A Secure Digital Content Delivery System Based on Partially Reconfigurable Hardware

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

[†] National Institute of Advanced Industrial Science and Technology (AIST)
1-1-1 Umezono, Tsukuba-shi, Ibaraki 305-8568, Japan
email: {hori.y, hirofumi.sakane, 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

We developed an FPGA-based content delivery system to securely distribute digital content on the Internet. With partial reconfigurability of a Xilinx Virtex-II Pro FPGA, the system provides a flexible single-chip solution for protecting digital content. In the system, a partial circuit must be downloaded from a server to the client terminal to play content. Content will be played if and only if the downloaded circuit is correctly combined (= interlocked*) with the circuit built in the terminal. Since each circuit has a unique I/O configuration, the downloaded circuit interlocks with the corresponding built-in circuit designed for a particular terminal. Thus, the interface of the circuit itself provides a novel authentication mechanism. In the present paper, we describe the detailed architecture of the proposed system and clarify the feasibility and effectiveness of this system experimentally using a single-chip partial reconfiguration. In addition, we discuss the fail-safe mechanisms, partially reconfigurable FPGA architecture, and future research necessary for the practical application of the system.*

1. Introduction

The expansion of broadband networks and the spread of Internet-access-ready mobile terminals have brought about the rapid growth of the on-line content market [1]. Digital content delivery systems, however, are always threatened by prevalent piracy, *i.e.*, system cracking and illegal copying of copyrighted content. Therefore, technology of Digital Rights Management (DRM) is of primary concern for content providers.

Based on the above considerations, various studies have examined the secure distribution of content on the Internet [2–4]. As observed in these studies, the security of content is guaranteed mainly by cryptographic technology, *e.g.*, RSA [5] and AES [6]. However, even though the implemented cryptographic algorithms are theoretically unbreakable, vulnerabilities could emerge when the algorithm is instantiated in the real world [7–11]. Since it is nearly impossible to completely eliminate design defects from recently developed complicated systems, confidential data will be made vulnerable by these flaws. Since attack methods that exploit such vulnerabilities continue to advance, content delivery systems require not only security but also flexibility in order to employ countermeasures against new piracy.

To develop a secure and flexible content delivery system, the use of the partial reconfigurability of an FPGA has been proposed [12, 13]. In the proposed system, a partial circuit must be downloaded from the server to the client terminal in order to play content. Content is properly played only when the downloaded circuit is correctly combined with the circuit built in the terminal. In the past work, the concept of this system was tested by emulating partial reconfiguration by two FPGAs. In our study, we implement the content protection mechanism on a single Virtex-II Pro FPGA. This paper describes the detailed architecture of the system and clarifies the effectiveness of the system experimentally using a single-chip partial reconfiguration.

2. System Architecture

2.1. Overview of the System

Figure 1 shows an overview of our FPGA-based content delivery system. The system consists of a server, client terminals, and networks connecting them. To play content on a client terminal, a user must download a partial circuit from the server. The downloaded circuit is called the *Content-Specific Circuit* (*CSC*), and the circuit built in the client terminal is called a *Terminal Built-in Circuit* (*TBC*). We use the term *interlock* to describe the condition in which CSC

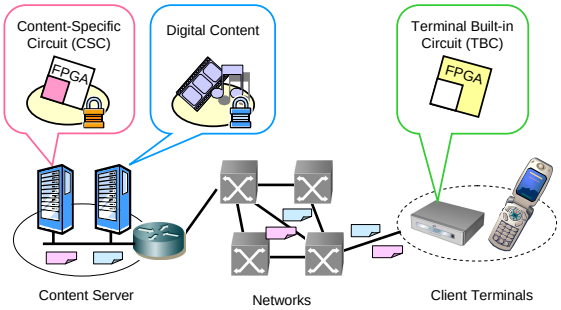

Figure 1. Overview of the content delivery system.

and TBC are correctly combined to work as intend. The key idea of the system is to implement modules *across* CSC and TBC so that the system works properly only when CSC and TBC are correctly interlocked.

2.2. Mechanisms of Content Protection

2.2.1 Authentication with I/O Configuration

To play content on the client terminal, proper CSC must be configured and interlocked with TBC, in other words, signals between CSC and TBC must be correctly connected. Since each TBC has a unique I/O configuration, CSC interlocks only with the TBC of a specific user. For this reason, even if a CSC is leaked and distributed on the network, the leaked CSC will not work on the terminal of another user.

2.2.2 Content-Specific Hardware Architecture for Illegal-Play Prevention

As mentioned earlier, algorithms implemented on the CSC vary depending on the content to be played. The CSC can be used for playing only specific content. Thus, playable content is determined by the architecture of the downloaded CSC. For this reason, even if a plain CSC bitstream is distributed on the network, it is difficult to determine which content is playable with the CSC.

2.2.3 Data and Algorithm Obfuscation

In the system, a partial bitstream of a key generating circuit, not a key itself, is transferred from the server. Even if the encrypted bitstream is obtained surreptitiously and decrypted for some reason, the bitstream is sufficiently intractable to most attackers. In addition, the behavior of the entire circuit will not be determined from the partial bitstream because it is merely a small fraction of the entire configuration data.

2.2.4 Single-chip Wiretapping-resistant Architecture

With partial reconfigurability of an FPGA, CSC and TBC are implemented on a single chip. Therefore, any communication between CSC and TBC cannot be wiretapped on the external buses. During the processing of key generation or content decryption, neither decipher keys nor intermediate data will be exposed to attackers.

2.2.5 Reactive Architecture Modification

As the architecture of recent devices and systems becomes increasingly complicated, it is nearly impossible to completely eliminate defects and security vulnerabilities in consumer electronics. In fact, new attack techniques exploiting such vulnerabilities have been frequently reported. With the reconfigurability of an FPGA, we can eliminate defects and vulnerabilities in a product even after shipment.

3. Implementation

3.1. Architecture of the Prototype System

Figure 2 shows the architecture of the prototype system. As the figure denotes, the *Content Key Generator* and the *Content Decryptor* are implemented across CSC and TBC. In the system, the CSC bitstream and content are encrypted/decrypted with AES128 [6]. We use a desktop computer as a server to transfer the encrypted CSC and content to the prototype system. For convenience, we define the symbols listed below.

D_{tid} : a 128-bit ID number given to each terminal.
D_{cid} : a 128-bit ID number given to each content.
D_{seed} : a 128-bit random number sent from the server.
D_{csc} : a configuration bitstream of CSC.
D_{cont} : original content data.
K_{csc} : a 128-bit key to encrypt/decrypt D_{csc} .
K_{cont} : a 128-bit key to encrypt/decrypt D_{cont} .
$\mathcal{E}\{D\}_K$: data D encrypted with the key K.

Note that the system uses two different keys: K_{csc} embedded in TBC and K_{cont} generated with the CSC-TBC mechanism. Figure 3 shows the detailed block diagram of the processing performed in CSC and TBC. The purpose of this implementation is to verify the feasibility of the CSC-TBC interlocking mechanism, and so the strength of the functions is not considered at this time. We implemented typical operations, (*e.g.*, exclusive OR, cyclic shift, and table reference), on CSC and TBC in order to estimate hardware utilization of a completed system.

We developed a prototype system with *REX2 series*, the FPGA boards produced by REXEON Technology, Inc. REX2 is equipped with a VirtexII-Pro(XC2VP70) [14]. Hardware utilization of CSC and TBC are given in Table 1. The size of the encrypted partial bitstream of CSC is about

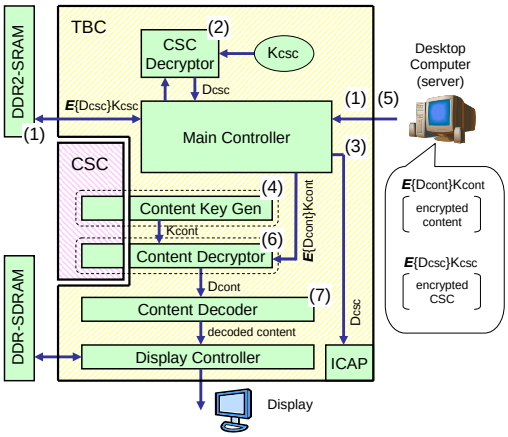

Figure 2. Block diagram of the prototype system.

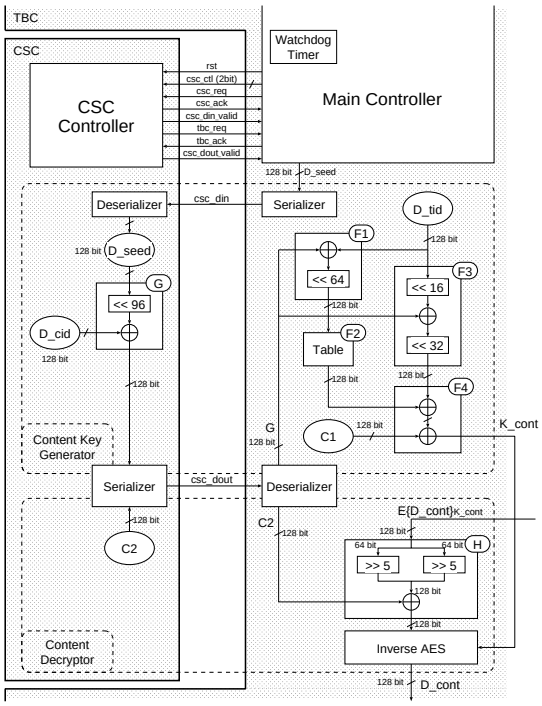

Figure 3. Diagram of the CSC-TBC processing.

75 kB, and processing time of decryption and reconfiguration of CSC is about 5.7 msec in total.

4. Experiment

4.1. Test Cases

In order to demonstrate the concept whereby the content is played only when (1) the signals between CSC and TBC are correctly connected and (2) CSC has proper functions for the content, we play a high-definition (1920 x 1080, 30 fps) movie using various CSC-TBC configurations. In order to

Table 1. The hardware utilization of TBC.

Resource	TBC	CSC	**Total**	**(%)**
Slice	311	11,881	**12,192**	**37%**
Block RAM	0	235	**235**	**72%**
MULT18x18	0	328	**328**	**16%**
BUFGMUX	-	-	**7**	**43%**
DCM	-	-	**7**	**88%**
ICAP	-	-	**1**	**100%**

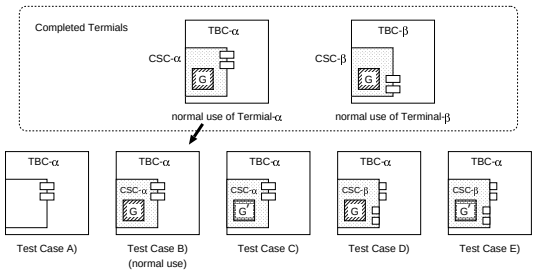

Figure 4. Test cases of the experiment.

verify the experimentally obtained results, we prepare two client terminals: Terminal-α and Terminal-β. Terminal-α works correctly if TBC-α interlocks with CSC-α, which has function G. Similarly, Terminal-β works correctly if TBC-β interlocks with CSC-β, which has function G.

The only difference between Terminal-α and Terminal-β is the positions of the two bus macros. In order to intentionally provide an erroneous function to CSC, we implement G', which returns the inverted value of G. With the symbols defined here, the test cases performed in the experiment are listed below and are also illustrated in Fig. 4.

A) CSC is not configured on the Terminal-α. In this case, the terminal is simply booted up and CSC is not yet downloaded.

B) CSC-α with function G is configured on TBC-α. This is a normal case, in which an authorized user is to play content on an appropriate terminal.

C) CSC-α with function G' is configured on TBC-α. In this case, a user attempts to play content, but the CSC is for different content.

D) CSC-β with function G is configured on TBC-α. In this case, a user attempts to configure the CSC of a different user, or a malicious person surreptitiously obtains the CSC of a user and attempts to play content on an unauthorized terminal.

E) CSC-β with function G' is configured on TBC-α. In this case, an unauthorized person attempts to play content with a fake CSC on a pseudo-terminal.

Table 2. Resultes of the experiment.

Test case	Result
Case A)	White noise displayed.
Case B)	Content properly played.
Case C)	White noise displayed.
Case D)	System halted.
Case E)	System halted.

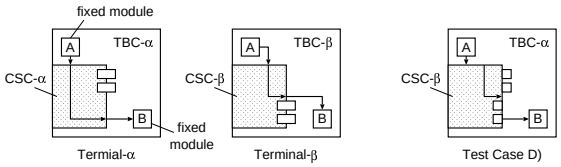

Figure 5. Inconsistent wiring in unsuccessful interlocking.

4.2. Experimental Results

The results of the experiment is given in Table 2. As shown in Table 2, the content is properly played only in the case B), where functions implemented on CSC are proper for the content and signals between CSC and TBC are correctly connected. The resultes shows that (1) the content-specific architecture of CSC can control key generation and content decryption so that only the targeted content is played on the terminal, and (2) the CSC-TBC interlocking mechanism can prevent content from being improperly played on unauthorized terminals.

5. Discussion

We focus on results D) and E), where the system halted after an incorrect CSC was configured. Note that fixed modules are allowed to use routing resources in PRR [15]. This means that the PRM also contains interconnections among fixed modules. Normally, the physical positions of these interconnections do not change even when the PRM is replaced by a different PRM, because the replacement PRM has the same wiring as the fixed modules.

In test cases D) and E), however, CSC-β presumably contains inconsistent wiring that cannot be connected to TBC-α. As noted in Fig. 5, the positions of the bus macros in CSC-β are different from those in CSC-α. Therefore, wiring of the fixed modules in CSC-β probably differs from that in CSC-α. This will cause the system to halt, while the Main Controller is permanently waiting for a signal from the CSC or the Watchdog Timer.

In a system in which a circuit is partially/entirely reconfigured, countermeasures against unexpected errors must be carefully devised. Since the architecture of the circuit itself is changed by reconfiguration, an erroneous bitstream could cause fatal damage to the system. In particular, in a reconfigurable system connected to the Internet, the system must be protected against malicious bitstreams sent by attackers.

6. Conclusions

We developed an FPGA-based content delivery system to securely distribute digital content on the Internet. The system effectively utilizes the partial reconfiguration of an FPGA to protect digital content. In the system, a partial circuit must be downloaded from the server to the client to play content. The content is properly played only when the downloaded circuit is correctly combined (= *interlocked*) with the circuit built in the terminal.

We implemented the interlocking mechanism on a Virtex-II Pro FPGA and tested the feasibility of the system. We verified that terminal authentication and content play control with partial reconfiguration worked successfully to protect the digital content.

In the future, we will implement a fail-safe mechanism to defend the system against erroneous bitstreams. A system failure can be caused by not only malicious attacks but also design defects. Further research on countermeasures against various attacks and errors is required.

References

[1] Digital Content Association of Japan, "Digital content white paper," 2006.

[2] A. Waller, G. Jones, T. Whitley, J. Edwards, D. Kaleshi, A. Munro, B. MacFarlane, and A. Wood, "Securing the delivery of digital content over the Internet," *Electronics & Communication Enginieering J.*, vol. 14, no. 5, pp. 239–248, 2002.

[3] J. Lee, S. O. Hwang, S.-W. Jeong, K. S. Yoon, C. S. Park, and J.-C. Ryou, "A DRM framework for distributing digital contents through the internet," *ETRI J.*, vol. 25, no. 6, pp. 423–436, 2003.

[4] J. Zhao, Y. Qi, and Z. Ma, "Secure multimedia streaming with trusted digital rights management," in *LCN'05*, 2005, pp. 817–821.

[5] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[6] National Institute of Standards and Technology, "Announcing the advanced encryption standard (AES)," FIPS PUB 197, Nov. 2001.

[7] R. Anderson, "Why cryptosystems fail," *Communications of the ACM*, vol. 37, no. 11, pp. 32–40, 1994.

[8] S. H. Weingart, "Physical security devices for computer subsystems: a survey of attacks and defenses," in *CHES'00*, 2000, pp. 302–317.

[9] M. Bond and R. Anderson, "API-level attacks on embedded systems," *Computer*, vol. 34, no. 10, pp. 67–75, 2001.

[10] S. Smith, "Fairy dust, secrets, and the real world," *IEEE Security & Privacy*, vol. 1, no. 1-2, pp. 89–93, 2003.

[11] R. Anderson, M. Bond, J. Clulow, and S. Skorobogatov, "Cryptographic processors—a survey," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 357–369, 2006.

[12] H. Yokoyama and K. Toda, "FPGA-based content protection system for embedded consumer electronics," in *RTCSA'05*, 2005, pp. 502–507.

[13] Y. Hori, H. Yokoyama, and K. Toda, "Secure content distribution system based on run-time partial reconfiguration," in *FPL'06*, 2006, pp. 637–640.

[14] *Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet v4.6*, Xilinx, Inc., 2007.

[15] *Early Access Partial Reconfiguration User Guide For ISE 8.1.01i*, Xilinx, Inc., 2006.