FC-SP-3 Combined mode encryption/integrity algorithm usage in SA Management protocol

- An algorithm that provides both encryption and integrity is called a combined mode algorithm (e.g, an Authenticated Encryption with Associated Data (AEAD) such as GCM)
- Such an algorithm is specified as an encryption transform in an SA payload.
- When using a combined mode algorithm, FC-SP-2 requires that an integrity algorithm of AUTH_NONE be provided as the transform type 3.
- IKEv2 (and CNSA) allows you to omit the type 3 transform.
- FC-SP-3 should allow the same option as IKEv2.
- This simplifies the SA Management protocol.

FC-SP-2 text:

6.3.2.2 (SA) Payload Structure

…

Each SA Proposal/Protocol structure is followed by one or more Transform structures. The number of different Transforms is generally determined by the protocol. CT_Authentication may have two Transforms, an integrity check algorithm and an encryption algorithm. ESP_Header may have two Transforms, an integrity check algorithm and an encryption algorithm. IKE generally has four Transforms, a Diffie-Hellman group, an integrity check algorithm, a pseudo-random function, and an encryption algorithm. If an algorithm that combines encryption and integrity protection is proposed, it shall be proposed as an encryption algorithm, and the AUTH_NONE integrity protection algorithm shall be proposed.

NOTE 25 – Unlike IKEv2, this standard specifies that the AUTH_NONE integrity protection algorithm is always proposed when a combined mode encryption algorithm is proposed.

And later in the same section:

NOTE 27 – Unlike IKEv2, this standard specifies that an integrity Transform be always proposed and allows an encryption Transform not to be proposed.

FC-SP-3 proposed text:

6.3.2.2 (SA) Payload Structure

…

Each SA Proposal/Protocol structure is followed by one or more Transform structures. The number of different Transforms is generally determined by the protocol. CT_Authentication may have two Transforms, an integrity check algorithm and an encryption algorithm. ESP_Header may have two Transforms, an integrity check algorithm and an encryption algorithm. IKE generally has four Transforms, a Diffie-Hellman group, an integrity check algorithm, a pseudo-random function, and an encryption algorithm. If an algorithm that combines encryption and integrity protection is proposed, it shall be proposed as an encryption algorithm and either include no integrity protection algorithm or the AUTH_NONE integrity protection algorithm.

~~NOTE 25 – Unlike IKEv2, this standard specifies that the AUTH_NONE integrity protection algorithm is always proposed when a combined mode encryption algorithm is proposed.~~

And later in the same section:

~~NOTE 27 – Unlike IKEv2, this standard specifies that an integrity Transform be always proposed and allows an encryption Transform not to be proposed.~~

Backups and discussion stuff

IKEv2 (RFC 7296)

3.3 Security Association Payload

“...Combined-mode ciphers include both integrity and encryption in a single encryption algorithm, and MUST either offer no integrity algorithm or a single integrity algorithm of "NONE", with no integrity algorithm being the RECOMMENDED method. “

From RFC9206 (CNSA for IPSec)

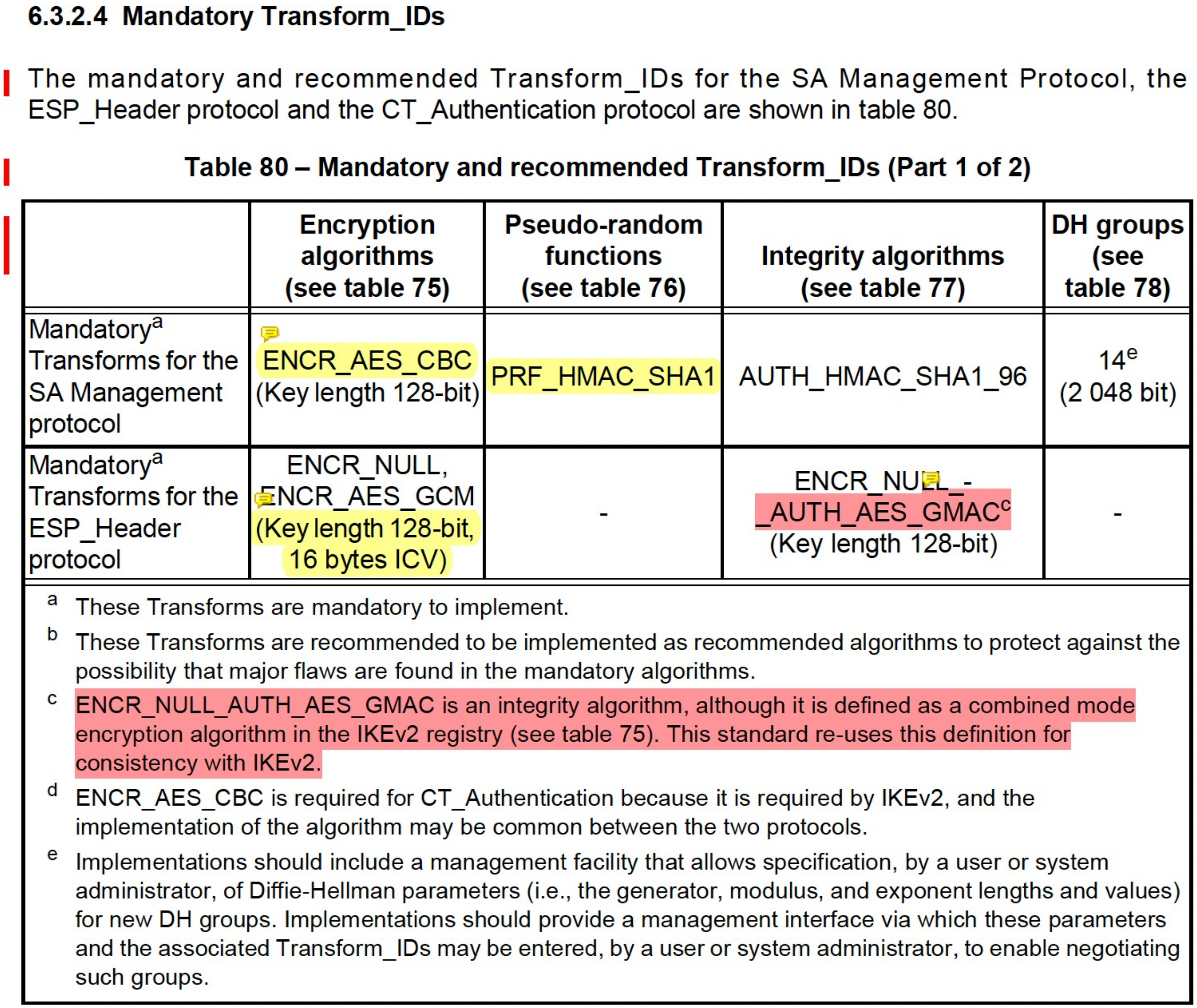
“ESP requires negotiation of both a confidentiality algorithm and an integrity algorithm. However, algorithms for Authenticated Encryption with Associated Data (AEAD) [RFC5116] do not require a separate integrity algorithm to be negotiated. In particular, since AES-GCM is an AEAD algorithm, ESP implementing AES-GCM MUST either offer no integrity algorithm or indicate the single integrity algorithm NONE (see Section 3.3 of [RFC7296]).”

Additional change in Annex E? (Examples of SA Management transactions)

In order to establish an SA for the ESP_Header protocol using the AES-GCM algorithm, that combines encryption and integrity protection, the SA_Initiator sends a single SA Proposal in the Security_Association payload. The SA Proposal contains a Transform of type 1 (i.e., Encryption Algorithm) with Transform_ID set to 20 (i.e., ENCR_AES_GCM with 16 bytes ICV) and a Transform of type 3 (i.e., Integrity Algorithm) with Transform_ID set to 0 (i.e., AUTH_NONE).

In order to establish an SA for the ESP_Header protocol using the AES-GMAC algorithm, that combines NULL encryption and integrity protection, the SA_Initiator sends a single SA Proposal in the Security_Association payload. The SA Proposal contains a Transform of type 1 (i.e., Encryption Algorithm) with Transform_ID set to 21 (i.e., ENCR_NULL_AUTH_AES_GMAC) and a Transform of Type 3 (i.e., Integrity Algorithm) with Transform_ID set to 0 (AUTH_NONE).

This needs help!!
(For Discussion on direction)



6.3.2.4 Mandatory Transform_IDs

The mandatory and recommended Transform_IDs for the SA Management Protocol, the ESP_Header protocol and the CT_Authentication protocol are shown in table 80.

Table 80 – Mandatory and recommended Transform_IDs (Part 1 of 2)

	Encryption algorithms (see table 75)	Pseudo-random functions (see table 76)	Integrity algorithms (see table 77)	DH groups (see table 78)
Mandatory[a] Transforms for the SA Management protocol	ENCR_AES_CBC (Key length 128-bit)	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	14[e] (2 048 bit)
Mandatory[a] Transforms for the ESP_Header protocol	ENCR_NULL, ENCR_AES_GCM (Key length 128-bit, 16 bytes ICV)	-	ENCR_NULL_- _AUTH_AES_GMAC[c] (Key length 128-bit)	-

[a] These Transforms are mandatory to implement.

[b] These Transforms are recommended to be implemented as recommended algorithms to protect against the possibility that major flaws are found in the mandatory algorithms.

[c] ENCR_NULL_AUTH_AES_GMAC is an integrity algorithm, although it is defined as a combined mode encryption algorithm in the IKEv2 registry (see table 75). This standard re-uses this definition for consistency with IKEv2.

[d] ENCR_AES_CBC is required for CT_Authentication because it is required by IKEv2, and the implementation of the algorithm may be common between the two protocols.

[e] Implementations should include a management facility that allows specification, by a user or system administrator, of Diffie-Hellman parameters (i.e., the generator, modulus, and exponent lengths and values) for new DH groups. Implementations should provide a management interface via which these parameters and the associated Transform_IDs may be entered, by a user or system administrator, to enable negotiating such groups.