Tackling the Security Issues of FPGA Partial Reconfiguration with Physical Unclonable Functions

Yohei Hori[1,2], **Toshihiro Katashita**[1,2] **and Akashi Satoh**[1]
[1]National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki, Japan
[2]CREST, Japan Science and Technology Agency, Chiyoda-ku, Tokyo, Japan

Abstract— *Protecting the confidentiality and integrity of data processed by field-programmable gate arrays (FPGAs) is a major concern in the field of electronics as the use of FPGAs is becoming increasingly widespread in various commercial, industrial and other products. Since the FPGA bitstream is essentially an electronic data stream, it is susceptible to eavesdropping and tampering during transport via a data bus or network. Such security issues clearly hinder the use of systems supporting partial reconfiguration, where users can design their own circuits or download and implement custom circuits from the Internet on demand.*

Although currently available high-end FPGAs feature cryptographic cores to counteract such security issues, in 2011 it was reported that a cryptosystem on an FPGA could be broken by means of a side-channel attack. The success of this type of attack indicates that, in the presence of state-of-the-art techniques, a fixed key in the memory constitutes a flaw in security-sensitive systems.

To address the problem of side-channel analysis, we have developed evaluation boards referred to as SASEBO, which is an acronym for Side-Channel Attack Evaluation Board and serves as a collective name for the entire range of evaluation boards developed thus far, namely SASEBO, SASEBO-G, -B, -R, -GII, -W, -RII and -GIII. Used in more than 30 countries, SASEBO is the world's most popular series of evaluation boards for side-channel analysis. The latest board (SASEBO-GIII) is equipped with the newest Xilinx Kintex-7 FPGA for cryptographic module evaluation and Spartan-6 for system control.

To resolve the security issue of embedded secret keys, we are also developing and evaluating Physical Unclonable Functions (PUFs) with SASEBO. A PUF is a circuit that generates a device-specific identifier by using the process variation of the device. Such variations are virtually unclonable, and thus the output of the PUF is considered to be a device fingerprint*, which is expected to be unique among devices. The generated identifier is used to communicate secret information, and there is no need to store that information in the device itself.*

In this paper, we outline security issues in the modern large-scale integration market, and we demonstrate the functionality of SASEBO boards. We also explain how PUFs can solve the abovementioned security problems concerning FPGAs.

Keywords: Dynamic Partial Reconfiguration (DPR), SASEBO, Side-Channel Analysis (SCA), Physical Unclonable Function (PUF), Authenticated Encryption

1. Introduction

Dynamic Partial Reconfiguration (DPR), or Partial Run-Time Reconfiguration (RTR) of Field-Programmable Gate Arrays (FPGAs), refers to the ability to replace a portion of a circuit with another module while the rest of the circuit remains fully operational. FPGAs of the Xilinx Virtex family are probably the most popular dynamically reconfigurable FPGAs, and recently Altera announced that their Stratix V FPGA also supports DPR. In a DPR system, a user can change the functionality of the system on demand by downloading a hardware module suitable for particular applications, performance requirements or environments. Similarly to downloadable software, such as JavaScript and ActiveX content, downloadable hardware services for reconfigurable hardware devices are expected to become available in the near future. The flexibility of DPR is expected to increase the versatility of such hardware systems as well as to improve their cost effectiveness and area efficiency. DPR also results in shorter configuration times and consequently makes reconfigurable computing more practical and operational. The application of DPR has been studied in the fields of content distribution [1], network processing [2], image processing [3], automotives [4], fault-tolerant and self-healing systems [5], and software defined radio [6] among others.

However, there are certain security issues to consider before DPR can be applied in practice. Since hardware configuration data (bitstreams) for FPGAs can be downloaded from the Internet, the bitstreams are always exposed to attackers on the network. As represented by *Side-Channel Analysis (SCA)*, which exploits power consumption measurement or electromagnetic emanation to obtain secret keys, technology which can be used for attacks is becoming more sophisticated every day, and simple encryption and authentication techniques might not always be sufficient for protecting confidential data. Indeed, according to recent reports, the bitstream security mechanisms of some FPGAs have been defeated by differential power analysis [7], [8]. In other words, secret information stored in the memory

can be extracted by state-of-the-art attacks. Therefore, we are currently addressing this security issue with *Physical Unclonable Functions (PUFs)*, which extract unclonable process variation of the devices and provide fingerprints which uniquely identify these devices.

In the following sections, we provide a brief introduction of the security issues concerning FPGAs, SCAs and PUFs.

1.1 Security Issues Concerning FPGAs

Since a bitstream is merely an electronic data stream, it is constantly exposed to threats such as illicit cloning, reverse engineering and other forms of tampering, and therefore bitstream encryption is essential for protecting FPGA IP cores. Encryption-only systems, however, are not sufficiently secure since they cannot prevent erroneous or malicious bitstreams from being used for configuration. Since DPR changes the hardware architecture of the circuit, an unauthorized bitstream can cause fatal, unrecoverable damage to the system or may cause secret information to leak through a network connection. Such a malicious bitstream is referred to as a *hardware virus* or a *hardware trojan*. Cryptographic schemes also provide DPR systems with solutions for preventing damage from being inflicted by such hardware viruses and trojans.

To use DPR systems in practice, mechanisms for bitstream protection, safe configuration and side-channel attack prevention should be implemented in accordance with the intended application of the specific system. In this regard, we have developed a secure DPR system using the Advanced Encryption Standard with the Galois/Counter Mode (AES-GCM) [9], [10], which is one of the latest Authenticated Encryption (AE) systems [11]. AE is a cryptographic algorithm that provides both message confidentiality and authenticity. Also, several studies on bitstream protection have been reported thus far [12]–[18].

Modern cryptography provides a reasonably good solution to the security issues associated with DPR systems. However, we must also take SCA into consideration in order to be able to counteract more sophisticated attacks since the secret key of the encryption core embedded into the FPGA can be revealed by SCA.

1.2 Side-Channel Analysis

SCA is a collective term for a range of non-invasive attacks targeting cryptographic modules which focuses on the power consumption, electromagnetic emanation, and the leakage of other information about the physical state of electronic devices. Using SCA, an attacker can extract secret information from inside the target without physically accessing the device. The cost of SCA attacks is usually low, requiring only basic equipment, such as a digital oscilloscope and a personal computer along with the target device. Therefore, SCA is a rather straightforward but powerful attack technique targeting cryptographic modules.

After Kocher et al. reported the first SCA (based on timing analysis) in [19] and subsequently *Simple Power Analysis (SPA)* and *Differential Power Analysis (DPA)* in [20], SCA has become widely recognized in both industry and academia as a serious problem concerning cryptographic modules. Many derivative attacks have been studied to date such as correlation power analysis [21], electromagnetic analysis [22], [23] and mutual information analysis [24].

In 2011, Moradi et al. successfully extracted the secret key of the encrypted bitstream from a Virtex-II Pro FPGA by recovering the three encryption keys of the Triple-DES algorithm from 25,000 power traces obtained during a single boot-up process [7]. It should be noted that the technique adopted by Moradi et al. required only 3 min to extract the key.

To facilitate the study of SCA at academic, industrial and governmental institutions, we have developed and distributed a standard experimental environment named SASEBO, or Side-channel Attack Standard Evaluation Board [25]. SASEBO is a collective name for a series of evaluation boards developed thus far, namely SASEBO, SASEBO-G, -B, -R, -GII, -W, -RII and -GIII. Used in more than 30 countries, the SASEBO boards are now the world's most popular SCA evaluation boards. It should be emphasized that SASEBO-GII and -GIII are also designed to support DPR system evaluation, where the target device can be (re)configured in various ways to study the feasibility and effectiveness of DPR systems. It is particularly important that one of the two FPGAs on these boards can be dynamically reconfigured under the control of the other FPGA. A detailed explanation of SASEBO will be given in Section 2.

1.3 Physical Unclonable Functions

In this context, storing a secret key in memory might not provide sufficient security with respect to sensitive information. Therefore, we look to PUFs as an effective solution to SCA attacks.

A PUF is an object that outputs a device-specific response based on its intrinsic physical characteristics. In this sense, the texture of paper can serve as a PUF, however here we consider PUFs in the context of semiconductors (silicon PUFs [26]). A silicon PUF (hereafter referred to simply as "PUF") is a circuit constructed on a semiconductor, and its purpose is to output a unique identifier (ID) based on variation in the device. By using a PUF for key generation, the secret key need not be embedded in the FPGA, which can protect the device against side-channel attacks. Another novelty associated with using PUFs for FPGAs is that different IDs can be generated from the *same* bitstream. Although bitstreams are common for all devices, device-specific data are generated as a result of physical differences between individual devices. Note that the bitstream itself does not necessarily include any secret information. As

a consequence, the bitstream of the PUF can be safely transferred over unsecured network channels.

Maes and Verbauwhede have categorized PUFs into non-electronic PUFs, analog electronic PUFs, delay-based intrinsic PUFs and memory-based intrinsic PUFs [27]. Among these, delay-based and memory-based PUFs can be applied to FPGAs. Examples of delay-based PUFs can be given with arbiter PUFs [28], ring oscillator (RO) PUFs [29], Glitch PUFs [30] and others, while examples of memory-based PUFs include SRAM PUFs [31], butterfly PUFs [32] and tri-state PUFs [33].

Our Pseudo-LFSR PUF (PL-PUF) [34], which is a delay-based type of PUF, was developed to eliminate certain shortcomings of existing PUFs. A conventional delay-based PUF outputs a response consisting of one or several bits from a challenge consisting of a long bitstream, and consequently has a low throughput. Additionally, some types of PUFs can be attacked by using machine learning to perform mathematical modeling of their signal delay characteristics. In contrast, PL-PUF efficiently outputs an N-bit response from an N-bit challenge, and the size of the PL-PUF circuit is reasonably small. Although the structure of PL-PUF is based on the Linear Feedback Shift Register (LFSR), in fact it does not contain a shift register; rather, it constitutes a large combinational logic. As a result of this structure, modeling its delay is considered to be exceedingly difficult. Furthermore, the challenge-response mapping of the PL-PUF can be varied depending on the active duration of the circuit, that is, a single PL-PUF behaves as though it consists of multiple PUF cores. The PL-PUF is explained in detail in Section 3.

1.4 Organization of this Paper

The remainder of this paper is organized as follows. Section 2 presents our SCA evaluation boards of the SASEBO family, where the details about the structure, functionality and various configuration mechanisms of the boards are explained. Section 3 introduces PL-PUF together with details of its structure and the results of its implementation on SASEBO-GII boards, and the effectiveness of PL-PUF is discussed on the basis of performance evaluation results. Furthermore, Section 4 presents a secure DPR system using authenticated encryption AES-GCM together with results regarding its structure and implementation. Finally, Section 5 summarizes the paper and the directions of future work.

2. SASEBO

SASEBO was our first version of an SCA evaluation board and is also the collective name for the entire series of evaluation boards developed thus far, namely SASEBO-G, -B, -R, -GII, -W, -RII and GIII. SASEBO is developed to provide an experimentation environment for SCA to researchers from various academic and industrial fields. SASEBO is currently the most common SCA evaluation board in the world, being used at more than 100 academic, governmental and industrial institutions in more than 30 countries. Images of the boards in the SASEBO family are shown in Fig. 1 through 6, and a summary of the functions of each board is given in Table 1.

After successful timing attacks and differential power analysis (DPA) were reported in 1996 and 1999 respectively, SCA attacks were recognized as a serious threat to the industry. However, at the time there was no common experimental environment for conducting SCA tests, so some research groups developed their own evaluation boards while others modified off-the-shelf boards to measure the power consumption of the chips. These experimental environments were drastically different from each other, which rendered the comparison of the experimental results obtained by different groups meaningless. Furthermore, even when a novel SCA experiment was conducted, performing independent confirmation experiments was virtually impossible since the original test environment in which the novel experiments were performed was unavailable to third-party research groups. The lack of a uniform and consistent test environment led to the development of SASEBO.

2.1 SASEBO-GIII

The latest board of the SASEBO family, SASEBO-GIII, is equipped with a Xilinx Kintex-7 FPGA as a testing cryptographic module and a Spartan-6 FPGA for implementing control logic. Its strongest advantage is in terms of expandability, with two standard FMC LPC[1] connectors. Therefore, off-the-shelf boards with an FMC connector, for example, HDMI cards, Ethernet cards and camera boards, can be connected to SASEBO-GIII. The configuration pins of Kintex-7 are connected to and controlled by Spartan-6, allowing the user to test complete and partial reconfigurations of the chip through the configuration pins.

Since SASEBO-GIII is not yet commercially available, in the following section we explain the functionality and configuration mechanisms of SASEBO-GII.

2.2 SASEBO-GII

The SASEBO-GII board is designed for developing secure DPR systems, as well as for improving the logic capacity and signal quality for advanced research on side-channel attacks. The FPGAs on SASEBO-GII can be configured via different interfaces: JTAG, SPI, SelectMAP and ICAP [35], and the designer can examine various configurations and evaluate the security of the developed configuration procedure.

In this section, first we explain the basic specifications of the SASEBO-GII board, after which we describe the various FPGA configuration patterns realized with the board.

2.2.1 Board Structure

The block diagram of the board is shown in Fig.7 and its basic features are summarized in Table 2.

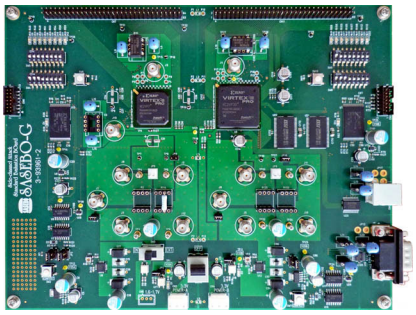

Figure 1: SASEBO-G

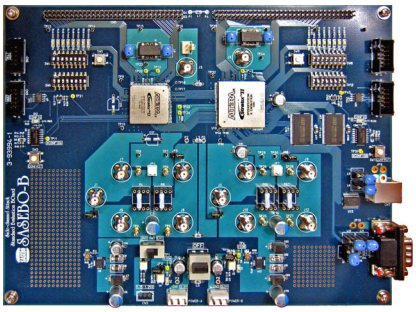

Figure 2: SASEBO-B

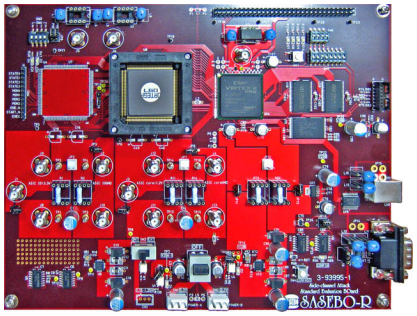

Figure 3: SASEBO-R

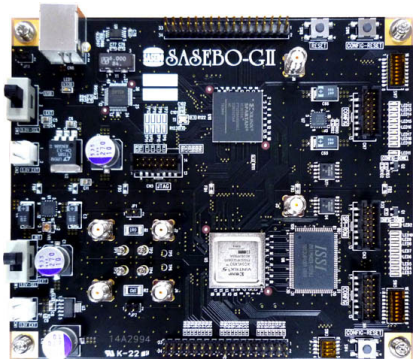

Figure 4: SASEBO-GII

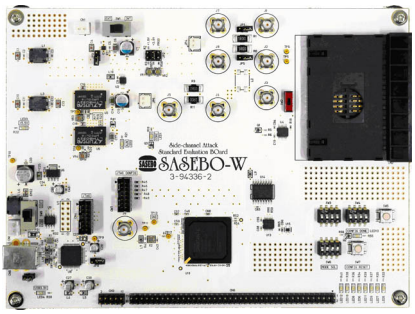

Figure 5: SASEBO-W

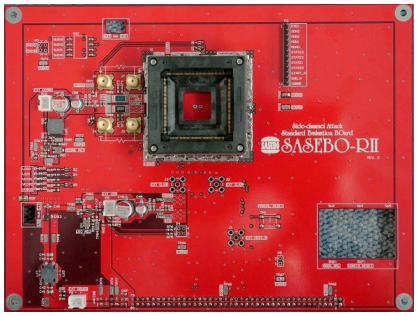

Figure 6: SASEBO-RII

Table 1: Summary of the SASEBO family.

Name	Year	Cryptographic Device	Control Device
SASEBO	2007	Virtex-II Pro (XC2VP7)	Virtex-II Pro (XC2VP30)
SASEBO-G	2008	Virtex-II Pro (XC2VP7)	Virtex-II Pro (XC2VP30)
SASEBO-B	2008	Stratix-II (EP2S15)	Stratix-II (EP2S30)
SASEBO-R	2008	LSI socket (QFP160)	Virtex-II Pro (XC2VP30)
SASEBO-GII	2009	Virtex-5 (XC5VLX30/50)	Spartan-3A (XC3S400A)
SASEBO-W	2010	Smartcard slot	Spartan-6 (XC6SLX150)
SASEBO-RII	2011	LSI socket (QFP160)	N/A
SASEBO-GIII	2012	Kintex-7 (TBD)	Spartan-6 (XC6S45LX)

Table 2: Basic specifications of SASEBO-GII

Size	$120x140x1.6$ mm^3, FR-4, six layers
Devices	xc5vlx30/50-ffg334 (for cryptographic circuit) xc3s50a-ftg (for control and interface circuit)
Power supply	5.0 V USB bus power / 5.0 V DC power supply 1.0 V internal regulators Alternative 1.0 V supply line for the FPGA
Monitoring points	Surface-mounted shunt resistors (1Ω) are inserted at the V_{CORE}, V_{IO} and GND lines
Local bus	38-bit bus between the FPGAs
I/F	USB
Clocks	24 MHz oscillator for control device JTAG, SPI-ROM, User-controllable SelectMAP, ICAP

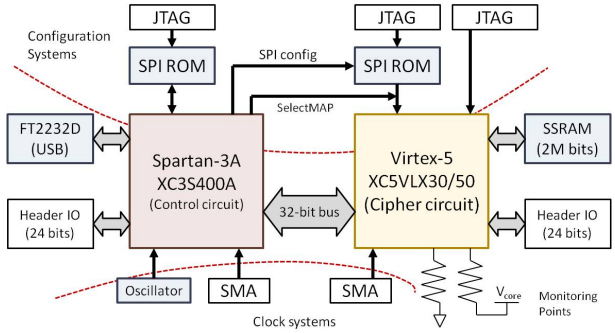

Figure 7: Block diagram of SASEBO-GII.

SASEBO-GII has two Xilinx FPGA devices—a Virtex-5 for cryptographic circuits and a Spartan-3A for interface and control circuits. Furthermore, there are two variants of Virtex-5, namely LX30 and LX50 for small and large logic circuits, respectively. Surface-mounted shunt resistors are soldered and SMA jumpers are inserted into the V_{CORE} and GND lines in order to improve the quality of power tracing. In addition, a surface-mounted device is chosen to reduce noise generated by the clock oscillator, whereas the previous SASEBO and SASEBO-G implementations use a PLL programmable crystal oscillator.

Power for operation can be supplied through the USB connector, or an external power source can also be used in cases where more stable power supply is necessary. The cryptographic device and the control device are equipped with their own V_{CORE} regulators, and the GND lines of the two parts are connected through inductors. This architecture also contributes towards further noise reduction. The size of SASEBO-GII has been reduced to 1/3 of that of SASEBO-G by removing the RS-232 interface, the monitoring points for power consumption for the control device, the FPGA configuration sequencer and the large header pins. As shown in the block diagram in Fig. 7, the wide local bus of SASEBO-G is emulated on the Virtex-5 FPGA. This simple and compact implementation also improves the quality of power tracing since it reduces the number of parasitic capacitances and resistances. In spite of its simplicity, SASEBO-GII provides high compatibility with SASEBO-G, and the same Verilog-HDL source code and control software [36], [37] designed for SASEBO-G can be used without modification for SASEBO-GII.

2.2.2 FPGA Configuration

SASEBO-GII allows for user-controllable configuration, where bitstreams are transmitted to the Virtex-5 SelectMAP interface or SPI-ROM through Spartan-3A. Thus, a JTAG cable is unnecessary for Virtex-5 configuration, although JTAG interfaces are still implemented for ordinary configuration and internal signal monitoring. Jumper pins on the board are used for selecting the configuration type.

Figure 8 illustrates the process of self-DPR of Virtex-5 through ICAP. In this case, a bitstream of a Partially Reconfigurable Module (PRM) is sent from the personal computer (PC) through Virtex-5. For the secure configuration, the integrity of the PRM bitstream should be checked, followed by decryption of the bitstream in the Virtex-5, after which the PRM is used to configure the device. The security of the DPR system with a single FPGA can therefore be examined with this configuration.

Figure 9 shows the configuration of Virtex-5 via the SelectMAP interface controlled by Spartan-3A. This type of configuration is useful for developing a device authentication protocol between the control logic and the FPGA. In addition to DPR, this configuration type can also be used for complete reconfiguration of the FPGA. Therefore, the security of completely reconfigurable systems can also be examined with this configuration setting.

Figures 10 and 11 show the configuration of Virtex-5 and Spartan-3A via the SPI, respectively. Configuration data are written to SPI-ROM through the JTAG interface or the FPGA. SPI-ROM is usually used for configuration during the booting process, in other words, the configuration data are automatically read from SPI-ROM after the system is powered on. If the FPGA writes configuration data to SPI-ROM, the function of the FPGA will be different the next time the system is booted. Additionally, SASEBO-GII can trigger an SPI-ROM configuration process while the system is operating, and therefore completely reconfigurable environments can be studied with this configuration.

2.3 Other Boards of the SASEBO Family

a) SASEBO and SASEBO-G/-B/-R: SASEBO, SASEBO-G, SASEBO-B and SASEBO-R are earlier members of the SASEBO family and were developed in collaboration with Tohoku University [37]. They have been discontinued and are currently unavailable on the market. SASEBO-G and SASEBO-R were replaced with SASEBO-GII and SASEBO-RII, respectively.

b) SASEBO-W: SASEBO-W was especially developed for studying and evaluating the security of smartcards. SASEBO-W is equipped with a card slot for a smartcard along with a Spartan-6 FPGA for implementing a relevant controller.

c) SASEBO-RII: SASEBO-RII is the latest version of the SASEBO-R series, and it was developed for ASIC evaluation. The architecture of SASEBO-RII is drastically different from that of SASEBO-R—the controlling FPGA is removed, and only an LSI socket is installed. SASEBO-RII is a daughter board of SASEBO-W, and SASEBO-W

[1]FPGA Mezzanine Card (Low-Pin Count).

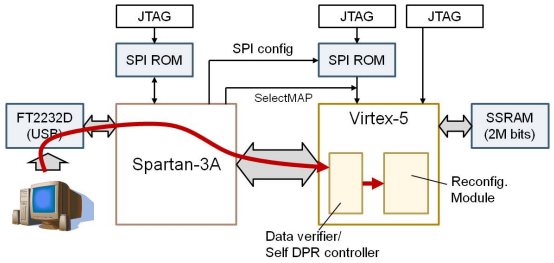

Figure 8: Dynamic partial reconfiguration (DPR) of Virtex-5.

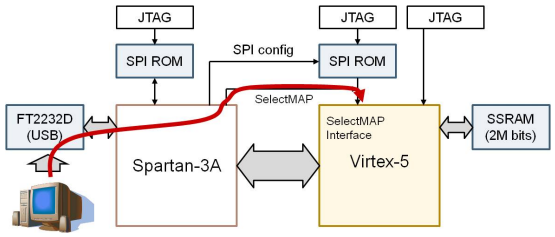

Figure 9: Virtex-5 configuration via the SelectMAP interface.

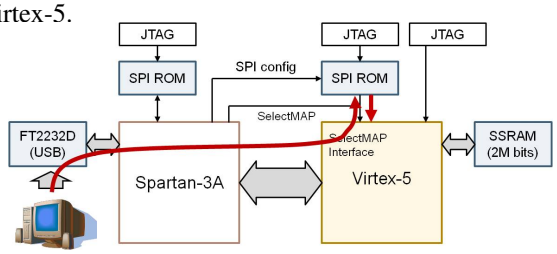

Figure 10: Virtex-5 configuration by implemented by updating SPI-ROM.

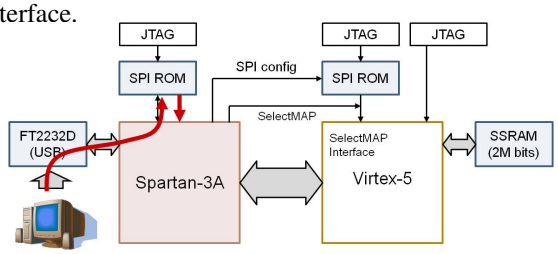

Figure 11: Spartan-3A configuration implemented by updating SPI-ROM.

is in charge of controlling the smartcard. The schematics of SASEBO-RII will be made available online under the condition that they be used for academic research, which would allow researchers to develop their own boards with LSI sockets of choice. This is expected to greatly reduce the cost of board development.

3. Pseudo-LFSR PUF

A PL-PUF is a delay PUF which is compact, efficient, multi-functional and resistant to attacks. PL-PUF does not contain a shift register; rather, it constitutes a large combinational logic based on the structure of LFSR. Figure 12 illustrates a 128-bit PL-PUF with the following primitive feedback polynomial [38]

$$x^{128} + x^{126} + x^{102} + x^{99} + 1. \quad (1)$$

Note that in PL-PUF the core logic (Fig 13) is not a register but an inverter, and thus PL-PUF constitutes a single combinational circuit. The output of PL-PUF oscillates since the output of the last core ($D_{out}(1)$) is fed back into the top core. The feedback signal is strongly affected by process variations in the device, and therefore the output of PL-PUF becomes sensitive to delays and consequently dependent on the device. The core logic does not necessarily have to be an inverter—it can be any combinational logic that efficiently extracts variations in the device.

PL-PUF realizes authentication based on a challenge-response pair (CRP). In the case of Fig. 12, the challenge is the 128-bit initial value supplied to the core logic, and the response (= ID) is the 128-bit output of the core logic. Note that the 128-bit ID is generated from a single 128-bit challenge, which is the remarkable novelty of PL-PUF realizing high throughput and high attack resistance.

After the initial value is set to each core logic, PL-PUF is activated for c clock cycles. This active cycle is referred to as an *active duration*, where the same PL-PUF can generate completely different outputs depending on the active duration c.

The features of PL-PUF can be summarized as follows.

- *Compactness*
  An inverter-based PL-PUF results in a small circuit. In the case of Fig. 12, it requires only 128 inverters and 3 XOR gates. By comparison, an arbiter-based PUF has two selector chains, and therefore a 128-stage arbiter PUF requires 256 multiplexers.
- *Efficiency*
  A PL-PUF efficiently outputs long IDs since all 128 bits of the ID are generated from a single 128-bit challenge. This is a notable advantage of the PL-PUF as compared to other PUFs, where only single-bit or several-bit output is generated from a long challenge. By comparison, an arbiter-based PUF usually requires 128 CRPs to obtain a 128-bit ID.
- *Multi-functionality*
  The output of the PL-PUF depends on the duration of the active clock cycles, and thus a single PL-PUF can be made to behave as multiple PUFs by changing the active duration. In other words, the challenge-response mapping of the PL-PUF can be easily changed without modifying its hardware structure. This property determines the unclonability of PL-PUF since cloning CRP mapping for all possible durations is considered impractical.

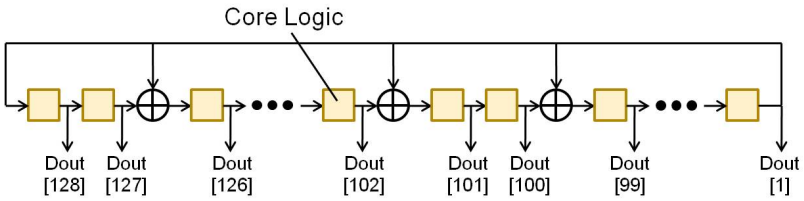

Figure 12: Structure of the PL-PUF.

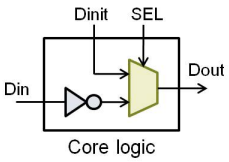

Figure 13: Structure of the core logic.

- *Reliability*
  A reliable PUF is expected to generate reproducible IDs which are unique to the device generating them. PL-PUF features both high reproducibility and uniqueness, as demonstrated below. In addition, the reliability of PL-PUF is configurable by changing the duration of the active clock cycles. Therefore, the user can choose a duration which corresponds to the preferred reliability.
- *Attack resistance*
  PL-PUF is expected to exhibit high resistance against attacks based on machine learning since modeling its delay would be exceedingly difficult. Furthermore, it outputs a 128-bit response at once from a single 128-bit challenge, in other words, the function of PL-PUF is

$$f : \{0,1\}^{128} \rightarrow \{0,1\}^{128} \quad (2)$$

  unlike the function of conventional PUFs

$$f : \{0,1\}^{128} \rightarrow \{0,1\}. \quad (3)$$

  Thus, learning the delay parameter from the 2^{128} output space would require an excessive number of CRPs, and thus it is considered impractical.

3.1 Experiments and Results

3.1.1 Quantitative and Statistical Analysis

First, we evaluated the performance of PL-PUF with respect to the quantitative indicators proposed in [34] (randomness, steadiness, correctness, diffuseness and uniqueness). The evaluation results are given in Table 3. Due to space limitations, only the results for Device 1 are given in the table. In the experiments, the active duration was varied between 1 and 16, and all performance indicators were in the range between 0 and 1, with 0 being the lowest and 1 being the highest.

Table 3: Performance of PL-PUF evaluated with respect to several quantitative indicators.

Active Duration	Randomness H	Steadiness S	Correctness C	Diffuseness D	Uniqueness U
1	0.984	0.982	0.979	0.988	0.656
2	0.975	0.966	0.960	0.987	0.728
3	0.964	0.954	0.947	0.985	0.746
4	0.967	0.925	0.913	0.989	0.755
5	0.966	0.878	0.859	0.990	0.766
6	0.944	0.804	0.775	0.988	0.772
7	0.969	0.726	0.686	0.989	0.776
8	0.960	0.622	0.572	0.988	0.772
9	0.967	0.516	0.460	0.985	0.773
10	0.964	0.415	0.357	0.978	0.771
11	0.966	0.324	0.269	0.974	0.760
12	0.964	0.253	0.203	0.958	0.756
13	0.964	0.200	0.155	0.950	0.744
14	0.962	0.165	0.126	0.929	0.739
15	0.965	0.145	0.109	0.914	0.738
16	0.963	0.131	0.097	0.900	0.734

As can be seen from the table, randomness and diffuseness are consistently high for all active durations. As a result, the entropy of PL-PUF is considered to be sufficiently high for cryptographic purposes. Also, the uniqueness of PL-PUF is markedly higher than that of the PUF in [39], and therefore PL-PUF is considered suitable for device identification as well. Furthermore, the steadiness and correctness are also high when the active duration is relatively short, although their values decrease as the active duration increases. This result indicates that PL-PUF can be suitable for device authentication when short active duration is used, while it can work as a high-quality random number generator in the case of long active duration.

3.1.2 Evaluation Results for Steadiness

Here, we assess the performance of PL-PUF with the biometric evaluation method, where the parameters *Fault Rejection Rate (FRR)* and *Fault Acceptance Rate (FAR)* are used as significant evaluation criteria. FRR represents the probability of a genuine input being rejected as a counterfeit one, while FAR represents the probability of a counterfeit input being accepted as a genuine one. FRR and FAR are derived from the intra-device Hamming distance (intra-HD) and the inter-device Hamming distance (inter-HD), where the former is the average HD between IDs generated by

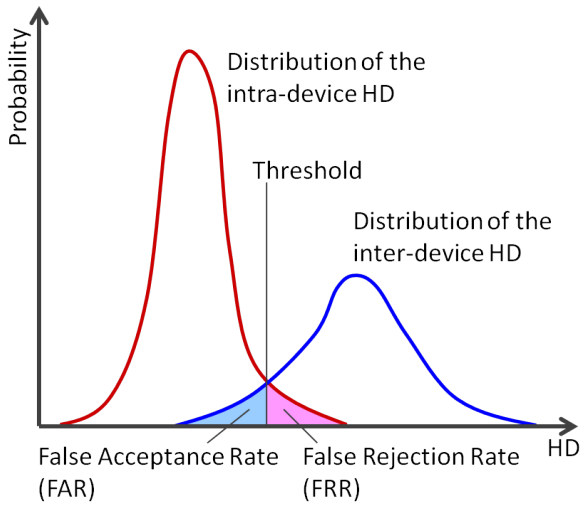

Figure 14: FRR and FAR of PUF.

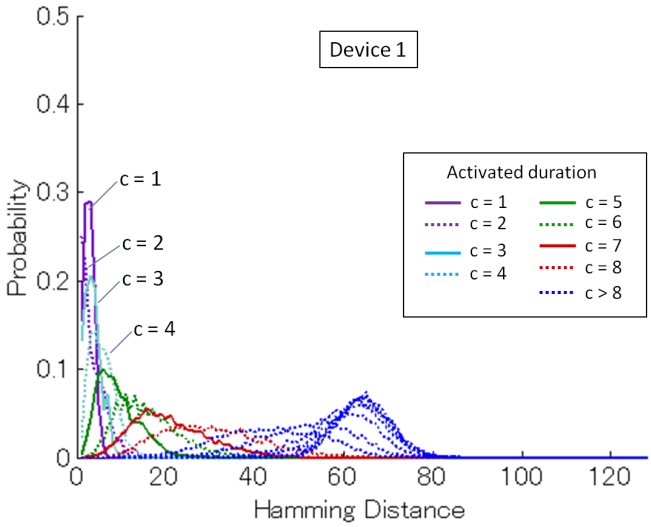

Figure 15: Distribution of intra-HD for Device 1.

the same device from the same challenge. If the intra-HD is small, the steadiness of the PUF is considered to be high. Furthermore, the inter-HD is the average HD between IDs generated by different devices from the same challenge. Since the ID length is 128 bits, the uniqueness of the PUF is considered to be high if the inter-HD is close to 64.

In Fig. 14, the curves on the left and right are the probability distributions of intra-HD and inter-HD, respectively. If the two curves cross, FAR and FRR take a non-zero value.

Figure 15 shows the probability distribution of the intra-HD for Device 1. As can be seen from the figure, when the active duration is short, the intra-device HD is rather small, and consequently the steadiness of the ID is high. On the other hand, the intra-device HD approaches 64 as the active duration increases, which indicates that the output of PL-PUF is almost purely random. This result shows that the active duration of PL-PUF should be reasonably short to obtain stable outputs.

3.1.3 Evaluation Results for Uniqueness

Figures 16-19 show the intra-HD for Device 1 and the inter-HDs between Device 1 and the other devices. The number of clock cycles for the active duration is set to 1, 4, 8 and 16, respectively. When the active duration is short, the shapes of the distributions of the intra- and inter-HD are sharp, and therefore FAR and FRR are both zero (Figs. 16 and 17). In Fig. 18, FAR and FRR become greater than zero but remain sufficiently low for Device 1 to be distinguishable from other devices. When the active duration becomes longer, Device 1 cannot be identified since its intra-HD and inter-HD distributions become indistinguishable from each other (Fig. 19).

As Figs. 16-19 show, the uniqueness of PL-PUF is high in the case of a sufficiently short active duration, although too short an active duration can fail to distinguish different PUFs.

4. Secure DPR Systems with PUF and AE

AE is a relatively new concept in cryptographic technology, providing both message encryption and authentication. Since both the confidentiality and the authenticity of bitstreams must be guaranteed, AE must be effectively applied to DPR systems. We developed a prototype of a secure DPR system using AES-GCM [14] and studied the relationship between the throughput and memory overhead of different AE modules [13]. As a result, we found that AE achieves high speed and area efficiency as compared with systems using separate encryption and authentication algorithms. However, the problem of the storage of the secret key remains since the secret key embedded into the chip can be extracted by an SCA attack. The use of PUFs is a promising approach for solving this problem.

The goal of our study is to build a secure DPR system by integrating AE and PL-PUF into the system. Such a DPR system is expected to be secure with respect to reverse engineering and hardware trojans since the bitstream of the system is protected by AE and therefore less vulnerable to SCA since PL-PUF eliminates the requirement that the secret key be stored in memory. Although there has been related work using PUF for protecting FPGA IP cores, PL-PUF is expected to realize higher throughput and considerably stronger protection against machine learning.

In this section, first we explain the AES-GCM algorithm, after which we show the implementation results and discuss the performance of our DPR system with AES-GCM. Fi-

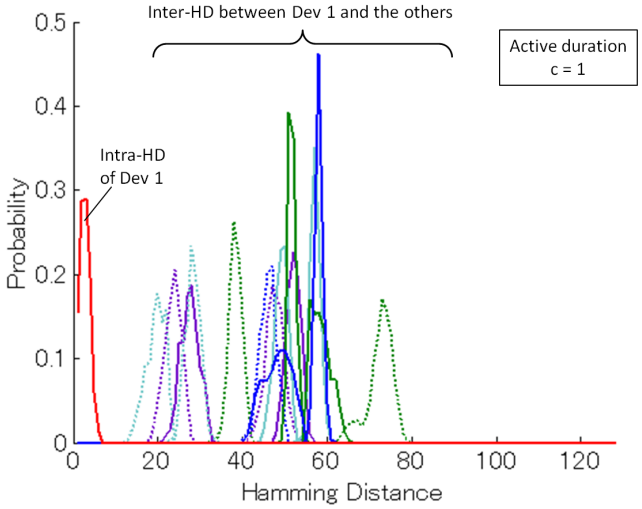

Figure 16: Distribution of the inter-HD for Device 1 for an active duration of 1.

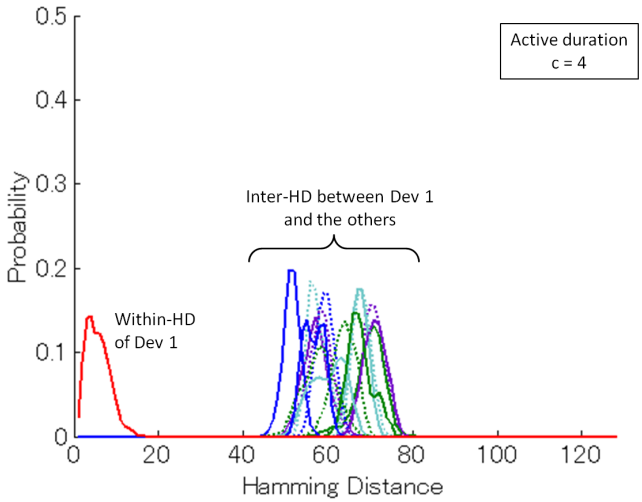

Figure 17: Distribution of the inter-HD for Device 1 for an active duration of 4.

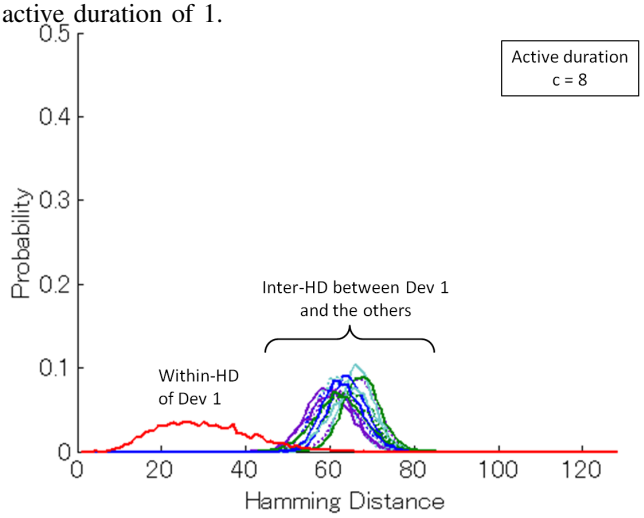

Figure 18: Distribution of the inter-HD for Device 1 for an active duration of 8.

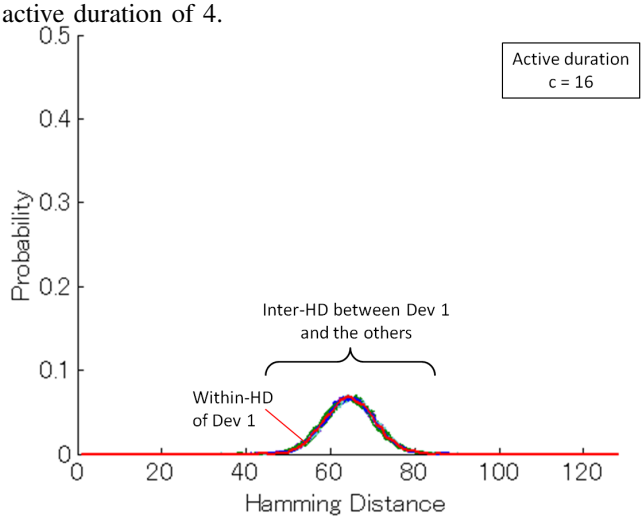

Figure 19: Distribution of the inter-HD for Device 1 for an active duration of 16.

nally, we present our ongoing project of a PUF-based secure video playback system.

4.1 AES-GCM

We chose AES-GCM [9], [10] as an AE algorithm for bitstream encryption and authentication. AES is a symmetric key block cipher algorithm standardized by the U.S. National Institute of Standards and Technology (NIST) [40]. While the previous standard (DES [41]) features a Feistel network architecture, AES employs a substitution-permutation network (SPN) architecture. The block length of AES is 128 bits, and the key length can be 128, 196 or 256 bits.

A block cipher algorithm can be applied to various modes of operation. GCM is one of the latest modes of operation standardized by NIST. Figure 20 shows an example demonstrating the operation of GCM.

The encryption and decryption scheme of GCM is based on the CTR mode of operation [42]. Thus, GCM can be highly parallelized and pipelined and is therefore suitable for hardware implementation, exhibiting a number of advantages ranging from compactness to high speed [43], [44]. There are other AE algorithms which are not necessarily suitable for hardware implementation as they cannot be parallelized or pipelined [45].

AES-GCM is an AE algorithm providing both message confidentiality and authenticity. GCM uses universal hashing in the finite field $GF(2^w)$ for generating a message authentication code (MAC). The additional merit of using $GF(2^w)$ is that the computational cost of multiplication under $GF(2^w)$ is lower than that for integer multiplication.

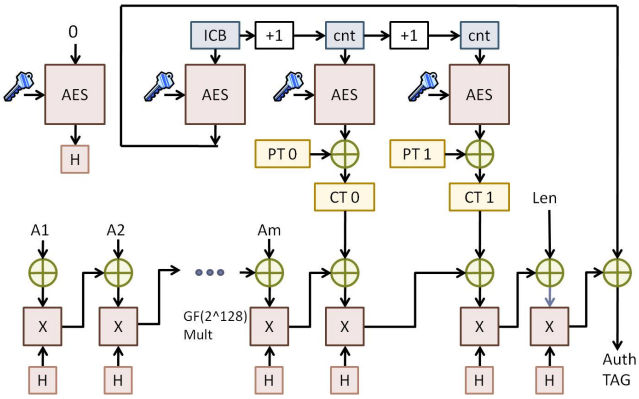

Figure 20: Example demonstrating the operation of the Galois/Counter Mode (GCM).

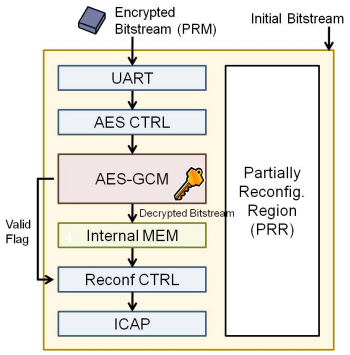

Figure 21: Overview of the proposed system with AES-GCM.

4.2 AES-GCM-based DPR System

Here, we introduce our AES-GCM-based DPR system [14]. Unlike other DPR systems, our system does not use an embedded processor to control partial reconfiguration. Rather, the input data and the ICAP control signals are directly connected to and controlled by the user logic. Thus, our system is free from the delay associated with processor buses. In Virtex-5, the maximum frequency of the ICAP interface is limited to 100 MHz, and thus the ideal throughput of the reconfiguration process is 3,200 Mbps.

Figure 21 shows a block diagram of the DPR system with bitstream encryption and authentication using AES-GCM. In this system, the lengths of the AES key and the initial vector are set to 128 bits and 96 bits, respectively.

As the main purpose of this study is to clarify the feasibility of AES-GCM for bitstream encryption and authentication, rather simple function blocks, for example, a 28-bit adder and a 28-bit subtractor, are used as PRM, which is connected to the static modules with two bus macros. The four most significant bits of the adder or the subtractor are output from PRM and connected to LEDs on the board. The PRR contains 80 slices, 640 LUTs and 320 registers. The size of the PRM bitstream is about 11KB.

The S-box of AES is implemented as a table using Block RAM. In AES-GCM, a 128-bit block is decrypted in 12 clock cycles. The last block of the message requires 12 clock cycles and an additional 10 clock cycles to calculate the authentication tag.

Table 4 shows the implementation results for the AES-GCM-based DPR system (PR-AES-GCM) along with AES-CBC and SHA-256-based DPR systems (PR-AES-SHA) for comparison. As the table shows in the case of PR-AES-GCM, the hardware resources used are fewer and the throughput is higher than for PR-AES-SHA.

4.3 Integration of a PUF into the DPR System

Since a PUF is considered a fingerprint of the device, it can be used for device authentication in a manner similar to biometrics. To perform biometric authentication, several CRPs are exchanged between the server and the device in the DPR system, in which the server knows the correct responses in advance. The actual responses are sent by the system to the server and compared to the correct responses. If the error rate is lower than a certain threshold, the system is successfully authenticated.

Note, however, that the simple use of a PUF provides neither a strict authentication scheme nor a solution to the problem of key exchange. In a secure DPR system, a (partial) bitstream of an FPGA is usually encrypted with a symmetric cipher. In light of the possibility of an SCA attack, the key should not be stored in the device in advance. Therefore, the key must be *generated* in the device in some way. Here, note that a PUF cannot provide exact reproducibility since the output of the PUF is affected by random fluctuations in the device. The sole use of a PUF cannot generate identical keys from the same challenge set, and thus error correction code (ECC) is often used in key generation. In this regard, a scheme known as a *fuzzy extractor* [46] is widely used for ECC-based key generation [31], [47], and other key generation methods have been recently reported, such as in [48] and [49].

As an ongoing project, we are developing a video playback system based on AE and PUFs. The development platform is SASEBO-GIII, and an FMC daughter board with an LSI socket for implementing ASIC PUFs is currently being developed. An off-the-shelf FMC board is used for HDMI input/output ports. Error correction as well as the computation of hashes and other parameters in the fuzzy extractor are implemented on Kintex-7 on a SASEBO-GIII. Figure 22 shows the development platform, including SASEBO-GIII and the HDMI FMC board.

It should be noted that an SCA attack against a fuzzy extractor was recently reported [50]. We believe that SASEBO-GIII is the most suitable platform for investigating the security of the proposed PUF-based video playback system since it supports simple and straightforward implementation

Table 4: Comparison of the performance of secure PR systems (14,112-byte PRM).

System	Device	Slice	Authentication	Decryption	Configuration	Overall
PR-AES-GCM	XC5VLX50T	2,687*	106.43 μs 1,067 Mbps		35.3 μs 3,200 Mbps	141.73 μs 797 Mbps
PR-AES-SHA256	XC5VLX50T	2,730*	160.97 μs 701 Mbps	97.14 μs 1,164 Mbps	35.3 μs 3,200 Mbps	196.27 μs 575 Mbps

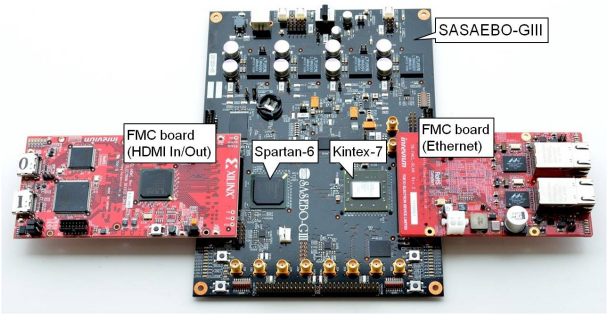

Figure 22: A PUF-based video playback system.

of SCA experiments. We are currently in the process of implementing the entire system, and the plans for future work include testing the security and feasibility of the system.

5. Conclusion

This paper introduced certain security issues associated with partial reconfiguration of FPGAs together with studies on counteracting these issues by using SASEBO, PUFs and secure DPR systems. Since FPGA bitstreams are electronic data downloaded from a host computer or the Internet, they are always susceptible to problems such as piracy, reverse engineering, tampering and hardware trojans. Authenticated encryption (AE), which guarantees both the confidentiality and the authenticity of the encrypted data, can serve as a solution to these problems, however, a recently reported type of attack referred to as SCA poses serious concern for the security of cryptographic systems. One solution to SCA attacks might be a PUF which generates device-specific IDs by using process variation of the device.

First, we introduced a family of boards named SASEBO (Side-channel Attack Standard Evaluation Board). The latest version of SASEBO, SASEBO-GIII, with the newest FPGA Kintex-7, will be made available in 2012 or 2013.

In addition, our PL-PUF is a compact and secure delay-based PUF achieving high throughput. Unlike other PUFs, PL-PUF outputs an N-bit response from an N-bit challenge, which enables fast and attack-resistant key generation. The values of both FAR and FRR of PL-PUF are rather small in the case of a short active duration, and therefore PL-PUF is suitable for device identification as well as for key generation with fuzzy extractors.

A direction of future work is to develop an entire video playback DPR system including AE, PUF, and fuzzy extractors along with the video decoders.

Acknowledgements

The parts of this work related to SASEBO, SASEBO-G, -B, -R and -GII was funded by the Ministry of Economy, Trade and Industry (METI), Japan. In addition, the part of this work related to SASEBO-W was funded by the Strategic International Research Cooperative Program (SICP), the Japan Science and Technology Agency (JST). Finally, the part of this work related to SASEBO-GIII was funded by the Core Research for Evolutional Science & Technology (CREST), JST.

References

[1] Y. Hori, H. Yokoyama, H. Sakane, and K. Toda, "A secure content delivery system based on a partially reconfigurable FPGA," *IEICE Trans. Inf.&Syst.*, vol. E91-D, no. 5, May 2008, (to be publised).

[2] C. Albrecht, J. Foag, R. Koch, and E. Maehle, "DynaCORE—a dynamically reconfigurable coprocessor architecture for network processors," in *PDP 2006*, 2006, pp. 101–108.

[3] M. Rummele-Werner, T. Perschke, L. Braun, M. Hubner, and J. Becker, "A FPGA based fast runtime reconfigurable real-time multi-object-tracker," in *ISCAS 2011*, 2011, pp. 853–856.

[4] J. Becker, M. Hubner, G. Hettich, R. Constapel, J. Eisenmann, and J. Luka, "Dynamic and partial FPGA exploitation," *Proc. IEEE*, vol. 95, no. 2, pp. 438–452, 2007.

[5] A. Akoglu, A. Sreeramareddy, and J. Josiah, "Fpga based distributed self healing architecture for reusable systems," *Cluster Computing*, vol. 12, no. 3, pp. 269–284, 2009.

[6] A. Mecwan and N. Gajjar, "Implementation of software defined radio on FPGA," in *NUiCONE 2011*. IEEE, 2011.

[7] A. Moradi, A. Barenghi, T. Kasper, and C. Parr, "On the vulnerability of FPGA bitstream encryption against power analysis attacks—extracting keys from Xilinx Virtex-II FPGAs," Cryptology ePrint Archive, 2011.

[8] A. Moradi, M. Kasper, and C. Paar, "On the portability of side-channel attacks –an analysis of the Xilinx Virtex 4 and Virtex 5 bitstream encryption mechanism," Cryptology ePrint Archive, 2011.

[9] D. A. McGrew and J. Viega, "The Galois/counter mode of operation (GCM)," May 2005, http://csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html.

[10] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*, SP 800-38D ed., National Institute of Standards and Technology, Nov. 2007.

[11] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," in *ASIACRYPT 2000*, 2000, pp. 531–545.

[12] A. Seffrin and S. Huss, "Ensuring secure information flow in partially reconfigurable architectures by means of process algebra analysis," in *TrustCom 2011*, 2011, pp. 443–450.

[13] Y. Hori, A. Satoh, H. Sakane, and K. Toda, "Bitstream encryption and authentication using AES-GCM in dynamically reconfigurable systems," in *IWSEC 2008*, 2008, pp. 261–278.

[14] ——, "Bitstream encryption and authentication with aes-gcm in dynamically reconfigurable systems," in *FPL 2008*, 2008, pp. 23–28.
[15] L. Bossuet and G. Gogniat, "Dynamically configurable security for SRAM FPGA bitstreams," *Int. J. Embedded Systems*, vol. 2, no. 1/2, pp. 73–85, 2006.
[16] M. M. Parelkar, "Authenticated encryption in hardware," Master's thesis, George Mason University, 2005.
[17] A. S. Zeineddini and K. Gaj, "Secure partial reconfiguration of FPGAs," in *ICFPT'05*, 2005, pp. 155–162.
[18] T. Kean, "Secure configuration of field programmable gate arrays," in *Field-Programmable Logic and Applications*, 2001, pp. 142–151.
[19] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, RSA, DSS, and other systems," in *CRYPTO'96*, 1996, pp. 104–113.
[20] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *CRYPTO'99*, 1999, pp. 388–397.
[21] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model," in *CHES 2004*, 2004, pp. 16–29.
[22] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *CHES'01*, vol. LNCS 2162, 2001, pp. 251–261.
[23] J. J. Quisquater and D. Samyde, "Electromagnetic analysis (EMA): Measures and countermeasures for smart card," in *e-Smart'01*, vol. LNCS 2140, 2001, pp. 200–210.
[24] B. Gierlichs, L. Batina, and P. Tuyls, "Mutual information analysis," in *CHES2008*, 2008.
[25] S. Akashi, K. Toshihiro, and S. Hirofumi, "Secure implementation of cryptographic modules—development of a standard evaluation environment for side channel attacks," *Synthesiology*, vol. 3, no. 1, pp. 56–65, 2010.
[26] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *CCS 2002*. ACM, 2002, pp. 148–160.
[27] R. Maes and I. Verbauwhede, "Physically unclonable functions: A study on the state of the art and future research directions," in *Towards Hardware-Intrinsic Security*, A.-R. Sadeghi and D. Naccache, Eds. Springer-Verlag, 2010, ch. 1, pp. 3–37.
[28] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Trans. VLSI Syst.*, vol. 13, no. 10, pp. 1200–1205, 2005.
[29] G. E. Suh and S. Devadas, "Physical physical unclonable functions for device authentication and secret key generation," in *DAC'07*, 2007, pp. 9–14.
[30] D. Suzuki and K. Shimizu, "The glitch PUF: A new delay-PUF architecture exploiting glitch shapes," in *Proc. CHES2010*, 2010, pp. 366–382.
[31] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *CHES'07*, 2007, pp. 63–80.
[32] S. S. Kumar, J. Guajardo, R. Maesyz, G.-J. Schrijen, and P. Tuyls, "The butterfly PUF," in *HOST'08*, 2008, pp. 67–70.
[33] E. Ozturk, G. Hammouri, and B. Sunar, "Physical unclonable function with tristate buffers," in *ISCAS'08*, 2008, pp. 3194–3197.
[34] Y. Hori, H. Kang, T. Katashita, and A. Satoh, "Pseudo-LFSR PUF: A compact, efficient and reliable physical unclonable function," in *ReConFig 2011*, 2011, pp. 223–228.
[35] *Virtex-5 User Guide*, Xilinx, Inc., 2007.
[36] "Side-channel attack standard evaluation board (sasebo)," http://www.rcis.aist.go.jp/special/SASEBO/, research Center for Information Security, National Institute of Advanced Industrial Science and Technology.
[37] "Cryptographic hardware project," http://www.aoki.ecei.tohoku.ac.jp/crypto/, aoki Lab., Tohoku University.
[38] M. George and P. Alfke, "Linear feedback shift registers in Virtex devices," Xilinx apprication note XAPP210, 2007.
[39] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh, "Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs," in *Proc. ReConFig2010*, 2010, pp. 298–303.
[40] U.S. Department of Commerce/National Institute of Standards and Technology, "Announcing the advanced encryption standard (AES)," FIPS PUB 197, Nov. 2001.
[41] ——, "Data encryption standard (DES)," FIPS PUB 46-3, 1999.
[42] M. Dworkin, *Recommendation for Block Cipher Modes of Operation*, SP 800-38A ed., National Institute of Standards and Technology, Dec. 2001.
[43] A. Satoh, "High-speed parallel hardware architecture for Galois counter mode," in *ISCAS'07*, 2007, pp. 1863–1866.
[44] A. Satoh, T. Sugawara, and T. Aoki, "High-speed pipelined hardware architecture for Galois counter mode," in *ISC'07*, 2007, pp. 118–129.
[45] D. A. McGrew and J. Viega, "The security and performance of the Galois/counter mode (GCM) of operation," in *INDOCRYPT 2004*, 2004, pp. 343–355.
[46] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM Journal of Computing*, vol. 38, no. 1, pp. 97–139, 2008.
[47] C. Bosch, J. Guajardo, A.-R. Sadegh, J. Shokrollahi, and P. Tuyls, "Efficient helper data key extractor on FPGAs," in *CHES'08*, 2008, pp. 181–197.
[48] M.-D. M. Yu and S. Devadas, "Secure and robust error correction for physical unclonable functions," *IEEE Design and Test of Computers*, vol. 27, no. 1, pp. 48–65, 2010.
[49] M.-d. M. Yu, R. Sowell, A. Singh, D. M'Raïhi, and S. Devadas, "Performance metrics and empirical results of a PUF cryptographic key generation ASIC," in *HOST 2012*, 2012.
[50] D. Merli, D. Schuster, F. Stumpf, and G. Sigl, "Side-channel analysis of pufs and fuzzy extractors," in *Trust 2011*, 2011, pp. 33–47.