SeReCon: a secure reconfiguration controller for self-reconfigurable systems

Krzysztof Kępa*, Fearghal Morgan and Krzysztof Kościuszkiewicz

Bio-Inspired and Reconfigurable Computing (BIRC) Group,
Electrical and Electronic Engineering, National University of Ireland,
Nun's Island, Galway, Ireland
Fax: + 353 91 494511
E-mail: krzysztof.kepa@poczta.fm
E-mail: fearghal.morgan@nuigalway.ie
E-mail: kosciuszkiewicz@ieee.org
*Corresponding author

Tomasz Surmacz

Institute of Computer Engineering, Control and Robotics (CECR),
Wroclaw University of Technology,
Janiszewskiego 11-17, 50-372 Wrocław, Poland
Fax: + 48 71 3212677
E-mail: tsurmacz@ict.pwr.wroc.pl

Abstract: A risk of covert insertion of circuitry into reconfigurable computing (RC) systems exists. This paper reviews risks of hardware attack on field programmable gate array (FPGA)-based RC systems and proposes a method for secure system credentials generation (unique, random and partially anonymous) and trusted self-reconfiguration, using a secure reconfiguration controller (SeReCon) and partial reconfiguration (PR).

SeReCon provides a root of trust (RoT) for RC systems, incorporating novel algorithms for security credentials generation and trusted design verification. Credentials are generated internally, during system certification. The private credential element never leaves the SeReCon security perimeter.

To provide integrity-maintaining self-reconfiguration, SeReCon performs analysis of each new IP core structure prior to reconfiguration. An unverified IP core can be used provided that its spatial isolation is retained. SeReCon provides encrypted storage for installed IP cores.

Resource usage for a prototype SeReCon system is presented. The protection provided by SeReCon is illustrated in a number of security attack scenarios.

Keywords: field programmable gate array; FPGA; partial reconfiguration; reconfigurable computing; trusted computing; critical embedded systems; design security; design assurance; design integrity; self-reconfiguration.

Reference to this paper should be made as follows: Kępa, K., Morgan, F., Kościuszkiewicz, K. and Surmacz, T. (2010) 'SeReCon: a secure reconfiguration controller for self-reconfigurable systems', *Int. J. Critical Computer-Based Systems*, Vol. 1, Nos. 1/2/3, pp.86–103.

Biographical notes: Krzysztof Kępa is a PhD candidate in College of Engineering and Informatics at National University of Ireland, Galway, Ireland. He received his MSc in Computer Science in 2005 from Wroclaw University of Technology (Wroclaw, Poland). In 2005, he joined Applied Optics Group at NUI Galway (Galway, Ireland) as Digital Systems Engineer. Since 2006, he is with Bio-inspired Electronics and Reconfigurable Computing (BIRC) Group, NUI Galway. His research interests include security of embedded systems, reconfigurable computing systems and hardware-software co-design.

Fearghal Morgan is the Director of the Bio-Inspired Electronics and Reconfigurable Computing (BIRC) Group at the National University of Ireland, Galway, Ireland (birc.nuigalway.ie). He received his BSc in Electrical and Electronic Engineering in 1978 and his PhD in 1986 (Queens University Belfast). He spent seven years as a Digital Systems Networks Products Designer with Digital Equipment Corporation (DEC), Ireland. His academic career includes four years in the Institute of Technology, Tallaght, Dublin prior to joining NUI Galway in 1996.

Krzysztof Kościuszkiewicz is a PhD candidate in College of Engineering and Informatics at National University of Ireland, Galway, Ireland. He received his MSc in Computer Science in 2005 from Wroclaw University of Technology (Wroclaw, Poland). His research interests are in the areas of run-time support for reconfigurable computing systems, formal verification methods and design of programming and hardware description languages.

Tomasz Surmacz is a Researcher and a Teacher. He received his MSc in Computer Engineering in 1994 after studying at Wroclaw University of Technology, Poland, Trinity College Dublin, Ireland and Staffordshire University, UK. In 2004, he received his PhD in Computer Science from Wroclaw. His areas of interest and research include security and reliability of computers and computer networks, development and integration of network services, concurrent programming, microcontrollers and embedded systems. He is currently employed at Institute of Computer Engineering, Control and Robotics at Wroclaw University of Technology.

1 Introduction

This paper proposes a novel method for secure system credentials generation (unique, random and partially-anonymous) and integrity protection (using trusted self-reconfiguration and design verification) in partial reconfiguration (PR) computing systems. The paper proposes and describes a secure reconfiguration controller (SeReCon) architecture.

A recent Department of Defense report (DSBTF, 2005) identifies several trends contributing to the threat of covert insertion of circuitry into computing hardware. Modifying hardware provides attackers with a fundamental advantage over software-based attacks (Agrawal et al., 2007; King et al., 2008). Attacks at the hardware level are more difficult to detect and to prevent than software changes. Defending against hardware intrusion is more difficult, as the offender has control over all system layers, including the software stack.

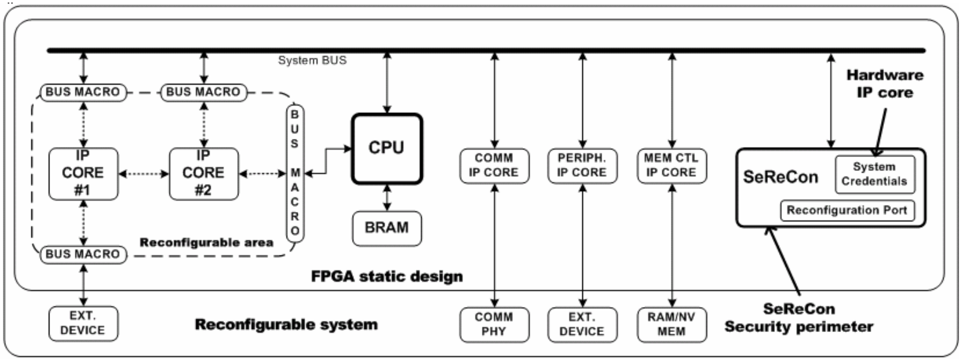
Reconfigurable computing (RC) is defined as: “the study of computation using reconfigurable devices” [Bobda, (2007), p.9]. RC systems offer hardware acceleration, reduced time-to-solution, reduced power consumption and improved fault-tolerance with respect to production defects. RC systems leverage intellectual property (IP) R&D costs, while providing benefits usually associated with expensive high-performance computing systems.

In RC systems, no protection layer exists below the system hardware layer. Without protective measures, reconfigurable hardware could be exposed to a range of attacks, with the addition of only a small amount of covert-inserted reconfigured hardware (Kepa et al., 2009). King et al. (2008) illustrate that an attacker can design hardware to support multiple attacks and demonstrate this concept using a system implemented in a field programmable gate array (FPGA).

FPGA-based RC systems are extensively used for rapid prototyping, in-system customisation, multi-modal computation and adaptive computing systems. Bobda (2007) surveys application domains which significantly benefit from the use of RC. The list includes pattern matching, video streaming, digital signal processing (DSP) using distributed arithmetic, adaptive controllers, adaptive cryptographic systems, software defined radio and high performance computing.

The runtime reconfiguration (RTR) paradigm enables RC systems (i.e. FPGAs) to perform (self-) reconfiguration (SR) (Blodget et al., 2003). SR can occur not only at the typical software level, but also at the configware level (Hartenstein, 2001). Configware defines a virtualised hardware platform on which the software is executed. Figure 1 presents the block diagram of the SeReCon-enabled SR system. Typically, the SR system contains a microcontroller (*CPU*), a number of application-specific accelerators (*IP core 1*, *2*) and a number of interfaces, e.g. for communication (*COMM*), external memory access (*EXT MEM*), device-specific IO (*PERIPH*) and self-reconfiguration (*ICAP*). SeReCon is an additional IP core connected to the base system.

Figure 1 Block diagram of the self-reconfigurable system including SeReCon



Note: SeReCon provides the system root of trust (RoT) and implements a two-phase integrity-maintaining SR.

The enabling technology for SR is PR, offered by some FPGA vendors (i.e. Atmel or Xilinx). PR provides full access to the FPGA configuration memory during system runtime. A system's ability to self-reconfigure using PR allows the software layer to modify the hardware configuration during runtime, e.g. to insert new hardware IP, without resetting the system. PR is facilitated using an internal configuration access port (ICAP) (Xilinx, 2005).

Reconfigurable systems typically include a number of IP cores. This leads to an unprecedented flexibility and freedom in adapting to temporal changes within the system, e.g. fault-tolerance (Streichert et al., 2006) or environmental adaptivity (Steiner and Athanas, 2009). However, PR consequently introduces risks to hardware system security on a scale associated to date only with the software domain. Extending hardware support for intrusion detection and handling is therefore required.

The most strict adversary model in embedded system design assumes that a security risk exists where a device is held by one entity and where secrets (i.e. IP) within the device are controlled (owned) by another entity. A secure system goal is to design systems which an attacker (user) cannot subvert, either by malice, accident, or trickery.

While the methodology of IP core reuse reduces design time and associated cost, the intensive growth of the market for pre-designed modules introduces concerns about protection of design IP rights and integrity of designs incorporating third party IP cores. Ideally, each of the design components should be formally specified, tested and verified, followed by certification by an external trusted authority (TAut). In reality, IP components are typically created through in-house design reuse, obtained from third party IP vendors, or generated using automated core generation tools, e.g. Xilinx CoreGen.

Design reuse in the RC system design flow results in IP cores of increasing complexity. As a consequence, attack methods can be generalised and are becoming obscured by the complexity of the RC system.

Current FPGA security measures implemented by vendors include:

a low cost device security (Trimberger, 2007); varying from no security to security-by-obscurity, e.g. closed bitstream format, design obfuscation etc.

b high-end device security; usually employing strong encryption with volatile, battery-backed tamper-proof key storage, configuration scrubbing (Drimer et al., 2008), complemented by Security Monitor IP core available to authorised users (McLean and Moore, 2007).

This paper proposes a novel method for secure system credentials generation and trusted self-reconfiguration in order to provide system integrity in RC systems using PR.

The architecture of an embedded FPGA-based SeReCon is described. SeReCon exploits PR and the trusted computing (TC) paradigm and performs autonomous analysis of the structure of IP cores prior to reconfiguration. This guarantees isolation of the RC system sub-modules and enforces inter-module communication only through module interfaces explicitly defined by the IP Vendor (IPVend). Thus, unverified IP cores can be used so long as the core provides an acceptable functionality and its spatial isolation is retained.

SeReCon incorporates two novel algorithms for building a system RoT and a two-phase integrity-maintaining self-reconfiguration. The protection provided by SeReCon is illustrated in a number of security attack scenarios. SeReCon aims to protect the integrity of self-reconfigurable systems, initially implemented using Xilinx technology, by mediating access to the FPGA ICAP and analysing incoming reconfiguration requests and IP cores during runtime. SeReCon employs authentication and encryption in order to provide authenticated protection to system security credentials (further referred to as *credentials*) and facilitate secure storage for analysed IP cores. SeReCon credentials are generated internally during the certification process and never leave the security perimeter of the SeReCon IP core. SeReCon assumes the implementation of a proposed minor modification to the FPGA fabric.

A prototype SeReCon system has been implemented and resource usage has been presented. The SeReCon implementation provides a generic, fixed footprint, single point of entry, public IP core, which manages runtime access to Xilinx FPGA configuration memory via the ICAP.

The paper is organised as follows. Section 2 reviews risks of hardware attack on self-reconfigurable computing systems and summarises reported work on the protection of RC hardware. Section 3 describes the various players which interact during the life cycle of a RC system. Section 4 proposes the SeReCon architecture. Section 5 reports on the SeReCon-enabled SR prototype implementation, highlights hardware-software partitioning issues and provides detailed insight into the operation of a prototype PR RC system using SeReCon. Section 6 concludes the paper and proposes future work.

2 Previous work

This section reviews risks of hardware attack on self-reconfigurable computing systems and summarises reported work on the protection of RC hardware.

RC system integrity protection deals with issues of malicious bitstream eavesdropping, device tamper-resistance etc. This section summarises reported work on the protection of RC system hardware. RC security counter-measures are mainly applied to high assurance systems. Hadzic et al. (1999) describe the threat of hardware viruses in FPGAs. Kean (2001) and Bossuet et al. (2006) highlight the vulnerability of volatile FPGAs to IP piracy and reverse engineering and propose bitstream encryption as a countermeasure. Wollinger et al. (2004) and Gogniat et al. (2006) survey FPGA security issues. Valette et al. (2006) list FPGA security features and emphasise configuration memory programming as the point of least security. Drimer (2007a) examines a wide range of attack mechanisms and countermeasures. Ravi et al. (2004) survey security challenges facing embedded systems.

In order to protect design integrity and design tampering of non-PR designs, encryption is a viable option which has been adopted by electronic design automation (EDA) tools and FPGA vendors (Xilinx, 2005). However, for PR designs, the existing ICAP does not fully protect against unrestricted FPGA configuration memory read back. Protection measures include imposing a block on the ICAP function when bitstream encryption is used (Xilinx, 2005). Under certain circumstances (e.g. where design IP protection is required, in high assurance systems design etc.) the FPGA vendor advises against the use of ICAP (Xilinx, 2007).

A common security model for trusted systems design is to trust the on-chip environment while assuming that the off-chip environment is untrustworthy (Suh et al., 2007). TC is a relatively new approach to system protection, proposed in 2003 by the Trusted Computing Group (TCG, 2003). TC introduces the idea of a hardware device capable of attesting, in a trustworthy way, certain system properties, thus establishing a RoT. The RoT in a secure system is defined as a component that must always behave in a defined manner, since its misbehaviour cannot be detected. The RoT contains at least the functions to enable a description of the system characteristics (i.e. system state) that affects the trustworthiness of the system, e.g. loaded OS modules, device drivers, etc. In TC and RC, the RoT may be based on a tamper-proof hardware element within the FPGA fabric. The RoT in RC systems can be partially implemented within user logic (configware), to provide a flexible security mechanism. The SeReCon proposed in this paper is such a system.

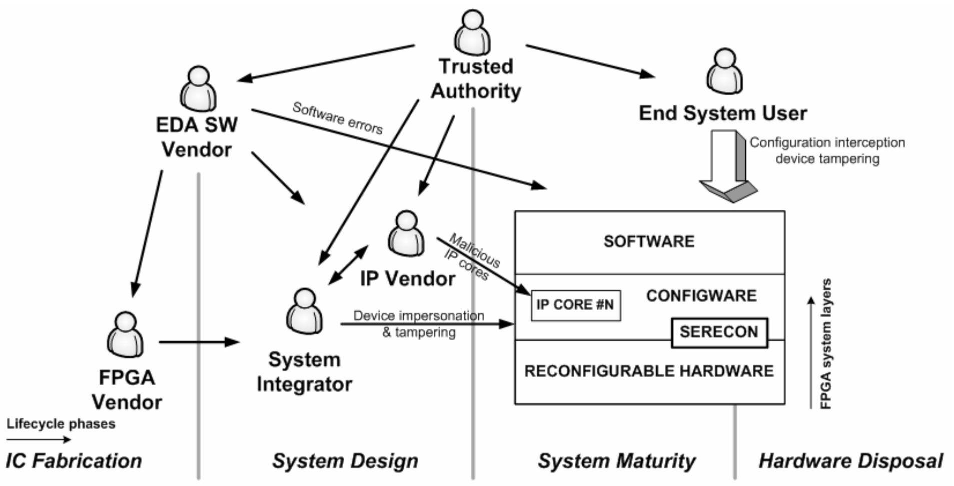
TC scheme for embedded RC systems has been previously reported by Glas et al. (2008). Glas proposes a protection model built upon trusted configuration attestation of the RC system state. The model assumes the use of certified and thus trusted modules. However, the growing number and complexity of available third party IP cores increases the risk of undetected malicious interaction even between certified cores.

In Suh et al. (2007), the authors discuss the AEGIS, a secure, single-chip processor and describe the techniques used to execute private and authenticated software from untrusted off-chip memory. Complementary work on IP core isolation is reported by Huffmire et al. (2007) and McLean and Moore (2007). The former does not support PR systems. The latter one is available only to authorised users (Xilinx, 2008).

Even if the IP core source code is available, it is vital to assure that the EDA tools used to produce the FPGA configuration bitstream are secure. Testing techniques can be used to show the presence of errors, but never to show the absence of errors (Dijkstra, 1979). Thompson (1984) discusses this issue and concludes that "You can't trust code that you did not totally create yourself. … No amount of source-level verification or scrutiny will protect you from using untrusted code".

This paper proposes a novel method of trusted design verification in order to provide system integrity protection in SR systems. The architecture of a novel embedded FPGA-based SeReCon is described. Figure 1 illustrates the block diagram of the SeReCon-enabled SR system. SeReCon incorporates novel algorithms for building a system RoT and a two-phase integrity-maintaining self-reconfiguration process.

Figure 2 Stack model, associated players and four phases of the reconfigurable system lifecycle



3 Reconfigurable system lifecycle

This section describes the various players which interact during the lifecycle of a RC system. Figure 2 illustrates the typical four-phase lifecycle, the interactions between the various parties (which introduce multilevel risks associated with the design flow and the RC system itself). The stack model of the SR system is also illustrated. Trust between players is limited. IP and EDA tool vendors seek appropriate IP protection against unauthorised design cloning, overbuilding and reverse engineering. Design houses seek methods to provide effective system security to protect design integrity in the field.

TAut is an authorisation and/or certification centre. TAut mediates the communication between players in order to provide the required element of trust. TAut is assumed to be trustworthy for all other entities and is usually not involved in the system development process.

The end-system user (user) is an end-customer who operates the RC system, possibly in hostile environments. The user requires the system to be secure, but could also try to gain personal profit by attempting to circumvent the implemented security countermeasures. At the end of system lifecycle, software can be erased, but the hardware platform often remains intact. The hardware recycling process can reveal some sensitive data, e.g. permanently embedded encryption keys. Under certain circumstances, even volatile memory can retain data (Skorobogatov, 2002; Tuan et al., 2007). This property may lead to disclosure of data or algorithms.

System integrator (SysInt) designs the RC system and provides it to the User. SysInt can issue a product upgrade in the field. A typical RC system consists of custom elements and multiple third party IP cores.

FPGA fabric vendor (FVend) provides the FPGA fabric. Risks involved with outsourcing IC fabrication are detailed in (DSBTF, 2005). Usually the FVend keeps the implementation details of the fabric confidential and guarantees quality of service and compliance to the FPGA specification. Verification by TAut is required to ensure that undocumented access to configuration memory is not possible therefore guarding against Trojan IC circuits (Agrawal et al., 2007).

EDA tool vendor (TVend) provides software tools for other parties and strives to ensure software quality. TVend can develop a strong reputation, based on long-term trusted activity, but cannot be trusted entirely (Thompson, 1984).

IPVend is an external entity which provides reusable components (IP cores) for the RC system. IPVend wishes to protect its own design secrets. IPVend is not directly involved in the system design process and is only aware of the system requirements to be met by its IP cores. IPVend guarantees compliance of the IP design to the specification.

The following assumptions have been made in defining the SeReCon system model:

a the FPGA device is trusted (i.e. Trojan-free) and provides hardware support for RoT (details are described in Section 4)

b the RC system comprises a number of integrated IP cores, configured using PR

c the IPVend explicitly declares some of IP core resources to be used as its communication interfaces

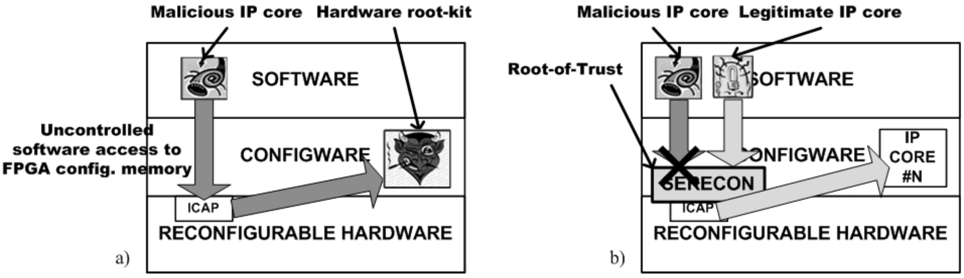
d IP cores are not trusted and their placement on the FPGA cannot overlap each other

e subliminal channel and side-channel attacks are not considered.

4 SeReCon architecture

This section proposes the SeReCon architecture. A novel algorithm is proposed for generating credentials in order to establish the secure RoT. SeReCon performs requested system reconfiguration on behalf of the system software. SeReCon aims to protect integrity of the RC system by mediating access to the ICAP and by analysing incoming reconfiguration requests during runtime. A two-phase self-reconfiguration process is implemented in order to improve performance of IP core activation. Figure 3a illustrates the stack model of a typical SR system and highlights the risks of unprotected software access to the reconfiguration interface. Figure 3b illustrates the proposed SeReCon-enabled SR system stack model.

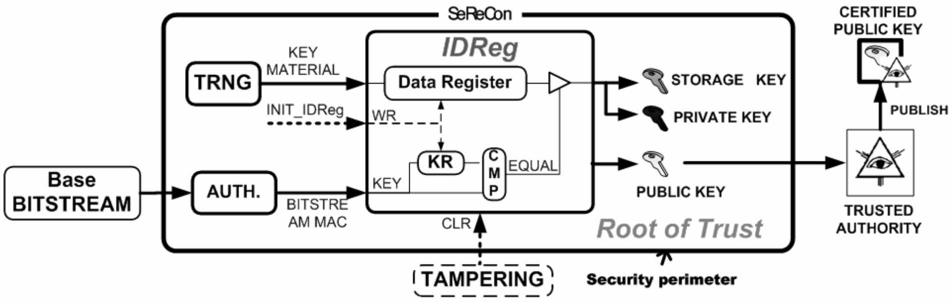
Figure 3 (a) Stack model of a self-reconfigurable system and risk of unprotected software access to the reconfiguration interface (b) the proposed SeReCon-enabled self-reconfigurable system stack model



In order to fulfil the TC definition of a system's RoT, SeReCon serves reconfiguration requests received from the software layer and interrupts the reconfiguration process when potentially malicious configuration is detected. SeReCon can be likened to a trusted OS boot-loader in the TC domain. SeReCon provides a facility to perform secure self-reconfiguration (via ICAP) in order to initialise the set of IP cores which constitute the legitimate system. Therefore, IP cores can be related to software applications that are loaded and activated under the control of the SeReCon. The SeReCon-enabled SR design (Figure 1) is the base FPGA configuration loaded after power up and contains only SeReCon. The base system is assumed to be secure, forming the RC RoT. SeReCon does not contain any proprietary (closed-source) IP cores and can therefore be freely audited. The external TAut confirms correct implementation of the SeReCon design. SysInt provides system credentials which TAut installs within SeReCon firmware. Finally, TAut encrypts the SR system bitstream in order to protect the sensitive part of the credentials between power-up cycles, i.e. the encryption keys and checksums used to protect IP core analysis reports produced by SeReCon (see Section 5). Credentials provide a unique identification of the RC system to the user and environment and are used to secure communication with SysInt and IPVend, e.g. in order to provide system upgrades. The SeReCon-based RoT is not likely to change during a product lifetime. However, if such a change is required, e.g. during a major system upgrade, trust has to be reestablished by the TAut and new credentials have to be generated.

In the above scenario, RoT security could be compromised through a successful attack on TAut. The feasibility of such an attack is based on the fact that TAut is aware of sensitive credential material. In order to mitigate this risk, this paper proposes an alternative method for controlled generation of unique, random and partially anonymous credentials (Figure 4). The method extends authenticated configuration proposed by Drimer (2007b) and uses true random number generator (TRNG) (Maiti et al., 2009). It is assumed that generation of two different base system designs (genuine and Trojan, i.e. a birthday attack), both having identical message authentication codes (MACs), is not feasible. The paper proposes a minor change to the FPGA fabric to provide a dedicated secure memory within the device. A fabric-embedded memory element, referred to as the ID Register (IDReg), is used to store the RC system *identity*, e.g. SeReCon credentials. The IDReg is hardwired to the MAC-generating module. The IDReg provides authenticated access to data register (DR) content. During an IDReg write access, input data is stored in the DR and the MAC is stored in the internal key register (KR). When IDReg data is requested, the current KEY input is compared with the KR content (Figure 4). The DR content is available only if values match, i.e. when the MAC of the current design is equal to the MAC used to store credentials. The IDReg should be non-volatile and include support for user-access and instant memory scrubbing ('one-shot zeroisation') upon tampering. Some FPGA devices already offer the battery-backed memory used for storing bitstream decryption key (Krueger, 2004). However, in current devices, user access is not supported.

Figure 4 Block diagram of the IDReg used for controlled generation of unique, random and partially anonymous security credentials



A novel algorithm is proposed for generating credentials in order to establish the secure RoT. TAut audits the process of credentials generation during the first execution of the SeReCon-based RoT. The algorithm steps are as follows:

S1 TAut audits the FPGA device, SeReCon source code and FPGA configuration bitstream and ensures generic environmental conditions (i.e. ambient temperature, FPGA voltage, etc) during the process

S2 SeReCon generates credentials (i.e. master symmetric-key, public-private key pair, etc.) using a TRNG (Simka et al., 2006; Maiti et al., 2009) and stores them in IDReg

S3 SeReCon reports the public-key to TAut

S4 TAut certifies and publishes the SeReCon public-key

S5 parties use certified credentials to authenticate the SeReCon during its lifetime

S5 SysInt and IPVend use AES encryption in order to communicate with SeReCon (one-time shared keys are generated using Diffie-Hellman key agreement protocol).

The algorithm has three important properties:

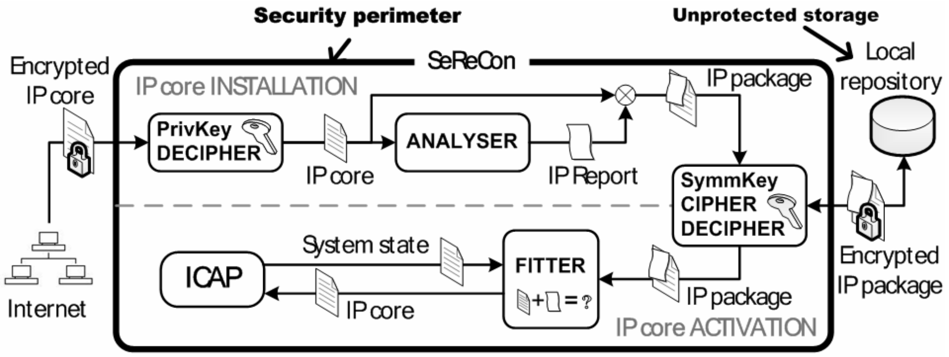
P1 Initial assumptions guarantee exclusive access to the sensitive part of the credentials (private crypto keys, etc.) only for the legitimate system i.e. SeReCon.

P2 The base configuration bitstream does not contain any credentials. Thus it can be audited in order to avoid vulnerabilities that might be introduced by third-party IP cores (Kepa et al., 2009)

P3 SeReCon RoT is immune to credentials leakage through a future successful attack on TAut.

A two-phase self-reconfiguration process is implemented in SeReCon in order to improve performance of IP core activation (Figure 5). During phase 1 the IP core is *installed* in the system, SeReCon performs analysis of its structure and generates a resource report which becomes an integral part of the installed IP core. This approach speeds up the subsequent reconfiguration process. In phase 2, when IP core *activation* is requested, SeReCon performs a controlled reconfiguration following verification of available resources and interfaces for the required IP core and the current system configuration. Only IP cores verified by SeReCon can be downloaded and configured in the RC system. The two-phase reconfiguration algorithm supports IP core spatial isolation (McLean and Moore, 2007) and allows dynamic instantiation of physical isolation primitives (Huffmire et al., 2007). The interface between SeReCon, IP cores and the rest of the system must be well defined so that activation of an IP core which eavesdrops on current IP cores can be prevented.

Figure 5 Two-phase SeReCon operation: IP core installation (top); IP core activation (bottom)



SeReCon analyses each IP core installed in the system and generates individual resource reports, which are attached to the IP core, encrypted and stored in an external repository. The analysis is performed once, before the IP core is activated. The report consists of IP core resource usage, location within the FPGA and IO interface. The report is extracted directly from the bitstream (Note and Rannaud, 2008; Krasteva et al., 2006; Kalte and Porrmann, 2006; Hübner et al., 2007). Also, integrity checksums for individual configuration frames are calculated and included in the report in order to prevent tampering. Subsequent IP module activation is based on successful completion of the analysis step.

SeReCon can detect the potential for damage to the FPGA fabric (routing short-circuit) and other parts of the system (e.g. misconfiguration of FPGA IO pins) before allowing PR with a new IP core via ICAP. If physical isolation between IP cores is required, a wrapping technique presented by Huffmire et al. (2007) can be applied by SeReCon, allowing runtime placement of isolated primitives (fences). In contrast to previous approaches, SeReCon not only verifies addressing of the configuration frame, but also performs analysis of its content.

When the functionality of the IP core is requested, e.g. during the system power-up or upon a reconfiguration request, SeReCon reads the IP core report and compares IP core resource requirements with the current system state, retrieved through ICAP (Figure 5). If no overlaps are detected and the communication interfaces match; the reconfiguration process is initiated.

In order to protect the integrity of the RC system, SeReCon must ensure the integrity of each installed IP core. In particular, it must guarantee that the analysis report has not been tampered with and that the corresponding IP core is unaltered. The amount of memory blocks [block RAM (BRAMs)] available in modern Xilinx FPGAs is in the order of megabits. It is not feasible to use this type of memory to store the complete configuration of an RC system as it might be comparable in size to the complete FPGA bitstream (tens of megabits). In order to address this issue, SeReCon implements secure (encrypted) storage (TCG, 2003) in the local repository, protecting integrity and confidentiality of data stored outside the RoT security perimeter, i.e. analysed IP cores and reports. When IP core analysis within SeReCon is complete, the core signature is generated and symmetric key encryption (AES-128) is used to protect the external IP core repository (Figure 5). This provides a protection mechanism which guarantees that the plain-text version of the installed IP core never leaves the SeReCon security perimeter.

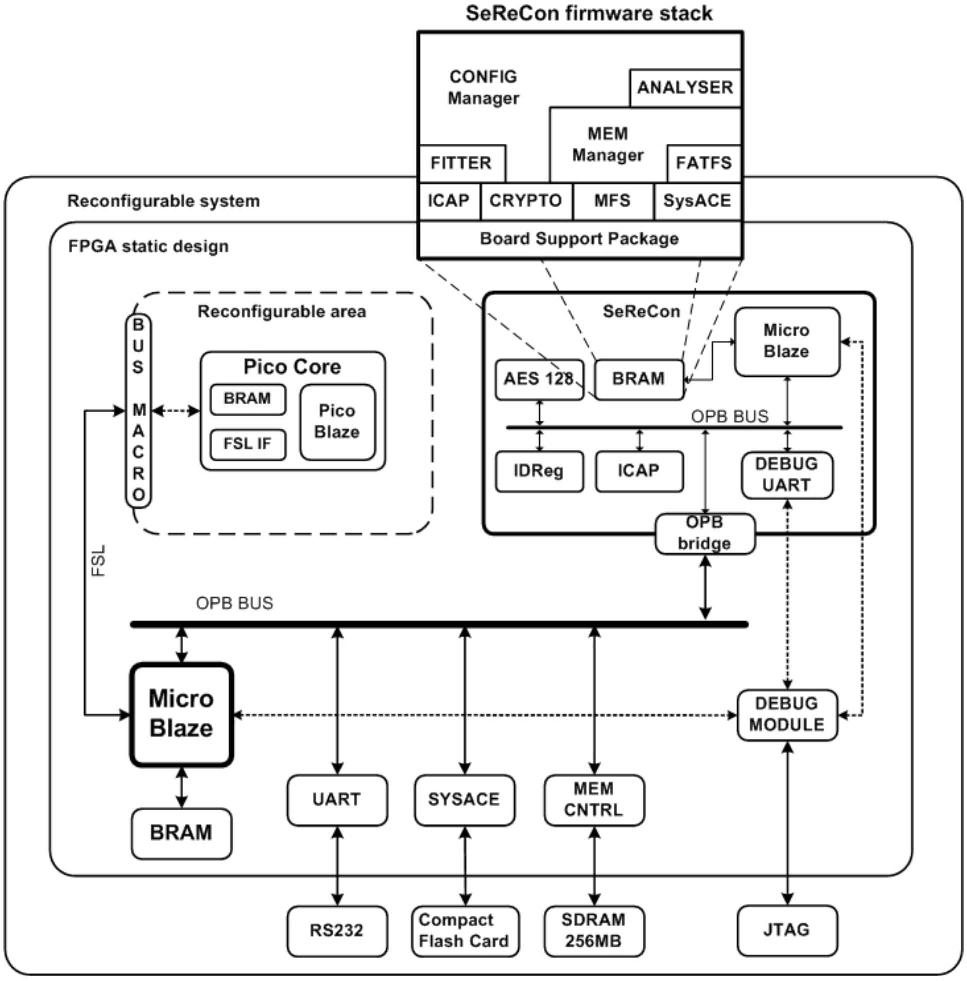
5 SeReCon implementation overview

This section reports on the SeReCon-enabled SR prototype implementation, highlights hardware-software partitioning issues and provides detailed insight into the operation of a prototype PR RC system using SeReCon. The SeReCon implementation provides a fixed-footprint public IP core, which manages runtime access to Xilinx FPGA configuration memory via the ICAP.

The prototype has been implemented on a Digilent XUP-V2P (XC2VP30) FPGA board. The SeReCon module and main system have been assembled using Xilinx Platform Studio Embedded Development Kit (v.9.1). The main system and SeReCon IP core are standalone MicroBlaze-based designs communicating through the OPB bus bridge. This allows SeReCon to reuse the resource-rich facility of the base RC system (256MB of SDRAM, CompactFlash etc.) without increasing its area footprint. The AES IP core is an open-hardware project available from OpenCores (Usselmann, 2002).

Figure 6 illustrates the block diagram of the SeReCon-enabled SR prototype system. The system contains two regions, namely the static area and the PR area. The static area includes the SeReCon module, along with most of the system IP cores and an additional JTAG interface for debugging purposes. A production version of the RC system should not contain any IP cores that are not part of the RoT.

Figure 6 Block diagram of SeReCon-enabled SR prototype system and SeReCon software stack

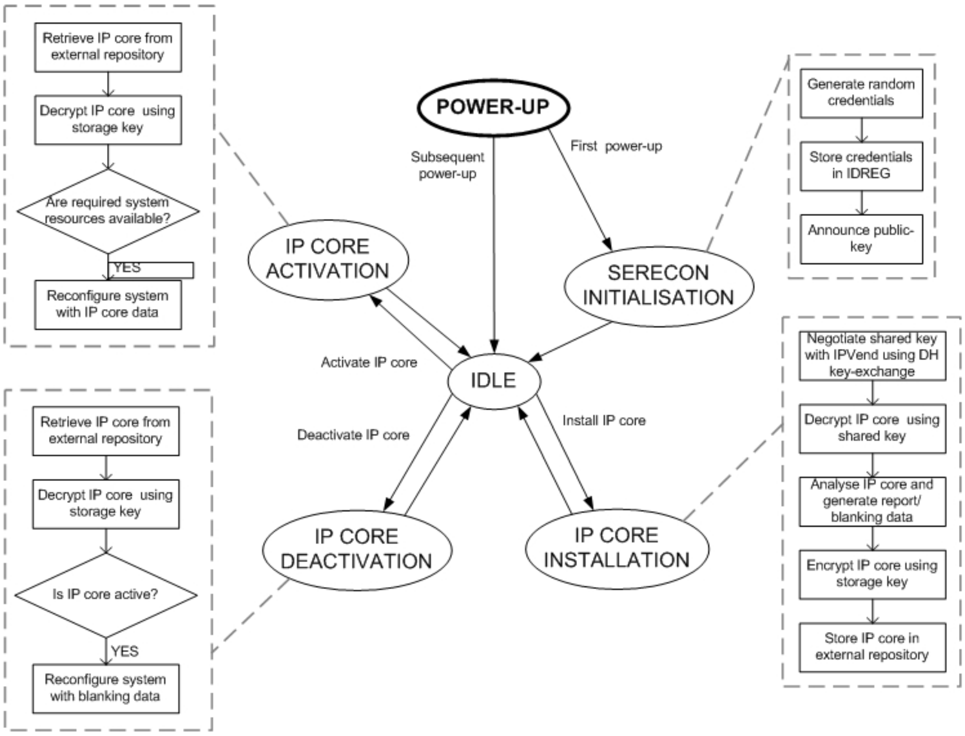


The PR area is used to test SeReCon functionality and contains the PicoCore IP. The PicoCore is a PicoBlaze-based accelerator module, used as a versatile accelerator for evaluation of the transparent runtime management of hardware tasks (Kosciuszkiewicz et al., 2007). Operation of the Picocore is defined by firmware stored in internal memory (BRAM). Firmware can be uploaded directly over a fast simplex link (FSL) or indirectly by updating the content of BRAM via ICAP.

The main functionality of SeReCon has been implemented in software running on a 32-bit CPU (Microblaze). Figure 6 also illustrates the SeReCon software stack. The configuration manager is the main SeReCon application used to serve reconfiguration requests from the main system. It uses a message passing interface API providing a message-based communication link between SeReCon IP core and the base RC system. The analyser and crypto APIs are in-house developed libraries. The analyser API supports IP core packet decomposition and analysis of configuration frames. The crypto API provides software for generating session keys (using elliptic curves and

Diffie-Hellman key agreement protocol) and wrapper for the AES IP core. Board support package (BSP) and standard libraries provided with EDK have been used to provide generic access to the ICAP, the memory file system (MFS) and the IP core repository located on the external compact flash card. The standard ICAP driver has been modified to support BRAM content encoding.

Figure 7 The SeReCon internal state diagram

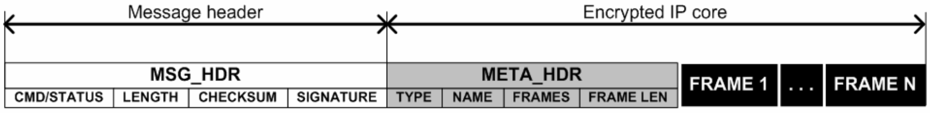

Retrieve IP core from external repository
Decrypt IP core using storage key
Are required system resources available?
YES
Reconfigure system with IP core data
POWER-UP
Subsequent power-up
First power-up
Generate random credentials
Store credentials in IDREG
Announce public-key
IP CORE ACTIVATION
SERECON INITIALISATION
Activate IP core
IDLE
Deactivate IP core
Install IP core
Negotiate shared key with IPVend using DH key-exchange
Decrypt IP core using shared key
Analyse IP core and generate report/ blanking data
Encrypt IP core using storage key
Store IP core in external repository
IP CORE DEACTIVATION
IP CORE INSTALLATION
Retrieve IP core from external repository
Decrypt IP core using storage key
Is IP core active?
YES
Reconfigure system with blanking data

This section describes the operation of the SeReCon prototype system. Figure 7 illustrates SeReCon internal state diagram.

- *SeReCon initialisation (RoT setup)*: as the IDReg implementation requires a modification to the FPGA fabric, the IDReg is currently emulated in software. Pre-generated credentials have been merged with SeReCon firmware and embedded into the initial bitstream as pre-initialised BRAM content. This scenario requires bitstream encryption, as a plain-text bitstream could lead to system compromise due to unauthorised bitstream analysis by an attacker. Virtex-II Pro devices do not support the PR port when bitstream encryption is used (Xilinx, 2005). Thus, newer architectures, e.g. Virtex-4/5/6, will be addressed in future work.

- *IP core installation*: a new IP core (constrained to predefined areas of the FPGA) is transferred to the device. Prior to installation, the IP core is downloaded to the system and stored in external memory (SDRAM). Custom protocol is used to communicate with SeReCon over the shared region of the external memory where the packet holding the IP core is located. After IP downloading, the system extends the IP core with the message header in order to allow consistency checking by SeReCon. Figure 8 illustrates the internal structure of SeReCon reconfiguration packet. The message header of each IP core contains control and status commands for SeReCon and the integrity checksum optionally signed by the IPVend. Each IP core includes a meta-header with information provided by the IPVend, i.e. IP core interface definition and the sequence of configuration frames.

Figure 8 Internal structure of reconfiguration packet for SeReCon; message header and encrypted IP core



During the installation phase, SeReCon analyses IP core configuration frames and creates an integrity checksum and resource report. Analysis performed includes identification of configuration frame address ranges (type, column, row) and routing usage. Also, the analyser generates complementary blanking data which is used to scrub the occupied area within the FPGA upon receipt of the request to deactivate the IP core. The resource report and generated checksum are merged with the analysed IP core and blanking data. The resulting package is encrypted using AES and SeReCon credentials and is stored in external non-volatile memory (compact flash card).

- *IP core activation*: when IP core activation is requested SeReCon downloads the related resource report from external memory, decrypts it and compares it with the current system state (obtained through the ICAP). If requested resources and interfaces are available, IP core configuration frames are decrypted one by one. The incremental checksum of each frame is calculated and verified against the resource report and the frame is sent to the ICAP port. Once this step is successfully completed, the IP core is ready for use. If any of the checks fail, the received configuration is discarded.

- *IP core deactivation*: different IP cores can occupy the same area over the lifetime of a system. The IP core deactivation process is similar to the activation process. Upon receipt of a deactivation request SeReCon verifies whether the IP core is currently active and selects the blanking data required to scrub the area to be deactivated.

Table 1 provides the statistics for the initial implementation of the SeReCon IP core.

Results indicate a significant impact on BRAM consumption (64%) due to the non-optimised software implementation. In order to place results in a broader context, Table 1 also includes SeReCon resource requirements as a percentage of the capacity of the largest available Xilinx FPGA device which supports partial-reconfiguration. SeReCon uses 2% of the XC5VLX330T user-logic resources along with 13% BRAM usage.

Table 1 FPGA resource usage for SeReCon implementation (synthesised using Xilinx ISE 9.1 with default settings)

SeReCon module	*LUTs*	*FFs*	*BRAMs*	*HW MULs (DSP)*
MicroBlaze	1862	1464	0	3 (0)
OPB bus	169	11	0	0
OPB BRAM IF	33	40	64	0
ICAP	182	151	1	0
OPB2OPB bridge	21	146	0	0
AES IP (enc + dec)	2005	815	22	0
Summary	*4272*	*4439*	*87x18 Kbit*	*3*
XCV2P30 resources	30816	30816	136x18 Kbit	136
% of device capacity	13.9	14.4	*64*	2.2
XC5VLX330T resources	*207360*	*207360*	*324x36 Kbit*	*192(192)*
Estimated % of dev. cap.	*2*	*2.1*	*13.4*	*1.6*

6 Conclusions and future work

This paper proposes a novel method for unique, random and partially anonymous system credentials generation and integrity protection using trusted self-reconfiguration and design verification. The paper describes a SeReCon architecture as a RoT for FPGA-based partially RC systems. The RoT is built upon the proposed SeReCon element and an extended FPGA fabric incorporating a bitstream authentication module and an ID Register (IDReg). The open architecture of SeReCon IP core can be audited by a TAut. Special care has to be taken by the TAut during the certification process of the SeReCon-enabled RC system, as its quality and the quality of secure key material generated by SeReCon is fundamental to correct system operation.

A novel method for installing and managing security credentials within SR systems has been proposed. SeReCon security credentials are generated internally, during the system certification process, rather than using pre-generated material. Since the TAut monitors the generation process and certifies only the resulting public-key, the disclosure of the private-key and other credentials are not required. Therefore, the private part of the credentials never leaves the security perimeter of SeReCon. This supports the public audit of the FPGA base configuration bitstream and thus the system RoT. The proposed algorithm for credentials generation offers increased security of the system RoT, based on a proposed minor FPGA fabric modification to provide the IDReg primitive and authenticated bitstream decryption.

SeReCon exploits paradigms of TC and PR and performs autonomous analysis of the structure of IP cores prior to reconfiguration. Runtime verification of physical placement of the IP core results in improved integrity protection during system self-reconfiguration and assures that even distrusted (uncertified) IP cores can be used so long as the core provides acceptable functionality and retains its spatial isolation. This could enable more convenient reuse of IP cores provided by external third-party IPVend.

SeReCon implements the policy of *integrity-maintaining* self-reconfiguration. This policy guarantees isolation of the RC system sub-modules and enforces inter-module communication only through module interfaces explicitly defined by the IPVend. This protects against system manipulation with potentially malicious overwriting configuration and/or eavesdropping of module communication links.

SeReCon employs authentication and encryption from the software domain in order to provide authenticated access to its credentials and secure storage of installed IP cores. Adding IP core signature checking to SeReCon may be an alternative way of assuring application locking, thus providing system security and IP rights protection.

A prototype SeReCon-enabled system has been implemented and resource usage has been presented. The SeReCon implementation provides a generic, fixed footprint, single point of entry, public IP core, which manages runtime access to Xilinx FPGA configuration memory via the ICAP. Implementation results indicate the need for a balanced hardware-software partitioning of the RoT design, as SeReCon memory (BRAM) requirements may be prohibitive for low and mid-sized FPGA devices.

Future work will consider time and resource-efficient algorithms for runtime bitstream analysis and security considerations for birthday attacks on material provided for TAut signing. Practical analysis of tradeoffs between the encryption methods used and self-reconfiguration time and efficient models of IP digital rights management will be investigated. Further work will also incorporate Xilinx Virtex-5/6 support for SeReCon-enabled PR using base bitstream encryption.

Acknowledgements

The authors wish to acknowledge Irish Research Council for Science, Engineering and Technology (IRCSET) for financial support of this work through the EMBARK initiative.

References

Agrawal, D., Baktir, S., Karakoyunlu, D., Rohatgi, P. and Sunar, B. (2007) 'Trojan detection using IC fingerprinting', *SP'07: Proceedings of the IEEE Symposium on Security and Privacy*, Berkeley/Oakland, CA, USA, May 20–23.

Blodget, B., James-Roxby, P., Keller, E., McMillan, S. and Sundararajan, P. (2003) 'A self-reconfiguring platform', *Proceedings of Field Programmable Logic and Application, 13th International Conference, FPL 2003i*, Lisbon, Portugal, September 1–3.

Bobda, C. (2007) *Introduction to Reconfigurable Computing Architectures, Algorithms and Applications*, Springer.

Bossuet, L., Gogniat, G. and Burleson, W. (2006) 'Dynamically configurable security for sram fpga bitstreams', *IJES*, Vol. 2, Nos. 1/2, pp.73–85.

Defense Science Board Task Force (DSBTF) (2005) 'High performance microchip supply', Report of United Stated Department of Defense, available at http://www.acq.osd.mil/dsb/reports/2005-02-HPMS_Report_Final.pdf (accessed on 1/12/2008).

Dijkstra, E. (1979) 'Structured programming', in E.N. Yourdon (Ed.): *Classics in Software Engineering*, pp.41–48, Yourdon Press, Upper Saddle River, NJ.

Drimer, S. (2007a) 'Volatile FPGA design security – a survey', available at: http://www.cl.cam.ac.uk/~sd410/papers/fpga_ security.pdf, accessed on 29/10/2007.

Drimer, S. (2007b) 'Authentication of FPGA bitstreams: why and how', *Proceedings of Workshop on Applied Reconfigurable Computing and Applications*, Mangaratiba, Brazil, March 27–29.

Drimer, S., Moore J. and Lesea, A. (2008) 'Circuit for and method of implementing a plurality of circuits on a programmable logic device', US Patent No. 7408381, Issued August 2008, Filed February 2006.

Glas, B., Klimm, A., Sander, O., Mueller-Glaser, K. and Becker, J. (2008) 'A system architecture for reconfigurable trusted platforms', *Proceedings of Design, Automation and Test in Europe DATE '08*, Nice, France, April 20–24.

Gogniat, G., Wolf, T. and Burleson, W. (2006) 'Reconfigurable security support for embedded systems', *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, Hyat Regency Kauai, January 4–7.

Hadzic, I., Udani, S. and Smith, J.M. (1999) 'FPGA viruses', *FPL '99: Proceedings of the 9th International Workshop on Field-Programmable Logic and Applications*, Glasgow, UK, August 30–September 1.

Hartenstein, R. (2001) 'Reconfigurable computing: a new business model-and its impact on soc design', *Proceedings Euromicro Symposium on Digital Systems, Design*, Warsaw, Poland, September 4–6.

Hübner, M., Braun, L., Becker, J., Claus, C. and Stechele, W. (2007) 'Physical configuration on-line visualization of Xilinx Virtex-II FPGAs', *VLSI, 2007. ISVLSI '07. IEEE Computer Society Annual Symposium*, Porto Alegre, Brazil, May 9–11.

Huffmire, T., Brotherton, B., Wang, G., Sherwood, T., Kastner, R., Levin, T., Nguyen, T. and Irvine, C. (2007) 'Moats and drawbridges: an isolation primitive for reconfigurable hardware based systems', *SP'07: Proceedings of the IEEE Symposium on Security and Privacy*, Berkeley/Oakland, CA, USA, May 20–23.

Kalte, H. and Porrmann, M. (2006) 'Replica2pro: task relocation by bitstream manipulation in Virtex-II/Pro FPGAs', *CF '06: Proceedings of the 3rd Conference on Computing Frontiers*, Ischia, Italy, May 3–5.

Kean, T. (2001) 'Secure configuration of field programmable gate arrays', *FPL '01: Proceedings of the 11th International Conference on Field-Programmable Logic and Applications*, Belfast, Northern Ireland, UK, August 27–29.

Kepa, K., Morgan, F. and Kosciuszkiewicz, K. (2009) 'Design assurance strategy and toolset for partially reconfigurable FPGA systems', *Special Issue of ACM Transactions on Reconfigurable Technology and Systems (TRETS)*.

King, S.T., Tucek, J., Cozzie, A., Weihang Jiang, C.G. and Zhou, Y. (2008) 'Designing and implementing malicious hardware', *Proceedings of First USENIX Workshop on Large-Scale Exploits and Emergent Threats*.

Kosciuszkiewicz, K., Morgan, F. and Kepa, K. (2007) 'Run-time management of reconfigurable hardware tasks using embedded Linux', *Proceedings International Conference on Field-Programmable Technology ICFPT 2007*, Kokurakita, Kitakyushu, Japan, December 12th–14th.

Krasteva, Y.E., de la Torre, E., Riesgo, T. and Joly, D. (2006) 'Virtex-II FPGA bitstream manipulation: application to reconfiguration control systems', *Proceedings of FPL '06. International Conference on Field Programmable Logic and Applications*, *2006*, Madrid, Spain, August 28–30.

Krueger, R. and Xilinx (2004) 'XAPP766: using high security features in Virtex-II series FPGAs', available at http://www.xilinx.com/support/documentation/application_notes/xapp766.pdf (accessed on 01/12/2008).

Maiti, A., Nagesh, R., Reddy, A. and Schaumont, P. (2009) 'Physical unclonable function and true random number generator: a compact and scalable implementation', *Proceedings of GLSVLSI '09: Proceedings of the 19th ACM Great Lakes symposium on VLSI*, Boston Area, MA, USA, May 10–12.

McLean, M. and More, J. (2007) 'Fpga-based single chip cryptographic solution', available at http://www.sdrforum.org/pages/sdr06/sdr06_papers/1.3/1.3-04.pdf (accessed on 01/12/2008).

Note, J.B. and Rannaud, E. (2008) 'From the bitstream to the netlist', *FPGA '08: Proceedings of the 16th International ACM/SIGDA Symposium on Field Programmable Gate Arrays*, Monterey, California, USA, February 18–20.

Ravi, S., Raghunathan, A., Kocher, P. and Hattangady, S. (2004) 'Security in embedded systems: Design challenges', *Trans. on Embedded Computing Sys.*, Vol. 3, No. 3, pp.461–491.

Simka, M., Drutarovsky, M., Fischer, V. and Fayolle, J. (2006) 'Model of a true random number generator aimed at cryptographic applications', *Proceedings IEEE International Symposium on Circuits and Systems ISCAS*, Island of Kos, Greece, May 21–24.

Skorobogatov, S. (2002) 'Low temperature data remanence in static ram', available at http://www.cl.cam.ac.uk/techreports/ UCAM-CL-TR-536.pdf (accessed on 01/12/2008).

Steiner, N. and Athanas, P. (2009) 'Hardware autonomy and space systems', *Proc. IEEE Aerospace Conference*, Big Sky, MO, USA, March 7–14.

Streichert, T., Koch, D., Haubelt, C. and Teich, J. (2006) 'Modeling and design of fault-tolerant and self-adaptive reconfigurable networked embedded systems', *EURASIP Journal on Embedded Systems*, Vol. 2006, pp.1–15.

Suh, G.E., O'Donnell, C.W. and Devadas, S. (2007) 'Aegis: a single-chip secure processor', *IEEE Des. Test*, Vol. 24, No. 6, pp.570–580.

Thompson, K. (1984) 'Reflections on trusting trust', *Commun. ACM*, Vol. 27, No. 8, pp.761–763.

Trimberger, S. (2007) 'Trusted design in FPGAs', *Proceedings of the 44th Annual Design Automation Conference*, San Diego, CA, USA, June 04–08.

Trusted Computing Group (TCG) (2003) Available at http://www.trustedcomputinggroup.org (accessed on 01/12/2008).

Tuan, T., Strader, T. and Trimberger, S. (2007) 'Analysis of data remanence in a 90nm FPGA', *Proceedings IEEE Custom Integrated Circuits Conference CICC '07*, San Jose, CA, USA, September 16–19.

Usselmann, R. (2002) 'AES (Rijndael) IP Core', available at http://www.opencores.org/project,aes_core.

Valette, N., Torres, L., Sassatelli, G. and Bancel, F. (2006) 'Securing embedded programmable gate arrays in secure circuits', *Proceedings of Parallel and Distributed Processing Symposium*, Rhodes Island, Greece, April 25–29.

Wollinger, T., Guajardo, J. and Paar, C. (2004) 'Security on FPGAs: state-of-the-art implementations and attacks', *Transactions on Embedded Computing Systems*, Vol. 3, No. 3, pp.534–574.

Xilinx (2005) 'Virtex-II pro user guide', available at http://www.xilinx.com/support/documentation/user_guides/ug012.pdf (accessed on 01/12/2008).

Xilinx (2007) 'Virtex-5 configuration user guide', available at: http://www.xilinx.com/support/documentation/user_guides/ug191.pdf (accessed on 01/12/2008).

Xilinx (2008) 'Single chip crypto', available at http://www.xilinx.com/esp/aero_def/crypto.htm (accessed on 01/12/2008).