**3GPP TSG-SA3 Meeting #116 *S3-242414-r2***

Jeju, South Korea, 20th - 24th May 2024

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.1* | | | | | | | | |
| **DRAFT CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **S3-242531** | **CR** |  | **rev** |  | **Current version:** | **18.0.0** |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **X** | Radio Access Network |  | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | Remove insecure usage of “aes-gcm” and “aes-gmac” | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Ericsson | | | | | | | | | |
| ***Source to TSG:*** | S3 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | CryptoSP | | | | |  | ***Date:*** | | | 2024-05-13 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **F** |  | | | | | ***Release:*** | | | Rel-19 |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-15 (Release 15) Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | The use of “aes-gcm” and “aes-gmac” are utterly broken and insecure in the way they are used in IMS. A single key and salt are used for at least four associations and it is very likely that IVs are 64-bit counters all starting with the value zero. A single (key, nonce) collision enables an attacker to compromise the authentication subkey H inside the algorithms, which allows the attacker to forge any number of chosen messages. This attack is practical.  AES-CBC is secure when used together with an integrity algorithm in IPsec. No practical vulnerability is known for HMAC-SHA-1, which relies on the preimage security of the hash function. AES-CBC and HMAC-SHA-1 are therefore strongly preferred over “aes-gcm” and “aes-gmac” in IMS.  The use of RFC 3329 instead of IKEv2 puts additional requirement on the key expansion in Annex I. This is not clear from RFC 3329 or 33.203 and was likely missed when “aes-gcm” and “aes-gmac” were added. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | - “aes-gcm” and “aes-gmac” are removed.  - The sentence that AES-CBC is not recommended is removed.  - A note on requirements for new algorithms is added.  - Added that “null” shall also be used with “aes-gcm-us”. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | The completely insecure GCM and GMAC might be negotiated leading to attackers forging any number of chosen messages. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 2, Annex H, Annex I | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **X** | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  | **X** | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  | **X** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

## \*\*\*\*\*\* START OF CHANGES **\*\*\*\***

Annex H (normative):  
The use of "Security Mechanism Agreement for SIP Sessions" [21] for security mode set-up

The BNF syntax of RFC 3329 [21] is defined for negotiating security associations for semi-manually keyed IPsec or TLS in the following way:

security-client = "Security-Client" HCOLON sec-mechanism \*(COMMA sec-mechanism)

security-server = "Security-Server" HCOLON sec-mechanism \*(COMMA sec-mechanism)

security-verify = "Security-Verify" HCOLON sec-mechanism \*(COMMA sec-mechanism)

sec-mechanism = mechanism-name \*(SEMI mech-parameters)

mechanism-name = "ipsec-3gpp" / "tls"

mech-parameters = ( preference / algorithm / protocol / mode / encrypt-algorithm / spi‑c / spi‑s / port‑c / port‑s )

preference = "q" EQUAL qvalue

qvalue = ( "0" [ "." 0\*3DIGIT ] ) / ( "1" [ "." 0\*3("0") ] )

algorithm = "alg" EQUAL ("hmac-sha-1-96" / "aes-gmac- us " / "null" )

protocol = "prot" EQUAL ( "ah" / "esp" )

mode = "mod" EQUAL ( "trans" / "tun" / "UDP-enc-tun" )

encrypt-algorithm = "ealg" EQUAL ("aes-cbc" / "aes-gcm- us" / "null" )

spi‑c = "spi‑c" EQUAL spivalue

spi‑s = "spi‑s" EQUAL spivalue

spivalue = 10DIGIT; 0 to 4294967295

port‑c = "port‑c" EQUAL port

port‑s = "port‑s" EQUAL port

port = 1\*DIGIT

The changes compared to RFC 3329 [21] are:

"alg" parameter: Addition of , "aes-gmac- us" and "null". Removal of "hmac-md5-96"

"ealg" parameter: Addition of "aes-cbc and "aes-gcm- us". Removal of "des-ede3-cbc"

"mod" parameter: Addition of "UDP-enc-tun"

"hmac-sha-1-96" is not recommended.

The use of security association parameters is specified in clauses 7.1, 7.2, M.7.1 and M.7.2 of the present document. The parameters described by the BNF above have the following semantics:

Mechanism-name: For manually keyed IPsec, this field includes the value "ipsec-3gpp". "ipsec‑3gpp" mechanism extends the general negotiation procedure of RFC 3329 [21] in the following way:

1 The server shall store the Security-Client header received in the request before sending the response with the Security-Server header.

2 The client shall include the Security-Client header in the first protected request. In other words, the first protected request shall include both Security-Verify and Security-Client header fields.

3 The server shall check that the content of Security-Client headers received in previous steps (1 and 2) are the same.

Mech-parameters: Of the mech-parameters, only preference is relevant when the mechanism-name has the value "tls".

Preference: As defined in RFC 3329 [21].

Algorithm: Defines the authentication algorithm. The algorithm parameter is mandatory. The value "aes-gmac-us" refers to the authentication algorithm ENCR\_NULL\_AUTH\_AES\_GMAC defined in IETF RFC 4543 [74]. The value "null" shall only be used with encryption algorithm "aes-gcm-us".

Protocol: Defines the IPsec protocol. May have a value "ah" or "esp". If no Protocol parameter is present, the value will be "esp".

NOTE 1: According to clause 6 only "esp" (RFC 4303 [54]) is allowed for use in IMS.

Mode: Defines the mode in which the IPsec protocol is used. May have a value "trans" for transport mode, and value "tun" for tunneling mode. If no Mode parameter is present, the value will be "trans".

NOTE 2: Void.

Encrypt-algorithm: If present, defines the encryption algorithm. The value "aes-cbc" refers to the algorithm defined in IETF RFC 3602 [22]. The value "aes-gcm-us" refers to the encryption algorithm AES-GCM with a 16 octet ICV defined in IETF RFC 4106 [73]. If no Encrypt-algorithm parameter is present, the algorithm will be "null". The value "aes-gcm-us" shall only be used with authentication algorithm equal to "null".

Spi‑c: Defines the SPI number of the inbound SA at the protected client port.

Spi‑s: Defines the SPI number of the inbound SA at the protected server port.

Port‑c: Defines the protected client port.

Port‑s: Defines the protected server port.

It is assumed that the underlying IPsec implementation supports selectors that allow all transport protocols supported by SIP to be protected with a single SA.

## \*\*\*\*\*\* NEXT CHANGE **\*\*\*\***

Annex I (normative):  
Key expansion functions for IPsec ESP

**Integrity Keys:**

If the selected authentication algorithm is HMAC-SHA-1-96 then IKESP is obtained from IKIM by appending 32 zero bits to the end of IKIM to create a 160‑bit string.

If selected authentication algorithm is AES-GMAC as specified in RFC 4543 [74] with 128 bit key then IKESP = IKIM.

The salt value specified in Section 3.2 of RFC 4543 [74] shall be derived using the key derivation function KDF defined in Annex B of TS 33.220 [66]. The input Key to the KDF function shall be equal to the concatenation of CKIM and IKIM: CKIM || IKIM. The input S to the KDF function shall be formed from the following parameters:

- FC = 0x58.

- P0 = "AES\_GMAC\_SALT" .

- L0 = length of the string “AES\_GMAC\_SALT” (i.e. 0x00 0x0D).

The salt value for each IPsec SA shall consist of the 32 least significant bits of the 256 bits of the KDF output XOR’d with the 2 bits — one bit representing for the direction of the SA ("0" for UE to P-CSCF, "1" for P-CSCF to UE) and one bit representing for the role of the source (UE or P-CSCF) of the SA ("0" for client, "1" for server). The direction bit will be XOR’d with the LSB of the 32-bit string, which is extracted from the 256-bit output of the KDF. The role bit will be XOR’d with the second LSB of the 32-bit string, which is extracted from the 256-bit output of the KDF.

"hmac-sha-1-96" is not recommended.

**Encryption Keys:**

If selected encryption algorithm is AES‑CBC as specified in RFC 3602 [22] with 128 bit key then CKESP = CKIM .

If selected encryption algorithm is AES‑GCM as specified in RFC 4106 [73] with 128 bit key then CKESP = CKIM. The salt value specified in Section 4 of RFC 4106 [73] shall be derived using the key derivation function KDF defined in Annex B of TS 33.220 [66]. The input Key to the KDF function shall be equal to the concatenation of CKIM and IKIM: CKIM || IKIM. The input S to the KDF function shall be formed from the following parameters:

- FC = 0x59

- P0 = “AES\_GCM\_SALT”

- L0 = length of the string “AES\_GCM\_SALT” (i.e. 0x00 0x0C)

The salt value for each IPsec SA shall consist of the 32 least significant bits of the 256 bits of the KDF output XOR’d with the 2 bits — one bit representing for the direction of the SA ("0" for UE to P-CSCF, "1" for P-CSCF to UE) and one bit representing for the role of the source (UE or P-CSCF) of the SA ("0" for client, "1" for server). The direction bit will be XOR’d with the LSB of the 32-bit string, which is extracted from the 256-bit output of the KDF. The role bit will be XOR’d with the second LSB of the 32-bit string, which is extracted from the 256-bit output of the KDF.

NOTE 1: IKEv2 derives a new key for each ESP association. In IMS, which uses RFC 3329 [21], IKESP and CKESP are reused for an unlimited number of ESP associations. This puts additional requirements on the key expansion and needs to be considered when adding new algorithms.

## \*\*\*\*\*\* END OF CHANGES **\*\*\*\***