Vulnerability of RSA Algorithm

Aleksandra V. Markelova
Information Security Department
Bauman Moscow State Technical University
Moscow, Russia
markelova_bmstu@mail.ru

***Abstract*—This paper is dedicated to ROCA-vulnerability that was detected by scientists from Masaryk University, Czech. Their investigation offers low-cost algorithm of factorization of RSA module for special type of keys generated by some widely used cryptographic library. They proposed a practical factorization method for various key lengths including 1024 and 2048 bits. This attack requires no additional information except for the value of the public key and does not depend on a weak or a faulty random number generator. We examine the possibility of modification of type of keys to embed the trapdoor with universal protection into key generator. In some cases we can design Secretly Embedded Trapdoor with Universal Protection in the generator of RSA key. This problem is serious and relevant for all closed (so-called black-box) implementations of cryptographic algorithm in user's library or device. The first section of this article ("Introduction") is devoted to the history of the issue. It also describes the damage caused by vulnerability ROCA. The second section ("Fingerprint of weak keys") describes the criterion that the key pair is vulnerable to ROCA. The third section ("Factorization") is dedicated to the attack ROCA. It also estimates the running time of the algorithm. In the fourth section ("The trapdoor with universal protection") we will consider the possibility of using SETUP mechanism in the implementation of RSA.**

Keywords— information security, cryptanalysis, vulnerability, ROCA, RSA, Coppersmith's algorithm, factorization, weak keys, kleptography, trapdoor with protection, backdoor, SETUP

I. INTRODUCTION

On November, 2017 Estonian government made a statement about blocking 760 000 certificate of ID cards issued after October 16, 2014. The reason for this statement was the vulnerability ROCA (Return of Coppersmith's Attack) discovered by scientists from Masaryk University, Czech [1].

Toomas Ilves, the former president of Estonia, said that he believed millions of people in countries had been affected by the ROCA flaw, but their authorities were remaining "silent". In particular, according to the researchers of the Enigma Bridge, similar problems are possible with ID card of Spain.

There is a fast detection algorithm to verify whether a particular key is vulnerable to attack. This verification is based on the properties of the public moduli.

Vulnerable keys were also found in some authentication tokens, in the TPM (Trusted Platform Modules), in PGP. Google, HP, Lenovo and Fujitsu released updates for their software products susceptible to this attack.

Recall that the public key of the RSA algorithm is a pair (n, e), where n is the product of two large primes and $\gcd(e, \phi(n))=1$. Private key is number d such that $ed=1 \pmod{\phi(n)}$. Some implementations also store prime divisors of n as part of the private key.

Thus, RSA requires two large random primes p and q, that can be obtained by generating a random candidate number (usually with half of the bits of n) and then testing it for primality. If the candidate is found to be composite, the process is repeated with a different candidate.

Since the RSA algorithm is very popular, many researches are devoted to its reliability [2, 3, 4, 5, 6, 7].

RSA security is based on the integer factorization problem.

The most effective modern factorization algorithms (such as quadratic sieve [8], number field sieve [9], special number field sieve [10, 11]) have subexponential complexity [12] and in the general case do not allow hacking RSA with large key lengths.

However, if the numbers p and q are a special type, then the time of factorization of the number $n = pq$ can be reduced.

II. FINGERPRINT OF WEAK KEYS

There is no common practice for developers of cryptographic library how to generate RSA key pair. But there are many recommendations regarding how to select suitable primes p and q [13, 14, 15, 16] to be later used to compute the private key and public moduli.

In 2016, scientists from Masaryk University analyzed implementations of RSA algorithm and key pairs from 22 open- and closed- source libraries and from 16 different smart cards [17]. In particular, the library RSALib used in Estonian ID cards was investigated. This library utilizes an acceleration algorithm called "Fast Prime".

The foundations of "Fast Prime" date back to the year 2000. According to its developers, its use started around ten years later after thorough reviews. As a sub-part of one cryptographic software library which is supplied to customers as a basis for their own development, this software function was certified by the BSI (Federal Office for Information Security) in Germany.

When compared to other implementations and theoretical expectations on distribution of prime numbers, the keys from RSALib exhibited a non-uniform distribution of (p mod x) and (n mod x) for small primes x.

Further studies have shown that all RSA primes generated by the RSALib have the following form [1]:

$$p=k*M + (65537^a \bmod M) \quad (1)$$

The integers k and a are unknown, and RSA primes differ only in their values of a and k for keys of the same size. The integer M is fixed for each key size (table I) and equal to the product of the first successive primes:

$$M = P_m = \prod p_i = 2*3*5*7*\ldots*p_m \quad (2)$$

TABLE I.

Key size	M
512	P_{39}#=167#
1024	P_{71}#=353#
2048	P_{126}#=701#
3072	P_{126}#=701#
4096	P_{225}#=1427#

In this case the following comparison takes place:

$$n = 65537^c \bmod M \quad (3)$$

The existence of the discrete logarithm $c = log_{65537}n \bmod M$ is used as the fingerprint of weakness of the key pair. In general, verification of solvability of the comparison (3) can be difficult [18]. But if M is of the form (2) then the problem is easily solved.

The number of residues modulo M for which there exists a logarithm at base 65537 is equal to ord_M65537. For randomly generated prime numbers, the remainders from dividing their product by M are distributed uniformly in the multiplicative group of residues modulo M. Therefore, the probability for the number n to satisfy the comparison (3) is $ord_M65537/\phi(M)$.

For the M used, the value of ord_M65537 is much less than $\phi(M)$. For example, $ord_M65537=2^{62,09}$, $\phi(M)=2^{215,98}$ for RSA-512. Thus, the probability of the false positive result does not exceed $2^{62-216}=2^{-154}$. This probability is even smaller for larger keys (table II).

So the existence of the discrete logarithm is the strong fingerprint of the weak keys.

TABLE II.

Key size	Size of M	ord_M65537
512	$2^{219,19}$	$2^{62,09}$
1024	$2^{474,92}$	$2^{134,73}$
2048	$2^{970,96}$	$2^{255,78}$
3072	$2^{970,96}$	$2^{255,78}$
4096	$2^{1962,19}$	$2^{434,69}$

Online and offline versions of this test are already developed. They are freely available on the Internet.

III. FACTORIZATION

Algorithm ROCA iterates over values of a in (1) and use Coppersmith's algorithm [19] to attempt to find k. To reduce the search, the modulus M is replaced by its divisor M', for which $ord_{M'}65537$ is small and $log_2M'>log_2n/4$. The number M' is selected once for each RSA key size (table III).

TABLE III.

Key size	Size of M'	$ord_{M'}65537$ /2
512	$2^{140,77}$	$2^{19,20}$
1024	$2^{474,92}$	$2^{29,04}$
2048	$2^{970,96}$	$2^{34,29}$
3072	$2^{552,50}$	$2^{99,29}$
4096	$2^{1098,42}$	$2^{55,05}$

The size of M' (condition $log_2M'>log_2n/4$) is chosen to apply the modification of the Coppersmith's attack [19], which is successful in case of knowing $l/4$ the least significant bits of the number p, where l is the key size.

Coppersmith's attack was repeatedly modified. Now there are various attacks on RSA based on Coppersmith's algorithm. For example, factorization algorithms have been developed for cases when the lowest bits of the number p are known or when primes p and q share bits in the middle ([20, 21, 22, 23]).

In attacks of this class, we choose a polynomial $f(x)$ having a small root x_0 in the residue field:

$$f(x_0) = 0 \bmod p \quad (4)$$

$$|x_0|<X \quad (5)$$

Then we construct a polynomial $g(x)$ satisfying the following conditions:

$$g(x) = \sum a_i f_i(x), \quad (6)$$

a_i is integer,

$f_i(x)$ and $f(x)$ have the same roots modulo p, (7)

$$|g(x_0)|<p. \quad (8)$$

The coefficients a_i are chosen by the LLL-lattice method [24]. It follows from (6) and (7) that $g(x)$ has the same roots modulo p as $f(x)$. Well then $g(x_0)=0 \bmod p$. Taking into account (8), we see that $g(x_0)=0$. Thus x_0 can be found by standard methods for finding the roots of a polynomial (e.g., the Berlekamp-Zassenhaus algorithm [25, 26]).

We denote the process of finding x_0 as *Coppersmith(f(x), n, β, m, t, X)*, where n is RSA-modulus, β is the upper bound for the ratio of $\log_2 p$ and $\log_2 n$, m and t are optimization parameters of Coppersmith algorithm, X is from (5).

Algorithm ROCA works as follows:

1. $c' = \log_{65537} n \bmod M'$

2. $\beta=0.5$, $X=2*n\beta/M'$

3. For all a' in $[c'/2; (c'+ ord_{M'}65537)/2]$:

3.1 $f(x) = x + (M'^{-1} \bmod n)*(65537^{a'} \bmod M')(\bmod n)$

3.2. $k' = Coppersmith(f(x), n, \beta, m, t, X)$

3.3. $p=k'*M' + (65537^{a'} \bmod M')$

3.4 If p is a nontrivial divisor of n, then finish.

The running time of this algorithm is shown in tables IV and V. Two time values were explicitly checked by the scientists from Masaryk University on the university cluster.

TABLE IV.

Key size	Factorization on Intel Xeon E5-2650 v3 @3GHz Q2/2014
512	1,93 hours (verified)
1024	97,1 days (verified)
2048	140,8 years
3072	$2,84*10^{25}$ years
4096	$1,28*10^{9}$ years

Others were extrapolated based on known algorithm properties and processing power.

The algorithm is well suited for parallel computations, since the approbation of different values of a' can pass independently of each other.

For example, you can rent 1000 cores on Amazon AWS. In this case it will take 45 minutes to find the 1024-bit key. It will take about 17 days to find the 2048-bit key. At the same time the cost of the attack will not increase, because it is calculated based on the price of one processor hour.

TABLE V.

Key size	Factorization on 2x Intel E5-2666 v3@2.90GHz (Amazon AWS, c4 instance) and resource rental cost
512	0,63 hours, $0,063
1024	31,71 days, $76
2048	45,98 years, $40 305
3072	$9,28*10^{24}$ years, $8,13*10^{27}$
4096	$4,18*10^{8}$ years, $3,66*10^{11}$

Note that 4096-bit RSA key is not practically factorizable now, but may become so, if the attack is improved.

This is very possible, since the Coppersmith's algorithm and the lattice-based method are constantly improving [27]

IV. THE TRAPDOOR WITH UNIVERSAL PROTECTION

Did the developers of the RSALib library know about the vulnerability of their key generator? If they knew, then such an implementation could be considered as an implementation with a trapdoor (or a backdoor). But in this case the special form (2) of the number M makes it possible for any observer to restore the prime numbers (1). Because of this, an access to the backdoor is provided not only for developers or authorized special agency, but also for attacker.

The difference between a trapdoor and a backdoor is the degree of protection from a third-party offender [28]. The science of trapdoors is called "kleptography" [29]. It was shown that a cryptosystem, when implemented as a black-box (i.e., when the user has only input/output access to the hardware or software cryptographic facility), can be designed such that it gives a unique advantage to the attacker. This is accomplished using SETUP (Secretly Embedded Trapdoor with Universal Protection) mechanisms. SETUP is a system, hacking which will be available only to the developer.

This topic was examined in detail by Young A. and Yung M. [30]. In particular, they considered a number of kleptographic attacks on the RSA algorithm [30, 31, 32].

Issues of the implementation of SETUP-mechanisms are an actual direction of modern cryptography. Bellare, Paterson, and Rogaway initiated a formal study of such attacks on symmetric key encryption algorithms, demonstrating that kleptographic attacks can be mounted in broad generality against randomized components of cryptographic systems [33]. Russell, Tang, Yung and Zhou enlarged the scope of work on the problem by permitting adversarial subversion of (randomized) key generation; in particular, they initiated the study of cryptography in the complete subversion model, where all relevant cryptographic primitives are subject to kleptographic attacks [34].

Obviously, the most effective attack for the intruder and the greatest danger to the user are backdoors and trapdoors into the key pair generator. The simplest attack on RSA key generator is using a fixed prime number p [30]. But this attack is detectable and it is breakable without reverse engineering.

The problem of detectability in the attack can be avoided by using pseudorandom values instead of random values and having the initial seed known only to the device and the attacker. This type of the backdoor is similar to the implementation of the key generator in the RSALib library.

The fundamental weakness in the pseudorandom number generator backdoor attack is that once an intermediate seed is exposed, the future operation of the device is compromised [30]. Modern methods of protection against backdoor in pseudorandom generators assume randomness checking at any time without any notification, so the device is forced to behave honest [35].

Degabriele, Paterson, Schuldt and Woodage conducted a full-scale study of pseudorandom number generators [36] and give efficient constructions of BPRGs (backdoored pseudorandom number generators) for which, given a single generator output, Big Brother can recover the initial state and, therefore, all outputs of the BPRG. They give an impossibility result: they provide a bound on the number of previous phases that Big Brother can compromise as a function of the state-size of the generator: smaller states provide more limited backdooring opportunities for Big Brother.

Leaving aside the moral and legal aspects of the issue, we will consider whether it is possible to construct a trapdoor with universal protection based on the same idea as the vulnerability of ROCA.

Let's choose numbers M and w satisfying the following conditions:

1) the decomposition of the number M into prime factors is known

2) $\phi(M)$ is decomposed into small primes:

$$p_i \mid \phi(M): p_i < C \tag{9}$$

3) value $ord_M w$ is small:

$$ord_M w < \delta \tag{10}$$

4) the size of M approximately equal to optimal Coppersmith's algorithm parameters:

$$l = log_2 M \sim log_2 n / 4 \tag{11}$$

5) there are many primes of the form

$$p = k*M + (w^a \bmod M) \tag{12}$$

The last condition ensures that the generator will construct many different prime numbers.

The set (C, δ, l) specifies the parameters for selecting the numbers M and w. If you know this set, M and w then you can restore the user's private key from the public key applying the ROCA analog.

If M has a large prime divisor, then it is not possible to estimate the distribution (p mod x) and (n mod x) for all numbers x of this size. Then an attacker will not receive any additional information about the key.

V. CONCLUSIONS

The vulnerability of ROCA once again recalled the importance of analyzing the used cryptographic libraries. In some cases, problems can be identified by evaluating the specificity of the generated keys.

However, the RSA algorithm allows you to build in the key generator secretly embedded trapdoor with universal protection. The feature of this trapdoor is that it is not detectable for the analyst (it is impossible even to prove its presence in the implementation), but it allows the developer to calculate the user's private keys.

REFERENCES

[1] Nemec M., Sys M., Svenda P., Klinec D., Matyas V. The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli. 2017. CCS'17 Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, p. 1631-1648. DOI: 10.1145/3133956.3133969.

[2] Nitaj A. (2012) A New Attack on RSA and CRT-RSA. In: Mitrokotsa A., Vaudenay S. (eds) Progress in Cryptology – AFRICACRYPT 2012. AFRICACRYPT 2012. Lecture Notes in Computer Science, vol 7374, p. 221-233. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-31410-0_14.

[3] Nitaj A. A new attack on RSA with two or three decryption exponents. Journal of Applied Mathematics and Computing (2013) Vol. 42, Issue 1-2, p. 309–319. DOI: 10.1007/s12190-012-0618-0.

[4] Peng L., Hu L., Lu Y., Xu J., Huang Z.. Cryptanalysis of Dual RSA Designs, Codes and Cryptography. (2017) Vol. 83, Issue 1, p. 1-21. DOI: 10.1007/s10623-016-0196-5.

[5] Barbu G. et al. (2013) Combined Attack on CRT-RSA. In: Kurosawa K., Hanaoka G. (eds) Public-Key Cryptography – PKC 2013. Lecture Notes in Computer Science, vol 7778, p. 198-215. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-36362-7_13.

[6] Bunder M., Nitaj A., Susilo W., Tonien J. (2016) A New Attack on Three Variants of the RSA Cryptosystem. In: Liu J., Steinfeld R. (eds) Information Security and Privacy. ACISP 2016. Lecture Notes in Computer Science, vol 9723, p. 258-268. Springer, Cham. DOI: 978-3-319-40367-0_16.

[7] Bauer A., Jaulmes E., Lomné V., Prouff E., Roche T. (2014) Side-Channel Attack against RSA Key Generation Algorithms. In: Batina L., Robshaw M. (eds) Cryptographic Hardware and Embedded Systems – CHES 2014. CHES 2014. Lecture Notes in Computer Science, vol 8731, p. 223-241. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-662-44709-3_13.

[8] Pomerance C., Analysis and Comparison of Some Integer Factoring Algorithms, in Computational Methods in Number Theory, Part I, H.W. Lenstra, Jr. and R. Tijdeman, eds., Math. Centre Tract 154, Amsterdam, 1982, p. 89-139.

[9] Lenstra A.K., Lenstra H.W., Jr., Manasse M.S., Pollard J.M. (1990). The number field sieve. STOC '90 Proceedings of the twenty-second annual ACM symposium on Theory of computing, p. 564-572, ISBN 0-89791-361-2. DOI: 10.1145/100216.100295.

[10] Lenstra A.K., Lenstra H.W., Jr., Manasse M.S., Pollard J.M. (1993). The Factorization of the Ninth Fermat Number. Mathematics of

Computation T. 61 (1993): p. 319–349, DOI: 10.1090/S0025-5718-1993-1182953-4.

[11] Buhler J.P., Lenstra H.W., Pomerance C. (1993) Factoring integers with the number field sieve. In: Lenstra A.K., Lenstra H.W. (eds) The development of the number field sieve. Lecture Notes in Mathematics, vol 1554, p. 50-94. Springer, Berlin, Heidelberg. DOI: 10.1007/BFb0091539.

[12] Pomerance C. A tale of two sieves. Notices Amer. Math. Soc. 43 (1996), p. 1473-1485.

[13] Gordon J.(1985) Strong Primes are Easy to Find. In: Beth T., Cot N., Ingemarsson I. (eds) Advances in Cryptology. EUROCRYPT 1984. Lecture Notes in Computer Science, vol 209, p. 216-223. Springer, Berlin, Heidelberg. DOI: 10.1007/3-540-39757-4_19.

[14] Loebenberger D., Nusken M.. Notions for RSA Integers. In International Journal of Applied Cryptography, Vol. 3, No. 2 (2014), p. 116–138. DOI: 10.1504/IJACT.2014.062723.

[15] Maurer U.M.. Fast generation of prime numbers and secure public-key cryptographic parameters. Journal of Cryptology, Vol. 8, Issue 3 (1995), p. 123–155. DOI: 10.1007/BF00202269.

[16] Benhamouda F., Ferradi H., Géraud R., Naccache D. (2017) Non-interactive Provably Secure Attestations for Arbitrary RSA Prime Generation Algorithms. In: Foley S., Gollmann D., Snekkenes E. (eds) Computer Security – ESORICS 2017. ESORICS 2017. Lecture Notes in Computer Science, vol 10492, p. 206-223. Springer, Cham. DOI: 978-3-319-66402-6_13.

[17] Svenda P., Nemec M., Sekan P., Kvasnovskyy R., Formanek D., Komarek D., Matyas V.. 2016. The Million-Key Question – Investigating the Origins of RSA Public Keys. In The 25th USENIX Security Symposium (USENIX Security'16). USENIX, p. 893–910. DOI: 10.13140/rg.2.1.3759.3848.

[18] Markelova A.V. Solvability of the problem of taking the discrete logarithm. Moscow University Mathematics Bulletin, Vol. 63, Issue 6 (2008), p. 225-228. DOI: 10.3103/S0027132208060016.

[19] Coppersmith D. (1996) Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known. In: Maurer U. (eds) Advances in Cryptology — EUROCRYPT '96. EUROCRYPT 1996. Lecture Notes in Computer Science, vol 1070, p. 178–189. Springer, Berlin, Heidelberg. DOI: 10.1007/3-540-68339-9_16.

[20] Lu Y., Zhang R., Lin D. (2013) Factoring RSA Modulus with Known Bits from Both p and q: A Lattice Method. In: Lopez J., Huang X., Sandhu R. (eds) Network and System Security. NSS 2013. Lecture Notes in Computer Science, vol 7873, p. 393-404. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-38631-2_29.

[21] Akchiche O., Khadir O.. Factoring RSA moduli with primes sharing bits in the middle. AAECC (2017), p. 1–15. DOI: 10.1007/s00200-017-0340-0.

[22] Nitaj A. (2013) An Attack on RSA Using LSBs of Multiples of the Prime Factors. In: Youssef A., Nitaj A., Hassanien A.E. (eds) Progress in Cryptology – AFRICACRYPT 2013. AFRICACRYPT 2013. Lecture Notes in Computer Science, vol 7918, p. 297-310. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-38553-7_17.

[23] May A. (2009) Using LLL-Reduction for Solving RSA and Factorization Problems. In: Nguyen P., Vallée B. (eds) The LLL Algorithm. Information Security and Cryptography, p. 315–348. Springer, Berlin, Heidelberg. DOI: 0.1007/978-3-642-02295-1_10.

[24] Lenstra A.K., Lenstra H.W., Lovász L. Factoring polynomials with rational coefficients. Math. Ann. 261, 4 (1982), p. 515–534. DOI: 10.1007/BF01457454.

[25] Berlekamp E.R. Factoring Polynomials Over Large Finite Fields. Math. Comp. 24 (1970), p. 713-735. DOI: 10.1090/S0025-5718-1970-0276200-X.

[26] Cantor D.G. and Zassenhaus H. A New Algorithm for Factoring Polynomials Over Finite Fields. Math. Comp. 36, 154 (1981), p. 587–592. DOI: 10.2307/2007663.

[27] Lu Y., Peng L., Kunihiro N. (2018) Recent Progress on Coppersmith's Lattice-Based Method: A Survey. In: Takagi T., Wakayama M., Tanaka K., Kunihiro N., Kimoto K., Duong D. (eds) Mathematical Modelling for Next-Generation Cryptography. Mathematics for Industry, vol 29, p. 297-312. Springer, Singapore. DOI: 10.1007/978-981-10-5065-7_16

[28] Zhukov A.E. Cryptosystems with embedded trapdoors. BYTE Russia, 2007 (№101), p.45-51.

[29] Young A., Yung M. (1997) Kleptography: Using Cryptography Against Cryptography. In: Fumy W. (eds) Advances in Cryptology — EUROCRYPT '97. EUROCRYPT 1997. Lecture Notes in Computer Science, vol 1233, p. 62-74. Springer, Berlin, Heidelberg. DOI: 10.1007/3-540-69053-0_6.

[30] Young A., Yung M. Malicious Cryptography. Exposing Cryptovirology. Wiley Publishing, Inc. 2004.

[31] Young A., Yung M. (2006) A Space Efficient Backdoor in RSA and Its Applications. In: Preneel B., Tavares S. (eds) Selected Areas in Cryptography. SAC 2005. Lecture Notes in Computer Science, vol 3897, p .128-143. Springer, Berlin, Heidelberg. DOI: 10.1007/11693383_9.

[32] Young A., Yung M. (2016) Cryptography as an Attack Technology: Proving the RSA/Factoring Kleptographic Attack. In: Ryan P., Naccache D., Quisquater JJ. (eds) The New Codebreakers. Lecture Notes in Computer Science, vol 9100, p. 243-255. Springer, Berlin, Heidelberg. DOI: 978-3-662-49301-4_16.

[33] Bellare M., Paterson K.G., Rogaway P. (2014) Security of Symmetric Encryption against Mass Surveillance. In: Garay J.A., Gennaro R. (eds) Advances in Cryptology – CRYPTO 2014. Lecture Notes in Computer Science, vol 8616, p. 1-19. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-662-44371-2_1.

[34] Russell A., Tang Q., Yung M., Zhou HS. (2016) Cliptography: Clipping the Power of Kleptographic Attacks. In: Cheon J., Takagi T. (eds) Advances in Cryptology – ASIACRYPT 2016. Lecture Notes in Computer Science, vol 10032, p. 34-64. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-662-53890-6_2.

[35] Hanzlik L., Kluczniak K., Kutyłowski M. (2017) Controlled Randomness – A Defense Against Backdoors in Cryptographic Devices. In: Phan RW., Yung M. (eds) Paradigms in Cryptology – Mycrypt 2016. Malicious and Exploratory Cryptology. Mycrypt 2016. Lecture Notes in Computer Science, vol 10311, p 215-232. Springer, Cham. DOI: 10.1007/978-3-319-61273-7_11.

[36] Degabriele J.P., Paterson K.G., Schuldt J.C.N., Woodage J. (2016) Backdoors in Pseudorandom Number Generators: Possibility and Impossibility Results. In: Robshaw M., Katz J. (eds) Advances in Cryptology – CRYPTO 2016. Lecture Notes in Computer Science, vol 9814, p. 403-432. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-662-53018-4_15.