Key Establishment and Pairing Management Protocol for Downloadable Conditional Access System Host Devices

Han-Seung Koo, O-Hyung Kwon, and Soo In Lee

In this paper, we investigate the possible security threats to downloadable conditional access system (DCAS) host devices. We then propose a DCAS secure micro (SM) and transport processor (TP) security protocol that counters identified security threats using a secure key establishment and pairing management scheme. The proposed protocol not only resists disclosed SM ID and TP ID threats and indirect connection between TA and TP threats, but also meets some desirable security attributes such as known key secrecy, perfect forward secrecy, key compromised impersonation, unknown key-share, and key control.

Keywords: Downloadable conditional access system (DCAS), DCAS protocol, key establishment protocol, pairing protocol.

I. Introduction

General conditional access systems (CASs) for digital television systems operate with CA servers and scramblers in a multiple system operator (MSO) headend, and a set-top box (STB) having a security hardware module, for example, a smartcard or CableCARD [1], that contains a CA client application in its memory at the user side [2]-[5]. Here, a CA client application means a security application program that performs a CAS operation protocol with the CA servers in an MSO headend. Since a CA client application contains proprietary functions of CAS vendors, it is unique for each CAS. Therefore, if a CAS vendor wants to upgrade the CA client application, or the MSO wants to change a CAS to another CA vendor's system, they have to physically change the security hardware module including the upgraded CA client application or the other CA vendor's client application in its memory. As a result, the MSO needs not only time for replacement of security hardware modules, but also extra budget for newly issuing them.

Recently, to eliminate the inefficiency of the CAS described above, the solution of a downloadable CAS (DCAS) was introduced by CableLabs [6]. DCAS is a security system for a digital cable system that delivers a CA client application as an image format, called a CA client image, from the MSO headend to a secure micro (SM), which is the security hardware module of DCAS, through a hybrid fiber and coaxial (HFC) network. By utilizing the DCAS, MSO can replace the already existing CA client application in the SM on-line. As a result, not only is there no

Manuscript received Sept. 15, 2009; revised Mar. 1, 2010; accepted Mar. 8, 2010.

Han-Seung Koo (phone: +82 42 860 1625, email: koohs@etri.re.kr), O-Hyung Kwon (email: ohkwon@etri.re.kr), and Soo In Lee (email: silee@etri.re.kr) are with the Broadcasting & Telecommunications Convergence Research Laboratory, ETRI, Daejeon, Rep. of Korea.

doi:10.4218/etrij.10.1409.0077

additional budget required for issuing a new security hardware module when the MSO wants to change an existing CAS to another CA vendor's system, but the CAS can also quickly upgrade the old CA client application in the SM to a new one. Note that a DCAS host also contains a transport processor (TP) for descrambling the encrypted video streams of pay programs [6]-[9] along with an SM, and SM and TP are called DCAS host devices [6].

To securely download a CA client image, the DCAS authenticates the SM and sends the CA client image in an encrypted form. For this, authentication and CA client image encryption key establishment between the MSO DCAS headend and SM are performed through the DCAS network protocol [7], [8]. After SM authentication and CA client image encryption key establishment are done, the DCAS download server in the MSO DCAS headend delivers the encrypted CA client image to the SM. Once the SM receives the encrypted CA client image, the SM decrypts the image and loads it into the memory of the SM. After that, the CA client image plays a role in the CA client application.

Even though the fundamental requirement of the DCAS is securely downloading a CA client image to the SM, which can be accomplished through the DCAS network protocol [7], [8], the DCAS needs the following additional requirements to securely operate. First, the DCAS has to provide the confidentiality of control words [2]-[5] that are supposed to be delivered from the SM to the TP. Control words are symmetric encryption keys which are used to scramble and descramble video streams. Therefore, control words should be confidentially delivered from a headend to a descrambler [3], [4]. For this, CAS utilizes a hierarchical key management scheme to efficiently deliver the decryption keys for control words to a CA client application located in security hardware module [2], [5]. In general, a descrambler is in a security hardware module such as CableCARD [1], so CAS does not care about the confidentiality of control words delivered from a CA client application to a descrambler. However, since the TP, which has a role as a descrambler in the DCAS, is physically separated from the security hardware module, the DCAS has to provide confidentiality to control words when they are sent from the SM and TP. Second, the DCAS has to manage the pair of SM and TP. This requirement is needed to prevent a service leak that can be caused from the sharing of an SM, including a legitimate CA client application in the group of users. As a result, we need another DCAS security protocol that establishes an encryption key of control words between the SM and TP and manages SM-TP paring. In this paper, we propose the DCAS security protocol, satisfying the additional requirements described above, and name it the DCAS SM and TP security protocol.

We first describe the DCAS architecture overview in section II and discuss the functional and security requirements for the proposed protocol in section III. Next, we describe the proposed DCAS SM and TP security protocol. Section V details an analysis of the proposed protocol based on the identified functional and security requirements. The final section summarizes the results.

II. DCAS Architecture Overview

As shown in Fig. 1, the DCAS consists of a trusted authority (TA), MSO DCAS headend, and DCAS host at the customer premises [7]. The TA issues identification information to the SM and TP and authenticates them. The MSO DCAS headend establishes a CA client image encryption key with the SM and sends an encrypted CA client image to the SM. Finally, the DCAS host is a two-way digital cable set-top that supports downloading the CA client image in the memory of the SM and descrambling the encrypted video streams at the TP.

The overall DCAS operation flows based on the DCAS entities are shown in Fig. 2 with following descriptions.

Step 1. The TA issues identification information of the SM and TP.

Step 2. If the DCAS host is connected to the MSO network, the authentication proxy (AP) and SM establish a CA client image encryption key between them through the DCAS network protocol [7], [8] after the TA authenticates the SM.

Step 3. The DCAS personalize server (PS) sends the encrypted CA client image via download servers to the SM.

Step 4. The SM boot loader decrypts the CA client image and loads the image into the memory of the SM.

Step 5. The CAS in the MSO headend sends entitlement keys for the subscribed pay-programs using an entitlement management message (EMM) [3], [4] and the encrypted control words using an entitlement control message (ECM) [3],

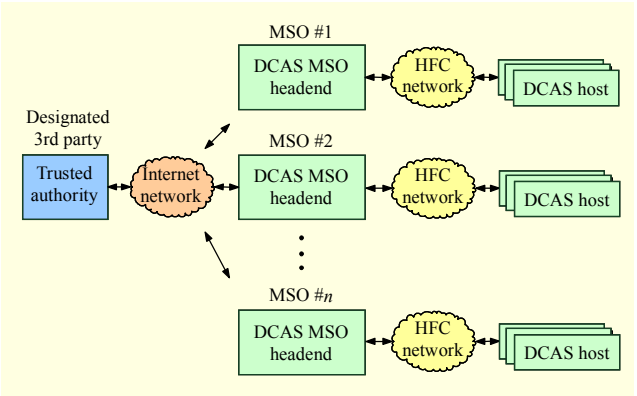

Fig. 1. Generalized DCAS architecture.

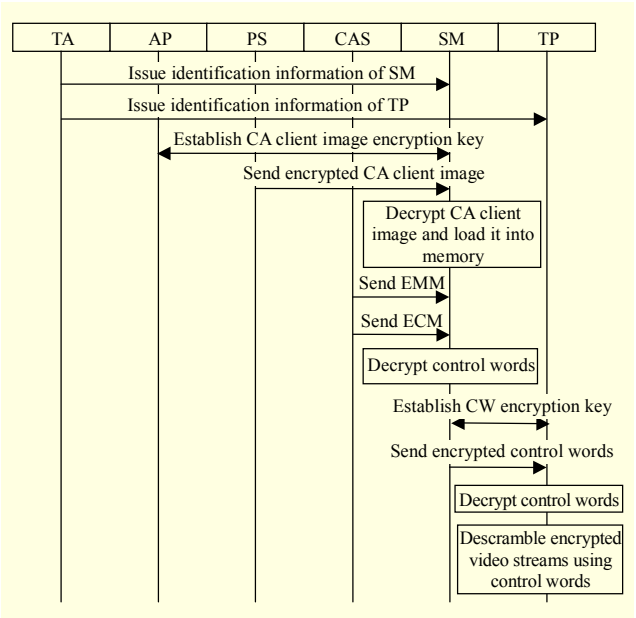

Fig. 2. Overall DCAS operation flows.

[4] to the CA client application in the SM.

Step 6. The CA client application in the SM decrypts control words with the entitlement keys and sends the control words to the TP in an encrypted form. Note that the SM and TP establish the control words encryption key to provide confidentiality when they are sent from the SM to TP.

Step 7. The TP descrambles the encrypted video stream using the control words and sends the cleared video stream to the user's premises, such as their television.

Note that the flows from the fifth to the seventh steps are the same as in a general CAS operation [2]-[5] except that the control words are delivered from the SM to TP.

1. Trusted Authority

The TA has connections to all MSO headends as a designated third party and has the following principle functions. The TA issues identification information of the SM and TP to each of their manufacturers. This identification information is typically included in the Common Name field of X.509 certificates The TA also performs a DCAS host device authentication process based on the identification information received from the SM via the AP in the MSO DCAS headend.

2. MSO DCAS Headend

The MSO DCAS headend consists of an AP, PS, CAS, and download servers, as shown in Fig. 3. The AP server is a proxy of the TA and performs a mutual authentication process through a DCAS network protocol [7], [8] between the AP and SM. Note that while the AP performs the authentication

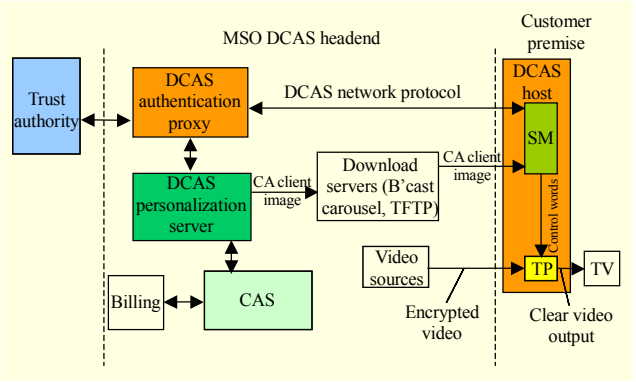

Fig. 3. Block diagram of MSO DCAS headend and DCAS host.

process, it sends identification information of the DCAS host devices to the TA, and the TA relays the authentication results to the AP. The PS contains the source of all CA client images for distribution, downloading, and management. If the mutual authentication process between the AP and SM is successfully performed, the DCAS PS sends the CA client image to the SM through the download servers after encrypting it with the CA client image encryption key. The CAS generates entitlements keys and control words for pay programs and securely delivers them through the EMM and ECM to the CA client application in the SM [2]-[5].

3. DCAS Host

The DCAS host is a two-way digital cable set-top supporting not only the functions of OpenCable [10] and DOCSIS [11], but also DCAS security functions [7], [8]. Since an SM and a TP have to protect security parameters stored in their memory from physical security attacks [12], they should be designed to meet FIPS 140-2 Level 3 [13] or CC EAL Level 5+ [14] as specified in [6] and [15].

The SM is a highly secure chip that performs a DCAS network protocol [7], [8] for mutual authentication with the AP. The type of SM device can be either an embedded chip on a set-top motherboard or removable device [7]. The SM also securely stores security parameters in its memory and operates SM client applications to decrypt control words and send them to the TP. Note that if the SM does not have any CA client application in its memory, we call it a virgin SM. Otherwise, the SM is called a non-virgin SM.

The TP is a descrambler containing multiple descrambling algorithms. The TP receives control words from the SM and uses them for descrambling the encrypted video streams [15]. Note that the TP has no direct communication channel to the MSO DCAS headend and TA. Therefore, the TP has to receive any necessary data from the TA or MSO DCAS headend through the SM.

III. Requirements for DCAS SM and TP Security Protocol

In this section, we describe the functional and security requirements that should be considered when designing the DCAS SM and TP security protocol.

1. Functional Requirements

A. Support SM-TP Pairing

Assume that there is a user who has a removable SM and subscribes to pay programs after a CA client application has loaded into an SM. Then, any user who has a removable SM can watch the subscribed pay programs by inserting the removable SM into any DCAS host connected in the same MSO network. In other words, a group of users can watch pay programs by sharing the removable SM if just one member of the group subscribes to the pay programs. However, this service leak caused by SM sharing cannot be acceptable to the MSO from a subscriber management point of view.

To prevent the service leak described above, the DCAS requires that the DCAS SM and TP security protocol has a SM-TP pairing function [8]. This function is the management of an SM and TP pair from the time when the DCAS host is first connected to the MSO network after it is released from the set-top manufacturer. Once the SM and TP are paired, the TA permanently maintains the pair information and never changes it. Note that the DCAS also requires that the SM-TP pairing scheme should be applied to not only a removable SM, but also an embedded SM to prevent an SM replacement attack that might be made by high-level attackers. Here, the high level attackers are a very well-financed and experienced engineering organization up to and including a CAS vendor that might benefit if the system were to be broken.

B. Control Words Encryption Key (CWEK) Management between SM and TP

If the control words are delivered in plaintext from an SM to a TP, an adversary can possibly watch pay programs by using the disclosed control words for decrypting the scrambled video streams. Therefore, the SM must provide confidentiality for the control words by encrypting them with the control words encryption key (CWEK).

Since a successful CWEK establishment between an SM and a TP means that the SM believes the TP as its correct pair, or vice versa, CWEK establishment must not be performed before the TA confirms that the SM and TP are correctly paired. For this reason, the DCAS SM and TP security protocol should make sure that the SM and TP establish a CWEK after they have received confirmation that the SM and TP are correctly paired from the TA.

2. Security Requirements

Here, we describe possible security threats to the DCAS SM and TP security protocol. Additionally, we list some desirable security attributes [16] that are necessary for the CWEK establishment protocol.

A. Security Threats to DCAS SM and TP security protocol

Since an SM_ID and a TP_ID are inserted into the Common Name field of an X.509 certificate without confidentiality, an attacker can possibly obtain these values. After an attacker acquires an SM_ID and a TP_ID, he may attempt to authorize a forged SM by sending the acquired SM_ID and TP_ID to the TA. Therefore, the TA must authenticate the SM_ID and TP_ID before it starts the SM-TP pairing validation process.

Since there is no direct communication channel between a TP and TA, a TP can receive the results of an SM-TP pairing validation check from the TA only through the SM. In this circumstance, it is possible that a forged SM can try to establish a CWEK with the TP even though the SM does not actually receive a notification that the SM and TP were correctly paired from the TA. Therefore, the DCAS SM and TP security protocol must ensure the TP that the message from the TA has not been manipulated by the SM.

B. Security Attributes for Key Establishment Protocol

Known Key Security. An adversary must not be allowed to generate a CWEK using knowledge that can be obtained from some other CWEKs. For this type of security, the DCAS SM and TP security protocol must generate a CWEK that is unique for all sessions.

Perfect Forward Secrecy. Even though long-term private keys are compromised, an adversary must not be allowed to find previous CWEKs established between entities. For this type of security, the DCAS SM and TP security protocol must use a random value for CWEK generation and guarantee that the random value cannot be extracted from the CWEK.

Key-Compromised Impersonation. If a long-term private key of an entity, say A, is compromised, then clearly an adversary that knows this value can impersonate A to other entities. However, this loss must not allow an adversary to impersonate other entities to A.

Unknown Key-Share Attack. From this unknown key-share attack, an adversary C can make one entity, say A, believe that the CWEK is shared with C when it is in fact shared with a different entity, B. To counter this attack, the DCAS SM and TP security protocol must provide a process of key confirmation between entities.

Key Control. Neither entity should be able to force the CWEK to a preselected value.

IV. Proposed DCAS SM and TP Security Protocol

In this section, the processes of the proposed DCAS SM and TP security protocol are described in detail. In the first subsection, we define the notations and symbols used in the rest of this paper. We also describe some assumptions and DCAS network protocol messages that are needed to understand the process of the proposed DCAS SM and TP security protocol. In the second subsection, we describe the detailed procedures of the proposed DCAS SM and TP security protocol.

1. Preliminaries

A. Notations

In the rest of this paper, the following notations are used for the proposed DCAS SM and TP security protocol.

- Pub(X): RSA public key of X
- Prv(X): RSA private key of X
- E(k,m): encrypt a message 'm' with key 'k'. RSAES-OAEP is used to encrypt a message.
- S(k,m): digital signature for a message 'm' with signing key 'k'. RSASSA-PSS is used for message signing.
- H(m): SHA-256 hashing for a message 'm'
- HMAC(k,m): HMAC-SHA1 for a message 'm' with key 'k'
- X||Y: concatenation of 'X' and 'Y'
- Cert(X): X.509 certificate of 'X'
- PRF(X) : pseudo random function using a method described in FIPS 186-2 appendix 3.3. Here 'X' is a seed value.
- $X_{msb(Y)}$: 'Y' bits from MSB of 'X'

B. Symbol Definitions

In the rest of this paper, the following symbols are used for the proposed DCAS SM and TP security protocol.

- TP_ID: value of identification of TP
- SM_ID: value of identification of SM
- KeyPairingID: value of concatenation with SM_ID and TP_ID, that is, SM_ID||TP_ID
- CWEK: control words encryption key used to encrypt control words
- KPK: key pairing key. TA generates the KPK if KeyPairingID is valid
- HMAC_KEY: HMAC secret key. SM uses HMAC_KEY to generate a HMAC value for the message including control words.
- RAND: 320 bit random number generated with algorithm 1 defined in FIPS 186-2 appendix 3.3
- K_i: pre-shared key having the size of 128 bits [8]. TA uniquely assigns three K_i to each SM.

C. DCAS Network Protocol Messages

The DCAS network protocol [7], [8] provides a SecurityAnnounce message and DCASDownload message that can trigger an SM to enter into a download for a new CA client image. Upon SM boot-up, the SM bootloader sets filters that instruct the SM's interface manager to forward the first received instance of a SecurityAnnounce or DCASDownload message to the SM bootloader.

A SecurityAnnounce message is broadcasted from an AP to all SMs connected to the MSO network. The purpose of this message is providing versions of the current CA client image and MSO network information to the SM. If a change in SM client version and MSO network information is indicated for this SM, the SM continues the authentication and download processes as defined in [7], [8]. Note that this message also includes an AP X.509 certificate.

The function of a download message is almost the same as that of a SecurityAnnounce message except that the AP multicasts or unicasts this message. The purpose of this message is triggering a certain SM or group of SMs to start the authentication and download processes.

D. Assumptions

An SM has its X.509 certificate signed by the TA Root private key, RSA private key, TA root X.509 certificate, and three K_i (K_1, K_2, K_3) in its secure memory. All security parameters of the SM are injected by the SM manufacturer before it is shipped to the DCAS host manufacturer. The Common Name field in an SM X.509 certificate includes the SM_ID.

A TP has its X.509 certificate signed by the TA root private key, RSA private key, and TA root X.509 certificate, in its secure memory. All security parameters of the TP are injected by the TP manufacturer before it is shipped to the DCAS host manufacturer. The Common Name field in a TP X.509 certificate includes the TP_ID.

A TA issues AP X.509 certificates signed by the TA root private key to all APs. All APs also have a TA root X.509 certificate.

2. Protocol Description

As shown in Fig. 4, the proposed DCAS SM and TP security

protocol consists of three sequences of phases.

Initialization. The SM monitors the triggering conditions of the protocol and starts the protocol procedures if it meets one of the triggering conditions. Then, the SM and TP exchange their own X.509 certificates with each other.

Pairing. The TA validates the SM-TP pairing status based on the value of the SM_ID and TP_ID delivered from the SM. If the TA can judge that the SM-TP pairing status is valid, it generates a KPK and sends the KPK to the SM through the AP.

CWEK Generation. The SM and TP establish a CWEK and HMAC_KEY in this phase. For this, the SM sends a KPK for CWEK generation and a random value for HMAC_KEY generation to the TP. The key establishment confirmation process also takes place by comparing the hashed value of each key.

In the phases of pairing and CWEK generation, we use RSA digital signatures for four messages at SM and TP sides as shown in Fig. 4. Certainly, digital signature is an expensive solution as regards computational complexity. However, commonly-available off-the-shelf commercial security chip products, including the Infinieon [17] SLE-66, SLE-88, SLD-9630, and STM ST-19, implement hardware acceleration for calculation of signature and secure hashing. Therefore, there is no problem utilizing digital signature for four messages during the proposed DCAS SM and TP security protocol from the operation speed point of view.

A. Initialization

The detailed procedures of the initialization phase are as follows.

Step 1. The SM sends its X.509 certificate to the TP through a TPCertReq message if the SM meets one of the following conditions.

- An SM is powered up or reset.
- A virgin SM receives a SecurityAnnounce message from an AP.
- An SM receives a SecurityAnnounce message from an AP right after the SM moves to another MSO network, or
- A non-virgin SM is requested to update CA client images from an AP via a DCASDownload message.

Note that an SM can achieve the AP X.509 certificate from

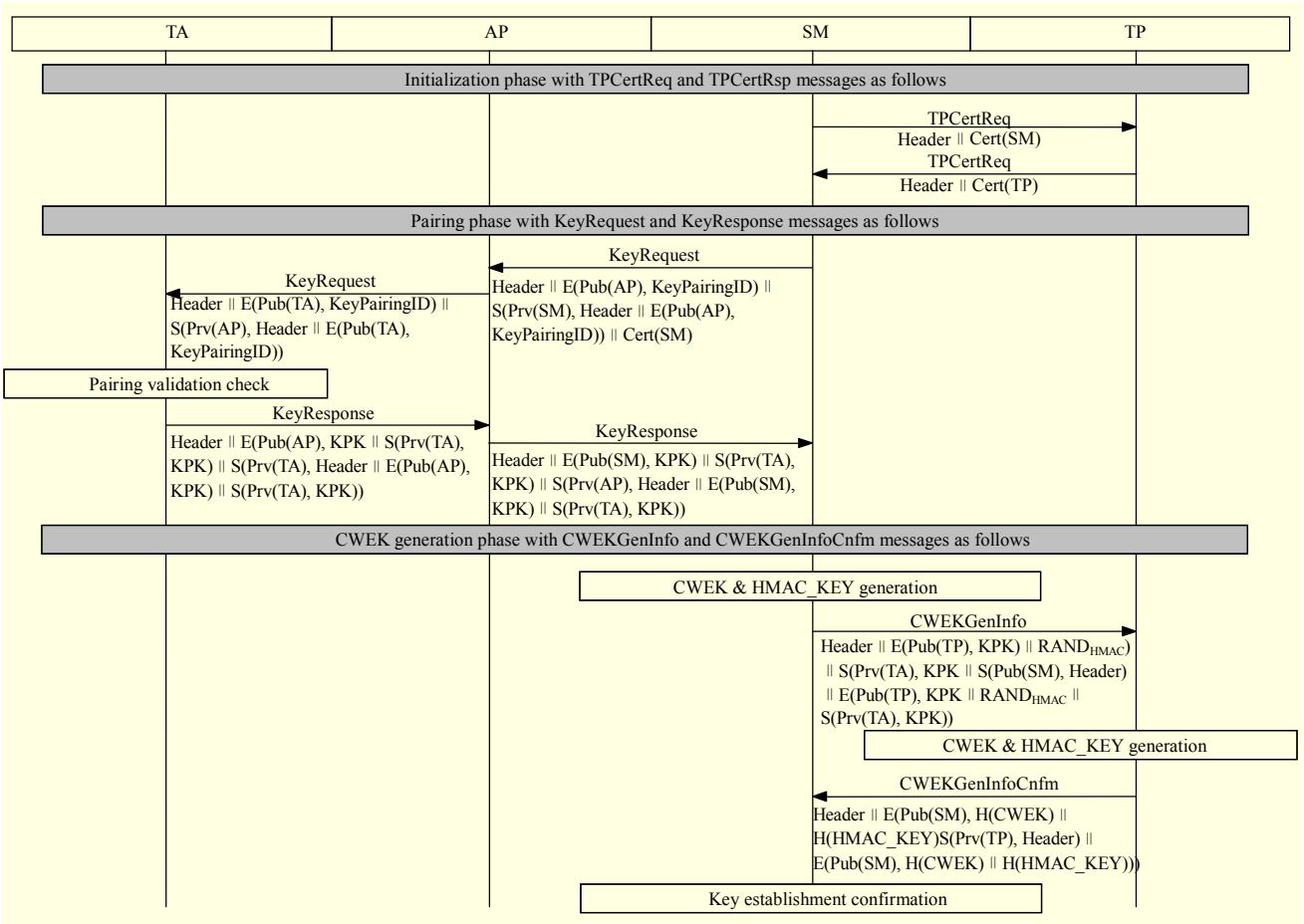

Fig. 4. Sequence diagram of proposed DCAS SM and TP security protocol.

the SecurityAnnounce message from the AP as mentioned earlier in this section.

Step 2. Right after the TP receives a TPCertReq message from the SM, the TP verifies the signature of the SM X.509 certificate using the public key of the TA. Only if the TP can successfully verify the SM X.509 certificate, the TP stores the SM_ID and RSA public key of the SM extracted from the SM X.509 certificate in the secure area of nonvolatile memory. Otherwise, the TP terminates this protocol. Finally, the TP sends a TPCertRsp message including its X.509 certificate to the SM.

Step 3. Right after the SM receives the TPCertRsp message from the TP, the SM verifies the signature of the TP X.509 certificate using the public key of the TA. Only if the SM can successfully verify the TP X.509 certificate, the SM stores the TP_ID and RSA public key of the TP extracted from the TP X.509 certificate in the secure area of nonvolatile memory and goes to the next phase. Otherwise, the SM terminates this protocol.

B. Pairing

In this phase, the TA performs an SM-TP pairing validation check with the KeyPairingID received from the SM and TP. If the TA can judge that the validity of the KeyPairingID is correct based on the pairing state information (PSI) shown in Table 1, it generates a KPK and sends this key back to the SM.

The PSI is maintained by the TA based on the pairing state diagram shown in Fig. 5. As the pairing state diagram shows, we classify the PSI into three types. The first type is Virgin (0x00). The TA sets the PSI type as Virgin (0x00) when it issues identification information of the SM and TP, and there have been no SM-TP pairing validation check requests from MSO DCAS headend for them. The second type is Auth/Paired (0x01). The TA changes the PSI type from Virgin (0x00) or Paired Only (0x10) to Auth/Paired (0x01) when the DCAS host devices in either a Virgin (0x00) state or Paired Only (0x10) state are connected to the MSO network and have passed the SM-TP pairing validation check in the TA. The third type is Paired Only (0x10). The TA sets the PSI type to Paired Only (0x10) when the DCAS host devices in an Auth/Paired (0x01) state leave the MSO network.

The generation method of the security parameters is defined in Table 2. The following are detailed descriptions of the pairing phase.

Step 1. The SM sends a KeyPairingID and SM X.509 certificate through the KeyRequest message to the AP. As shown in Fig. 4, the KeyPairingID is encrypted with the public key of the AP, and the content is signed with the private key of the SM. Note that an SM X.509 certificate is added to the tail of this message without encryption.

Table 1. Pairing state table.

SM state	TP state	Pairing state information
0x00	0x00	Virgin
0x01	0x01	Auth/Paired
0x10	0x10	Paired Only

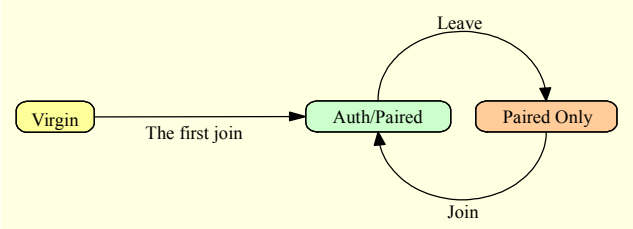

Fig. 5. Pairing state diagram.

Table 2. Generation method of KeyPairingID and KPK.

Parameters	Generation method
KeyPairingID	SM_ID ‖ TP_ID
KPK	PRF(H (Ki_1 ‖ Ki_2 ‖ Ki_3 ‖ SM_ID ‖ TP_ID ‖ RAND))$_{msb(160)}$

Step 2. The AP receives the KeyRequest message from the SM and verifies the digital signature of the message. The AP also stores the SM X.509 certificate for future communication with the SM. If the AP fails to verify the digital signature of the message, it discards this message and terminates the protocol. Otherwise, the AP decrypts the KeyPairingID and generates a KeyRequest message including E(Pub(TA), KeyPairingID) instead of E(Pub(AP), KeyPairingID). After that, the AP sends this KeyRequest message to the TA. At this time, an SM X.509 certificate is not attached to this message.

Step 3. The TA receives the KeyRequest message from the AP and verifies the digital signature of the message. If the TA fails to verify the digital signature of the message, it discards this message and terminates the protocol. Otherwise, the TA decrypts the KeyPairingID and starts to validate the KeyPairingID based on the PSI as shown in Fig. 6. The identification validation procedures at a TA shown in Fig. 6 are carried out as follows. After the TA receives a KeyRequest message from the AP, it extracts the SM_ID and TP_ID from the KeyPairingID. Then, the TA searches the PSI regarding the SM_ID and TP_ID from its own database. If the TA fails to find the record in its database regarding the SM_ID and TP_ID, it terminates the protocol. If the PSI for the SM_ID and TP_ID is equal to Virgin (0x00), the TA changes the PSI from Virgin (0x00) to Auth/Paired (0x01) and judges that the SM_ID and TP_ID have successfully passed the SM-TP pairing validation

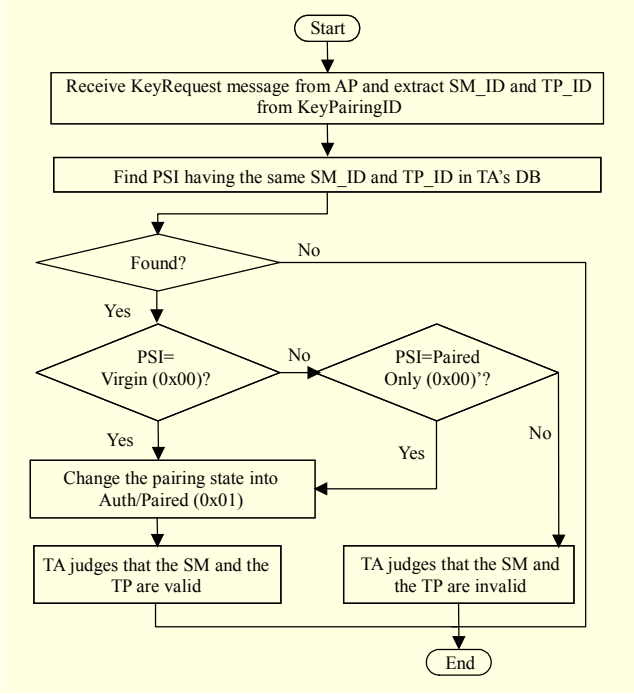

Fig. 6. Identification validation procedures at TA.

check. If the PSI for the SM_ID and TP_ID is not equal to Virgin (0x00) but the same as Paired Only (0x10), the TA changes the PSI from Paired Only (0x10) to Auth/Paired (0x01) and judges that the SM_ID and TP_ID have successfully passed the SM-TP pairing validation check. For all other cases, the TA judges that the SM_ID and TP_ID have failed to pass the SM-TP pairing validation check.

After finishing the SM-TP pairing validation check, the TA generates a KeyResponse message including the encrypted KPK and signed KPK. At this time, the TA generates a KPK, which is uniquely assigned to the SM, using the generation method shown in Table 2. Otherwise, the TA sets all bytes of the KPK as '0xff" to indicate that the SM_ID and TP_ID pairing validation result is a failure. Finally, the TA sends the KeyResponse message to the AP.

Step 4. The AP receives the KeyResponse message from the TP and verifies the digital signature of the message. If the AP fails to verify the digital signature of the message, it discards this message and terminates the protocol. Otherwise, the AP generates the KeyResponse message including E(Pub(SM), KPK) instead of E(Pub(AP), KPK) and sends this message to the SM. At this time, the signed value of the KPK, for example, S(Prv(TA),KPK), is inserted into the message content as it is received from the TA.

C. CWEK Generation

In this phase, the SM and TP establish a CWEK that will be used for encrypting the control words. For this, the SM sends the KPK to the TP first, and the SM and TP establish a CWEK using the shared KPK. As described in the pairing phase, a KPK is generated by the TA and delivered to the SM in encrypted form using an RSA public key.

Table 3. Generation method of CWEK and HMAC_KEY.

Parameters	Generation method
CWEK	H(KPK ‖SM_ID‖ TP_ID)msb(128)
HMAC_KEY	H($RAND_{HMAC}$‖ SM_ID‖TP_ID)msb(160)

The generation method of the security parameters are defined in Table 3. The following are detailed descriptions of the CWEK generation phase.

Step 1. The SM receives the KeyResponse message from the AP and verifies the digital signature of the message. If the SM fails to verify the digital signature of the message, it discards this message and terminates the protocol. Otherwise, the SM decrypts the KPK and verifies S(Prv(TA),KPK) with the decrypted value of the KPK using the public key of the TA. Note that the TP already has the TA root certificate in its memory as we described in section IV. If the SM fails to verify the digital signature of the KPK, it also discards the KeyResponse message and terminates the protocol. Otherwise, the SM generates the CWEK and HMAC_KEY as shown in Table 3. Then, the SM also generates a CWEKGenInfo message including a KPK and $RAND_{HMAC}$ except when all bytes of the KPK are '0xff". If all bytes of KPK are '0xff", it terminates this DCAS SM and TP security protocol since the value of '0xff" means that the SM_ID and TP_ID pairing validation result is a failure. Finally, the SM sends the CWEKGenInfo message to the TP. Note that the value of S(Prv(TA),KPK) is inserted into the message content as it is received from the AP through a KeyResponse message.

Step 2. The TP receives the CWEKGenInfo message from the SM and verifies the digital signature of the message. If the TP fails to verify the digital signature of the message, it discards this message and terminates the protocol. Otherwise, the TP decrypts the KPK and $RAND_{HMAC}$. After that, the TP verifies S(Prv(TA),KPK) with the decrypted value of KPK using the public key of the TA. Note that the TP already has the TA root certificate in its memory as we described in section IV. If the TP fails to verify the digital signature of the KPK, it also discards the CWEKGenInfo message and terminates the protocol. Otherwise, the TP generates a CWEK and HMAC_KEY with the KPK and $RAND_{HMAC}$ as shown in Table 3. Finally, the TP generates a CWEKGenInfoCnfm message including H(CWEK)‖H(HMAC_KEY) and sends this message to the SM.

Step 3. The SM receives the CWEKGenInfoCnfm message from the TP and verifies the digital signature of the message. If the SM fails to verify the digital signature of the message, it discards this message and terminates the protocol. Otherwise, the SM decrypts H(CWEK)||H(HMAC_KEY) from the CWEKGenInfoCnfm message and generates the hashed value with the CWEK and HMAC_KEY that were generated in Step 1. Finally, the SM compares the hashed values received from the TP with those generated by the SM itself. If the two hashed values are mismatched, the SM terminates the protocol.

Step 4. After the SM and TP share the same CWEK and HMAC_KEY, the CWEK is used for encrypting the control words with the symmetric encryption algorithm, and HMAC_KEY is used for applying the HMAC algorithm to the messages, which includes control words, for the purpose of message authentication. Note that the rekeying period to send updated control words from the CAS headend to the SM is very short. For example, the rekeying period can be from 1 to 20 seconds. [2]-[5]. Therefore, the SM also has to deliver control words from the CAS headend to the TP whenever SM receives the updated control words from the headend. In this circumstance, the primary decision criteria for a message authentication algorithm should be computational overload, not security. As a result, we select HMAC algorithm instead of digital signature algorithm for the practical reason of reducing computational overload.

V. Analysis

1. Functional Requirements

The proposed protocol satisfies the following functional requirements.

Support SM-TP Pairing. The TA validates the SM-TP pairing based on the values of the SM_ID and TP_ID and the pairing state table. Once the SM_ID and TP_ID are registered and their pairing state becomes Auth/Paired from Virgin, the pair of SM_ID and TP_ID is never changed and is maintained permanently by the TA. Therefore, there is no way to receive validation from the TA if an attacker sends unmatched pair information after the SM_ID and TP_ID are set to Auth/Paired by the TA.

CWEK Establishment. The SM and TP establish a CWEK by utilizing the KPK, SM_ID, and TP_ID in the CWEK generation phase. Since the SM and TP receive the same KPK in the pairing phase and exchange their own ID values in the initialization phase, they can generate the same CWEK at the same time.

To generate a CWEK, the pairing phase must be performed prior to the CWEK generation phase since the KPK is necessary to generate a CWEK. Moreover, the TA confidentially issues a valid KPK to an authenticated SM only after the TA receives a correct pair of SM_ID and TP_ID. Therefore, we can counter an attack that tries to generate a CWEK without the SM and TP pairing validation process at the TA.

2. Security Requirement

The proposed protocol satisfies the following security requirements.

Disclosed SM_ID and TP_ID Threat. To counter this threat, the proposed protocol signs the KeyRequest message that is sent from the SM to the AP, and from the AP to the TA, with an RSA private key using RSASSA-PSS. Moreover, the chips of the SM and TP are designed to meet the requirements of FIPS 140-2 Level 3 or EAL Level 5+ to protect identification information stored in the memory from a physical security attack. Therefore, we can authenticate the values of the SM_ID and TP_ID as long as attackers cannot hack the PSASSA-PSS authentication algorithm or physical security of FIPS 140-2 Level 3 or EAL Level 5+.

Indirect Connection between TA and TP threat. To counter this threat, the TA sends the KPK to the SM and TP along with its signature signed by the TA. Since the TP generates a CWEK with the KPK only if it can verify the KPK signature with the public key of the TA, we can prevent the TP from receiving a forged KPK from the SM.

Known Key Security. The SM and TP generate a CWEK with the KPK received from the TA at every session, and the TA guarantees that the KPK is uniquely assigned to every SM. Therefore, we can be sure that there is a unique CWEK for every session.

Perfect Forward Secrecy. It is clear that the KPK used for CWEK generation is a random value. We can also guarantee that the KPK cannot be extracted from a CWEK as long as the one-way property of the SHA-256 hash function is not broken.

Key-Compromised Impersonation. Suppose the SM RSA private key is disclosed. An attacker who knows this value can clearly impersonate the SM. However, an attacker must know the RSA private key of the TP for a successful impersonation of an attacker's TP to an SM. However, this is impossible unless an attacker also compromises the RSA private key of the TP. Therefore, we can counter this 'key-compromised impersonation' threat as long as the other entity's long-term private key is not compromised.

Unknown Key-Share Attack. In the CWEK generation phase, the SM does not use the generated CWEK for control words encryption before it can confirm that the SM and TP have established the same CWEK through the key establishment confirmation process. Therefore, we can counter this unknown key-share attack with the key establishment confirmation

process in the pairing phase.

Key Control. Since the SM and TP generate a CWEK with the KPK after they receive the KPK from the TA and verify its signature with the public key of the TA, it is also clearly impossible that the SM or TP can force a CWEK to a preselected value.

VI. Conclusion

In this paper, we proposed a DCAS SM and TP security protocol, which is used for generating a control words encryption key and managing SM and TP pairing. It is secure in the sense that it not only counters a disclosed SM ID and TP ID threat and an indirect connection between a TA and TP threat, but it also meets some desirable security attributes such as known key secrecy, perfect forward secrecy, key compromised impersonation, unknown key-share, and key control. Since the proposed protocol provides additional security functions that the DCAS network protocol cannot support, we can accomplish a more secure DCAS by utilizing the proposed protocol with the DCAS network protocol.

References

[1] OpenCable™ Specifications: CableCARD Interface 2.0 Specification, CableLabs, OC-SP-CCIF2.0-I19-090904, 2009.

[2] T. Jiang, S. Zheng, and B. Liu, "Key Distribution Based on Hierarchical Access Control for Conditional Access System in DTV Broadcast," *IEEE Trans. Consum. Electron.*, vol. 50, 2004, pp. 225-230.

[3] B.M. Macq and J.J. Quisquater, "Cryptology for Digital TV Broadcasting," *Proc. IEEE*, 1995, pp. 944-957.

[4] EBU Project Group, "Functional Model of a Conditional Access System," *EBU Technical Review*, 1995, pp. 64-77.

[5] F.K. Tu, C.S. Laih, and H.H. Tung, "On Key Distribution Management for Conditional Access System Onpay-TV System," *IEEE Trans. Consum. Electron.*, vol. 45, 1999, pp. 151-158.

[6] DCAS Host License Agreement, CableLabs, http://www.opencable.com/downloads/DCAS_New.pdf.

[7] W.L. Helms, J.B. Carlucci, and J.K. Schnitzer, "Downloadable Security and Protection Methods and Apparatus," in Patent Publication Number: 20080098212; Application Number: 2006-584208, US, 2006.

[8] Y. Jeong et al., "A Novel Protocol for Downloadable CAS," *IEEE Trans. Consum. Electron.*, vol. 54, 2008, pp. 1236-1243.

[9] M. Borza and A. Hawtin, "The Future of Open Cable Systems: Conditional Access Migrates to DCAS," *Information Quarterly*, vol. 7, 2008, pp. 60-63.

[10] OpenCable™ Specifications: OpenCable Host Device 2.1 Core Functional Requirements, CableLabs, OC-SP-HOST2.1-CFR-I08-090508, 2009.

[11] Data-Over-Cable Service Interface Specifications: Radio Frequency Interface Specification, CableLabs, CM-SP-RFIv2.0-C02-090422, 2009.

[12] S.H. Weingart, "Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses," *Lecture Notes in Computer Science*, 2000, pp. 302-317.

[13] Security Requirement for Cryptographic Modules: NIST, FIPS PUB 140-2, 2001.

[14] "Common Criteria for Information Technology Security Evaluation," ver. 2.1, *Technical Report*, http://www.commoncriteria.org/docs/index.html, 1999.

[15] Next Generation Network Architecture Plan: NGNA LLC, 2004.

[16] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*, Springer: Berlin, 2003.

[17] Infineon, http://www.infineon.com.

Han-Seung Koo received the BS, MS, and PhD degrees in electronic engineering from Chungnam National University, Korea, in 1999, 2001, and 2008, respectively. Since 2001, he has been with ETRI, Daejeon, Korea. He was a member of OCAP middleware test focus team and DOCSIS 3.0 security specification development focus team in 2002 and 2005 to 2006, respectively, in CableLabs, Colorado, USA. Currently, he also serves as the rapporteur for ITU-T SG9 Q.3 in Geneva, Switzerland. His current interest includes security protocols, key management algorithms, and conditional access system architecture for digital broadcasting systems.

O-Hyung Kwon received the BS, MS, and PhD degrees in electrical engineering from Sogang University, Seoul, Korea, in 1981, 1983, and 2004, respectively. In 1983, he joined ETRI, Daejeon, Korea, where he has been working on digital cable TV broadcasting system technologies. His main research interests include digital cable TV broadcasting systems, digital cable communication systems, digital watermarking technologies, and conditional access systems.

Soo In Lee received the MS and PhD degrees in electronics engineering from Kyungpook National University, Daegu, Korea, in 1989 and 1996, respectively. In 1990, he joined ETRI, Daejeon, Korea, where he has been working on broadcasting system technologies. Currently, he serves as the director for Broadcasting System Research Group. His research interests include terrestrial DTV and DMB systems, digital CATV systems, and 3DTV systems.