2007 International Symposium on Field-Programmable Custom Computing Machines

Establishing Chain of Trust in Reconfigurable Hardware

Thomas Eisenbarth, Tim Güneysu, Christof Paar,
Ahmad-Reza Sadeghi, Marko Wolf
Horst Görtz Institute for IT Security, Bochum, Germany
{eisenbarth,gueneysu,cpaar,sadeghi,mwolf}@crypto.rub.de

Russell Tessier
University of Massachusetts Amherst, USA
tessier@ecs.umass.edu

Abstract

Facing ubiquitous threats like computer viruses, trojans and theft of intellectual property, Trusted Computing (TC) is an emerging technology towards building trustworthy computing platforms. A recent initiative by the Trusted Computing Group (TCG) specifies the use of Trusted Platform Modules (TPM), currently implemented as dedicated, cost-effective crypto-chips mounted on the main board of computer systems.

In this paper we propose implementations for TC functionalities based on more flexible and versatile approaches for reconfigurable and embedded architectures. Our approach allows for (i) a scalable design and update of TPM functionalities in embedded systems, (ii) the integration of the TPM hardware in the chain of trust to bind applications to the underlying TPM and the reconfigurable hardware, and (iii) the design of vendor independent TPMs.

Trusted Embedded Computing

Recent IT applications demand for computing devices that can provide sophisticated functional and security requirements. In this context embedded systems play an increasingly important role. Prominent examples are pay-TV, games, firmware updates and cellular phones. Progress in cryptography and information security has led to a variety of solutions, however, the reliability of these solutions strongly depends on the security of the respective computing device. In this context Trusted Computing (TC) seems to be a promising technology.

Currently, most realizations of TC are based on a Trusted Platform Module (TPM) implemented as a dedicated hardware chip[1] similar to a smart-card that is assumed to be securely bound to the computing device. A TPM provides basic security services for authenticated boot, data sealing and binding as well as system identification (attestation). However, the TCG does not consider hardware attacks, e.g., tapping attacks on the bus between TPM and microprocessor as well as on the TPM chip itself.

We address these challenges by presenting integrated solutions based on reconfigurable hardware architectures with TC functionalities, in particular including the TPM hardware itself in the so-called chain of trust.

Our fundamental idea is to place a TPM in the reconfigurable fabric of an FPGA (cf. Fig. 1). This approach enables a tight but flexible integration of the main system with the TPM and prevents easy attacks on the system interconnects. One of the main security issues, besides protection of the application logic, is to protect the integrity of the TPM against manipulation, replays and cloning. This issue is addressed by including the configuration bit stream of the FPGA in the chain of trust and binding it to the underlying FPGA hardware.

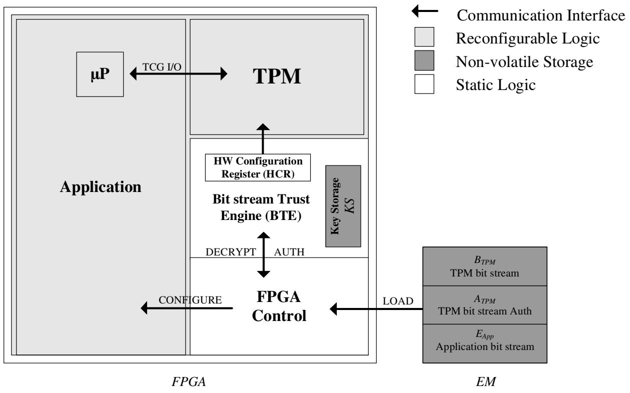

Figure 1. Architecture of a Trusted FPGA.

We assume the static control logic of an FPGA to allow partial configuration of the FPGA fabric, configuring the TPM and the application bit streams independently using the CONFIGURE interface. Furthermore, we introduce a *Bit Stream Trust Engine* (BTE) providing means to load, decrypt and authenticate bit streams using LOAD, DECRYPT

[1] TPM hardware implementations are already available, e.g., from Atmel, Broadcom, Infineon, Sinosun,STMicroelectronics, and Winbond.

0-7695-2940-2/07 $25.00 © 2007 IEEE
DOI 10.1109/FCCM.2007.23

and AUTH interfaces[2]. Furthermore, the BTE provides a protected and non-volatile key storage KS to store associated keys K_{Auth} and K_{Dec}. During the load and authentication of the bit stream, the BTE generates a unique credential for each bit stream which is stored in the Hardware Configuration Register (HCR). The TPM can access the HCR from the fabric for establishing and initializing the chain of trust. On power-up, the FPGA performs the following steps:

1. The FPGA controller loads the TPM bit stream B_{TPM} and the corresponding authentication information A_{TPM} from the external memory EM. After a successful load and authentication, BTE generates a credential value C_{TPM} for the TPM and writes it into the HCR.

2. The TPM requires access to its secret state $\mathcal{T} = (EK, SRK, TD)$ where EK, SRK, TD are security relevant keys and data to be stored permanently on the FPGA. Since $\mathcal{T}$ should be accessible exclusively to the authorized TPM we need a corresponding Access Control Logic (ACL) to avoid any unauthorized access. A possible implementation of the ACL is a non-volatile memory component on the same chip or package with an credential controlled access path, e.g., using the credential provided by the BTE derived uniquely from the bit stream. Alternatively, this ACL system can be combined with encryption to store only a secret key on-chip, thus saving memory, while the actual data is stored in separate off-chip memory.

3. The TPM accesses the read-only HCR and uses its contents as a (hashed) root value for the chain of trust. In this way the unique credential C_{TPM} of the bit stream authentication can be included in the trust chain 'measuring' the underlying hardware.

4. The application bit stream E_{App} is loaded, decrypted, authenticated and 'measured' in the same way. The credential of E_{App} is stored in the TPM, providing a 'measurement' of the application hardware.

One solution to protect the sensitive state $\mathcal{T}$ is the use of an ACL that identifies an authentic TPM using its credential C_{TPM} computed by the BTE. On the first load of the TPM, the ACL is reset to the TPM credential. Henceforth, the ACL can identify when an authorized TPM bit stream is loaded and only grants access to the state $\mathcal{T}$ to this authentic and known TPM instance.

The second solution for exclusive access to the TPM state $\mathcal{T}$ is to use Physically Unclonable Functions (PUF). *Silicon* or *Delay PUFs* (DPUF) can be realized on FPGAs and are able to generate a secret key K_{DPUF} by measuring gate delays [1]. The TPM issuer can include a DPUF in the reconfigurable TPM by choosing a circuit C at a specific location in the FPGA fabric to generate gate delays based on a fixed input m. Both, the location C as well as the input m, are kept in the encrypted bit stream and are only known to the TPM issuer. Hence, each TPM can generate its own device-specific and unique credential $K_{\mathsf{DPUF}} = \mathsf{DPUF}(C, m)$ to access its ACL controlled state $\mathcal{T}$.

Enhancing an FPGA with TC mechanisms in the reconfigurable logic provides an enhanced chain of trust, tight system integration, the capability of flexible updates of TPM functionality, and vendor independence. Flexible updates allow for elimination of security weaknesses in hardware without exchanging the entire system. Vendor independence is especially important for institutions which are not willing to trust external parties on their proprietary TPM implementations.

The transformation of FPGAs into trusted embedded computing platforms requires some modifications to the FPGA architectures which are not available today. We assume an SRAM-based FPGA that provides symmetric bit stream encryption, partial reconfiguration and a small amount of non-volatile (key) storage. Besides the ACL controlled non-volatile memory, we demand bit stream authentication that is not yet commonly available in FPGA devices. For authentication the BTE can use a Message Authentication Code (MAC), which ideally can use the same cryptographic mechanisms used for the bit stream encryption without the need for significant modification of the FPGA control logic [2]. For all other cryptographic components of the TPM which will be implemented in the fabric of the FPGA, a multitude of proposals for efficient implementation are available from literature.

References

[1] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Silicon physical random functions. *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 148–160, 2002.

[2] National Institute of Standards and Technology (NIST). Recommendation for block cipher modes of operation – the CMAC mode for authentication. NIST Special Publication SP 800-38B, 2005.

[2] Recent FPGA types from Altera and Xilinx already provide LOAD and DECRYPT interfaces. However, authentication is not provided.