Configurable computing for high-security/high-performance ambient systems *

Guy Gogniat[1], Wayne Burleson[2], and Lilian Bossuet[1]

[1]Laboratory of Electronic and REal Time Systems (LESTER),
University of South Britanny (UBS), Lorient, France
guy.gogniat@univ-ubs.fr; lilian.bossuet@univ-ubs.fr
[2]Department of Electrical and Computer Engineering,
University of Massachusetts, Amherst, MA 01003-9284 USA
burleson@ecs.umass.edu

Abstract. This paper stresses why configurable computing is a promising target to guarantee the hardware security of ambient systems. Many works have focused on configurable computing to demonstrate its efficiency but as far as we know none have addressed the security issue from system to circuit levels. This paper recalls main hardware attacks before focusing on issues to build secure systems on configurable computing. Two complementary views are presented to provide a guide for security and main issues to make them a reality are discussed. As the security at the system and architecture levels is enforced by agility significant aspects related to that point are presented and illustrated through the AES algorithm. The goal of this paper is to make designers aware of that configurable computing is not just hardware accelerators for security primitives as most studies have focused on but a real solution to provide high-security/high-performance for the whole system.

1 Introduction

Configurable computing research area has been deeply studied these last ten years. Today its maturity is largely admitted and many works have demonstrated its efficiency. As a consequence, configurable computing is now widely used in embedded systems to provide system performances and flexibility. At the same time pervasive computing is becoming a reality which enables interconnecting systems in a huge network [1]. As all entities can communicate to exchange data the critical question of security is unavoidable. Privacy and confidentiality is a major issue for users [2]. At the hardware level configurable computing offers numerous interesting features to efficiently handle this point. However, since now most studies have focused on the use of configurable computing to speed up security primitives dealing with architectural optimizations whereas it

* This work is supported by the French DGA DSP/SREA under contract no. ERE 04 60 00 010

is also mandatory to consider configurable computing at all levels from system to circuit. In this paper the problem of hardware security related to configurable computing is addressed. A quick review of hardware attacks is first presented in order to emphasize the main features a system must provide to be secure. Then configurable computing is analyzed to demonstrate its ability to address these features. Based on this information a guide for hardware security using configurable computing is proposed. Main issues to make this guide a reality are then discussed. Finally, as most of these concepts rely on the agility property provided by configurable computing a case study dealing with the AES security primitive is proposed to illustrate that point.

2 Hardware attacks and counter-measures

Hardware attacks Two types of attacks are considered depending on the way the attack is performed: active or passive attack. Active attack which corresponds to an alteration of the normal device operation can be further refined into three subtypes. Irreversible attack is a physical attack and corresponds to chip destruction or modification for reverse-engineering. After this type of attack the device does not perform its initial computation or not at all. Reversible attack consists in punctually moving the device out of its specified operation modes so as to move it into a weak state or to gain information from a computation fault [3]. Reversible attack can be or not detectable. When detected the device can react in order not to leak any information (e.g., by erasing a private key). Non detectable attacks can be for example glitch attacks on clock or power. It is very difficult to detect these types of attacks as there is always a compromise between reliability and efficiency. When too sensitive the detection is not reliable since it can detect some normal variations that do not correspond to any attacks and when not enough sensitive the detection is neither reliable since it cannot detect some attacks. Detectable attack is for example black box attack, fault injection and power or temperature reduction. For a whole system they are difficult to handle. However one has to keep in mind that given enough time, resources and motivation an attacker can break any system [4]. Passive attacks enable to deduce secrets from the analysis of the correlation between the legal information output by the device and the side-channel information (i.e., current, power, electromagnetism) [3]. In that case the device computes normally and the attack is more sophisticated since it relies only on the statistical evolution of the peripheral information. Examples of passive attacks are timing, power and electromagnetic emission analysis [5].

Counter-measures So what conclusions from these attacks must be drawn in order to increase system security at the hardware level? To be safe a system should:

– Not provide any information (i.e., data leaks) in order to disable passive attacks. The system must be symptom-free [3].

- Be continuously aware of its state and notably of its vulnerability in order to react if necessary. The system must be security-aware.
- Analyze its states and its environments in order to detect any irregular activity. The system must embed distributed sensors and monitors to be activity-aware.
- Be agile in order to react rapidly to an attack or to anticipate an attack. Be agile to be a ble to update security mechanisms as long as attacks evolve. The system must provide agility.
- Be tamper resistant in order to resist to physical attacks. The system must be robust [6][7].

And at the same time, the system must provide high performance to run the applications. Throughput, latency, area, power, energy are examples of parameters that are mandatory to run actual applications. So where is the solution, what technology provides these characteristics?

3 Configurable computing

Configurable computing presents several major advantages to deal with both security and performance compared to dedicated hardware components and processors. Some aspects are not specific to configurable computing but are more related to design at logic and circuit levels as for example symptom-free and robustness. But one major feature is required when dealing with security, adaptability (this term gathers the notions of awareness and agility) that is not provided by dedicated hardware components. Processors also provide adaptability through code update but they do not meet with the high performance requirements. Configurable computing provides many interesting features to be selected as a high-security/high-performance target. One key feature that underlies all others is the dynamic nature of configurable computing. Dynamic configuration enables to react and adapt rapidly in order to provide efficient architecture for performance and security. Irregular activities can be detected and the system can react objectively. This dynamism can be performed at run time or not, depending on the requirements.
High performance with configurable computing has been deeply studied these last ten years. In the special case of security these studies have focused and still focus on high performance for security primitives (typically cryptography) [8][9]. However this vision of security is only a part of the challenge to be addressed to provide secure system since it deals mainly with architecture optimizations. In the following a vision of security that gathers all the issues from system to circuit levels is proposed. It is essential to enlarge today vision of security since configurable computing may not be a single part of the system but the whole system.

4 Secure systems and configurable computing

When dealing with security it is important to determine what you want to protect and for how long you want to protect it. Furthermore defining the proper security boundary is critical for designing a flexible yet provably secure system [4]. In order to address these points it is essential to analyze what the different issues to provide secure configurable computing are. The very important idea is to classify and to define what the boundary of each part of a configurable system is and what design levels for security have to be targeted. Depending on the design and the security policy one or several security issues have to be considered. The designer has to be aware of these boundaries and design levels in order to guide his safe design building. In the following two complementary views are proposed in order to deal with security and configurable computing, the first one is related to system parts (system boundaries) and the second one to system security layers (security hierarchy).

4.1 Configurable Computing Security Space

The view of a system and of its different parts enables to highlight what the issues to build secure systems are. Three domains have to be addressed: Configurable Security Module, Secure Configurable System, and Configurable Design Security. Each domain focuses on a specific point and is detailed hereafter.

Configurable Security Module A Configurable Security Module is a part of the whole system and performs some security primitives (e.g., cryptography, data filtering). A system generally embeds several Configurable Security Modules. Many works have focused to define efficient Configurable Security Modules dealing with very interesting and optimized architectures [8][9], however when dealing with agility it is also essential to define what the rules to switch or to update a module are. Thus, a security module controller is needed in order to manage the agility (i.e., flexibility) provided within the Configurable Security Module. Typical security module controller control tasks are related to configuration context to change or to adapt the functionality of the module [10].

Secure Configurable System The Secure Configurable System domain deals with the security of the whole system to mainly perform intrusion prevention and detection. To build a Secure Configurable System three main points have to be considered: security awareness, activity awareness through distributed sensors and monitors, and agility. Security awareness is required in order to build a system that is aware of its state in order to anticipate and to detect possible attacks. Distributed sensors and monitors build the security network that enables the system to be aware of its activity [10]. Agility enables the system to react in order to modify its state to defeat an attack. Different levels of reaction are considered depending on the type of attack, reflex or global. Typical monitor control tasks are related to sensors and protocols analysis, monitor state exchange, reaction management, and monitor agility.

Configurable Design Security A configurable computing module/system is defined through the configuration data since each hardware execution context is defined through a specific configuration. The configuration data represents the design of the module/system; it may contain private information (i.e., intellectual property) that needs to be protected from adversaries to prevent reverse-engineering. The design security is provided through cryptography and needs a dedicated Configurable Security Module that performs the cryptography primitives (i.e., authentication, encryption). When the configuration data is protected the Configurable Security Module enables to configure the system without leaking any information about the design it embeds [11].

Depending on the security policy one or all domains have to be considered. Right now at the hardware level most studies have focused on Configurable Security Module and Configurable Design Security however it is essential to deal with the system level architecture to enable the visions of ubiquitous computing [10]. It is important to keep in mind that building a secure system has some overhead costs, so defining the right security boundary is important to meet with design constraints and to provide power efficient system [12]. Configurable computing enables to provide security/performance trade-off dynamically which promotes dynamic evolution of the system to manage dynamic security policy.

4.2 Configurable Computing Security Hierarchy

The previous view was dealing with the different parts of a system which is important for the designer in order not to disregard any parts of the security barriers. Another view is important which is related to the different hierarchical levels of a design from system to circuit levels. As each level provides specific weaknesses specific mechanisms need to be defined in order to build a global secure system (i.e., defense in depth). Depending on the requirements several levels have to be considered however as previously mentioned it is important to clearly define what the security boundaries for a system to be protected are. In the following main issues dealing with each level are presented.

Secure System Level At the system level configurable computing is seen as the global system (it corresponds to the Secure Configurable System in the previous view). At that level it is important to continuously monitor the activity of the system to detect irregular sequence of computation [10]. Another important feature is to keep the system as a moving target in order not to enable attackers to get a signature of the system or to identify some sensitive parts of the system [4]. This mobility should be provided for both system parts and monitors.

Secure Architecture Level At the architectural level the architecture of a module is considered (it corresponds to the Configurable Security Module in the previous view). Critical modules are typically cryptography primitives. The architecture of these modules has to be flexible, efficient and fault tolerant.

Another important feature for security is to provide symptom-free and security-aware algorithms and modules in order to disable side-channel attacks.

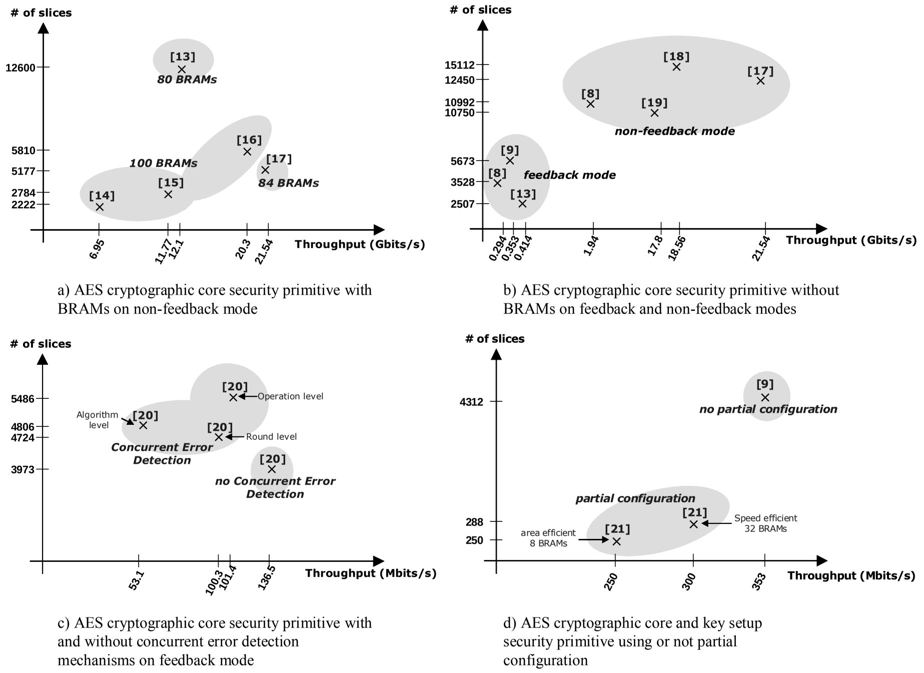
Secure Logic Level At the logic level the design of gates is targeted. The main point that has to be considered is to provide symptom-free gates (e.g., balance the computation time, synchronize the inputs, leave no evidence of previous computation) [3]. Gates need to be fault tolerant in order to be reliable. Reliability has to be a major concern since fault injection can break security barriers.

Secure Circuit Level At the circuit level the transistors and the physical process are considered. The goal is to strengthen the hardware physical shielding against for example RAM overwriting, optical induced fault, clock or power glitch attacks. An essential issue at that level is to define sensors that enable to prevent attacks by detecting them.

Configurable Computing Security Hierarchy is a complex structure and each level has its importance in order to provide a defense in depth. There are still many open problems to provide such a hierarchy from physical to system level concepts. The key idea is always to provide symptom-free and security-aware devices; the strength of configurable computing is its inherent agility using dynamic configuration. It enables to keep the system moving and to strengthen the security barriers when needed. Another essential issue when dealing with embedded systems is to provide the just right barrier and efficiency in order to keep alive the system functionality as long as possible (i.e., power-aware systems). Configurable computing provides this capability but the mechanisms to control this adaptability still need to be defined. As most of the concepts presented in the two previous views rely on agility, in the following a discussion dealing with that point is provided.

5 AES (Rijndael) security primitive agility case study

To illustrate the concepts related to agility an analysis of a Configurable Security Module is proposed. The case study deals with the AES security primitive. This case study is based on published works dealing with configurable architecture. All the selected implementations have been performed on Xilinx Virtex FPGA which is a fine grain configurable architecture. For that architecture the configuration memory relies on a 1D configuration array. More precisely it is a column based configuration array, hence partial configuration can be performed only column by column. For security issues, this type of configuration memory does not provide full flexibility but still enables partial dynamic configuration to perform security scenarios. Figure 1 gathers all the different implementations and represents them in four charts; each chart corresponds to some specific parameters. Figure 1.a corresponds to the AES cryptographic core security primitive with BRAMs (i.e.,



a) AES cryptographic core security primitive with BRAMs on non-feedback mode

b) AES cryptographic core security primitive without BRAMs on feedback and non-feedback modes

c) AES cryptographic core security primitive with and without concurrent error detection mechanisms on feedback mode

d) AES cryptographic core and key setup security primitive using or not partial configuration

Fig. 1. Agility design space for the AES security primitive: throughput/area/reliability trade-offs

embedded RAM) on non-feedback mode [13][14][15][16][17]. Thus for these studies key setup management is not considered. Concerning agility all the solutions are based on static and full configuration. The configuration is defined through predefined configuration data and performed using remote-configuration. The configuration time is on average tens of ms, since full configuration is performed. The security module controller is not addressed in these studies since the implementations are static. Figure 1.a highlights that various area/throughput trade-offs can be provided depending on the implementation. This is important to dynamically adapt the performance and to deal with security of the module. From the security side it enables the global system to behave as a moving target and from the performance side it allows to dynamically consider different throughputs depending on the actual application requirements. Figure 1.b corresponds to the AES cryptographic core security primitive without BRAMs on feedback [8][9][13] and non-feedback modes [8][17][18][19]. As previously key setup management is not considered. Solutions [8], [9] and [13] correspond to feedback mode while the others to non-feedback mode. Feedback solutions provide throughput on average hundreds of Mbits/s whereas non-feedback solutions are around tens of Gbits/s. Same remarks as previously can be done concerning agility characteristics; static, full and predefined configuration is considered. The goal in these studies is to promote high throughput while reducing area and dealing with a specific execution mode. However as explained in this paper dynamism and reliability have to be considered also.

Figure 1.c is interesting since it proposes different solutions that manage fault detection which guaranties reliability; essential feature for security. Fault detection can be performed at different levels of granularity from algorithm to operation level [20]. Performance/reliability trade-off is interesting since finer level of granularity enables reduced fault detection latency and then promotes fast reaction against an attack. But this efficiency is at the price of area overhead. No error detection leads to better performance, thus is it important to dynamically adapt the level of protection depending on the environment and the state of the system. Concerning agility static, full and predefined configuration is considered.

Finally figure 1.d provides some interesting values since solutions dealing with dynamic configuration are proposed. In [9] full configuration with predefined configuration data is implemented whereas in [21] partial configuration with dynamic configuration data is carried out. In both cases remote-configuration is performed since the Configurable Security Module is seen as an agile hardware accelerator. Both solutions also deal with key setup management, in [9] it is performed within the module so that the architecture is generic and in [21] it is performed by the remote-processor which enables to provide key-specific architecture. In [21] the remote-processor implements the security module controller that computes the new configuration when new keys have to be taken into account by the cryptography core. This type of execution enables a large flexibility since the configuration data can be defined at run-time. However, in that case the computation time to define the new configuration data is in the range of 63-153 ms, which can be prohibitive for some applications. The reconfiguration time for a new configuration data is not critical (around tens of μs) since only partial configuration is performed. As it can be seen on figure 1.d partial configuration enables to significantly save area compared to a generic implementation since in that case the architecture is specialized for each key. the security policy supported by the security module controllers are not explicitly presented in these papers. Figure 1 highlights that various solutions can be implemented for a same security primitive hence various area/throughput/reliability trade-offs can be considered. Agility enables to promote these trade-offs and then to adapt dynamically both performance and security to actual execution context. A last point which is important is related to power consumption. All previous studies did not deal with that point however for ambient system it is an essential feature. In [12], energy efficient solutions are proposed for the AES security primitive. In this case the important metric is Gbits/joule which is very relevant since ambient systems are mobile.

In conclusion of that part it is important for designers that have to build Configurable Security Module to be aware of all these trade-offs in order to promote agility and to meet with performance. Studies dealing thoroughly with configuration power consumption, secure communication links and security module controller policy are still required in order to propose secure Configurable Security Module and by extension secure systems. However agility provides many keys to build high-security/high-performance systems.

6 Conclusion

Configurable computing presents significant features to target high-security/high-performance ambient systems. However today these features are partially addressed and it is time to extend the vision of security using configurable computing since not only some parts but the whole system will be embedded within configurable systems. In this paper an analysis of major issues dealing with security at the hardware level using configurable computing is proposed. The goal of this paper is to stress that configurable computing is not just hardware accelerators for security primitives as most studies have focused on. Actually this point is part of the global system when dealing with configurable embedded computing. For that purpose two complementary views are proposed in order to guide the designer when facing with the difficult problem of system security and key aspects related to agility are presented and illustrated through the AES security primitive. Clearly there are still many issues to make security commonplace dealing with configurable computing and to define the overhead costs that imply security mechanisms at the hardware level but this paper aims to propose a first step toward a security design guide using configurable computing to meet with high-security/high-performance ambient system requirements.

References

1. C. Plessl, R. Enzler, H.W.J.B.M.P.L.T., Troster, G.: The case for reconfigurable hardware in wearable computing. In Springer-Verlag, ed.: Personal and Ubiquitous Computing. Volume 7. (2003) 299–308
2. Xenakis, C., Merakos, L.: Security in third generation mobile networks. In: Computer Communications. Volume 27. (2004) 638–650
3. Guilley, S., Pacalet, R.: Soc securiy: a war against side-channels. In of the Telecommunications, A., ed.: Systeme sur puce electronique pour les telecommunications. Volume 59. (2004)
4. Cravotta, N.: Prying eyes. In: EDN. (2002) http://www.edn.com/toc-archive/2002/20020926.html.
5. F-X. Standaert, L. Van Oldeneel tot Oldenzeel, D.S., Quisquater, J.J.: Power analysis of fpgas: How practical is the attack? In Heidelberg, S.V., ed.: international conference on Field-Programmable Logic and its Applications (FPL 2003). Volume LNCS 2778. (2003) 701–711
6. Anderson, R., Kuhn, M.: Tamper resistance - a cautionary note. In: Second USENIX Workshop on Electronic Commerce Proceedings, Oakland, California, USA (1996)
7. Wollinger, T., Paar, C.: Security aspects of fpgas in cryptographic applications. In Rosenstiel, W., Lysaght, P., eds.: New Algorithms, Architectures, and Applications for Reconfigurable Computing, Kluwer (2004)

8. A. J. Elbirt, W. Yip, B.C., Paar, C.: An fpga-based performance evaluation of the aes block cipher candidate algorithm finalists. In: IEEE Transactions on Very Large Scale Integration (VLSI) Systems. Volume 9. (2001) 545–557
9. Dandalis, A., Prasanna, V.: An adaptive cryptography engine for internet protocol security architectures. In: ACM Transactions on Design Automation of Electronic Systems (TODAES). Volume 9. (2004) 333–353
10. G. Gogniat, T.W., Burleson, W.: Configurable security architecture for networked embedded systems. Technical report, ECE Department, University of Massachusetts, Amherst, USA (2004)
11. L. Bossuet, G.G., Burleson, W.: Dynamically configurable security for sram fpga bitstreams. In: 11th Reconfigurable Architectures Workshop (RAW 2004), Santa F, New Mexico, USA (2004)
12. Schaumont, P., Verbauwhede, I.: Domain specific tools and methods for application in security processor design. In: Kluwer Journal for Design Automation of Embedded Systems. (2002) 365–383
13. Gaj, K., Chodowiec, P.: Fast implementation and fair comparison of the final candidates for advanced encryption standard using field programmable gate arrays. In Springer-Verlag, ed.: RSA Security Conf. - Cryptographer's Trac, San Francisco, CA, USA (2001) 84–99
14. McLoone, M., McCanny, J.: High performance single-chip fpga rijndael algorithm implementations. In: Third International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2001), Paris, France (2001) 65–76
15. F-X. Standaert, G. Rouvroy, J.J.Q., Legat, J.D.: A methodology to implement block ciphers in reconfigurable hardware and its application to fast and compact aes rijndael. In: ACM/SIGDA eleventh international symposium on Field programmable gate arrays (FPGA 2003), Monterey, California, USA (2003) 216–224
16. G. P. Saggese, A. Mazzeo, N.M., Strollo, A.G.M.: An fpga-based performance analysis of the unrolling, tiling, and pipelining of the aes algorithm. In Heidelberg, S.V., ed.: international conference on Field-Programmable Logic and its Applications (FPL 2003). Volume LNCS 2778. (2003) 292–302
17. Hodjat, A., Verbauwhede, I.: A 21.54 gbits/s fully pipelined aes processor on fpga. In: IEEE Symposium on Field -Programmable Custom Computing Machines (FCCM 2004). (2004)
18. F-X. Standaert, G. Rouvroy, J.J.Q.J.D.L.: Efficient implementation of rijndael encryption in reconfigurable hardware: Improvements and design tradeoffs. In Springer, ed.: Cryptographic Hardware and Embedded Systems (CHES 2003). Volume Lecture Notes in Computer Science 2779., Cologne, Germany (2003) 334–350
19. K.U. Jarvinen, M.T., Skytta, J.: A fully pipelined memoryless 17.8 gbps aes-128 encryptor. In: ACM/SIGDA eleventh international symposium on Field programmable gate arrays (FPGA 2003), Monterey, California, USA (2003) 207–215
20. R. Karri, K. Wu, P.M., Kim, Y.: Concurrent error detection schemes for fault-based side-channel cryptanalysis of symmetric block ciphers. In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. Volume 21. (2002)
21. McMillan, S., Cameron, C.: Jbits implementation of the advanced encryption standard (rijndael). In: international conference on Field-Programmable Logic and its Applications (FPL 2001), Belfast, Ireland (2001)