Evaluating the Randomness of the RNG in a Commercial Smart Card

Wellinton Costa Ribeiro[1], Marcus Tadeu Pinheiro Silva[1]

[1]Departamento de Eletrônica e Biomédica – Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Campus II – 30510-000 – Belo Horizonte – MG – Brazil

wcostaee@gmail.com, tadeu@cefetmg.br

Abstract. *This paper brings results concerning the quality evaluation for the pseudo-random number generator (PRNG) in a commercial smart card. The RNG is a fundamental part for the cryptography carried out in several applications. We have acquired a huge quantity of random numbers from three samples of a commercial smart card. These data were evaluated using the statistical computation package developed by National Institute of Standards and Technology. In order to be used as gold benchmark and to validate our methodology, we have also tested the true random number generator (TRNG) included in a commercial integrated circuit. Our results show that the card PRNG owns quality too inferior than the TRNG. Due to card vendor confidentiality policy is not possible state the tested PRNG is base for the device cryptography. However, if this occurs, results lead us to conclude the tested PRNG is not adequate to provide the required security in the systems that adopt the evaluated smart card.*

1. Introduction

Typically, smart cards include a cryptographic unit, because the applications of this type of device require high security. However, eventual attackers will has total access to the smart card, what will allow to them act without restrictions existing in other applications where cryptography is used to guarantee security. Smart cards are also more vulnerable due to limitations imposed to them, including small dimensions, low cost and reduced power consumption [Akram 2012]. These factors lead to a security level lower than desirable, inasmuch as the cryptographic unit design can not to adopt the better solutions for some aspects. For example, the designer is forced to use a pseudo-random number generator (PRNG), while the ideal would be use a true random number generator (TRNG) instead.

In recent years, some researchers have reported failures in the cryptographic unit of commercial smart cards. The majority of these works has focused in the power consumption analysis methods in order to determine the private keys from the devices, what indicates necessity of new countermeasures from the vendors to avoid this vulnerability [Balasch 2012]. Other part of these works showed that the private keys can be determined when the PRNG in the smart card presents a weak randomness [Bernstein 2013].

In this paper we present results about the quality assessment for the PRNG in a commercial smart card. In order to be a reference for our methodology we have also applied it to the TRNG included in a commercial integrated circuit (IC). Although the tested smart card has resource to provide direct access to the number generated in the PRNG, the device documentation makes none assertion about to be exclusively this PRNG the base for the cryptography.

2. Qualifying an RNG

When the internal structure of an RNG is not exactly known, assessment of its quality can be carried out only by statistical methods. So, a myriad of statistical tests has been developed to qualify the level of randomness for the sequences produced by an RNG. However, even the best RNG will eventually produce sequences that fail in one or other test. Consequently, the quality judgment for an RNG is a cumbersome process, involving generation of huge sequences and the assessment these sequences using many diverse tests. Several researchers have developed software tools to make this process easier [L'Ecuyer 2007][Marsaglia 2002][Rukhin 2010]. For the analysis in this paper we chose the Statistical Test Suite (STS) developed by National Institute of Standards and Technology (NIST) [Rukhin 2010].

3. Data Acquisition Setup

In this work three identical smart cards were tested. These cards includes only a contactless interface, which is compliant with the radio-frequency identification (RFID) standard ISO 14443B, being devices focused in access control applications, as stated in vendor documentation. As usual in RFID standards, the communication data rate is very limited. Furthermore, in the command to obtain data from the PRNG, only one byte is made available after each access. So, with the tested smart card the maximum data throughput achieved was 150 bps. An RFID reader was developed specifically for the work. This reader includes a microcontroller, a text LCD, a real time clock (RTC) with battery, a serial interface for communication with the host personal computer (PC) and a 1 MByte flash memory, the Intel i82802 IC. In order to validate the data acquisition setup and provide a gold benchmark for the results, we also carried out the generation of random numbers from the TRNG included in the i82802 [Taylor 2011]. Figure 1 shows the system block diagram and Figure 2 the corresponding assembled prototype. The firmware for the reader microcontroller was developed using C language, while for the PC was developed a data acquisition program written in Java. Concerning code execution, the smart card used is not user programmable, owning a proprietary operating system (OS) written in non-volatile memory during the manufacture. Further information about this OS is not available, in reason of the vendor confidentiality policy.

4. Methodology

For the three smart cards and i82802, we acquired huge bit streams, which were stored in files on a PC that controlled the data acquisition experiments. For the cards these bits streams were acquired in two modes. One these modes, called *on-off*, consisted of cycles on/off for the card, i.e., for each new number generated the card was power-off, from the interruption of the reader RF field, and some time after it was power-on again to gene-

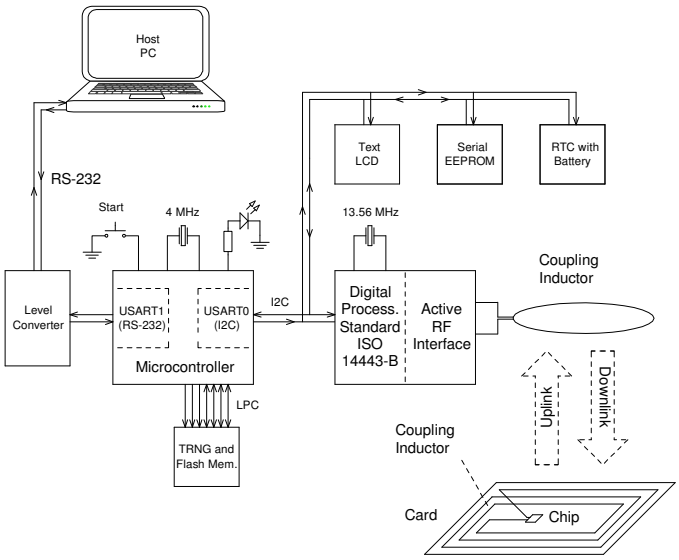

Figure 1. Block diagram of the system developed for this work.

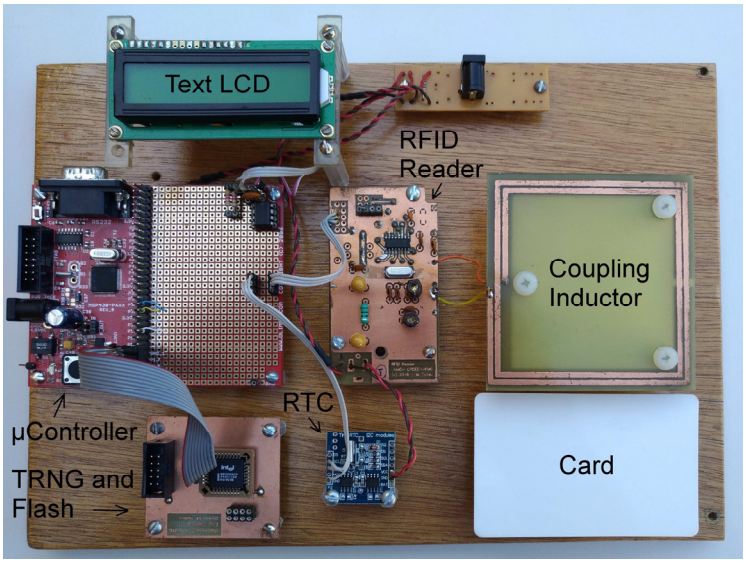

Figure 2. Prototype corresponding to figure 1.

rate the next number. In the other mode, called *continuous*, the card was continuously powered during the whole bit stream acquisition. The bit streams were processed by STS in blocks of n bits. When testing a RNG, in each statistical test is indispensable a high value for n, typically equal or upper to 10^5. Besides that, only is possible to qualify the generator if were acquired m sequences of n bits, since it is the rate of sequences validated as random that will establish the generator judgment in each test. The STS assumes an approval rating equal or upper 96% of the m sequences will qualify the generator as satisfactory. Our experiments were carried out with n equal to 10^5, m equal to 10^2, in compliance with values commonly reported in the literature [Boorghany

Table 1. Rate of passing sequences on the NIST-STS. Tests for mode continuous. A generator is approved for a rate better or equal to 96%, except for the Universal and FFT tests, where the threshold is 90%. Not approval condition is indicated by an underlined value.

	i82802 (%)	Card 1 (%)	Card 2 (%)	Card 3 (%)
Frequency	99	93	81	73
Block Frequency	100	100	94	62
Cumul. Sums (fwd.)	98	90	81	74
Cumul. Sums (rev.)	99	94	81	73
Runs	99	94	75	88
Longest Run	100	71	80	20
Rank	100	99	99	29
FFT	100	100	70	0
Overlapping Templates	99	98	97	39
Approximate Entropy	100	6	12	0
Random Excursions	100	100	100	100
Random Exc. Variant	100	100	100	100
Serial	98	94	68	0
Linear Complexity	97	97	99	87
Universal	100	90	70	60

2014]. FFT and Universal tests of the STS require $n \geq 10^6$. So, for these specific tests, we have utilized n=10^6, m=10 and adopted an approval threshold equal to 90%. Due to RF interface low data rate, acquisition of 10 Mbits from each card was a time consuming task, with an approximate duration of 18.5 hours and 7 days, for the continuous and on/off modes, respectively. Concerning the i82802, the acquisition time was equal to 59 minutes and for this generator only the continuous mode was used.

5. Results

Tables 1 and 2 show the results in the continuous mode. Table 1 gives the results for 15 tests of NIST-STS. The i82802 IC has passed all tests. By contrast, the PRNG into cards 1, 2 and 3 has failed in 7, 10, and 13 tests, respectively. In addition to the tests in Table 1, NIST-STS also includes the Non-Overlapping Templates (NOT) test, whose results are presented in Table 2. NOT test validates each one of the m sequences over 148 different 9 bits templates; each template owns a non-periodicity character in its bit pattern. For example, Table 2 shows the i82802 failed only for one (1) in 148 templates. Specifically, for the template "101010000", where 5% of sequences failed; until 4% qualifies the generator as "passed in test". In the case of the smart cards the worst performance was for the number 3, where all templates leads to a not approval result and in the worst case 72% of sequences were rejected. If we consider each one of the 148 processed templates as a test itself, we will obtain a set of 163 tests. The results for

Table 2. Results for the NIST-STS Non-Overlapping Templates (NOT) test, mode continuous.

	i82802	Card 1	Card 2	Card 3
number failed templates	1	43	23	148
worst rate of not passed	5%	20%	13%	72%

the i82802 are compatible with the expected for a TRNG, since only one test had a not approval result and, even in this case, the rejection was by a minimum margin. Meanwhile, the card 2, that with better evaluation, failed in 33 tests. However, in some tests even this better card had a high rejection result, reaching 88% in the Approximate Entropy test.

Table 3 presents the results for the cards in on-off mode. It is immediately seen that the card 3 had a performance yet worst than in continuous mode. Between the cards 1 and 2, again the card 2 showed a better result. However, in general all cards had weak performance in comparison with the continuous mode. The on-off mode reported in Table 3 makes possible to examine the aspect of seed for the PRNG, although indirectly [Rank 2003]. It can be stated that the seed generation in the tested cards is very far from

Table 3. Rate of passing sequences on the NIST-STS. Tests for mode on-off. NC means "not carried out". Not approval condition is indicated by an underlined value

	Card 1 (%)	Card 2 (%)	Card 3 (%)
Frequency	15	21	12
Block Frequency	24	33	36
Cumul. Sums (fwd)	2	14	6
Cumul. Sums (rev)	4	15	6
Runs	5	7	4
Longest Run	32	32	22
Rank	85	88	75
FFT	0	0	0
Overlapping Templates	42	58	34
Approximate Entropy	11	10	14
Random Excursions	NC	NC	NC
Random Exc. Variant	NC	NC	NC
Serial	82	85	68
Linear Complexity	99	100	99
Universal	0	0	1

an ideal entropy source, leading to weak randomness shown.

From the results, we can note the linear complexity test was that where the cards obtained the best approval level. In the Table 1, only the card 3 was not approved in this test, but by a not significant margin. In Table 3 this was the only test where the cards were approved. The focus of the linear complexity test is the length of a linear feedback shift register (LFSR) [Rukhin 2010]. So, in general way, the results allows to infer that the tested smart card includes an RNG based on LFSR, as commonly happens in the commercial designs [Akram 2012].

6. Conclusion

In this paper we have presented results concerning the quality evaluation for the RNG in a commercial smart card. Although it is not possible to claim that the evaluated RNG is effectively used in the card cryptographic unit, if that happen, the obtained results shows the smart card will represent significant weakness in the system security. In particular, one of the tested cards has presented so weak results that we believe in some manufacturing problem. If the proportion of these defective cards in a production batch becomes significant, it will arise an additional concern about the system security. Also, we have shown the convenience in adopt a TRNG device as gold benchmark, since it allow to make sure that is valid the used methodology.

Acknowledgment

The authors express your thanks to the Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG) by support given to this work through its research scholarship program managed by the Departamento de Pesquisa e Pós-Graduação (DPPG).

References

Akram, R.N. et al. (2012). Pseudorandom number generation in smart cards: an implementation, performance and randomness analysis. In *5th IEEE International Conference on New Technologies, Mobility and Security (NTMS)*, May 7-10, Istanbul, Turkey.

Balasch, J. et al. (2012). Power analysis of Atmel CryptoMemory - recovering keys from secure EEPROMs. In *Topics in Cryptology - CT-RSA 2012, The Cryptographers' Track at the RSA Conference*, Lecture Notes in Computer Science 7178, O. Dunkelman (ed.), Springer-Verlag, pages 19-34.

Bernstein, D.J. et al. (2013). Factoring RSA keys from certified smart cards: Coppersmith in the wild. In *International Conference on the Theory and Application of Cryptology and Information Security*, Dec. 1-5, Bangalore, India, pages 341-360.

Boorghany, A. et al. (2014). Random data key generation evaluation of some commercial tokens and smart cards. In *Proc. of 11th International ISC Conference on Information Security and Cryptology*, Sept. 3-4, Tehran, Iran, pages 49-54.

L'Ecuyer, P., and Simard, R. (2007). TestU01: A C library for empirical testing of random number generators. In *ACM Transactions on Mathematical Software*, 33, (4), article 22.

Marsaglia, G., and Tsang, W. W. (2002). Some difficult-to-pass tests of randomness. In *Journal of Statistical Software*, 7,(3), pages 1-9.

Rankl, W. and Effing, W. (2003). Smart Card Handbook. 3rd edition, Wiley, Chichester, England, pages 210-213.

Rukhin, A. et. al. (2010). A statistical test suite for random and pseudo-random number generators for cryptographic applications. *NIST special publication 800-22*, National Institute of Standards and Technology, USA.

Taylor, G. and Cox, G. (2011). Digital randomness. In *IEEE Spectrum*, vol. 48, no. 9, pages 32-58.

