Summary of Errata and Clarifications to the HDCP DiiVA Specification Rev 2.0

Page 11, Section 2.2, replace Figure 2.1 with the following

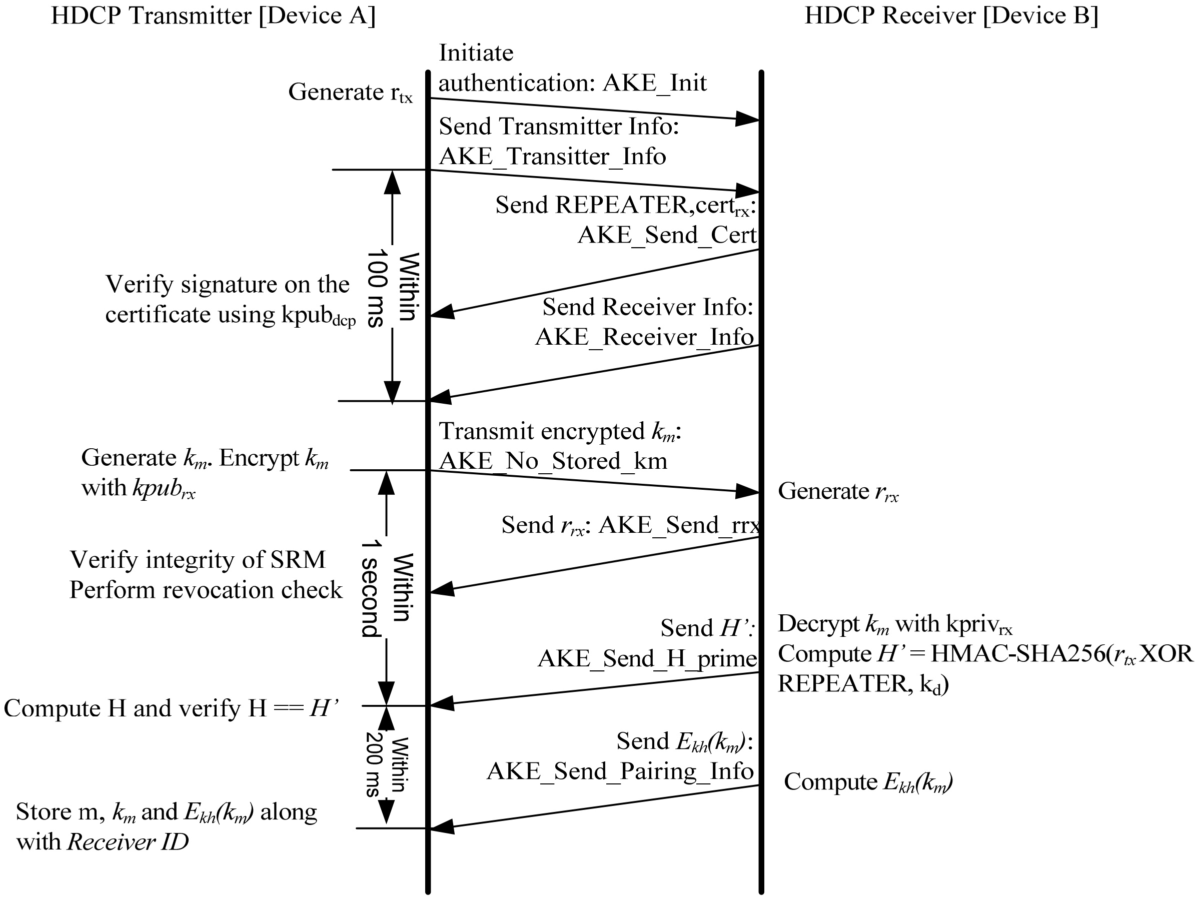
HDCP Transmitter [Device A]
HDCP Receiver [Device B]
Generate r_{tx}
Initiate authentication: AKE_Init
Send Transmitter Info: AKE_Transitter_Info
Send REPEATER,$cert_{rx}$: AKE_Send_Cert
Within 100 ms
Verify signature on the certificate using $kpub_{dcp}$
Send Receiver Info: AKE_Receiver_Info
Generate k_m. Encrypt k_m with $kpub_{rx}$
Transmit encrypted k_m: AKE_No_Stored_km
Generate r_{rx}
Send r_{rx}: AKE_Send_rrx
Within 1 second
Verify integrity of SRM
Perform revocation check
Send H': AKE_Send_H_prime
Decrypt k_m with $kpriv_{rx}$
Compute H' = HMAC-SHA256(r_{tx} XOR REPEATER, k_d)
Compute H and verify H == H'
Within 200 ms
Send $E_{kh}(k_m)$: AKE_Send_Pairing_Info
Compute $E_{kh}(k_m)$
Store m, k_m and $E_{kh}(k_m)$ along with *Receiver ID*

Figure 2.1. Authentication and Key Exchange (Without Stored km)

Page 11, Section 2.2, replace Figure 2.2 with the following

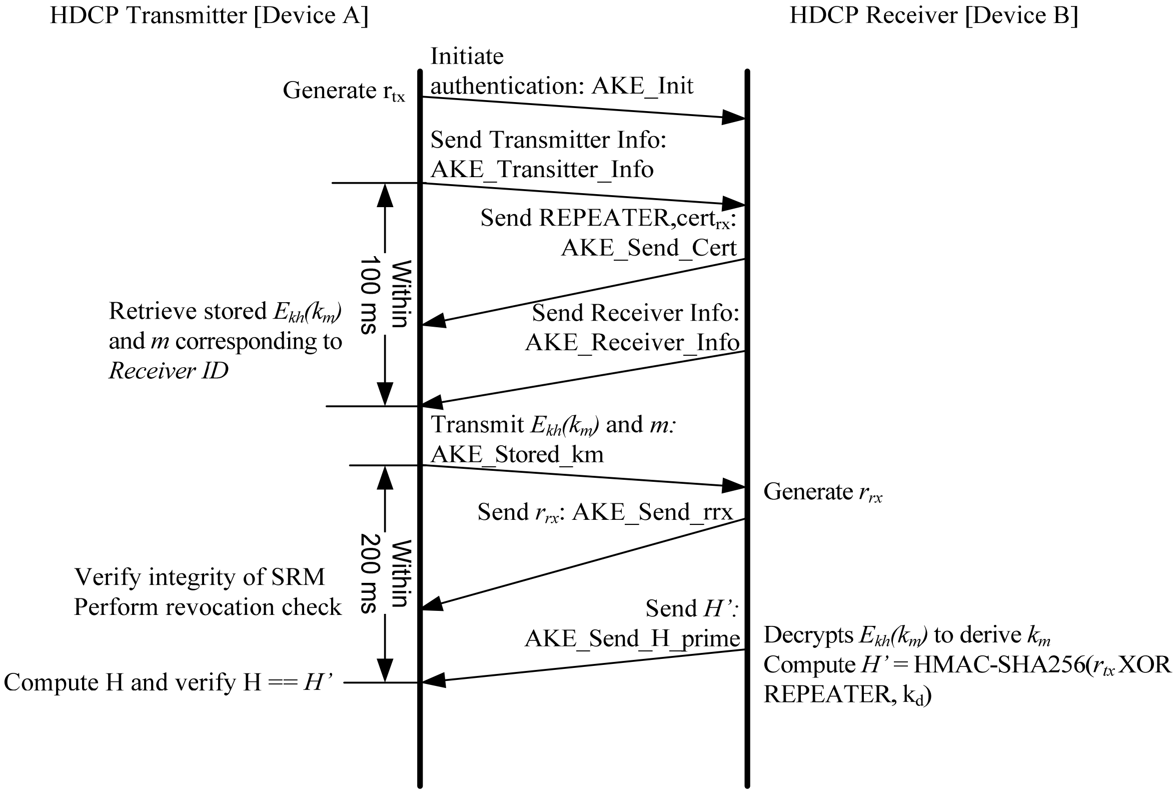
HDCP Transmitter [Device A]
HDCP Receiver [Device B]
Initiate authentication: AKE_Init
Generate r_{tx}
Send Transmitter Info: AKE_Transitter_Info
Send REPEATER,$cert_{rx}$: AKE_Send_Cert
Within 100 ms
Retrieve stored $E_{kh}(k_m)$ and m corresponding to *Receiver ID*
Send Receiver Info: AKE_Receiver_Info
Transmit $E_{kh}(k_m)$ and m: AKE_Stored_km
Generate r_{rx}
Send r_{rx}: AKE_Send_rrx
Within 200 ms
Verify integrity of SRM
Perform revocation check
Send H': AKE_Send_H_prime
Decrypts $E_{kh}(k_m)$ to derive k_m
Compute H' = HMAC-SHA256(r_{tx} XOR REPEATER, k_d)
Compute H and verify H == H'

Figure 2.2. Authentication and Key Exchange (With Stored km)

Page 12, Section 2.2, insert the following step

- Sends AKE_Transmitter_Info message to the HDCP Receiver before sending either AKE_No_Stored_km or AKE_Stored_km message to the receiver.

in between steps “Initiates authentication by sending the initiation message, AKE_Init, containing a 64-bit pseudo-random value (r_{tx}) “ and “Receives AKE_Send_Cert from the receiver containing REPEATER and $cert_{rx}$ values. REPEATER indicates whether the connected receiver is an HDCP Repeater”.

Page 12, Section 2.2, insert the following step

- Receives AKE_Receiver_Info message from the receiver. The contents of the message may be ignored. If AKE_Receiver_Info message is not received within 100 ms from the transmission of AKE_Transmitter_Info message, the HDCP Transmitter aborts the authentication protocol (See Section 2.7 on handling authentication failures).

in between steps “Receives AKE_Send_Cert from the receiver containing REPEATER and $cert_{rx}$ values. REPEATER indicates whether the connected receiver is an HDCP Repeater” and “Extracts Receiver ID from $cert_{rx}$”.

Page 13, Section 2.2, insert the following step

- Sends AKE_Receiver_Info message to the transmitter after sending the AKE_Send_Cert message to the transmitter. The contents of the AKE_Transmitter_Info message may be ignored.

in between steps “Sends AKE_Send_Cert message in response to AKE_Init” and “Generates and sends 64-bit r_{rx} as part of the AKE_Send_rrx message immediately after receiving either AKE_No_Stored_km or AKE_Stored_km message from the transmitter. r_{rx} must be generated only after either AKE_No_Stored_km or AKE_Stored_km message is received from the transmitter”.

Throughout the specification, replace all references to RTT_Response with LC_Send_L_prime.

Page 25, Section 2.9, replace the first paragraph under “State A1. Exchange k_m” with the following

> In this state, the HDCP Transmitter initiates authentication by sending AKE_Init message containing r_{tx} to the HDCP Receiver and sends AKE_Transmitter_Info message to the HDCP Receiver. It receives AKE_Send_Cert from the receiver containing REPEATER and $cert_{rx}$ and AKE_Receiver_Info message. If the HDCP Transmitter does not receive AKE_Receiver_Info message within 100 ms of the transmission of AKE_Transmitter_Info message, the HDCP Transmitter aborts the authentication protocol.

Page 29, Section 2.10, replace the first paragraph under “State B1. Compute k_m” with the following

> In this state, the HDCP Receiver sends AKE_Send_Cert message in response to AKE_Init, sends AKE_Receiver_Info message to the transmitter, generates and sends r_{rx} as part of AKE_Send_rrx message. If AKE_No_Stored_km is received, it decrypts k_m with $kpriv_{rx}$, calculates H’. It sends AKE_Send_H_prime message immediately after computation of H’ to ensure that the message is received by the transmitter within the specified one second timeout at the transmitter.

Page 33, Section 2.11.2, replace the first paragraph under “State F1. Exchange k_m” with the following

> In this state, the downstream side initiates authentication by sending AKE_Init message containing r_{tx} to the HDCP Receiver and sends AKE_Transmitter_Info message to the HDCP Receiver. It receives AKE_Send_Cert from the receiver containing REPEATER and $cert_{rx}$ and AKE_Receiver_Info message. If the downstream side does not receive AKE_Receiver_Info message within 100 ms of the transmission of

AKE_Transmitter_Info message, the HDCP Transmitter aborts the authentication protocol.

Page 35, Section 2.11.3, replace the first paragraph under “State C1. Compute k_m” with the following

In this state, the upstream (HDCP Receiver) side sends AKE_Send_Cert message in response to AKE_Init, sends AKE_Receiver_Info message to the transmitter, generates and sends r_{rx} as part of AKE_Send_rrx message. If AKE_No_Stored_km is received, it decrypts k_m with $kpriv_{rx}$, calculates H’. It sends AKE_Send_H_prime message immediately after computation of H’ to ensure that the message is received by the transmitter within the specified one second timeout at the transmitter

Page 60, Section 4.1, replace Table 4.1 with the following

Message Type	**msg_id Value**
Null message	1
AKE_Init	2
AKE_Send_Cert	3
AKE_No_Stored_km	4
AKE_Stored_km	5
AKE_Send_rrx	6
AKE_Send_H_prime	7
AKE_Send_Pairing_Info	8
LC_Init	9
LC_Send_L_prime	10
SKE_Send_Eks	11
RepeaterAuth_Send_ReceiverID_List	12
RTT_Ready	13
RTT_Challenge	14
Reserved	15
Reserved	16
Reserved	17
Reserved	18
AKE_Transmitter_Info	19
AKE_Receiver_Info	20
Reserved	21-31

Table 4.1. Values for msg_id

Page 60, Section 4.1, replace the fourth paragraph with the following

Note:

- The use of the Null message and Reserved values for msg_id are not defined in this specification. HDCP Devices must be capable of receiving the Null message and messages with reserved msg_id values and must ignore these messages.

Page 62, Replace Section 4.2.9 with the following

4.2.9 RTT_Ready (Receiver to Transmitter)

Syntax	No. of Bytes	Identifier

RTT_Ready {
msg_id
}
	1	uint

Table 4.10. RTT_Ready

Add Section 4.2.14

4.2.14 AKE_Transmitter_Info (Transmitter to Receiver)

A Receiver that receives an AKE_Transmitter_Info message that is longer than specified must read the VERSION and TRANSMITTER_CAPABILITY_MASK parameters and must ignore the additional bytes.

The HDCP Transmitter must set VERSION to 0x00.

Syntax	No. of Bytes	Identifier
AKE_Transmitter_Info{		
msg_id	1	uint
VERSION	1	uint
TRANSMITTER_CAPABILITY_MASK	2	uint
}		

Table 4.15. AKE_Transmitter_Info Payload

Parameter	No. of Bytes	Description
TRANSMITTER_CAPABILITY_MASK	2	Bits 15:1: Reserved zeros. Bit 0: Reserved – Do not define in future.

Table 4.16. TRANSMITTER_CAPABILITY_MASK

Add Section 4.2.15

4.2.15 AKE_Receiver_Info (Receiver to Transmitter)

A transmitter that receives an AKE_Receiver_Info message that is longer than specified must read the VERSION and RECEIVER_CAPABILITY_MASK parameters and must ignore the additional bytes.

The HDCP Receiver must set VERSION to 0x00.

Syntax	No. of Bytes	Identifier
AKE_Receiver_Info{		
msg_id	1	uint
VERSION	1	uint
RECEIVER_CAPABILITY_MASK	2	uint
}		

Table 4.17. AKE_Receiver_Info Payload

Parameter	No. of Bytes	Description
RECEIVER_CAPABILITY_MASK	2	Bits 15:1: Reserved zeros. Bit 0: Reserved – Do not define in future.

Table 4.18. RECEIVER_CAPABILITY_MASK