SECURE CONTENT DISTRIBUTION SYSTEM BASED ON RUN-TIME PARTIAL HARDWARE RECONFIGURATION

Yohei Hori [†], *Hiroyuki Yokoyama* [‡] *and Kenji Toda* [†]

[†] National Institute of Advanced Industrial Science and Technology (AIST)
1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan
email: {hori.y, k-toda}@aist.go.jp

[‡] KDDI R&D Laboratories, Inc.
2-1-15 Ohara, Fujimino-shi, Saitama 356-8502, Japan
email: yokoyama@kddilabs.jp

ABSTRACT

A secure content distribution system is prototyped based on run-time partial reconfigurability of an FPGA. The system provides a robust content protection scheme for online content download services. The key idea is to divide the security module in a user terminal into Content-Specific Circuit (CSC) and Terminal Build-in Circuit (TBC) and to dynamically reconfigure CSC. CSC is customized for each content and transferred from a server in the form of encrypted configuration data. TBC is a uniquely identifiable processing unit that is combined with particular CSC to decrypt and decode contents. A content is properly decrypted and played by the security module only if its CSC is interlocked with the authorized TBC. To realize this CSC-TBC interlock authentication mechanism, partial reconfigurability of the FPGA is essential. This paper discusses the robustness and feasibility of the content distribution system through a proof-of-concept demonstration.

1. INTRODUCTION

Nowadays the online content market continues growing with the improvement of wired and wireless network infrastructures [1]. In online business, digital contents are always threatened by *piracy*, for instance illegal copying, cracking and diffusion of copyrighted contents. Packet streams on networks or data stored in local storage are easily copied and possibly cracked to extract original contents. Therefore technology to protect digital contents, i.e. Digital Rights Management (DRM), is the primary concern for digital content providers. However, content protection mechanisms are usually provided by application software or middleware running on a user terminal. In this case, once a user terminal is infected by malware such as virus and worm, all data and software in the file system can be accessed and abused by unexpected users.

For the protection of digital contents, hardware-based approaches will be more secure because (1) algorithms employed in the system are concealed and difficult to be analyzed, (2) existing malware cannot interfere with the data processing in circuits, and (3) neither plain data nor intermediate processing data appears even on local bus lines. In addition, flexibility of a Field-Programmable Gate Array (FPGA) enables us to introduce more powerful countermeasures against piracy. Authors have proposed a secure content protection system based on reconfigurable hardware [2]. This paper presents further discussion on robustness and feasibility of the reconfiguration-based content distribution system through a proof-of-concept demonstration.

The key feature of the system is that the data processing module in a user terminal is divided into two circuits: *Content-Specific Circuit* (CSC) and *Terminal Build-in Circuit* (TBC). CSC is a reconfigurable module that is customized for a specific content. Configuration data of CSC is transferred from a server in the form of encrypted bitstream. TBC is uniquely designed for each terminal. TBC covers the functionality of key generation, data decryption and image decoding, but is not workable until proper CSC is combined. A content is correctly replayed only if the corresponding CSC is configured and interlocked with the TBC in the licensed terminal.

2. RELATED WORKS

One of the effective approaches to ensure security of a system is to build dedicated hardware where a program is executed in the specific manner, ideally in the form of an encrypted code. Abyss [3], Citadel [4] and Dyad [5] are examples of such system. Disadvantages of these systems are that cost is quite expensive and architecture of the system is in-

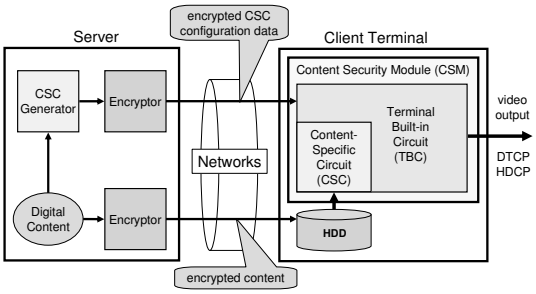

Fig. 1. Overview of the content distribution system.

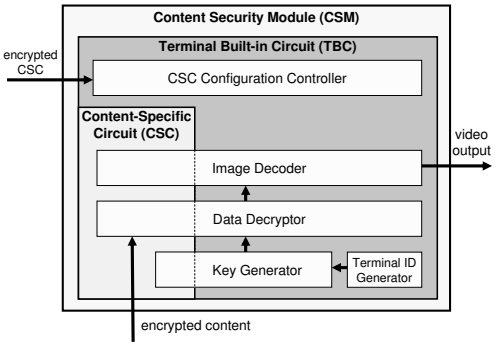

Fig. 2. Block diagram of Content Security Module.

flexible. Consequently the systems are difficult to maintain the robustness against new piracy.

To resolve the problems of cost and inflexibility, some researchers prototyped their secure processor with a Field-Programmable Gate Array (FPGA)[6, 7]. Also, a method to securely reconfigure a device via networks is reported[8].

Trusted Computing Group (TCG) [9] is an industry standards body constituted to define and promote standards for trusted computing. The core authentication mechanism for secure computing introduced by TCG is attestation by Trusted Platform Module (TPM). TPM attests trustworthiness of peripherals and guarantees integrity of the platform.

3. CONTENT DISTRIBUTION SYSTEM

3.1. Architecture of the System

Figure 1 shows functional architecture of the content distribution system. The system consists of a server, client terminals and networks connecting them. The core module of the content distribution system is Content Security Module (CSM). Figure 2 illustrates the block diagram of CSM. CSM consists of important processing modules, e.g. a key generator, a data decryptor and an image decoder. CSM contains two submodules, CSC and TBC.

3.1.1. Content-Specific Circuit

CSC is a reconfigurable part of CSM. CSC is customized for each content and is configured every time a content is replayed. CSC is not configured at the time system is booted up. When a user is to play a content, encrypted CSC configuration data is downloaded from a server, configured on the user terminal and combined with TBC.

Reconfiguration of CSC is also available for a content whose decryption and decompression algorithms frequently alter during it is replayed. With run-time partial reconfigurability of an FPGA, for example, changing a decipher key by reconfiguring CSC is possible during TBC is still in operational.

3.1.2. Terminal Built-in Circuit

TBC is uniquely identifiable module in the system. TBC consists of various components including a key generator, a data decryptor, a video decoder, an ID generator and a partial reconfiguration controller.

Though TBC is not a content-specific module, TBC can be also reconfigured for other reasons. Reconfigurability of TBC is useful to update, modify or replace the content protection mechanisms of the system. For example, changing architecture of whole system according to day or month would be effective to maintain system security. In addition, reconfigurability of CSC/TBC allows us to reactively provide counter measures against new piracy.

3.2. Mechanisms of Content Protection

The CSC-TBC architecture offers quite secure authentication mechanisms to the system.

To guarantee the communication between CSC and TBC, all signals crossing circuits' boundary must be transmitted through tri-state buffers. If the tri-state buffers are always located at the same places, CSC and TBC are correctly interfaced every time CSC is reconfigured. We call this correctly interfaced state *interlocked*. If unexpected CSC configuration data is downloaded whose tri-state buffers are differently located, CSC does not interlocked with TBC and the system does not work properly.

Because CSC is transferred in the form of encrypted configuration data which can be decrypted only by the authorized terminal, the content is still in safe in the case where the configuration data is tapped on networks. Even in the worst situation where the CSC configuration data is tapped and decrypted by a pirate, no decisive data (e.g. a decipher key) discloses because decrypted data is just a part of a circuit. Algorithms employed in CSC are difficult to be analyzed because they are hardwarelized, moreover, CSC does not output any significant information without being interlocked with a particular TBC.

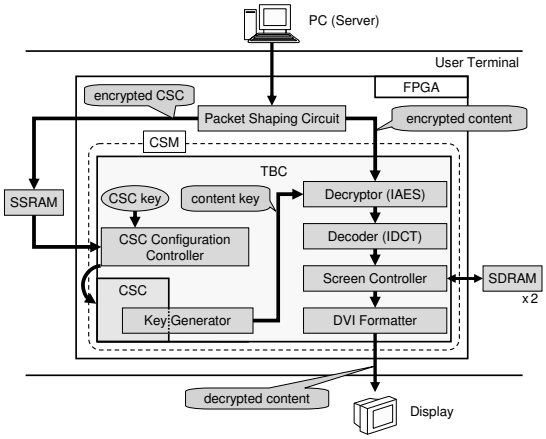

Fig. 3. Block diagram of the prototype system.

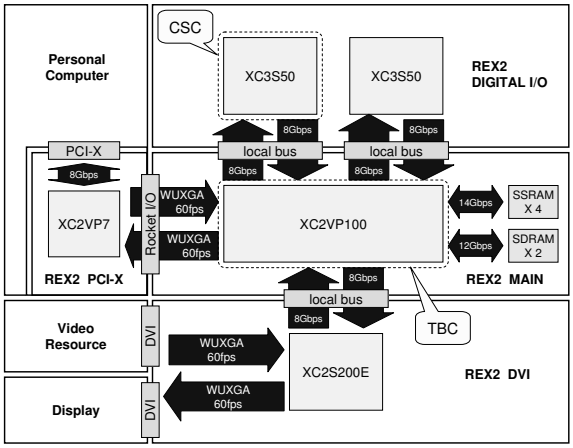

Fig. 4. Board configuration of the prototype system.

4. IMPLEMENTATION

4.1. Architecture of the Prototype

Figure 3 shows the functional architecture of the prototype system. The objective of this implementation is to proof the concept of the CSC-TBC mechanism. To avoid the restriction of column-based reconfiguration architecture and floorplanning of busmacros [10], CSC and TBC are tentatively implemented on two chips separately. CSC is implemented on Spartan-3 and TBC on VirtexII-Pro. CSC and TBC communicate each other to generate a key to decrypt a content. CSC is configured with boundary-scan protocol under the control of TBC. The details of the experimental setup of the prototype system is explained in Section 4.2.

The procedure for playing a content in the prototype system is described as follows. $\boldsymbol{K}_{csc}$ is a secret key embedded in the terminal to decrypt CSC configuration data ($\boldsymbol{D}_{csc}$). $\boldsymbol{K}_{cont}$ is generated after CSC is interlocked with TBC to decrypt an content ($\boldsymbol{D}_{cont}$). In the following expranation, encrypted data $\boldsymbol{D}$ is expressed as $\boldsymbol{E}\{\boldsymbol{D}\}$.

1. $\boldsymbol{E}\{\boldsymbol{D}_{csc}\}$ is sent from the PC to the system and stored in SSRAM.
2. $\boldsymbol{E}\{\boldsymbol{D}_{csc}\}$ is loaded from SSRAM and sent to CSC Decryptor. If $\boldsymbol{E}\{\boldsymbol{D}_{csc}\}$ is downloaded by the authorized terminal, it is decrypted with $\boldsymbol{K}_{csc}$.
3. CSC is configured with $\boldsymbol{D}_{csc}$ by Configuration Controller. In the experimentation, CSC is configured with boundary-scan protocol and implemented on Spartan3.
4. If CSC is interlocked with authorized TBC, $\boldsymbol{K}_{cont}$ is correctly generated. If CSC is not interlocked with TBC, an erronicous key is generated.
5. If the generated key is authentic, the encrypted content is properly decrypted, decoded and replayed.

Table 1. Hardware utilization of the prototype system (XC2VP100 [11]).

Resource	Utilization		(%)
Slices	10,289/	33,088	31%
LUTs	13,599/	66,176	20%
Block RAMs	235/	328	71%
GCLKs	8/	16	50%
DCMs	7/	8	87%
IOBs	388/	966	38%
GTs	5/	20	25%

4.2. Experimental Setup

The CSC-TBC authentication mechanism is applied to a Full HD (FHD, 1080p) image processor to test its feasibility. A content is of FHD resolution and encrypted with AES (CBC mode, 128-bit block size, 128-bit key). The prototype of the system is developed with *REX2* series experimental FPGA boards produced by REXEON Technology, Inc.[1] Figure 4 illustrates the board configuration of the prototype system.

As described in Fig. 4, CSC is implemented on Spartan3 and TBC on VirtexII-Pro. TBC, SSRAM and SDRAM work at 100 MHz; CSC works at 25 MHz.

4.3. Results

Firstly, hardware utilization of TBC implemented on VirtexII-Pro (XC2VP100) is given in Table 1. Table 1 shows that the amount of utilized logical resources in the system is small, in contrast that of Block RAM is relatively large. Thus, reducing Block RAM utilization has an impact on cost-reduction especially when the system is commercialized.

[1] REXEON Technology, Inc. is an entrepreneurial company fostered by AIST. For more information, see http://www.rexeon.com/.

Table 2. Hardware utilization of CSC (XC3S50 [12]).

Resource	Utilization		(%)
Slices	229/	768	29%
LUTs	237/	1,536	12%
Block RAMs	0/	4	0%
GCLKs	1/	16	12%
DCMs	1/	8	13%
IOBs	14/	124	11%

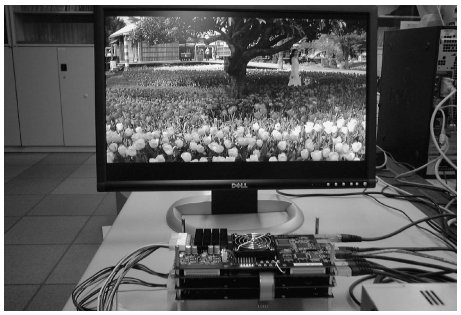

Fig. 5. Demonstration of the content distribution system.

Secondly, hardware utilization of CSC implemented on Spartan3 (XC3S50) is given in Table 2. As given in Table 2, hardware utilization of CSC is quite small with all the resources utilized less than 30%. The minimum reconfigurable area in VirtexII-Pro XC2VP100 is 4-column rectangle, and CSC in this experiment is able to be implemented in this area. In the case where CSC is implemented in 4-column rectangle, configuration data of CSC is reduced to about 13 kByte. Since CSC bitstream is transfered via networks, it is an important advantage that the size of CSC configuration data is diminutive.

Lastly, the demonstration of the content distribution system is given in Fig. 5. The figure shows that all of the following procedures are properly performed: (1) $\boldsymbol{E}\{\boldsymbol{D}_{csc}\}$ is transferred from PC and decrypted with $\boldsymbol{K}_{csc}$, (2) CSC is configured and interlocked with TBC, (3) $\boldsymbol{K}_{cont}$ is generated with the CSC-TBC mechanism, and (4) $\boldsymbol{E}\{\boldsymbol{D}_{cont}\}$ transferred from PC is decrypted, decoded and replayed.

5. CONCLUSIONS

This paper presented a secure content distribution system utilizing partial reconfigurability of an FPGA. Since preventing illegal copy of streaming contents is difficult, we focused on the mechanism to disable illegal use of downloaded contents.

The hardware-based authentication given in this paper provides a secure content protection scheme in online content distribution business. The key idea of the system is to divide a security module in the user terminal into Content-Specific Circuit (CSC) and Terminal Build-in Circuit (TBC) and to dynamically reconfigure the CSC. A content is properly replayed only after its correct CSC is downloaded from a server, configured in the user terminal and interlocked with authorized TBC. This CSC-TBC interlock architecture provides quite robust authentication mechanism for following reasons: (1) user terminal is incomplete and unworkable until correct CSC is interlocked with authorized TBC, (2) algorithms employed in the system are hardwarelized and difficult to be analyzed, (3) even CSC is wiretapped and cracked by an unauthorized user, the decrypted data is just a part of a circuit and no decisive data is disclosed to anyone.

To test the feasibility of CSC-TBC interlock authentication mechanism, a prototype system is developed with state-of-the-art REX2 series FPGA boards. The CSC-TBS mechanism is applied to a full high-definition (1920x1080) image processor to replay encrypted contents. In the experimentation, the encrypted movies streamed from the server are successfully replayed. The experimentation demonstrates that decryption of CSC configuration data, configuration of CSC, interlock of CSC and TBC, generation of the decipher key and decryption of the content are all performed correctly under the CSC-TBC mechanism.

As future works, firstly, implementing CSC and TBC on a single FPGA is the main subject that should be tackled. Secondly, architecture and hardware algorithms of CSC should be studied to enhance robustness of the system. Lastly, introducing tamper-resistant technology to the system is left for further research.

6. REFERENCES

[1] "Online paid content U.S. market spending report," Online Publishers Assosication, 2005.

[2] H. Yokoyama and K. Toda, "FPGA-based content protection system for embedded consumer electronics," in *Proc. RTCSA*, 2005, pp. 502–507.

[3] S. White and L. Comerford, "ABYSS: A trusted architecture for software protection," in *IEEE Symp. Security and Privacy*, 1987, pp. 38–51.

[4] S. White, W. Weingart, W. Arnold, and E. Palmer, "Introduction to the Citadel architecture: Security in physically exposed environments," Distributed Security Systems Group, IBM Thomas J. Watson Reserch Center, Tech. Rep., 1991.

[5] D. Tygar and B. Yee, "Dyad: A system for using physically secure coprocessors," Dept. Comput. Sci., Carnegie Mellon Univ.," CMU-CS-91-140R, 1991.

[6] G. E. Suh, C. W. O'Donnell, and S. Devadas, "AEGIS: A single-chip secure processor," in *Information Security Technical Report*. Elsevier, 2005, vol. 10, pp. 63–73.

[7] J. Zambreno, D. Honbo, A. Choudhary, R. Simha, and B. Narahari, "High-performance software protection using reconfigurable architectures," *Proc. IEEE*, vol. 94, no. 2, pp. 419–431, 2006.

[8] H. Song, J. Lu, J. Lockwood, and J. Moscola, "Secure remote control of field-programmable network devices," in *FCCM'04*, 2004, pp. 334–335.

[9] "Trusted Computing Group (TCG)," https://www.trustedcomputinggroup.org/home/.

[10] *Virtex FPGA Series Configuration and Readback, XAPP138, v2.8*, Xilinx, Inc., 2005.

[11] *Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet v4.0*, Xilinx, Inc., San Jose, CA, 2004.

[12] *Spartan-3 FPGA Familiy: Complete Data Sheet*, Xilinx, Inc., San Jose, CA, 2005.