e9 d3 f3 78 19 1c 38 40 8d df f4 01 fb fb 0b 14
00000010: 2e 20 24 08 00 00 00 02 00 00 01 19 21 00 00 fd
```

(19) Extracts ciphertext from message

```
00000000: 00 16 cf 92 8a 87 4c 02 79 31 04 22 c3 d9 5f fd
00000010: 5a 19 23 62 25 d1 99 c2 af 75 4d f1 3c ac c0 c1
00000020: c7 db d0 fd 93 ac 6d 25 b4 19 01 e6 df e8 51 c2
00000030: 88 a9 8a 26 92 98 ec ce c1 2f cf ca ce 9b 5a 6d
00000040: 4c 8b cf 97 63 5a a3 e6 46 49 0f 1f 05 54 00 49
00000050: 6b d8 14 f4 e2 ee b3 66 2a 13 9b dd 63 53 7a 82
00000060: 2a d8 bf 48 aa db 79 21 d3 d8 ac b1 ac 8f 9b 41
00000070: a7 49 81 95 d7 54 46 e2 00 9b 17 3a ab 9a 4c 8f
00000080: 19 9e ac 61 cc f6 02 47 a1 7e f4 48 5b e7 3c a7
00000090: 53 dc 03 9e ea 5f c4 99 60 6e db 6a 21 fe 7c 7b
000000A0: 11 ed bf 44 59 73 fa 65 01 98 e4 e6 10 63 87 27
000000B0: 8b f0 8c bb 94 52 dd 97 ee dc ce 88 c4 45 b4 16
000000C0: f2 8b d4 74 cb 46 38 57 f4 44 88 23 44 06 d9 91
000000D0: 00 ea 81 2c e7 f6 66 0f a8 45 0f 1d 8c 2d f1 02
000000E0: a2 06 78 c7 e0
```

(20) Decrypts ciphertext and verifies ICV using K3i as K_msg, resulting in plaintext

```
00000000: 28 00 00 2c 00 00 00 28 01 01 08 03 43 87 64 8d
00000010: 6c 9e 28 ff 03 00 00 08 01 00 00 20 03 00 00 08
00000020: 02 00 00 09 00 00 00 08 04 00 00 22 22 00 00 24
00000030: 6c 83 67 41 1b 45 94 1d 79 94 51 2d 3f 7d 1e ce
00000040: 06 76 a6 09 cc a9 3a 8f f8 17 81 ff 28 08 5a 4c
00000050: 29 00 00 88 00 22 00 00 04 db 0b d3 9a ac 83 f3
00000060: e9 9d a9 11 c3 12 f6 df f6 ae 99 38 55 20 1f 83
00000070: c8 28 ed 14 f9 68 88 77 ac 78 36 41 7a d7 93 a7
00000080: ee 4c 6a d7 f2 50 24 f5 a8 7b 03 28 22 9f a4 66
00000090: 11 20 57 64 56 7c 36 3c 72 c7 91 0a 1c fd 64 54
000000A0: f1 17 97 6a 35 48 dc 8f 85 97 20 12 2f 35 55 58
000000B0: 9b ca 7a 84 f3 01 cf ca 78 e7 41 87 d3 3f 0f 2b
000000C0: 6d 78 59 ad f2 f2 c2 97 db 0b 75 6e 00 38 a2 72
000000D0: 8d 17 6b 44 f9 8b 95 66 00 00 00 0c 00 00 40 01
000000E0: 00 00 00 04 00
```

(21) Parses received message

```
Create Child SA
E9D3F378191C3840.8DDFF401FBFB0B14.00000002 IKEv2 I->R[281]
  E[253]{
    SA[44]{
      P[40](#1:IKE:4387648D6C9E28FF:3#){
        Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
        PRF=PRF_HMAC_STREEBOG_512,
        KE=GOST3410_2012_512}},
    NONCE[36]{6C8367...085A4C},
    KE[136](GOST3410_2012_512){04DB0B...8B9566},
    N[12](SET_WINDOW_SIZE){4}}
```

(22) Generates random SPIr for new IKE SA

```
00000000: 82 d9 fa f8 74 49 b9 36
```

(23) Generates random IKE nonce Nr

```
00000000: 5a 2d d2 68 c6 85 5d 32 d4 7b 0b 8e ae 7d c9 81
00000010: be 3e 69 c1 bb f5 ae 89 55 59 c7 48 bc 96 43 7b
```

(24) Generates ephemeral private key

```
00000000: b9 ea c6 c1 84 db 39 54 e3 e7 74 be 02 e0 c9 0b
00000010: 5c b9 72 03 d4 fc a2 3f b6 cf 71 8d 4f f4 b4 c5
00000020: 21 1c 93 f9 86 cc 6b cb db ff 78 51 5b b6 48 e8
00000030: 44 ce c0 83 c9 d0 b8 90 08 94 db 29 9f bb c2 1a
```

(25) Computes public key

```
00000000: b9 f9 27 a8 96 70 7a 03 58 c2 39 58 63 2d 50 20
00000010: bf 69 c0 1d a6 de d4 4d 65 aa 26 c6 8f 9f e9 e9
00000020: 4b bb da 1d 2f d3 60 2d 18 33 04 9b b2 25 a6 07
00000030: ac 58 1b fc 3c 5b 1e f3 4b c0 f9 cb 90 14 c6 80
00000040: 6e c3 73 c1 4a f7 5c 27 dd 2a e1 ba 94 9c f7 06
00000050: 68 92 19 8e 85 67 f9 d2 d1 ea 3c 16 16 b9 3f 0c
00000060: 8b 2d 2e d6 20 14 7e 27 18 d3 23 9e 2a 99 41 40
00000070: 6a 41 c5 3f 79 9c a7 22 79 15 98 1d 98 b5 ac 4a
```

(26) Computes shared key

```
00000000: dd e7 44 39 1c d9 66 cf d2 24 a4 bb 0a 57 b3 3e
00000010: 1a 8f 5d 07 11 4d c3 47 87 1a 13 ec 84 26 03 f8
00000020: ea 93 5a f5 23 a3 45 71 ff 5f f2 3d 59 43 3a 5e
00000030: eb 5e 79 fa 0e 62 9e bc af ca e4 ee 7a 81 3a 84
```

(27) Computes SKEYSEED for new SA

```
00000000: ec 5f 4f 15 ce d7 7d 2f 12 fb a1 df 5f 44 aa 88
00000010: 6a ef 45 e4 04 97 86 95 15 1b 3c ac 31 cc 57 a3
00000020: f0 f4 92 89 33 00 76 2b e9 fd 8b c2 ed 8b e7 36
00000030: cb 17 59 55 9e cc 22 14 72 a5 79 27 27 1d 06 62
```

(28) Computes SK_d for new SA

```
00000000: 08 58 14 7d eb c9 41 7f 7f a2 86 66 bf d4 76 37
00000010: 04 27 4e bc 5d 63 f7 07 79 62 69 7a 69 3c da 7a
00000020: d5 4d 6f 08 1e 14 51 66 2f 94 0d bd 29 45 9c b0
00000030: 51 26 09 4b 47 52 ba 19 98 a5 c2 65 af 84 a1 34
```

(29) Computes SK_ei for new SA

```
00000000: 18 0a 4f 98 7d a4 21 6c 68 84 94 1f d9 28 49 b9
00000010: 05 30 f8 aa 43 02 7e 0d aa d3 27 e9 8c 9a 39 9a
00000020: 03 a0 05 b7 b2 2d f9 90 bb 6c ff ca
```

(30) Computes SK_er for new SA

```
00000000: 47 dc aa 71 4a 8b 66 13 d8 09 79 c7 8c 72 0a 78
00000010: 06 48 6d 4f 1f 53 3a 91 1d b7 2c 86 f5 f1 4e 00
00000020: 84 57 87 2b 38 70 63 27 8c dd 88 78
```

(31) Creates message

```
Create Child SA
E9D3F378191C3840.8DDFF401FBFB0B14.00000002 IKEv2 I<=R[281]
  E[253]{
    SA[44]{
      P[40](#1:IKE:82D9FAF87449B936:3#){
        Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
        PRF=PRF_HMAC_STREEBOG_512,
        KE=GOST3410_2012_512}},
    NONCE[36]{5A2DD2...96437B},
    KE[136](GOST3410_2012_512){B9F927...B5AC4A},
    N[12](SET_WINDOW_SIZE){64}}
```

(32) Uses previously computed key K3r

```
00000000: e8 af 72 c4 c3 55 a2 6a fb ad 37 fd b4 b9 7f d6
00000010: f6 c8 cc 32 3f 50 32 40 06 86 ce 85 1b 02 28 f3
```

(33) Composes MGM nonce

```
00000000: 00 00 00 01 65 20 72 e7 0a 1e ff 7d da ba 17 31
```

(34) Composes AAD

```
00000000: e9 d3 f3 78 19 1c 38 40 8d df f4 01 fb fb 0b 14
00000010: 2e 20 24 20 00 00 00 02 00 00 01 19 21 00 00 fd
```

(35) Composes plaintext

```
00000000: 28 00 00 2c 00 00 00 28 01 01 08 03 82 d9 fa f8
00000010: 74 49 b9 36 03 00 00 08 01 00 00 20 03 00 00 08
00000020: 02 00 00 09 00 00 00 08 04 00 00 22 22 00 00 24
00000030: 5a 2d d2 68 c6 85 5d 32 d4 7b 0b 8e ae 7d c9 81
00000040: be 3e 69 c1 bb f5 ae 89 55 59 c7 48 bc 96 43 7b
00000050: 29 00 00 88 00 22 00 00 b9 f9 27 a8 96 70 7a 03
00000060: 58 c2 39 58 63 2d 50 20 bf 69 c0 1d a6 de d4 4d
00000070: 65 aa 26 c6 8f 9f e9 e9 4b bb da 1d 2f d3 60 2d
00000080: 18 33 04 9b b2 25 a6 07 ac 58 1b fc 3c 5b 1e f3
00000090: 4b c0 f9 cb 90 14 c6 80 6e c3 73 c1 4a f7 5c 27
000000A0: dd 2a e1 ba 94 9c f7 06 68 92 19 8e 85 67 f9 d2
000000B0: d1 ea 3c 16 16 b9 3f 0c 8b 2d 2e d6 20 14 7e 27
000000C0: 18 d3 23 9e 2a 99 41 40 6a 41 c5 3f 79 9c a7 22
000000D0: 79 15 98 1d 98 b5 ac 4a 00 00 00 0c 00 00 40 01
000000E0: 00 00 00 40 00
```

(36) Encrypts plaintext using K3r as K_msg, resulting in ciphertext

```
00000000: fd ee 4c 8f 78 ff b6 0c fc 65 bb ef db 53 56 a2
00000010: d3 2d 4f 59 ff 28 38 eb 76 0b 40 5e 8d 52 e8 c1
00000020: b9 75 22 b4 bb 71 8f 16 3a 97 0e 4d 95 ef bc 84
00000030: 46 c6 77 1e 4b 14 73 46 89 ed d4 b4 54 a2 64 19
00000040: 67 b2 98 7e 8b d4 45 31 17 1e e4 ae f4 24 44 42
00000050: dd 55 a0 49 fe 08 59 d0 a1 16 69 60 8a 8e 54 d2
00000060: 02 6d ae 17 5f 32 bf 14 78 f0 86 47 26 bf fb 6b
00000070: 7c 17 f7 f5 62 b6 d6 a0 e5 f3 c2 af b5 28 ee d0
00000080: 9b 22 8c e6 d0 58 4d 48 18 6d dd 3e 4e 33 66 ac
00000090: a2 29 1f 3b 62 4a e6 4a 8c 98 18 8b 21 73 a5 88
000000A0: 49 09 3b 27 88 20 40 6b a5 fc 08 37 c7 ac c9 0f
000000B0: 5d 69 87 7c 37 c8 c7 fd d8 72 6d ad ac 22 27 ca
000000C0: 93 d6 bd 6a 55 2a 1a 8b 2e 84 b4 0a 35 d3 ac d5
000000D0: 99 c9 ac d5 6f 03 94 bf ca f5 53 e5 a5 74 57 de
000000E0: 6a 5a 26 b8 e4
```

(37) Computes ICV using K3r as K_msg

```
00000000: 04 2f 99 3f 02 19 56 c4 0d 0b 7a 45
```

(38) Composes IV

```
00000000: 00 00 00 00 00 00 00 01
```

(39) Sends message, peer receives message

```
       10.111.10.171:54294<-10.111.15.45:500 [281]

00000000: e9 d3 f3 78 19 1c 38 40 8d df f4 01 fb fb 0b 14
00000010: 2e 20 24 20 00 00 00 02 00 00 01 19 21 00 00 fd
00000020: 00 00 00 00 00 00 00 01 fd ee 4c 8f 78 ff b6 0c
00000030: fc 65 bb ef db 53 56 a2 d3 2d 4f 59 ff 28 38 eb
00000040: 76 0b 40 5e 8d 52 e8 c1 b9 75 22 b4 bb 71 8f 16
00000050: 3a 97 0e 4d 95 ef bc 84 46 c6 77 1e 4b 14 73 46
00000060: 89 ed d4 b4 54 a2 64 19 67 b2 98 7e 8b d4 45 31
00000070: 17 1e e4 ae f4 24 44 42 dd 55 a0 49 fe 08 59 d0
00000080: a1 16 69 60 8a 8e 54 d2 02 6d ae 17 5f 32 bf 14
00000090: 78 f0 86 47 26 bf fb 6b 7c 17 f7 f5 62 b6 d6 a0
000000A0: e5 f3 c2 af b5 28 ee d0 9b 22 8c e6 d0 58 4d 48
000000B0: 18 6d dd 3e 4e 33 66 ac a2 29 1f 3b 62 4a e6 4a
000000C0: 8c 98 18 8b 21 73 a5 88 49 09 3b 27 88 20 40 6b
000000D0: a5 fc 08 37 c7 ac c9 0f 5d 69 87 7c 37 c8 c7 fd
000000E0: d8 72 6d ad ac 22 27 ca 93 d6 bd 6a 55 2a 1a 8b
000000F0: 2e 84 b4 0a 35 d3 ac d5 99 c9 ac d5 6f 03 94 bf
00000100: ca f5 53 e5 a5 74 57 de 6a 5a 26 b8 e4 04 2f 99
00000110: 3f 02 19 56 c4 0d 0b 7a 45
```

Initiator's actions:

(40) Extracts IV from message

```
00000000: 00 00 00 00 00 00 00 01
```

(41) Uses previously computed key K3r

```
00000000: e8 af 72 c4 c3 55 a2 6a fb ad 37 fd b4 b9 7f d6
00000010: f6 c8 cc 32 3f 50 32 40 06 86 ce 85 1b 02 28 f3
```

(42) Composes MGM nonce

```
00000000: 00 00 00 01 65 20 72 e7 0a 1e ff 7d da ba 17 31
```

(43) Extracts ICV from message

```
00000000: 04 2f 99 3f 02 19 56 c4 0d 0b 7a 45
```

(44) Extracts AAD from message

```
00000000: e9 d3 f3 78 19 1c 38 40 8d df f4 01 fb fb 0b 14
00000010: 2e 20 24 20 00 00 00 02 00 00 01 19 21 00 00 fd
```

(45) Extracts ciphertext from message

```
00000000: fd ee 4c 8f 78 ff b6 0c fc 65 bb ef db 53 56 a2
00000010: d3 2d 4f 59 ff 28 38 eb 76 0b 40 5e 8d 52 e8 c1
00000020: b9 75 22 b4 bb 71 8f 16 3a 97 0e 4d 95 ef bc 84
00000030: 46 c6 77 1e 4b 14 73 46 89 ed d4 b4 54 a2 64 19
00000040: 67 b2 98 7e 8b d4 45 31 17 1e e4 ae f4 24 44 42
00000050: dd 55 a0 49 fe 08 59 d0 a1 16 69 60 8a 8e 54 d2
00000060: 02 6d ae 17 5f 32 bf 14 78 f0 86 47 26 bf fb 6b
00000070: 7c 17 f7 f5 62 b6 d6 a0 e5 f3 c2 af b5 28 ee d0
00000080: 9b 22 8c e6 d0 58 4d 48 18 6d dd 3e 4e 33 66 ac
00000090: a2 29 1f 3b 62 4a e6 4a 8c 98 18 8b 21 73 a5 88
000000A0: 49 09 3b 27 88 20 40 6b a5 fc 08 37 c7 ac c9 0f
000000B0: 5d 69 87 7c 37 c8 c7 fd d8 72 6d ad ac 22 27 ca
000000C0: 93 d6 bd 6a 55 2a 1a 8b 2e 84 b4 0a 35 d3 ac d5
000000D0: 99 c9 ac d5 6f 03 94 bf ca f5 53 e5 a5 74 57 de
000000E0: 6a 5a 26 b8 e4
```

(46) Decrypts ciphertext and verifies ICV using K3r as K_msg, resulting in plaintext

```
00000000: 28 00 00 2c 00 00 00 28 01 01 08 03 82 d9 fa f8
00000010: 74 49 b9 36 03 00 00 08 01 00 00 20 03 00 00 08
00000020: 02 00 00 09 00 00 00 08 04 00 00 22 22 00 00 24
00000030: 5a 2d d2 68 c6 85 5d 32 d4 7b 0b 8e ae 7d c9 81
00000040: be 3e 69 c1 bb f5 ae 89 55 59 c7 48 bc 96 43 7b
00000050: 29 00 00 88 00 22 00 00 b9 f9 27 a8 96 70 7a 03
00000060: 58 c2 39 58 63 2d 50 20 bf 69 c0 1d a6 de d4 4d
00000070: 65 aa 26 c6 8f 9f e9 e9 4b bb da 1d 2f d3 60 2d
00000080: 18 33 04 9b b2 25 a6 07 ac 58 1b fc 3c 5b 1e f3
00000090: 4b c0 f9 cb 90 14 c6 80 6e c3 73 c1 4a f7 5c 27
000000A0: dd 2a e1 ba 94 9c f7 06 68 92 19 8e 85 67 f9 d2
000000B0: d1 ea 3c 16 16 b9 3f 0c 8b 2d 2e d6 20 14 7e 27
000000C0: 18 d3 23 9e 2a 99 41 40 6a 41 c5 3f 79 9c a7 22
000000D0: 79 15 98 1d 98 b5 ac 4a 00 00 00 0c 00 00 40 01
000000E0: 00 00 00 40 00
```

(47) Parses received message

```
Create Child SA
E9D3F378191C3840.8DDFF401FBFB0B14.00000002 IKEv2 R=>I[281]
  E[253]{
    SA[44]{
      P[40](#1:IKE:82D9FAF87449B936:3#){
        Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
        PRF=PRF_HMAC_STREEBOG_512,
        KE=GOST3410_2012_512}},
    NONCE[36]{5A2DD2...96437B},
    KE[136](GOST3410_2012_512){B9F927...B5AC4A},
    N[12](SET_WINDOW_SIZE){64}}
```

(48) Computes shared key

```
00000000: dd e7 44 39 1c d9 66 cf d2 24 a4 bb 0a 57 b3 3e
00000010: 1a 8f 5d 07 11 4d c3 47 87 1a 13 ec 84 26 03 f8
00000020: ea 93 5a f5 23 a3 45 71 ff 5f f2 3d 59 43 3a 5e
00000030: eb 5e 79 fa 0e 62 9e bc af ca e4 ee 7a 81 3a 84
```

(49) Computes SKEYSEED for new SA

```
00000000: ec 5f 4f 15 ce d7 7d 2f 12 fb a1 df 5f 44 aa 88
00000010: 6a ef 45 e4 04 97 86 95 15 1b 3c ac 31 cc 57 a3
00000020: f0 f4 92 89 33 00 76 2b e9 fd 8b c2 ed 8b e7 36
00000030: cb 17 59 55 9e cc 22 14 72 a5 79 27 27 1d 06 62
```

(50) Computes SK_d for new SA

```
00000000: 08 58 14 7d eb c9 41 7f 7f a2 86 66 bf d4 76 37
00000010: 04 27 4e bc 5d 63 f7 07 79 62 69 7a 69 3c da 7a
00000020: d5 4d 6f 08 1e 14 51 66 2f 94 0d bd 29 45 9c b0
00000030: 51 26 09 4b 47 52 ba 19 98 a5 c2 65 af 84 a1 34
```

(51) Computes SK_ei for new SA

```
00000000: 18 0a 4f 98 7d a4 21 6c 68 84 94 1f d9 28 49 b9
00000010: 05 30 f8 aa 43 02 7e 0d aa d3 27 e9 8c 9a 39 9a
00000020: 03 a0 05 b7 b2 2d f9 90 bb 6c ff ca
```

(52) Computes SK_er for new SA

```
00000000: 47 dc aa 71 4a 8b 66 13 d8 09 79 c7 8c 72 0a 78
00000010: 06 48 6d 4f 1f 53 3a 91 1d b7 2c 86 f5 f1 4e 00
00000020: 84 57 87 2b 38 70 63 27 8c dd 88 78
```

A.1.3. Sub-Scenario 3: ESP SAs Rekeying with PFS Using the CREATE_CHILD_SA Exchange

```
Initiator                             Responder

HDR, SK {N(REKEY_SA), SAi, Ni,
     KEi, TSi, TSr [,N+]}     --->
                              <---    HDR, SK {SAr, Nr,
                                           KEr, TSi, TSr [,N+]}
```

Initiator's actions:

(1) Generates random IKE nonce Ni

```
00000000: 59 52 b2 58 00 b7 d3 f9 c3 31 23 16 6f c2 d1 d7
00000010: 07 8b 99 fb 24 cf 24 30 a3 ce a6 fe d3 0f 20 9b
```

(2) Generates ephemeral private key

```
00000000: 2f b9 df 43 dc 50 f5 17 59 c0 c7 21 ac ca 03 7a
00000010: 55 87 f9 bb a6 5a 9e d4 46 98 15 c9 3a 6b 40 91
00000020: e6 99 f4 f2 e5 88 14 e7 d8 9f 98 b1 59 21 05 52
00000030: f0 b0 ce dc 8e c6 db 1f 9d a9 4a 6d 95 f2 cb 3d
```

(3) Computes public key

```
00000000: 1c 55 08 b9 01 f5 76 6a 01 27 97 2d 38 b1 4a 5c
00000010: b7 43 f1 64 24 ef 76 75 50 ce 4f 6f 59 ca 96 ae
00000020: 54 85 9c 94 8d 04 91 62 3a 0c b6 6e 77 59 81 40
00000030: 69 bf bb 80 f7 7c 29 ee 9f 9e 0c 83 b6 08 fc 43
00000040: b8 c6 66 36 e5 eb a0 43 c2 56 fa 52 f9 99 b6 95
00000050: 34 4c cd 49 1f c7 83 9e d7 d9 ca e3 a5 d0 3c aa
00000060: e8 ee ed 2c dd 5c 81 49 ab 3c d4 fa 15 4e 29 5f
00000070: 7c cd b2 f1 c1 d2 6f 8f a7 74 4d 6a d8 8a c3 60
```

(4) Selects SPI for new incoming ESP SA

```
00000000: a4 fe 65 a1
```

(5) Creates message

```
Create Child SA
4387648D6C9E28FF.82D9FAF87449B936.00000000 IKEv2 R<-I[341]
  E[313]{
    N[12](ESP:0ADE5FCD:REKEY_SA),
    SA[40]{
      P[36](#1:ESP:A4FE65A1:3#){
        Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
        KE=GOST3410_2012_512,
        ESN=Off}},
    NONCE[36]{5952B2...0F209B},
    KE[136](GOST3410_2012_512){1C5508...8AC360},
    TSi[24](1#){10.1.1.2},
    TSr[24](1#){10.0.0.0-10.0.0.255},
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(6) Computes K1i (i1 = 0)

```
00000000: 17 ec f1 84 33 9a c3 e3 93 e1 21 d7 65 3b 6c 83
00000010: d4 ae 9c 29 5b 12 cc b3 c5 0c 48 19 49 eb c0 ba
```

(7) Computes K2i (i2 = 0)

```
00000000: 2d 33 c0 55 87 f2 ee ce ac 1a f2 28 64 c6 f5 ad
00000010: de 2d be 7a a8 92 d0 a6 20 bc ef 25 29 7b 56 9f
```

(8) Computes K3i (i3 = 0)

```
00000000: c9 41 22 b5 39 b7 d2 3f c4 4d a6 ae 88 2e ff b4
00000010: f4 c0 90 9c bd bc 63 56 14 62 e8 8f 90 1a e7 eb
```

(9) Composes MGM nonce

```
00000000: 00 00 00 00 03 a0 05 b7 b2 2d f9 90 bb 6c ff ca
```

(10) Composes AAD

```
00000000: 43 87 64 8d 6c 9e 28 ff 82 d9 fa f8 74 49 b9 36
00000010: 2e 20 24 08 00 00 00 00 00 00 01 55 29 00 01 39
```

(11) Composes plaintext

```
00000000: 21 00 00 0c 03 04 40 09 0a de 5f cd 28 00 00 28
00000010: 00 00 00 24 01 03 04 03 a4 fe 65 a1 03 00 00 08
00000020: 01 00 00 20 03 00 00 08 04 00 00 22 00 00 00 08
00000030: 05 00 00 00 22 00 00 24 59 52 b2 58 00 b7 d3 f9
00000040: c3 31 23 16 6f c2 d1 d7 07 8b 99 fb 24 cf 24 30
00000050: a3 ce a6 fe d3 0f 20 9b 2c 00 00 88 00 22 00 00
00000060: 1c 55 08 b9 01 f5 76 6a 01 27 97 2d 38 b1 4a 5c
00000070: b7 43 f1 64 24 ef 76 75 50 ce 4f 6f 59 ca 96 ae
00000080: 54 85 9c 94 8d 04 91 62 3a 0c b6 6e 77 59 81 40
00000090: 69 bf bb 80 f7 7c 29 ee 9f 9e 0c 83 b6 08 fc 43
000000A0: b8 c6 66 36 e5 eb a0 43 c2 56 fa 52 f9 99 b6 95
000000B0: 34 4c cd 49 1f c7 83 9e d7 d9 ca e3 a5 d0 3c aa
000000C0: e8 ee ed 2c dd 5c 81 49 ab 3c d4 fa 15 4e 29 5f
000000D0: 7c cd b2 f1 c1 d2 6f 8f a7 74 4d 6a d8 8a c3 60
000000E0: 2d 00 00 18 01 00 00 00 07 00 00 10 00 00 ff ff
000000F0: 0a 01 01 02 0a 01 01 02 29 00 00 18 01 00 00 00
00000100: 07 00 00 10 00 00 ff ff 0a 00 00 00 0a 00 00 ff
00000110: 29 00 00 08 00 00 40 0a 00 00 00 08 00 00 40 0b
00000120: 00
```

(12) Encrypts plaintext using K3i as K_msg, resulting in ciphertext

```
00000000: 00 9b 13 cb cb f1 18 53 fc 81 2e 75 c3 03 e0 ca
00000010: 55 c1 fb 55 c0 29 40 48 fc 20 f4 a8 51 5b 97 6b
00000020: c6 07 4c 7d 45 54 51 0f 18 7f 43 a4 df 4b e8 e3
00000030: b4 eb 68 24 4b f0 1c df 8f 1e a2 21 31 02 29 68
00000040: 38 4d 68 fd 42 66 34 3e 82 46 f0 17 02 bf 65 19
00000050: b0 f7 09 62 0d 12 6a 7e ad 76 57 0d 19 55 cf 01
00000060: 89 9c 7e f5 5a fa 20 4f 8c 6d a4 83 b9 94 ad 4e
00000070: 2a 46 08 5a 58 a1 4b 8e 53 2b a4 e6 3b fc 33 de
00000080: cf cb ee 50 6d a1 9f e4 94 06 19 39 39 6b 7e 4b
00000090: 83 f7 07 c0 bb 15 21 8d 8f 2d 5f 6c f6 97 68 21
000000A0: 3c ce c6 67 82 00 8f f3 d7 d6 c3 f2 87 47 b8 b9
000000B0: a3 0f f8 e2 0a 62 e8 f5 98 df bc f0 02 6a 3f 47
000000C0: c4 f0 24 a4 80 95 bf cf 32 5a a5 22 3c a5 a8 f1
000000D0: 57 d6 3b b8 06 1c b6 d7 c7 b3 58 e7 ee 69 eb 31
000000E0: d6 09 db 8b 8a 1d 2b a1 f7 46 e5 b9 99 13 73 30
000000F0: 1f ed 0c 82 4b cc ce 5e 25 79 1b ff 8b ca f0 b2
00000100: 1e 7e 70 03 66 c7 7b 6c 10 92 f2 34 b6 e9 ce bb
00000110: 65 ce d4 b5 99 f3 70 78 5f 06 f4 fe 0a 3c 00 28
00000120: 68
```

(13) Computes ICV using K3i as K_msg

```
00000000: fc 85 a4 7e 0b 41 77 54 ef 1a 03 cb
```

(14) Composes IV

```
00000000: 00 00 00 00 00 00 00 00
```

(15) Sends message, peer receives message

```
       10.111.10.171:54294->10.111.15.45:500 [341]

00000000: 43 87 64 8d 6c 9e 28 ff 82 d9 fa f8 74 49 b9 36
00000010: 2e 20 24 08 00 00 00 00 00 00 01 55 29 00 01 39
00000020: 00 00 00 00 00 00 00 00 00 9b 13 cb cb f1 18 53
00000030: fc 81 2e 75 c3 03 e0 ca 55 c1 fb 55 c0 29 40 48
00000040: fc 20 f4 a8 51 5b 97 6b c6 07 4c 7d 45 54 51 0f
00000050: 18 7f 43 a4 df 4b e8 e3 b4 eb 68 24 4b f0 1c df
00000060: 8f 1e a2 21 31 02 29 68 38 4d 68 fd 42 66 34 3e
00000070: 82 46 f0 17 02 bf 65 19 b0 f7 09 62 0d 12 6a 7e
00000080: ad 76 57 0d 19 55 cf 01 89 9c 7e f5 5a fa 20 4f
00000090: 8c 6d a4 83 b9 94 ad 4e 2a 46 08 5a 58 a1 4b 8e
000000A0: 53 2b a4 e6 3b fc 33 de cf cb ee 50 6d a1 9f e4
000000B0: 94 06 19 39 39 6b 7e 4b 83 f7 07 c0 bb 15 21 8d
000000C0: 8f 2d 5f 6c f6 97 68 21 3c ce c6 67 82 00 8f f3
000000D0: d7 d6 c3 f2 87 47 b8 b9 a3 0f f8 e2 0a 62 e8 f5
000000E0: 98 df bc f0 02 6a 3f 47 c4 f0 24 a4 80 95 bf cf
000000F0: 32 5a a5 22 3c a5 a8 f1 57 d6 3b b8 06 1c b6 d7
00000100: c7 b3 58 e7 ee 69 eb 31 d6 09 db 8b 8a 1d 2b a1
00000110: f7 46 e5 b9 99 13 73 30 1f ed 0c 82 4b cc ce 5e
00000120: 25 79 1b ff 8b ca f0 b2 1e 7e 70 03 66 c7 7b 6c
00000130: 10 92 f2 34 b6 e9 ce bb 65 ce d4 b5 99 f3 70 78
00000140: 5f 06 f4 fe 0a 3c 00 28 68 fc 85 a4 7e 0b 41 77
00000150: 54 ef 1a 03 cb
```

Responder's actions:

(16) Extracts IV from message

```
00000000: 00 00 00 00 00 00 00 00
```

(17) Computes K1i (i1 = 0)

```
00000000: 17 ec f1 84 33 9a c3 e3 93 e1 21 d7 65 3b 6c 83
00000010: d4 ae 9c 29 5b 12 cc b3 c5 0c 48 19 49 eb c0 ba
```

(18) Computes K2i (i2 = 0)

```
00000000: 2d 33 c0 55 87 f2 ee ce ac 1a f2 28 64 c6 f5 ad
00000010: de 2d be 7a a8 92 d0 a6 20 bc ef 25 29 7b 56 9f
```

(19) Computes K3i (i3 = 0)

```
00000000: c9 41 22 b5 39 b7 d2 3f c4 4d a6 ae 88 2e ff b4
00000010: f4 c0 90 9c bd bc 63 56 14 62 e8 8f 90 1a e7 eb
```

(20) Composes MGM nonce

```
00000000: 00 00 00 00 03 a0 05 b7 b2 2d f9 90 bb 6c ff ca
```

(21) Extracts ICV from message

```
00000000: fc 85 a4 7e 0b 41 77 54 ef 1a 03 cb
```

(22) Extracts AAD from message

```
00000000: 43 87 64 8d 6c 9e 28 ff 82 d9 fa f8 74 49 b9 36
00000010: 2e 20 24 08 00 00 00 00 00 00 01 55 29 00 01 39
```

(23) Extracts ciphertext from message

```
00000000: 00 9b 13 cb cb f1 18 53 fc 81 2e 75 c3 03 e0 ca
00000010: 55 c1 fb 55 c0 29 40 48 fc 20 f4 a8 51 5b 97 6b
00000020: c6 07 4c 7d 45 54 51 0f 18 7f 43 a4 df 4b e8 e3
00000030: b4 eb 68 24 4b f0 1c df 8f 1e a2 21 31 02 29 68
00000040: 38 4d 68 fd 42 66 34 3e 82 46 f0 17 02 bf 65 19
00000050: b0 f7 09 62 0d 12 6a 7e ad 76 57 0d 19 55 cf 01
00000060: 89 9c 7e f5 5a fa 20 4f 8c 6d a4 83 b9 94 ad 4e
00000070: 2a 46 08 5a 58 a1 4b 8e 53 2b a4 e6 3b fc 33 de
00000080: cf cb ee 50 6d a1 9f e4 94 06 19 39 39 6b 7e 4b
00000090: 83 f7 07 c0 bb 15 21 8d 8f 2d 5f 6c f6 97 68 21
000000A0: 3c ce c6 67 82 00 8f f3 d7 d6 c3 f2 87 47 b8 b9
000000B0: a3 0f f8 e2 0a 62 e8 f5 98 df bc f0 02 6a 3f 47
000000C0: c4 f0 24 a4 80 95 bf cf 32 5a a5 22 3c a5 a8 f1
000000D0: 57 d6 3b b8 06 1c b6 d7 c7 b3 58 e7 ee 69 eb 31
000000E0: d6 09 db 8b 8a 1d 2b a1 f7 46 e5 b9 99 13 73 30
000000F0: 1f ed 0c 82 4b cc ce 5e 25 79 1b ff 8b ca f0 b2
00000100: 1e 7e 70 03 66 c7 7b 6c 10 92 f2 34 b6 e9 ce bb
00000110: 65 ce d4 b5 99 f3 70 78 5f 06 f4 fe 0a 3c 00 28
00000120: 68
```

(24) Decrypts ciphertext and verifies ICV using K3i as K_msg, resulting in plaintext

```
00000000: 21 00 00 0c 03 04 40 09 0a de 5f cd 28 00 00 28
00000010: 00 00 00 24 01 03 04 03 a4 fe 65 a1 03 00 00 08
00000020: 01 00 00 20 03 00 00 08 04 00 00 22 00 00 00 08
00000030: 05 00 00 00 22 00 00 24 59 52 b2 58 00 b7 d3 f9
00000040: c3 31 23 16 6f c2 d1 d7 07 8b 99 fb 24 cf 24 30
00000050: a3 ce a6 fe d3 0f 20 9b 2c 00 00 88 00 22 00 00
00000060: 1c 55 08 b9 01 f5 76 6a 01 27 97 2d 38 b1 4a 5c
00000070: b7 43 f1 64 24 ef 76 75 50 ce 4f 6f 59 ca 96 ae
00000080: 54 85 9c 94 8d 04 91 62 3a 0c b6 6e 77 59 81 40
00000090: 69 bf bb 80 f7 7c 29 ee 9f 9e 0c 83 b6 08 fc 43
000000A0: b8 c6 66 36 e5 eb a0 43 c2 56 fa 52 f9 99 b6 95
000000B0: 34 4c cd 49 1f c7 83 9e d7 d9 ca e3 a5 d0 3c aa
000000C0: e8 ee ed 2c dd 5c 81 49 ab 3c d4 fa 15 4e 29 5f
000000D0: 7c cd b2 f1 c1 d2 6f 8f a7 74 4d 6a d8 8a c3 60
000000E0: 2d 00 00 18 01 00 00 00 07 00 00 10 00 00 ff ff
000000F0: 0a 01 01 02 0a 01 01 02 29 00 00 18 01 00 00 00
00000100: 07 00 00 10 00 00 ff ff 0a 00 00 00 0a 00 00 ff
00000110: 29 00 00 08 00 00 40 0a 00 00 00 08 00 00 40 0b
00000120: 00
```

(25) Parses received message

```
Create Child SA
4387648D6C9E28FF.82D9FAF87449B936.00000000 IKEv2 I->R[341]
  E[313]{
    N[12](ESP:0ADE5FCD:REKEY_SA),
    SA[40]{
      P[36](#1:ESP:A4FE65A1:3#){
        Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
        KE=GOST3410_2012_512,
        ESN=Off}},
    NONCE[36]{5952B2...0F209B},
    KE[136](GOST3410_2012_512){1C5508...8AC360},
    TSi[24](1#){10.1.1.2},
    TSr[24](1#){10.0.0.0-10.0.0.255},
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(26) Generates random IKE nonce Nr

```
00000000: f1 c1 3f 5e c4 c9 70 81 cb 1f 57 fe af 3d 80 37
00000010: 92 a9 ff 96 db 8f 3f 31 0a db 84 d1 24 d5 94 12
```

(27) Generates ephemeral private key

```
00000000: 2e 75 2f 5d 6c f0 9a 59 af 47 8d e1 2a a5 aa f5
00000010: c1 ef 9a fb e0 16 5e d9 59 6a c5 96 e8 88 14 62
00000020: 03 81 90 4f 18 d1 60 18 fe dc 9a a1 61 b3 8b c0
00000030: bf e0 d9 a0 d5 2b f2 7b 6b 60 f5 b9 4d e9 0b 36
```

(28) Computes public key

```
00000000: de 1d 91 64 c3 3e 58 4a b3 3e 55 5d 3e f6 5b cb
00000010: b5 c6 1c 09 cb 9a 17 91 81 13 5f 46 ce 52 98 c5
00000020: 1e bb 77 96 c9 04 03 2d f4 e5 23 f9 75 e3 ef a8
00000030: 53 52 b4 75 9c 00 55 7b 09 75 49 55 c1 65 7c 4d
00000040: 67 77 00 0a bc cd bc 4c 34 c3 b3 85 ed 86 7d 3b
00000050: 9f f7 15 ea 55 b5 e4 1e 45 d9 b0 4f 69 3f ee 7c
00000060: 89 0e 09 3d 4b 35 2e 8a 3c 0c 33 20 c3 54 7b 44
00000070: db 9f c7 96 a0 1e 9e ae b4 bd 29 73 b6 80 2d 00
```

(29) Selects SPI for new incoming ESP SA

```
00000000: 29 0a 8e 3f
```

(30) Computes keys for new ESP SAs

```
00000000: 4e c4 99 c2 d9 e8 fc 7f 26 fa cf df 20 8f a2 5c
00000010: 85 f8 e3 0c f7 fd 11 5b 5f 80 ba c4 e6 70 8b e4
00000020: 0b 90 d7 8f bd d4 c5 bd c4 31 6f 0b
00000000: 3c cc d8 46 72 44 68 c6 41 84 d2 22 ea 39 7c e8
00000010: aa 83 66 11 3a 26 4d 7b 07 52 6b c7 65 25 73 9d
00000020: 0f 3d 80 bc 8c 34 ff 07 31 11 5e d2
```

(31) Creates message

```
Create Child SA
4387648D6C9E28FF.82D9FAF87449B936.00000000 IKEv2 I<=R[337]
  E[309]{
    SA[40]{
      P[36](#1:ESP:290A8E3F:3#){
        Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
        KE=GOST3410_2012_512,
        ESN=Off}},
    NONCE[36]{F1C13F...D59412},
    KE[136](GOST3410_2012_512){DE1D91...802D00},
    TSi[24](1#){10.1.1.2},
    TSr[24](1#){10.0.0.0-10.0.0.255},
    N[8](ADDITIONAL_TS_POSSIBLE),
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(32) Computes K1r (i1 = 0)

```
00000000: 0c 45 d2 29 64 b8 72 57 11 10 3b a0 c2 66 d8 63
00000010: 34 f5 22 43 bf 6b 9a 1b 67 d6 d2 d8 fc 87 75 38
```

(33) Computes K2r (i2 = 0)

```
00000000: a9 92 d9 92 1f 15 13 bd db 61 83 43 58 2d dd e6
00000010: 66 28 4f 5d 71 47 a9 d4 8e 31 2e 95 37 f8 c5 d2
```

(34) Computes K3r (i3 = 0)

```
00000000: c1 ca 4f dd 2d 02 55 a4 11 9a 10 08 43 2d 61 ea
00000010: 52 68 83 c5 ec 92 53 24 01 b0 a2 0b d2 8f 72 78
```

(35) Composes MGM nonce

```
00000000: 00 00 00 00 84 57 87 2b 38 70 63 27 8c dd 88 78
```

(36) Composes AAD

```
00000000: 43 87 64 8d 6c 9e 28 ff 82 d9 fa f8 74 49 b9 36
00000010: 2e 20 24 20 00 00 00 00 00 00 01 51 21 00 01 35
```

(37) Composes plaintext

```
00000000: 28 00 00 28 00 00 00 24 01 03 04 03 29 0a 8e 3f
00000010: 03 00 00 08 01 00 00 20 03 00 00 08 04 00 00 22
00000020: 00 00 00 08 05 00 00 00 22 00 00 24 f1 c1 3f 5e
00000030: c4 c9 70 81 cb 1f 57 fe af 3d 80 37 92 a9 ff 96
00000040: db 8f 3f 31 0a db 84 d1 24 d5 94 12 2c 00 00 88
00000050: 00 22 00 00 de 1d 91 64 c3 3e 58 4a b3 3e 55 5d
00000060: 3e f6 5b cb b5 c6 1c 09 cb 9a 17 91 81 13 5f 46
00000070: ce 52 98 c5 1e bb 77 96 c9 04 03 2d f4 e5 23 f9
00000080: 75 e3 ef a8 53 52 b4 75 9c 00 55 7b 09 75 49 55
00000090: c1 65 7c 4d 67 77 00 0a bc cd bc 4c 34 c3 b3 85
000000A0: ed 86 7d 3b 9f f7 15 ea 55 b5 e4 1e 45 d9 b0 4f
000000B0: 69 3f ee 7c 89 0e 09 3d 4b 35 2e 8a 3c 0c 33 20
000000C0: c3 54 7b 44 db 9f c7 96 a0 1e 9e ae b4 bd 29 73
000000D0: b6 80 2d 00 2d 00 00 18 01 00 00 00 07 00 00 10
000000E0: 00 00 ff ff 0a 01 01 02 0a 01 01 02 29 00 00 18
000000F0: 01 00 00 00 07 00 00 10 00 00 ff ff 0a 00 00 00
00000100: 0a 00 00 ff 29 00 00 08 00 00 40 02 29 00 00 08
00000110: 00 00 40 0a 00 00 00 08 00 00 40 0b 00
```

(38) Encrypts plaintext using K3r as K_msg, resulting in ciphertext

```
00000000: 42 73 5f 2b 14 a0 27 ca 3c 90 67 80 3c 3d 99 02
00000010: 1c 08 c8 67 03 0f 69 f1 c3 64 43 a6 59 74 ce b0
00000020: d7 5d 29 58 53 3a f6 c3 20 04 56 ba 2e af 14 9b
00000030: 2d a3 93 15 2c e5 15 e6 59 2b 7f 47 94 7f 90 82
00000040: ce d3 64 cc 89 92 04 c6 bc 7b ce 61 c6 1d 7f a5
00000050: 45 1c 27 e6 0b 78 1a f2 75 8f 3e 47 53 8e d7 16
00000060: 11 f4 26 04 ae 5e d5 b8 84 b6 ac e6 20 28 da ca
00000070: da 84 fe 0d c4 4d 29 2f 58 30 fe 93 f6 59 04 4a
00000080: 9b aa 97 99 5b 5e 74 9c 5d 45 d5 99 42 16 8c ab
00000090: 62 cb 9f 14 5f f5 25 92 34 5c 8d 61 45 44 55 6d
000000A0: 3d 80 b0 39 f0 39 0b 43 8a f9 b7 b7 17 41 34 ce
000000B0: 36 bf e3 e7 1a 68 61 72 0e f1 91 24 89 ab d7 e9
000000C0: a9 b1 87 38 a1 c0 4c 42 4e 47 62 28 9e d7 1f 02
000000D0: 13 40 69 38 31 f1 91 87 ec 54 11 0a 2d d9 25 15
000000E0: 15 16 37 b7 71 94 11 49 5e f7 28 90 c5 1e 6b 07
000000F0: d9 cf 06 a2 a2 33 0e e0 25 67 db a6 17 11 27 60
00000100: c8 21 f7 79 63 aa b0 f9 7b 95 03 a7 8d 2e d7 df
00000110: 58 e7 30 ab d3 c8 f1 24 40 69 fc 3f bf
```

(39) Computes ICV using K3r as K_msg

```
00000000: 3a 2d 3c 6b 87 43 ed 6e 80 ab 27 e2
```

(40) Composes IV

```
00000000: 00 00 00 00 00 00 00 00
```

(41) Sends message, peer receives message

```
       10.111.10.171:54294<-10.111.15.45:500 [337]

00000000: 43 87 64 8d 6c 9e 28 ff 82 d9 fa f8 74 49 b9 36
00000010: 2e 20 24 20 00 00 00 00 00 00 01 51 21 00 01 35
00000020: 00 00 00 00 00 00 00 00 42 73 5f 2b 14 a0 27 ca
00000030: 3c 90 67 80 3c 3d 99 02 1c 08 c8 67 03 0f 69 f1
00000040: c3 64 43 a6 59 74 ce b0 d7 5d 29 58 53 3a f6 c3
00000050: 20 04 56 ba 2e af 14 9b 2d a3 93 15 2c e5 15 e6
00000060: 59 2b 7f 47 94 7f 90 82 ce d3 64 cc 89 92 04 c6
00000070: bc 7b ce 61 c6 1d 7f a5 45 1c 27 e6 0b 78 1a f2
00000080: 75 8f 3e 47 53 8e d7 16 11 f4 26 04 ae 5e d5 b8
00000090: 84 b6 ac e6 20 28 da ca da 84 fe 0d c4 4d 29 2f
000000A0: 58 30 fe 93 f6 59 04 4a 9b aa 97 99 5b 5e 74 9c
000000B0: 5d 45 d5 99 42 16 8c ab 62 cb 9f 14 5f f5 25 92
000000C0: 34 5c 8d 61 45 44 55 6d 3d 80 b0 39 f0 39 0b 43
000000D0: 8a f9 b7 b7 17 41 34 ce 36 bf e3 e7 1a 68 61 72
000000E0: 0e f1 91 24 89 ab d7 e9 a9 b1 87 38 a1 c0 4c 42
000000F0: 4e 47 62 28 9e d7 1f 02 13 40 69 38 31 f1 91 87
00000100: ec 54 11 0a 2d d9 25 15 15 16 37 b7 71 94 11 49
00000110: 5e f7 28 90 c5 1e 6b 07 d9 cf 06 a2 a2 33 0e e0
00000120: 25 67 db a6 17 11 27 60 c8 21 f7 79 63 aa b0 f9
00000130: 7b 95 03 a7 8d 2e d7 df 58 e7 30 ab d3 c8 f1 24
00000140: 40 69 fc 3f bf 3a 2d 3c 6b 87 43 ed 6e 80 ab 27
00000150: e2
```

Initiator's actions:

(42) Extracts IV from message

```
00000000: 00 00 00 00 00 00 00 00
```

(43) Computes K1r (i1 = 0)

```
00000000: 0c 45 d2 29 64 b8 72 57 11 10 3b a0 c2 66 d8 63
00000010: 34 f5 22 43 bf 6b 9a 1b 67 d6 d2 d8 fc 87 75 38
```

(44) Computes K2r (i2 = 0)

```
00000000: a9 92 d9 92 1f 15 13 bd db 61 83 43 58 2d dd e6
00000010: 66 28 4f 5d 71 47 a9 d4 8e 31 2e 95 37 f8 c5 d2
```

(45) Computes K3r (i3 = 0)

```
00000000: c1 ca 4f dd 2d 02 55 a4 11 9a 10 08 43 2d 61 ea
00000010: 52 68 83 c5 ec 92 53 24 01 b0 a2 0b d2 8f 72 78
```

(46) Composes MGM nonce

```
00000000: 00 00 00 00 84 57 87 2b 38 70 63 27 8c dd 88 78
```

(47) Extracts ICV from message

```
00000000: 3a 2d 3c 6b 87 43 ed 6e 80 ab 27 e2
```

(48) Extracts AAD from message

```
00000000: 43 87 64 8d 6c 9e 28 ff 82 d9 fa f8 74 49 b9 36
00000010: 2e 20 24 20 00 00 00 00 00 00 01 51 21 00 01 35
```

(49) Extracts ciphertext from message

```
00000000: 42 73 5f 2b 14 a0 27 ca 3c 90 67 80 3c 3d 99 02
00000010: 1c 08 c8 67 03 0f 69 f1 c3 64 43 a6 59 74 ce b0
00000020: d7 5d 29 58 53 3a f6 c3 20 04 56 ba 2e af 14 9b
00000030: 2d a3 93 15 2c e5 15 e6 59 2b 7f 47 94 7f 90 82
00000040: ce d3 64 cc 89 92 04 c6 bc 7b ce 61 c6 1d 7f a5
00000050: 45 1c 27 e6 0b 78 1a f2 75 8f 3e 47 53 8e d7 16
00000060: 11 f4 26 04 ae 5e d5 b8 84 b6 ac e6 20 28 da ca
00000070: da 84 fe 0d c4 4d 29 2f 58 30 fe 93 f6 59 04 4a
00000080: 9b aa 97 99 5b 5e 74 9c 5d 45 d5 99 42 16 8c ab
00000090: 62 cb 9f 14 5f f5 25 92 34 5c 8d 61 45 44 55 6d
000000A0: 3d 80 b0 39 f0 39 0b 43 8a f9 b7 b7 17 41 34 ce
000000B0: 36 bf e3 e7 1a 68 61 72 0e f1 91 24 89 ab d7 e9
000000C0: a9 b1 87 38 a1 c0 4c 42 4e 47 62 28 9e d7 1f 02
000000D0: 13 40 69 38 31 f1 91 87 ec 54 11 0a 2d d9 25 15
000000E0: 15 16 37 b7 71 94 11 49 5e f7 28 90 c5 1e 6b 07
000000F0: d9 cf 06 a2 a2 33 0e e0 25 67 db a6 17 11 27 60
00000100: c8 21 f7 79 63 aa b0 f9 7b 95 03 a7 8d 2e d7 df
00000110: 58 e7 30 ab d3 c8 f1 24 40 69 fc 3f bf
```

(50) Decrypts ciphertext and verifies ICV using K3r as K_msg, resulting in plaintext

```
00000000: 28 00 00 28 00 00 00 24 01 03 04 03 29 0a 8e 3f
00000010: 03 00 00 08 01 00 00 20 03 00 00 08 04 00 00 22
00000020: 00 00 00 08 05 00 00 00 22 00 00 24 f1 c1 3f 5e
00000030: c4 c9 70 81 cb 1f 57 fe af 3d 80 37 92 a9 ff 96
00000040: db 8f 3f 31 0a db 84 d1 24 d5 94 12 2c 00 00 88
00000050: 00 22 00 00 de 1d 91 64 c3 3e 58 4a b3 3e 55 5d
00000060: 3e f6 5b cb b5 c6 1c 09 cb 9a 17 91 81 13 5f 46
00000070: ce 52 98 c5 1e bb 77 96 c9 04 03 2d f4 e5 23 f9
00000080: 75 e3 ef a8 53 52 b4 75 9c 00 55 7b 09 75 49 55
00000090: c1 65 7c 4d 67 77 00 0a bc cd bc 4c 34 c3 b3 85
000000A0: ed 86 7d 3b 9f f7 15 ea 55 b5 e4 1e 45 d9 b0 4f
000000B0: 69 3f ee 7c 89 0e 09 3d 4b 35 2e 8a 3c 0c 33 20
000000C0: c3 54 7b 44 db 9f c7 96 a0 1e 9e ae b4 bd 29 73
000000D0: b6 80 2d 00 2d 00 00 18 01 00 00 00 07 00 00 10
000000E0: 00 00 ff ff 0a 01 01 02 0a 01 01 02 29 00 00 18
000000F0: 01 00 00 00 07 00 00 10 00 00 ff ff 0a 00 00 00
00000100: 0a 00 00 ff 29 00 00 08 00 00 40 02 29 00 00 08
00000110: 00 00 40 0a 00 00 00 08 00 00 40 0b 00
```

(51) Parses received message

```
Create Child SA
4387648D6C9E28FF.82D9FAF87449B936.00000000 IKEv2 R=>I[337]
  E[309]{
    SA[40]{
      P[36](#1:ESP:290A8E3F:3#){
        Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
        KE=GOST3410_2012_512,
        ESN=Off}},
    NONCE[36]{F1C13F...D59412},
    KE[136](GOST3410_2012_512){DE1D91...802D00},
    TSi[24](1#){10.1.1.2},
    TSr[24](1#){10.0.0.0-10.0.0.255},
    N[8](ADDITIONAL_TS_POSSIBLE),
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(52) Computes keys for new ESP SAs

```
00000000: 4e c4 99 c2 d9 e8 fc 7f 26 fa cf df 20 8f a2 5c
00000010: 85 f8 e3 0c f7 fd 11 5b 5f 80 ba c4 e6 70 8b e4
00000020: 0b 90 d7 8f bd d4 c5 bd c4 31 6f 0b
00000000: 3c cc d8 46 72 44 68 c6 41 84 d2 22 ea 39 7c e8
00000010: aa 83 66 11 3a 26 4d 7b 07 52 6b c7 65 25 73 9d
00000020: 0f 3d 80 bc 8c 34 ff 07 31 11 5e d2
```

A.1.4. Sub-Scenario 4: IKE SA Deletion Using the INFORMATIONAL Exchange

```
Initiator                                  Responder

HDR, SK {D}             --->
                        <---               HDR, SK { }
```

Initiator's actions:

(1) Creates message

```
Informational
4387648D6C9E28FF.82D9FAF87449B936.00000003 IKEv2 R<-I[61]
  E[33]{
    D[8](IKE)}
```

(2) Uses previously computed key K3i

```
00000000: c9 41 22 b5 39 b7 d2 3f c4 4d a6 ae 88 2e ff b4
00000010: f4 c0 90 9c bd bc 63 56 14 62 e8 8f 90 1a e7 eb
```

(3) Composes MGM nonce

```
00000000: 00 00 00 03 03 a0 05 b7 b2 2d f9 90 bb 6c ff ca
```

(4) Composes AAD

```
00000000: 43 87 64 8d 6c 9e 28 ff 82 d9 fa f8 74 49 b9 36
00000010: 2e 20 25 08 00 00 00 03 00 00 00 3d 2a 00 00 21
```

(5) Composes plaintext

```
00000000: 00 00 00 08 01 00 00 00 00
```

(6) Encrypts plaintext using K3i as K_msg, resulting in ciphertext

```
00000000: 3e 17 6f 6c 23 48 06 e9 fd
```

(7) Computes ICV using K3i as K_msg

```
00000000: 23 7b a2 fc d5 1c 6f 2c c0 1e 21 e4
```

(8) Composes IV

```
00000000: 00 00 00 00 00 00 00 03
```

(9) Sends message, peer receives message

```
       10.111.10.171:54294->10.111.15.45:500 [61]

00000000: 43 87 64 8d 6c 9e 28 ff 82 d9 fa f8 74 49 b9 36
00000010: 2e 20 25 08 00 00 00 03 00 00 00 3d 2a 00 00 21
00000020: 00 00 00 00 00 00 00 03 3e 17 6f 6c 23 48 06 e9
00000030: fd 23 7b a2 fc d5 1c 6f 2c c0 1e 21 e4
```

Responder's actions:

(10) Extracts IV from message

```
00000000: 00 00 00 00 00 00 00 03
```

(11) Uses previously computed key K3i

```
00000000: c9 41 22 b5 39 b7 d2 3f c4 4d a6 ae 88 2e ff b4
00000010: f4 c0 90 9c bd bc 63 56 14 62 e8 8f 90 1a e7 eb
```

(12) Composes MGM nonce

```
00000000: 00 00 00 03 03 a0 05 b7 b2 2d f9 90 bb 6c ff ca
```

(13) Extracts ICV from message

```
00000000: 23 7b a2 fc d5 1c 6f 2c c0 1e 21 e4
```

(14) Extracts AAD from message

```
00000000: 43 87 64 8d 6c 9e 28 ff 82 d9 fa f8 74 49 b9 36
00000010: 2e 20 25 08 00 00 00 03 00 00 00 3d 2a 00 00 21
```

(15) Extracts ciphertext from message

```
00000000: 3e 17 6f 6c 23 48 06 e9 fd
```

(16) Decrypts ciphertext and verifies ICV using K3i as K_msg, resulting in plaintext

```
00000000: 00 00 00 08 01 00 00 00 00
```

(17) Parses received message

```
Informational
4387648D6C9E28FF.82D9FAF87449B936.00000003 IKEv2 I->R[61]
  E[33]{
    D[8](IKE)}
```

(18) Creates message

```
Informational
4387648D6C9E28FF.82D9FAF87449B936.00000003 IKEv2 I<=R[53]
  E[25]{}
```

(19) Uses previously computed key K3r

```
00000000: c1 ca 4f dd 2d 02 55 a4 11 9a 10 08 43 2d 61 ea
00000010: 52 68 83 c5 ec 92 53 24 01 b0 a2 0b d2 8f 72 78
```

(20) Composes MGM nonce

```
00000000: 00 00 00 03 84 57 87 2b 38 70 63 27 8c dd 88 78
```

(21) Composes AAD

```
00000000: 43 87 64 8d 6c 9e 28 ff 82 d9 fa f8 74 49 b9 36
00000010: 2e 20 25 20 00 00 00 03 00 00 00 35 00 00 00 19
```

(22) Composes plaintext

```
00000000: 00
```

(23) Encrypts plaintext using K3r as K_msg, resulting in ciphertext

```
00000000: f1
```

(24) Computes ICV using K3r as K_msg

```
00000000: 38 3b 47 ed 04 4d af 44 b8 59 9a ce
```

(25) Composes IV

```
00000000: 00 00 00 00 00 00 00 03
```

(26) Sends message, peer receives message

```
       10.111.10.171:54294<-10.111.15.45:500 [53]

00000000: 43 87 64 8d 6c 9e 28 ff 82 d9 fa f8 74 49 b9 36
00000010: 2e 20 25 20 00 00 00 03 00 00 00 35 00 00 00 19
00000020: 00 00 00 00 00 00 00 03 f1 38 3b 47 ed 04 4d af
00000030: 44 b8 59 9a ce
```

Initiator's actions:

(27) Extracts IV from message

```
00000000: 00 00 00 00 00 00 00 03
```

(28) Uses previously computed key K3r

```
00000000: c1 ca 4f dd 2d 02 55 a4 11 9a 10 08 43 2d 61 ea
00000010: 52 68 83 c5 ec 92 53 24 01 b0 a2 0b d2 8f 72 78
```

(29) Composes MGM nonce

```
00000000: 00 00 00 03 84 57 87 2b 38 70 63 27 8c dd 88 78
```

(30) Extracts ICV from message

```
00000000: 38 3b 47 ed 04 4d af 44 b8 59 9a ce
```

(31) Extracts AAD from message

```
00000000: 43 87 64 8d 6c 9e 28 ff 82 d9 fa f8 74 49 b9 36
00000010: 2e 20 25 20 00 00 00 03 00 00 00 35 00 00 00 19
```

(32) Extracts ciphertext from message

```
00000000: f1
```

(33) Decrypts ciphertext and verifies ICV using K3r as K_msg, resulting in plaintext

```
00000000: 00
```

(34) Parses received message

```
Informational
4387648D6C9E28FF.82D9FAF87449B936.00000003 IKEv2 R=>I[53]
  E[25]{}
```

A.2. Scenario 2

In this scenario, peers establish, rekey, and delete an IKE SA and ESP SAs using the following prerequisites:

- Peers authenticate each other using digital signatures.
- Initiator's ID is "CN=IKE Interop Test Client, O=ELVIS-PLUS, C=RU" of type ID_DER_ASN1_DN:

```
00000010: 30 44 31 20 30 1e 06 03 55 04 03 13 17 49 4b 45
00000020: 20 49 6e 74 65 72 6f 70 20 54 65 73 74 20 43 6c
00000030: 69 65 6e 74 31 13 30 11 06 03 55 04 0a 13 0a 45
00000040: 4c 56 49 53 2d 50 4c 55 53 31 0b 30 09 06 03 55
00000050: 04 06 13 02 52 55
```

- Responder's ID is "CN=IKE Interop Test Server, O=ELVIS-PLUS, C=RU" of type ID_DER_ASN1_DN:

```
00000010: 30 44 31 20 30 1e 06 03 55 04 03 13 17 49 4b 45
00000020: 20 49 6e 74 65 72 6f 70 20 54 65 73 74 20 53 65
00000030: 72 76 65 72 31 13 30 11 06 03 55 04 0a 13 0a 45
00000040: 4c 56 49 53 2d 50 4c 55 53 31 0b 30 09 06 03 55
00000050: 04 06 13 02 52 55
```

- No NAT is present between the peers, but using UDP encapsulation is forced by the initiator by setting the NAT_DETECTION_SOURCE_IP notification data to all zeroes.
- IKE fragmentation is used in the IKE_AUTH exchange.
- IKE SA is created with the following transforms:
  - ENCR_MAGMA_MGM_KTREE
  - PRF_HMAC_STREEBOG_512
  - GOST3410_2012_256

- ESP SAs are created with the following transforms:
  - ENCR_MAGMA_MGM_KTREE
  - ESN off

The certificates for this scenario were obtained from the public testing CA service <https://testgost2012.cryptopro.ru/certsrv/>.

The initiator's certificate private key (little endian):

```
0000000000: 76 e9 dd b3 f3 a2 08 a2 4e a5 81 9c ae 41 da b4
0000000010: 77 3c 1d d5 dc eb af e6 58 b1 47 d2 d8 29 ce 71
0000000020: 18 a9 85 5d 28 5b 3c e3 23 bd 80 ac 2f 00 cc b6
0000000030: 61 4c 42 a1 65 61 02 cf 33 eb 1f 5f 02 ce 8a b9
```

The initiator's certificate:

```
0000000000: 30 82 04 f7 30 82 04 a4 a0 03 02 01 02 02 13 7c
0000000010: 00 03 da a8 9e 1e ff 9e 79 05 fb bb 00 01 00 03
0000000020: da a8 30 0a 06 08 2a 85 03 07 01 01 03 02 30 82
0000000030: 01 0a 31 18 30 16 06 05 2a 85 03 64 01 12 0d 31
0000000040: 32 33 34 35 36 37 38 39 30 31 32 33 31 1a 30 18
0000000050: 06 08 2a 85 03 03 81 03 01 01 12 0c 30 30 31 32
0000000060: 33 34 35 36 37 38 39 30 31 2f 30 2d 06 03 55 04
0000000070: 09 0c 26 d1 83 d0 bb 2e 20 d0 a1 d1 83 d1 89 d1
0000000080: 91 d0 b2 d1 81 d0 ba d0 b8 d0 b9 20 d0 b2 d0 b0
0000000090: d0 bb 20 d0 b4 2e 20 31 38 31 0b 30 09 06 03 55
00000000A0: 04 06 13 02 52 55 31 19 30 17 06 03 55 04 08 0c
00000000B0: 10 d0 b3 2e 20 d0 9c d0 be d1 81 d0 ba d0 b2 d0
00000000C0: b0 31 15 30 13 06 03 55 04 07 0c 0c d0 9c d0 be
00000000D0: d1 81 d0 ba d0 b2 d0 b0 31 25 30 23 06 03 55 04
00000000E0: 0a 0c 1c d0 9e d0 9e d0 9e 20 22 d0 9a d0 a0 d0
00000000F0: 98 d0 9f d0 a2 d0 9e 2d d0 9f d0 a0 d0 9e 22 31
0000000100: 3b 30 39 06 03 55 04 03 0c 32 d0 a2 d0 b5 d1 81
0000000110: d1 82 d0 be d0 b2 d1 8b d0 b9 20 d0 a3 d0 a6 20
0000000120: d0 9e d0 9e d0 9e 20 22 d0 9a d0 a0 d0 98 d0 9f
0000000130: d0 a2 d0 9e 2d d0 9f d0 a0 d0 9e 22 30 1e 17 0d
0000000140: 32 31 31 30 30 31 30 36 31 30 31 30 5a 17 0d 32
0000000150: 32 30 31 30 31 30 36 32 30 31 30 5a 30 44 31 20
0000000160: 30 1e 06 03 55 04 03 13 17 49 4b 45 20 49 6e 74
0000000170: 65 72 6f 70 20 54 65 73 74 20 43 6c 69 65 6e 74
0000000180: 31 13 30 11 06 03 55 04 0a 13 0a 45 4c 56 49 53
0000000190: 2d 50 4c 55 53 31 0b 30 09 06 03 55 04 06 13 02
00000001A0: 52 55 30 81 aa 30 21 06 08 2a 85 03 07 01 01 01
00000001B0: 02 30 15 06 09 2a 85 03 07 01 02 01 02 01 06 08
00000001C0: 2a 85 03 07 01 01 02 03 03 81 84 00 04 81 80 ee
00000001D0: 2f 0a 0e 09 1e 7e 04 ef ba 5b 62 a2 52 86 e1 9c
00000001E0: 24 50 30 50 b0 b4 8a 37 35 b5 fc af 28 94 ec b5
00000001F0: 9b 92 41 5b 69 e2 c9 ba 24 de 6a 72 c4 ef 44 bb
0000000200: 89 a1 05 14 1b 87 3d 6a a3 72 3e 17 ca 7f 39 28
0000000210: ce 16 8b dd 07 52 87 6a 0d 77 42 6d 99 2b 46 2c
0000000220: fd 4b b2 7c d7 c7 17 08 12 54 63 47 9d 14 3d 61
0000000230: ed f2 95 ab 11 80 69 02 a7 66 60 50 7e a4 53 6d
0000000240: ad 01 49 b2 16 8a 95 1d cf 1a 57 93 56 14 5e a3
```

```
0000000250: 82 02 59 30 82 02 55 30 0e 06 03 55 1d 0f 01 01
0000000260: ff 04 04 03 02 05 a0 30 13 06 03 55 1d 25 04 0c
0000000270: 30 0a 06 08 2b 06 01 05 05 07 03 11 30 1d 06 03
0000000280: 55 1d 0e 04 16 04 14 40 81 b1 d1 18 75 f0 da 6b
0000000290: 3c 50 5f cd 73 1d d9 77 f2 d7 c1 30 1f 06 03 55
00000002A0: 1d 23 04 18 30 16 80 14 9b 85 5e fb 81 dc 4d 59
00000002B0: 07 51 63 cf be df da 2c 7f c9 44 3c 30 82 01 0f
00000002C0: 06 03 55 1d 1f 04 82 01 06 30 82 01 02 30 81 ff
00000002D0: a0 81 fc a0 81 f9 86 81 b5 68 74 74 70 3a 2f 2f
00000002E0: 74 65 73 74 67 6f 73 74 32 30 31 32 2e 63 72 79
00000002F0: 70 74 6f 70 72 6f 2e 72 75 2f 43 65 72 74 45 6e
0000000300: 72 6f 6c 6c 2f 21 30 34 32 32 21 30 34 33 35 21
0000000310: 30 34 34 31 21 30 34 34 32 21 30 34 33 65 21 30
0000000320: 34 33 32 21 30 34 34 62 21 30 34 33 39 25 32 30
0000000330: 21 30 34 32 33 21 30 34 32 36 25 32 30 21 30 34
0000000340: 31 65 21 30 34 31 65 21 30 34 31 65 25 32 30 21
0000000350: 30 30 32 32 21 30 34 31 61 21 30 34 32 30 21 30
0000000360: 34 31 38 21 30 34 31 66 21 30 34 32 32 21 30 34
0000000370: 31 65 2d 21 30 34 31 66 21 30 34 32 30 21 30 34
0000000380: 31 65 21 30 30 32 32 28 31 29 2e 63 72 6c 86 3f
0000000390: 68 74 74 70 3a 2f 2f 74 65 73 74 67 6f 73 74 32
00000003A0: 30 31 32 2e 63 72 79 70 74 6f 70 72 6f 2e 72 75
00000003B0: 2f 43 65 72 74 45 6e 72 6f 6c 6c 2f 74 65 73 74
00000003C0: 67 6f 73 74 32 30 31 32 28 31 29 2e 63 72 6c 30
00000003D0: 81 da 06 08 2b 06 01 05 05 07 01 01 04 81 cd 30
00000003E0: 81 ca 30 44 06 08 2b 06 01 05 05 07 30 02 86 38
00000003F0: 68 74 74 70 3a 2f 2f 74 65 73 74 67 6f 73 74 32
0000000400: 30 31 32 2e 63 72 79 70 74 6f 70 72 6f 2e 72 75
0000000410: 2f 43 65 72 74 45 6e 72 6f 6c 6c 2f 72 6f 6f 74
0000000420: 32 30 31 38 2e 63 72 74 30 3f 06 08 2b 06 01 05
0000000430: 05 07 30 01 86 33 68 74 74 70 3a 2f 2f 74 65 73
0000000440: 74 67 6f 73 74 32 30 31 32 2e 63 72 79 70 74 6f
0000000450: 70 72 6f 2e 72 75 2f 6f 63 73 70 32 30 31 32 67
0000000460: 2f 6f 63 73 70 2e 73 72 66 30 41 06 08 2b 06 01
0000000470: 05 05 07 30 01 86 35 68 74 74 70 3a 2f 2f 74 65
0000000480: 73 74 67 6f 73 74 32 30 31 32 2e 63 72 79 70 74
0000000490: 6f 70 72 6f 2e 72 75 2f 6f 63 73 70 32 30 31 32
00000004A0: 67 73 74 2f 6f 63 73 70 2e 73 72 66 30 0a 06 08
00000004B0: 2a 85 03 07 01 01 03 02 03 41 00 21 ee 3b e1 fd
00000004C0: 0f 36 90 92 c4 a2 35 26 e8 dc 4e b8 ef 89 40 70
00000004D0: d2 91 39 bc 79 a6 e2 f7 c1 06 bd d5 d6 ff 72 a5
00000004E0: 6c f2 c0 c3 75 e9 ca 67 81 c1 93 96 b4 bd 18 12
00000004F0: 4c 37 f7 d9 73 d6 4c 8a a6 c4 0a
```

```
   0 1271: SEQUENCE {
   4 1188:  SEQUENCE {
   8    3:   [0] {
  10    1:    INTEGER 2
         :     }
  13   19:   INTEGER
         : 7c 00 03 da a8 9e 1e ff 9e 79 05 fb bb 00 01 00
         : 03 da a8
  34   10:   SEQUENCE {
  36    8:    OBJECT IDENTIFIER
         :     gost2012Signature256 (1 2 643 7 1 1 3 2)
         :     }
  46  266:   SEQUENCE {
  50   24:    SET {
  52   22:     SEQUENCE {
  54    5:      OBJECT IDENTIFIER '1 2 643 100 1'
  61   13:      NumericString '1234567890123'
         :       }
         :      }
  76   26:    SET {
  78   24:     SEQUENCE {
  80    8:      OBJECT IDENTIFIER '1 2 643 3 131 1 1'
  90   12:      NumericString '001234567890'
         :       }
         :      }
 104   47:    SET {
 106   45:     SEQUENCE {
 108    3:      OBJECT IDENTIFIER
         :       streetAddress (2 5 4 9)
 113   38:      UTF8String 'ул. Сущёвский вал д. 18'
         :       }
         :      }
 153   11:    SET {
 155    9:     SEQUENCE {
 157    3:      OBJECT IDENTIFIER
         :       countryName (2 5 4 6)
 162    2:      PrintableString 'RU'
         :       }
         :      }
 166   25:    SET {
 168   23:     SEQUENCE {
 170    3:      OBJECT IDENTIFIER
         :       stateOrProvinceName (2 5 4 8)
 175   16:      UTF8String 'г. Москва'
         :       }
         :      }
 193   21:    SET {
 195   19:     SEQUENCE {
 197    3:      OBJECT IDENTIFIER
         :       localityName (2 5 4 7)
 202   12:      UTF8String 'Москва'
         :       }
         :      }
 216   37:    SET {
 218   35:     SEQUENCE {
 220    3:      OBJECT IDENTIFIER
         :       organizationName (2 5 4 10)
 225   28:      UTF8String 'ООО "КРИПТО-ПРО"'
         :       }
         :      }
 255   59:    SET {
 257   57:     SEQUENCE {
 259    3:      OBJECT IDENTIFIER
         :       commonName (2 5 4 3)
 264   50:      UTF8String
         :       'Тестовый УЦ ООО "КРИПТО-ПРО"'
         :       }
         :      }
         :     }
 316   30:   SEQUENCE {
```

```
318   13:    UTCTime 01/10/2021 06:10:10 GMT
333   13:    UTCTime 01/01/2022 06:20:10 GMT
        :     }
348   68:   SEQUENCE {
350   32:    SET {
352   30:     SEQUENCE {
354    3:      OBJECT IDENTIFIER
        :       commonName (2 5 4 3)
359   23:      PrintableString 'IKE Interop Test Client'
        :       }
        :      }
384   19:    SET {
386   17:     SEQUENCE {
388    3:      OBJECT IDENTIFIER
        :       organizationName (2 5 4 10)
393   10:      PrintableString 'ELVIS-PLUS'
        :       }
        :      }
405   11:    SET {
407    9:     SEQUENCE {
409    3:      OBJECT IDENTIFIER
        :       countryName (2 5 4 6)
414    2:      PrintableString 'RU'
        :       }
        :      }
        :     }
418  170:   SEQUENCE {
421   33:    SEQUENCE {
423    8:     OBJECT IDENTIFIER
        :      gost2012PublicKey512 (1 2 643 7 1 1 1 2)
433   21:     SEQUENCE {
435    9:      OBJECT IDENTIFIER
        :       cryptoPro2012Sign512A (1 2 643 7 1 2 1 2 1)
446    8:      OBJECT IDENTIFIER
        :       gost2012Digest512 (1 2 643 7 1 1 2 3)
        :       }
        :      }
456  132:    BIT STRING, encapsulates {
460  128:     OCTET STRING
        : ee 2f 0a 0e 09 1e 7e 04 ef ba 5b 62 a2 52 86 e1
        : 9c 24 50 30 50 b0 b4 8a 37 35 b5 fc af 28 94 ec
        : b5 9b 92 41 5b 69 e2 c9 ba 24 de 6a 72 c4 ef 44
        : bb 89 a1 05 14 1b 87 3d 6a a3 72 3e 17 ca 7f 39
        : 28 ce 16 8b dd 07 52 87 6a 0d 77 42 6d 99 2b 46
        : 2c fd 4b b2 7c d7 c7 17 08 12 54 63 47 9d 14 3d
        : 61 ed f2 95 ab 11 80 69 02 a7 66 60 50 7e a4 53
        : 6d ad 01 49 b2 16 8a 95 1d cf 1a 57 93 56 14 5e
        :      }
        :     }
591  601:   [3] {
595  597:    SEQUENCE {
599   14:     SEQUENCE {
601    3:      OBJECT IDENTIFIER
        :       keyUsage (2 5 29 15)
606    1:      BOOLEAN TRUE
609    4:      OCTET STRING, encapsulates {
611    2:       BIT STRING 5 unused bits
        :        '101'B
```

```
           :        }
           :       }
   615   19:     SEQUENCE {
   617    3:      OBJECT IDENTIFIER
           :       extKeyUsage (2 5 29 37)
   622   12:      OCTET STRING, encapsulates {
   624   10:       SEQUENCE {
   626    8:        OBJECT IDENTIFIER
           :         ipsecIKE (1 3 6 1 5 5 7 3 17)
           :         }
           :        }
           :       }
   636   29:     SEQUENCE {
   638    3:      OBJECT IDENTIFIER
           :       subjectKeyIdentifier (2 5 29 14)
   643   22:      OCTET STRING, encapsulates {
   645   20:       OCTET STRING
           : 40 81 b1 d1 18 75 f0 da 6b 3c 50 5f cd 73 1d d9
           : 77 f2 d7 c1
           :        }
           :       }
   667   31:     SEQUENCE {
   669    3:      OBJECT IDENTIFIER
           :       authorityKeyIdentifier (2 5 29 35)
   674   24:      OCTET STRING, encapsulates {
   676   22:       SEQUENCE {
   678   20:        [0]
           : 9b 85 5e fb 81 dc 4d 59 07 51 63 cf be df da 2c
           : 7f c9 44 3c
           :         }
           :        }
           :       }
   700  271:     SEQUENCE {
   704    3:      OBJECT IDENTIFIER
           :       cRLDistributionPoints (2 5 29 31)
   709  262:      OCTET STRING, encapsulates {
   713  258:       SEQUENCE {
   717  255:        SEQUENCE {
   720  252:         [0] {
   723  249:          [0] {
   726  181:           [6]
           :         'http://testgost2012.cryptopro.ru/CertEnroll/!042'
           :         '2!0435!0441!0442!043e!0432!044b!0439%20!0423!042'
           :         '6%20!041e!041e!041e%20!0022!041a!0420!0418!041f!'
           :         '0422!041e-!041f!0420!041e!0022(1).crl'
   910   63:           [6]
           :         'http://testgost2012.cryptopro.ru/CertEnroll/test'
           :         'gost2012(1).crl'
           :            }
           :           }
           :          }
           :         }
           :        }
           :       }
   975  218:     SEQUENCE {
   978    8:      OBJECT IDENTIFIER
           :       authorityInfoAccess (1 3 6 1 5 5 7 1 1)
   988  205:      OCTET STRING, encapsulates {
```

```
 991  202:       SEQUENCE {
 994   68:        SEQUENCE {
 996    8:         OBJECT IDENTIFIER
         :          caIssuers (1 3 6 1 5 5 7 48 2)
1006   56:         [6]
         :         'http://testgost2012.cryptopro.ru/CertEnroll/root'
         :         '2018.crt'
         :          }
1064   63:        SEQUENCE {
1066    8:         OBJECT IDENTIFIER
         :          ocsp (1 3 6 1 5 5 7 48 1)
1076   51:         [6]
         :         'http://testgost2012.cryptopro.ru/ocsp2012g/ocsp.'
         :         'srf'
         :          }
1129   65:        SEQUENCE {
1131    8:         OBJECT IDENTIFIER
         :          ocsp (1 3 6 1 5 5 7 48 1)
1141   53:         [6]
         :         'http://testgost2012.cryptopro.ru/ocsp2012gst/ocs'
         :         'p.srf'
         :          }
         :         }
         :        }
         :       }
         :      }
         :     }
         :    }
1196   10:  SEQUENCE {
1198    8:   OBJECT IDENTIFIER
         :    gost2012Signature256 (1 2 643 7 1 1 3 2)
         :    }
1208   65:  BIT STRING
         : 21 ee 3b e1 fd 0f 36 90 92 c4 a2 35 26 e8 dc 4e
         : b8 ef 89 40 70 d2 91 39 bc 79 a6 e2 f7 c1 06 bd
         : d5 d6 ff 72 a5 6c f2 c0 c3 75 e9 ca 67 81 c1 93
         : 96 b4 bd 18 12 4c 37 f7 d9 73 d6 4c 8a a6 c4 0a
         :   }
```

The responder's certificate private key (little endian):

```
0000000000: cb 73 0c 81 6f ac 6d 81 9f 82 ae 15 a9 08 12 17
0000000010: d3 1b 97 64 b7 1c 34 0d d3 dd 90 1f 15 8c 9b 06
```

The responder's certificate:

```
0000000000: 30 82 04 b2 30 82 04 5f a0 03 02 01 02 02 13 7c
0000000010: 00 03 d9 02 ec f9 34 3e c8 aa d6 59 00 01 00 03
0000000020: d9 02 30 0a 06 08 2a 85 03 07 01 01 03 02 30 82
0000000030: 01 0a 31 18 30 16 06 05 2a 85 03 64 01 12 0d 31
0000000040: 32 33 34 35 36 37 38 39 30 31 32 33 31 1a 30 18
0000000050: 06 08 2a 85 03 03 81 03 01 01 12 0c 30 30 31 32
0000000060: 33 34 35 36 37 38 39 30 31 2f 30 2d 06 03 55 04
0000000070: 09 0c 26 d1 83 d0 bb 2e 20 d0 a1 d1 83 d1 89 d1
```

```
  0000000080: 91 d0 b2 d1 81 d0 ba d0 b8 d0 b9 20 d0 b2 d0 b0
  0000000090: d0 bb 20 d0 b4 2e 20 31 38 31 0b 30 09 06 03 55
  00000000A0: 04 06 13 02 52 55 31 19 30 17 06 03 55 04 08 0c
  00000000B0: 10 d0 b3 2e 20 d0 9c d0 be d1 81 d0 ba d0 b2 d0
  00000000C0: b0 31 15 30 13 06 03 55 04 07 0c 0c d0 9c d0 be
  00000000D0: d1 81 d0 ba d0 b2 d0 b0 31 25 30 23 06 03 55 04
  00000000E0: 0a 0c 1c d0 9e d0 9e d0 9e 20 22 d0 9a d0 a0 d0
  00000000F0: 98 d0 9f d0 a2 d0 9e 2d d0 9f d0 a0 d0 9e 22 31
  0000000100: 3b 30 39 06 03 55 04 03 0c 32 d0 a2 d0 b5 d1 81
  0000000110: d1 82 d0 be d0 b2 d1 8b d0 b9 20 d0 a3 d0 a6 20
  0000000120: d0 9e d0 9e d0 9e 20 22 d0 9a d0 a0 d0 98 d0 9f
  0000000130: d0 a2 d0 9e 2d d0 9f d0 a0 d0 9e 22 30 1e 17 0d
  0000000140: 32 31 30 39 33 30 31 33 32 34 30 36 5a 17 0d 32
  0000000150: 31 31 32 33 30 31 33 33 34 30 36 5a 30 44 31 20
  0000000160: 30 1e 06 03 55 04 03 13 17 49 4b 45 20 49 6e 74
  0000000170: 65 72 6f 70 20 54 65 73 74 20 53 65 72 76 65 72
  0000000180: 31 13 30 11 06 03 55 04 0a 13 0a 45 4c 56 49 53
  0000000190: 2d 50 4c 55 53 31 0b 30 09 06 03 55 04 06 13 02
  00000001A0: 52 55 30 66 30 1f 06 08 2a 85 03 07 01 01 01 01
  00000001B0: 30 13 06 07 2a 85 03 02 02 24 00 06 08 2a 85 03
  00000001C0: 07 01 01 02 02 03 43 00 04 40 5b b3 14 3e f4 70
  00000001D0: c1 70 d7 f3 27 25 d8 53 7c e6 de 6d 8c 29 f6 b2
  00000001E0: 32 64 56 dc b1 77 f2 3d fa f4 2a 5c f3 74 86 7f
  00000001F0: 04 72 51 c1 cf b3 43 36 f5 95 a2 af 05 47 57 1a
  0000000200: 55 c0 78 a4 9d 64 26 b8 61 14 a3 82 02 59 30 82
  0000000210: 02 55 30 0e 06 03 55 1d 0f 01 01 ff 04 04 03 02
  0000000220: 05 a0 30 13 06 03 55 1d 25 04 0c 30 0a 06 08 2b
  0000000230: 06 01 05 05 07 03 11 30 1d 06 03 55 1d 0e 04 16
  0000000240: 04 14 e0 d3 f0 09 ad ce 6c a5 47 ba 9b f7 a6 a5
  0000000250: 1b 06 14 ba a5 43 30 1f 06 03 55 1d 23 04 18 30
  0000000260: 16 80 14 9b 85 5e fb 81 dc 4d 59 07 51 63 cf be
  0000000270: df da 2c 7f c9 44 3c 30 82 01 0f 06 03 55 1d 1f
  0000000280: 04 82 01 06 30 82 01 02 30 81 ff a0 81 fc a0 81
  0000000290: f9 86 81 b5 68 74 74 70 3a 2f 2f 74 65 73 74 67
  00000002A0: 6f 73 74 32 30 31 32 2e 63 72 79 70 74 6f 70 72
  00000002B0: 6f 2e 72 75 2f 43 65 72 74 45 6e 72 6f 6c 6c 2f
  00000002C0: 21 30 34 32 32 21 30 34 33 35 21 30 34 34 31 21
  00000002D0: 30 34 34 32 21 30 34 33 65 21 30 34 33 32 21 30
  00000002E0: 34 34 62 21 30 34 33 39 25 32 30 21 30 34 32 33
  00000002F0: 21 30 34 32 36 25 32 30 21 30 34 31 65 21 30 34
  0000000300: 31 65 21 30 34 31 65 25 32 30 21 30 30 32 32 21
  0000000310: 30 34 31 61 21 30 34 32 30 21 30 34 31 38 21 30
  0000000320: 34 31 66 21 30 34 32 32 21 30 34 31 65 2d 21 30
  0000000330: 34 31 66 21 30 34 32 30 21 30 34 31 65 21 30 30
  0000000340: 32 32 28 31 29 2e 63 72 6c 86 3f 68 74 74 70 3a
  0000000350: 2f 2f 74 65 73 74 67 6f 73 74 32 30 31 32 2e 63
  0000000360: 72 79 70 74 6f 70 72 6f 2e 72 75 2f 43 65 72 74
  0000000370: 45 6e 72 6f 6c 6c 2f 74 65 73 74 67 6f 73 74 32
  0000000380: 30 31 32 28 31 29 2e 63 72 6c 30 81 da 06 08 2b
  0000000390: 06 01 05 05 07 01 01 04 81 cd 30 81 ca 30 44 06
  00000003A0: 08 2b 06 01 05 05 07 30 02 86 38 68 74 74 70 3a
  00000003B0: 2f 2f 74 65 73 74 67 6f 73 74 32 30 31 32 2e 63
  00000003C0: 72 79 70 74 6f 70 72 6f 2e 72 75 2f 43 65 72 74
  00000003D0: 45 6e 72 6f 6c 6c 2f 72 6f 6f 74 32 30 31 38 2e
  00000003E0: 63 72 74 30 3f 06 08 2b 06 01 05 05 07 30 01 86
  00000003F0: 33 68 74 74 70 3a 2f 2f 74 65 73 74 67 6f 73 74
  0000000400: 32 30 31 32 2e 63 72 79 70 74 6f 70 72 6f 2e 72
  0000000410: 75 2f 6f 63 73 70 32 30 31 32 67 2f 6f 63 73 70
```

```
0000000420: 2e 73 72 66 30 41 06 08 2b 06 01 05 05 07 30 01
0000000430: 86 35 68 74 74 70 3a 2f 2f 74 65 73 74 67 6f 73
0000000440: 74 32 30 31 32 2e 63 72 79 70 74 6f 70 72 6f 2e
0000000450: 72 75 2f 6f 63 73 70 32 30 31 32 67 73 74 2f 6f
0000000460: 63 73 70 2e 73 72 66 30 0a 06 08 2a 85 03 07 01
0000000470: 01 03 02 03 41 00 a5 39 5f ca 48 e1 c2 93 c1 e0
0000000480: 8a 64 74 0f 6b 86 a2 15 9b 46 29 d0 42 71 4f ce
0000000490: e7 52 d7 d7 3d aa 47 ce cf 52 63 8f 26 b2 17 5f
00000004A0: ad 96 57 76 ea 5f d0 87 bb 12 29 e4 06 0e e1 5f
00000004B0: fd 59 81 fb 34 6d
```

```
   0 1202: SEQUENCE {
   4 1119:  SEQUENCE {
   8    3:   [0] {
  10    1:    INTEGER 2
         :     }
  13   19:   INTEGER
         : 7c 00 03 d9 02 ec f9 34 3e c8 aa d6 59 00 01 00
         : 03 d9 02
  34   10:   SEQUENCE {
  36    8:    OBJECT IDENTIFIER
         :     gost2012Signature256 (1 2 643 7 1 1 3 2)
         :     }
  46  266:   SEQUENCE {
  50   24:    SET {
  52   22:     SEQUENCE {
  54    5:      OBJECT IDENTIFIER '1 2 643 100 1'
  61   13:      NumericString '1234567890123'
         :       }
         :      }
  76   26:    SET {
  78   24:     SEQUENCE {
  80    8:      OBJECT IDENTIFIER '1 2 643 3 131 1 1'
  90   12:      NumericString '001234567890'
         :       }
         :      }
 104   47:    SET {
 106   45:     SEQUENCE {
 108    3:      OBJECT IDENTIFIER
         :       streetAddress (2 5 4 9)
 113   38:      UTF8String 'ул. Сущёвский вал д. 18'
         :       }
         :      }
 153   11:    SET {
 155    9:     SEQUENCE {
 157    3:      OBJECT IDENTIFIER
         :       countryName (2 5 4 6)
 162    2:      PrintableString 'RU'
         :       }
         :      }
 166   25:    SET {
 168   23:     SEQUENCE {
 170    3:      OBJECT IDENTIFIER
         :       stateOrProvinceName (2 5 4 8)
 175   16:      UTF8String 'г. Москва'
         :       }
```

```
        :      }
193   21:    SET {
195   19:     SEQUENCE {
197    3:      OBJECT IDENTIFIER
        :       localityName (2 5 4 7)
202   12:      UTF8String 'Москва'
        :       }
        :      }
216   37:    SET {
218   35:     SEQUENCE {
220    3:      OBJECT IDENTIFIER
        :       organizationName (2 5 4 10)
225   28:      UTF8String 'ООО "КРИПТО-ПРО"'
        :       }
        :      }
255   59:    SET {
257   57:     SEQUENCE {
259    3:      OBJECT IDENTIFIER
        :       commonName (2 5 4 3)
264   50:      UTF8String
        :       'Тестовый УЦ ООО "КРИПТО-ПРО"'
        :       }
        :      }
        :     }
316   30:   SEQUENCE {
318   13:    UTCTime 30/09/2021 13:24:06 GMT
333   13:    UTCTime 30/12/2021 13:34:06 GMT
        :     }
348   68:   SEQUENCE {
350   32:    SET {
352   30:     SEQUENCE {
354    3:      OBJECT IDENTIFIER
        :       commonName (2 5 4 3)
359   23:      PrintableString 'IKE Interop Test Server'
        :       }
        :      }
384   19:    SET {
386   17:     SEQUENCE {
388    3:      OBJECT IDENTIFIER
        :       organizationName (2 5 4 10)
393   10:      PrintableString 'ELVIS-PLUS'
        :       }
        :      }
405   11:    SET {
407    9:     SEQUENCE {
409    3:      OBJECT IDENTIFIER
        :       countryName (2 5 4 6)
414    2:      PrintableString 'RU'
        :       }
        :      }
        :     }
418  102:   SEQUENCE {
420   31:    SEQUENCE {
422    8:     OBJECT IDENTIFIER
        :      gost2012PublicKey256 (1 2 643 7 1 1 1 1)
432   19:     SEQUENCE {
434    7:      OBJECT IDENTIFIER
        :       cryptoProSignXA (1 2 643 2 2 36 0)
```

```
  443    8:       OBJECT IDENTIFIER
          :        gost2012Digest256 (1 2 643 7 1 1 2 2)
          :        }
          :       }
  453   67:     BIT STRING, encapsulates {
  456   64:      OCTET STRING
          : 5b b3 14 3e f4 70 c1 70 d7 f3 27 25 d8 53 7c e6
          : de 6d 8c 29 f6 b2 32 64 56 dc b1 77 f2 3d fa f4
          : 2a 5c f3 74 86 7f 04 72 51 c1 cf b3 43 36 f5 95
          : a2 af 05 47 57 1a 55 c0 78 a4 9d 64 26 b8 61 14
          :       }
          :      }
  522  601:    [3] {
  526  597:     SEQUENCE {
  530   14:      SEQUENCE {
  532    3:       OBJECT IDENTIFIER
          :        keyUsage (2 5 29 15)
  537    1:       BOOLEAN TRUE
  540    4:       OCTET STRING, encapsulates {
  542    2:        BIT STRING 5 unused bits
          :         '101'B
          :         }
          :        }
  546   19:      SEQUENCE {
  548    3:       OBJECT IDENTIFIER
          :        extKeyUsage (2 5 29 37)
  553   12:       OCTET STRING, encapsulates {
  555   10:        SEQUENCE {
  557    8:         OBJECT IDENTIFIER
          :          ipsecIKE (1 3 6 1 5 5 7 3 17)
          :          }
          :         }
          :        }
  567   29:      SEQUENCE {
  569    3:       OBJECT IDENTIFIER
          :        subjectKeyIdentifier (2 5 29 14)
  574   22:       OCTET STRING, encapsulates {
  576   20:        OCTET STRING
          : e0 d3 f0 09 ad ce 6c a5 47 ba 9b f7 a6 a5 1b 06
          : 14 ba a5 43
          :         }
          :        }
  598   31:      SEQUENCE {
  600    3:       OBJECT IDENTIFIER
          :        authorityKeyIdentifier (2 5 29 35)
  605   24:       OCTET STRING, encapsulates {
  607   22:        SEQUENCE {
  609   20:         [0]
          : 9b 85 5e fb 81 dc 4d 59 07 51 63 cf be df dA 2C
          : 7f C9 44 3c
          :          }
          :         }
          :        }
  631  271:      SEQUENCE {
  635    3:       OBJECT IDENTIFIER
          :        cRLDistributionPoints (2 5 29 31)
  640  262:       OCTET STRING, encapsulates {
  644  258:        SEQUENCE {
```

```
 648  255:        SEQUENCE {
 651  252:         [0] {
 654  249:          [0] {
 657  181:           [6]
         :         'http://testgost2012.cryptopro.ru/CertEnroll/!042'
         :         '2!0435!0441!0442!043e!0432!044b!0439%20!0423!042'
         :         '6%20!041e!041e!041e%20!0022!041a!0420!0418!041f!'
         :         '0422!041e-!041f!0420!041e!0022(1).crl'
 841   63:           [6]
         :         'http://testgost2012.cryptopro.ru/CertEnroll/test'
         :         'gost2012(1).crl'
         :            }
         :           }
         :          }
         :         }
         :        }
         :       }
 906  218:     SEQUENCE {
 909    8:      OBJECT IDENTIFIER
         :       authorityInfoAccess (1 3 6 1 5 5 7 1 1)
 919  205:      OCTET STRING, encapsulates {
 922  202:       SEQUENCE {
 925   68:        SEQUENCE {
 927    8:         OBJECT IDENTIFIER
         :          caIssuers (1 3 6 1 5 5 7 48 2)
 937   56:         [6]
         :         'http://testgost2012.cryptopro.ru/CertEnroll/root'
         :         '2018.crt'
         :          }
 995   63:        SEQUENCE {
 997    8:         OBJECT IDENTIFIER
         :          ocsp (1 3 6 1 5 5 7 48 1)
1007   51:         [6]
         :         'http://testgost2012.cryptopro.ru/ocsp2012g/ocsp.'
         :          'srf'
         :          }
1060   65:        SEQUENCE {
1062    8:         OBJECT IDENTIFIER
         :          ocsp (1 3 6 1 5 5 7 48 1)
1072   53:         [6]
         :         'http://testgost2012.cryptopro.ru/ocsp2012gst/ocs'
         :          'p.srf'
         :          }
         :         }
         :        }
         :       }
         :      }
         :     }
         :    }
1127   10:  SEQUENCE {
1129    8:   OBJECT IDENTIFIER
         :    gost2012Signature256 (1 2 643 7 1 1 3 2)
         :    }
1139   65:  BIT STRING
         : a5 39 5f ca 48 e1 c2 93 c1 e0 8a 64 74 0f 6b 86
         : a2 15 9b 46 29 d0 42 71 4f ce e7 52 d7 d7 3d aa
         : 47 ce cf 52 63 8f 26 b2 17 5f ad 96 57 76 ea 5f
```

```
           : d0 87 bb 12 29 e4 06 0e e1 5f fd 59 81 fb 34 6d
           :   }
```

CA certificate:

```
  0000000000: 30 82 05 1c 30 82 04 c9 a0 03 02 01 02 02 10 3b
  0000000010: 20 8a e5 fd 46 68 86 49 a0 50 fa af a8 83 93 30
  0000000020: 0a 06 08 2a 85 03 07 01 01 03 02 30 82 01 0a 31
  0000000030: 18 30 16 06 05 2a 85 03 64 01 12 0d 31 32 33 34
  0000000040: 35 36 37 38 39 30 31 32 33 31 1a 30 18 06 08 2a
  0000000050: 85 03 03 81 03 01 01 12 0c 30 30 31 32 33 34 35
  0000000060: 36 37 38 39 30 31 2f 30 2d 06 03 55 04 09 0c 26
  0000000070: d1 83 d0 bb 2e 20 d0 a1 d1 83 d1 89 d1 91 d0 b2
  0000000080: d1 81 d0 ba d0 b8 d0 b9 20 d0 b2 d0 b0 d0 bb 20
  0000000090: d0 b4 2e 20 31 38 31 0b 30 09 06 03 55 04 06 13
  00000000A0: 02 52 55 31 19 30 17 06 03 55 04 08 0c 10 d0 b3
  00000000B0: 2e 20 d0 9c d0 be d1 81 d0 ba d0 b2 d0 b0 31 15
  00000000C0: 30 13 06 03 55 04 07 0c 0c d0 9c d0 be d1 81 d0
  00000000D0: ba d0 b2 d0 b0 31 25 30 23 06 03 55 04 0a 0c 1c
  00000000E0: d0 9e d0 9e d0 9e 20 22 d0 9a d0 a0 d0 98 d0 9f
  00000000F0: d0 a2 d0 9e 2d d0 9f d0 a0 d0 9e 22 31 3b 30 39
  0000000100: 06 03 55 04 03 0c 32 d0 a2 d0 b5 d1 81 d1 82 d0
  0000000110: be d0 b2 d1 8b d0 b9 20 d0 a3 d0 a6 20 d0 9e d0
  0000000120: 9e d0 9e 20 22 d0 9a d0 a0 d0 98 d0 9f d0 a2 d0
  0000000130: 9e 2d d0 9f d0 a0 d0 9e 22 30 1e 17 0d 31 38 30
  0000000140: 39 31 32 31 30 31 39 33 30 5a 17 0d 32 33 30 39
  0000000150: 31 32 31 30 32 38 35 35 5a 30 82 01 0a 31 18 30
  0000000160: 16 06 05 2a 85 03 64 01 12 0d 31 32 33 34 35 36
  0000000170: 37 38 39 30 31 32 33 31 1a 30 18 06 08 2a 85 03
  0000000180: 03 81 03 01 01 12 0c 30 30 31 32 33 34 35 36 37
  0000000190: 38 39 30 31 2f 30 2d 06 03 55 04 09 0c 26 d1 83
  00000001A0: d0 bb 2e 20 d0 a1 d1 83 d1 89 d1 91 d0 b2 d1 81
  00000001B0: d0 ba d0 b8 d0 b9 20 d0 b2 d0 b0 d0 bb 20 d0 b4
  00000001C0: 2e 20 31 38 31 0b 30 09 06 03 55 04 06 13 02 52
  00000001D0: 55 31 19 30 17 06 03 55 04 08 0c 10 d0 b3 2e 20
  00000001E0: d0 9c d0 be d1 81 d0 ba d0 b2 d0 b0 31 15 30 13
  00000001F0: 06 03 55 04 07 0c 0c d0 9c d0 be d1 81 d0 ba d0
  0000000200: b2 d0 b0 31 25 30 23 06 03 55 04 0a 0c 1c d0 9e
  0000000210: d0 9e d0 9e 20 22 d0 9a d0 a0 d0 98 d0 9f d0 a2
  0000000220: d0 9e 2d d0 9f d0 a0 d0 9e 22 31 3b 30 39 06 03
  0000000230: 55 04 03 0c 32 d0 a2 d0 b5 d1 81 d1 82 d0 be d0
  0000000240: b2 d1 8b d0 b9 20 d0 a3 d0 a6 20 d0 9e d0 9e d0
  0000000250: 9e 20 22 d0 9a d0 a0 d0 98 d0 9f d0 a2 d0 9e 2d
  0000000260: d0 9f d0 a0 d0 9e 22 30 66 30 1f 06 08 2a 85 03
  0000000270: 07 01 01 01 01 30 13 06 07 2a 85 03 02 02 23 01
  0000000280: 06 08 2a 85 03 07 01 01 02 02 03 43 00 04 40 98
  0000000290: 1f fd a9 50 cd 21 86 30 f4 59 06 72 a9 d6 3d 6b
  00000002A0: c0 33 82 06 46 37 e3 dc 21 4a b1 f8 9f b7 56 ec
  00000002B0: a5 2d b5 81 87 b6 9d c2 2e df fd 09 33 53 9c 18
  00000002C0: 32 ac d7 42 2e 09 a5 f4 36 a3 a5 c1 d2 22 f0 a3
  00000002D0: 82 01 fe 30 82 01 fa 30 36 06 05 2a 85 03 64 6f
  00000002E0: 04 2d 0c 2b 22 d0 9a d1 80 d0 b8 d0 bf d1 82 d0
  00000002F0: be d0 9f d1 80 d0 be 20 43 53 50 22 20 28 d0 b2
  0000000300: d0 b5 d1 80 d1 81 d0 b8 d1 8f 20 34 2e 30 29 30
  0000000310: 82 01 21 06 05 2a 85 03 64 70 04 82 01 16 30 82
  0000000320: 01 12 0c 2b 22 d0 9a d1 80 d0 b8 d0 bf d1 82 d0
```

```
0000000330: be d0 9f d1 80 d0 be 20 43 53 50 22 20 28 d0 b2
0000000340: d0 b5 d1 80 d1 81 d0 b8 d1 8f 20 34 2e 30 29 0c
0000000350: 41 d0 a3 d0 b4 d0 be d1 81 d1 82 d0 be d0 b2 d0
0000000360: b5 d1 80 d1 8f d1 8e d1 89 d0 b8 d0 b9 20 d1 86
0000000370: d0 b5 d0 bd d1 82 d1 80 20 22 d0 9a d1 80 d0 b8
0000000380: d0 bf d1 82 d0 be d0 9f d1 80 d0 be 20 d0 a3 d0
0000000390: a6 22 0c 4f d0 a1 d0 b5 d1 80 d1 82 d0 b8 d1 84
00000003A0: d0 b8 d0 ba d0 b0 d1 82 20 d1 81 d0 be d0 be d1
00000003B0: 82 d0 b2 d0 b5 d1 82 d1 81 d1 82 d0 b2 d0 b8 d1
00000003C0: 8f 20 e2 84 96 20 d0 a1 d0 a4 2f 30 30 30 2d 30
00000003D0: 30 30 30 20 d0 be d1 82 20 30 30 2e 30 30 2e 30
00000003E0: 30 30 30 0c 4f d0 a1 d0 b5 d1 80 d1 82 d0 b8 d1
00000003F0: 84 d0 b8 d0 ba d0 b0 d1 82 20 d1 81 d0 be d0 be
0000000400: d1 82 d0 b2 d0 b5 d1 82 d1 81 d1 82 d0 b2 d0 b8
0000000410: d1 8f 20 e2 84 96 20 d0 a1 d0 a4 2f 30 30 30 2d
0000000420: 30 30 30 30 20 d0 be d1 82 20 30 30 2e 30 30 2e
0000000430: 30 30 30 30 30 0b 06 03 55 1d 0f 04 04 03 02 01
0000000440: 86 30 0f 06 03 55 1d 13 01 01 ff 04 05 30 03 01
0000000450: 01 ff 30 1d 06 03 55 1d 0e 04 16 04 14 9b 85 5e
0000000460: fb 81 dc 4d 59 07 51 63 cf be df da 2c 7f c9 44
0000000470: 3c 30 12 06 09 2b 06 01 04 01 82 37 15 01 04 05
0000000480: 02 03 01 00 01 30 25 06 03 55 1d 20 04 1e 30 1c
0000000490: 30 08 06 06 2a 85 03 64 71 01 30 08 06 06 2a 85
00000004A0: 03 64 71 02 30 06 06 04 55 1d 20 00 30 23 06 09
00000004B0: 2b 06 01 04 01 82 37 15 02 04 16 04 14 c8 da 66
00000004C0: cb b6 97 d2 3e c9 67 1d c2 5b 64 3a ab dc bb cf
00000004D0: 69 30 0a 06 08 2a 85 03 07 01 01 03 02 03 41 00
00000004E0: 3e 95 cd d8 1f 95 bd 09 ab 73 82 f5 04 e0 f2 66
00000004F0: 12 32 82 9b 2b 03 cc 4b c0 b3 73 f8 e7 0d d6 bd
0000000500: 83 c8 27 2d 01 c1 ec ef 65 5d ac 77 fd dd da 9d
0000000510: 04 e2 bf e8 02 7f 87 36 1b cf ac 7a 28 9c 21 fe
```

```
   0 1308: SEQUENCE {
   4 1225:  SEQUENCE {
   8    3:   [0] {
  10    1:    INTEGER 2
         :     }
  13   16:   INTEGER
         : 3b 20 8a e5 fd 46 68 86 49 a0 50 fa af a8 83 93
  31   10:   SEQUENCE {
  33    8:    OBJECT IDENTIFIER
         :     gost2012Signature256 (1 2 643 7 1 1 3 2)
         :     }
  43  266:   SEQUENCE {
  47   24:    SET {
  49   22:     SEQUENCE {
  51    5:      OBJECT IDENTIFIER '1 2 643 100 1'
  58   13:      NumericString '1234567890123'
         :       }
         :      }
  73   26:    SET {
  75   24:     SEQUENCE {
  77    8:      OBJECT IDENTIFIER '1 2 643 3 131 1 1'
  87   12:      NumericString '001234567890'
         :       }
         :      }
```

```
101   47:     SET {
103   45:      SEQUENCE {
105    3:       OBJECT IDENTIFIER
        :        streetAddress (2 5 4 9)
110   38:       UTF8String 'ул. Сущёвский вал д. 18'
        :        }
        :       }
150   11:     SET {
152    9:      SEQUENCE {
154    3:       OBJECT IDENTIFIER
        :        countryName (2 5 4 6)
159    2:       PrintableString 'RU'
        :        }
        :       }
163   25:     SET {
165   23:      SEQUENCE {
167    3:       OBJECT IDENTIFIER
        :        stateOrProvinceName (2 5 4 8)
172   16:       UTF8String 'г. Москва'
        :        }
        :       }
190   21:     SET {
192   19:      SEQUENCE {
194    3:       OBJECT IDENTIFIER
        :        localityName (2 5 4 7)
199   12:       UTF8String 'Москва'
        :        }
        :       }
213   37:     SET {
215   35:      SEQUENCE {
217    3:       OBJECT IDENTIFIER
        :        organizationName (2 5 4 10)
222   28:       UTF8String 'ООО "КРИПТО-ПРО"'
        :        }
        :       }
252   59:     SET {
254   57:      SEQUENCE {
256    3:       OBJECT IDENTIFIER
        :        commonName (2 5 4 3)
261   50:       UTF8String
        :        'Тестовый УЦ ООО "КРИПТО-ПРО"'
        :        }
        :       }
        :      }
313   30:    SEQUENCE {
315   13:     UTCTime 12/09/2018 10:19:30 GMT
330   13:     UTCTime 12/09/2023 10:28:55 GMT
        :      }
345  266:    SEQUENCE {
349   24:     SET {
351   22:      SEQUENCE {
353    5:       OBJECT IDENTIFIER '1 2 643 100 1'
360   13:       NumericString '1234567890123'
        :        }
        :       }
375   26:     SET {
377   24:      SEQUENCE {
379    8:       OBJECT IDENTIFIER '1 2 643 3 131 1 1'
389   12:      NumericString '001234567890'
        :       }
        :      }
403   47:    SET {
405   45:     SEQUENCE {
407    3:      OBJECT IDENTIFIER
        :       streetAddress (2 5 4 9)
412   38:      UTF8String 'ул. Сущёвский вал д. 18'
        :       }
        :      }
452   11:    SET {
454    9:     SEQUENCE {
456    3:      OBJECT IDENTIFIER
        :       countryName (2 5 4 6)
461    2:      PrintableString 'RU'
        :       }
        :      }
465   25:    SET {
467   23:     SEQUENCE {
469    3:      OBJECT IDENTIFIER
        :       stateOrProvinceName (2 5 4 8)
474   16:      UTF8String 'г. Москва'
        :       }
        :      }
492   21:    SET {
494   19:     SEQUENCE {
496    3:      OBJECT IDENTIFIER
        :       localityName (2 5 4 7)
501   12:      UTF8String 'Москва'
        :       }
        :      }
515   37:    SET {
517   35:     SEQUENCE {
519    3:      OBJECT IDENTIFIER
        :       organizationName (2 5 4 10)
524   28:      UTF8String 'ООО "КРИПТО-ПРО"'
        :       }
        :      }
554   59:    SET {
556   57:     SEQUENCE {
558    3:      OBJECT IDENTIFIER
        :       commonName (2 5 4 3)
563   50:      UTF8String
        :       'Тестовый УЦ ООО "КРИПТО-ПРО"'
        :       }
        :      }
        :     }
615  102:   SEQUENCE {
617   31:    SEQUENCE {
619    8:     OBJECT IDENTIFIER
        :      gost2012PublicKey256 (1 2 643 7 1 1 1 1)
629   19:     SEQUENCE {
631    7:      OBJECT IDENTIFIER
        :       cryptoProSignA (1 2 643 2 2 35 1)
640    8:      OBJECT IDENTIFIER
        :       gost2012Digest256 (1 2 643 7 1 1 2 2)
        :       }
        :      }
```

```
  650   67:    BIT STRING, encapsulates {
  653   64:     OCTET STRING
          : 98 1f fd a9 50 cd 21 86 30 f4 59 06 72 a9 d6 3d
          : 6b c0 33 82 06 46 37 e3 dc 21 4a b1 f8 9f b7 56
          : ec a5 2d b5 81 87 b6 9d c2 2e df fd 09 33 53 9c
          : 18 32 ac d7 42 2e 09 a5 f4 36 a3 a5 c1 d2 22 f0
          :      }
          :     }
  719  510:   [3] {
  723  506:    SEQUENCE {
  727   54:     SEQUENCE {
  729    5:      OBJECT IDENTIFIER '1 2 643 100 111'
  736   45:      OCTET STRING, encapsulates {
  738   43:       UTF8String
          :        '"КриптоПро CSP" (версия 4.0)'
          :        }
          :       }
  783  289:     SEQUENCE {
  787    5:      OBJECT IDENTIFIER '1 2 643 100 112'
  794  278:      OCTET STRING, encapsulates {
  798  274:       SEQUENCE {
  802   43:        UTF8String
          :         '"КриптоПро CSP" (версия 4.0)'
  847   65:        UTF8String
          :         'Удостоверяющий центр "КриптоПро УЦ"'
  914   79:        UTF8String
          :         'Сертификат соответствия № СФ/000-0000 от 00.00.'
          :         '0000'
  995   79:        UTF8String
          :         'Сертификат соответствия № СФ/000-0000 от 00.00.'
          :         '0000'
          :         }
          :        }
          :       }
 1076   11:     SEQUENCE {
 1078    3:      OBJECT IDENTIFIER
          :       keyUsage (2 5 29 15)
 1083    4:      OCTET STRING, encapsulates {
 1085    2:       BIT STRING 1 unused bit
          :        '1100001'B
          :        }
          :       }
 1089   15:     SEQUENCE {
 1091    3:      OBJECT IDENTIFIER
          :       basicConstraints (2 5 29 19)
 1096    1:      BOOLEAN TRUE
 1099    5:      OCTET STRING, encapsulates {
 1101    3:       SEQUENCE {
 1103    1:        BOOLEAN TRUE
          :         }
          :        }
          :       }
 1106   29:     SEQUENCE {
 1108    3:      OBJECT IDENTIFIER
          :       subjectKeyIdentifier (2 5 29 14)
 1113   22:      OCTET STRING, encapsulates {
 1115   20:       OCTET STRING
          : 9b 85 5e fb 81 dc 4d 59 07 51 63 cf be df da 2c
```

```
         : 7f c9 44 3c
         :        }
         :       }
1137   18:     SEQUENCE {
1139    9:      OBJECT IDENTIFIER
         :       cAKeyCertIndexPair (1 3 6 1 4 1 311 21 1)
1150    5:      OCTET STRING, encapsulates {
1152    3:       INTEGER 65537
         :        }
         :       }
1157   37:     SEQUENCE {
1159    3:      OBJECT IDENTIFIER
         :       certificatePolicies (2 5 29 32)
1164   30:      OCTET STRING, encapsulates {
1166   28:       SEQUENCE {
1168    8:        SEQUENCE {
1170    6:         OBJECT IDENTIFIER '1 2 643 100 113 1'
         :          }
1178    8:        SEQUENCE {
1180    6:         OBJECT IDENTIFIER '1 2 643 100 113 2'
         :          }
1188    6:        SEQUENCE {
1190    4:         OBJECT IDENTIFIER
         :          anyPolicy (2 5 29 32 0)
         :          }
         :         }
         :        }
         :       }
1196   35:     SEQUENCE {
1198    9:      OBJECT IDENTIFIER
         :       certSrvPreviousCertHash (1 3 6 1 4 1 311 21 2)
1209   22:      OCTET STRING, encapsulates {
1211   20:       OCTET STRING
         : c8 da 66 cb b6 97 d2 3e c9 67 1d c2 5b 64 3a ab
         : dc bb cf 69
         :        }
         :       }
         :      }
         :     }
         :    }
1233   10:  SEQUENCE {
1235    8:   OBJECT IDENTIFIER
         :    gost2012Signature256 (1 2 643 7 1 1 3 2)
         :    }
1245   65:  BIT STRING
         : 3e 95 cd d8 1f 95 bd 09 ab 73 82 f5 04 e0 f2 66
         : 12 32 82 9b 2b 03 cc 4b c0 b3 73 f8 e7 0d d6 bd
         : 83 c8 27 2d 01 c1 ec ef 65 5d ac 77 fd dd da 9d
         : 04 e2 bf e8 02 7f 87 36 1b cf ac 7a 28 9c 21 fe
         :   }
```

This scenario includes four sub-scenarios, which are described below.

A.2.1. Sub-Scenario 1: Establishment of IKE and ESP SAs Using the IKE_SA_INIT and the IKE_AUTH Exchanges

```
Initiator                                 Responder

HDR, SAi1, KEi, Ni [,N+]      --->
                              <---    HDR, N(INVALID_KE_PAYLOAD)

HDR, SAi1, KEi, Ni [,N+]      --->
                              <---    HDR, SAr1, KEr, Nr
                                           [,CERTREQ] [,N+]

HDR, SK {IDi, [CERT,]
     [CERTREQ,] [IDr,] [N+,]
     AUTH, SAi2, TSi, TSr}    --->
                              <---    HDR, SK {IDr, [CERT,] [N+,]
                                           AUTH, SAr2, TSi, TSr}
```

Initiator's actions:

(1) Generates random SPIi for IKE SA

```
00000000: 92 80 e0 82 2e 75 87 78
```

(2) Generates random IKE nonce Ni

```
00000000: 98 44 d5 40 ef 89 46 f4 55 20 0a 55 73 dc ad 73
00000010: dd 2a 6f a8 31 f8 49 05 f5 8e 17 a2 6c cc 01 1f
```

(3) Generates ephemeral private key (512 bit)

```
00000000: 82 fb 1c 90 c3 a3 c2 16 7f 76 15 5d 69 06 f8 47
00000010: 3e fe 83 3e 21 cd e7 a4 e5 cd d9 71 ef d3 c5 db
00000020: 7e de 50 70 48 96 90 01 0c 81 02 b9 4b 56 f6 47
00000030: cb 27 40 25 58 55 80 32 e9 59 17 10 3b 0f eb 3b
```

(4) Computes public key

```
00000000: 89 77 c6 d7 2b 08 5d d5 48 b1 ea 5d 99 c5 03 09
00000010: c6 62 fe d7 7d 84 a4 d8 8b 9b a5 c8 3a 7a 05 86
00000020: e2 0d 8d 9b 5d ce 01 18 e2 d2 da 73 83 ee 30 ad
00000030: 49 88 44 6f bd 18 78 b4 bb da c9 df 1a ca d1 2a
00000040: 05 98 75 da 9e 9a 21 e4 db 71 8f af d1 96 c7 8b
00000050: de 9a b2 98 f7 55 bb 74 38 34 a4 da 47 ab 86 15
00000060: d4 c8 33 70 b7 02 79 b8 7f c2 97 6d 03 8f 2d 08
00000070: d7 ab ac 85 4c bf 5a f6 27 57 ad fe 61 50 5e 45
```

(5) Creates message

```
IKE SA Init
9280E0822E758778.0000000000000000.00000000 IKEv2 R<-I[328]
  SA[52]{
    P[48](#1:IKE::5#){
      Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
                 ENCR_MAGMA_MGM_KTREE,
      PRF=PRF_HMAC_STREEBOG_512,
      KE=GOST3410_2012_512,
         GOST3410_2012_256}},
  KE[136](GOST3410_2012_512){8977C6...505E45},
  NONCE[36]{9844D5...CC011F},
  N[28](NAT_DETECTION_SOURCE_IP){000000...000000},
  N[28](NAT_DETECTION_DESTINATION_IP){7D2124...4E6F10},
  N[8](IKEV2_FRAGMENTATION_SUPPORTED),
  N[12](SIGNATURE_HASH_ALGORITHMS){STREEBOG_256, STREEBOG_512}
```

(6) Sends message, peer receives message

```
       10.111.10.171:54294->10.111.15.45:500 [328]

00000000: 92 80 e0 82 2e 75 87 78 00 00 00 00 00 00 00 00
00000010: 21 20 22 08 00 00 00 00 00 00 01 48 22 00 00 34
00000020: 00 00 00 30 01 01 00 05 03 00 00 08 01 00 00 20
00000030: 03 00 00 08 01 00 00 21 03 00 00 08 02 00 00 09
00000040: 03 00 00 08 04 00 00 22 00 00 00 08 04 00 00 21
00000050: 28 00 00 88 00 22 00 00 89 77 c6 d7 2b 08 5d d5
00000060: 48 b1 ea 5d 99 c5 03 09 c6 62 fe d7 7d 84 a4 d8
00000070: 8b 9b a5 c8 3a 7a 05 86 e2 0d 8d 9b 5d ce 01 18
00000080: e2 d2 da 73 83 ee 30 ad 49 88 44 6f bd 18 78 b4
00000090: bb da c9 df 1a ca d1 2a 05 98 75 da 9e 9a 21 e4
000000A0: db 71 8f af d1 96 c7 8b de 9a b2 98 f7 55 bb 74
000000B0: 38 34 a4 da 47 ab 86 15 d4 c8 33 70 b7 02 79 b8
000000C0: 7f c2 97 6d 03 8f 2d 08 d7 ab ac 85 4c bf 5a f6
000000D0: 27 57 ad fe 61 50 5e 45 29 00 00 24 98 44 d5 40
000000E0: ef 89 46 f4 55 20 0a 55 73 dc ad 73 dd 2a 6f a8
000000F0: 31 f8 49 05 f5 8e 17 a2 6c cc 01 1f 29 00 00 1c
00000100: 00 00 40 04 00 00 00 00 00 00 00 00 00 00 00 00
00000110: 00 00 00 00 00 00 00 00 29 00 00 1c 00 00 40 05
00000120: 7d 21 24 87 89 d7 95 71 bd a2 2d 22 9d 51 d0 71
00000130: e9 4e 6f 10 29 00 00 08 00 00 40 2e 00 00 00 0c
00000140: 00 00 40 2f 00 06 00 07
```

Responder's actions:

(7) Parses received message

```
IKE SA Init
9280E0822E758778.0000000000000000.00000000 IKEv2 I->R[328]
  SA[52]{
    P[48](#1:IKE::5#){
      Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
                 ENCR_MAGMA_MGM_KTREE,
      PRF=PRF_HMAC_STREEBOG_512,
      KE=GOST3410_2012_512,
         GOST3410_2012_256}},
  KE[136](GOST3410_2012_512){8977C6...505E45},
  NONCE[36]{9844D5...CC011F},
  N[28](NAT_DETECTION_SOURCE_IP){000000...000000},
  N[28](NAT_DETECTION_DESTINATION_IP){7D2124...4E6F10},
  N[8](IKEV2_FRAGMENTATION_SUPPORTED),
  N[12](SIGNATURE_HASH_ALGORITHMS){STREEBOG_256, STREEBOG_512}
```

(8) Creates message

```
IKE SA Init
9280E0822E758778.0000000000000000.00000000 IKEv2 I<=R[38]
  N[10](INVALID_KE_PAYLOAD){GOST3410_2012_256}
```

(9) Sends message, peer receives message

```
       10.111.10.171:54294<-10.111.15.45:500 [38]

00000000: 92 80 e0 82 2e 75 87 78 00 00 00 00 00 00 00 00
00000010: 29 20 22 20 00 00 00 00 00 00 00 26 00 00 00 0a
00000020: 00 00 00 11 00 21
```

Initiator's actions:

(10) Parses received message

```
IKE SA Init
9280E0822E758778.0000000000000000.00000000 IKEv2 R=>I[38]
  N[10](INVALID_KE_PAYLOAD){GOST3410_2012_256}}
```

(11) Generates ephemeral private key (256 bit)

```
00000000: b9 7c ac df 01 43 44 dd 54 92 33 63 4a 6e da 64
00000010: 38 5b 6a 9c c0 3c 6c 41 c5 02 eb 63 d1 e6 24 21
```

(12) Computes public key

```
00000000: 7d b0 49 81 88 6d 1b 02 b2 a6 35 c5 8b ea 90 8c
00000010: 3e 16 de e5 43 13 22 0b ad f5 89 9f 7f 85 54 2d
00000020: 3e db 1e de 85 f7 d5 5d 6f 83 c5 d0 31 bd 31 49
00000030: dd 29 c5 16 16 7d ec 86 16 d8 85 e6 e4 50 ab 46
```

(13) Creates message

```
IKE SA Init
9280E0822E758778.0000000000000000.00000000 IKEv2 R<-I[264]
  SA[52]{
    P[48](#1:IKE::5#){
      Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
                 ENCR_MAGMA_MGM_KTREE,
      PRF=PRF_HMAC_STREEBOG_512,
      KE=GOST3410_2012_512,
         GOST3410_2012_256}},
  KE[72](GOST3410_2012_256){7DB049...50AB46},
  NONCE[36]{9844D5...CC011F},
  N[28](NAT_DETECTION_SOURCE_IP){000000...000000},
  N[28](NAT_DETECTION_DESTINATION_IP){7D2124...4E6F10},
  N[8](IKEV2_FRAGMENTATION_SUPPORTED),
  N[12](SIGNATURE_HASH_ALGORITHMS){STREEBOG_256, STREEBOG_512}
```

(14) Sends message, peer receives message

```
       10.111.10.171:54294->10.111.15.45:500 [264]

00000000: 92 80 e0 82 2e 75 87 78 00 00 00 00 00 00 00 00
00000010: 21 20 22 08 00 00 00 00 00 00 01 08 22 00 00 34
00000020: 00 00 00 30 01 01 00 05 03 00 00 08 01 00 00 20
00000030: 03 00 00 08 01 00 00 21 03 00 00 08 02 00 00 09
00000040: 03 00 00 08 04 00 00 22 00 00 00 08 04 00 00 21
00000050: 28 00 00 48 00 21 00 00 7d b0 49 81 88 6d 1b 02
00000060: b2 a6 35 c5 8b ea 90 8c 3e 16 de e5 43 13 22 0b
00000070: ad f5 89 9f 7f 85 54 2d 3e db 1e de 85 f7 d5 5d
00000080: 6f 83 c5 d0 31 bd 31 49 dd 29 c5 16 16 7d ec 86
00000090: 16 d8 85 e6 e4 50 ab 46 29 00 00 24 98 44 d5 40
000000A0: ef 89 46 f4 55 20 0a 55 73 dc ad 73 dd 2a 6f a8
000000B0: 31 f8 49 05 f5 8e 17 a2 6c cc 01 1f 29 00 00 1c
000000C0: 00 00 40 04 00 00 00 00 00 00 00 00 00 00 00 00
000000D0: 00 00 00 00 00 00 00 00 29 00 00 1c 00 00 40 05
000000E0: 7d 21 24 87 89 d7 95 71 bd a2 2d 22 9d 51 d0 71
000000F0: e9 4e 6f 10 29 00 00 08 00 00 40 2e 00 00 00 0c
00000100: 00 00 40 2f 00 06 00 07
```

Responder's actions:

(15) Parses received message

```
IKE SA Init
9280E0822E758778.0000000000000000.00000000 IKEv2 I->R[264]
  SA[52]{
    P[48](#1:IKE::5#){
      Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
                 ENCR_MAGMA_MGM_KTREE,
      PRF=PRF_HMAC_STREEBOG_512,
      KE=GOST3410_2012_512,
         GOST3410_2012_256}},
  KE[72](GOST3410_2012_256){7DB049...50AB46},
  NONCE[36]{9844D5...CC011F},
  N[28](NAT_DETECTION_SOURCE_IP){000000...000000},
  N[28](NAT_DETECTION_DESTINATION_IP){7D2124...4E6F10},
  N[8](IKEV2_FRAGMENTATION_SUPPORTED),
  N[12](SIGNATURE_HASH_ALGORITHMS){STREEBOG_256, STREEBOG_512}
```

(16) Generates random SPIr for IKE SA

```
00000000: db 57 8d 97 de 11 9d 1e
```

(17) Generates random IKE nonce Nr

```
00000000: 6c de 24 c1 2c 0a 10 d5 c3 fe 55 e8 7e 90 30 66
00000010: ee 54 5b 24 1c 3c 01 dd b3 98 06 ae d3 b5 00 48
```

(18) Generates ephemeral private key

```
00000000: 46 fd 19 da 1c 77 e8 4c 12 69 cf c8 a2 2a 0b e9
00000010: 70 db c1 2c 9f 6d 88 0a 70 71 22 03 68 c6 fd 2d
```

(19) Computes public key

```
00000000: 49 c2 40 f6 ac 35 f1 70 a7 c2 37 5e 9a 78 3c 09
00000010: 59 8d 55 3b 30 5b 64 58 db 2f 3c 36 f4 b1 db ad
00000020: ff c8 f4 b2 bd 14 cf 96 5b b2 d6 80 51 69 67 06
00000030: bd 16 39 0e 6d 07 83 e4 9d ed fd 04 f1 9e 07 a2
```

(20) Computes hash of CA public key

```
00000000: 5e 9e 50 5f 58 b0 a5 7a 33 45 83 49 66 0f 1c 3c
00000010: 7a 67 71 98
```

(21) Creates message

```
IKE SA Init
9280E0822E758778.DB578D97DE119D1E.00000000 IKEv2 I<=R[273]
  SA[36]{
    P[32](#1:IKE::3#){
      Encryption=ENCR_MAGMA_MGM_KTREE,
      PRF=PRF_HMAC_STREEBOG_512,
      KE=GOST3410_2012_256}},
  KE[72](GOST3410_2012_256){49C240...9E07A2},
  NONCE[36]{6CDE24...B50048},
  N[28](NAT_DETECTION_SOURCE_IP){A4DCA3...2F5B3F},
  N[28](NAT_DETECTION_DESTINATION_IP){BA7D7A...7AB7C9},
  CERTREQ[25](X.509 Cert){5E9E50...677198},
  N[8](IKEV2_FRAGMENTATION_SUPPORTED),
  N[12](SIGNATURE_HASH_ALGORITHMS){STREEBOG_256, STREEBOG_512}
```

(22) Sends message, peer receives message

```
       10.111.10.171:54294<-10.111.15.45:500 [273]

00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 21 20 22 20 00 00 00 00 00 00 01 11 22 00 00 24
00000020: 00 00 00 20 01 01 00 03 03 00 00 08 01 00 00 21
00000030: 03 00 00 08 02 00 00 09 00 00 00 08 04 00 00 21
00000040: 28 00 00 48 00 21 00 00 49 c2 40 f6 ac 35 f1 70
00000050: a7 c2 37 5e 9a 78 3c 09 59 8d 55 3b 30 5b 64 58
00000060: db 2f 3c 36 f4 b1 db ad ff c8 f4 b2 bd 14 cf 96
00000070: 5b b2 d6 80 51 69 67 06 bd 16 39 0e 6d 07 83 e4
00000080: 9d ed fd 04 f1 9e 07 a2 29 00 00 24 6c de 24 c1
00000090: 2c 0a 10 d5 c3 fe 55 e8 7e 90 30 66 ee 54 5b 24
000000A0: 1c 3c 01 dd b3 98 06 ae d3 b5 00 48 29 00 00 1c
000000B0: 00 00 40 04 a4 dc a3 62 54 e8 4b 53 2b ff e7 d2
000000C0: 26 83 f3 8f 28 2f 5b 3f 26 00 00 1c 00 00 40 05
000000D0: ba 7d 7a b8 48 82 72 f6 30 91 b6 ae 2b dd fb 48
000000E0: ba 7a b7 c9 29 00 00 19 04 5e 9e 50 5f 58 b0 a5
000000F0: 7a 33 45 83 49 66 0f 1c 3c 7a 67 71 98 29 00 00
00000100: 08 00 00 40 2e 00 00 00 0c 00 00 40 2f 00 06 00
00000110: 07
```

Initiator's actions:

(23) Parses received message

```
IKE SA Init
9280E0822E758778.DB578D97DE119D1E.00000000 IKEv2 R=>I[273]
  SA[36]{
    P[32](#1:IKE::3#){
      Encryption=ENCR_MAGMA_MGM_KTREE,
      PRF=PRF_HMAC_STREEBOG_512,
      KE=GOST3410_2012_256}},
  KE[72](GOST3410_2012_256){49C240...9E07A2},
  NONCE[36]{6CDE24...B50048},
  N[28](NAT_DETECTION_SOURCE_IP){A4DCA3...2F5B3F},
  N[28](NAT_DETECTION_DESTINATION_IP){BA7D7A...7AB7C9},
  CERTREQ[25](X.509 Cert){5E9E50...677198},
  N[8](IKEV2_FRAGMENTATION_SUPPORTED),
  N[12](SIGNATURE_HASH_ALGORITHMS){STREEBOG_256, STREEBOG_512}
```

(24) Computes shared key

```
00000000: bd 04 9d 0f 9c 5f 58 af c7 e4 01 bc 18 59 01 7c
00000010: 88 28 f9 f2 9f 33 01 5d 49 9a 7d 14 74 d4 31 ac
```

(25) Computes SKEYSEED

```
00000000: 9b ed 6c 79 64 b3 de 3a e4 9e dd 62 04 5a f0 8b
00000010: 43 88 33 d4 e6 9e 73 16 a1 1a 9e b2 b4 19 13 c5
00000020: d0 6d fb 86 40 11 c3 02 bb e5 a3 b5 e4 4a c4 c0
00000030: 9d 18 c6 94 de c3 c5 14 82 e7 a2 51 fe c4 98 ca
```

(26) Computes SK_d

```
00000000: c2 21 15 fd d3 99 3b 2a 43 60 c4 59 34 b0 be 3f
00000010: 53 ef 6e b1 dd 88 ad 72 55 dd 83 22 5c 6f e1 d6
00000020: 1f 1e ab 06 f9 41 cb c8 ea f9 dc fc 19 a0 2d bf
00000030: 9a 0a 3f 3a 9a 45 1f 08 b6 a9 2c 62 52 b7 26 34
```

(27) Computes SK_ei

```
00000000: 18 4e 4e 0f 36 28 bf 3c 9c 04 8e 93 bf a0 77 53
00000010: 91 34 12 81 42 e6 4e 62 7f db a5 ed 98 60 50 ff
00000020: b4 e1 3e 23
```

(28) Computes SK_er

```
00000000: e9 27 59 2f 09 49 68 1e 0e 62 db c6 19 06 73 13
00000010: cf da 5c 02 27 3e 4a b4 78 98 b4 86 d0 e9 34 f4
00000020: a5 bb 18 2f
```

(29) Computes SK_pi

```
00000000: 30 2c 10 8d 0f 61 47 00 f1 40 4f a9 4f af b5 30
00000010: 11 ba 5f 24 39 32 85 12 4e 7e 71 75 50 15 a6 93
00000020: c3 d0 5e 40 2e 21 8e b1 59 09 cd a4 eb b4 91 68
00000030: 29 42 fe e2 d8 76 8f a6 96 55 1f ab 6c 9b 00 f8
```

(30) Computes SK_pr

```
00000000: 6f 81 72 cb 96 58 fb 0e 17 70 b6 b9 1f a9 69 a9
00000010: fc c7 27 4f b4 e1 85 90 a0 c7 9f f9 72 11 61 2a
00000020: 35 b7 b7 96 d3 6a bb a5 aa b1 b8 34 8d 99 c6 f3
00000030: 2b fc 32 56 c1 94 71 04 55 bd 89 6a bf c3 8b fe
```

(31) Computes prf(SK_pi, IDi)

```
00000000: ce e8 8b d1 7e 3c 83 32 eb d1 29 08 de dc 71 f4
00000010: 8f ba 09 b8 ca 5b 10 e2 f4 44 29 5c 97 7b 26 01
00000020: a4 ba 83 c8 ea 40 92 0f 88 18 bd e7 e1 c9 45 cf
00000030: ff 99 48 05 0d f4 93 a6 cd 54 46 d7 eb 7a 52 94
```

(32) Uses private key for signing (little endian)

```
00000000: 76 E9 DD B3 F3 A2 08 A2 4E A5 81 9C AE 41 DA B4
00000010: 77 3C 1D D5 DC EB AF E6 58 B1 47 D2 D8 29 CE 71
00000020: 18 A9 85 5D 28 5B 3C E3 23 BD 80 AC 2F 00 CC B6
00000030: 61 4C 42 A1 65 61 02 CF 33 EB 1F 5F 02 CE 8A B9
```

(33) Uses random number for signing

```
00000000: 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
00000010: 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
00000020: 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
00000030: 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
```

(34) Computes signature using algorithm id-tc26-signwithdigest-gost3410-12-512

```
00000000: 6a 3e 59 0d 72 1e 55 a3 c0 d1 2f 8a 9b 4e 44 10
00000010: 58 59 bd 62 9e e7 12 31 e5 7d 01 53 f3 84 40 dd
00000020: ac 73 ed 09 3a 10 d9 6e 7f eb 80 6c 11 9e 91 f3
00000030: 7c 3c b0 55 f7 4b ec 0e 78 36 10 95 02 09 86 b3
00000040: 27 04 2a 83 3c 89 36 1b 73 cf 7b c9 e0 df a2 07
00000050: 12 1e 69 52 4d 89 1b de 6e 48 d1 34 fa 21 78 22
00000060: 88 2e 30 86 c0 80 0a 2d 74 af 08 ff 35 75 a5 79
00000070: e3 85 40 22 6b a8 42 f6 72 24 bf 29 87 58 a8 20
```

(35) Computes K1i (i1 = 0)

```
00000000: 3c 57 d7 c8 9f 50 98 fc 86 81 d6 8a 4e 5d 83 c6
00000010: 1e 42 e6 e7 60 67 05 8d f5 2e 10 13 12 15 32 58
```

(36) Computes K2i (i2 = 0)

```
00000000: 0b 88 0a 1b c8 3e 61 79 82 08 db 13 31 08 63 3c
00000010: 17 62 17 cb 7d 18 ce 70 37 84 85 f4 89 49 d0 06
```

(37) Computes K3i (i3 = 0)

```
00000000: 18 63 41 67 49 6e cf 48 56 71 4d aa 42 63 5c 11
00000010: 2e 26 5b e2 7b c7 53 a4 09 82 e5 5a 7e f4 65 4d
```

(38) Selects SPI for incoming ESP SA

```
00000000: 6c 0c a5 70
```

(39) Computes hash of CA public key

```
00000000: 5e 9e 50 5f 58 b0 a5 7a 33 45 83 49 66 0f 1c 3c
00000010: 7a 67 71 98
```

(40) Creates message splitting it into 4 fragments

```
IKE SA Auth
#9280E0822E758778.DB578D97DE119D1E.00000001 IKEv2 R<-I[1847]
  E[1819]->4*EF[...]{
    IDi[78](DN){CN=IKE Interop Test Client,O=ELVIS-PLUS,C=RU},
    CERT[1280](X.509 Cert){308204...A6C40A},
    CERTREQ[25](X.509 Cert){5E9E50...677198},
    IDr[78](DN){CN=IKE Interop Test Server,O=ELVIS-PLUS,C=RU},
    AUTH[149](Sig){id-tc26-signwithdigest-gost3410-12-512[12]:
              6A3E59...58A820},
    N[8](INITIAL_CONTACT),
    N[12](SET_WINDOW_SIZE){4},
    CP[16](REQUEST){IP4.Address[0], IP4.DNS[0]},
    SA[56]{
      P[52](#1:ESP:6C0CA570:5#){
        Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
                   ENCR_MAGMA_MGM_KTREE,
                   ENCR_KUZNYECHIK_MGM_MAC_KTREE,
                   ENCR_MAGMA_MGM_MAC_KTREE,
        ESN=Off}},
    TSi[40](2#){10.111.10.171:icmp:8.0, 0.0.0.0-255.255.255.255},
    TSr[40](2#){10.0.0.2:icmp:8.0, 10.0.0.0-10.0.0.255},
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(41) Composes MGM nonce (fragment 1)

```
00000000: 00 00 00 00 b4 e1 3e 23
```

(42) Composes AAD (fragment 1)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 08 00 00 00 01 00 00 02 20 23 00 02 04
00000020: 00 01 00 04
```

(43) Composes plaintext (fragment 1)

```
00000000: 25 00 00 4e 09 00 00 00 30 44 31 20 30 1e 06 03
00000010: 55 04 03 13 17 49 4b 45 20 49 6e 74 65 72 6f 70
00000020: 20 54 65 73 74 20 43 6c 69 65 6e 74 31 13 30 11
00000030: 06 03 55 04 0a 13 0a 45 4c 56 49 53 2d 50 4c 55
00000040: 53 31 0b 30 09 06 03 55 04 06 13 02 52 55 26 00
00000050: 05 00 04 30 82 04 f7 30 82 04 a4 a0 03 02 01 02
00000060: 02 13 7c 00 03 da a8 9e 1e ff 9e 79 05 fb bb 00
00000070: 01 00 03 da a8 30 0a 06 08 2a 85 03 07 01 01 03
00000080: 02 30 82 01 0a 31 18 30 16 06 05 2a 85 03 64 01
00000090: 12 0d 31 32 33 34 35 36 37 38 39 30 31 32 33 31
000000A0: 1a 30 18 06 08 2a 85 03 03 81 03 01 01 12 0c 30
000000B0: 30 31 32 33 34 35 36 37 38 39 30 31 2f 30 2d 06
000000C0: 03 55 04 09 0c 26 d1 83 d0 bb 2e 20 d0 a1 d1 83
000000D0: d1 89 d1 91 d0 b2 d1 81 d0 ba d0 b8 d0 b9 20 d0
000000E0: b2 d0 b0 d0 bb 20 d0 b4 2e 20 31 38 31 0b 30 09
000000F0: 06 03 55 04 06 13 02 52 55 31 19 30 17 06 03 55
00000100: 04 08 0c 10 d0 b3 2e 20 d0 9c d0 be d1 81 d0 ba
00000110: d0 b2 d0 b0 31 15 30 13 06 03 55 04 07 0c 0c d0
00000120: 9c d0 be d1 81 d0 ba d0 b2 d0 b0 31 25 30 23 06
00000130: 03 55 04 0a 0c 1c d0 9e d0 9e d0 9e 20 22 d0 9a
00000140: d0 a0 d0 98 d0 9f d0 a2 d0 9e 2d d0 9f d0 a0 d0
00000150: 9e 22 31 3b 30 39 06 03 55 04 03 0c 32 d0 a2 d0
00000160: b5 d1 81 d1 82 d0 be d0 b2 d1 8b d0 b9 20 d0 a3
00000170: d0 a6 20 d0 9e d0 9e d0 9e 20 22 d0 9a d0 a0 d0
00000180: 98 d0 9f d0 a2 d0 9e 2d d0 9f d0 a0 d0 9e 22 30
00000190: 1e 17 0d 32 31 31 30 30 31 30 36 31 30 31 30 5a
000001A0: 17 0d 32 32 30 31 30 31 30 36 32 30 31 30 5a 30
000001B0: 44 31 20 30 1e 06 03 55 04 03 13 17 49 4b 45 20
000001C0: 49 6e 74 65 72 6f 70 20 54 65 73 74 20 43 6c 69
000001D0: 65 6e 74 31 13 30 11 06 03 55 04 0a 13 0a 45 4c
000001E0: 56 49 53 2d 50 4c 55 53 31 0b 30 00
```

(44) Encrypts plaintext using K3i as K_msg, resulting in ciphertext (fragment 1)

```
00000000: 03 45 60 11 15 25 f5 45 bb 0e f4 25 26 e2 14 8c
00000010: a7 01 82 f6 9c 6e 42 f1 a3 9b 9e ac a6 dd 0d 9c
00000020: ff 79 15 ed b9 0c 81 a0 b4 29 61 fb 55 1b c1 73
00000030: 4d de 1f b2 5f 1f cb 84 5d 12 24 85 52 c4 f2 1d
00000040: 01 a7 92 ad 55 4d 90 d0 58 d2 1a 5e f6 dc 4e 73
00000050: d4 9b 08 66 d7 64 de 10 e6 75 69 20 e3 7b 6c f0
00000060: 4b 8b ff 60 39 f1 19 31 72 dd c1 09 33 5b 1d 56
00000070: ee 0c 1c 42 d7 f3 04 d3 5b 9a 6e cf 7f b3 1f ac
00000080: 34 a6 ee e0 ac 87 b8 88 99 75 a6 ae dc b5 30 38
00000090: eb 3d 48 fd cc 69 64 f8 c6 61 ce e9 e1 24 ba aa
000000A0: 25 5e e6 ea 8b 0c ef 20 31 bf a9 ae 6d e2 82 d4
000000B0: ab 2c d7 af ca 62 fe bd 7c 8f a9 dc d3 63 05 d7
000000C0: ba 92 56 66 44 ad 5d 9d 1e 9a 27 2e 22 6e 5b 0c
000000D0: af 84 6b c6 a7 cf ca 72 f8 8e d3 a1 bc d4 7c 5b
000000E0: 7e 26 7f b3 05 d8 62 ef ad d6 07 70 d7 4b 33 e4
000000F0: 26 84 e6 eb 5b 65 5c a7 71 29 45 15 d9 b0 83 6a
00000100: 52 5f a9 d8 dd f1 d8 62 c7 d7 3d e9 69 0e c5 b1
00000110: e1 de 20 6c 3d 5f f7 f7 9f f6 a5 7b 4d a5 4e e9
00000120: b4 c4 c2 7d cc 43 62 77 57 37 d3 40 48 b2 c0 5b
00000130: 48 ab d0 94 79 ef 3d 04 e3 d8 6d 42 56 ed cd 94
00000140: b4 23 2c fa f0 6b 39 ad 41 a3 b3 8f ec b8 6c ef
00000150: e1 98 3a b2 fb a8 fd 21 96 8a bf 3a 65 47 8a e9
00000160: 69 60 44 02 2c ec 7a 86 74 fe 1d 9b 08 5e b8 5e
00000170: f8 ca 37 20 5f a7 74 8c 12 88 f2 d8 9e d4 94 29
00000180: c2 db f9 fb 35 a0 cf 21 2b da 8b 9e cc 52 84 eb
00000190: c4 12 39 3e e6 18 fb f7 57 6c b5 1e 10 3d 11 9c
000001A0: 29 9c 41 73 69 d8 d0 9d 71 2b 77 66 87 65 51 19
000001B0: db 27 a0 dd aa 64 ba fd c0 5f e1 4e da 7c 20 fc
000001C0: 8c 13 ab 2d c2 9c 37 9d 7e 51 cb 29 03 10 52 dc
000001D0: f8 09 61 cc 12 9a a0 8e 1b e4 52 f8 72 bd 7a 86
000001E0: db 93 7c 55 b8 1e 7f 21 d4 e6 02 f2
```

(45) Computes ICV using K3i as K_msg (fragment 1)

```
00000000: b1 51 cd e6 dc 64 12 1c
```

(46) Composes IV (fragment 1)

```
00000000: 00 00 00 00 00 00 00 00
```

(47) Composes MGM nonce (fragment 2)

```
00000000: 00 00 00 01 b4 e1 3e 23
```

(48) Composes AAD (fragment 2)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 08 00 00 00 01 00 00 02 20 00 00 02 04
00000020: 00 02 00 04
```

(49) Composes plaintext (fragment 2)

```
00000000: 09 06 03 55 04 06 13 02 52 55 30 81 aa 30 21 06
00000010: 08 2a 85 03 07 01 01 01 02 30 15 06 09 2a 85 03
00000020: 07 01 02 01 02 01 06 08 2a 85 03 07 01 01 02 03
00000030: 03 81 84 00 04 81 80 ee 2f 0a 0e 09 1e 7e 04 ef
00000040: ba 5b 62 a2 52 86 e1 9c 24 50 30 50 b0 b4 8a 37
00000050: 35 b5 fc af 28 94 ec b5 9b 92 41 5b 69 e2 c9 ba
00000060: 24 de 6a 72 c4 ef 44 bb 89 a1 05 14 1b 87 3d 6a
00000070: a3 72 3e 17 ca 7f 39 28 ce 16 8b dd 07 52 87 6a
00000080: 0d 77 42 6d 99 2b 46 2c fd 4b b2 7c d7 c7 17 08
00000090: 12 54 63 47 9d 14 3d 61 ed f2 95 ab 11 80 69 02
000000A0: a7 66 60 50 7e a4 53 6d ad 01 49 b2 16 8a 95 1d
000000B0: cf 1a 57 93 56 14 5e a3 82 02 59 30 82 02 55 30
000000C0: 0e 06 03 55 1d 0f 01 01 ff 04 04 03 02 05 a0 30
000000D0: 13 06 03 55 1d 25 04 0c 30 0a 06 08 2b 06 01 05
000000E0: 05 07 03 11 30 1d 06 03 55 1d 0e 04 16 04 14 40
000000F0: 81 b1 d1 18 75 f0 da 6b 3c 50 5f cd 73 1d d9 77
00000100: f2 d7 c1 30 1f 06 03 55 1d 23 04 18 30 16 80 14
00000110: 9b 85 5e fb 81 dc 4d 59 07 51 63 cf be df da 2c
00000120: 7f c9 44 3c 30 82 01 0f 06 03 55 1d 1f 04 82 01
00000130: 06 30 82 01 02 30 81 ff a0 81 fc a0 81 f9 86 81
00000140: b5 68 74 74 70 3a 2f 2f 74 65 73 74 67 6f 73 74
00000150: 32 30 31 32 2e 63 72 79 70 74 6f 70 72 6f 2e 72
00000160: 75 2f 43 65 72 74 45 6e 72 6f 6c 6c 2f 21 30 34
00000170: 32 32 21 30 34 33 35 21 30 34 34 31 21 30 34 34
00000180: 32 21 30 34 33 65 21 30 34 33 32 21 30 34 34 62
00000190: 21 30 34 33 39 25 32 30 21 30 34 32 33 21 30 34
000001A0: 32 36 25 32 30 21 30 34 31 65 21 30 34 31 65 21
000001B0: 30 34 31 65 25 32 30 21 30 30 32 32 21 30 34 31
000001C0: 61 21 30 34 32 30 21 30 34 31 38 21 30 34 31 66
000001D0: 21 30 34 32 32 21 30 34 31 65 2d 21 30 34 31 66
000001E0: 21 30 34 32 30 21 30 34 31 65 21 00
```

(50) Encrypts plaintext using K3i as K_msg, resulting in ciphertext (fragment 2)

```
00000000: 3c b1 b4 aa 04 56 27 1b 45 04 f7 70 1b 17 16 16
00000010: 85 16 ee b3 88 7d 08 64 2d 24 b8 1d 7e ac c9 72
00000020: 73 07 d3 d9 ef 5d 08 8b 47 97 5a 98 53 00 ec 13
00000030: cc 5a 46 7b 16 a2 14 6a f1 ea 17 71 9b 75 1d 46
00000040: 9d 6d 8c 3a a2 b2 75 c5 c9 4c 16 56 73 03 16 40
00000050: 42 fe a2 5a cc c7 ed 37 91 b1 eb e5 56 2a 01 bc
00000060: a2 83 ac 05 f1 a7 56 e5 f2 bb f4 18 7f 05 82 14
00000070: 70 de af 44 d4 cc a9 0a 95 6d c1 96 11 3d cf e1
00000080: aa 27 f1 87 60 d2 32 c1 1e 91 bf 60 00 5f d3 fb
00000090: a4 55 2e f0 0b 08 14 ed a3 63 54 4c b8 7b 5c 71
000000A0: 69 d1 3b 0c 6c 93 f3 99 2e fe 36 98 90 a1 05 ee
000000B0: 35 d2 da f8 81 59 f5 17 23 33 40 99 99 42 37 b0
000000C0: 0d 94 0a bd 00 cf 1c be 0e d0 13 93 e2 27 5a a5
000000D0: c5 e8 a0 25 5a 2d ad 6c b4 bc 64 37 05 ac cd 22
000000E0: 92 13 83 ab e8 87 93 29 82 dc 47 b4 1c 92 4d 36
000000F0: ef ba 10 3d 42 2d d6 2c d5 6b 95 99 2d 17 61 c4
00000100: c5 13 ed 55 a5 e5 b2 65 ac 25 24 21 c4 25 7f 6f
00000110: 68 fb ce 8f 17 60 e9 ac 9c 52 9f d5 d4 a7 14 35
00000120: 89 a4 1f de 21 a9 51 3c 1d 73 00 10 ba a6 7c 24
00000130: fb b9 20 21 5e df 63 8a c8 1f b1 55 05 5a 70 a8
00000140: b5 f4 23 9e 22 c0 2a 7c a5 11 01 c3 5e 3d 52 2a
00000150: b8 1d c5 19 b5 55 cc 8e f0 8d 6e 93 36 10 cd e3
00000160: c8 a5 a6 2e 90 53 fa 92 64 16 6c 4f da 9b e5 f8
00000170: 91 c5 ea b4 60 64 db ed d5 bc fc 3a 73 62 ce b2
00000180: ff 7a 15 95 0d 77 00 ee 5c a8 c5 89 2f 39 13 59
00000190: dd 52 ea 11 ae 28 82 36 be aa 29 68 4c f6 63 d5
000001A0: 93 a5 54 3d 8f 13 26 0a 87 34 b9 81 1c 2c cd d5
000001B0: 79 3a 65 6d 1c 6e 32 be b0 77 b7 b3 e4 ae b8 72
000001C0: f9 44 59 e9 14 46 67 56 93 ca 70 d1 ac 25 05 62
000001D0: f7 55 c2 9e 2e 11 a7 29 01 24 77 4a 6f 1c ba f6
000001E0: 4a 4f 83 75 29 1e c7 a9 68 29 02 d0
```

(51) Computes ICV using K3i as K_msg (fragment 2)

```
00000000: b4 68 c7 4d eb dd bd 92
```

(52) Composes IV (fragment 2)

```
00000000: 00 00 00 00 00 00 00 01
```

(53) Composes MGM nonce (fragment 3)

```
00000000: 00 00 00 02 b4 e1 3e 23
```

(54) Composes AAD (fragment 3)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 08 00 00 00 01 00 00 02 20 00 00 02 04
00000020: 00 03 00 04
```

(55) Composes plaintext (fragment 3)

```
00000000: 30 30 32 32 28 31 29 2e 63 72 6c 86 3f 68 74 74
00000010: 70 3a 2f 2f 74 65 73 74 67 6f 73 74 32 30 31 32
00000020: 2e 63 72 79 70 74 6f 70 72 6f 2e 72 75 2f 43 65
00000030: 72 74 45 6e 72 6f 6c 6c 2f 74 65 73 74 67 6f 73
00000040: 74 32 30 31 32 28 31 29 2e 63 72 6c 30 81 da 06
00000050: 08 2b 06 01 05 05 07 01 01 04 81 cd 30 81 ca 30
00000060: 44 06 08 2b 06 01 05 05 07 30 02 86 38 68 74 74
00000070: 70 3a 2f 2f 74 65 73 74 67 6f 73 74 32 30 31 32
00000080: 2e 63 72 79 70 74 6f 70 72 6f 2e 72 75 2f 43 65
00000090: 72 74 45 6e 72 6f 6c 6c 2f 72 6f 6f 74 32 30 31
000000A0: 38 2e 63 72 74 30 3f 06 08 2b 06 01 05 05 07 30
000000B0: 01 86 33 68 74 74 70 3a 2f 2f 74 65 73 74 67 6f
000000C0: 73 74 32 30 31 32 2e 63 72 79 70 74 6f 70 72 6f
000000D0: 2e 72 75 2f 6f 63 73 70 32 30 31 32 67 2f 6f 63
000000E0: 73 70 2e 73 72 66 30 41 06 08 2b 06 01 05 05 07
000000F0: 30 01 86 35 68 74 74 70 3a 2f 2f 74 65 73 74 67
00000100: 6f 73 74 32 30 31 32 2e 63 72 79 70 74 6f 70 72
00000110: 6f 2e 72 75 2f 6f 63 73 70 32 30 31 32 67 73 74
00000120: 2f 6f 63 73 70 2e 73 72 66 30 0a 06 08 2a 85 03
00000130: 07 01 01 03 02 03 41 00 21 ee 3b e1 fd 0f 36 90
00000140: 92 c4 a2 35 26 e8 dc 4e b8 ef 89 40 70 d2 91 39
00000150: bc 79 a6 e2 f7 c1 06 bd d5 d6 ff 72 a5 6c f2 c0
00000160: c3 75 e9 ca 67 81 c1 93 96 b4 bd 18 12 4c 37 f7
00000170: d9 73 d6 4c 8a a6 c4 0a 24 00 00 19 04 5e 9e 50
00000180: 5f 58 b0 a5 7a 33 45 83 49 66 0f 1c 3c 7a 67 71
00000190: 98 27 00 00 4e 09 00 00 00 30 44 31 20 30 1e 06
000001A0: 03 55 04 03 13 17 49 4b 45 20 49 6e 74 65 72 6f
000001B0: 70 20 54 65 73 74 20 53 65 72 76 65 72 31 13 30
000001C0: 11 06 03 55 04 0a 13 0a 45 4c 56 49 53 2d 50 4c
000001D0: 55 53 31 0b 30 09 06 03 55 04 06 13 02 52 55 29
000001E0: 00 00 95 0e 00 00 00 0c 30 0a 06 00
```

(56) Encrypts plaintext using K3i as K_msg, resulting in ciphertext (fragment 3)

```
00000000: e7 72 d9 51 90 b1 a2 bc 81 8d d6 56 bf 7a 81 e0
00000010: 1a a1 70 8b 35 a0 7e 5f e8 df 58 3d 75 5d d2 4c
00000020: 4c ce 17 77 3f 28 9c ca 7a a4 23 23 f0 c7 ff ff
00000030: 98 ee e3 1a 27 39 4d 90 1a b7 5b 44 11 16 11 3a
00000040: ea bf 83 66 da 92 2a 3a 3d bd b5 40 c8 bc f6 ed
00000050: cb 1d 5a 8e 30 f0 06 72 dc 6c da c1 45 7b e8 25
00000060: ca 93 2a b2 fe 4a db 00 90 e3 31 78 26 8d ae c8
00000070: 39 66 80 7d e5 01 5f 21 d6 c3 40 46 19 e4 43 9d
00000080: 23 c6 c1 18 06 49 bd f5 dc 8c 1b 19 b0 60 0c a3
00000090: ad f5 5c 57 e8 8e 37 e6 ea b6 79 11 b8 f1 16 ba
000000A0: a6 d9 09 1f 0d e0 3c 07 b8 ce 9d 11 a3 c6 f7 e4
000000B0: 62 e8 94 7b ad b9 8a 6b 9c f1 f8 43 cf 7e fc 5e
000000C0: 44 ab bf b1 88 f5 67 1e 84 5f 82 63 f3 13 89 55
000000D0: f5 ef 86 c3 db 48 37 f8 26 3c c4 6d a5 fc b5 69
000000E0: 56 0d 2d f3 c0 98 dd e7 53 da 0a 28 87 2f 38 ab
000000F0: a9 ec 60 a6 c4 54 c6 68 e7 6b e3 4b 54 bf b5 82
00000100: 44 c9 b9 45 bc 9e f5 58 d8 76 63 92 cd 52 ec 82
00000110: 80 d6 43 86 10 16 eb 7b 32 e4 ee ba ec 09 b6 4f
00000120: 35 1a bf da d7 de 40 fa b5 d2 40 f2 73 09 2d 52
00000130: 83 bd 56 a6 6b d3 9f 8a c2 c5 66 c6 6b 22 fb 6a
00000140: 00 b2 8a ac 9d 8b fc 8d 41 af 80 92 16 51 e2 cb
00000150: 89 62 9b 77 2b 1e 38 01 df fc 1f 81 2d 95 8b 9e
00000160: 1d 1e ad 9c c0 0d fc 77 6e 35 13 16 26 28 1a 29
00000170: 19 7f f8 08 5a 0f 09 4f 6f ba 7f 4c 5b cd 0c c2
00000180: 71 ab ea 82 a2 d2 d1 1b 17 fd dc c3 54 03 85 14
00000190: f4 90 47 2e 67 d7 93 c3 67 7e 8a f7 43 1a b3 41
000001A0: 32 f7 b0 58 38 6e 24 c8 96 d9 94 d3 54 89 2d 61
000001B0: 10 a9 9c 22 51 52 02 c9 b7 8d cc 5b 28 6d cb 55
000001C0: 5d 2f 97 8a 8f 3f 27 56 73 eb ec 5d e4 64 91 49
000001D0: 3b 88 f2 0a fc ed a5 67 a9 e3 71 ef 31 ce a0 33
000001E0: fc d8 ea 4d 1e 3f dc 89 c8 89 e2 c3
```

(57) Computes ICV using K3i as K_msg (fragment 3)

```
00000000: 54 4f 9b aa dd af bd ca
```

(58) Composes IV (fragment 3)

```
00000000: 00 00 00 00 00 00 00 02
```

(59) Composes MGM nonce (fragment 4)

```
00000000: 00 00 00 03 b4 e1 3e 23
```

(60) Composes AAD (fragment 4)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 08 00 00 00 01 00 00 01 7a 00 00 01 5e
00000020: 00 04 00 04
```

(61) Composes plaintext (fragment 4)

```
00000000: 08 2a 85 03 07 01 01 03 03 6a 3e 59 0d 72 1e 55
00000010: a3 c0 d1 2f 8a 9b 4e 44 10 58 59 bd 62 9e e7 12
00000020: 31 e5 7d 01 53 f3 84 40 dd ac 73 ed 09 3a 10 d9
00000030: 6e 7f eb 80 6c 11 9e 91 f3 7c 3c b0 55 f7 4b ec
00000040: 0e 78 36 10 95 02 09 86 b3 27 04 2a 83 3c 89 36
00000050: 1b 73 cf 7b c9 e0 df a2 07 12 1e 69 52 4d 89 1b
00000060: de 6e 48 d1 34 fa 21 78 22 88 2e 30 86 c0 80 0a
00000070: 2d 74 af 08 ff 35 75 a5 79 e3 85 40 22 6b a8 42
00000080: f6 72 24 bf 29 87 58 a8 20 29 00 00 08 00 00 40
00000090: 00 2f 00 00 0c 00 00 40 01 00 00 00 04 21 00 00
000000A0: 10 01 00 00 00 00 01 00 00 00 03 00 00 2c 00 00
000000B0: 38 00 00 00 34 01 03 04 05 6c 0c a5 70 03 00 00
000000C0: 08 01 00 00 20 03 00 00 08 01 00 00 21 03 00 00
000000D0: 08 01 00 00 22 03 00 00 08 01 00 00 23 00 00 00
000000E0: 08 05 00 00 00 2d 00 00 28 02 00 00 00 07 01 00
000000F0: 10 08 00 08 00 0a 6f 0a ab 0a 6f 0a ab 07 00 00
00000100: 10 00 00 ff ff 00 00 00 00 ff ff ff ff 29 00 00
00000110: 28 02 00 00 00 07 01 00 10 08 00 08 00 0a 00 00
00000120: 02 0a 00 00 02 07 00 00 10 00 00 ff ff 0a 00 00
00000130: 00 0a 00 00 ff 29 00 00 08 00 00 40 0a 00 00 00
00000140: 08 00 00 40 0b 00
```

(62) Encrypts plaintext using K3i as K_msg, resulting in ciphertext (fragment 4)

```
00000000: e0 8a 0b 04 ee f8 47 c2 52 96 71 9f 9d 39 0c 91
00000010: ea 6a 16 7c 80 31 a0 fd 76 cc c4 f1 8f 1a d3 be
00000020: fa 78 6b df c1 c6 73 83 be 36 69 c4 8a 87 ed 11
00000030: 90 31 a8 fd f9 0a 5c e4 d4 23 c9 e6 b3 96 ac b6
00000040: 8e bd fc 27 58 79 9f cc 8b ac 6b 59 e4 70 4b 05
00000050: 23 16 ed 49 25 f3 de 02 2e ce ae 86 e8 b4 ca b4
00000060: 96 ad 5b f6 2b c2 47 33 6f da f3 97 3c 13 ed 1f
00000070: 7a da 93 b5 69 6a b5 10 93 38 75 ea b7 34 a3 87
00000080: b6 83 c7 da 8a a1 d9 2a 0b 22 e2 ab 63 2b 57 2b
00000090: 88 e3 ea be 7b fc dc 26 ac b8 bb 15 96 f9 c2 f4
000000A0: 60 17 e4 09 18 ae 78 b8 73 02 6b 0e 20 cc b1 cd
000000B0: b4 4d 94 7f f3 16 28 9a d2 bd 26 77 4b a5 85 56
000000C0: b1 81 8b 9c c3 0a 7f 67 fe 6a 61 15 f1 45 66 f3
000000D0: 36 fc a5 bb 1f d7 6d e7 1d 9f 3f b5 cc 60 19 48
000000E0: 17 f7 08 28 1c 58 9f 2b 7a 0b b9 50 bd 02 ea b8
000000F0: 1e 03 1f 52 6a 7a fc e5 b4 6b 00 cf 0d 83 1f d2
00000100: 3f f2 ad 43 d4 86 6e c1 88 d2 87 d6 1f ac a3 30
00000110: 7b c1 5b 6a 3d 4c 20 72 5d 2c ca bf 87 a2 ce 1d
00000120: b3 fa c7 7c 22 cd 66 fc be 49 22 32 17 ee 6e 5e
00000130: 62 c1 ca 12 2b 5d 3d 7b ae b5 3e 53 c5 98 05 1f
00000140: 42 53 49 d1 2c c2
```

(63) Computes ICV using K3i as K_msg (fragment 4)

```
00000000: d2 25 f1 d0 38 65 b7 b6
```

(64) Composes IV (fragment 4)

```
00000000: 00 00 00 00 00 00 00 03
```

(65) Sends message fragment (1), peer receives message fragment (1)

```
       10.111.10.171:54295->10.111.15.45:4500 [548]

00000000: 00 00 00 00 92 80 e0 82 2e 75 87 78 db 57 8d 97
00000010: de 11 9d 1e 35 20 23 08 00 00 00 01 00 00 02 20
00000020: 23 00 02 04 00 01 00 04 00 00 00 00 00 00 00 00
00000030: 03 45 60 11 15 25 f5 45 bb 0e f4 25 26 e2 14 8c
00000040: a7 01 82 f6 9c 6e 42 f1 a3 9b 9e ac a6 dd 0d 9c
00000050: ff 79 15 ed b9 0c 81 a0 b4 29 61 fb 55 1b c1 73
00000060: 4d de 1f b2 5f 1f cb 84 5d 12 24 85 52 c4 f2 1d
00000070: 01 a7 92 ad 55 4d 90 d0 58 d2 1a 5e f6 dc 4e 73
00000080: d4 9b 08 66 d7 64 de 10 e6 75 69 20 e3 7b 6c f0
00000090: 4b 8b ff 60 39 f1 19 31 72 dd c1 09 33 5b 1d 56
000000A0: ee 0c 1c 42 d7 f3 04 d3 5b 9a 6e cf 7f b3 1f ac
000000B0: 34 a6 ee e0 ac 87 b8 88 99 75 a6 ae dc b5 30 38
000000C0: eb 3d 48 fd cc 69 64 f8 c6 61 ce e9 e1 24 ba aa
000000D0: 25 5e e6 ea 8b 0c ef 20 31 bf a9 ae 6d e2 82 d4
000000E0: ab 2c d7 af ca 62 fe bd 7c 8f a9 dc d3 63 05 d7
000000F0: ba 92 56 66 44 ad 5d 9d 1e 9a 27 2e 22 6e 5b 0c
00000100: af 84 6b c6 a7 cf ca 72 f8 8e d3 a1 bc d4 7c 5b
00000110: 7e 26 7f b3 05 d8 62 ef ad d6 07 70 d7 4b 33 e4
00000120: 26 84 e6 eb 5b 65 5c a7 71 29 45 15 d9 b0 83 6a
00000130: 52 5f a9 d8 dd f1 d8 62 c7 d7 3d e9 69 0e c5 b1
00000140: e1 de 20 6c 3d 5f f7 f7 9f f6 a5 7b 4d a5 4e e9
00000150: b4 c4 c2 7d cc 43 62 77 57 37 d3 40 48 b2 c0 5b
00000160: 48 ab d0 94 79 ef 3d 04 e3 d8 6d 42 56 ed cd 94
00000170: b4 23 2c fa f0 6b 39 ad 41 a3 b3 8f ec b8 6c ef
00000180: e1 98 3a b2 fb a8 fd 21 96 8a bf 3a 65 47 8a e9
00000190: 69 60 44 02 2c ec 7a 86 74 fe 1d 9b 08 5e b8 5e
000001A0: f8 ca 37 20 5f a7 74 8c 12 88 f2 d8 9e d4 94 29
000001B0: c2 db f9 fb 35 a0 cf 21 2b da 8b 9e cc 52 84 eb
000001C0: c4 12 39 3e e6 18 fb f7 57 6c b5 1e 10 3d 11 9c
000001D0: 29 9c 41 73 69 d8 d0 9d 71 2b 77 66 87 65 51 19
000001E0: db 27 a0 dd aa 64 ba fd c0 5f e1 4e da 7c 20 fc
000001F0: 8c 13 ab 2d c2 9c 37 9d 7e 51 cb 29 03 10 52 dc
00000200: f8 09 61 cc 12 9a a0 8e 1b e4 52 f8 72 bd 7a 86
00000210: db 93 7c 55 b8 1e 7f 21 d4 e6 02 f2 b1 51 cd e6
00000220: dc 64 12 1c
```

(66) Sends message fragment (2), peer receives message fragment (2)

```
       10.111.10.171:54295->10.111.15.45:4500 [548]

00000000: 00 00 00 00 92 80 e0 82 2e 75 87 78 db 57 8d 97
00000010: de 11 9d 1e 35 20 23 08 00 00 00 01 00 00 02 20
00000020: 00 00 02 04 00 02 00 04 00 00 00 00 00 00 00 01
00000030: 3c b1 b4 aa 04 56 27 1b 45 04 f7 70 1b 17 16 16
00000040: 85 16 ee b3 88 7d 08 64 2d 24 b8 1d 7e ac c9 72
00000050: 73 07 d3 d9 ef 5d 08 8b 47 97 5a 98 53 00 ec 13
00000060: cc 5a 46 7b 16 a2 14 6a f1 ea 17 71 9b 75 1d 46
00000070: 9d 6d 8c 3a a2 b2 75 c5 c9 4c 16 56 73 03 16 40
00000080: 42 fe a2 5a cc c7 ed 37 91 b1 eb e5 56 2a 01 bc
00000090: a2 83 ac 05 f1 a7 56 e5 f2 bb f4 18 7f 05 82 14
000000A0: 70 de af 44 d4 cc a9 0a 95 6d c1 96 11 3d cf e1
000000B0: aa 27 f1 87 60 d2 32 c1 1e 91 bf 60 00 5f d3 fb
000000C0: a4 55 2e f0 0b 08 14 ed a3 63 54 4c b8 7b 5c 71
000000D0: 69 d1 3b 0c 6c 93 f3 99 2e fe 36 98 90 a1 05 ee
000000E0: 35 d2 da f8 81 59 f5 17 23 33 40 99 99 42 37 b0
000000F0: 0d 94 0a bd 00 cf 1c be 0e d0 13 93 e2 27 5a a5
00000100: c5 e8 a0 25 5a 2d ad 6c b4 bc 64 37 05 ac cd 22
00000110: 92 13 83 ab e8 87 93 29 82 dc 47 b4 1c 92 4d 36
00000120: ef ba 10 3d 42 2d d6 2c d5 6b 95 99 2d 17 61 c4
00000130: c5 13 ed 55 a5 e5 b2 65 ac 25 24 21 c4 25 7f 6f
00000140: 68 fb ce 8f 17 60 e9 ac 9c 52 9f d5 d4 a7 14 35
00000150: 89 a4 1f de 21 a9 51 3c 1d 73 00 10 ba a6 7c 24
00000160: fb b9 20 21 5e df 63 8a c8 1f b1 55 05 5a 70 a8
00000170: b5 f4 23 9e 22 c0 2a 7c a5 11 01 c3 5e 3d 52 2a
00000180: b8 1d c5 19 b5 55 cc 8e f0 8d 6e 93 36 10 cd e3
00000190: c8 a5 a6 2e 90 53 fa 92 64 16 6c 4f da 9b e5 f8
000001A0: 91 c5 ea b4 60 64 db ed d5 bc fc 3a 73 62 ce b2
000001B0: ff 7a 15 95 0d 77 00 ee 5c a8 c5 89 2f 39 13 59
000001C0: dd 52 ea 11 ae 28 82 36 be aa 29 68 4c f6 63 d5
000001D0: 93 a5 54 3d 8f 13 26 0a 87 34 b9 81 1c 2c cd d5
000001E0: 79 3a 65 6d 1c 6e 32 be b0 77 b7 b3 e4 ae b8 72
000001F0: f9 44 59 e9 14 46 67 56 93 ca 70 d1 ac 25 05 62
00000200: f7 55 c2 9e 2e 11 a7 29 01 24 77 4a 6f 1c ba f6
00000210: 4a 4f 83 75 29 1e c7 a9 68 29 02 d0 b4 68 c7 4d
00000220: eb dd bd 92
```

(67) Sends message fragment (3), peer receives message fragment (3)

```
       10.111.10.171:54295->10.111.15.45:4500 [548]

00000000: 00 00 00 00 92 80 e0 82 2e 75 87 78 db 57 8d 97
00000010: de 11 9d 1e 35 20 23 08 00 00 00 01 00 00 02 20
00000020: 00 00 02 04 00 03 00 04 00 00 00 00 00 00 00 02
00000030: e7 72 d9 51 90 b1 a2 bc 81 8d d6 56 bf 7a 81 e0
00000040: 1a a1 70 8b 35 a0 7e 5f e8 df 58 3d 75 5d d2 4c
00000050: 4c ce 17 77 3f 28 9c ca 7a a4 23 23 f0 c7 ff ff
00000060: 98 ee e3 1a 27 39 4d 90 1a b7 5b 44 11 16 11 3a
00000070: ea bf 83 66 da 92 2a 3a 3d bd b5 40 c8 bc f6 ed
00000080: cb 1d 5a 8e 30 f0 06 72 dc 6c da c1 45 7b e8 25
00000090: ca 93 2a b2 fe 4a db 00 90 e3 31 78 26 8d ae c8
000000A0: 39 66 80 7d e5 01 5f 21 d6 c3 40 46 19 e4 43 9d
000000B0: 23 c6 c1 18 06 49 bd f5 dc 8c 1b 19 b0 60 0c a3
000000C0: ad f5 5c 57 e8 8e 37 e6 ea b6 79 11 b8 f1 16 ba
000000D0: a6 d9 09 1f 0d e0 3c 07 b8 ce 9d 11 a3 c6 f7 e4
000000E0: 62 e8 94 7b ad b9 8a 6b 9c f1 f8 43 cf 7e fc 5e
000000F0: 44 ab bf b1 88 f5 67 1e 84 5f 82 63 f3 13 89 55
00000100: f5 ef 86 c3 db 48 37 f8 26 3c c4 6d a5 fc b5 69
00000110: 56 0d 2d f3 c0 98 dd e7 53 da 0a 28 87 2f 38 ab
00000120: a9 ec 60 a6 c4 54 c6 68 e7 6b e3 4b 54 bf b5 82
00000130: 44 c9 b9 45 bc 9e f5 58 d8 76 63 92 cd 52 ec 82
00000140: 80 d6 43 86 10 16 eb 7b 32 e4 ee ba ec 09 b6 4f
00000150: 35 1a bf da d7 de 40 fa b5 d2 40 f2 73 09 2d 52
00000160: 83 bd 56 a6 6b d3 9f 8a c2 c5 66 c6 6b 22 fb 6a
00000170: 00 b2 8a ac 9d 8b fc 8d 41 af 80 92 16 51 e2 cb
00000180: 89 62 9b 77 2b 1e 38 01 df fc 1f 81 2d 95 8b 9e
00000190: 1d 1e ad 9c c0 0d fc 77 6e 35 13 16 26 28 1a 29
000001A0: 19 7f f8 08 5a 0f 09 4f 6f ba 7f 4c 5b cd 0c c2
000001B0: 71 ab ea 82 a2 d2 d1 1b 17 fd dc c3 54 03 85 14
000001C0: f4 90 47 2e 67 d7 93 c3 67 7e 8a f7 43 1a b3 41
000001D0: 32 f7 b0 58 38 6e 24 c8 96 d9 94 d3 54 89 2d 61
000001E0: 10 a9 9c 22 51 52 02 c9 b7 8d cc 5b 28 6d cb 55
000001F0: 5d 2f 97 8a 8f 3f 27 56 73 eb ec 5d e4 64 91 49
00000200: 3b 88 f2 0a fc ed a5 67 a9 e3 71 ef 31 ce a0 33
00000210: fc d8 ea 4d 1e 3f dc 89 c8 89 e2 c3 54 4f 9b aa
00000220: dd af bd ca
```

(68) Sends message fragment (4), peer receives message fragment (4)

```
       10.111.10.171:54295->10.111.15.45:4500 [382]

00000000: 00 00 00 00 92 80 e0 82 2e 75 87 78 db 57 8d 97
00000010: de 11 9d 1e 35 20 23 08 00 00 00 01 00 00 01 7a
00000020: 00 00 01 5e 00 04 00 04 00 00 00 00 00 00 00 03
00000030: e0 8a 0b 04 ee f8 47 c2 52 96 71 9f 9d 39 0c 91
00000040: ea 6a 16 7c 80 31 a0 fd 76 cc c4 f1 8f 1a d3 be
00000050: fa 78 6b df c1 c6 73 83 be 36 69 c4 8a 87 ed 11
00000060: 90 31 a8 fd f9 0a 5c e4 d4 23 c9 e6 b3 96 ac b6
00000070: 8e bd fc 27 58 79 9f cc 8b ac 6b 59 e4 70 4b 05
00000080: 23 16 ed 49 25 f3 de 02 2e ce ae 86 e8 b4 ca b4
00000090: 96 ad 5b f6 2b c2 47 33 6f da f3 97 3c 13 ed 1f
000000A0: 7a da 93 b5 69 6a b5 10 93 38 75 ea b7 34 a3 87
000000B0: b6 83 c7 da 8a a1 d9 2a 0b 22 e2 ab 63 2b 57 2b
000000C0: 88 e3 ea be 7b fc dc 26 ac b8 bb 15 96 f9 c2 f4
000000D0: 60 17 e4 09 18 ae 78 b8 73 02 6b 0e 20 cc b1 cd
000000E0: b4 4d 94 7f f3 16 28 9a d2 bd 26 77 4b a5 85 56
000000F0: b1 81 8b 9c c3 0a 7f 67 fe 6a 61 15 f1 45 66 f3
00000100: 36 fc a5 bb 1f d7 6d e7 1d 9f 3f b5 cc 60 19 48
00000110: 17 f7 08 28 1c 58 9f 2b 7a 0b b9 50 bd 02 ea b8
00000120: 1e 03 1f 52 6a 7a fc e5 b4 6b 00 cf 0d 83 1f d2
00000130: 3f f2 ad 43 d4 86 6e c1 88 d2 87 d6 1f ac a3 30
00000140: 7b c1 5b 6a 3d 4c 20 72 5d 2c ca bf 87 a2 ce 1d
00000150: b3 fa c7 7c 22 cd 66 fc be 49 22 32 17 ee 6e 5e
00000160: 62 c1 ca 12 2b 5d 3d 7b ae b5 3e 53 c5 98 05 1f
00000170: 42 53 49 d1 2c c2 d2 25 f1 d0 38 65 b7 b6
```

Responder's actions:

(69) Computes shared key

```
00000000: bd 04 9d 0f 9c 5f 58 af c7 e4 01 bc 18 59 01 7c
00000010: 88 28 f9 f2 9f 33 01 5d 49 9a 7d 14 74 d4 31 ac
```

(70) Computes SKEYSEED

```
00000000: 9b ed 6c 79 64 b3 de 3a e4 9e dd 62 04 5a f0 8b
00000010: 43 88 33 d4 e6 9e 73 16 a1 1a 9e b2 b4 19 13 c5
00000020: d0 6d fb 86 40 11 c3 02 bb e5 a3 b5 e4 4a c4 c0
00000030: 9d 18 c6 94 de c3 c5 14 82 e7 a2 51 fe c4 98 ca
```

(71) Computes SK_d

```
00000000: c2 21 15 fd d3 99 3b 2a 43 60 c4 59 34 b0 be 3f
00000010: 53 ef 6e b1 dd 88 ad 72 55 dd 83 22 5c 6f e1 d6
00000020: 1f 1e ab 06 f9 41 cb c8 ea f9 dc fc 19 a0 2d bf
00000030: 9a 0a 3f 3a 9a 45 1f 08 b6 a9 2c 62 52 b7 26 34
```

(72) Computes SK_ei

```
00000000: 18 4e 4e 0f 36 28 bf 3c 9c 04 8e 93 bf a0 77 53
00000010: 91 34 12 81 42 e6 4e 62 7f db a5 ed 98 60 50 ff
00000020: b4 e1 3e 23
```

(73) Computes SK_er

```
00000000: e9 27 59 2f 09 49 68 1e 0e 62 db c6 19 06 73 13
00000010: cf da 5c 02 27 3e 4a b4 78 98 b4 86 d0 e9 34 f4
00000020: a5 bb 18 2f
```

(74) Computes SK_pi

```
00000000: 30 2c 10 8d 0f 61 47 00 f1 40 4f a9 4f af b5 30
00000010: 11 ba 5f 24 39 32 85 12 4e 7e 71 75 50 15 a6 93
00000020: c3 d0 5e 40 2e 21 8e b1 59 09 cd a4 eb b4 91 68
00000030: 29 42 fe e2 d8 76 8f a6 96 55 1f ab 6c 9b 00 f8
```

(75) Computes SK_pr

```
00000000: 6f 81 72 cb 96 58 fb 0e 17 70 b6 b9 1f a9 69 a9
00000010: fc c7 27 4f b4 e1 85 90 a0 c7 9f f9 72 11 61 2a
00000020: 35 b7 b7 96 d3 6a bb a5 aa b1 b8 34 8d 99 c6 f3
00000030: 2b fc 32 56 c1 94 71 04 55 bd 89 6a bf c3 8b fe
```

(76) Extracts IV from message (fragment 1)

```
00000000: 00 00 00 00 00 00 00 00
```

(77) Computes K1i (i1 = 0)

```
00000000: 3c 57 d7 c8 9f 50 98 fc 86 81 d6 8a 4e 5d 83 c6
00000010: 1e 42 e6 e7 60 67 05 8d f5 2e 10 13 12 15 32 58
```

(78) Computes K2i (i2 = 0)

```
00000000: 0b 88 0a 1b c8 3e 61 79 82 08 db 13 31 08 63 3c
00000010: 17 62 17 cb 7d 18 ce 70 37 84 85 f4 89 49 d0 06
```

(79) Computes K3i (i3 = 0)

```
00000000: 18 63 41 67 49 6e cf 48 56 71 4d aa 42 63 5c 11
00000010: 2e 26 5b e2 7b c7 53 a4 09 82 e5 5a 7e f4 65 4d
```

(80) Composes MGM nonce (fragment 1)

```
00000000: 00 00 00 00 b4 e1 3e 23
```

(81) Extracts ICV from message (fragment 1)

```
00000000: b1 51 cd e6 dc 64 12 1c
```

(82) Extracts AAD from message (fragment 1)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 08 00 00 00 01 00 00 02 20 23 00 02 04
00000020: 00 01 00 04
```

(83) Extracts ciphertext from message (fragment 1)

```
00000000: 03 45 60 11 15 25 f5 45 bb 0e f4 25 26 e2 14 8c
00000010: a7 01 82 f6 9c 6e 42 f1 a3 9b 9e ac a6 dd 0d 9c
00000020: ff 79 15 ed b9 0c 81 a0 b4 29 61 fb 55 1b c1 73
00000030: 4d de 1f b2 5f 1f cb 84 5d 12 24 85 52 c4 f2 1d
00000040: 01 a7 92 ad 55 4d 90 d0 58 d2 1a 5e f6 dc 4e 73
00000050: d4 9b 08 66 d7 64 de 10 e6 75 69 20 e3 7b 6c f0
00000060: 4b 8b ff 60 39 f1 19 31 72 dd c1 09 33 5b 1d 56
00000070: ee 0c 1c 42 d7 f3 04 d3 5b 9a 6e cf 7f b3 1f ac
00000080: 34 a6 ee e0 ac 87 b8 88 99 75 a6 ae dc b5 30 38
00000090: eb 3d 48 fd cc 69 64 f8 c6 61 ce e9 e1 24 ba aa
000000A0: 25 5e e6 ea 8b 0c ef 20 31 bf a9 ae 6d e2 82 d4
000000B0: ab 2c d7 af ca 62 fe bd 7c 8f a9 dc d3 63 05 d7
000000C0: ba 92 56 66 44 ad 5d 9d 1e 9a 27 2e 22 6e 5b 0c
000000D0: af 84 6b c6 a7 cf ca 72 f8 8e d3 a1 bc d4 7c 5b
000000E0: 7e 26 7f b3 05 d8 62 ef ad d6 07 70 d7 4b 33 e4
000000F0: 26 84 e6 eb 5b 65 5c a7 71 29 45 15 d9 b0 83 6a
00000100: 52 5f a9 d8 dd f1 d8 62 c7 d7 3d e9 69 0e c5 b1
00000110: e1 de 20 6c 3d 5f f7 f7 9f f6 a5 7b 4d a5 4e e9
00000120: b4 c4 c2 7d cc 43 62 77 57 37 d3 40 48 b2 c0 5b
00000130: 48 ab d0 94 79 ef 3d 04 e3 d8 6d 42 56 ed cd 94
00000140: b4 23 2c fa f0 6b 39 ad 41 a3 b3 8f ec b8 6c ef
00000150: e1 98 3a b2 fb a8 fd 21 96 8a bf 3a 65 47 8a e9
00000160: 69 60 44 02 2c ec 7a 86 74 fe 1d 9b 08 5e b8 5e
00000170: f8 ca 37 20 5f a7 74 8c 12 88 f2 d8 9e d4 94 29
00000180: c2 db f9 fb 35 a0 cf 21 2b da 8b 9e cc 52 84 eb
00000190: c4 12 39 3e e6 18 fb f7 57 6c b5 1e 10 3d 11 9c
000001A0: 29 9c 41 73 69 d8 d0 9d 71 2b 77 66 87 65 51 19
000001B0: db 27 a0 dd aa 64 ba fd c0 5f e1 4e da 7c 20 fc
000001C0: 8c 13 ab 2d c2 9c 37 9d 7e 51 cb 29 03 10 52 dc
000001D0: f8 09 61 cc 12 9a a0 8e 1b e4 52 f8 72 bd 7a 86
000001E0: db 93 7c 55 b8 1e 7f 21 d4 e6 02 f2
```

(84) Decrypts ciphertext and verifies ICV using K3i as K_msg, resulting in plaintext (fragment 1)

```
00000000: 25 00 00 4e 09 00 00 00 30 44 31 20 30 1e 06 03
00000010: 55 04 03 13 17 49 4b 45 20 49 6e 74 65 72 6f 70
00000020: 20 54 65 73 74 20 43 6c 69 65 6e 74 31 13 30 11
00000030: 06 03 55 04 0a 13 0a 45 4c 56 49 53 2d 50 4c 55
00000040: 53 31 0b 30 09 06 03 55 04 06 13 02 52 55 26 00
00000050: 05 00 04 30 82 04 f7 30 82 04 a4 a0 03 02 01 02
00000060: 02 13 7c 00 03 da a8 9e 1e ff 9e 79 05 fb bb 00
00000070: 01 00 03 da a8 30 0a 06 08 2a 85 03 07 01 01 03
00000080: 02 30 82 01 0a 31 18 30 16 06 05 2a 85 03 64 01
00000090: 12 0d 31 32 33 34 35 36 37 38 39 30 31 32 33 31
000000A0: 1a 30 18 06 08 2a 85 03 03 81 03 01 01 12 0c 30
000000B0: 30 31 32 33 34 35 36 37 38 39 30 31 2f 30 2d 06
000000C0: 03 55 04 09 0c 26 d1 83 d0 bb 2e 20 d0 a1 d1 83
000000D0: d1 89 d1 91 d0 b2 d1 81 d0 ba d0 b8 d0 b9 20 d0
000000E0: b2 d0 b0 d0 bb 20 d0 b4 2e 20 31 38 31 0b 30 09
000000F0: 06 03 55 04 06 13 02 52 55 31 19 30 17 06 03 55
00000100: 04 08 0c 10 d0 b3 2e 20 d0 9c d0 be d1 81 d0 ba
00000110: d0 b2 d0 b0 31 15 30 13 06 03 55 04 07 0c 0c d0
00000120: 9c d0 be d1 81 d0 ba d0 b2 d0 b0 31 25 30 23 06
00000130: 03 55 04 0a 0c 1c d0 9e d0 9e d0 9e 20 22 d0 9a
00000140: d0 a0 d0 98 d0 9f d0 a2 d0 9e 2d d0 9f d0 a0 d0
00000150: 9e 22 31 3b 30 39 06 03 55 04 03 0c 32 d0 a2 d0
00000160: b5 d1 81 d1 82 d0 be d0 b2 d1 8b d0 b9 20 d0 a3
00000170: d0 a6 20 d0 9e d0 9e d0 9e 20 22 d0 9a d0 a0 d0
00000180: 98 d0 9f d0 a2 d0 9e 2d d0 9f d0 a0 d0 9e 22 30
00000190: 1e 17 0d 32 31 31 30 30 31 30 36 31 30 31 30 5a
000001A0: 17 0d 32 32 30 31 30 31 30 36 32 30 31 30 5a 30
000001B0: 44 31 20 30 1e 06 03 55 04 03 13 17 49 4b 45 20
000001C0: 49 6e 74 65 72 6f 70 20 54 65 73 74 20 43 6c 69
000001D0: 65 6e 74 31 13 30 11 06 03 55 04 0a 13 0a 45 4c
000001E0: 56 49 53 2d 50 4c 55 53 31 0b 30 00
```

(85) Extracts IV from message (fragment 2)

```
00000000: 00 00 00 00 00 00 00 01
```

(86) Uses previously computed key K3i

```
00000000: 18 63 41 67 49 6e cf 48 56 71 4d aa 42 63 5c 11
00000010: 2e 26 5b e2 7b c7 53 a4 09 82 e5 5a 7e f4 65 4d
```

(87) Composes MGM nonce (fragment 2)

```
00000000: 00 00 00 01 b4 e1 3e 23
```

(88) Extracts ICV from message (fragment 2)

```
00000000: b4 68 c7 4d eb dd bd 92
```

(89) Extracts AAD from message (fragment 2)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 08 00 00 00 01 00 00 02 20 00 00 02 04
00000020: 00 02 00 04
```

(90) Extracts ciphertext from message (fragment 2)

```
00000000: 3c b1 b4 aa 04 56 27 1b 45 04 f7 70 1b 17 16 16
00000010: 85 16 ee b3 88 7d 08 64 2d 24 b8 1d 7e ac c9 72
00000020: 73 07 d3 d9 ef 5d 08 8b 47 97 5a 98 53 00 ec 13
00000030: cc 5a 46 7b 16 a2 14 6a f1 ea 17 71 9b 75 1d 46
00000040: 9d 6d 8c 3a a2 b2 75 c5 c9 4c 16 56 73 03 16 40
00000050: 42 fe a2 5a cc c7 ed 37 91 b1 eb e5 56 2a 01 bc
00000060: a2 83 ac 05 f1 a7 56 e5 f2 bb f4 18 7f 05 82 14
00000070: 70 de af 44 d4 cc a9 0a 95 6d c1 96 11 3d cf e1
00000080: aa 27 f1 87 60 d2 32 c1 1e 91 bf 60 00 5f d3 fb
00000090: a4 55 2e f0 0b 08 14 ed a3 63 54 4c b8 7b 5c 71
000000A0: 69 d1 3b 0c 6c 93 f3 99 2e fe 36 98 90 a1 05 ee
000000B0: 35 d2 da f8 81 59 f5 17 23 33 40 99 99 42 37 b0
000000C0: 0d 94 0a bd 00 cf 1c be 0e d0 13 93 e2 27 5a a5
000000D0: c5 e8 a0 25 5a 2d ad 6c b4 bc 64 37 05 ac cd 22
000000E0: 92 13 83 ab e8 87 93 29 82 dc 47 b4 1c 92 4d 36
000000F0: ef ba 10 3d 42 2d d6 2c d5 6b 95 99 2d 17 61 c4
00000100: c5 13 ed 55 a5 e5 b2 65 ac 25 24 21 c4 25 7f 6f
00000110: 68 fb ce 8f 17 60 e9 ac 9c 52 9f d5 d4 a7 14 35
00000120: 89 a4 1f de 21 a9 51 3c 1d 73 00 10 ba a6 7c 24
00000130: fb b9 20 21 5e df 63 8a c8 1f b1 55 05 5a 70 a8
00000140: b5 f4 23 9e 22 c0 2a 7c a5 11 01 c3 5e 3d 52 2a
00000150: b8 1d c5 19 b5 55 cc 8e f0 8d 6e 93 36 10 cd e3
00000160: c8 a5 a6 2e 90 53 fa 92 64 16 6c 4f da 9b e5 f8
00000170: 91 c5 ea b4 60 64 db ed d5 bc fc 3a 73 62 ce b2
00000180: ff 7a 15 95 0d 77 00 ee 5c a8 c5 89 2f 39 13 59
00000190: dd 52 ea 11 ae 28 82 36 be aa 29 68 4c f6 63 d5
000001A0: 93 a5 54 3d 8f 13 26 0a 87 34 b9 81 1c 2c cd d5
000001B0: 79 3a 65 6d 1c 6e 32 be b0 77 b7 b3 e4 ae b8 72
000001C0: f9 44 59 e9 14 46 67 56 93 ca 70 d1 ac 25 05 62
000001D0: f7 55 c2 9e 2e 11 a7 29 01 24 77 4a 6f 1c ba f6
000001E0: 4a 4f 83 75 29 1e c7 a9 68 29 02 d0
```

(91) Decrypts ciphertext and verifies ICV using K3i as K_msg, resulting in plaintext (fragment 2)

```
00000000: 09 06 03 55 04 06 13 02 52 55 30 81 aa 30 21 06
00000010: 08 2a 85 03 07 01 01 01 02 30 15 06 09 2a 85 03
00000020: 07 01 02 01 02 01 06 08 2a 85 03 07 01 01 02 03
00000030: 03 81 84 00 04 81 80 ee 2f 0a 0e 09 1e 7e 04 ef
00000040: ba 5b 62 a2 52 86 e1 9c 24 50 30 50 b0 b4 8a 37
00000050: 35 b5 fc af 28 94 ec b5 9b 92 41 5b 69 e2 c9 ba
00000060: 24 de 6a 72 c4 ef 44 bb 89 a1 05 14 1b 87 3d 6a
00000070: a3 72 3e 17 ca 7f 39 28 ce 16 8b dd 07 52 87 6a
00000080: 0d 77 42 6d 99 2b 46 2c fd 4b b2 7c d7 c7 17 08
00000090: 12 54 63 47 9d 14 3d 61 ed f2 95 ab 11 80 69 02
000000A0: a7 66 60 50 7e a4 53 6d ad 01 49 b2 16 8a 95 1d
000000B0: cf 1a 57 93 56 14 5e a3 82 02 59 30 82 02 55 30
000000C0: 0e 06 03 55 1d 0f 01 01 ff 04 04 03 02 05 a0 30
000000D0: 13 06 03 55 1d 25 04 0c 30 0a 06 08 2b 06 01 05
000000E0: 05 07 03 11 30 1d 06 03 55 1d 0e 04 16 04 14 40
000000F0: 81 b1 d1 18 75 f0 da 6b 3c 50 5f cd 73 1d d9 77
00000100: f2 d7 c1 30 1f 06 03 55 1d 23 04 18 30 16 80 14
00000110: 9b 85 5e fb 81 dc 4d 59 07 51 63 cf be df da 2c
00000120: 7f c9 44 3c 30 82 01 0f 06 03 55 1d 1f 04 82 01
00000130: 06 30 82 01 02 30 81 ff a0 81 fc a0 81 f9 86 81
00000140: b5 68 74 74 70 3a 2f 2f 74 65 73 74 67 6f 73 74
00000150: 32 30 31 32 2e 63 72 79 70 74 6f 70 72 6f 2e 72
00000160: 75 2f 43 65 72 74 45 6e 72 6f 6c 6c 2f 21 30 34
00000170: 32 32 21 30 34 33 35 21 30 34 34 31 21 30 34 34
00000180: 32 21 30 34 33 65 21 30 34 33 32 21 30 34 34 62
00000190: 21 30 34 33 39 25 32 30 21 30 34 32 33 21 30 34
000001A0: 32 36 25 32 30 21 30 34 31 65 21 30 34 31 65 21
000001B0: 30 34 31 65 25 32 30 21 30 30 32 32 21 30 34 31
000001C0: 61 21 30 34 32 30 21 30 34 31 38 21 30 34 31 66
000001D0: 21 30 34 32 32 21 30 34 31 65 2d 21 30 34 31 66
000001E0: 21 30 34 32 30 21 30 34 31 65 21 00
```

(92) Extracts IV from message (fragment 3)

```
00000000: 00 00 00 00 00 00 00 02
```

(93) Uses previously computed key K3i

```
00000000: 18 63 41 67 49 6e cf 48 56 71 4d aa 42 63 5c 11
00000010: 2e 26 5b e2 7b c7 53 a4 09 82 e5 5a 7e f4 65 4d
```

(94) Composes MGM nonce (fragment 3)

```
00000000: 00 00 00 02 b4 e1 3e 23
```

(95) Extracts ICV from message (fragment 3)

```
00000000: 54 4f 9b aa dd af bd ca
```

(96) Extracts AAD from message (fragment 3)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 08 00 00 00 01 00 00 02 20 00 00 02 04
00000020: 00 03 00 04
```

(97) Extracts ciphertext from message (fragment 3)

```
00000000: e7 72 d9 51 90 b1 a2 bc 81 8d d6 56 bf 7a 81 e0
00000010: 1a a1 70 8b 35 a0 7e 5f e8 df 58 3d 75 5d d2 4c
00000020: 4c ce 17 77 3f 28 9c ca 7a a4 23 23 f0 c7 ff ff
00000030: 98 ee e3 1a 27 39 4d 90 1a b7 5b 44 11 16 11 3a
00000040: ea bf 83 66 da 92 2a 3a 3d bd b5 40 c8 bc f6 ed
00000050: cb 1d 5a 8e 30 f0 06 72 dc 6c da c1 45 7b e8 25
00000060: ca 93 2a b2 fe 4a db 00 90 e3 31 78 26 8d ae c8
00000070: 39 66 80 7d e5 01 5f 21 d6 c3 40 46 19 e4 43 9d
00000080: 23 c6 c1 18 06 49 bd f5 dc 8c 1b 19 b0 60 0c a3
00000090: ad f5 5c 57 e8 8e 37 e6 ea b6 79 11 b8 f1 16 ba
000000A0: a6 d9 09 1f 0d e0 3c 07 b8 ce 9d 11 a3 c6 f7 e4
000000B0: 62 e8 94 7b ad b9 8a 6b 9c f1 f8 43 cf 7e fc 5e
000000C0: 44 ab bf b1 88 f5 67 1e 84 5f 82 63 f3 13 89 55
000000D0: f5 ef 86 c3 db 48 37 f8 26 3c c4 6d a5 fc b5 69
000000E0: 56 0d 2d f3 c0 98 dd e7 53 da 0a 28 87 2f 38 ab
000000F0: a9 ec 60 a6 c4 54 c6 68 e7 6b e3 4b 54 bf b5 82
00000100: 44 c9 b9 45 bc 9e f5 58 d8 76 63 92 cd 52 ec 82
00000110: 80 d6 43 86 10 16 eb 7b 32 e4 ee ba ec 09 b6 4f
00000120: 35 1a bf da d7 de 40 fa b5 d2 40 f2 73 09 2d 52
00000130: 83 bd 56 a6 6b d3 9f 8a c2 c5 66 c6 6b 22 fb 6a
00000140: 00 b2 8a ac 9d 8b fc 8d 41 af 80 92 16 51 e2 cb
00000150: 89 62 9b 77 2b 1e 38 01 df fc 1f 81 2d 95 8b 9e
00000160: 1d 1e ad 9c c0 0d fc 77 6e 35 13 16 26 28 1a 29
00000170: 19 7f f8 08 5a 0f 09 4f 6f ba 7f 4c 5b cd 0c c2
00000180: 71 ab ea 82 a2 d2 d1 1b 17 fd dc c3 54 03 85 14
00000190: f4 90 47 2e 67 d7 93 c3 67 7e 8a f7 43 1a b3 41
000001A0: 32 f7 b0 58 38 6e 24 c8 96 d9 94 d3 54 89 2d 61
000001B0: 10 a9 9c 22 51 52 02 c9 b7 8d cc 5b 28 6d cb 55
000001C0: 5d 2f 97 8a 8f 3f 27 56 73 eb ec 5d e4 64 91 49
000001D0: 3b 88 f2 0a fc ed a5 67 a9 e3 71 ef 31 ce a0 33
000001E0: fc d8 ea 4d 1e 3f dc 89 c8 89 e2 c3
```

(98) Decrypts ciphertext and verifies ICV using K3i as K_msg, resulting in plaintext (fragment 3)

```
00000000: 30 30 32 32 28 31 29 2e 63 72 6c 86 3f 68 74 74
00000010: 70 3a 2f 2f 74 65 73 74 67 6f 73 74 32 30 31 32
00000020: 2e 63 72 79 70 74 6f 70 72 6f 2e 72 75 2f 43 65
00000030: 72 74 45 6e 72 6f 6c 6c 2f 74 65 73 74 67 6f 73
00000040: 74 32 30 31 32 28 31 29 2e 63 72 6c 30 81 da 06
00000050: 08 2b 06 01 05 05 07 01 01 04 81 cd 30 81 ca 30
00000060: 44 06 08 2b 06 01 05 05 07 30 02 86 38 68 74 74
00000070: 70 3a 2f 2f 74 65 73 74 67 6f 73 74 32 30 31 32
00000080: 2e 63 72 79 70 74 6f 70 72 6f 2e 72 75 2f 43 65
00000090: 72 74 45 6e 72 6f 6c 6c 2f 72 6f 6f 74 32 30 31
000000A0: 38 2e 63 72 74 30 3f 06 08 2b 06 01 05 05 07 30
000000B0: 01 86 33 68 74 74 70 3a 2f 2f 74 65 73 74 67 6f
000000C0: 73 74 32 30 31 32 2e 63 72 79 70 74 6f 70 72 6f
000000D0: 2e 72 75 2f 6f 63 73 70 32 30 31 32 67 2f 6f 63
000000E0: 73 70 2e 73 72 66 30 41 06 08 2b 06 01 05 05 07
000000F0: 30 01 86 35 68 74 74 70 3a 2f 2f 74 65 73 74 67
00000100: 6f 73 74 32 30 31 32 2e 63 72 79 70 74 6f 70 72
00000110: 6f 2e 72 75 2f 6f 63 73 70 32 30 31 32 67 73 74
00000120: 2f 6f 63 73 70 2e 73 72 66 30 0a 06 08 2a 85 03
00000130: 07 01 01 03 02 03 41 00 21 ee 3b e1 fd 0f 36 90
00000140: 92 c4 a2 35 26 e8 dc 4e b8 ef 89 40 70 d2 91 39
00000150: bc 79 a6 e2 f7 c1 06 bd d5 d6 ff 72 a5 6c f2 c0
00000160: c3 75 e9 ca 67 81 c1 93 96 b4 bd 18 12 4c 37 f7
00000170: d9 73 d6 4c 8a a6 c4 0a 24 00 00 19 04 5e 9e 50
00000180: 5f 58 b0 a5 7a 33 45 83 49 66 0f 1c 3c 7a 67 71
00000190: 98 27 00 00 4e 09 00 00 00 30 44 31 20 30 1e 06
000001A0: 03 55 04 03 13 17 49 4b 45 20 49 6e 74 65 72 6f
000001B0: 70 20 54 65 73 74 20 53 65 72 76 65 72 31 13 30
000001C0: 11 06 03 55 04 0a 13 0a 45 4c 56 49 53 2d 50 4c
000001D0: 55 53 31 0b 30 09 06 03 55 04 06 13 02 52 55 29
000001E0: 00 00 95 0e 00 00 00 0c 30 0a 06 00
```

(99) Extracts IV from message (fragment 4)

```
00000000: 00 00 00 00 00 00 00 03
```

(100) Uses previously computed key K3i

```
00000000: 18 63 41 67 49 6e cf 48 56 71 4d aa 42 63 5c 11
00000010: 2e 26 5b e2 7b c7 53 a4 09 82 e5 5a 7e f4 65 4d
```

(101) Composes MGM nonce (fragment 4)

```
00000000: 00 00 00 03 b4 e1 3e 23
```

(102) Extracts ICV from message (fragment 4)

```
00000000: d2 25 f1 d0 38 65 b7 b6
```

(103) Extracts AAD from message (fragment 4)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 08 00 00 00 01 00 00 01 7a 00 00 01 5e
00000020: 00 04 00 04
```

(104) Extracts ciphertext from message (fragment 4)

```
00000000: e0 8a 0b 04 ee f8 47 c2 52 96 71 9f 9d 39 0c 91
00000010: ea 6a 16 7c 80 31 a0 fd 76 cc c4 f1 8f 1a d3 be
00000020: fa 78 6b df c1 c6 73 83 be 36 69 c4 8a 87 ed 11
00000030: 90 31 a8 fd f9 0a 5c e4 d4 23 c9 e6 b3 96 ac b6
00000040: 8e bd fc 27 58 79 9f cc 8b ac 6b 59 e4 70 4b 05
00000050: 23 16 ed 49 25 f3 de 02 2e ce ae 86 e8 b4 ca b4
00000060: 96 ad 5b f6 2b c2 47 33 6f da f3 97 3c 13 ed 1f
00000070: 7a da 93 b5 69 6a b5 10 93 38 75 ea b7 34 a3 87
00000080: b6 83 c7 da 8a a1 d9 2a 0b 22 e2 ab 63 2b 57 2b
00000090: 88 e3 ea be 7b fc dc 26 ac b8 bb 15 96 f9 c2 f4
000000A0: 60 17 e4 09 18 ae 78 b8 73 02 6b 0e 20 cc b1 cd
000000B0: b4 4d 94 7f f3 16 28 9a d2 bd 26 77 4b a5 85 56
000000C0: b1 81 8b 9c c3 0a 7f 67 fe 6a 61 15 f1 45 66 f3
000000D0: 36 fc a5 bb 1f d7 6d e7 1d 9f 3f b5 cc 60 19 48
000000E0: 17 f7 08 28 1c 58 9f 2b 7a 0b b9 50 bd 02 ea b8
000000F0: 1e 03 1f 52 6a 7a fc e5 b4 6b 00 cf 0d 83 1f d2
00000100: 3f f2 ad 43 d4 86 6e c1 88 d2 87 d6 1f ac a3 30
00000110: 7b c1 5b 6a 3d 4c 20 72 5d 2c ca bf 87 a2 ce 1d
00000120: b3 fa c7 7c 22 cd 66 fc be 49 22 32 17 ee 6e 5e
00000130: 62 c1 ca 12 2b 5d 3d 7b ae b5 3e 53 c5 98 05 1f
00000140: 42 53 49 d1 2c c2
```

(105) Decrypts ciphertext and verifies ICV using K3i as K_msg, resulting in plaintext (fragment 4)

```
00000000: 08 2a 85 03 07 01 01 03 03 6a 3e 59 0d 72 1e 55
00000010: a3 c0 d1 2f 8a 9b 4e 44 10 58 59 bd 62 9e e7 12
00000020: 31 e5 7d 01 53 f3 84 40 dd ac 73 ed 09 3a 10 d9
00000030: 6e 7f eb 80 6c 11 9e 91 f3 7c 3c b0 55 f7 4b ec
00000040: 0e 78 36 10 95 02 09 86 b3 27 04 2a 83 3c 89 36
00000050: 1b 73 cf 7b c9 e0 df a2 07 12 1e 69 52 4d 89 1b
00000060: de 6e 48 d1 34 fa 21 78 22 88 2e 30 86 c0 80 0a
00000070: 2d 74 af 08 ff 35 75 a5 79 e3 85 40 22 6b a8 42
00000080: f6 72 24 bf 29 87 58 a8 20 29 00 00 08 00 00 40
00000090: 00 2f 00 00 0c 00 00 40 01 00 00 00 04 21 00 00
000000A0: 10 01 00 00 00 00 01 00 00 00 03 00 00 2c 00 00
000000B0: 38 00 00 00 34 01 03 04 05 6c 0c a5 70 03 00 00
000000C0: 08 01 00 00 20 03 00 00 08 01 00 00 21 03 00 00
000000D0: 08 01 00 00 22 03 00 00 08 01 00 00 23 00 00 00
000000E0: 08 05 00 00 00 2d 00 00 28 02 00 00 00 07 01 00
000000F0: 10 08 00 08 00 0a 6f 0a ab 0a 6f 0a ab 07 00 00
00000100: 10 00 00 ff ff 00 00 00 00 ff ff ff ff 29 00 00
00000110: 28 02 00 00 00 07 01 00 10 08 00 08 00 0a 00 00
00000120: 02 0a 00 00 02 07 00 00 10 00 00 ff ff 0a 00 00
00000130: 00 0a 00 00 ff 29 00 00 08 00 00 40 0a 00 00 00
00000140: 08 00 00 40 0b 00
```

(106) Reassembles message from received fragments and parses it

```
IKE SA Auth
#9280E0822E758778.DB578D97DE119D1E.00000001 IKEv2 I->R[1847]
  4*EF[...]->E[1819]{
    IDi[78](DN){CN=IKE Interop Test Client,O=ELVIS-PLUS,C=RU},
    CERT[1280](X.509 Cert){308204...A6C40A},
    CERTREQ[25](X.509 Cert){5E9E50...677198},
    IDr[78](DN){CN=IKE Interop Test Server,O=ELVIS-PLUS,C=RU},
    AUTH[149](Sig){id-tc26-signwithdigest-gost3410-12-512[12]:
              6A3E59...58A820},
    N[8](INITIAL_CONTACT),
    N[12](SET_WINDOW_SIZE){4},
    CP[16](REQUEST){IP4.Address[0], IP4.DNS[0]},
    SA[56]{
      P[52](#1:ESP:6C0CA570:5#){
        Encryption=ENCR_KUZNYECHIK_MGM_KTREE,
                   ENCR_MAGMA_MGM_KTREE,
                   ENCR_KUZNYECHIK_MGM_MAC_KTREE,
                   ENCR_MAGMA_MGM_MAC_KTREE,
        ESN=Off}},
    TSi[40](2#){10.111.10.171:icmp:8.0, 0.0.0.0-255.255.255.255},
    TSr[40](2#){10.0.0.2:icmp:8.0, 10.0.0.0-10.0.0.255},
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(107) Computes prf(SK_pi, IDi)

```
00000000: ce e8 8b d1 7e 3c 83 32 eb d1 29 08 de dc 71 f4
00000010: 8f ba 09 b8 ca 5b 10 e2 f4 44 29 5c 97 7b 26 01
00000020: a4 ba 83 c8 ea 40 92 0f 88 18 bd e7 e1 c9 45 cf
00000030: ff 99 48 05 0d f4 93 a6 cd 54 46 d7 eb 7a 52 94
```

(108) Uses initiator's public key

```
00000010: EE 2F 0A 0E 09 1E 7E 04 EF BA 5B 62 A2 52 86 E1
00000020: 9C 24 50 30 50 B0 B4 8A 37 35 B5 FC AF 28 94 EC
00000030: B5 9B 92 41 5B 69 E2 C9 BA 24 DE 6A 72 C4 EF 44
00000040: BB 89 A1 05 14 1B 87 3D 6A A3 72 3E 17 CA 7F 39
00000050: 28 CE 16 8B DD 07 52 87 6A 0D 77 42 6D 99 2B 46
00000060: 2C FD 4B B2 7C D7 C7 17 08 12 54 63 47 9D 14 3D
00000070: 61 ED F2 95 AB 11 80 69 02 A7 66 60 50 7E A4 53
00000080: 6D AD 01 49 B2 16 8A 95 1D CF 1A 57 93 56 14 5E
```

(109) Verifies signature from AUTH payload using algorithm id-tc26-signwithdigest-gost3410-12-512

```
00000000: 6a 3e 59 0d 72 1e 55 a3 c0 d1 2f 8a 9b 4e 44 10
00000010: 58 59 bd 62 9e e7 12 31 e5 7d 01 53 f3 84 40 dd
00000020: ac 73 ed 09 3a 10 d9 6e 7f eb 80 6c 11 9e 91 f3
00000030: 7c 3c b0 55 f7 4b ec 0e 78 36 10 95 02 09 86 b3
00000040: 27 04 2a 83 3c 89 36 1b 73 cf 7b c9 e0 df a2 07
00000050: 12 1e 69 52 4d 89 1b de 6e 48 d1 34 fa 21 78 22
00000060: 88 2e 30 86 c0 80 0a 2d 74 af 08 ff 35 75 a5 79
00000070: e3 85 40 22 6b a8 42 f6 72 24 bf 29 87 58 a8 20
```

(110) Computes keys for ESP SAs

```
00000000: 98 ab 7e db 78 03 a1 e6 c7 21 43 ee b9 7f 5f 56
00000010: 45 bb 51 cd 0b b7 09 a1 af 34 02 87 69 4d 7b a0
00000020: 1d 14 a0 cc
00000000: 70 31 4d 57 94 8b 7e 5c 6f 29 d5 68 1b fd 43 2b
00000010: 19 4e 64 6d 8f 8a 8d 1e ba 72 24 59 c7 0c de 81
00000020: e2 04 84 af
```

(111) Computes prf(SK_pr,IDr)

```
00000000: 7d c8 6a 33 12 02 5c 21 1f ab dc 83 0b 01 a5 27
00000010: 82 a2 f2 1f 64 c6 e9 5e 0e c0 4c e5 d9 11 8d 8e
00000020: b9 5c ef fa b0 a3 37 75 94 20 7c e4 60 60 ed 9d
00000030: fa 5e cb 7e e7 79 05 ab fb 51 1b 03 a8 2c c5 6a
```

(112) Uses private key for signing (little endian)

```
00000000: CB 73 0C 81 6F AC 6D 81 9F 82 AE 15 A9 08 12 17
00000010: D3 1B 97 64 B7 1C 34 0D D3 DD 90 1F 15 8C 9B 06
```

(113) Uses random number for signing

```
00000000: 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
00000010: 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
```

(114) Computes signature using algorithm id-tc26-signwithdigest-gost3410-12-256

```
00000000: c8 40 af f7 46 6f 7b eb d2 b9 1c 5a 80 d0 00 93
00000010: c2 5e 44 16 40 47 f7 8e 61 9c da a5 16 94 83 c5
00000020: 68 5f e8 4d 03 e7 c2 cd 08 07 b8 f3 46 66 6d 05
00000030: 76 c0 d5 e7 60 1d 59 49 09 45 52 c4 95 a7 5a d3
```

(115) Computes K1r (i1 = 0)

```
00000000: 35 e4 d1 65 2e ec 24 89 e4 c9 58 b1 b9 05 1b 83
00000010: 62 5e 65 d7 61 73 d9 1c cf 84 60 64 b9 f2 e7 51
```

(116) Computes K2r (i2 = 0)

```
00000000: 86 8c 89 42 41 d7 30 da 1a 4a 67 69 3a 32 4d 38
00000010: f3 54 02 9f f7 7d b7 bc 5a ee 3b 60 2b 3f 05 56
```

(117) Computes K3r (i3 = 0)

```
00000000: 31 95 e8 c6 67 af 42 d8 ce f1 e8 99 c6 8b 2a c2
00000010: 29 aa 3d c0 ff 18 5f 3d 79 4a 14 6b 9f ac d0 bb
```

(118) Selects SPI for incoming ESP SA

```
00000000: 34 ff 8a 25
```

(119) Creates message splitting it into 4 fragments

```
IKE SA Auth
#9280E0822E758778.DB578D97DE119D1E.00000001 IKEv2 I<=R[1563]
  E[1535]->4*EF[...]{
    IDr[78](DN){CN=IKE Interop Test Server,O=ELVIS-PLUS,C=RU},
    CERT[1211](X.509 Cert){308204...FB346D},
    AUTH[85](Sig){id-tc26-signwithdigest-gost3410-12-256[12]:
             C840AF...A75AD3},
    N[8](INITIAL_CONTACT),
    N[12](SET_WINDOW_SIZE){64},
    CP[16](REPLY){IP4.Address[4]=10.1.1.3},
    SA[32]{
      P[28](#1:ESP:34FF8A25:2#){
        Encryption=ENCR_MAGMA_MGM_KTREE,
        ESN=Off}},
    TSi[24](1#){10.1.1.3},
    TSr[24](1#){10.0.0.0-10.0.0.255},
    N[8](ADDITIONAL_TS_POSSIBLE),
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(120) Composes MGM nonce (fragment 1)

```
00000000: 00 00 00 00 a5 bb 18 2f
```

(121) Composes AAD (fragment 1)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 20 00 00 00 01 00 00 02 20 24 00 02 04
00000020: 00 01 00 04
```

(122) Composes plaintext (fragment 1)

```
00000000: 25 00 00 4e 09 00 00 00 30 44 31 20 30 1e 06 03
00000010: 55 04 03 13 17 49 4b 45 20 49 6e 74 65 72 6f 70
00000020: 20 54 65 73 74 20 53 65 72 76 65 72 31 13 30 11
00000030: 06 03 55 04 0a 13 0a 45 4c 56 49 53 2d 50 4c 55
00000040: 53 31 0b 30 09 06 03 55 04 06 13 02 52 55 27 00
00000050: 04 bb 04 30 82 04 b2 30 82 04 5f a0 03 02 01 02
00000060: 02 13 7c 00 03 d9 02 ec f9 34 3e c8 aa d6 59 00
00000070: 01 00 03 d9 02 30 0a 06 08 2a 85 03 07 01 01 03
00000080: 02 30 82 01 0a 31 18 30 16 06 05 2a 85 03 64 01
00000090: 12 0d 31 32 33 34 35 36 37 38 39 30 31 32 33 31
000000A0: 1a 30 18 06 08 2a 85 03 03 81 03 01 01 12 0c 30
000000B0: 30 31 32 33 34 35 36 37 38 39 30 31 2f 30 2d 06
000000C0: 03 55 04 09 0c 26 d1 83 d0 bb 2e 20 d0 a1 d1 83
000000D0: d1 89 d1 91 d0 b2 d1 81 d0 ba d0 b8 d0 b9 20 d0
000000E0: b2 d0 b0 d0 bb 20 d0 b4 2e 20 31 38 31 0b 30 09
000000F0: 06 03 55 04 06 13 02 52 55 31 19 30 17 06 03 55
00000100: 04 08 0c 10 d0 b3 2e 20 d0 9c d0 be d1 81 d0 ba
00000110: d0 b2 d0 b0 31 15 30 13 06 03 55 04 07 0c 0c d0
00000120: 9c d0 be d1 81 d0 ba d0 b2 d0 b0 31 25 30 23 06
00000130: 03 55 04 0a 0c 1c d0 9e d0 9e d0 9e 20 22 d0 9a
00000140: d0 a0 d0 98 d0 9f d0 a2 d0 9e 2d d0 9f d0 a0 d0
00000150: 9e 22 31 3b 30 39 06 03 55 04 03 0c 32 d0 a2 d0
00000160: b5 d1 81 d1 82 d0 be d0 b2 d1 8b d0 b9 20 d0 a3
00000170: d0 a6 20 d0 9e d0 9e d0 9e 20 22 d0 9a d0 a0 d0
00000180: 98 d0 9f d0 a2 d0 9e 2d d0 9f d0 a0 d0 9e 22 30
00000190: 1e 17 0d 32 31 30 39 33 30 31 33 32 34 30 36 5a
000001A0: 17 0d 32 31 31 32 33 30 31 33 33 34 30 36 5a 30
000001B0: 44 31 20 30 1e 06 03 55 04 03 13 17 49 4b 45 20
000001C0: 49 6e 74 65 72 6f 70 20 54 65 73 74 20 53 65 72
000001D0: 76 65 72 31 13 30 11 06 03 55 04 0a 13 0a 45 4c
000001E0: 56 49 53 2d 50 4c 55 53 31 0b 30 00
```

(123) Encrypts plaintext using K3r as K_msg, resulting in ciphertext (fragment 1)

```
00000000: 73 f2 45 3e fb 6a 26 28 67 7d 14 e3 bf 0a 90 74
00000010: c9 95 6a 40 d5 4e a6 77 cf 58 2e b8 ae 52 f4 25
00000020: f7 82 bc d9 f0 74 4e 38 51 90 07 70 27 f8 01 27
00000030: 17 da f4 ba bc 1e 02 0b 73 ec cc 7b f8 b3 68 64
00000040: f3 48 65 33 3b ab ac 19 11 d3 f7 78 b4 f8 d1 3f
00000050: 6d 46 93 37 a6 58 48 3a 7d d0 8a 9c 84 ab de eb
00000060: 0d d4 8d ab 75 20 18 27 42 fe 24 ee ba c4 a4 6e
00000070: db 80 68 3c 84 7e d6 36 50 d4 1b 1c bc c5 9f 18
00000080: 41 af 48 52 c1 7e a2 f0 e4 bc 0a 3c 64 34 81 ca
00000090: df 96 ba 51 91 f1 06 13 b2 04 23 c8 70 3a ea 64
000000A0: e9 ea ce c2 db aa 12 90 28 0c 9d f9 89 02 a8 5e
000000B0: 66 f5 6e ce dd e7 2c 4a 45 54 de 5e b8 76 73 67
000000C0: 2d a3 a0 52 91 74 ff b7 eb e4 ea d1 2b 04 76 f7
000000D0: ff 4b 1c b8 45 7e 8a 60 e7 1e ec 13 3e c1 d8 d0
000000E0: 78 be f4 79 77 06 ce 76 04 64 ad e7 10 19 65 2b
000000F0: 45 66 23 3d 34 7a 40 6c 36 c0 20 73 47 d8 7a b6
00000100: 2b 0f 56 04 7a c0 41 ab 18 23 11 78 7f 4f d4 f5
00000110: 7d 2e 06 a5 15 ee de 84 9f c2 0a f6 c8 1e a4 30
00000120: 70 42 07 c8 5e 97 08 69 12 27 58 c3 c7 b7 db 7a
00000130: 8c 50 3a 3a 5c bf 3a a7 73 40 8f 9c 18 f6 13 77
00000140: 63 c1 60 06 36 a1 43 ab 88 08 c9 cc ad f2 88 ca
00000150: 84 bd 45 e0 8e d9 27 a3 07 f2 63 79 b0 a8 62 9f
00000160: 5f ba dc a7 f5 54 b8 4f 4f bb 1e a2 16 4b 4f 2d
00000170: d4 08 4e 45 c2 c0 60 3b 73 df 6b 35 3a fe 38 2e
00000180: 25 75 fc be 89 4c d2 7a 9c 1f b4 41 a6 31 d3 3d
00000190: 39 a6 d1 c4 47 94 44 30 3a 2b 23 22 ba c0 a9 df
000001A0: dc 1c 90 8d d1 e8 13 f9 08 68 5a 94 98 c7 3f 47
000001B0: 77 79 b5 bb fb 22 56 4b 38 55 48 e8 14 d4 01 eb
000001C0: 63 e9 17 da 24 69 9a 6d dc 1e 25 06 ef 77 10 46
000001D0: ad 99 ad 9c 54 4f d4 68 64 ea 05 1d ef 29 ea 0e
000001E0: 3c 1c 7e 27 cf 59 76 42 5b 02 04 b8
```

(124) Computes ICV using K3r as K_msg (fragment 1)

```
00000000: 96 08 17 ed ef 01 4d a0
```

(125) Composes IV (fragment 1)

```
00000000: 00 00 00 00 00 00 00 00
```

(126) Composes MGM nonce (fragment 2)

```
00000000: 00 00 00 01 a5 bb 18 2f
```

(127) Composes AAD (fragment 2)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 20 00 00 00 01 00 00 02 20 00 00 02 04
00000020: 00 02 00 04
```

(128) Composes plaintext (fragment 2)

```
00000000: 09 06 03 55 04 06 13 02 52 55 30 66 30 1f 06 08
00000010: 2a 85 03 07 01 01 01 01 30 13 06 07 2a 85 03 02
00000020: 02 24 00 06 08 2a 85 03 07 01 01 02 02 03 43 00
00000030: 04 40 5b b3 14 3e f4 70 c1 70 d7 f3 27 25 d8 53
00000040: 7c e6 de 6d 8c 29 f6 b2 32 64 56 dc b1 77 f2 3d
00000050: fa f4 2a 5c f3 74 86 7f 04 72 51 c1 cf b3 43 36
00000060: f5 95 a2 af 05 47 57 1a 55 c0 78 a4 9d 64 26 b8
00000070: 61 14 a3 82 02 59 30 82 02 55 30 0e 06 03 55 1d
00000080: 0f 01 01 ff 04 04 03 02 05 a0 30 13 06 03 55 1d
00000090: 25 04 0c 30 0a 06 08 2b 06 01 05 05 07 03 11 30
000000A0: 1d 06 03 55 1d 0e 04 16 04 14 e0 d3 f0 09 ad ce
000000B0: 6c a5 47 ba 9b f7 a6 a5 1b 06 14 ba a5 43 30 1f
000000C0: 06 03 55 1d 23 04 18 30 16 80 14 9b 85 5e fb 81
000000D0: dc 4d 59 07 51 63 cf be df da 2c 7f c9 44 3c 30
000000E0: 82 01 0f 06 03 55 1d 1f 04 82 01 06 30 82 01 02
000000F0: 30 81 ff a0 81 fc a0 81 f9 86 81 b5 68 74 74 70
00000100: 3a 2f 2f 74 65 73 74 67 6f 73 74 32 30 31 32 2e
00000110: 63 72 79 70 74 6f 70 72 6f 2e 72 75 2f 43 65 72
00000120: 74 45 6e 72 6f 6c 6c 2f 21 30 34 32 32 21 30 34
00000130: 33 35 21 30 34 34 31 21 30 34 34 32 21 30 34 33
00000140: 65 21 30 34 33 32 21 30 34 34 62 21 30 34 33 39
00000150: 25 32 30 21 30 34 32 33 21 30 34 32 36 25 32 30
00000160: 21 30 34 31 65 21 30 34 31 65 21 30 34 31 65 25
00000170: 32 30 21 30 30 32 32 21 30 34 31 61 21 30 34 32
00000180: 30 21 30 34 31 38 21 30 34 31 66 21 30 34 32 32
00000190: 21 30 34 31 65 2d 21 30 34 31 66 21 30 34 32 30
000001A0: 21 30 34 31 65 21 30 30 32 32 28 31 29 2e 63 72
000001B0: 6c 86 3f 68 74 74 70 3a 2f 2f 74 65 73 74 67 6f
000001C0: 73 74 32 30 31 32 2e 63 72 79 70 74 6f 70 72 6f
000001D0: 2e 72 75 2f 43 65 72 74 45 6e 72 6f 6c 6c 2f 74
000001E0: 65 73 74 67 6f 73 74 32 30 31 32 00
```

(129) Encrypts plaintext using K3r as K_msg, resulting in ciphertext (fragment 2)

```
00000000: b1 c8 8d ae d9 6f 91 7e 5a 6a 2d 8c e0 d6 28 3e
00000010: 10 59 46 12 a1 1e fa 53 c3 58 ec 4e a9 a5 92 0c
00000020: fa 5e cf a3 33 4a 8b b7 56 66 54 d9 9c 64 2e b6
00000030: 4d 03 3f 77 a8 17 88 f6 23 e0 2e 56 a6 a2 4c 4d
00000040: 6e e3 09 8a 2e 31 a1 85 1c cf ce 95 e7 73 93 8e
00000050: 9c 5a 7b 3b 49 75 96 69 d4 b0 46 f7 74 b0 0d 5d
00000060: 91 3b 6d 2b a4 46 cc 5c d9 a8 38 c0 6b ad 73 35
00000070: 09 aa c7 4c 91 8a 84 1c dd 3f e1 44 f7 c5 9c 61
00000080: 0e b7 03 6b 84 cc 8e 93 5b d5 f6 7e 71 3a f4 2c
00000090: 98 14 ad 47 e3 c3 70 dc e3 3e c0 a5 e0 e4 6d 01
000000A0: 44 78 7f e3 b7 6c cb 44 29 59 96 e9 84 6d 9d 18
000000B0: 89 66 16 07 46 a4 cd 72 a6 0e bd d2 a7 1c f7 21
000000C0: f0 d1 67 a9 0d 1c c4 c8 30 bd 26 1f 53 7d 61 8b
000000D0: ad 6f ef 3e 2c 6e 7e 69 b9 92 72 66 65 b6 06 22
000000E0: 49 a1 a8 f1 2f 02 dd 41 bf f5 d1 f6 7c 93 25 6e
000000F0: 52 8b a9 3f b5 40 97 02 bb 7c f5 33 a6 60 52 b8
00000100: 4f 3e 80 6c 38 cf e4 8b 15 fd d0 66 75 c1 bf bb
00000110: ac fc ac 01 c3 11 8e 0b 3e e9 2c 1b 5d b9 9f f6
00000120: 2f d7 e8 3c c7 a9 25 8b aa 6e c6 49 6d 6f df 42
00000130: 53 0e ba 70 54 d2 af c3 4d 02 e1 48 42 c5 45 53
00000140: 25 59 66 25 c7 3c c6 c2 e2 99 e2 bb 47 a4 a7 be
00000150: 6c 92 0d 3b 4c ab 6e d7 23 05 ea 73 07 62 e8 c0
00000160: e8 78 47 af 54 c8 67 8f dd 32 59 8d 87 ac 42 0e
00000170: 21 15 c4 f7 66 dc 02 cf 55 c2 e3 4d 8e 91 7a fd
00000180: d7 4d 20 b0 6f 67 78 58 08 9c ba 05 8b b0 9c 16
00000190: 20 51 75 12 96 e2 d5 28 ac 3e 50 26 04 6f 59 02
000001A0: 28 e0 ec 2c da 70 4a 9c 15 5a 2e 52 01 e6 4e 1e
000001B0: 10 6d 8d 5d 2a 81 69 0e 54 d0 5e 13 82 82 84 9a
000001C0: ac a6 0e 69 4e 17 5c c1 8a 71 f8 b4 80 3b 7a e5
000001D0: b8 1f 09 4a 02 14 24 07 af 6a 14 d9 52 8e da d3
000001E0: 58 23 68 71 27 b2 9a 03 09 f7 80 51
```

(130) Computes ICV using K3r as K_msg (fragment 2)

```
00000000: 89 bd 07 12 fc 3f 15 8d
```

(131) Composes IV (fragment 2)

```
00000000: 00 00 00 00 00 00 00 01
```

(132) Composes MGM nonce (fragment 3)

```
00000000: 00 00 00 02 a5 bb 18 2f
```

(133) Composes AAD (fragment 3)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 20 00 00 00 01 00 00 02 20 00 00 02 04
00000020: 00 03 00 04
```

(134) Composes plaintext (fragment 3)

```
00000000: 28 31 29 2e 63 72 6c 30 81 da 06 08 2b 06 01 05
00000010: 05 07 01 01 04 81 cd 30 81 ca 30 44 06 08 2b 06
00000020: 01 05 05 07 30 02 86 38 68 74 74 70 3a 2f 2f 74
00000030: 65 73 74 67 6f 73 74 32 30 31 32 2e 63 72 79 70
00000040: 74 6f 70 72 6f 2e 72 75 2f 43 65 72 74 45 6e 72
00000050: 6f 6c 6c 2f 72 6f 6f 74 32 30 31 38 2e 63 72 74
00000060: 30 3f 06 08 2b 06 01 05 05 07 30 01 86 33 68 74
00000070: 74 70 3a 2f 2f 74 65 73 74 67 6f 73 74 32 30 31
00000080: 32 2e 63 72 79 70 74 6f 70 72 6f 2e 72 75 2f 6f
00000090: 63 73 70 32 30 31 32 67 2f 6f 63 73 70 2e 73 72
000000A0: 66 30 41 06 08 2b 06 01 05 05 07 30 01 86 35 68
000000B0: 74 74 70 3a 2f 2f 74 65 73 74 67 6f 73 74 32 30
000000C0: 31 32 2e 63 72 79 70 74 6f 70 72 6f 2e 72 75 2f
000000D0: 6f 63 73 70 32 30 31 32 67 73 74 2f 6f 63 73 70
000000E0: 2e 73 72 66 30 0a 06 08 2a 85 03 07 01 01 03 02
000000F0: 03 41 00 a5 39 5f ca 48 e1 c2 93 c1 e0 8a 64 74
00000100: 0f 6b 86 a2 15 9b 46 29 d0 42 71 4f ce e7 52 d7
00000110: d7 3d aa 47 ce cf 52 63 8f 26 b2 17 5f ad 96 57
00000120: 76 ea 5f d0 87 bb 12 29 e4 06 0e e1 5f fd 59 81
00000130: fb 34 6d 29 00 00 55 0e 00 00 00 0c 30 0a 06 08
00000140: 2a 85 03 07 01 01 03 02 c8 40 af f7 46 6f 7b eb
00000150: d2 b9 1c 5a 80 d0 00 93 c2 5e 44 16 40 47 f7 8e
00000160: 61 9c da a5 16 94 83 c5 68 5f e8 4d 03 e7 c2 cd
00000170: 08 07 b8 f3 46 66 6d 05 76 c0 d5 e7 60 1d 59 49
00000180: 09 45 52 c4 95 a7 5a d3 29 00 00 08 00 00 40 00
00000190: 2f 00 00 0c 00 00 40 01 00 00 00 40 21 00 00 10
000001A0: 02 00 00 00 00 01 00 04 0a 01 01 03 2c 00 00 20
000001B0: 00 00 00 1c 01 03 04 02 34 ff 8a 25 03 00 00 08
000001C0: 01 00 00 21 00 00 00 08 05 00 00 00 2d 00 00 18
000001D0: 01 00 00 00 07 00 00 10 00 00 ff ff 0a 01 01 03
000001E0: 0a 01 01 03 29 00 00 18 01 00 00 00
```

(135) Encrypts plaintext using K3r as K_msg, resulting in ciphertext (fragment 3)

```
00000000: 08 e0 86 04 1f 8a c9 b5 68 cd 96 10 ab 59 99 3a
00000010: 54 7b a9 fa d7 60 46 ec c3 bf bd 8f fa 03 ed 41
00000020: 49 13 ca 8c 9c b8 0c df 81 25 e2 30 ca cb 65 b9
00000030: 16 55 8e 67 f4 b3 7c b8 91 66 76 7c a4 15 98 a3
00000040: 3a c9 48 64 e4 ce 9f 64 67 5d bb 7c 03 23 9e c9
00000050: 81 3f da 48 ee a6 2a d8 fb ac 77 ce ed c2 a4 d9
00000060: 24 d3 71 99 fc 71 2b 6c 10 d3 c3 4b b5 37 e2 55
00000070: 5f d5 ee c0 d6 ff 66 15 8c e5 63 26 96 cd 3f 49
00000080: 2b da 51 94 55 6e 2e e5 2e d1 b4 91 81 50 85 8a
00000090: 84 bd fe 52 ec ce 1b 6b bd 7d 12 b4 de a5 88 c4
000000A0: b7 78 d3 3d 2d 46 ef dc 0f 91 43 be 08 7a ba fa
000000B0: b3 2a c2 17 30 99 79 ae 3a 00 f0 3f 47 4a 9b 11
000000C0: 4d 7b 1b 28 0a 44 5b 1a af 35 4d c3 2b 6b be 11
000000D0: 89 03 b9 de cf 37 57 53 1e a4 f3 3f ce 52 a6 d8
000000E0: 7e 9d d8 d4 2f 9f f5 8f 3c c6 cb 2f 56 e0 97 2d
000000F0: b2 0e 10 66 3b 3c ec 34 50 99 a3 7d 42 ec 96 eb
00000100: 87 48 72 2c 0a 6d af b9 4b 62 48 89 36 01 21 ab
00000110: 8e 79 10 54 9c 83 ab a9 8a 6c 37 c7 ac dc a1 7e
00000120: 41 0e 58 de da aa 95 71 fb 34 50 8a ef 37 0b c4
00000130: 56 ca 4b 2c 75 b7 c7 d9 74 22 c2 65 1a e4 4f 94
00000140: 20 f6 e9 44 f1 69 5e d2 18 d3 30 2e 85 74 25 be
00000150: 2a 88 e2 ce fe 75 ca fa 25 f9 2e 88 8c ed 6f dd
00000160: c3 c5 53 2e da 14 fd 96 28 4a b7 81 3a b3 d5 44
00000170: 26 e2 84 21 f2 5c 0a ed bf c4 34 1c a4 91 5e f3
00000180: 47 ef 0e 9e fb ee 34 95 5d 21 72 43 c9 63 af b4
00000190: f2 98 4a 36 57 77 fc e7 57 52 b2 4d bf 34 2a 98
000001A0: ea 70 cd d7 a9 da 4c 0d 19 05 d4 1e dd 36 c7 c4
000001B0: 31 54 18 2a ef 0e 30 44 97 31 15 57 cd d4 88 52
000001C0: 4e 42 c8 20 89 8d 35 7b 8e 03 96 b4 74 fb ec 3b
000001D0: 14 c2 64 49 92 f2 1f 3d ff 84 2d 92 4c b9 01 04
000001E0: 3d 0a 2a 28 33 de 43 44 6b cf 79 0e
```

(136) Computes ICV using K3r as K_msg (fragment 3)

```
00000000: 7d 7c 57 8f 91 d0 c9 eb
```

(137) Composes IV (fragment 3)

```
00000000: 00 00 00 00 00 00 00 02
```

(138) Composes MGM nonce (fragment 4)

```
00000000: 00 00 00 03 a5 bb 18 2f
```

(139) Composes AAD (fragment 4)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 20 00 00 00 01 00 00 00 5e 00 00 00 42
00000020: 00 04 00 04
```

(140) Composes plaintext (fragment 4)

```
00000000: 00 07 00 00 10 00 00 ff ff 0a 00 00 00 0a 00 00
00000010: ff 29 00 00 08 00 00 40 02 29 00 00 08 00 00 40
00000020: 0a 00 00 00 08 00 00 40 0b 00
```

(141) Encrypts plaintext using K3r as K_msg, resulting in ciphertext (fragment 4)

```
00000000: 81 fa 5d 7a 67 13 b7 93 f4 2c 01 b8 d1 02 8c ab
00000010: 8e 80 47 25 6e c5 69 e3 0c 84 cd 35 9a 0f 7a cc
00000020: 0a 92 7a 74 77 dc ba 60 ac 4a
```

(142) Computes ICV using K3r as K_msg (fragment 4)

```
00000000: 6c 27 70 e0 8a 82 bd 4b
```

(143) Composes IV (fragment 4)

```
00000000: 00 00 00 00 00 00 00 03
```

(144) Sends message fragment (1), peer receives message fragment (1)

```
       10.111.10.171:54295<-10.111.15.45:4500 [548]

00000000: 00 00 00 00 92 80 e0 82 2e 75 87 78 db 57 8d 97
00000010: de 11 9d 1e 35 20 23 20 00 00 00 01 00 00 02 20
00000020: 24 00 02 04 00 01 00 04 00 00 00 00 00 00 00 00
00000030: 73 f2 45 3e fb 6a 26 28 67 7d 14 e3 bf 0a 90 74
00000040: c9 95 6a 40 d5 4e a6 77 cf 58 2e b8 ae 52 f4 25
00000050: f7 82 bc d9 f0 74 4e 38 51 90 07 70 27 f8 01 27
00000060: 17 da f4 ba bc 1e 02 0b 73 ec cc 7b f8 b3 68 64
00000070: f3 48 65 33 3b ab ac 19 11 d3 f7 78 b4 f8 d1 3f
00000080: 6d 46 93 37 a6 58 48 3a 7d d0 8a 9c 84 ab de eb
00000090: 0d d4 8d ab 75 20 18 27 42 fe 24 ee ba c4 a4 6e
000000A0: db 80 68 3c 84 7e d6 36 50 d4 1b 1c bc c5 9f 18
000000B0: 41 af 48 52 c1 7e a2 f0 e4 bc 0a 3c 64 34 81 ca
000000C0: df 96 ba 51 91 f1 06 13 b2 04 23 c8 70 3a ea 64
000000D0: e9 ea ce c2 db aa 12 90 28 0c 9d f9 89 02 a8 5e
000000E0: 66 f5 6e ce dd e7 2c 4a 45 54 de 5e b8 76 73 67
000000F0: 2d a3 a0 52 91 74 ff b7 eb e4 ea d1 2b 04 76 f7
00000100: ff 4b 1c b8 45 7e 8a 60 e7 1e ec 13 3e c1 d8 d0
00000110: 78 be f4 79 77 06 ce 76 04 64 ad e7 10 19 65 2b
00000120: 45 66 23 3d 34 7a 40 6c 36 c0 20 73 47 d8 7a b6
00000130: 2b 0f 56 04 7a c0 41 ab 18 23 11 78 7f 4f d4 f5
00000140: 7d 2e 06 a5 15 ee de 84 9f c2 0a f6 c8 1e a4 30
00000150: 70 42 07 c8 5e 97 08 69 12 27 58 c3 c7 b7 db 7a
00000160: 8c 50 3a 3a 5c bf 3a a7 73 40 8f 9c 18 f6 13 77
00000170: 63 c1 60 06 36 a1 43 ab 88 08 c9 cc ad f2 88 ca
00000180: 84 bd 45 e0 8e d9 27 a3 07 f2 63 79 b0 a8 62 9f
00000190: 5f ba dc a7 f5 54 b8 4f 4f bb 1e a2 16 4b 4f 2d
000001A0: d4 08 4e 45 c2 c0 60 3b 73 df 6b 35 3a fe 38 2e
000001B0: 25 75 fc be 89 4c d2 7a 9c 1f b4 41 a6 31 d3 3d
000001C0: 39 a6 d1 c4 47 94 44 30 3a 2b 23 22 ba c0 a9 df
000001D0: dc 1c 90 8d d1 e8 13 f9 08 68 5a 94 98 c7 3f 47
000001E0: 77 79 b5 bb fb 22 56 4b 38 55 48 e8 14 d4 01 eb
000001F0: 63 e9 17 da 24 69 9a 6d dc 1e 25 06 ef 77 10 46
00000200: ad 99 ad 9c 54 4f d4 68 64 ea 05 1d ef 29 ea 0e
00000210: 3c 1c 7e 27 cf 59 76 42 5b 02 04 b8 96 08 17 ed
00000220: ef 01 4d a0
```

(145) Sends message fragment (2), peer receives message fragment (2)

```
       10.111.10.171:54295<-10.111.15.45:4500 [548]

00000000: 00 00 00 00 92 80 e0 82 2e 75 87 78 db 57 8d 97
00000010: de 11 9d 1e 35 20 23 20 00 00 00 01 00 00 02 20
00000020: 00 00 02 04 00 02 00 04 00 00 00 00 00 00 00 01
00000030: b1 c8 8d ae d9 6f 91 7e 5a 6a 2d 8c e0 d6 28 3e
00000040: 10 59 46 12 a1 1e fa 53 c3 58 ec 4e a9 a5 92 0c
00000050: fa 5e cf a3 33 4a 8b b7 56 66 54 d9 9c 64 2e b6
00000060: 4d 03 3f 77 a8 17 88 f6 23 e0 2e 56 a6 a2 4c 4d
00000070: 6e e3 09 8a 2e 31 a1 85 1c cf ce 95 e7 73 93 8e
00000080: 9c 5a 7b 3b 49 75 96 69 d4 b0 46 f7 74 b0 0d 5d
00000090: 91 3b 6d 2b a4 46 cc 5c d9 a8 38 c0 6b ad 73 35
000000A0: 09 aa c7 4c 91 8a 84 1c dd 3f e1 44 f7 c5 9c 61
000000B0: 0e b7 03 6b 84 cc 8e 93 5b d5 f6 7e 71 3a f4 2c
000000C0: 98 14 ad 47 e3 c3 70 dc e3 3e c0 a5 e0 e4 6d 01
000000D0: 44 78 7f e3 b7 6c cb 44 29 59 96 e9 84 6d 9d 18
000000E0: 89 66 16 07 46 a4 cd 72 a6 0e bd d2 a7 1c f7 21
000000F0: f0 d1 67 a9 0d 1c c4 c8 30 bd 26 1f 53 7d 61 8b
00000100: ad 6f ef 3e 2c 6e 7e 69 b9 92 72 66 65 b6 06 22
00000110: 49 a1 a8 f1 2f 02 dd 41 bf f5 d1 f6 7c 93 25 6e
00000120: 52 8b a9 3f b5 40 97 02 bb 7c f5 33 a6 60 52 b8
00000130: 4f 3e 80 6c 38 cf e4 8b 15 fd d0 66 75 c1 bf bb
00000140: ac fc ac 01 c3 11 8e 0b 3e e9 2c 1b 5d b9 9f f6
00000150: 2f d7 e8 3c c7 a9 25 8b aa 6e c6 49 6d 6f df 42
00000160: 53 0e ba 70 54 d2 af c3 4d 02 e1 48 42 c5 45 53
00000170: 25 59 66 25 c7 3c c6 c2 e2 99 e2 bb 47 a4 a7 be
00000180: 6c 92 0d 3b 4c ab 6e d7 23 05 ea 73 07 62 e8 c0
00000190: e8 78 47 af 54 c8 67 8f dd 32 59 8d 87 ac 42 0e
000001A0: 21 15 c4 f7 66 dc 02 cf 55 c2 e3 4d 8e 91 7a fd
000001B0: d7 4d 20 b0 6f 67 78 58 08 9c ba 05 8b b0 9c 16
000001C0: 20 51 75 12 96 e2 d5 28 ac 3e 50 26 04 6f 59 02
000001D0: 28 e0 ec 2c da 70 4a 9c 15 5a 2e 52 01 e6 4e 1e
000001E0: 10 6d 8d 5d 2a 81 69 0e 54 d0 5e 13 82 82 84 9a
000001F0: ac a6 0e 69 4e 17 5c c1 8a 71 f8 b4 80 3b 7a e5
00000200: b8 1f 09 4a 02 14 24 07 af 6a 14 d9 52 8e da d3
00000210: 58 23 68 71 27 b2 9a 03 09 f7 80 51 89 bd 07 12
00000220: fc 3f 15 8d
```

(146) Sends message fragment (3), peer receives message fragment (3)

```
       10.111.10.171:54295<-10.111.15.45:4500 [548]

00000000: 00 00 00 00 92 80 e0 82 2e 75 87 78 db 57 8d 97
00000010: de 11 9d 1e 35 20 23 20 00 00 00 01 00 00 02 20
00000020: 00 00 02 04 00 03 00 04 00 00 00 00 00 00 00 02
00000030: 08 e0 86 04 1f 8a c9 b5 68 cd 96 10 ab 59 99 3a
00000040: 54 7b a9 fa d7 60 46 ec c3 bf bd 8f fa 03 ed 41
00000050: 49 13 ca 8c 9c b8 0c df 81 25 e2 30 ca cb 65 b9
00000060: 16 55 8e 67 f4 b3 7c b8 91 66 76 7c a4 15 98 a3
00000070: 3a c9 48 64 e4 ce 9f 64 67 5d bb 7c 03 23 9e c9
00000080: 81 3f da 48 ee a6 2a d8 fb ac 77 ce ed c2 a4 d9
00000090: 24 d3 71 99 fc 71 2b 6c 10 d3 c3 4b b5 37 e2 55
000000A0: 5f d5 ee c0 d6 ff 66 15 8c e5 63 26 96 cd 3f 49
000000B0: 2b da 51 94 55 6e 2e e5 2e d1 b4 91 81 50 85 8a
000000C0: 84 bd fe 52 ec ce 1b 6b bd 7d 12 b4 de a5 88 c4
000000D0: b7 78 d3 3d 2d 46 ef dc 0f 91 43 be 08 7a ba fa
000000E0: b3 2a c2 17 30 99 79 ae 3a 00 f0 3f 47 4a 9b 11
000000F0: 4d 7b 1b 28 0a 44 5b 1a af 35 4d c3 2b 6b be 11
00000100: 89 03 b9 de cf 37 57 53 1e a4 f3 3f ce 52 a6 d8
00000110: 7e 9d d8 d4 2f 9f f5 8f 3c c6 cb 2f 56 e0 97 2d
00000120: b2 0e 10 66 3b 3c ec 34 50 99 a3 7d 42 ec 96 eb
00000130: 87 48 72 2c 0a 6d af b9 4b 62 48 89 36 01 21 ab
00000140: 8e 79 10 54 9c 83 ab a9 8a 6c 37 c7 ac dc a1 7e
00000150: 41 0e 58 de da aa 95 71 fb 34 50 8a ef 37 0b c4
00000160: 56 ca 4b 2c 75 b7 c7 d9 74 22 c2 65 1a e4 4f 94
00000170: 20 f6 e9 44 f1 69 5e d2 18 d3 30 2e 85 74 25 be
00000180: 2a 88 e2 ce fe 75 ca fa 25 f9 2e 88 8c ed 6f dd
00000190: c3 c5 53 2e da 14 fd 96 28 4a b7 81 3a b3 d5 44
000001A0: 26 e2 84 21 f2 5c 0a ed bf c4 34 1c a4 91 5e f3
000001B0: 47 ef 0e 9e fb ee 34 95 5d 21 72 43 c9 63 af b4
000001C0: f2 98 4a 36 57 77 fc e7 57 52 b2 4d bf 34 2a 98
000001D0: ea 70 cd d7 a9 da 4c 0d 19 05 d4 1e dd 36 c7 c4
000001E0: 31 54 18 2a ef 0e 30 44 97 31 15 57 cd d4 88 52
000001F0: 4e 42 c8 20 89 8d 35 7b 8e 03 96 b4 74 fb ec 3b
00000200: 14 c2 64 49 92 f2 1f 3d ff 84 2d 92 4c b9 01 04
00000210: 3d 0a 2a 28 33 de 43 44 6b cf 79 0e 7d 7c 57 8f
00000220: 91 d0 c9 eb
```

(147) Sends message fragment (4), peer receives message fragment (4)

```
       10.111.10.171:54295<-10.111.15.45:4500 [98]

00000000: 00 00 00 00 92 80 e0 82 2e 75 87 78 db 57 8d 97
00000010: de 11 9d 1e 35 20 23 20 00 00 00 01 00 00 00 5e
00000020: 00 00 00 42 00 04 00 04 00 00 00 00 00 00 00 03
00000030: 81 fa 5d 7a 67 13 b7 93 f4 2c 01 b8 d1 02 8c ab
00000040: 8e 80 47 25 6e c5 69 e3 0c 84 cd 35 9a 0f 7a cc
00000050: 0a 92 7a 74 77 dc ba 60 ac 4a 6c 27 70 e0 8a 82
00000060: bd 4b
```

Initiator's actions:

(148) Extracts IV from message (fragment 1)

```
00000000: 00 00 00 00 00 00 00 00
```

(149) Computes K1r (i1 = 0)

```
00000000: 35 e4 d1 65 2e ec 24 89 e4 c9 58 b1 b9 05 1b 83
00000010: 62 5e 65 d7 61 73 d9 1c cf 84 60 64 b9 f2 e7 51
```

(150) Computes K2r (i2 = 0)

```
00000000: 86 8c 89 42 41 d7 30 da 1a 4a 67 69 3a 32 4d 38
00000010: f3 54 02 9f f7 7d b7 bc 5a ee 3b 60 2b 3f 05 56
```

(151) Computes K3r (i3 = 0)

```
00000000: 31 95 e8 c6 67 af 42 d8 ce f1 e8 99 c6 8b 2a c2
00000010: 29 aa 3d c0 ff 18 5f 3d 79 4a 14 6b 9f ac d0 bb
```

(152) Composes MGM nonce (fragment 1)

```
00000000: 00 00 00 00 a5 bb 18 2f
```

(153) Extracts ICV from message (fragment 1)

```
00000000: 96 08 17 ed ef 01 4d a0
```

(154) Extracts AAD from message (fragment 1)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 20 00 00 00 01 00 00 02 20 24 00 02 04
00000020: 00 01 00 04
```

(155) Extracts ciphertext from message (fragment 1)

```
00000000: 73 f2 45 3e fb 6a 26 28 67 7d 14 e3 bf 0a 90 74
00000010: c9 95 6a 40 d5 4e a6 77 cf 58 2e b8 ae 52 f4 25
00000020: f7 82 bc d9 f0 74 4e 38 51 90 07 70 27 f8 01 27
00000030: 17 da f4 ba bc 1e 02 0b 73 ec cc 7b f8 b3 68 64
00000040: f3 48 65 33 3b ab ac 19 11 d3 f7 78 b4 f8 d1 3f
00000050: 6d 46 93 37 a6 58 48 3a 7d d0 8a 9c 84 ab de eb
00000060: 0d d4 8d ab 75 20 18 27 42 fe 24 ee ba c4 a4 6e
00000070: db 80 68 3c 84 7e d6 36 50 d4 1b 1c bc c5 9f 18
00000080: 41 af 48 52 c1 7e a2 f0 e4 bc 0a 3c 64 34 81 ca
00000090: df 96 ba 51 91 f1 06 13 b2 04 23 c8 70 3a ea 64
000000A0: e9 ea ce c2 db aa 12 90 28 0c 9d f9 89 02 a8 5e
000000B0: 66 f5 6e ce dd e7 2c 4a 45 54 de 5e b8 76 73 67
000000C0: 2d a3 a0 52 91 74 ff b7 eb e4 ea d1 2b 04 76 f7
000000D0: ff 4b 1c b8 45 7e 8a 60 e7 1e ec 13 3e c1 d8 d0
000000E0: 78 be f4 79 77 06 ce 76 04 64 ad e7 10 19 65 2b
000000F0: 45 66 23 3d 34 7a 40 6c 36 c0 20 73 47 d8 7a b6
00000100: 2b 0f 56 04 7a c0 41 ab 18 23 11 78 7f 4f d4 f5
00000110: 7d 2e 06 a5 15 ee de 84 9f c2 0a f6 c8 1e a4 30
00000120: 70 42 07 c8 5e 97 08 69 12 27 58 c3 c7 b7 db 7a
00000130: 8c 50 3a 3a 5c bf 3a a7 73 40 8f 9c 18 f6 13 77
00000140: 63 c1 60 06 36 a1 43 ab 88 08 c9 cc ad f2 88 ca
00000150: 84 bd 45 e0 8e d9 27 a3 07 f2 63 79 b0 a8 62 9f
00000160: 5f ba dc a7 f5 54 b8 4f 4f bb 1e a2 16 4b 4f 2d
00000170: d4 08 4e 45 c2 c0 60 3b 73 df 6b 35 3a fe 38 2e
00000180: 25 75 fc be 89 4c d2 7a 9c 1f b4 41 a6 31 d3 3d
00000190: 39 a6 d1 c4 47 94 44 30 3a 2b 23 22 ba c0 a9 df
000001A0: dc 1c 90 8d d1 e8 13 f9 08 68 5a 94 98 c7 3f 47
000001B0: 77 79 b5 bb fb 22 56 4b 38 55 48 e8 14 d4 01 eb
000001C0: 63 e9 17 da 24 69 9a 6d dc 1e 25 06 ef 77 10 46
000001D0: ad 99 ad 9c 54 4f d4 68 64 ea 05 1d ef 29 ea 0e
000001E0: 3c 1c 7e 27 cf 59 76 42 5b 02 04 b8
```

(156) Decrypts ciphertext and verifies ICV using K3r as K_msg, resulting in plaintext (fragment 1)

```
00000000: 25 00 00 4e 09 00 00 00 30 44 31 20 30 1e 06 03
00000010: 55 04 03 13 17 49 4b 45 20 49 6e 74 65 72 6f 70
00000020: 20 54 65 73 74 20 53 65 72 76 65 72 31 13 30 11
00000030: 06 03 55 04 0a 13 0a 45 4c 56 49 53 2d 50 4c 55
00000040: 53 31 0b 30 09 06 03 55 04 06 13 02 52 55 27 00
00000050: 04 bb 04 30 82 04 b2 30 82 04 5f a0 03 02 01 02
00000060: 02 13 7c 00 03 d9 02 ec f9 34 3e c8 aa d6 59 00
00000070: 01 00 03 d9 02 30 0a 06 08 2a 85 03 07 01 01 03
00000080: 02 30 82 01 0a 31 18 30 16 06 05 2a 85 03 64 01
00000090: 12 0d 31 32 33 34 35 36 37 38 39 30 31 32 33 31
000000A0: 1a 30 18 06 08 2a 85 03 03 81 03 01 01 12 0c 30
000000B0: 30 31 32 33 34 35 36 37 38 39 30 31 2f 30 2d 06
000000C0: 03 55 04 09 0c 26 d1 83 d0 bb 2e 20 d0 a1 d1 83
000000D0: d1 89 d1 91 d0 b2 d1 81 d0 ba d0 b8 d0 b9 20 d0
000000E0: b2 d0 b0 d0 bb 20 d0 b4 2e 20 31 38 31 0b 30 09
000000F0: 06 03 55 04 06 13 02 52 55 31 19 30 17 06 03 55
00000100: 04 08 0c 10 d0 b3 2e 20 d0 9c d0 be d1 81 d0 ba
00000110: d0 b2 d0 b0 31 15 30 13 06 03 55 04 07 0c 0c d0
00000120: 9c d0 be d1 81 d0 ba d0 b2 d0 b0 31 25 30 23 06
00000130: 03 55 04 0a 0c 1c d0 9e d0 9e d0 9e 20 22 d0 9a
00000140: d0 a0 d0 98 d0 9f d0 a2 d0 9e 2d d0 9f d0 a0 d0
00000150: 9e 22 31 3b 30 39 06 03 55 04 03 0c 32 d0 a2 d0
00000160: b5 d1 81 d1 82 d0 be d0 b2 d1 8b d0 b9 20 d0 a3
00000170: d0 a6 20 d0 9e d0 9e d0 9e 20 22 d0 9a d0 a0 d0
00000180: 98 d0 9f d0 a2 d0 9e 2d d0 9f d0 a0 d0 9e 22 30
00000190: 1e 17 0d 32 31 30 39 33 30 31 33 32 34 30 36 5a
000001A0: 17 0d 32 31 31 32 33 30 31 33 33 34 30 36 5a 30
000001B0: 44 31 20 30 1e 06 03 55 04 03 13 17 49 4b 45 20
000001C0: 49 6e 74 65 72 6f 70 20 54 65 73 74 20 53 65 72
000001D0: 76 65 72 31 13 30 11 06 03 55 04 0a 13 0a 45 4c
000001E0: 56 49 53 2d 50 4c 55 53 31 0b 30 00
```

(157) Extracts IV from message (fragment 2)

```
00000000: 00 00 00 00 00 00 00 01
```

(158) Uses previously computed key K3r

```
00000000: 31 95 e8 c6 67 af 42 d8 ce f1 e8 99 c6 8b 2a c2
00000010: 29 aa 3d c0 ff 18 5f 3d 79 4a 14 6b 9f ac d0 bb
```

(159) Composes MGM nonce (fragment 2)

```
00000000: 00 00 00 01 a5 bb 18 2f
```

(160) Extracts ICV from message (fragment 2)

```
00000000: 89 bd 07 12 fc 3f 15 8d
```

(161) Extracts AAD from message (fragment 2)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 20 00 00 00 01 00 00 02 20 00 00 02 04
00000020: 00 02 00 04
```

(162) Extracts ciphertext from message (fragment 2)

```
00000000: b1 c8 8d ae d9 6f 91 7e 5a 6a 2d 8c e0 d6 28 3e
00000010: 10 59 46 12 a1 1e fa 53 c3 58 ec 4e a9 a5 92 0c
00000020: fa 5e cf a3 33 4a 8b b7 56 66 54 d9 9c 64 2e b6
00000030: 4d 03 3f 77 a8 17 88 f6 23 e0 2e 56 a6 a2 4c 4d
00000040: 6e e3 09 8a 2e 31 a1 85 1c cf ce 95 e7 73 93 8e
00000050: 9c 5a 7b 3b 49 75 96 69 d4 b0 46 f7 74 b0 0d 5d
00000060: 91 3b 6d 2b a4 46 cc 5c d9 a8 38 c0 6b ad 73 35
00000070: 09 aa c7 4c 91 8a 84 1c dd 3f e1 44 f7 c5 9c 61
00000080: 0e b7 03 6b 84 cc 8e 93 5b d5 f6 7e 71 3a f4 2c
00000090: 98 14 ad 47 e3 c3 70 dc e3 3e c0 a5 e0 e4 6d 01
000000A0: 44 78 7f e3 b7 6c cb 44 29 59 96 e9 84 6d 9d 18
000000B0: 89 66 16 07 46 a4 cd 72 a6 0e bd d2 a7 1c f7 21
000000C0: f0 d1 67 a9 0d 1c c4 c8 30 bd 26 1f 53 7d 61 8b
000000D0: ad 6f ef 3e 2c 6e 7e 69 b9 92 72 66 65 b6 06 22
000000E0: 49 a1 a8 f1 2f 02 dd 41 bf f5 d1 f6 7c 93 25 6e
000000F0: 52 8b a9 3f b5 40 97 02 bb 7c f5 33 a6 60 52 b8
00000100: 4f 3e 80 6c 38 cf e4 8b 15 fd d0 66 75 c1 bf bb
00000110: ac fc ac 01 c3 11 8e 0b 3e e9 2c 1b 5d b9 9f f6
00000120: 2f d7 e8 3c c7 a9 25 8b aa 6e c6 49 6d 6f df 42
00000130: 53 0e ba 70 54 d2 af c3 4d 02 e1 48 42 c5 45 53
00000140: 25 59 66 25 c7 3c c6 c2 e2 99 e2 bb 47 a4 a7 be
00000150: 6c 92 0d 3b 4c ab 6e d7 23 05 ea 73 07 62 e8 c0
00000160: e8 78 47 af 54 c8 67 8f dd 32 59 8d 87 ac 42 0e
00000170: 21 15 c4 f7 66 dc 02 cf 55 c2 e3 4d 8e 91 7a fd
00000180: d7 4d 20 b0 6f 67 78 58 08 9c ba 05 8b b0 9c 16
00000190: 20 51 75 12 96 e2 d5 28 ac 3e 50 26 04 6f 59 02
000001A0: 28 e0 ec 2c da 70 4a 9c 15 5a 2e 52 01 e6 4e 1e
000001B0: 10 6d 8d 5d 2a 81 69 0e 54 d0 5e 13 82 82 84 9a
000001C0: ac a6 0e 69 4e 17 5c c1 8a 71 f8 b4 80 3b 7a e5
000001D0: b8 1f 09 4a 02 14 24 07 af 6a 14 d9 52 8e da d3
000001E0: 58 23 68 71 27 b2 9a 03 09 f7 80 51
```

(163) Decrypts ciphertext and verifies ICV using K3r as K_msg, resulting in plaintext (fragment 2)

```
00000000: 09 06 03 55 04 06 13 02 52 55 30 66 30 1f 06 08
00000010: 2a 85 03 07 01 01 01 01 30 13 06 07 2a 85 03 02
00000020: 02 24 00 06 08 2a 85 03 07 01 01 02 02 03 43 00
00000030: 04 40 5b b3 14 3e f4 70 c1 70 d7 f3 27 25 d8 53
00000040: 7c e6 de 6d 8c 29 f6 b2 32 64 56 dc b1 77 f2 3d
00000050: fa f4 2a 5c f3 74 86 7f 04 72 51 c1 cf b3 43 36
00000060: f5 95 a2 af 05 47 57 1a 55 c0 78 a4 9d 64 26 b8
00000070: 61 14 a3 82 02 59 30 82 02 55 30 0e 06 03 55 1d
00000080: 0f 01 01 ff 04 04 03 02 05 a0 30 13 06 03 55 1d
00000090: 25 04 0c 30 0a 06 08 2b 06 01 05 05 07 03 11 30
000000A0: 1d 06 03 55 1d 0e 04 16 04 14 e0 d3 f0 09 ad ce
000000B0: 6c a5 47 ba 9b f7 a6 a5 1b 06 14 ba a5 43 30 1f
000000C0: 06 03 55 1d 23 04 18 30 16 80 14 9b 85 5e fb 81
000000D0: dc 4d 59 07 51 63 cf be df da 2c 7f c9 44 3c 30
000000E0: 82 01 0f 06 03 55 1d 1f 04 82 01 06 30 82 01 02
000000F0: 30 81 ff a0 81 fc a0 81 f9 86 81 b5 68 74 74 70
00000100: 3a 2f 2f 74 65 73 74 67 6f 73 74 32 30 31 32 2e
00000110: 63 72 79 70 74 6f 70 72 6f 2e 72 75 2f 43 65 72
00000120: 74 45 6e 72 6f 6c 6c 2f 21 30 34 32 32 21 30 34
00000130: 33 35 21 30 34 34 31 21 30 34 34 32 21 30 34 33
00000140: 65 21 30 34 33 32 21 30 34 34 62 21 30 34 33 39
00000150: 25 32 30 21 30 34 32 33 21 30 34 32 36 25 32 30
00000160: 21 30 34 31 65 21 30 34 31 65 21 30 34 31 65 25
00000170: 32 30 21 30 30 32 32 21 30 34 31 61 21 30 34 32
00000180: 30 21 30 34 31 38 21 30 34 31 66 21 30 34 32 32
00000190: 21 30 34 31 65 2d 21 30 34 31 66 21 30 34 32 30
000001A0: 21 30 34 31 65 21 30 30 32 32 28 31 29 2e 63 72
000001B0: 6c 86 3f 68 74 74 70 3a 2f 2f 74 65 73 74 67 6f
000001C0: 73 74 32 30 31 32 2e 63 72 79 70 74 6f 70 72 6f
000001D0: 2e 72 75 2f 43 65 72 74 45 6e 72 6f 6c 6c 2f 74
000001E0: 65 73 74 67 6f 73 74 32 30 31 32 00
```

(164) Extracts IV from message (fragment 3)

```
00000000: 00 00 00 00 00 00 00 02
```

(165) Uses previously computed key K3r

```
00000000: 31 95 e8 c6 67 af 42 d8 ce f1 e8 99 c6 8b 2a c2
00000010: 29 aa 3d c0 ff 18 5f 3d 79 4a 14 6b 9f ac d0 bb
```

(166) Composes MGM nonce (fragment 3)

```
00000000: 00 00 00 02 a5 bb 18 2f
```

(167) Extracts ICV from message (fragment 3)

```
00000000: 7d 7c 57 8f 91 d0 c9 eb
```

(168) Extracts AAD from message (fragment 3)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 20 00 00 00 01 00 00 02 20 00 00 02 04
00000020: 00 03 00 04
```

(169) Extracts ciphertext from message (fragment 3)

```
00000000: 08 e0 86 04 1f 8a c9 b5 68 cd 96 10 ab 59 99 3a
00000010: 54 7b a9 fa d7 60 46 ec c3 bf bd 8f fa 03 ed 41
00000020: 49 13 ca 8c 9c b8 0c df 81 25 e2 30 ca cb 65 b9
00000030: 16 55 8e 67 f4 b3 7c b8 91 66 76 7c a4 15 98 a3
00000040: 3a c9 48 64 e4 ce 9f 64 67 5d bb 7c 03 23 9e c9
00000050: 81 3f da 48 ee a6 2a d8 fb ac 77 ce ed c2 a4 d9
00000060: 24 d3 71 99 fc 71 2b 6c 10 d3 c3 4b b5 37 e2 55
00000070: 5f d5 ee c0 d6 ff 66 15 8c e5 63 26 96 cd 3f 49
00000080: 2b da 51 94 55 6e 2e e5 2e d1 b4 91 81 50 85 8a
00000090: 84 bd fe 52 ec ce 1b 6b bd 7d 12 b4 de a5 88 c4
000000A0: b7 78 d3 3d 2d 46 ef dc 0f 91 43 be 08 7a ba fa
000000B0: b3 2a c2 17 30 99 79 ae 3a 00 f0 3f 47 4a 9b 11
000000C0: 4d 7b 1b 28 0a 44 5b 1a af 35 4d c3 2b 6b be 11
000000D0: 89 03 b9 de cf 37 57 53 1e a4 f3 3f ce 52 a6 d8
000000E0: 7e 9d d8 d4 2f 9f f5 8f 3c c6 cb 2f 56 e0 97 2d
000000F0: b2 0e 10 66 3b 3c ec 34 50 99 a3 7d 42 ec 96 eb
00000100: 87 48 72 2c 0a 6d af b9 4b 62 48 89 36 01 21 ab
00000110: 8e 79 10 54 9c 83 ab a9 8a 6c 37 c7 ac dc a1 7e
00000120: 41 0e 58 de da aa 95 71 fb 34 50 8a ef 37 0b c4
00000130: 56 ca 4b 2c 75 b7 c7 d9 74 22 c2 65 1a e4 4f 94
00000140: 20 f6 e9 44 f1 69 5e d2 18 d3 30 2e 85 74 25 be
00000150: 2a 88 e2 ce fe 75 ca fa 25 f9 2e 88 8c ed 6f dd
00000160: c3 c5 53 2e da 14 fd 96 28 4a b7 81 3a b3 d5 44
00000170: 26 e2 84 21 f2 5c 0a ed bf c4 34 1c a4 91 5e f3
00000180: 47 ef 0e 9e fb ee 34 95 5d 21 72 43 c9 63 af b4
00000190: f2 98 4a 36 57 77 fc e7 57 52 b2 4d bf 34 2a 98
000001A0: ea 70 cd d7 a9 da 4c 0d 19 05 d4 1e dd 36 c7 c4
000001B0: 31 54 18 2a ef 0e 30 44 97 31 15 57 cd d4 88 52
000001C0: 4e 42 c8 20 89 8d 35 7b 8e 03 96 b4 74 fb ec 3b
000001D0: 14 c2 64 49 92 f2 1f 3d ff 84 2d 92 4c b9 01 04
000001E0: 3d 0a 2a 28 33 de 43 44 6b cf 79 0e
```

(170) Decrypts ciphertext and verifies ICV using K3r as K_msg, resulting in plaintext (fragment 3)

```
00000000: 28 31 29 2e 63 72 6c 30 81 da 06 08 2b 06 01 05
00000010: 05 07 01 01 04 81 cd 30 81 ca 30 44 06 08 2b 06
00000020: 01 05 05 07 30 02 86 38 68 74 74 70 3a 2f 2f 74
00000030: 65 73 74 67 6f 73 74 32 30 31 32 2e 63 72 79 70
00000040: 74 6f 70 72 6f 2e 72 75 2f 43 65 72 74 45 6e 72
00000050: 6f 6c 6c 2f 72 6f 6f 74 32 30 31 38 2e 63 72 74
00000060: 30 3f 06 08 2b 06 01 05 05 07 30 01 86 33 68 74
00000070: 74 70 3a 2f 2f 74 65 73 74 67 6f 73 74 32 30 31
00000080: 32 2e 63 72 79 70 74 6f 70 72 6f 2e 72 75 2f 6f
00000090: 63 73 70 32 30 31 32 67 2f 6f 63 73 70 2e 73 72
000000A0: 66 30 41 06 08 2b 06 01 05 05 07 30 01 86 35 68
000000B0: 74 74 70 3a 2f 2f 74 65 73 74 67 6f 73 74 32 30
000000C0: 31 32 2e 63 72 79 70 74 6f 70 72 6f 2e 72 75 2f
000000D0: 6f 63 73 70 32 30 31 32 67 73 74 2f 6f 63 73 70
000000E0: 2e 73 72 66 30 0a 06 08 2a 85 03 07 01 01 03 02
000000F0: 03 41 00 a5 39 5f ca 48 e1 c2 93 c1 e0 8a 64 74
00000100: 0f 6b 86 a2 15 9b 46 29 d0 42 71 4f ce e7 52 d7
00000110: d7 3d aa 47 ce cf 52 63 8f 26 b2 17 5f ad 96 57
00000120: 76 ea 5f d0 87 bb 12 29 e4 06 0e e1 5f fd 59 81
00000130: fb 34 6d 29 00 00 55 0e 00 00 00 0c 30 0a 06 08
00000140: 2a 85 03 07 01 01 03 02 c8 40 af f7 46 6f 7b eb
00000150: d2 b9 1c 5a 80 d0 00 93 c2 5e 44 16 40 47 f7 8e
00000160: 61 9c da a5 16 94 83 c5 68 5f e8 4d 03 e7 c2 cd
00000170: 08 07 b8 f3 46 66 6d 05 76 c0 d5 e7 60 1d 59 49
00000180: 09 45 52 c4 95 a7 5a d3 29 00 00 08 00 00 40 00
00000190: 2f 00 00 0c 00 00 40 01 00 00 00 40 21 00 00 10
000001A0: 02 00 00 00 00 01 00 04 0a 01 01 03 2c 00 00 20
000001B0: 00 00 00 1c 01 03 04 02 34 ff 8a 25 03 00 00 08
000001C0: 01 00 00 21 00 00 00 08 05 00 00 00 2d 00 00 18
000001D0: 01 00 00 00 07 00 00 10 00 00 ff ff 0a 01 01 03
000001E0: 0a 01 01 03 29 00 00 18 01 00 00 00
```

(171) Extracts IV from message (fragment 4)

```
00000000: 00 00 00 00 00 00 00 03
```

(172) Uses previously computed key K3r

```
00000000: 31 95 e8 c6 67 af 42 d8 ce f1 e8 99 c6 8b 2a c2
00000010: 29 aa 3d c0 ff 18 5f 3d 79 4a 14 6b 9f ac d0 bb
```

(173) Composes MGM nonce (fragment 4)

```
00000000: 00 00 00 03 a5 bb 18 2f
```

(174) Extracts ICV from message (fragment 4)

```
00000000: 6c 27 70 e0 8a 82 bd 4b
```

(175) Extracts AAD from message (fragment 4)

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 35 20 23 20 00 00 00 01 00 00 00 5e 00 00 00 42
00000020: 00 04 00 04
```

(176) Extracts ciphertext from message (fragment 4)

```
00000000: 81 fa 5d 7a 67 13 b7 93 f4 2c 01 b8 d1 02 8c ab
00000010: 8e 80 47 25 6e c5 69 e3 0c 84 cd 35 9a 0f 7a cc
00000020: 0a 92 7a 74 77 dc ba 60 ac 4a
```

(177) Decrypts ciphertext and verifies ICV using K3r as K_msg, resulting in plaintext (fragment 4)

```
00000000: 00 07 00 00 10 00 00 ff ff 0a 00 00 00 0a 00 00
00000010: ff 29 00 00 08 00 00 40 02 29 00 00 08 00 00 40
00000020: 0a 00 00 00 08 00 00 40 0b 00
```

(178) Reassembles message from received fragments and parses it

```
IKE SA Auth
#9280E0822E758778.DB578D97DE119D1E.00000001 IKEv2 R=>I[1563]
  4*EF[...]->E[1535]{
    IDr[78](DN){CN=IKE Interop Test Server,O=ELVIS-PLUS,C=RU},
    CERT[1211](X.509 Cert){308204...FB346D},
    AUTH[85](Sig){id-tc26-signwithdigest-gost3410-12-256[12]:
              C840AF...A75AD3},
    N[8](INITIAL_CONTACT),
    N[12](SET_WINDOW_SIZE){64},
    CP[16](REPLY){IP4.Address[4]=10.1.1.3},
    SA[32]{
      P[28](#1:ESP:34FF8A25:2#){
        Encryption=ENCR_MAGMA_MGM_KTREE,
        ESN=Off}},
    TSi[24](1#){10.1.1.3},
    TSr[24](1#){10.0.0.0-10.0.0.255},
    N[8](ADDITIONAL_TS_POSSIBLE),
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(179) Computes prf(SK_pr, IDr)

```
00000000: 7d c8 6a 33 12 02 5c 21 1f ab dc 83 0b 01 a5 27
00000010: 82 a2 f2 1f 64 c6 e9 5e 0e c0 4c e5 d9 11 8d 8e
00000020: b9 5c ef fa b0 a3 37 75 94 20 7c e4 60 60 ed 9d
00000030: fa 5e cb 7e e7 79 05 ab fb 51 1b 03 a8 2c c5 6a
```

(180) Uses responder's public key

```
00000000: 5B B3 14 3E F4 70 C1 70 D7 F3 27 25 D8 53 7C E6
00000010: DE 6D 8C 29 F6 B2 32 64 56 DC B1 77 F2 3D FA F4
00000020: 2A 5C F3 74 86 7F 04 72 51 C1 CF B3 43 36 F5 95
00000030: A2 AF 05 47 57 1A 55 C0 78 A4 9D 64 26 B8 61 14
```

(181) Verifies signature from AUTH payload using algorithm id-tc26-signwithdigest-gost3410-12-256

```
00000000: c8 40 af f7 46 6f 7b eb d2 b9 1c 5a 80 d0 00 93
00000010: c2 5e 44 16 40 47 f7 8e 61 9c da a5 16 94 83 c5
00000020: 68 5f e8 4d 03 e7 c2 cd 08 07 b8 f3 46 66 6d 05
00000030: 76 c0 d5 e7 60 1d 59 49 09 45 52 c4 95 a7 5a d3
```

(182) Computes keys for ESP SAs

```
00000000: 98 ab 7e db 78 03 a1 e6 c7 21 43 ee b9 7f 5f 56
00000010: 45 bb 51 cd 0b b7 09 a1 af 34 02 87 69 4d 7b a0
00000020: 1d 14 a0 cc
00000000: 70 31 4d 57 94 8b 7e 5c 6f 29 d5 68 1b fd 43 2b
00000010: 19 4e 64 6d 8f 8a 8d 1e ba 72 24 59 c7 0c de 81
00000020: e2 04 84 af
```

A.2.2. Sub-Scenario 2: IKE SA Rekeying Using the CREATE_CHILD_SA Exchange

```
Initiator                                 Responder

HDR, SK {SAi, Ni, KEi [,N+]}  --->
                              <---    HDR, SK {SAr, Nr, KEr [,N+]}
```

Initiator's actions:

(1) Generates random SPIi for new IKE SA

```
00000000: fd d9 35 89 50 d5 db 22
```

(2) Generates random IKE nonce Ni

```
00000000: 2e 98 99 76 4a 67 1e d9 17 27 32 f2 6d 3a 93 3c
00000010: 7f 21 2b 0e 59 90 cf 2a 7f 85 53 c5 ed 8a ec 37
```

(3) Generates ephemeral private key

```
00000000: 29 2c 72 52 e0 6c fd 39 1d 55 04 e9 cf af 82 29
00000010: 89 09 ff 1c ab b2 dd a5 88 f0 34 fd 2c 57 d2 28
```

(4) Computes public key

```
00000000: 13 78 88 b1 0f 09 65 43 94 53 b7 26 5d 2a 8b 29
00000010: 5f a9 d6 73 a2 d0 64 6c 98 0f 02 44 d5 5a 1d 13
00000020: 7b b4 4d 18 81 c3 ee 48 35 18 a7 71 ce 4f fa 45
00000030: b0 e9 74 63 37 58 32 7c ff a5 e4 98 b5 02 d4 ef
```

(5) Creates message

```
Create Child SA
#9280E0822E758778.DB578D97DE119D1E.00000002 IKEv2 R<-I[213]
  E[185]{
    SA[44]{
      P[40](#1:IKE:FDD9358950D5DB22:3#){
        Encryption=ENCR_MAGMA_MGM_KTREE,
        PRF=PRF_HMAC_STREEBOG_512,
        KE=GOST3410_2012_256}},
    NONCE[36]{2E9899...8AEC37},
    KE[72](GOST3410_2012_256){137888...02D4EF},
    N[12](SET_WINDOW_SIZE){4}}
```

(6) Computes K3i (i3 = 1)

```
00000000: da 26 f7 b5 4c 4c 97 23 3f e2 cb 53 23 82 1b 2a
00000010: 40 3c 95 e1 78 2a 8f 3d 1b 0f a4 d3 ab c3 98 3d
```

(7) Composes MGM nonce

```
00000000: 00 00 00 00 b4 e1 3e 23
```

(8) Composes AAD

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 2e 20 24 08 00 00 00 02 00 00 00 d5 21 00 00 b9
```

(9) Composes plaintext

```
00000000: 28 00 00 2c 00 00 00 28 01 01 08 03 fd d9 35 89
00000010: 50 d5 db 22 03 00 00 08 01 00 00 21 03 00 00 08
00000020: 02 00 00 09 00 00 00 08 04 00 00 21 22 00 00 24
00000030: 2e 98 99 76 4a 67 1e d9 17 27 32 f2 6d 3a 93 3c
00000040: 7f 21 2b 0e 59 90 cf 2a 7f 85 53 c5 ed 8a ec 37
00000050: 29 00 00 48 00 21 00 00 13 78 88 b1 0f 09 65 43
00000060: 94 53 b7 26 5d 2a 8b 29 5f a9 d6 73 a2 d0 64 6c
00000070: 98 0f 02 44 d5 5a 1d 13 7b b4 4d 18 81 c3 ee 48
00000080: 35 18 a7 71 ce 4f fa 45 b0 e9 74 63 37 58 32 7c
00000090: ff a5 e4 98 b5 02 d4 ef 00 00 00 0c 00 00 40 01
000000A0: 00 00 00 04 00
```

(10) Encrypts plaintext using K3i as K_msg, resulting in ciphertext

```
00000000: f4 d1 2b 1e 51 65 d1 0b 7f 38 c6 16 3f 6e 5e f7
00000010: e0 48 24 15 6a 45 50 51 1a 6e fb 1c 1d b8 52 75
00000020: 80 56 e4 da fb e5 fe 42 08 71 79 99 ef 17 7a 03
00000030: fc c3 c6 b0 15 a5 72 a4 1b de e2 b5 e6 46 56 73
00000040: 3f 78 57 9e 6b b4 05 4c 86 91 c3 61 00 2d 9b 89
00000050: c0 0c 8b 11 0b 41 e7 92 16 7f f8 f6 5d ef f4 29
00000060: 27 ef ba 8c 5f 30 fd a9 12 4c 5f 8d e9 39 97 48
00000070: 9a e1 6a 91 01 c7 8c 94 aa 3b 89 bb 54 40 3b f1
00000080: 8d 2b 0e 75 d8 f6 98 d2 74 e4 b7 2f f5 ac a0 41
00000090: df 73 7f 1c 37 18 b9 79 8e 9d 6f ea e5 8a b6 9f
000000A0: 35 d9 d4 b3 cd
```

(11) Computes ICV using K3i as K_msg

```
00000000: 49 96 ac 4c 3f c4 fc 1d
```

(12) Composes IV

```
00000000: 00 00 00 00 01 00 00 00
```

(13) Sends message, peer receives message

```
     10.111.10.171:54295->10.111.15.45:4500 [217]

00000000: 00 00 00 00 92 80 e0 82 2e 75 87 78 db 57 8d 97
00000010: de 11 9d 1e 2e 20 24 08 00 00 00 02 00 00 00 d5
00000020: 21 00 00 b9 00 00 00 00 01 00 00 00 f4 d1 2b 1e
00000030: 51 65 d1 0b 7f 38 c6 16 3f 6e 5e f7 e0 48 24 15
00000040: 6a 45 50 51 1a 6e fb 1c 1d b8 52 75 80 56 e4 da
00000050: fb e5 fe 42 08 71 79 99 ef 17 7a 03 fc c3 c6 b0
00000060: 15 a5 72 a4 1b de e2 b5 e6 46 56 73 3f 78 57 9e
00000070: 6b b4 05 4c 86 91 c3 61 00 2d 9b 89 c0 0c 8b 11
00000080: 0b 41 e7 92 16 7f f8 f6 5d ef f4 29 27 ef ba 8c
00000090: 5f 30 fd a9 12 4c 5f 8d e9 39 97 48 9a e1 6a 91
000000A0: 01 c7 8c 94 aa 3b 89 bb 54 40 3b f1 8d 2b 0e 75
000000B0: d8 f6 98 d2 74 e4 b7 2f f5 ac a0 41 df 73 7f 1c
000000C0: 37 18 b9 79 8e 9d 6f ea e5 8a b6 9f 35 d9 d4 b3
000000D0: cd 49 96 ac 4c 3f c4 fc 1d
```

Responder's actions:

(14) Extracts IV from message

```
00000000: 00 00 00 00 01 00 00 00
```

(15) Computes K3i (I = 1)

```
00000000: da 26 f7 b5 4c 4c 97 23 3f e2 cb 53 23 82 1b 2a
00000010: 40 3c 95 e1 78 2a 8f 3d 1b 0f a4 d3 ab c3 98 3d
```

(16) Composes MGM nonce

```
00000000: 00 00 00 00 b4 e1 3e 23
```

(17) Extracts ICV from message

```
00000000: 49 96 ac 4c 3f c4 fc 1d
```

(18) Extracts AAD from message

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 2e 20 24 08 00 00 00 02 00 00 00 d5 21 00 00 b9
```

(19) Extracts ciphertext from message

```
00000000: f4 d1 2b 1e 51 65 d1 0b 7f 38 c6 16 3f 6e 5e f7
00000010: e0 48 24 15 6a 45 50 51 1a 6e fb 1c 1d b8 52 75
00000020: 80 56 e4 da fb e5 fe 42 08 71 79 99 ef 17 7a 03
00000030: fc c3 c6 b0 15 a5 72 a4 1b de e2 b5 e6 46 56 73
00000040: 3f 78 57 9e 6b b4 05 4c 86 91 c3 61 00 2d 9b 89
00000050: c0 0c 8b 11 0b 41 e7 92 16 7f f8 f6 5d ef f4 29
00000060: 27 ef ba 8c 5f 30 fd a9 12 4c 5f 8d e9 39 97 48
00000070: 9a e1 6a 91 01 c7 8c 94 aa 3b 89 bb 54 40 3b f1
00000080: 8d 2b 0e 75 d8 f6 98 d2 74 e4 b7 2f f5 ac a0 41
00000090: df 73 7f 1c 37 18 b9 79 8e 9d 6f ea e5 8a b6 9f
000000A0: 35 d9 d4 b3 cd
```

(20) Decrypts ciphertext and verifies ICV using K3i as K_msg, resulting in plaintext

```
00000000: 28 00 00 2c 00 00 00 28 01 01 08 03 fd d9 35 89
00000010: 50 d5 db 22 03 00 00 08 01 00 00 21 03 00 00 08
00000020: 02 00 00 09 00 00 00 08 04 00 00 21 22 00 00 24
00000030: 2e 98 99 76 4a 67 1e d9 17 27 32 f2 6d 3a 93 3c
00000040: 7f 21 2b 0e 59 90 cf 2a 7f 85 53 c5 ed 8a ec 37
00000050: 29 00 00 48 00 21 00 00 13 78 88 b1 0f 09 65 43
00000060: 94 53 b7 26 5d 2a 8b 29 5f a9 d6 73 a2 d0 64 6c
00000070: 98 0f 02 44 d5 5a 1d 13 7b b4 4d 18 81 c3 ee 48
00000080: 35 18 a7 71 ce 4f fa 45 b0 e9 74 63 37 58 32 7c
00000090: ff a5 e4 98 b5 02 d4 ef 00 00 00 0c 00 00 40 01
000000A0: 00 00 00 04 00
```

(21) Parses received message

```
Create Child SA
#9280E0822E758778.DB578D97DE119D1E.00000002 IKEv2 I->R[213]
  E[185]{
    SA[44]{
      P[40](#1:IKE:FDD9358950D5DB22:3#){
        Encryption=ENCR_MAGMA_MGM_KTREE,
        PRF=PRF_HMAC_STREEBOG_512,
        KE=GOST3410_2012_256}},
    NONCE[36]{2E9899...8AEC37},
    KE[72](GOST3410_2012_256){137888...02D4EF},
    N[12](SET_WINDOW_SIZE){4}}
```

(22) Generates random SPIr for new IKE SA

```
00000000: 81 27 5d a2 98 90 1a 06
```

(23) Generates random IKE nonce Nr

```
00000000: cf 8e 80 0f 84 c9 d8 50 06 a4 02 b5 19 2a 0f a0
00000010: d7 f4 db 70 ca f1 2b 9b 02 ce 92 8d 97 20 43 96
```

(24) Generates ephemeral private key

```
00000000: af 9a 62 7d d3 b8 23 d2 49 7f f9 0a 9d f2 55 8c
00000010: ae 9c 48 ad f5 a4 ee a5 f6 24 5f 48 3c f8 42 0d
```

(25) Computes public key

```
00000000: ba 9c bb 8d c4 51 68 1c 63 50 9c 5b 78 c2 93 be
00000010: 52 9b 7a a0 6b 14 1e 0f 52 d4 a3 0e 71 d7 5b 4c
00000020: aa 58 af 26 21 d9 b2 92 87 1c d9 7a 89 6f c2 7d
00000030: 7d 95 96 39 a2 36 37 8f f4 b9 1d 2f a8 b7 f5 c9
```

(26) Computes shared key

```
00000000: ae 27 a3 df af 7d bb ad f4 5c 19 64 c9 27 eb 41
00000010: 14 fc 1a f8 25 cc 93 50 a2 64 5f 04 67 0a 74 cb
```

(27) Computes SKEYSEED for new SA

```
00000000: 31 2b 7f 6a 24 23 8f ed b6 ac 40 a7 58 2e 28 54
00000010: 47 53 76 20 05 c7 00 c8 87 c1 51 68 93 40 7e 2d
00000020: ed 14 c4 78 9a f4 12 e7 f0 19 4d 4d 12 45 0d 42
00000030: e4 b2 29 e5 57 b4 90 cc cf d5 94 84 b4 59 5e b9
```

(28) Computes SK_d for new SA

```
00000000: 38 ec b5 1c 33 77 f8 62 29 9f 00 d9 98 5f a4 4c
00000010: ea c7 97 31 01 b9 39 ce 16 2c 1c 30 dd 53 d8 97
00000020: 48 49 cd ca 82 7b 57 55 e4 5a 33 1c 80 e6 b9 1f
00000030: 2c 80 b2 e5 48 8a 23 9d 8e 42 32 ed 4f 63 3a f1
```

(29) Computes SK_ei for new SA

```
00000000: 17 1c 7c 08 bd 1a 3d 50 58 e1 13 58 9d c4 21 c6
00000010: a3 44 e5 c1 f5 14 e8 22 ed 94 03 2e 76 47 b1 8d
00000020: 2b 3d 3b 2f
```

(30) Computes SK_er for new SA

```
00000000: 4a a9 b7 36 1d 2c e1 e0 dc 55 b6 45 0a 38 f1 9a
00000010: 83 cb 8f 79 57 5e df d8 5f 5e 22 a8 36 bd 3a 4a
00000020: d2 f6 27 21
```

(31) Creates message

```
Create Child SA
#9280E0822E758778.DB578D97DE119D1E.00000002 IKEv2 I<=R[213]
  E[185]{
    SA[44]{
      P[40](#1:IKE:81275DA298901A06:3#){
        Encryption=ENCR_MAGMA_MGM_KTREE,
        PRF=PRF_HMAC_STREEBOG_512,
        KE=GOST3410_2012_256}},
    NONCE[36]{CF8E80...204396},
    KE[72](GOST3410_2012_256){BA9CBB...B7F5C9},
    N[12](SET_WINDOW_SIZE){64}}
```

(32) Computes K3r (i3 = 1)

```
00000000: 9b 6c de 40 b4 63 c4 85 db 09 b7 24 f4 60 fa d0
00000010: 1f d3 f3 fa e9 f8 e9 03 0c 34 cb 51 52 51 5b 56
```

(33) Composes MGM nonce

```
00000000: 00 00 00 00 a5 bb 18 2f
```

(34) Composes AAD

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 2e 20 24 20 00 00 00 02 00 00 00 d5 21 00 00 b9
```

(35) Composes plaintext

```
00000000: 28 00 00 2c 00 00 00 28 01 01 08 03 81 27 5d a2
00000010: 98 90 1a 06 03 00 00 08 01 00 00 21 03 00 00 08
00000020: 02 00 00 09 00 00 00 08 04 00 00 21 22 00 00 24
00000030: cf 8e 80 0f 84 c9 d8 50 06 a4 02 b5 19 2a 0f a0
00000040: d7 f4 db 70 ca f1 2b 9b 02 ce 92 8d 97 20 43 96
00000050: 29 00 00 48 00 21 00 00 ba 9c bb 8d c4 51 68 1c
00000060: 63 50 9c 5b 78 c2 93 be 52 9b 7a a0 6b 14 1e 0f
00000070: 52 d4 a3 0e 71 d7 5b 4c aa 58 af 26 21 d9 b2 92
00000080: 87 1c d9 7a 89 6f c2 7d 7d 95 96 39 a2 36 37 8f
00000090: f4 b9 1d 2f a8 b7 f5 c9 00 00 00 0c 00 00 40 01
000000A0: 00 00 00 40 00
```

(36) Encrypts plaintext using K3r as K_msg, resulting in ciphertext

```
00000000: 6e a0 bc 5e 58 16 91 db 1f e0 22 20 b6 75 fd e6
00000010: e0 01 a7 86 0c 9c a6 77 ef cd f6 be e4 c8 31 18
00000020: c7 7f 68 58 d8 85 75 6c 1d 4a 0e 66 09 86 7c 84
00000030: 30 a7 2e f0 26 2b 19 da c5 25 34 5b 19 f0 97 86
00000040: 54 ca 08 92 65 9c e3 92 4d ee 92 0a a0 86 d7 3f
00000050: 4d d9 f2 7e 32 48 b3 9f ea 54 d2 96 99 42 30 6b
00000060: b0 b4 fe 5d 4a fc 8c ff 54 f6 2f b7 ca 7b 83 01
00000070: 36 85 57 78 b3 74 84 72 9d 94 2f 6f ae 4e 26 bb
00000080: 6e 06 84 2b ac f8 99 29 31 ad 7b dc db c0 0f 19
00000090: 5f 06 42 2d 90 d2 6a 05 8a 41 ee 24 e2 49 a5 b6
000000A0: 61 e8 cb 46 3c
```

(37) Computes ICV using K3r as K_msg

```
00000000: dc c4 ca 6d 07 cf 31 a8
```

(38) Composes IV

```
00000000: 00 00 00 00 01 00 00 00
```

(39) Sends message, peer receives message

```
       10.111.10.171:54295<-10.111.15.45:4500 [217]

00000000: 00 00 00 00 92 80 e0 82 2e 75 87 78 db 57 8d 97
00000010: de 11 9d 1e 2e 20 24 20 00 00 00 02 00 00 00 d5
00000020: 21 00 00 b9 00 00 00 00 01 00 00 00 6e a0 bc 5e
00000030: 58 16 91 db 1f e0 22 20 b6 75 fd e6 e0 01 a7 86
00000040: 0c 9c a6 77 ef cd f6 be e4 c8 31 18 c7 7f 68 58
00000050: d8 85 75 6c 1d 4a 0e 66 09 86 7c 84 30 a7 2e f0
00000060: 26 2b 19 da c5 25 34 5b 19 f0 97 86 54 ca 08 92
00000070: 65 9c e3 92 4d ee 92 0a a0 86 d7 3f 4d d9 f2 7e
00000080: 32 48 b3 9f ea 54 d2 96 99 42 30 6b b0 b4 fe 5d
00000090: 4a fc 8c ff 54 f6 2f b7 ca 7b 83 01 36 85 57 78
000000A0: b3 74 84 72 9d 94 2f 6f ae 4e 26 bb 6e 06 84 2b
000000B0: ac f8 99 29 31 ad 7b dc db c0 0f 19 5f 06 42 2d
000000C0: 90 d2 6a 05 8a 41 ee 24 e2 49 a5 b6 61 e8 cb 46
000000D0: 3c dc c4 ca 6d 07 cf 31 a8
```

Initiator's actions:

(40) Extracts IV from message

```
00000000: 00 00 00 00 01 00 00 00
```

(41) Computes K3r (i3 = 1)

```
00000000: 9b 6c de 40 b4 63 c4 85 db 09 b7 24 f4 60 fa d0
00000010: 1f d3 f3 fa e9 f8 e9 03 0c 34 cb 51 52 51 5b 56
```

(42) Composes MGM nonce

```
00000000: 00 00 00 00 a5 bb 18 2f
```

(43) Extracts ICV from message

```
00000000: dc c4 ca 6d 07 cf 31 a8
```

(44) Extracts AAD from message

```
00000000: 92 80 e0 82 2e 75 87 78 db 57 8d 97 de 11 9d 1e
00000010: 2e 20 24 20 00 00 00 02 00 00 00 d5 21 00 00 b9
```

(45) Extracts ciphertext from message

```
00000000: 6e a0 bc 5e 58 16 91 db 1f e0 22 20 b6 75 fd e6
00000010: e0 01 a7 86 0c 9c a6 77 ef cd f6 be e4 c8 31 18
00000020: c7 7f 68 58 d8 85 75 6c 1d 4a 0e 66 09 86 7c 84
00000030: 30 a7 2e f0 26 2b 19 da c5 25 34 5b 19 f0 97 86
00000040: 54 ca 08 92 65 9c e3 92 4d ee 92 0a a0 86 d7 3f
00000050: 4d d9 f2 7e 32 48 b3 9f ea 54 d2 96 99 42 30 6b
00000060: b0 b4 fe 5d 4a fc 8c ff 54 f6 2f b7 ca 7b 83 01
00000070: 36 85 57 78 b3 74 84 72 9d 94 2f 6f ae 4e 26 bb
00000080: 6e 06 84 2b ac f8 99 29 31 ad 7b dc db c0 0f 19
00000090: 5f 06 42 2d 90 d2 6a 05 8a 41 ee 24 e2 49 a5 b6
000000A0: 61 e8 cb 46 3c
```

(46) Decrypts ciphertext and verifies ICV using K3r as K_msg, resulting in plaintext

```
00000000: 28 00 00 2c 00 00 00 28 01 01 08 03 81 27 5d a2
00000010: 98 90 1a 06 03 00 00 08 01 00 00 21 03 00 00 08
00000020: 02 00 00 09 00 00 00 08 04 00 00 21 22 00 00 24
00000030: cf 8e 80 0f 84 c9 d8 50 06 a4 02 b5 19 2a 0f a0
00000040: d7 f4 db 70 ca f1 2b 9b 02 ce 92 8d 97 20 43 96
00000050: 29 00 00 48 00 21 00 00 ba 9c bb 8d c4 51 68 1c
00000060: 63 50 9c 5b 78 c2 93 be 52 9b 7a a0 6b 14 1e 0f
00000070: 52 d4 a3 0e 71 d7 5b 4c aa 58 af 26 21 d9 b2 92
00000080: 87 1c d9 7a 89 6f c2 7d 7d 95 96 39 a2 36 37 8f
00000090: f4 b9 1d 2f a8 b7 f5 c9 00 00 00 0c 00 00 40 01
000000A0: 00 00 00 40 00
```

(47) Parses received message

```
Create Child SA
#9280E0822E758778.DB578D97DE119D1E.00000002 IKEv2 R=>I[213]
  E[185]{
    SA[44]{
      P[40](#1:IKE:81275DA298901A06:3#){
        Encryption=ENCR_MAGMA_MGM_KTREE,
        PRF=PRF_HMAC_STREEBOG_512,
        KE=GOST3410_2012_256}},
    NONCE[36]{CF8E80...204396},
    KE[72](GOST3410_2012_256){BA9CBB...B7F5C9},
    N[12](SET_WINDOW_SIZE){64}}
```

(48) Computes shared key

```
00000000: ae 27 a3 df af 7d bb ad f4 5c 19 64 c9 27 eb 41
00000010: 14 fc 1a f8 25 cc 93 50 a2 64 5f 04 67 0a 74 cb
```

(49) Computes SKEYSEED for new SA

```
00000000: 31 2b 7f 6a 24 23 8f ed b6 ac 40 a7 58 2e 28 54
00000010: 47 53 76 20 05 c7 00 c8 87 c1 51 68 93 40 7e 2d
00000020: ed 14 c4 78 9a f4 12 e7 f0 19 4d 4d 12 45 0d 42
00000030: e4 b2 29 e5 57 b4 90 cc cf d5 94 84 b4 59 5e b9
```

(50) Computes SK_d for new SA

```
00000000: 38 ec b5 1c 33 77 f8 62 29 9f 00 d9 98 5f a4 4c
00000010: ea c7 97 31 01 b9 39 ce 16 2c 1c 30 dd 53 d8 97
00000020: 48 49 cd ca 82 7b 57 55 e4 5a 33 1c 80 e6 b9 1f
00000030: 2c 80 b2 e5 48 8a 23 9d 8e 42 32 ed 4f 63 3a f1
```

(51) Computes SK_ei for new SA

```
00000000: 17 1c 7c 08 bd 1a 3d 50 58 e1 13 58 9d c4 21 c6
00000010: a3 44 e5 c1 f5 14 e8 22 ed 94 03 2e 76 47 b1 8d
00000020: 2b 3d 3b 2f
```

(52) Computes SK_er for new SA

```
00000000: 4a a9 b7 36 1d 2c e1 e0 dc 55 b6 45 0a 38 f1 9a
00000010: 83 cb 8f 79 57 5e df d8 5f 5e 22 a8 36 bd 3a 4a
00000020: d2 f6 27 21
```

A.2.3. Sub-Scenario 3: ESP SAs Rekeying without PFS Using the CREATE_CHILD_SA Exchange

```
Initiator                                 Responder

HDR, SK {N(REKEY_SA), SAi, Ni,
     TSi, TSr [,N+]}          --->
                              <---    HDR, SK {SAr, Nr,
                                           TSi, TSr [,N+]}
```

Initiator's actions:

(1) Generates random IKE nonce Ni

```
00000000: b5 48 18 7d 30 d8 ea 49 20 d0 9d 42 de 9e 91 ce
00000010: b3 1c 41 85 37 66 d8 9e c6 a6 f8 08 93 f4 48 23
```

(2) Computes K1i (i1 = 0)

```
00000000: 28 b9 3c 93 ea db 74 38 64 87 8a 28 8d e0 38 5c
00000010: 14 cb ea 9f 67 58 a6 ee e2 2d c9 37 bb c8 41 69
```

(3) Computes K2i (i2 = 0)

```
00000000: 75 11 35 65 e6 29 70 2a d9 7d 38 a8 3a e3 aa 8a
00000010: 9e fb 80 af f5 52 71 be c9 c6 c3 4b 4b 40 96 44
```

(4) Computes K3i (i3 = 0)

```
00000000: 45 6f 03 f7 ad 75 eb e9 52 b8 8f 0d e8 36 47 69
00000010: 4d 2e f2 ba 15 e6 8c 89 1c 99 62 64 fb 0e 70 0a
```

(5) Selects SPI for new incoming ESP SA

```
00000000: 9a 8c 6a 9b
```

(6) Creates message

```
Create Child SA
#FDD9358950D5DB22.81275DA298901A06.00000000 IKEv2 R<-I[193]
  E[165]{
    N[12](ESP:6C0CA570:REKEY_SA),
    SA[32]{
      P[28](#1:ESP:9A8C6A9B:2#){
        Encryption=ENCR_MAGMA_MGM_KTREE,
        ESN=Off}},
    NONCE[36]{B54818...F44823},
    TSi[24](1#){10.1.1.3},
    TSr[24](1#){10.0.0.0-10.0.0.255},
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(7) Composes MGM nonce

```
00000000: 00 00 00 00 2b 3d 3b 2f
```

(8) Composes AAD

```
00000000: fd d9 35 89 50 d5 db 22 81 27 5d a2 98 90 1a 06
00000010: 2e 20 24 08 00 00 00 00 00 00 00 c1 29 00 00 a5
```

(9) Composes plaintext

```
00000000: 21 00 00 0c 03 04 40 09 6c 0c a5 70 28 00 00 20
00000010: 00 00 00 1c 01 03 04 02 9a 8c 6a 9b 03 00 00 08
00000020: 01 00 00 21 00 00 00 08 05 00 00 00 2c 00 00 24
00000030: b5 48 18 7d 30 d8 ea 49 20 d0 9d 42 de 9e 91 ce
00000040: b3 1c 41 85 37 66 d8 9e c6 a6 f8 08 93 f4 48 23
00000050: 2d 00 00 18 01 00 00 00 07 00 00 10 00 00 ff ff
00000060: 0a 01 01 03 0a 01 01 03 29 00 00 18 01 00 00 00
00000070: 07 00 00 10 00 00 ff ff 0a 00 00 00 0a 00 00 ff
00000080: 29 00 00 08 00 00 40 0a 00 00 00 08 00 00 40 0b
00000090: 00
```

(10) Encrypts plaintext using K3i as K_msg, resulting in ciphertext

```
00000000: 47 71 bb 57 2a 1a 58 a6 44 cb 60 d4 8e 5c cc 0a
00000010: b9 34 0f 34 80 cf a2 38 54 f6 70 3b 98 4e 8f 9f
00000020: 3b 5c 5a 04 06 dc e9 d4 d3 54 c6 4d 73 09 10 c5
00000030: 4e 26 c4 27 fd cb 54 e1 cf e0 fd b4 9f f8 00 41
00000040: 41 c8 58 b2 c9 3a d8 e0 19 40 a3 89 ee 26 d4 84
00000050: 69 e9 52 68 d5 e1 ee f0 89 6e d3 95 34 62 ad 2e
00000060: e6 77 17 b8 6c 25 52 7f d8 70 9c 36 0b c8 1d 1a
00000070: 43 50 82 2a be b6 31 ff 2f 43 11 f7 d0 60 bf 62
00000080: b9 08 c3 09 a3 78 fb 5e 76 57 91 5d 48 1c aa d2
00000090: a3
```

(11) Computes ICV using K3i as K_msg

```
00000000: b3 05 bd 43 2f 87 0c 3f
```

(12) Composes IV

```
00000000: 00 00 00 00 00 00 00 00
```

(13) Sends message, peer receives message

```
       10.111.10.171:54295->10.111.15.45:4500 [197]

00000000: 00 00 00 00 fd d9 35 89 50 d5 db 22 81 27 5d a2
00000010: 98 90 1a 06 2e 20 24 08 00 00 00 00 00 00 00 c1
00000020: 29 00 00 a5 00 00 00 00 00 00 00 00 47 71 bb 57
00000030: 2a 1a 58 a6 44 cb 60 d4 8e 5c cc 0a b9 34 0f 34
00000040: 80 cf a2 38 54 f6 70 3b 98 4e 8f 9f 3b 5c 5a 04
00000050: 06 dc e9 d4 d3 54 c6 4d 73 09 10 c5 4e 26 c4 27
00000060: fd cb 54 e1 cf e0 fd b4 9f f8 00 41 41 c8 58 b2
00000070: c9 3a d8 e0 19 40 a3 89 ee 26 d4 84 69 e9 52 68
00000080: d5 e1 ee f0 89 6e d3 95 34 62 ad 2e e6 77 17 b8
00000090: 6c 25 52 7f d8 70 9c 36 0b c8 1d 1a 43 50 82 2a
000000A0: be b6 31 ff 2f 43 11 f7 d0 60 bf 62 b9 08 c3 09
000000B0: a3 78 fb 5e 76 57 91 5d 48 1c aa d2 a3 b3 05 bd
000000C0: 43 2f 87 0c 3f
```

Responder's actions:

(14) Extracts IV from message

```
00000000: 00 00 00 00 00 00 00 00
```

(15) Computes K1i (i1 = 0)

```
00000000: 28 b9 3c 93 ea db 74 38 64 87 8a 28 8d e0 38 5c
00000010: 14 cb ea 9f 67 58 a6 ee e2 2d c9 37 bb c8 41 69
```

(16) Computes K2i (i2 = 0)

```
00000000: 75 11 35 65 e6 29 70 2a d9 7d 38 a8 3a e3 aa 8a
00000010: 9e fb 80 af f5 52 71 be c9 c6 c3 4b 4b 40 96 44
```

(17) Computes K3i (i3 = 0)

```
00000000: 45 6f 03 f7 ad 75 eb e9 52 b8 8f 0d e8 36 47 69
00000010: 4d 2e f2 ba 15 e6 8c 89 1c 99 62 64 fb 0e 70 0a
```

(18) Composes MGM nonce

```
00000000: 00 00 00 00 2b 3d 3b 2f
```

(19) Extracts ICV from message

```
00000000: b3 05 bd 43 2f 87 0c 3f
```

(20) Extracts AAD from message

```
00000000: fd d9 35 89 50 d5 db 22 81 27 5d a2 98 90 1a 06
00000010: 2e 20 24 08 00 00 00 00 00 00 00 c1 29 00 00 a5
```

(21) Extracts ciphertext from message

```
00000000: 47 71 bb 57 2a 1a 58 a6 44 cb 60 d4 8e 5c cc 0a
00000010: b9 34 0f 34 80 cf a2 38 54 f6 70 3b 98 4e 8f 9f
00000020: 3b 5c 5a 04 06 dc e9 d4 d3 54 c6 4d 73 09 10 c5
00000030: 4e 26 c4 27 fd cb 54 e1 cf e0 fd b4 9f f8 00 41
00000040: 41 c8 58 b2 c9 3a d8 e0 19 40 a3 89 ee 26 d4 84
00000050: 69 e9 52 68 d5 e1 ee f0 89 6e d3 95 34 62 ad 2e
00000060: e6 77 17 b8 6c 25 52 7f d8 70 9c 36 0b c8 1d 1a
00000070: 43 50 82 2a be b6 31 ff 2f 43 11 f7 d0 60 bf 62
00000080: b9 08 c3 09 a3 78 fb 5e 76 57 91 5d 48 1c aa d2
00000090: a3
```

(22) Decrypts ciphertext and verifies ICV using K3i as K_msg, resulting in plaintext

```
00000000: 21 00 00 0c 03 04 40 09 6c 0c a5 70 28 00 00 20
00000010: 00 00 00 1c 01 03 04 02 9a 8c 6a 9b 03 00 00 08
00000020: 01 00 00 21 00 00 00 08 05 00 00 00 2c 00 00 24
00000030: b5 48 18 7d 30 d8 ea 49 20 d0 9d 42 de 9e 91 ce
00000040: b3 1c 41 85 37 66 d8 9e c6 a6 f8 08 93 f4 48 23
00000050: 2d 00 00 18 01 00 00 00 07 00 00 10 00 00 ff ff
00000060: 0a 01 01 03 0a 01 01 03 29 00 00 18 01 00 00 00
00000070: 07 00 00 10 00 00 ff ff 0a 00 00 00 0a 00 00 ff
00000080: 29 00 00 08 00 00 40 0a 00 00 00 08 00 00 40 0b
00000090: 00
```

(23) Parses received message

```
Create Child SA
#FDD9358950D5DB22.81275DA298901A06.00000000 IKEv2 I->R[193]
  E[165]{
    N[12](ESP:6C0CA570:REKEY_SA),
    SA[32]{
      P[28](#1:ESP:9A8C6A9B:2#){
        Encryption=ENCR_MAGMA_MGM_KTREE,
        ESN=Off}},
    NONCE[36]{B54818...F44823},
    TSi[24](1#){10.1.1.3},
    TSr[24](1#){10.0.0.0-10.0.0.255},
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(24) Generates random IKE nonce Nr

```
00000000: 41 5e a7 ed 7e 65 d3 ff d3 df ed 5f b5 c8 5c 60
00000010: 2b 9c 15 14 eb 52 97 b7 fc aa 33 c4 64 f3 58 06
```

(25) Selects SPI for new incoming ESP SA

```
00000000: 15 4f 35 39
```

(26) Computes keys for new ESP SAs

```
00000000: 6a b6 a0 e7 05 d3 51 16 6f 4f b9 d6 59 0c c8 69
00000010: 43 70 cf 6f 0d 32 c3 7d 92 75 00 4b 0a 76 35 67
00000020: 64 0e 3a fe
00000000: 65 56 1c 79 27 cb c6 d6 8c b8 69 0f 40 00 d2 0a
00000010: c1 49 1c d1 86 88 db 88 ae f3 be 82 0c 71 b7 c9
00000020: 6c cf a3 64
```

(27) Creates message

```
Create Child SA
#FDD9358950D5DB22.81275DA298901A06.00000000 IKEv2 I<=R[189]
  E[161]{
    SA[32]{
      P[28](#1:ESP:154F3539:2#){
        Encryption=ENCR_MAGMA_MGM_KTREE,
        ESN=Off}},
    NONCE[36]{415EA7...F35806},
    TSi[24](1#){10.1.1.3},
    TSr[24](1#){10.0.0.0-10.0.0.255},
    N[8](ADDITIONAL_TS_POSSIBLE),
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(28) Computes K1r (i1 = 0)

```
00000000: 51 49 d5 41 33 91 45 dd ff 04 f5 05 e5 21 39 f2
00000010: 3a 71 1c 18 ef 39 94 1e dd 0c 70 e5 14 12 43 0a
```

(29) Computes K2r (i2 = 0)

```
00000000: 0e 8f 21 54 2e fc 81 79 57 c4 c9 0b e0 25 9a 59
00000010: 29 26 0e 86 20 bf d4 e6 00 32 23 43 ae f0 11 52
```

(30) Computes K3r (i3 = 0)

```
00000000: 92 b8 b2 d6 7a 2d e1 db 5f e1 39 d2 57 c8 24 5f
00000010: f6 22 54 de fc 35 35 c9 24 cf a5 4a e1 5d 75 71
```

(31) Composes MGM nonce

```
00000000: 00 00 00 00 d2 f6 27 21
```

(32) Composes AAD

```
00000000: fd d9 35 89 50 d5 db 22 81 27 5d a2 98 90 1a 06
00000010: 2e 20 24 20 00 00 00 00 00 00 00 bd 21 00 00 a1
```

(33) Composes plaintext

```
00000000: 28 00 00 20 00 00 00 1c 01 03 04 02 15 4f 35 39
00000010: 03 00 00 08 01 00 00 21 00 00 00 08 05 00 00 00
00000020: 2c 00 00 24 41 5e a7 ed 7e 65 d3 ff d3 df ed 5f
00000030: b5 c8 5c 60 2b 9c 15 14 eb 52 97 b7 fc aa 33 c4
00000040: 64 f3 58 06 2d 00 00 18 01 00 00 00 07 00 00 10
00000050: 00 00 ff ff 0a 01 01 03 0a 01 01 03 29 00 00 18
00000060: 01 00 00 00 07 00 00 10 00 00 ff ff 0a 00 00 00
00000070: 0a 00 00 ff 29 00 00 08 00 00 40 02 29 00 00 08
00000080: 00 00 40 0a 00 00 00 08 00 00 40 0b 00
```

(34) Encrypts plaintext using K3r as K_msg, resulting in ciphertext

```
00000000: 2e c7 13 73 4c cc f8 f3 51 71 ac d9 7a 6e 20 2c
00000010: 68 70 bb 8f 82 42 2a 14 e3 8d b8 25 10 9a 1f b6
00000020: 51 ef c5 35 50 bf df 8e 96 bc 94 5a e5 4d 9d 99
00000030: 9a 14 36 d1 4b 61 e1 de 3b 0d 12 94 e5 72 60 00
00000040: 0f 9d dd 2b e1 97 25 4c 5c ee 48 2e 9b f7 d8 9e
00000050: 01 6b 1d 92 b7 c1 7f 16 81 0f e2 e3 14 1c 27 c7
00000060: 35 e9 e3 fd b8 fc 5d fb a2 ee 2f f9 b0 17 39 ca
00000070: f1 2e b1 13 99 e0 da 10 1a 29 74 26 a3 63 ce 09
00000080: 6a f9 1b 67 4a f2 fb 0f 17 5e 48 1a 93
```

(35) Computes ICV using K3r as K_msg

```
00000000: 57 b4 30 41 07 50 b1 cc
```

(36) Composes IV

```
00000000: 00 00 00 00 00 00 00 00
```

(37) Sends message, peer receives message

```
       10.111.10.171:54295<-10.111.15.45:4500 [193]

00000000: 00 00 00 00 fd d9 35 89 50 d5 db 22 81 27 5d a2
00000010: 98 90 1a 06 2e 20 24 20 00 00 00 00 00 00 00 bd
00000020: 21 00 00 a1 00 00 00 00 00 00 00 00 2e c7 13 73
00000030: 4c cc f8 f3 51 71 ac d9 7a 6e 20 2c 68 70 bb 8f
00000040: 82 42 2a 14 e3 8d b8 25 10 9a 1f b6 51 ef c5 35
00000050: 50 bf df 8e 96 bc 94 5a e5 4d 9d 99 9a 14 36 d1
00000060: 4b 61 e1 de 3b 0d 12 94 e5 72 60 00 0f 9d dd 2b
00000070: e1 97 25 4c 5c ee 48 2e 9b f7 d8 9e 01 6b 1d 92
00000080: b7 c1 7f 16 81 0f e2 e3 14 1c 27 c7 35 e9 e3 fd
00000090: b8 fc 5d fb a2 ee 2f f9 b0 17 39 ca f1 2e b1 13
000000A0: 99 e0 da 10 1a 29 74 26 a3 63 ce 09 6a f9 1b 67
000000B0: 4a f2 fb 0f 17 5e 48 1a 93 57 b4 30 41 07 50 b1
000000C0: cc
```

Initiator's actions:

(38) Extracts IV from message

```
00000000: 00 00 00 00 00 00 00 00
```

(39) Computes K1r (i1 = 0)

```
00000000: 51 49 d5 41 33 91 45 dd ff 04 f5 05 e5 21 39 f2
00000010: 3a 71 1c 18 ef 39 94 1e dd 0c 70 e5 14 12 43 0a
```

(40) Computes K2r (i2 = 0)

```
00000000: 0e 8f 21 54 2e fc 81 79 57 c4 c9 0b e0 25 9a 59
00000010: 29 26 0e 86 20 bf d4 e6 00 32 23 43 ae f0 11 52
```

(41) Computes K3r (i3 = 0)

```
00000000: 92 b8 b2 d6 7a 2d e1 db 5f e1 39 d2 57 c8 24 5f
00000010: f6 22 54 de fc 35 35 c9 24 cf a5 4a e1 5d 75 71
```

(42) Composes MGM nonce

```
00000000: 00 00 00 00 d2 f6 27 21
```

(43) Extracts ICV from message

```
00000000: 57 b4 30 41 07 50 b1 cc
```

(44) Extracts AAD from message

```
00000000: fd d9 35 89 50 d5 db 22 81 27 5d a2 98 90 1a 06
00000010: 2e 20 24 20 00 00 00 00 00 00 00 bd 21 00 00 a1
```

(45) Extracts ciphertext from message

```
00000000: 2e c7 13 73 4c cc f8 f3 51 71 ac d9 7a 6e 20 2c
00000010: 68 70 bb 8f 82 42 2a 14 e3 8d b8 25 10 9a 1f b6
00000020: 51 ef c5 35 50 bf df 8e 96 bc 94 5a e5 4d 9d 99
00000030: 9a 14 36 d1 4b 61 e1 de 3b 0d 12 94 e5 72 60 00
00000040: 0f 9d dd 2b e1 97 25 4c 5c ee 48 2e 9b f7 d8 9e
00000050: 01 6b 1d 92 b7 c1 7f 16 81 0f e2 e3 14 1c 27 c7
00000060: 35 e9 e3 fd b8 fc 5d fb a2 ee 2f f9 b0 17 39 ca
00000070: f1 2e b1 13 99 e0 da 10 1a 29 74 26 a3 63 ce 09
00000080: 6a f9 1b 67 4a f2 fb 0f 17 5e 48 1a 93
```

(46) Decrypts ciphertext and verifies ICV using K3r as K_msg, resulting in plaintext

```
00000000: 28 00 00 20 00 00 00 1c 01 03 04 02 15 4f 35 39
00000010: 03 00 00 08 01 00 00 21 00 00 00 08 05 00 00 00
00000020: 2c 00 00 24 41 5e a7 ed 7e 65 d3 ff d3 df ed 5f
00000030: b5 c8 5c 60 2b 9c 15 14 eb 52 97 b7 fc aa 33 c4
00000040: 64 f3 58 06 2d 00 00 18 01 00 00 00 07 00 00 10
00000050: 00 00 ff ff 0a 01 01 03 0a 01 01 03 29 00 00 18
00000060: 01 00 00 00 07 00 00 10 00 00 ff ff 0a 00 00 00
00000070: 0a 00 00 ff 29 00 00 08 00 00 40 02 29 00 00 08
00000080: 00 00 40 0a 00 00 00 08 00 00 40 0b 00
```

(47) Parses received message

```
Create Child SA
#FDD9358950D5DB22.81275DA298901A06.00000000 IKEv2 R=>I[189]
  E[161]{
    SA[32]{
      P[28](#1:ESP:154F3539:2#){
        Encryption=ENCR_MAGMA_MGM_KTREE,
        ESN=Off}},
    NONCE[36]{415EA7...F35806},
    TSi[24](1#){10.1.1.3},
    TSr[24](1#){10.0.0.0-10.0.0.255},
    N[8](ADDITIONAL_TS_POSSIBLE),
    N[8](ESP_TFC_PADDING_NOT_SUPPORTED),
    N[8](NON_FIRST_FRAGMENTS_ALSO)}
```

(48) Computes keys for new ESP SAs

```
00000000: 6a b6 a0 e7 05 d3 51 16 6f 4f b9 d6 59 0c c8 69
00000010: 43 70 cf 6f 0d 32 c3 7d 92 75 00 4b 0a 76 35 67
00000020: 64 0e 3a fe
00000000: 65 56 1c 79 27 cb c6 d6 8c b8 69 0f 40 00 d2 0a
00000010: c1 49 1c d1 86 88 db 88 ae f3 be 82 0c 71 b7 c9
00000020: 6c cf a3 64
```

A.2.4. Sub-Scenario 4: IKE SA Deletion Using the INFORMATIONAL Exchange

```
Initiator                             Responder

HDR, SK {D}           --->
                      <---            HDR, SK { }
```

Initiator's actions:

(1) Creates message

```
Informational
#FDD9358950D5DB22.81275DA298901A06.00000003 IKEv2 R<-I[57]
  E[29]{
    D[8](IKE)}
```

(2) Uses previously computed key K3i

```
00000000: 45 6f 03 f7 ad 75 eb e9 52 b8 8f 0d e8 36 47 69
00000010: 4d 2e f2 ba 15 e6 8c 89 1c 99 62 64 fb 0e 70 0a
```

(3) Composes MGM nonce

```
00000000: 00 00 00 03 2b 3d 3b 2f
```

(4) Composes AAD

```
00000000: fd d9 35 89 50 d5 db 22 81 27 5d a2 98 90 1a 06
00000010: 2e 20 25 08 00 00 00 03 00 00 00 39 2a 00 00 1d
```

(5) Composes plaintext

```
00000000: 00 00 00 08 01 00 00 00 00
```

(6) Encrypts plaintext using K3i as K_msg, resulting in ciphertext

```
00000000: 4f ff 67 66 41 9c d3 ec 8e
```

(7) Computes ICV using K3i as K_msg

```
00000000: d2 bf 0e b7 8f c5 53 03
```

(8) Composes IV

```
00000000: 00 00 00 00 00 00 00 03
```

(9) Sends message, peer receives message

```
       10.111.10.171:54295->10.111.15.45:4500 [61]

00000000: 00 00 00 00 fd d9 35 89 50 d5 db 22 81 27 5d a2
00000010: 98 90 1a 06 2e 20 25 08 00 00 00 03 00 00 00 39
00000020: 2a 00 00 1d 00 00 00 00 00 00 00 03 4f ff 67 66
00000030: 41 9c d3 ec 8e d2 bf 0e b7 8f c5 53 03
```

Responder's actions:

(10) Extracts IV from message

```
00000000: 00 00 00 00 00 00 00 03
```

(11) Uses previously computed key K3i

```
00000000: 45 6f 03 f7 ad 75 eb e9 52 b8 8f 0d e8 36 47 69
00000010: 4d 2e f2 ba 15 e6 8c 89 1c 99 62 64 fb 0e 70 0a
```

(12) Composes MGM nonce

```
00000000: 00 00 00 03 2b 3d 3b 2f
```

(13) Extracts ICV from message

```
00000000: d2 bf 0e b7 8f c5 53 03
```

(14) Extracts AAD from message

```
00000000: fd d9 35 89 50 d5 db 22 81 27 5d a2 98 90 1a 06
00000010: 2e 20 25 08 00 00 00 03 00 00 00 39 2a 00 00 1d
```

(15) Extracts ciphertext from message

```
00000000: 4f ff 67 66 41 9c d3 ec 8e
```

(16) Decrypts ciphertext and verifies ICV using K3i as K_msg, resulting in plaintext

```
00000000: 00 00 00 08 01 00 00 00 00
```

(17) Parses received message

```
Informational
#FDD9358950D5DB22.81275DA298901A06.00000003 IKEv2 I->R[57]
  E[29]{
    D[8](IKE)}
```

(18) Creates message

```
Informational
#FDD9358950D5DB22.81275DA298901A06.00000003 IKEv2 I<=R[49]
  E[21]{}
```

(19) Uses previously computed key K3r

```
00000000: 92 b8 b2 d6 7a 2d e1 db 5f e1 39 d2 57 c8 24 5f
00000010: f6 22 54 de fc 35 35 c9 24 cf a5 4a e1 5d 75 71
```

(20) Composes MGM nonce

```
00000000: 00 00 00 03 d2 f6 27 21
```

(21) Composes AAD

```
00000000: fd d9 35 89 50 d5 db 22 81 27 5d a2 98 90 1a 06
00000010: 2e 20 25 20 00 00 00 03 00 00 00 31 00 00 00 15
```

(22) Composes plaintext

```
00000000: 00
```

(23) Encrypts plaintext using K3r as K_msg, resulting in ciphertext

```
00000000: a8
```

(24) Computes ICV using K3r as K_msg

```
00000000: ef 77 21 c9 8b c1 eb 98
```

(25) Composes IV

```
00000000: 00 00 00 00 00 00 00 03
```

(26) Sends message, peer receives message

```
       10.111.10.171:54295<-10.111.15.45:4500 [53]

00000000: 00 00 00 00 fd d9 35 89 50 d5 db 22 81 27 5d a2
00000010: 98 90 1a 06 2e 20 25 20 00 00 03 00 00 00 31
00000020: 00 00 00 15 00 00 00 00 00 00 03 a8 ef 77 21
00000030: c9 8b c1 eb 98
```

Initiator's actions:

(27) Extracts IV from message

```
00000000: 00 00 00 00 00 00 00 03
```

(28) Uses previously computed key K3r

```
00000000: 92 b8 b2 d6 7a 2d e1 db 5f e1 39 d2 57 c8 24 5f
00000010: f6 22 54 de fc 35 35 c9 24 cf a5 4a e1 5d 75 71
```

(29) Composes MGM nonce

```
00000000: 00 00 00 03 d2 f6 27 21
```

(30) Extracts ICV from message

```
00000000: ef 77 21 c9 8b c1 eb 98
```

(31) Extracts AAD from message

```
00000000: fd d9 35 89 50 d5 db 22 81 27 5d a2 98 90 1a 06
00000010: 2e 20 25 20 00 00 00 03 00 00 00 31 00 00 00 15
```

(32) Extracts ciphertext from message

```
00000000: a8
```

(33) Decrypts ciphertext and verifies ICV using K3r as K_msg, resulting in plaintext

```
00000000: 00
```

(34) Parses received message

```
Informational
#FDD9358950D5DB22.81275DA298901A06.00000003 IKEv2 R=>I[49]
  E[21]{}
```

Author's Address

Valery Smyslov
ELVIS-PLUS
PO Box 81
Moscow (Zelenograd)
124460
Russian Federation
Phone: +7 495 276 0211
Email: svan@elvis.ru