Design of a secure digital recording protection system with network connected devices

Hyoungshick Kim
Department of Software
Sungkyunkwan University
Suwon, Republic of Korea
Email: hyoung@skku.edu

***Abstract*—We present a new recording protection system for network connected devices that belong to a user's personal network. To avoid illegal redistribution of the recorded contents through Internet, a proper content protection technology is required. However, conventional recording protection systems all lack sharing of the recorded contents without an online connection to a remote content service. Our recording protection system is designed to avoid complicated membership controls which are generally required in the existing solutions. To show the feasibility of the proposed recording protection system, we implemented a prototype with popularly used encryption algorithms (AES and RSA-OAEP). The experimental results of our prototype demonstrate that the execution time overheads incurred by the proposed system seem acceptable enough.**

I. Introduction

As network technologies were becoming more widespread, building a small and private network with smartphone, laptop, desktop and IPTV has become essential to users.

Many industrial standard organizations [1] such as Digital Living Network Alliance (DLNA) [2] and Universal Plug and Play (UPnP) forum [3] have made significant efforts to develop practical standard technologies for sharing data over a home network. Those home networking technologies seem to promise a major shift in our ways of consuming digital contents. For example, an IPTV subscriber can record her favorite television programs on PC through Internet for later viewing. In such a scenario, the most challenging issue is to restrict the redistribution of recorded contents within a limited set of *authorized* devices (e.g., the subscriber's personal devices) only. Without a proper copy protection mechanism for the recorded contents, content providers do not allow users to record their high-quality premium contents on either network attached storage or PC because the recorded contents can freely be controlled and distributed by users without any restriction. On behalf of movie studios and cable operators, the Digital Video Broadcasting Project (DVB) and Cable Television Laboratories, Inc. (CableLabs) developed several technical guidelines to ensure that digital content players and recorders must meet such content providers' security requirements [4]. However, those strong restrictions make take away consumers' rights and abilities to share lawfully acquired contents even within their personal devices.

In this paper, we address how to securely share the recorded contents with *authorized* network connected devices belong to a user's personal network without violating the contents providers' security requirements. We consider a passive attacker (e.g., a potentially malicious user) who can access the contents recorded by a recording device. The attacker's goal is to illegally share the recorded contents with unauthorized playback devices.

We propose a new design of recording protection systems based on public-key cryptography. Unlike the existing solutions (e.g., [5]) requiring complicated membership controls, the proposed system can simply support the secure data sharing of recorded contents between *authorized* devices.

To show the feasibility of the proposed recording protection system, we analyzed the execution time overheads of the proposed system through a few experiments. The experimental results demonstrate that its execution time overheads are marginal compared with the total execution cost (within 4% and 8% of the total execution time for the *record* and *playback* operations, respectively, when the recorded file size is greater than or equal to 40MB).

The rest of this paper is organized as follows. In Section II, we briefly introduce the architecture of conventional recording protection systems and discuss its challenging issues. In Section III, we outline the design of the proposed recording protection system and demonstrate its advantages over the conventional systems. In Section IV, we evaluate the performance of the proposed recording protection system. Finally, we conclude in Section V.

II. Security issues in recording systems

Traditional recording protection systems are generally designed for a single device such as a Digital Video Recorder (DVR) or a Personal Video Recorder (PVR) that records the received content in a digital format to data storage within a device. In the middle of recording, a recorded content is stored in encrypted format by a content protection technology to prevent the redistribution of the recorded content through the Internet.

In general, in a proprietary recording protection system, the content recording operation is processed as follows: (1) An encoded content c is encrypted with a content encryption key ck generated by the *key manager* in a recording device while being recorded. (2) The content encryption key ck is encrypted with the recording device's own device key dk again. The device key dk must be securely embedded in each recording device. (3) The encrypted content $E_{ck}(c)$ and the encrypted content encryption key $E_{dk}(ck)$ are stored in data storage. (4) When a user wants to play the (encrypted) content $E_{ck}(c)$ in his or her playback device, $E_{dk}(ck)$ is first decrypted with the recording device's key dk, and $E_{ck}(c)$ is then decrypted with the obtained content key ck at the content key decryption module in the recording device. We note that the encrypted content encryption key $E_{dk}(ck)$ can be decrypted only by the same recording device that previously recorded the content c because each recording device has its own unique device key dk individually [6]. (5) The decrypted content c is transferred to the playback device over a digital interface (e.g., HDMI, DVI or VGA). Figure 1 shows the overall architecture of a conventional recording protection system. In this figure, the dashed lines represent the communication channels within each device, and the solid lines represent the communication channels between network connected devices.

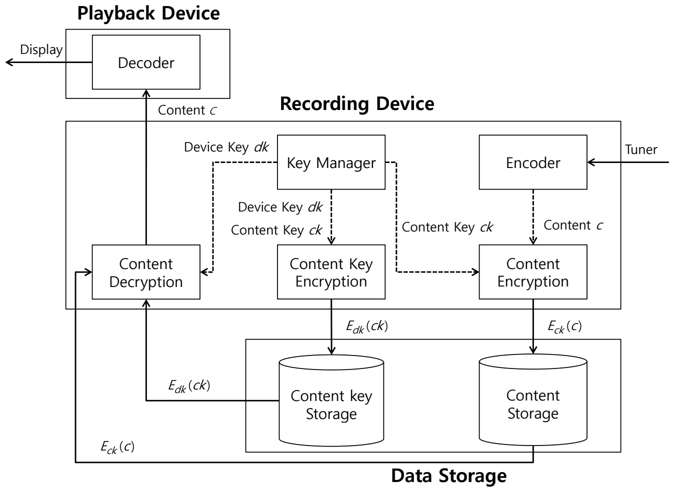


Fig. 1. Overview of a conventional recording protection system.

Here, we assume that the transferred content c for digital output should be securely protected. When the playback device and the recording device are incorporated into a single device, this assumption is valid if we neglect sophisticated attackers who can monitor input/output buses on a single physical device. However, the playback device or the data storage could be deployed as independent devices. In such a situation (i.e., when a playback device is deployed as an independent device), the digital interface between the playback device and the recording device must be protected by using a proper link protection technology such as DTCP-IP [7].

In practice, many content service providers allow users to share some paid contents between their private devices for enhancing usability and user experience. In order to enable sharing paid digital contents between multiple playback devices, the notion of *authorized domain* was introduced [5], [8]. The basic mechanism is to define a set of devices as an *authorized domain*, and to allow users to share their contents between the devices belong to the domain. In a recording system supporting the domain concept, protected contents can be generated with a common domain key which is shared between multiple playback devices within the same domain rather than a device key that is managed by a single device (i.e., DVR) alone. Therefore, the encrypted contents with the domain key can effectively be shared between all the devices that belong to the domain. Most commercial content protection solutions have already provided such a domain management functionality.

For recording protection systems, however, conventional copy protection solutions supporting the authorized domain functionality have some security weaknesses without a permanent Internet connection.

To construct a user's authorized domain, the most challenging issue is to deploy a secure and usable key management scheme for sharing a domain key between recording and playback devices. The overall security of a domain-based content protection system depends on the secrecy of the domain key. In most cases, domain management is typically governed by an Internet-based remote domain server because content service providers do not want to lose their rights of controlling domain membership.

An important task of the domain management service is to revoke illegal devices that have been compromised and cloned. Unsurprisingly, a domain key could be exposed to non-authorized devices unless a proper device revocation scheme is deployed. Maintaining, however, up-to-date revocation status information such as Certificate Revocation Lists (CRLs) is difficult in local devices that are incapable of a permanent online connection. In addition, the cost of updating a new domain key for all domain devices is also very expensive when some domain members are frequently changed over time. As devices join or leave a domain, the domain key need to be promptly updated for backward and forward secrecy properties. Since most key distribution schemes for dynamic membership are complicated and require a high message complexity, domain-based content protection technologies may not directly be applied to a recording system. In this context, synchronization issue of the updated domain keys between authorized devices should also be considered because some devices might be essentially turned off or temporally disconnected.

III. Proposed recording protection system

We present a recording system to *securely* share the contents recorded by a recording device with *authorized* network connected devices (e.g., the devices belong to the recording device owner's personal network) only.

A recording system is generally composed of two types of devices: (1) recording device and (2) playback device. Interestingly, the data storage can be regarded as a *insecure* communication channel between those devices. From this perspective, a conventional recording protection system can be regarded as a cryptographic protocol to securely protect the files on the data storage with a secret (domain) key that is commonly shared between those devices. This view also gives us a key insight into how to securely design a recording protection system.

We suggest using a *public key encryption* to provide the flexibility needed to add new playback devices easily without complicated membership operations. When a *copy-protected* content is recorded, the content encryption key ck is encrypted with previously designated playback devices' public keys instead of the recording device's own secret key or the domain key in an authorized domain. Therefore, the stored content encryption key ck can be only decrypted by designated playback devices with the corresponding private keys. That is, in the proposed recording protection system, the content recording operation is processed as follows: (1) An encoded content c is encrypted with a content encryption key ck generated by the *key manager* in a recording device while being recorded. (2) The content encryption key ck is encrypted with an *authorized* playback device's public keys pk where the playback device holds a pair of keys (pk, sk). The playback device's secret key sk must be securely embedded in each playback device. (3) The encrypted content $E_{ck}(c)$ and the encrypted content encryption key $E_{pk}(ck)$ are stored in data storage. (4) When a user wants to play the (encrypted) content $E_{ck}(c)$ in his or her playback device, $E_{pk}(ck)$ is first decrypted with the playback device's secret key sk, and $E_{ck}(c)$ is then decrypted with the obtained content key ck at the *content key decryption* module in the playback device. Figure 2 shows the overall architecture of the proposed recording protection system.

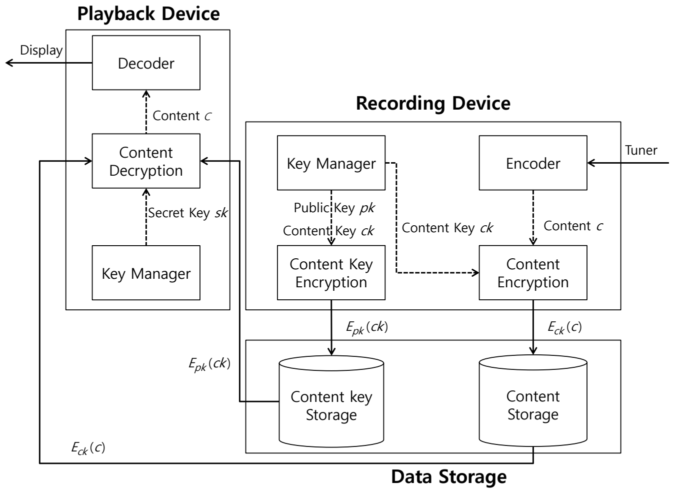


Fig. 2. Overview of the proposed recording protection system.

The main advantage of the proposed system is that the playback device can freely access the protected contents on the data storage without the assistance of the recording device. Also, this approach can be flexibly extended with multiple *authorized* playback devices belong to the recording device owner's personal network. To support multiple authorized playback devices, the recording device encrypts the content encryption key with those playback devices' public keys, respectively. The maximum number of authorized playback devices could be defined by a specific policy enforced by a content service provider.

Moreover, with respect to security, this approach is more preferable compared with the domain-based approach. As mentioned beforehand, it is difficult to maintain the backward and forward secrecy in domain-based content protection technologies. However, the backward and forward secrecy must be naturally satisfied in the proposed system because the public key of the removed (or added) device is never used in future (or previously) recorded contents. Another interesting feature is that the protected content encryption key is inherently bound to a specific device key unlike a domain key in an authorized domain. In general, the device key can be securely protected at the hardware and/or firmware level while software-based protection would often be used for the domain key.

For registration of the authorized playback devices, various implementation ways could be considered. For those implementations, a playback device's public key should be securely registered to the recording device through an authenticated channel between the playback device and the recording device (e.g., using public-key digital certificate). The simplest implementation of the public key registration is to securely embed the playback device's public key into the recording device during manufacturing processes. In practice, a recording device can be packaged with playback devices as a set of home appliances. In this case, the use of the embedded public key would be an acceptable solution.

IV. Performance evaluation

To show the feasibility and effectiveness of the proposed recording protection system, we implemented a prototype using the `PyCryto` library for encryption and decryption operations and performed a few experiments with the prototype implementation. The experiments were conducted with an Intel Xeon E3-1240 (3.40GHz), running on the Windows 10 with 8GB RAM. For symmetric key encryption algorithm, we used AES-128 with CBC mode [9]. For public key encryption algorithm, we used 2048-bit RSA-OAEP [10].

When a file is encrypted and decrypted, we measured the execution-time incurred by encryption and decryption operations. To decrease the bias associated with the performance realized from the testing samples, we repeated the test procedure 400 times with varying numbers of authorized playback devices from 1 to 9 and sizes of the files to be encrypted from 20MB to 100MB, respectively. The execution-time overheads were measured using the method `time.clock()` because `time.clock()` is generally reliable on the Windows platform.

We compared the performance of the proposed recording protection system with the conventional recording protection system that uses a *single symmetric encryption* operation for content key encryption. The purpose of the experiments shows that the proposed recording protection system can practically be implemented without incurring significant overhead compared with the conventional recording protection system.

First of all, we performed an experiment with 20MB of file size to show the effects of the number of authorized playback devices. The average execution time results for content encryption/decryption and content key encryption/decryption operations are shown in Figure 3.

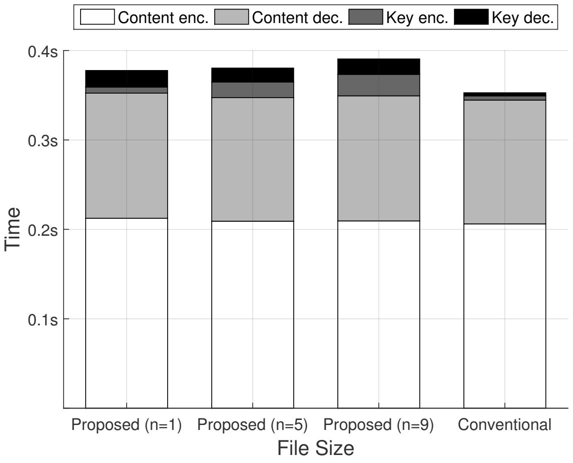
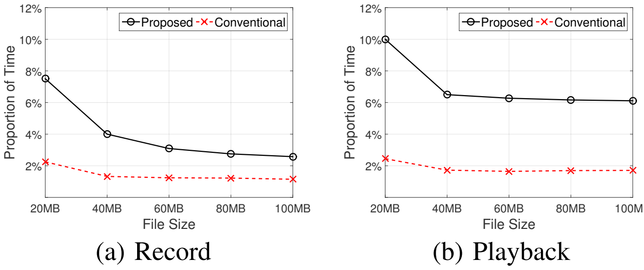


Fig. 3. Average execution time of recording protection systems with varying numbers (n) of authorized playback devices from 1 to 9.

The experimental results show that the execution time overheads incurred by content key encryption and decryption operations (darker colors in Figure 3) for the proposed recording protection system are comparable to those for the conventional recording protection system. In fact, as shown in Figure 3, the total execution time was overwhelmingly dominated by content encryption and decryption operations although there was a slight increase in the execution time for content key encryption and decryption operations with the number n of authorized playback devices.

We now move to the discussion on the performance of the recording protection systems with file size. To demonstrate this, we fixed the number of authorized playback devices as 5. The proportion of the execution time of the content key related operations during the *record* phase is shown in Figure 4(a). Unsurprisingly, the proportion of the execution time of the content key related operations during the *record* phase dramatically decreases as the file size increases. This is because the execution time of content key related operations is independent from the recorded content file size. We can see that the execution time overheads incurred by the content key related operations are marginal compared with the total execution time (within 4% of the total execution time when the recorded file size is greater than or equal to 40MB). As the file size increased from 20MB to 100MB, the gap between the proposed and conventional recording protection systems was rather reduced.

The proportion of the execution time of content key related operations during the *playback* phase is shown in Figure 4(b). Again, the proportion of the execution time of the content key related operations during the *playback* phase also decreases as the file size increases. The execution time overhead incurred by the content key related operations is about 6% of the total execution time when the recorded file size is greater than or equal to 40MB.



(a) Record (b) Playback

Fig. 4. Proportion of the execution time of content key related operations with varying sizes of the files to be encrypted from 20MB to 100MB.

V. Conclusion

In this paper, we presented a new recording protection system against the illegal distribution and promotion of pirated digital contents.

Unlike the conventional recording protection systems requiring complicated membership management operations, the proposed recording protection system is simple but practical. Our main idea is to use a public key cryptosystem to securely protect a communication channel between a recording device and a playback device. Consequently, the recorded contents can be flexibly played back with authorized playback devices by encrypting content encryption keys with their public keys, respectively.

Acknowledgment

This work was supported by the NRFK (No. 2014R1A1A1003707), ITRC (IITP-2016-R0992-16-1006), and ICT R&D program (No. B0717-16-0116).

References

[1] G. M. Toschi, L. B. Campos, and C. E. Cugnasca, "Home automation networks: A survey," *Computer Standards & Interfaces*, vol. 50, pp. 42–54, 2017.

[2] *DLNA's Overview and Vision*, [online] http://www.dlna.org, Digital Living Network Alliance.

[3] B. Miller, T. Nixon, C. Tai, and M. Wood, "Home networking with universal plug and play," *Communications Magazine, IEEE*, vol. 39, no. 12, pp. 104–109, 2001.

[4] *Bluebook A094r2*, [online] http://www.dvb.org/technology/dvb-cpcm/index.xml, DVB Content Protection and Copy Management.

[5] B. C. Popescu, B. Crispo, A. S. Tanenbaum, and F. L. Kamperman, "A DRM Security Architecture for Home Networks," in *Proceedings of the 4th ACM Workshop on Digital Rights Management*, 2004.

[6] N. Sakamoto, K. Muguruma, N. Koshino, S. Chiba, and M. Sakurai, "A digital hdtv receiver with home networking function and digital content storage," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 3, pp. 831–835, 2005.

[7] *Digital Transmission Content Protection Specification Volume 1*, [online] http://www.dtcp.com, Digital Transmission Licensing Administrator.

[8] F. L. A. J. Kamperman, L. Szostek, and W. Baks, "Marlin Common Domain: Authorized Domains in Marlin technology," in *Proceedings of the 4th IEEE Consumer Communications and Networking Conference*, 2007.

[9] J. Daemen and V. Rijmen, *The Design of Rijndael*. Springer-Verlag New York, 2002.

[10] M. Bellare and P. Rogaway, "Optimal Asymmetric Encryption — How to Encrypt with RSA," in *Advances in Cryptology — EUROCRYPT'94*. Springer Berlin Heidelberg, 1995.