IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Draft Text for 12.15 Security | | | | |
| Date: 2020-07-30 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Hitoshi Morioka | SRC Software | Fukuoka, JAPAN |  | [hmorioka@src-soft.com](mailto:hmorioka@src-soft.com) |
|  |  |  |  |  |

Abstract

This document describes a TGbc draft text proposal for clause 12 security.

*Yellow marked numbers are temporal and to be assigned by ANA.*

12. Security

*Add the following new subclause in clause 12.*

12.bc Frame authentication for eBCS

12.bc.1 General

eBCS provides one-way frame authentication mechanisms that do not require key negotiation between a transmitter and receivers.

eBCS DL cases use one of the following four types of frame authentication.

* Public Key Frame Authentication (PKFA)
* Hash Chain Frame Authentication (HCFA) with Instant Authentication
* Hash Chain Frame Authentication (HCFA) without Instant Authentication
* No frame authentication with mandatory higher layer source authentication (HLSA)

eBCS UL cases use PKFA or HLSA.

While the PKFA is suitable for occasional small data transfer or time sensitive data transfer, the HCFA is suitable for continuous content distribution such as live streaming or periodic file transfer.

The following preparations are required for both PKFA and HCFA before starting eBCS.

* The certificate(s) of the CA(s) (Certificate Authority) shall be installed into the eBCS receivers.  
  Note: The certificate of the CA(s) may be installed with an application like contents browser. The installation method is out of scope of this standard.
* The eBCS transmitter generates its own private key and public key pair. The public key shall be signed by one of the CAs of which the eBCS receiver have the certificate.

12.bc.2 eBCS public key frame authentication (PKFA)

12.bc.2.1 Signature of the eBCS Info frame

One of the following public key algorithms is used.

* RSA-2048
* ECDSA-P256
* Ed25519

The eBCS transmitter generates an eBCS Info frame when it receives data to be transmitted. The eBCS Info frame contains the following items.

* eBCS Info sequence number
* Timestamp
* Authentication algorithm
* Allowable time difference
* Length of the Certificate of the AP
* Certificate of the AP
* Content Information
* (Data)
* Signature

If the length of the eBCS Info frame is larger than the maximum MMPDU length (Table 9-25 Maximum data unit sizes (in octets) and durations (in microseconds)), the eBCS Info frame shall be fragmented as described in 11.bc.2.4 (eBCS Info frame fragmentation).

If the eBCS Info frame is not fragmented, fill all the fields according to 9.6.7.bc.5 (eBCS Info frame format) except the signature.

Generate signature as following:

Signature = Sign(The eBCS transmitter’s private key, SHAKE128(Transmitter’s MAC address | from the Sequence Number field to the last Content Information field in the eBCS Info frame))

Otherwise, only the first fragment contains the signature.

Signature = Sign(The eBCS transmitter’s private key, SHAKE128(Transmitter’s MAC address | from the Sequence Number field to the last of the first fragment))

And compute the hash value(s) for the following fragment(s).

HashValue = SHAKE128(Transmitter’s MAC address | from the Sequence Number field to the last of the fragment) (1)

The output length of SHAKE128 is 256bit.

Then the eBCS transmitter transmits the eBCS Info frame.

12.bc.2.2 Authentication of the eBCS Info frame

When the eBCS receiver receives the eBCS Info frame, the eBCS receiver shall authenticate it as follows:

1. If the eBCS Info frame is fragmented, the following procedures are applied only to the first fragment.
2. If the difference between the timestamp in the eBCS Info frame and the time of the eBCS receiver is greater than the allowable time difference in the eBCS Info frame, the eBCS Info frame shall be discarded.
3. Verify the certificate of the AP in the eBCS Info frame with the installed certificate of the CA. If the verification fails or the certificate of the CA that signed the certificate of the AP in the eBCS Info frame is not installed, the eBCS Info frame shall be discarded.
4. Verify the signature in the eBCS Info frame with the certificate of the AP in the eBCS Info frame. If the verification is failed, the eBCS Info frame shall be discarded.

If the authentication succeeds,

* The eBCS receiver caches the certificate of the AP and the allowable time difference in the eBCS Info frame.
* If data is present in the Content Information, the eBCS receiver processes the data in the Content Information field(s) in accordance with 11.bc.2.3 (eBCS Info frame generation and usage).

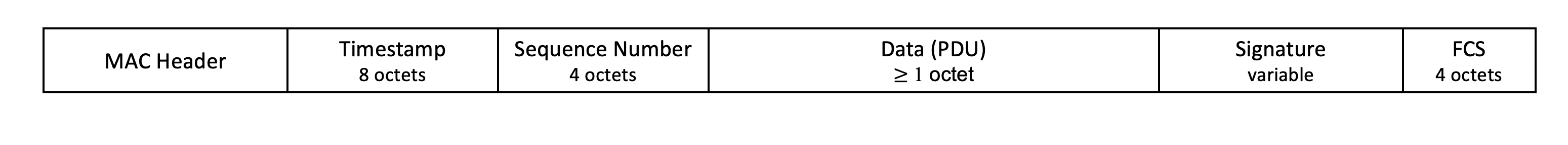
If the eBCS Info frame is fragmented, the eBCS receiver caches the hash value(s) of the fragment(s) and the eBCS receiver shall authenticate the following fragment(s) as following.

1. Compute the hash value of the fragment as described in the formula (1) in 12.bc.2.1(Signing).
2. If the computed hash value is equal to the cached hash value, the authentication succeeds. Otherwise, the fragment shall be discarded.

If the authentication succeeds, the eBCS receiver defragments the eBCS Info frame as described in 11.bc.2.5 (eBCS Info frame defragmentation).

12.bc.2.3 PKFA MPDU format

The MPDU format for PKFA is shown in Figure 12-bc1 (PKFA MPDU format).



**Figure 12-bc1 PKFA MPDU format**

The Timestamp field contains the elapsed time from 2020-01-01 00:00 UTC in milliseconds that indicates the time of the MPDU generation.

The Sequence Number field indicates the sequence number of the frame that starts from 0 and is incremented by each MPDU generation. In case of overflow, it is reset to 0.

The Signature field contains the signature for the MPDU. The signature is generated by the following formula.

Signature = Sign(The eBCS transmitter’s private key, SHAKE128(Transmitter’s MAC address | Timestamp | Sequence Number | Data))

12.bc.2.4 Authentication of a PKFA MPDU

When the eBCS receiver receives the PKFA MPDU, the eBCS receiver shall authenticate it as follows:

1. If the difference between the timestamp in the PKFA MPDU and the time of the eBCS receiver is greater than the cached allowable time difference, the PKFA MPDU shall be discarded.
2. Verify the signature in the PKFA MPDU by the cached certificate. If the verification is failed, the PKFA MPDU shall be discarded, otherwise decapsulate the PDU.

12.bc.2.5 Signature of the eBCS UL frame

One of the following public key algorithms is used.

* RSA-2048
* ECDSA-P256
* Ed25519

The eBCS transmitter generates an E-BCS UL frame when it receives data to be transmitted. The format of the E-BCS UL frame is described in 9.6.7.bc (eBCS UL frame format).

The signature is generated as following:

Signature = Sign(The eBCS transmitter’s private key, SHAKE128(Transmitter’s MAC address | from the Sequence Number field to the last field before the Frame Signature Length field in the eBCS UL frame))

Then the eBCS transmitter transmits the eBCS UL frame.

12.bc.2.6 Authentication of an eBCS UL frame

When an eBCS receiver receives an eBCS UL frame, the eBCS receiver shall authenticate it as follows:

1. If the Timestamp is present and the difference between the timestamp in the eBCS UL frame and the time of the eBCS receiver is greater than the configured value, the eBCS UL frame shall be discarded.
2. Verify the certificate of the STA in the eBCS UL frame using the installed certificate of the CA. If the verification fails or the certificate of the CA that signed the certificate of the STA in the eBCS UL frame is not installed, the eBCS UL frame shall be discarded.
3. Verify the signature in the eBCS UL frame using the certificate of the STA in the eBCS UL frame. If the verification fails, the eBCS UL frame shall be discarded.

If the authentication succeeds,

* The eBCS receiver processes the HLP Payload as described in 11.bc.3.2 (E-BCS UL operation at an eBCS AP).

12.bc.3 eBCS hash chain frame authentication (HCFA)

12.bc.3.1 General

The HCFA uses the digital signature and a modified TESLA (Timed Efficient Stream Loss-Tolerant Authentication, IETF RFC4082).

HCFA provides the following two authentication methods to authenticate each HCFA MPDU.

* HCFA authentication
* Instant authentication (optional)

Each authentication method uses a separate authenticator. HCFA authentication uses the HCFA authenticator, and the instant authentication uses the instant authenticator. The instant authenticator is optionally used to filter the malicious HCFA MPDU.

The HCFA uses both eBCS Info frames (9.6.7.bc.5 eBCS Info frame format) and Data frames. The frame sequence is shown in Figure 12-bc2 (eBCS HCFA frame sequence).

スクリーンショット が含まれている画像

自動的に生成された説明

**Figure 12-bc2 eBCS HCFA frame sequence**

The eBCS Info frames are transmitted periodically in the interval of dot11eBCSInfoInteval (*TI*). *TK* is the HCFA key change interval configured as dot11eBCSHCFAKeyChangeInterval. *TI* shall be a multiple of *TK*. The period between one eBCS Info frame and the next eBCS Info frame is called the “HCFA period”. Each HCFA period is identified by the HCFA sequence number.

Each content has an ID that is indicated in the Content ID subfield in the Content Information field in the eBCS Info frame

The period that uses the same HCFA authentication key is called “Key period”. Each Key period has its sequence number, Key sequence number, starting with 0 at the beginning of each HCFA period. Note that the Key sequence number is different from HCFA key indexes.

Each HCFA MPDU has a sequence number starting from 0 at the beginning of each Key period. The HCFA MPDU is identified by the following identifiers:

* HCFA sequence number
* Content ID
* Key sequence number
* Data sequence number

eBCSData(*s*, *c*, *k*, *d*) represents the HCFA MPDU in which the HCFA sequence number is *s*, the Content ID is *c*, the Key sequence number is *k* and the Data sequence number is *d*. IAuth(*s*, *c*, *k*, *d*) and HAuth(*s*, *c*, *k*, *d*) represent the instant authenticator and the HCFA authenticator for the eBCSData(*s*, *c*, *k*, *d*) respectively. eBCSInfo(*s*) represents the eBCS Info frame for which the HCFA sequence number is *s*.

For example, in case of two content streams, Content A and Content B, the identifiers are shown in Figure 12-bc3 (Identifiers example).

文字と写真のスクリーンショット

自動的に生成された説明

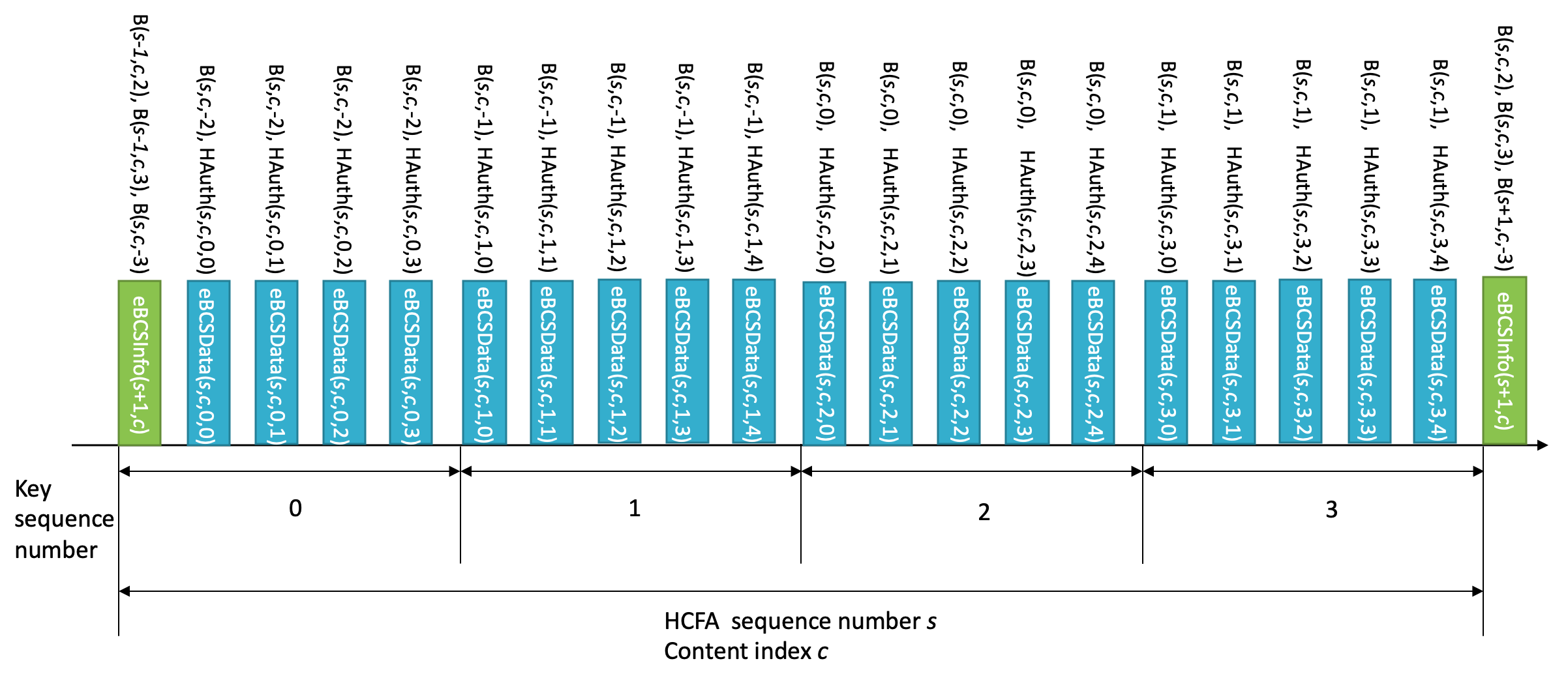
**Figure 12-bc3 Identifiers example**

The index of HCFA base key and HCFA authentication key is defined as B(*s, c, k*) and A(*s, c, k*) respectively where *s* is the HCFA sequence number, *c* is the content ID, *k* is the Key sequence number. The Key sequence number is different from the HCFA base/authentication key index. The HCFA base/authentication keys are used in the opposite order of the HCFA key generation. The relation between the HCFA base/authentication key index and the HCFA sequence number is shown in Table 12-bc1 (Relation between HCFA authentication key index and HCFA sequence number) where N is the number of HCFA authentication keys generated. An example of the HCFA key delivery is shown in Figure 12-bc4 (Example HCFA Key Delivery).

**Table 12-bc1 Relation between HCFA base/authentication key index and HCFA sequence number**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HCFA base/authentication key index | (N-1) | (N-2) | (N-3) | N-4 | N-5 | N-6 | … | 0 |
| HCFA sequence number | (-3) | (-2) | (-1) | 0 | 1 | 2 | … | N-4 |

Note: HCFA sequence number -3, -2 and -1 are used only for key verification.



**Figure 12-bc4 Example HCFA Key Delivery**

An eBCS Info frame and an eBCS Data frame may contain multiple instant authenticators. For example, eBCSData(*s, c, k, d*) may contain IAuth(*s, c, k, d*+1) and IAuth(*s, c, k, d*+3). In this case, the values 1 and 3 are called Hash Distance. The Hash Distance is configured in dot11eBCSHCFAHashDistance. Each instant authenticator is delivered with the frame identifier (*s, c, k, d*). An example of the instant authenticator delivery is shown in Figure 12-bc5 (Example Instant Authenticator Delivery).

スクリーンショット が含まれている画像

自動的に生成された説明

Figure 12-bc5 Example Instant Authenticator Delivery

12.bc.3.2 Key generation

HCFA is a one-way key chain authentication mechanism. The eBCS transmitter generates HCFA base keys and HCFA authentication keys for each content stream before each eBCS Info frame generation. SHAKE128 hash function is used for HCFA key generation. The output length of SHAKE128 is 256bit.

The HCFA base keys (*Bs,n*) are generated as follows:

*Bs,0* = Random value (256bit length)

*Bs,n* = SHAKE128(“eBCS HCFA base key” || *Bs,n-1*) (*n* >= 1)

where *s* is the sequence number of the generating eBCS Info frame.

The HCFA authentication keys (*A s,n*) are generated as follows:

*As,n* = SHAKE128(“eBCS HCFA authentication key” || *Bs,n*)

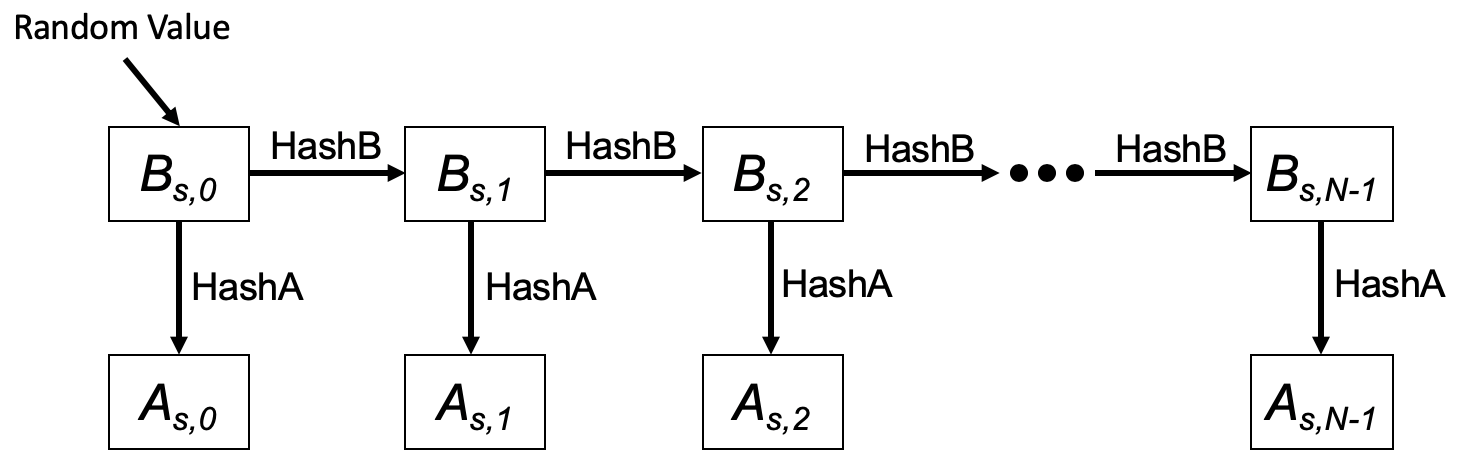
where *s* is the sequence number of the generating eBCS Info frame.

The number of the keys to be generated (*N*) is following:

*N* = *TI* / *TK* + 3

where *TI* is the eBCS Info frame transmission interval and *TK* is the HCFA key change interval.

The HCFA key generation scheme is shown in Figure 12-bc6 (HCFA key generation scheme).



**Figure 12-bc6 HCFA key generation scheme**

In this figure, HashB is the hash function to generate HCFA base keys and HashA is the hash function to generate HCFA authentication keys.

The HCFA keys are generated for each content stream.

12.bc.3.3 eBCS Info frame generation

An eBCS Info frame contains the following items that are related to frame authentication.

* HCFA sequence number
* Timestamp
* Certificate
* Signature
* HCFA key change interval
* Content Information
  + HCFA base key(s)
  + Instant authenticator(s) of HCFA MPDU(s) to be transmitted (optional)

The functions of the eBCS Info sequence number, the timestamp, the certificate and the signature are same as those of PKFA.

The eBCS Info sequence number is used as HCFA sequence number.

The HCFA key change interval, HCFA base key(s) and the instant authenticator(s) are present only in HCFA.

The HCFA key change interval is *TK*.

The HCFA base keys to be included in the eBCS Info frame of sequence number *s* are B(*s*, *c*, -3), B(*s*-1, *c*, 1) and B(*s*-1, *c*, 0) for all content streams, where *c* is the content ID. In case of the first eBCS Info frame, B(*s*-1, *c*, 1) and B(*s*-1, *c*, 0) are not present.

If instant authentication is used, the instant authenticator(s) with frame identifier (*s, c, k, d*) is present. In this case, the eBCS transmitter must buffer data packets to generate instant authenticators.

If the length of the eBCS Info frame is larger than the maximum MMPDU length (Table 9-25 Maximum data unit sizes (in octets) and durations (in microseconds)), the eBCS Info frame shall be fragmented as described in 11.bc.2.4 (eBCS Info frame fragmentation).

12.bc.3.4 HCFA MPDU generation

The HCFA MPDU contains the following items.

* Content data
* HCFA sequence number
* Content ID
* Key sequence number
* Data sequence number
* HCFA base key
* Instant authenticator(s)
* HCFA authenticator

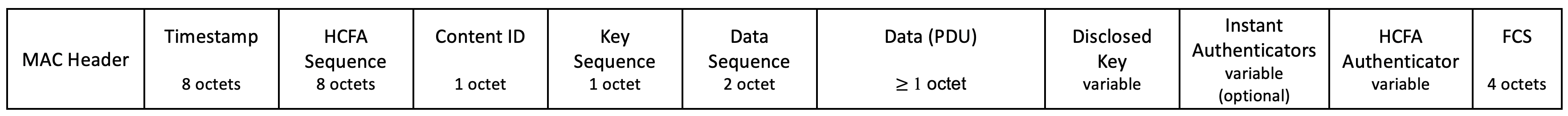
The HCFA sequence number, the Content ID, the Key sequence number and the Data sequence number are described in 12.bc.3.1 (General).

The HCFA base key contained in eBCSData(*s*, *c*, *k*, *d*) is B(*s*, *c*, *k*-2).

The instant authenticator(s) in the eBCSData(*s*, *c*, *k*, *d*) depends on the configured Hash Distance.

The HCFA authenticator in the eBCSData(*s*, *c*, *k*, *d*) is HAuth(*s*, *c*, *k*, *d*) with A(*s*, *c*, *k*).

The MPDU format for HCFA is shown in Figure 12-bc7 (PKFA MPDU format).



**Figure 12-bc7 HCFA MPDU format**

The Timestamp field contains the elapsed time from 2020-01-01 00:00 UTC in milliseconds that indicates the time of the MPDU generation.

The HCFA Sequence field contains the HCFA sequence number of the MPDU.

The Content ID field contains the Content ID of the MPDU.

The Key Sequence field contains the Key sequence number of the MPDU.

The Data Sequence field contains the Data sequence number of the MPDU.

The Disclosed Key field contains the HCFA key to be disclosed.

The Instant Authenticator field contains the instant authenticator(s) of the MPDU(s) that will be transmitted later.

The HCFA Authenticator field contains the HCFA authenticator of the MPDU.

The HCFA authenticator is the KMAC128 (NIST Special Publication 800-185) value of the MPDU with the HCFA authentication key.

HCFA Authenticator = KMAC128(A(s, c, k), Transmitter’s MAC address || from the Timestamp field to the Instant Authenticator field)

The instant authenticator is the hash value of the MPDU to be transmitted later that is generated as following:

Instant Authenticator = SHAKE128(Transmitter’s MAC address || from the Timestamp field to the Instant Authenticator field)

12.bc.3.4 eBCS Info frame reception

The received eBCS Info frame, eBCSInfo(*s*), is processed as following.

1. If the eBCS Info frame is fragmented, defragment it at first as described in 11.bc.2.4 (eBCS Info defragmentation).
2. If the difference between the timestamp in the eBCS Info frame and the time in the eBCS receiver’s clock is greater than the HCFA key change interval in the eBCS Info frame, the eBCS Info frame shall be discarded.
3. Verify the certificate of the AP in the eBCS Info frame using the installed certificate of the CA. If the verification fails or the certificate of the CA that signed the certificate of the AP in the eBCS Info frame is not installed, the eBCS Info frame shall be discarded.
4. Verify the signature in the eBCS Info frame using the certificate of the AP in the eBCS Info frame. If the verification fails, the eBCS Info frame shall be discarded.
5. If the HCFA base key(s) of the previous HCFA period, B(*s*-1, *c*, *N*-4) and B(*s*-1, *c*, *N*-5), is included and the HCFA MPDUs of the previous HCFA period to be authenticated are present, authenticate and process the eBCS Data frames as described in 12.bc.3.5 (eBCS Data frame reception).
6. Cache the HCFA sequence number, *s*, and the HCFA base key(s), B(*s*, *c*, 0), for the HCFA period of the eBCS Info frame.
7. If the instant authenticator(s) are present, cache the instant authenticators contained in the eBCS Info frame.

12.bc.3.4 HCFA MPDU reception

The HCFA MPDU shall be discarded until the eBCS Info frame from the BSS is received.

The received HCFA MPDU, eBCSData(*s*, *c*, *k*, *d*), is processed as following.

1. Compute B(*s*, *c*, *k*-3) from B(*s*, *c*, *k*-2) in the eBCSData(*s*, *c*, *k*, *d*). If the computed B(*s*, *c*, *k*-3) is different from the cached B(*s*, *c* , *k*-3), The HCFA MPDU shall be discarded.
2. If instant authentication is used and the instant authenticator of the eBCSData(*s*, *c*, *k*, *d*), IAuth(*s*, *c*, *k*, *d*), is cached, compute the hash value of the eBCSData(*s*, *c*, *k*, *d*). If the computed hash value is different from the cached instant authenticator, the HCFA MPDU shall be discarded.
3. If instant authentication is used and the instant authenticator of the eBCSData(*s*, *c*, *k*, *d*), IAuth(*s*, *c*, *k*, *d*), is not cached, the HCFA MPDU may be cached until the HCFA base key for the Key period is received, or the HCFA MPDU may be discarded.
4. If instant authentication is not used, the HCFA MPDU shall be cached until the HCFA base key for the Key period is received.
5. If HCFA MPDU(s) using the HCFA authentication key derived from the HCFA base key included in the HCFA MPDU, eBCSData(*s*,*c*,*k*-2,\*), is cached,
6. Derive the HCFA authentication key, A(*s*, *c*, *k*-2), from the HCFA base key, B(*s*, *c*, *k*-2).
7. Compute HCFA authenticator for the cached HCFA MPDU by using the HCFA authentication key.
8. If the computed HCFA authenticator is different from the HCFA authenticator in the cached HCFA MPDU, the cached HCFA MPDU shall be discarded.

Then forward the PDU to the higher layer.

Even in case of missing HCFA MPDUs, the eBCS receiver recovers HCFA keys. For example, if the eBCS receiver missed all HCFA MPDUs containing B(*s*, *c*, *k*) but still cached B(*s*, *c*, *k*-1) and received B(*s*, *c*, *k*+1), the eBCS receiver computes B(*s*, *c*, *k*) and B(*s*, *c*, *k*-1) as follows.

B(*s*, *c*, *k*) = SHAKE128(“eBCS base key” || B(*s*, *c*, *k*+1))

B(*s*, *c*, *k*-1) = SHAKE128(“eBCS base key” || B(*s*, *c*, *k*))

Then the eBCS receiver authenticates the HCFA base keys by comparing the computed B(*s*, *c*, *k*-1) and the cached B(*s*, *c*, *k*-1). After successful key authentication, the eBCS receiver authenticates eBCSData(*s*, *c*, *k*, \*) and eBCSData(*s*, *c*, *k*+1, \*).

12.bc.4 No frame authentication with mandatory higher layer source authentication (HLSA)

If neither PKFA nor HCFA is used, a content source authentication mechanism shall be provided by a higher layer. The higher layer source authentication mechanism is out of scope of this standard. In this case, eBCS Info frames and Data frames for DL or eBCS UL frames for UL are used.

Authentication of eBCS Info frames is optional if the eBCS Info frames include only HLSA content information. The eBCS AP may decide to use eBCS Info frame authentication or not.

If an eBCS Info frame includes the certificate of the AP, the eBCS receiver shall authenticate the eBCS Info frame as described in 12.bc.2.2 (Authentication of the eBCS Info frame).