FACTA UNIVERSITATIS
Series: **Electronics and Energetics** Vol. 31, N° 2, June 2018, pp. 157 - 167
https://doi.org/10.2298/FUEE1802157D

ENERGY-EFFICIENT CRYPTOGRAPHIC PRIMITIVES[*]

Elena Dubrova

Royal Institute of Technology (KTH), Stockholm,
Sweden

Abstract. *Our society greatly depends on services and applications provided by mobile communication networks. As billions of people and devices become connected, it becomes increasingly important to guarantee security of interactions of all players. In this talk we address several aspects of this important, many-folded problem. First, we show how to design cryptographic primitives which can assure integrity and confidentiality of transmitted messages while satisfying resource constrains of low-end low-cost wireless devices such as sensors or RFID tags. Second, we describe counter measures which can enhance the resistance of hardware implementing cryptographic algorithms to hardware Trojans.*

Key words: *Security, lightweight cryptography, cryptographic primitive, encryption, message authentication, hardware Trojan.*

1 Introduction

Today minimal or no security is typically provided to low-end low-cost wireless devices such as sensors or RFID tags in the conventional belief that the information they gather is of little concern to attackers. However, case studies have shown that a compromised sensor can be used as a stepping stone to mount an attack on a wireless network. For example, in the attack described in [1], wireless tire pressure sensors were hacked and used to access the automotive system.

As the number and type of connected devices grows, so are the security risks. Attacks are becoming more frequent and their effect is more global. The October 21st, 2016, massive distributed denial of service attack that made inaccessible millions of webpages has shown how vulnerable the Internet is today. Future wireless networks are expected to support security-critical services related to industrial automation, traffic safety, smart trans-

Received October 21, 2017; received in revised form January 29, 2018

Corresponding author: Elena Dubrova
Royal Institute of Technology (KTH), Stockholm, Sweden
(E-mail: dubrova@kth.se)

[*] An earlier version of this paper was presented as an invited address at the Reed-Muller 2017 Workshop, Novi Sad, Serbia, May 24-25, 2017

port, smart grid, e-health, etc. The value of the information to which the low-end devices will have access via future wireless networks is expected to be much greater than the one today, hence the incentives for attackers will increase [2]. The damage caused by an individual actor may not be limited to a business or reputation, but could have a severe impact on public safety, national economy, and national security.

Many low-cost wireless devices, such as sensors or Radio Frequency IDentification (RFID) tags, work under severe resource constrains such as limited battery and computing power, little memory, and insufficient bandwidth. These devices must dedicate most of their available resources to executing core application functionality and have little resources left for implementing security [3]. To satisfy their constrains, it might be advantageous to combine techniques intended to assure high reliability of communication links (scrambling, checksums, forward error correction (FEC)) with cryptographic techniques intended to assure security. In Section 2 we show how functional similarities between error detection and data integrity protection can be exploited to efficiently combine these two functions in one.

In Section 3, we address another important problem - assuring data confidentiality. For communications over insecure networks, such as the Internet, data confidentiality is assured by encryption. In recent years, many cases of successful attacks on networks causing disclosure of private data have been reported. This increased user privacy concerns. We describe an efficient encryption algorithm which satisfies resource constrains of low-cost wireless devices and therefore enables encrypting all traffic and data transmitted using these devices.

In addition to assuring data integrity and confidentiality, it is equally important to trust hardware which implements cryptographic algorithms. If cryptographic hardware has a vulnerability, all the efforts in defending a system at network or software levels are wasted. For example, malicious alterations inserted into an integrated circuit at the design or manufacturing stage can open backdoors into a system in spite of cryptographic protection. In Section 4, we describe two countermeasures against a type of hardware Trojans which exploit non-zero aliasing probability of Built-In-Self-Test (BIST).

2 Message Authentication

To authenticate a message means to verify that the message

1. comes from the right sender (its *authenticity*), and
2. has not been modified (its *integrity*).

Clearly, data integrity protection can be implemented by using some n-bit message authentication code, e.g. keyed hash message authentication code (HMAC) [4] or cipher block chaining message authentication code (CBC-MAC) [5], *on the top* of an error-detecting code, e.g. n-bit cyclic redundancy check (CRC). However, such an approach expands the message by n bits and requires a separate encoding/decoding engine which is more complex than the CRC encoding/decoding engine.

On the other hand, if we simply *replace* an n-bit CRC with an n-bit HMAC or CBC-MAC, we cannot guarantee the detection of the same type of random errors as the CRC. For example, the detection of n-bit burst errors cannot be guaranteed. This may have a negative impact on the reliability of communication links. Only if we make the conventional CRC *cryptographically secure*, can we assure a certain level of security without sacrificing reliability.

The latter motivated the development of cryptographically secure CRCs. The core idea is to make the CRC generator polynomial variable and secret. The CRC presented by Krawczyk [6] is based on irreducible generator polynomials. The approach described in [7] uses a product of irreducible polynomials. The CRC proposed in [8] uses generator polynomials of type $(1+x)p(x)$, where $p(x)$ is a primitive polynomial. In all three cases, testing for irreducibility or primitivity is required, which is either time or memory consuming. Selecting an irreducible degree-n polynomial at random requires either selecting at random a degree-n polynomial ($O(n)$ time) and running a test for irreducibility ($\Omega(n^3)$ time [9]), or selecting at random a degree-n polynomial from a database of irreducible degree-n polynomials (roughly $2^n/n$ space). Note that the irreducibility test has to be done during key agreement, i.e. it incurs delay before the communication can start. Therefore, it is desirable to minimize the time spent on doing it as much as possible.

In [10], we presented a cryptographically secure CRC based on any randomly selected generator polynomial, with no requirements on irreducibility. This eliminates the need for irreducibility tests. It takes only $O(n)$ time to generate a random polynomial of degree n. The presented cryptographically secure CRC retains most of the implementation simplicity of the traditional CRC except that the LFSR implementing the encoding and decoding is required to have re-programmable connections. Similarly to previously proposed cryptographically secure CRCs, the new one enables combining the detection of random and malicious errors without increasing bandwidth.

However, using random polynomials as generator polynomials for the CRC gives an adversary a higher chance of braking authentication. In [10], we provided a detailed quantitative analysis of the achieved security as a function of message and CRC lengths and showed that the presented authentication scheme is particularly suitable for short messages. Since short messages (a few bytes to a few tens of bytes) are expected to be dominant in Machine-to-Machine (M2M) communications [11], this message authentication technique might be quite useful for resource-constrained M2M devices.

3 Encryption

Encryption is the process of transforming a message in such a way that only authorized parties can understand its content. Encryption assures message *confidentiality*.

Encryption can be performed using either a *block* or a *stream* cipher. Block ciphers have been studied for over 50 years [12]. Collected knowledge about their design and cryptanalysis made it possible to develop the Advanced Encryption Standard (AES) algorithm which is widely accepted and has strong resistance against various kind of attacks [13].

On the other hand, an active public investigation of stream ciphers began only about 20 years ago [14]. A common type of stream cipher is the *binary additive stream cipher*, in which the keystream, the plaintext, and the ciphertext are binary sequences. The keystream is produced by a *keystream generator* which takes a secret key and an initial value (IV) as a seed and generates a pseudo-random sequence of 0s and 1s. The ciphertext is then obtained by the bit-wise addition of the keystream and the plaintext.

To design a secure stream cipher which satisfies technical requirements of resource-constrained devices, a best trade-off between area and performance for a given security level should be sought. Previous stream cipher designs have either too high propagation delay (e.g. Grain family [15]) or use too many flip-flops (e.g. Trivium [16]) for a given security level. Thus, they optimize only one of the two important parameters - area or performance. In [17], we presented a methodology for designing a class of stream ciphers, called Espresso, which takes into account both parameters simultaneously, thus minimizing the hardware footprint and maximizing the throughput of the design.

First, by using Non-Linear Feedback Shift Registers (NLFSRs) implemented in the Galois configuration [18, 19], feedback functions can be made smaller. This allows us to reduce the propagation delay compared to Grain while at the same time decrease the size compared to Trivium. Due to the large number of feedback functions in Espresso, its maximum degree of parallelization cannot be made as high as in both Grain and Trivium. Still, by carefully choosing feedback functions, we are able to guarantee the maximum degree of parallelization four and a maximum-length NLFSR.

Second, to enable security analysis of Espresso, we transform the original Galois NLFSR to an NLFSR whose configuration resembles the Fibonacci configuration. The core idea of our method is to assure that all of the most biased linear approximations of the output Boolean function take inputs only from those stages of the Galois NLFSR which have a corresponding equivalent stage in the transformed NLFSR. As a result, traditional cryptanalysis techniques can be applied to our design as well.

According to our evaluation, Espresso is the fastest among the ciphers below 1500 GE, including Grain-128 and Trivium. Its 1-bit per cycle version has 1497 GE area, 2.22 Gbits/sec throughput and 232 ns latency, meeting requirements of most applications envisioned today. It is resistant to known attacks, including linear approximations, algebraic attacks, time-memory-data trade off attacks, chosen IV attacks, differential attacks and weak key attacks.

4 Hardware Trojans

A *hardware Trojan* is a malicious modification of a design that makes it possible to bypass or disable the security of a system [20]. Hardware Trojans has been known for a while, but previously it was very difficult to inject a Trojan into the supply chain. In today's globalized world in which multiple players are involved in the supply chain, this is no longer a problem.

Malicious changes can be introduced into a design, for example, by tampering with a CAD-tool which is used for circuit's synthesis [21]. The code of a CAD-tool is usually huge and it undergoes a continuous development.

So, several extra lines which modify the original design to inject a Trojan may easily get unnoticed in a multi-million line code. Alternatively, a third-party-made IP-block might contain a backdoor that can be used to steal secret keys or extract internal chip data. Circuit modifications can also be made at the manufacturing stage, potentially affecting all chips or just some selected ones. Today's chips contain billions of transistors, so it is very difficult to identify which of them are not a part of the original design [22]. Functional verification is further complicated by the fact that manufacturers are typically given a freedom to add redundant circuitry to a chip in order to increase manufacturing yield [23].

The presence of hardware Trojans can be difficult to prove. For example, some PCs are claimed to contain malicious circuit modifications that allow a person who knows the modifications to remotely access a PC without the user's knowledge [24]. However, it is still not confirmed if these claims are true or not. Some processors are suspected to contain backdoors deliberately implanted in their hardware Random Number Generators (RNGs) that make possible predicting RNG's output [25]. Again, it remains a conspiracy-theory story.

We do not know if these stores are true or not. However, we cannot discount a possibility that such attacks may take place if they are feasible to implement. For example, as demonstrated in [26], it is possible to reduce the security of a hardware RNG compliant with FIPS 140-2 [27] and NIST SP800-90 [28] standards from 128 to 32 bits by injecting stuck-at faults at the outputs of selected transistors. This can be done without disabling the BIST logic which checks RNG's functionality at each power-up, without failing BIST tests, and without failing any randomness tests. Stuck-at faults can be injected a very stealthy way by modifying dopant types in the active region of transistors. Such dopant-level Trojans do not require adding any extra logic to the original design and therefore do not change its layout. As a result, the Trojan-injected circuit appears legitimate at all wiring layers. Even with the advanced imaging methods such as Scanning Electron Microscopy (SEM) or Focused Ion Beam (FIB), it extremely difficult to detect changes made to the dopant in a large design implemented with nanoscale technologies. To detect changes in dopant types, in addition to all metal layers, the contact layer has to be examined. High-quality imaging of the contact layer is significantly more costly than imaging of a metal layer [29]. In addition, since only the dopants of a few transistors are modified, the change in the side-channel information is too small to be detected by side-channel analysis. Typically side-channel analysis can only detect sufficiently large Trojans that are at most three to four orders of magnitude smaller than the original design [30]. Trojans of a smaller size remain undetected.

The attack presented in [26] exploits the fact that aliasing probability of BIST is non-zero due to the compaction of circuit's output responses. *Aliasing probability* is the probability that a fault-free circuit is not distinguished from a faulty one. If an n-bit compactor is used, the aliasing probability of BIST is $1/2^n$ [31]. In the traditional BIST, the same set of test patterns is applied to a circuit under test at each test cycle, and therefore the same compacted output response, called *signature*, is expected. Therefore, an adversary who knows the set of BIST test patterns can select suitable values for the Trojan that result in the same signature as a fault-free circuit signature. Since the aliasing probability is $1/2^n$, in order to inject a Trojan which does not trigger BIST, an adversary has to make 2^{n-1} simulation trials on average. The typical size of a BIST output response compactor is 32 bits, so the attack is feasible in practice.

In [32,33], we presented two methods for modifying BIST to prevent such an attack. In the first method, we make the BIST test patterns dependent on a configurable key which is programed into a chip after the manufacturing stage. In the second method, we use a remote test management system which has sufficient computational resources to execute BIST using a different set of test patterns at each test cycle. In both cases, the manufacturer does not have BIST test patterns and thus does not know which circuit modifications produce the same signature as a fault-free circuit. Depending on the application requirements, the former approach might be preferable to the latter, or vice versa. For example, the latter countermeasure might be preferable for the Internet of Things applications with constrained resources, e.g. M2M communications, since it removes some of the BIST functionality from a device under test rather than adds an extra key to it.

5 Conclusion and Future Work

We described message authentication and encryption algorithms which can assure data integrity and confidentiality while satisfying technical requirements of resource-constrained devices. We also showed how to design countermeasures which can enhance the resistance of hardware implementing cryptographic algorithms to a type of hardware Trojans which exploit nonzero aliasing probability of BIST.

The presented message authentication approach seems a good candidate for simpler 5G radio types, such as the ones used for direct communication in sensor networks, and use cases with constrained resources such as M2M. Future work includes evaluating the impact on bandwidth. In the current wireless standard message formats two separate fields are typically used for the protection against random and malicious errors. These fields may be located on different layers, e.g. in LTE the CRC is located at the physical (PHY) layer while the message authentication code is located at the packet data convergence protocol (PDCP) layer. A good strategy might be to combine these two fields into the one at the PHY layer and use the a cryptographic CRC for the protection against both types of errors. However, implications for security and coverage caused by such a merge need to be investigated.

The presented stream cipher Espresso has 128-bit security. We are currently extending it to 256-bit security, in order to meet the requirements of post-quantum cryptography. We also consider integrating encryption and authentication within Espresso.

Regarding hardware Trojans, there is no "silver bullet" method that can protect against *all* possible types of Trojans or other attacks. In parallel with new countermeasures, more complex attacks are being developed. Moreover, we are dealing with a two-ended stick - a method originally designed as a countermeasure can be later turned into an attack, and vice versa. For example, advanced visual inspection methods for Trojan detection can be used by IP thefts to reverse-engineer chips. Similarly, if side-channel analysis techniques for detecting Trojans that affect only a tiny fraction of a design are invented, they are likely to give rise to more effective side-channel attacks. Future work remains investigating if there are attacks which can go around the presented countermeasures and what can be done to avoid them.

Acknowledgements

The author was supported by the research grant No SM14-0016 from the Swedish Foundation for Strategic Research.

References

[1] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study," in *Proceedings of the 19th USENIX Conference on Security*, Berkeley, CA, USA, 2010, pp. 21–21. [Online]. Available: http://dl.acm.org/citation.cfm?id=1929820.1929848

[2] Ericsson, “5G security,” 2015, www.ericsson.com/res/docs/whitepapers/ 5G-security.pdf.

[3] A. Juels, “RFID security and privacy: a research survey,” *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 2, pp. 381–394, Feb. 2006.

[4] M. Bellare, R. Canetti, and H. Krawczyk, “Keying hash functions for message authentication,” in *Advances in Cryptology - CRYPTO 96*, ser. Lecture Notes in Computer Science, N. Koblitz, Ed. Springer Berlin Heidelberg, 1996, vol. 1109, pp. 1–15. [Online]. Available: http://dx.doi.org/10.1007/3-540-68697-5_1

[5] M. Bellare, J. Kilian, and P. Rogaway, “The security of cipher block chaining,” in *Advances in Cryptology CRYPTO 94*, ser. Lecture Notes in Computer Science, Y. Desmedt, Ed. Springer Berlin Heidelberg, 1994, vol. 839, pp. 341–358. [Online]. Available: http://dx.doi.org/10.1007/3-540-48658-5_32

[6] H. Krawczyk, “LFSR-based hashing and authentication,” in *Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, ser. CRYPTO ’94. London, UK, UK: Springer-Verlag, 1994, pp. 129–139.

[7] E. Dubrova, M. Naslund, G. Selander, and F. Lindqvist, “Cryptographically secure CRC for lightweight message authentication,” Tech. Rep. 2015/035, January 2015, cryptology ePrint Archive. [Online]. Available: http://eprint.iacr.org/2015/035

[8] E. Dubrova, M. Naslund, and G. Selander, “CRC-based message authentication for 5G mobile technology,” in *Proceedings of 1st IEEE International Workshop on 5G Security*, August 2015.

[9] S. Gao and D. Panario, “Tests and constructions of irreducible polynomials over finite fields,” in *Foundations of Computational Mathematics*, F. Cucker and M. Shub, Eds. Springer Berlin Heidelberg, 1997, pp. 346–361. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-60539-0_27

[10] E. Dubrova, M. Naslund, G. Selander, and F. Lindqvist, “Message authentication based on cryptographically secure CRC without polynomial irreducibility test,” *Cryptography and Communications*, 2017.

[11] D. Boswarthick, O. Elloumi, and O. Hersent, *M2M Communications: A Systems Approach.* John Wiley & Sons, 2012.

[12] B. Schneier, *Applied cryptography (2nd ed.): protocols, algorithms, and source code in C.* New York, NY, USA: John Wiley & Sons, Inc., 1995.

[13] J. Daemen and V. Rijmen, “AES proposal: Rijndael,” April 2003, national Institute of Standards and Technology.

[14] M. Robshaw, “Stream ciphers,” Tech. Rep. TR - 701, July 1994. [Online]. Available: citeseer.ist.psu.edu/robshaw95stream.html

[15] M. Hell, T. Johansson, A. Maximov, and W. Meier, “The Grain family of stream ciphers,” *New Stream Cipher Designs: The eSTREAM Finalists, LNCS 4986*, pp. 179–190, 2008.

[16] C. Cannière and B. Preneel, "Trivium," *New Stream Cipher Designs: The eSTREAM Finalists, LNCS 4986*, pp. 244–266, 2008.

[17] E. Dubrova and M. Hell, "Espresso: A stream cipher for 5g wireless communication systems," *Cryptography and Communications*, pp. 1–17, 2015.

[18] E. Dubrova, "A transformation from the Fibonacci to the Galois NLFSRs," *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5263–5271, Nov. 2009.

[19] ——, "An equivalence-preserving transformation of shift register," *Sequences and Their Applications - SETA'2014, LNCS 8865*, pp. 187–199, 2014.

[20] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection," *IEEE Design Test of Computers*, vol. 27, no. 1, pp. 10–25, 2010.

[21] E. Brunvand, *Digital VLSI Chip Design with Cadence and Synopsys CAD Tools*. Pearson, 2009.

[22] E. Seligman, T. Schubert, and A. K. Kumar, *Formal Verification: An Essential Toolkit for Modern VLSI Design*. Morgan Kaufmann, 2015.

[23] P. Gupta and E. Papadopoulou, "Yield analysis and optimization," in *The Handbook of Algorithms for VLSI Physical Design Automation*. RC Press, 2011.

[24] S. Shah, "NSA, GCHQ ban Lenovo's PSs due to security concerns," July 2013.

[25] D. Goodin, "We cannot trust Intel's and Via's chip-based crypto FreeBSD developers say," Dec. 2013.

[26] G. Becker, F. Regazzoni, C. Paar, and W. P. Burleson, "Stealthy dopant-level hardware Trojans," *Proceedings of Cryptographic Hardware and Embedded Systems (CHES'2013), LNCS 8086*, pp. 197–214, 2013.

[27] Federal Information Processing Standards Publication, "Security requirements for cryptographic modules: FIPS PUB 140-2," 2001.

[28] E. Barker and J. Kelsey, "Recommendation for random number generation using deterministic random bit generators: NIST 800-90A," 2012.

[29] T. Sugawara, D. Suzuki, R. Fujii, S. Tawa, R. Hori, M. Shiozaki, and T. Fujimo, "Reversing stealthy dopant-level circuits," *Proceedings of Cryptographic Hardware and Embedded Systems (CHES'2014), LNCS 8731*, pp. 112–126, 2014.

[30] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," in *IEEE Symposium on Security and Privacy (SP'07)*, May 2007, pp. 296–310.

[31] M. Damiani, P. Olivo, M. Favalli, S. Ercolani, and B. Ricco, "Aliasing in signature analysis testing with multiple input shift registers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 12, pp. 1344–1353, 1990.

[32] E. Dubrova, M. Näslund, G. Carlsson, J. Fornehed, and B. Smeets, "Two countermeasures against hardware Trojans exploiting non-zero aliasing probability of BIST," *Journal of Signal Processing Systems*, pp. 1–11, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11265-016-1127-4

[33] E. Dubrova, M. M. Näslund, G. Carlsson, and B. Smeets, "Keyed logic BIST for Trojan detection in SoC," in *Proceedings of International Conference of System-on-Chip (SoC'2014)*, 2014.