Autonomous Physical Secret Functions and Clone-Resistant Identification

Weal Adi

Technical University of Braunschweig
Braunschweig, Germany
wadi@ieee.org

Abstract

Self configuring VLSI technology architectures offer a new environment for creating novel security functions. Two such functions for physical security architectures are proposed to be generated autonomously as unknown/secret internal functions. A cell-based FPGA technology architecture is deployed for generating two classes of self-constructed one-way physical secret functions, one representing a hash function and the other a ciphering function. The Hash function is a non-invertible mapping, where the cipher function should be invertible. The two sample architectures of the functions are inspired from the programmable cell structure of the selected FPGA technology. As the functions are internally created, their mapping structures can be kept completely secret and even unknown to anybody. Such units could be efficiently deployed for a novel physical security even when nothing is known about their exact architecture and mapping functions. Several new attractive application scenarios are demonstrated including a type of zero-knowledge proof of identity and clone-resistant physical units as well as secured dependency functions. It is also shown that such security mechanisms can be kept operational for some useful applications even if the secret-unknown functions are allowed to evolve and develop additional time-dependent and individual properties. Such security functions became recently possible after self-configuring VLSI architectures are available as a part of real microelectronic systems.

Keywords-Identification; *secret unknown hardware functions; clone-resitant units; secret-unknown physical-cipher, secret unknown hash-functions.*

1. Introduction

Physical Unclonable Functions (PUF) has been introduced making use of some inherent/intrinsic physical differences between devices to uniquely identify electronic devices [1]-[3]. The devices identified by such technique are expected to be perfectly unclonable as the existing large mappings (PUFs) are not practically reproducible even by the same manufacturer. However, the costly sensing and/or the inherent liability of such functions in electronic devices to be sensitive to temperature and voltage drifts could make PUF's technique often inadequate for many practical applications.

The main objective of this research is to devise practical techniques for creating non-ambiguous constructive differences in electronic devices such that each device would define autonomously a part of its structure in a non-predictable manner. The result should become a hard-wired physical structure which can serve for physical security tasks. The structures are created in such a way that they are kept intentionally both secret and non-reproducible, therefore practically hard to clone. This type of functionality becomes first possible through VLSI technologies having self-reconfiguration properties. Many new application scenarios could result out of such new structure properties. As the reconfiguration changes are not limited to the initialization phase of such devices, a dynamic time dependent evolution can even be considered in both space and time scale. Like all modern practically secure systems, no perfect security (unconditional security) is expected. However, there is also no reason to exclude the possibility of attaining practically smart security mechanisms, which may reach incrementally quite high confidence level approaching that of the PUF's or the biological genetic structures.

The paper is showing first a possible simplified creation strategy for a secret Hash and ciphering functions followed by showing some possible security application scenarios.

2. Background of the Research

Some Spartan 3 Xilinx FPGA's were fabricated with a unique secret serial number called a device DNA stored in a tamper resistant area with no cryptographic mechanisms to re-identify the device [4]. The initial designation of that identity was DNA, and it is at least quite interesting even no real cryptographic security is attained.

To inspire biological systems, the human uniqueness-properties and identification methodologies can be considered. The first identity a human being becomes is a given name and later an identity card, both endorsed by a trusted authority. By time progress, new personal properties are accumulatively acquired by human beings. The born DNA identity represent an intrinsic unique identity, which is mostly used to identify individuals in real life when all other identifications do not meet the degree of trust required. On the other hand, acquired identification properties as knowledge, skills, language and other profile properties of a living individual would differentiate persons even if they were born as twins and even if they were successfully cloned with the same DNA. Including such self created properties in the electronic device identification strategy could offer more stable and resilient security architectures. For example cloning a device in use would require practically additional seeking and tracing all relevant device transactions (including possibly secret ones) with the environment. This tends to be rather impossible in most practical application. Even if this could have been possible at some time and an attacker was able to clone a device, the system would detect discrepancy after some time, as both cloned and non-cloned units would exhibit different evolved identity properties for the same claimed unique name or serial number. In that case, the system administration would stabilize its system security by prohibiting both units and running more generic and intensive non-conventional proofs to pick out the illegal units. This is actually the reason why our dynamically growing human community system does not collapse easily and a remedy is mostly possible and successful with more or less efforts.

3. Self-Reconfiguring VLSI Technology

Self-reconfigurable physical hardware architecture could be deployed as a basic technology for creating (mutating) hardwired secret functions. In [7] a micro-involution function as a basic building block of a cipher was proposed to be self-generated in FPGA architecture. The example was based on the Xilinx programmable FPGA cell structure as that shown in Fig. 2. The involution structure was inspired from the existing programmable CLB architecture. The first target of this work is to extend the concept to generating hash-functions, which are simpler than involutions. The reason is that hash functions do not need to be invertible. The free programmable mappings capacity of a simple FPGA cell appears to be quite suitable for such cryptographic functions.

A CLB cell of Xilinx technology as shown in Fig. 1 includes 4 mapping look-up tables LUTs. Each table have got n=4 inputs resulting with a large number of different possible mapping functions in each cell

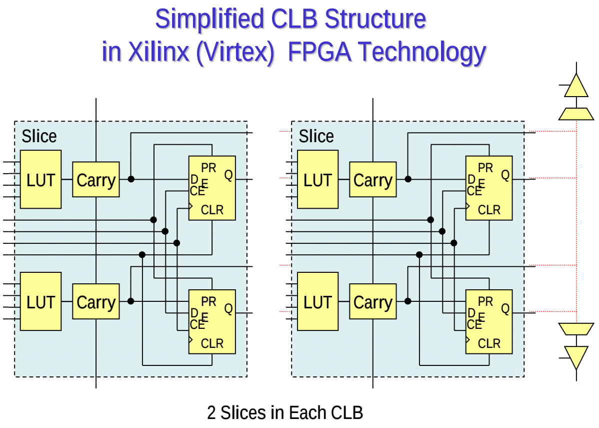


Figure 1. Basic Xilinx FPGA programmable cell

$$\text{Number of possible mappings in each cell} = 4 \cdot 2^{2^n} = 4 \cdot 2^{2^4} = 2^{18} \approx 256 \cdot 10^3$$

This LUT mapping can be seen as a micro-basic building block for ciphering and Hash functions. The average utilization of the programmable area in most applications leaves unused rest area in the chip. If the technology allows dynamic reconfiguration using on-chip reconfiguration controller, then unpredictable addition and deletion of free programmable micro-crypto structures can be self fabricated. If this process is kept autonomous, then the resulting functions can be kept unknown and secret to everybody if the devise is not invasively attacked. Programmable technologies offers also means like some hardware trap functions to prohibit reading the configured structures. Section 4 shows possible use of such secret (even unknown) Hash and cipher functions. Unknown ciphers/functions appear in the first look to be useless in the eyes of conventional security techniques.

3.1 Mutating Secret Hash Functions

Evolving physical hardware architecture would be the key technique to be deployed for constructing evolving accumulative properties. Fig 2 shows a basic architecture for non-predictable embodiment of evolving secret hash functions. An internal configuration controller should take care of the autonomous reconfiguration processes. The functional core areas FC’s are to be identified by a configuration controller and the useful free areas designated as Evolution Cores ECs are to be statically or dynamically monitored. The round spots represent single spare/unused cells with routable connectivity. A true random generator TRG is also required for constructing robust unknown-random crypto-functions.

Autonomous generation of useful and interconnection-safe hardware architecture is however not a trivial task. The function should obey the configuration design rules of the VLSI architecture, which are quit complex for a self-generating random process. Randomness is required to make the structure and behavior unpredictable and cryptographically resilient. At the same time, randomness could collide against the design rules. We will assume without denying possible obstacles that such difficulties are possible to overcome when seeking a real practical implementation.

The e-Mutation concept proposed in [5] was seen as a particular permanent change after chip fabrication. The application was restricted to changes in foreseen non-volatile memory without changing the hardware architecture.

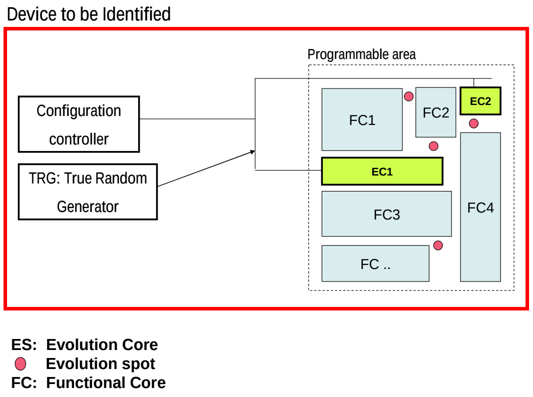


Figure 2. Embedding Evolutionary Structures in a FPGA Programmable Area

A standard know Hash function was assumed to reside on the chip ready connected to that secret memory. The resulting keyed hash function was then further refined and linked to protocols in a scenario to generate challenge-response markers designated as e-DNA markers inspired from the biological DNA concept [6]. In this work, more self-mutating functionalities are extended by deploying more sophisticated self reconfiguration technology.

3.2 Concepts for Secret Hash Architectures

To demonstrate a sample realization scenario, the CLB structure of Fig. 1 is used as a basic cell structure for the device under consideration. The basic requirement on a Hash function is to produce a computationally non invertible and non-predictable output for a given input. The Look-up table LUT unit represents a free programmable mapping which allows implementing highly non-linear mappings. Having a self-reconfiguring autonomous function, the LUT mapping could also be chosen as a true-random secret sequence loaded into the LUT. Fig. 3 shows a possible structural configuration scenario of the existing logical power in each CLB representing a micro non-linear state machine with 4-inputs and one output.

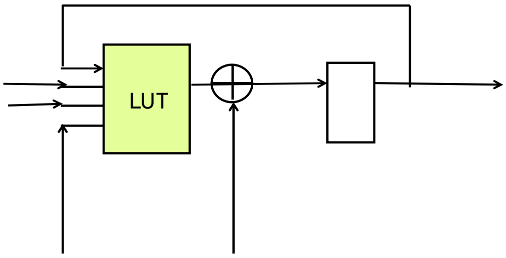


Figure 3. Micro Hash Function as a Micro Non-linear State Machine

Fig. 4 demonstrates a possible rectangular interconnection configuration of such basic cells in a spare reconfigurable chip area. Such secret Hash can be internally mapped into one free evolution core EC as shown in Fig. 2. This proposed architecture is just to demonstrate a basic strategy, which may need to be adapted to meet the realization constraints in such environments.

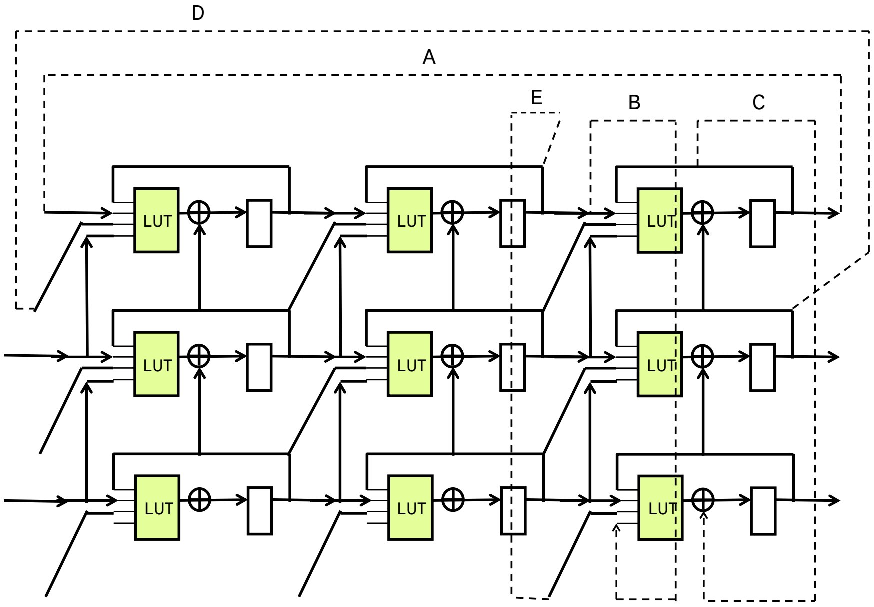


Figure 4. Array structure for a secret autonomous Hash-function

Another simple basic structure with non-linear non-predictable mapping often used for stream cipher constructions [9] is also proposed. The nonlinear feedback shift register (NFSR) architecture can be deployed as a basic structure having non-singular behavior over GF(2) is shown in Fig. 5. The non-singular property produces a single loop for the register's state sequence. This property is essential to avoid generating trivial output sequences. Notice that the function F in Fig. 5 can be any nonlinear function.

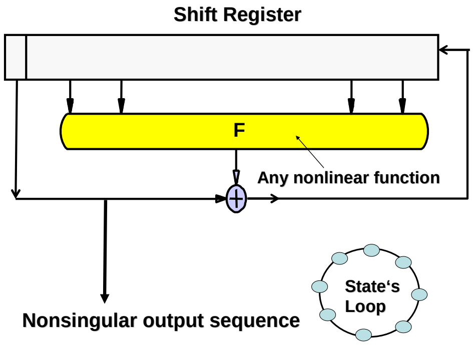


Figure 5. Non-Singular NLFS architecture

Fig. 6 shows a possible macro NLFSR unit mapped into one FPGA cell as a basic unit to be self-configured and autonomously embedded in a free area of a reconfigurable VLSI architecture. The number of input and output ports can be kept small (in best case one input and one output) to ease cascading such units within a chip in a non-predictable secret autonomous process.

Cascading many such cells possibly by interconnecting single-input single-output cells would result with a cryptographically relevant Has-function. As an example, cascading of 25 cells would come up with a 100-bit cascade of non-linear secret mapping. Again this is also conceptual cell architecture for cascading free evolution-spots as shown in Fig. 2.

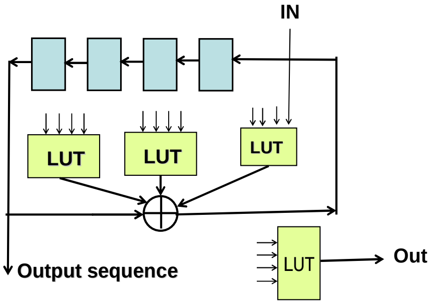


Figure 6. Possible Non-Singular NLFS Architecture in FPGA Cells

3.3 Dynamic function evolution:

Arranging and cascading such micro-machines as demonstrated in Figs. 3, 4 and 6 in an extendible fashion results with a dynamic hash architecture, which can be deployed as a Virtual-evolving structure. A simple possible initial scenario is to let each device start by having the same structure after fabrication and let the initialization of the device start by autonomously-defined random unknown selection of that structure as a reference permanent anchor configuration. Later extensions and reductions can be managed to produce evolved architectures for more sophisticated applications.

To simplify the procedure assume that the change can proceed by increasing or decreasing the array size in an unknown and unpredictable manner. The unpredictable change can be accomplished in its simplest way by configuring the LUT by unpredictable pattern from the true random generator TRG. The other evolution factor can be a true random change of the size of the array in both vertical and horizontal directions or by increasing or decreasing the chain of physically scattered E-spots shown in Fig. 2. Other evolution strategies can be adapted to fit to the individual properties of the deployed FPGA cell-technology and the intended security protocols.

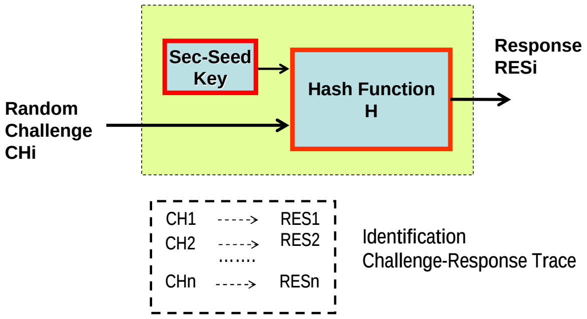
4. Security Application Scenarios

Two application scenarios for the resulting secret Hash-function and secret cipher are demonstrated.

4.1 Authentication using a Secret Hash functions

The "secret hash functions" are new enteritis essentially different from the conventional known Hash function designs. They can be however used in the same fashion to generate identification markers as challenge response pairs to uniquely identify physical devices. The procedure is sketched in Fig. 7. Notice that neither the knowledge of the secret seed key nor the knowledge of the Hash function is necessary to operate the challenge-response identification mechanism. This fact leads to the idea of accumulatively changing/evolving the hash function as a part of the device personality and keep tracing its evolution generations.

Notice that for the system functionality; neither secret seed key nor the Hash function H need to be known to anybody



If the challenge-response pairs table is kept secret and no challenge pattern is used more than once, then the system is practically secure!

Figure 7. Basic Challenge-Response Identification and Marker Principles

The same identification technique is used for PUF-based identification [1-3] with the difference that the secret seed key and the hash mapping are unchangeable as they are physically inherent unknown intrinsic mappings. Identification using PUFs is actually the ultimate solution if the inherent properties are consistently reproducible (i.e. device and PUF response are temperature, voltage and environment independent).

The proposed approach is generating a consistent unknown function similar to that of the PUF, however in a secret, self constructing and cryptologically secure manner. The simple evolution by changing just the secret seed key was recently proposed and discussed. Possible operation scenarios to manage such dynamic identity evolution in a practical environment were also shown in [5] and [6].

4.2 Secret cipher for clone-resistant units

The secret physical cipher is a new aspect for cryptographic application environment. A cipher was always assumed to be impossible to keep secret; therefore all security protocols are dealing with open cipher concept as a basic assumption. The fact of having the ability to generate a secret operational cipher delivers new application horizons. Two sample applications primitives are shown:

4.2.1 A Primitive for „Secured Dependency Scenario"

Fig. 8 shows „secured dependency" architecture. The application primitive is linking a process to be protected to a physical unit D0. The unit D0 includes a secret cipher. In the system initialization phase, all operational data for a certain process are stored encrypted by the secret cipher in a permanent open memory. After a certain short operating time, the process would not be possible to operate without having the physical unit D0. All process related data are stored after encryption by a secret cipher of D0 which no body knows. Even the ciphering key could be kept unknown or provided from outside D0 to add additional user controlled dependency to the system.

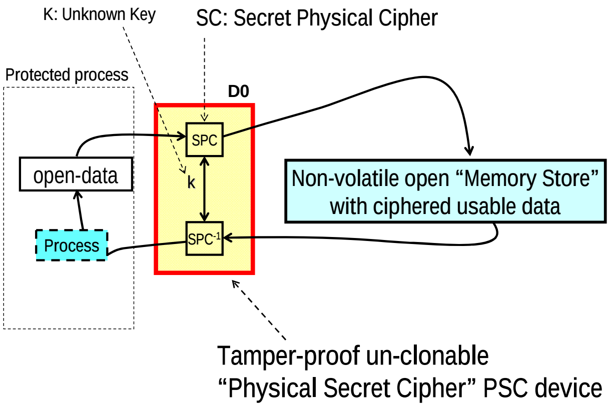

K: Unknown Key
SC: Secret Physical Cipher
Protected process
D0
SPC
open-data
k
Non-volatile open "Memory Store" with ciphered usable data
Process
SPC-1
Tamper-proof un-clonable "Physical Secret Cipher" PSC device

Figure 8. "Secured dependency" Primitive Using a Secret Cipher

A variety of other similar dependency scenarios can be constructed out of this primitive dependency scenario in a similar way.

4.2.2 A Security Primitive for "Physical Identity"

Fig. 9 shows a possible “Physical Identity Primitive” scenario. A trusted Authority TA can challenge a device D0 having a secret physical cipher SPC at some initial time point. A challenge response pair set Ci, Ri can be generated in a single set-up process after delivering a certain user dependent key K0. In later time point, only D0 would be able to deliver the correct response Ri for a certain selected Ci. In order to keep the Ri-Ci pairs used for just one time, an update and dynamic identity management protocol should refresh the challenge response process and avoid any challenge repetition in an open network. Updating process is out of the scope of this work.

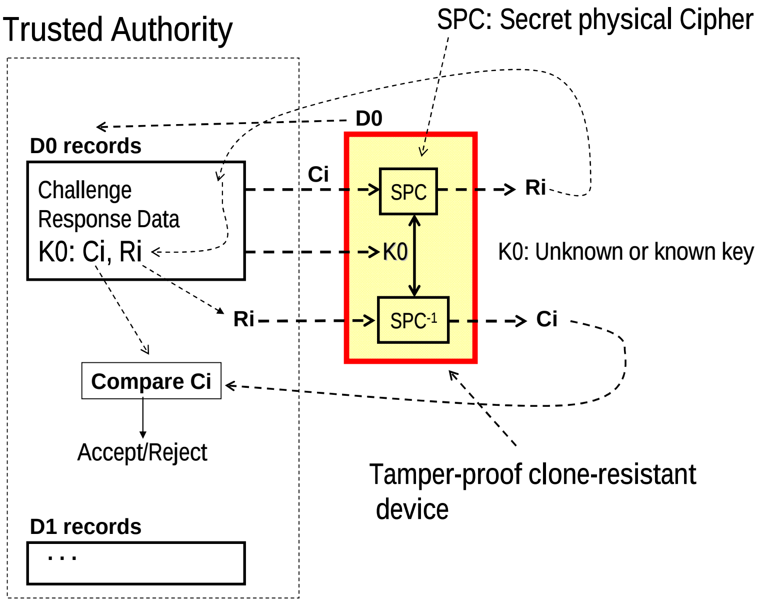


Figure 9. Clone-Resistant “Physical Identity Primitive” Using a Secret Cipher

5. Security Threats

Detailed security analysis is in preparation and is not yet available. The proposed architectures need also to be experimentally evaluated for each particular cell technology and evolution strategy. However, the following security-threats and concerns can be discussed:

1- The main hardware concern or challenge in that particular system resides however in the dilemma of designing and testing a hardware architecture, which should be allowed to run only one time to set up the initial reference configuration!

2- Due to the unpredictable random structure generation, the resulting functions could be a bad one and delivers information leakage or security relevant correlated behavior.

Countermeasures: The system controller could increase the number of cycles used to generate the response depending on the array size and its structure. However, the fact that neither, information about the LUT contents nor any information about the array’s size and its configuration exist, makes attempts to crack the system hopeless. The attacker would most probably be only encouraged to work on an attack if his work would come up with a general cracking methodology. This is quite un-expectable in such a varying environment and most probably leads to an early frustration as the structure is changing in a non-traceable manner. In addition to that, the probability of detecting a successful attack is inherently very high.

Security threats: Both PUF and the proposed constructive techniques require to trust the manufacturer in implementing the system according to the agreed specifications. Both systems can be exposed to invasive or replacement attacks as shown in Fig. 10. The invasive attack for PUFs is probably less efficient than the proposed technique, however the evolutionary properties of the proposed constructive technique seems to compensate that weakness in field operation. Therefore, if due to some temporary weakness resulting from the unpredictable random structure, a device can be easily cracked and cloned. As the structure is dynamic and in case of successful cloning at any time point, at least the cloned or the original device would fail to behave adequately after one or several transactions. In that case, both original and cloned device would then be detected and tagged. The danger of fraud is stopped until the case is cleared and the cloned device is identified by further investigations and made harmless. Therefore, the security of the whole system is still robust and stable.

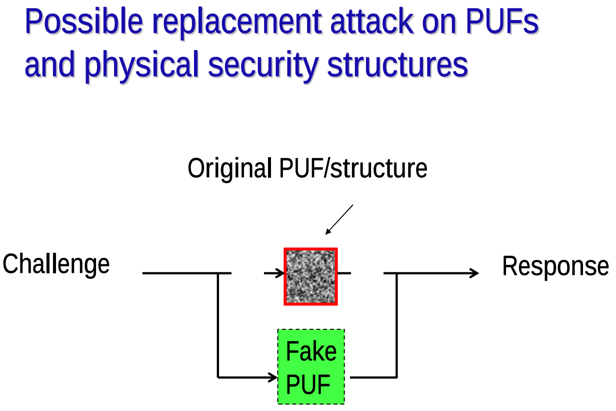

Possible replacement attack on PUFs
and physical security structures

Figure 10. Replacement Invasive Attack Scenario

A general design strategy for such physical security structures should target the following basic requirements: The proposed evolved structures should in its ultimate case spread over and penetrate into virtually all device activities throughout the lifetime of the device. In other words, the *evolution* should possibly ***diffuse*** in the whole physical area in an ***attacker-confusing*** non-predictable manner.

6. Summary and Conclusion

The paper is demonstrating new security structures concepts embedded in self-reconfigurable VLSI technology environment. The resulting secret ciphers and secret Hash functions exhibit new security application horizons due to the particular possibility of constructing autonomous practical secret unknown functions. Keeping functions secret was assumed as a non-realistic assumption in cryptographic systems. A simplified scenario for devising hardware physical secret functions was introduced for self reconfiguring architectures. Several new security application scenarios were shown. The initial results appear to be quite promising to establish new type or constructive and practical physical security infrastructure for a variety of new applications. Possible application areas are expected in vehicular and network security environment. Future efforts could include powerful architectures for the different reconfigurable environments and other new application scenarios. The self reconfiguration monitor in such systems is also a complex implementation part which is both technology dependent and security relevant.

References

[1] Gassend, B. Clarke, D. van Dijk, M. Devadas, S " Controlled physical random functions" Computer Security Applications Conference, 2002. Proceedings. 18th Annual, 2002,pp 149- 160

[2] G. Edward Suh*, Charles W. O'Donnell, Srinivas Devadas , " AEGIS: A single-chip secure processor". Information Security Technical Report (2005) 10,63e73

[3] P. Tuyls, RFID-Tags: Privacy and Security Issues, Philips Research

[4] Xilinx, FPGA Data book, Spartan 3 series, "Device DNA"

[5] Adi, Wael; Soudan, Bassel; "Bio-Inspired Electronic-Mutation with genetic properties for Secured Identification", Bio-inspired, Learning, and Intelligent Systems for Security, 2007. BLISS 2007. pp.133 – 136

[6] Wael Adi, "Clone-Resistant DNA-Like Secured Dynamic Identity," BLISS 2008, Bio-inspired, Learning and Intelligent Systems for Security, 2008, pp. 148-153

[7] Adi, Wael; N. Ouertani, A. Hanoun, B. Soudan, Deploying FPGA Self-Configurable Cell Structure for Micro Crypto-Functions. IEEE Symposium on Computers and Communications (ISCC'09), July 5 - 8, 2009

[8] F. Kargl, P. Papadimitratos, L. Buttyan, M. Müter, B. Wiedersheim, E. Schoch, T.-V. Thong, G. Calandriello, A. Held, A. Kung, and J.-P. Hubaux "Secure Vehicular Communications: Implementation, Performance, and Research Challenges" IEEE Communications Magazine, Vol. 46, No. 11, pp. 110-118

[9] Gammel, B.M.; Gottfert, R.; Kniffler, O.; An NLFSR-based stream cipher, IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006. Proceedings. Page(s):4 pp. – 2920