STRIBOB / WHIRLBOB Security Analysis Addendum

Last update: Tuesday 30th June, 2015

Markku-Juhani O. Saarinen

Tampere University of Technology
mjos@iki.fi

Abstract. This memo collects references to published cryptanalytic results which are directly relevant to the security evaluation of CAESAR first round algorithm STRIBOB and its second round tweaked variant, WHIRLBOB. During the first year after initial publication of STRIBOB and WHIRLBOB, no cryptanalytic breaks or other serious issues have emerged. The main difference in the security between the two variants is that WHIRLBOB allows easier creation of constant-time software implementations resistant to cache timing attacks.

Keywords: CAESAR, STRIBOB, WHIRLBOB, Streebog, Whirlpool, Cryptanalysis.

1 Introduction

STRIBOB [38,39,40] is an algorithm for Authenticated Encryption with Associated Data (AEAD), and a first round candidates in the CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) [17] competition. WHIRLBOB [41] is a tweak of STRIBOB for CAESAR second round.

Security evaluation in the CAESAR competition is based on published analyses [16]. The main purpose of this document is to contain linked references and very short summaries of all relevant published results. The STRIBOB security parameters, claims, and goals remain unmodified from the original submission:

Parameter	Bits	Macro Definition
Secret key size.	192	`CRYPTO_KEYBYTES 24`
Secret sequence number.	0	`CRYPTO_NSECBYTES 0`
Public sequence number (nonce).	128	`CRYPTO_NPUBBYTES 16`
Authentication tag (message expansion).	128	`CRYPTO_ABYTES 16`

Category	Effort	Attack Goal
Confidentiality for the plaintext.	2^{191}	Recover plaintext from ciphertext.
Integrity for the plaintext.	2^{127}	Forge valid ciphertext.
Integrity for the associated data.	2^{127}	Forge Associated Data.
Integrity for the public message number.	2^{127}	Forge public message number.

Here we assume that the secret key is entirely unknown to the attacker. The complexities are given for $P = 0.5$ success probability.

Attacks against Sponge AEADs and structure of this paper. A sponge-based AEAD can be shown to be insecure in essentially two ways:

- By demonstrating a vulnerability in the padding mechanism. See Section 2.
- By demonstrating a structural distinguisher for the sponge permutation π.

There are two unkeyed π permutations, π^{STRI} (Section 3) and π^{WHIRL} (Section 4), for STRIBOB and WHIRLBOB, respectively. Both operate on 512-bit blocks and have 12 round iterations. Due to their structural similarty, much of the same analysis applies to both variants. Each round consists of 8×8 - byte MDS matrix operaton L in binary field $\mathbb{F}_{2^8}$; permutation P of 64 state bytes; 64 invocations of a 8×8-bit S-Box S; and XOR addition of a 512-bit round constant. The compound round operation $L \circ P \circ S$ is referred to as "LPS core" in this work. We conclude in Section 5.

2 Attacks on the Sponge AEAD Mode BLNK

The two variants, STRIBOB [38,39,40] and WHIRLBOB [41], use exactly the same padding mechanism, BLNK. This padding mechanism is a variant of Saarinen's Blinker [37] padding, but limited to the CAESAR use case.

Some independent analysis of the mode has been recently published [22,23,43], essentially validating the claimed security bounds. No attacks have been reported against the padding mechanism, and hence the security bounds derived from DUPLEXWRAP [10,12] still apply.

In the Duplex construction of SpongeWrap additional padding is included for each input block; a secondary information bit called *frame bit* is used for domain separation. Sakura [11] uses additional frame bits to facilitate tree hashing. It is essential that the various bits of information such as the key, authenticated data, and authenticated ciphertext can be exactly "decoded" from the Sponge input to avoid trivial padding collisions. BLNK uses simpler embedded encoding. However, for our "effective capacity" c [38] the following Theorem holds:

Theorem 1 (Theorem 4 from [12]). *The DuplexWrap and BLNK authenticated encryption modes satisfy the following privacy and authentication security bounds:*

$$\text{Adv}^{\mathsf{priv}}_{\mathsf{sbob}}(\mathcal{A}) < \frac{D+T}{2^k} + \frac{D^2+4DT}{2^{c+1}}$$
$$\text{Adv}^{\mathsf{auth}}_{\mathsf{sbob}}(\mathcal{A}) < \frac{D+T}{2^k} + \frac{D^2+4DT}{2^{c+1}} + \frac{D}{2^t}.$$

against any single adversary $\mathcal{A}$ if $K \xleftarrow{\$} \{0,1\}^k$, tags of t bits are used, π is a randomly chosen permutation, D is the data complexity (number of queries to target), and T is the offline attack time complexity.

Proof. See Theorem 4 of [13] and related work [6,10]. See also [22,23].

Since $b = 512$, we choose a Sponge rate of $r = 256$ bits, which leaves capacity $c = b - r = 256$. We choose key size $k = 192$ and limit $D < 2^{56}$ (2^{64} bits) and $T < 2^{k-1}$. As our actual effective capacity is $c \approx 254$ ($\delta \leq 2$ effective capacity bits are lost due to domain separation bits [38]), a 192-bit security level is reached.

3 Original STRIBOB Permutation π^{STRI}

The STRIBOBr1 (designated STRIBOBr2d1 for Round 2) sponge permutation π is derived from the Russian GOST R 34.11-2012 "Streebog" hash standard [20].

Streebog is not a sponge-based construction and uses the LPS core in an entirely different way, yet the similarities allow certain types of security reductions between the two algorithms.

We first recall the structure of GOST R 34.11-2012 hash function. Streebog produces either a 256-bit or a 512-bit hash from a bit string of arbitrary size using the Merkle-Damgård [19,33] iterative method without any randomization.

Figure 1 gives an overview of the hashing process. Padded message M is processed in 512-bit blocks $M = m_0 \mid m_1 \mid \cdots \mid m_n$ by a compression function $h' = g_N(h, m_i)$. The chaining variable h also has 512 bits and N denotes the index bit offset of the input block. After the last message block, there are finalization steps involving two invocations of the compression function, first on the total bit length of input, and then on checksum ϵ, which is computed over all input blocks $\bmod 2^{512}$.

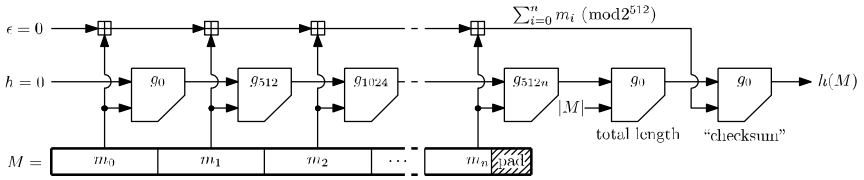

Fig. 1. Operation of Streebog with 512-bit output. For 256-bit hashes, the initial h value is changed to `0x010101..01` and the output $h(M)$ is truncated to 256 bits.

3.1 Streebog Compression Function $g_N(h, m)$

The compression function $h' = g_N(h, m)$ takes in a chaining variable h, message block m, a position index variable N, and produces a new chaining value h'. The compression function is built from a keyless 512-bit nonlinear permutation LPS and 512-bit vector XOR operations. The compression function has 12 rounds and a performs a total of 25 invocations of LPS:

$$
\begin{aligned}
[K_1, X_1] &= [\, \mathsf{LPS}(h \oplus N),\ m \,] \\
[K_{i+1}, X_{i+1}] &= [\, \mathsf{LPS}(K_i \oplus C_i), \mathsf{LPS}(X_i \oplus K_i) \,] \ \text{ for } \ 1 \le i \le 12 \\
g_N(h, m) &= K_{13} \oplus X_{13} \oplus h \oplus m.
\end{aligned}
$$

Figure 2 shows the structure of g. We can view it as a two-track substitution-permutation network where input value $h \oplus N$ and a set of 12 round constants C_i is used to key (via K_i) another substitution-permutation network operating on h . The outputs of the two tracks are finally XORed together with original values of h and m. We note that h together with offset N uniquely defines all K_i subkey values for each invocation of g.

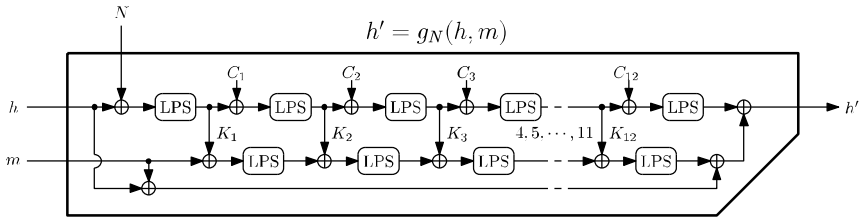

Fig. 2. Streebog compression function. All data paths, inputs, and outputs are 512-bit vectors. Here the $\oplus$ symbol denotes the XOR operation between two 512-bit vectors.

3.2 Security Relationship Between STRIBOB's π and Streebog's g

Only a single keyless permutation π is required in a in a sponge function. Structure of π is shown in Figure 3. We utilize the LPS transform and twelve round constants C_i of GOST R 34.11-2012 in our new design. For some vector of twelve 512-bit subkeys K_i we define a 512-bit permutation $\pi_K(X_1) = X_{13}$ with iteration

$$X_{i+1} = \mathsf{LPS}(X_i \oplus K_i) \text{ for } 1 \leq i \leq 12.$$

We assume that π_C and π_K are equally strong since both C and K consist of an essentially random set of subkeys. There is a straightforward intuitive security relation between π_K and a single instance of the full compression function g. We note that for the very first message block m, the subkeys K_i are always constant as they depend on the initial constant $h = 0$ alone. We can therefore write for first block:

$$h' = g_0(0, m) = \pi_K(m) \oplus m.$$

This indicates that a generic powerful attack against π is also an attack on g. A structural distinguishing attack against g of course does not imply a collision attack against Streebog as a whole.

After careful analysis, we conjecture that the π_C permutation offers *no structural distinguishers* that are not based on some trivial property such as a priori knowledge of output value of $\pi_C(x)$ for some particular x. This is a fundamental requirement for a Sponge based design. We use π_C alone in our final construction.

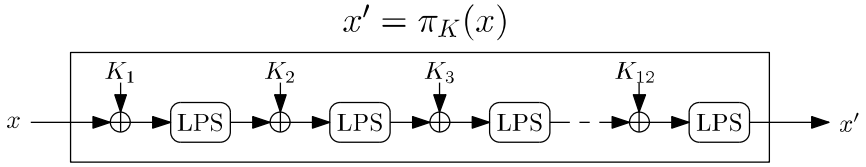

Fig. 3. Structure of the 512-bit permutations π used by STRIBOB and WHIRLBOB.

3.3 Security of LPS

LPS gets all of its non-linearity from the 8-bit S-box S, which has been designed to offer resistance against classical methods of cryptanalysis. Its differential bound [14] is $P = \frac{8}{256}$ and best linear approximation [30] holds with $P = \frac{28}{128}$. No algebraic weaknesses have been found (S-Box design was "randomized" for this purpose [41]).

The creation method for the round constants K was recently published [36] in response to a "malicious" variant of Streebog [5]. Some background on the creation of the L and S is given in [41]. We hope to be able to publish the full creation criteria for Sreebog's L and S too. The same S-Box S is also used by the new block cipher standard proposal [44].

The linear transform L is not randomly constructed even though it is expressed without explanation as a 64×64 binary matrix in [20]. L in fact has a byte-oriented structure as an MDS matrix with $\mathbb{F}_{2^8}$ arithmetic in a similar fashion as AES, even though this is not mentioned in the standard specification [24,34]. We use this equivalent description.

Many structural observations on AES-like ciphers also apply to LPS [8]: S and L are effectively mix together the bits of the eight 64-bit rows. P swaps rows and columns and after two rounds each input bit affects each output bit of the 512-bit state. Adjusted to its state size, LPS has similar per-round avalanche to AES (each input byte affects each output byte after two rounds) and similar resistance to Square attacks. The best theoretical Square attack is effective against six rounds [26].

Recently, pre-image attacks against Streebog were considered in [4], distinguishers in [3,29], and rebound attacks in [2]. Each one of these attacks is only effective against reduced variants of the Streebog compression function.

4 WHIRLBOB Permutation π^{WHIRL}

The π_{WHIRL} permutation is lifted from WhirlPool 3.0 hash function [7] in straightforward fashion, and an analogous security relationship (to STRIBOB / Streebog) exists between the WhirlPool and WHIRLBOB [41]. WhirlPool is an ISO standard [21].

If we use the AES-style notation of Whirlpool, S is equivalent to `SubBytes`, P corresponds to `ShiftColumns`, L to `MixRows`, followed by `AddRoundKey`. Vast AES research literature is also largely applicable to both WhirlPool and WHIRLBOB [8,18].

Structural similarity and equivalent differential and linear bounds indicate that the security levels of STRIBOB and WHIRLBOB are essentially equivalent. However, the WhirlPool structure has received even more (15 years) of cryptanalysis [25,42] than StreeBog, and allows constant-time and light-weight hardware and software implementation [15,41]. The structure is therefore easier to secure against cache timing attacks [1,9,35,45].

The only effective attack against 10-round variant is the Rebound Attack [27,28,32]. We firmly believe that increase of rounds from 10 to 12 makes WHIRLBOB resistant to these attacks. Even addition of a single round would increase the work factor of those attacks by a significant factor. Since we are using a sponge mode with $r = 256$ $c = 256$, an attacker only has control over half of the state, and therefore STRIBOB and WHIRLBOB designs have a good security margin against these attacks [31].

5 Conclusions

STRIBOB and WHIRLBOB are built from conservative, very well understood cryptographic elements. Based on our review of latest research, no attacks are known against full versions of these ciphers, and a comfortable security margin remains.

Furthermore, especially WHIRLBOB allows constant-time implementation on many platforms, making it resistant to side-channel attacks that make secure software implementation of AES difficult.

Extensive, directly applicable research into the security of DuplexWrap, WhirlPool, Streebog, and AES-like structures warrants an exceptional level of confidence for the long-term security of both STRIBOB and WHIRLBOB.

References

1. Onur Aciiçmez, Werner Schindler, and Çetin Kaya Koç. Cache based remote timing attack on the AES. In Masayuki Abe, editor, *CT-RSA 2007*, volume 4377 of *LNCS*, pages 271–286. Springer, 2007. doi:10.1007/11967668_18.
2. Riham AlTawy, Aleksandar Kircanskim, and Amr M. Youssef. Rebound attacks on stribog. In Dong-Guk Han Hyang-Sook Lee, editor, *ICISC 2013*, volume 8565 of *LNCS*, pages 175–188. Springer, 2014. doi:10.1007/978-3-319-12160-4_11.
3. Riham AlTawy and Amr M. Youssef. Integral distinguishers for reduced-round stribog. IACR ePrint 2013/684, 2013. URL: https://eprint.iacr.org/2013/648.
4. Riham AlTawy and Amr M. Youssef. Preimage attacks on reduced-round stribog. In David Pointcheval and Damien Vergnaud, editors, *AFRICACRYPT 2014*, volume 8469 of *LNCS*, pages 109–125. Springer, 2014. doi:10.1007/978-3-319-06734-6_7.
5. Riham AlTawy and Amr M. Youssef. Watch your constants: Malicious streebog. IACR ePrint 2014/879, 2014. URL: https://eprint.iacr.org/2014/879.
6. Elena Andreeva, Bart Mennink, and Bart Preneel. Security reductions of the second round SHA-3 candidates. IACR ePrint 2010/381, July 2010. URL: https://eprint.iacr.org/2010/381.
7. Paulo S. L. M. Barreto and Vincent Rijmen. The Whirlpool hashing function. NESSIE Algorithm Specification, 2000, Revised May 2003. URL: http://www.larc.usp.br/~pbarreto/WhirlpoolPage.html.
8. Christof Beierle, Philipp Jovanovic, Martin M. Lauridsen, Gregor Leander, and Christian Rechberger. Analyzing permutations for AES-like ciphers: Understanding ShiftRows. In Kaisa Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*, pages 37–58. Springer, 2015. doi:10.1007/978-3-319-16715-2_3.
9. D. J. Bernstein. Cache-timing attacks on AES. Technical report, University of Chigaco, 2005.
10. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the sponge: Single-pass authenticated encryption and other applications. In A. Miri and S. Vaudenay, editors, *SAC 2011*, volume 7118 of *LNCS*, pages 320–337. Springer, 2011. doi:10.1007/978-3-642-28496-0_19.
11. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sakura: a flexible coding for tree hashing. IACR ePrint 2013/231, April 2013. URL: https://eprint.iacr.org/2013/231.
12. Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. CAESAR submission: Keyak v1. CAESAR First Round Submission, March 2014. URL: http://competitions.cr.yp.to/round1/keyakv1.pdf.

13. Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. CAESAR submission: Keyak v1. CAESAR First Round Submission, March 2014. URL: http://competitions.cr.yp.to/round1/keyakv1.pdf.
14. Eli Biham and Adi Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer, 1993. doi:10.1007/978-1-4613-9314-6.
15. Billy B. Brumley. Secure and fast implementations of two involution ciphers. In T. Aura, K. Järvinen, and K. Nyberg, editors, *NordSec '10*, volume 7127 of *LNCS*, pages 269–282. Springer, 2012. doi:10.1007/978-3-642-27937-9_19.
16. CAESAR. CAESAR call for submissions, January 2014. URL: http://competitions.cr.yp.to/caesar-call.html.
17. CAESAR. CAESAR: Competition for authenticated encryption: Security, applicability, and robustness, January 2014. URL: http://competitions.cr.yp.to/caesar.html.
18. Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - the Advanced Encryption Standard*. Springer, 2002. doi:dx.doi.org/10.1007/978-3-662-04722-4.
19. Ivan Damgård. A design principle for hash functions. In Gilles Brassard, editor, *CRYPTO '89*, volume 435 of *LNCS*, pages 416–427. Springer, 1989. doi:10.1007/0-387-34805-0_39.
20. GOST. Information technology. cryptographic protection of information, hash function. GOST R 34.11-2012, 2012. (In Russian). URL: http://protect.gost.ru/v.aspx?control=7&id=180209.
21. ISO/IEC. Information technology – security techniques – hash-functions – part 3: Dedicated hash-functions. ISO/IEC 10118-3:2004, 2004.
22. Philipp Jovanovic, Atul Luykx, and Bart Mennink. Beyond $2^{c/2}$ security in sponge-based authenticated encryption modes. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014*, volume 8873 of *LNCS*, pages 85–104. Springer, 2002. doi:10.1007/978-3-662-45611-8_5.
23. Philipp Jovanovic, Atul Luykx, and Bart Mennink. Beyond $2^{c/2}$ security in sponge-based authenticated encryption modes. IACR ePrint 2014/373, May 2014. URL: https://eprint.iacr.org/2014/373.
24. Oleksandr Kazymyrov and Valentyna Kazymyrova. Algebraic aspects of the Russian hash standard GOST R 34.11-2012. In *CTCrypt '13, June 23-24, 2013, Ekaterinburg, Russia*, 2013. IACR ePrint 2013/556. URL: https://eprint.iacr.org/2013/556.
25. Lars R. Knudsen. Non-random properties of reduced-round Whirlpool. NESSIE public report, NES/DOC/UIB/WP5/016/2, August 2002. URL: https://www.cosic.esat.kuleuven.be/nessie/reports/phase2/uibwp5-016-2.pdf.
26. Lars R. Knudsen and David Wagner. Integral cryptanalysis (extended abstract). In Joan Daemen and Vincent Rijmen, editors, *FSE 2002*, volume 2365 of *LNCS*, pages 112–127. Springer, 2002. doi:10.1007/3-540-45661-9_9.
27. Mario Lamberger, Florian Mendel, Christian Rechberger, Vincent Rijmen, and Martin Schläffer. Rebound distinguishers: Results on the full whirlpool compression function. In Mitsuru Matsui, editor, *ASIACRYPT '09*, volume 5912 of *LNCS*, pages 126–143. Springer, 2009. doi:10.1007/978-3-642-10366-7_8.
28. Mario Lamberger, Florian Mendel, Martin Schläffer, Christian Rechberger, and Vincent Rijmen. The rebound attack and subspace distinguishers: Application to Whirlpool. *J. Cryptology*, 28:257–296, 2015. doi:10.1007/s00145-013-9166-5.
29. Bingke Ma, Bao Li, Ronglin Hao, and Xiaoqian Li. Improved cryptanalysis on reduced-round GOST and Whirlpool hash function. In Ioana Boureanu, Philippe Owesarski, and Serge Vaudenay, editors, *ACNS 2014*, volume 8479 of *LNCS*, pages 289–307. Springer, 2014. doi:10.1007/978-3-319-07536-5_18.

30. Mitsuru Matsui. Linear cryptoanalysis method for DES cipher. In T. Helleseth, editor, *EUROCRYPT '93*, volume 765 of *LNCS*, pages 386–397. Springer, 1994. doi:10.1007/3-540-48285-7_33.
31. Florian Mendel. Personal Communication, June 2014.
32. Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. The rebound attack: Cryptanalysis of reduced Whirlpool and Grøstl. In Orr Dunkelman, editor, *FSE 2009*, volume 5665 of *LNCS*, pages 260–276. Springer, 2009. doi:10.1007/978-3-642-03317-9_16.
33. Ralph C. Merkle. *Secrecy, Authenticatication, and public key systems*. PhD thesis, Stanford University, 1979.
34. NIST. Advanced Encryption Standard (AES). Federal Information Processing Standards Publication FIPS 197, November 2001. URL: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.
35. Dag Arne Osvik, Adi Shamir, and Eran Tromer. Cache attacks and countermeasures: The case of AES. In David Pointcheval, editor, *CT-RSA 2006*, volume 3860 of *LNCS*, pages 1–20. Springer, 2006. doi:10.1007/11605805_1.
36. Vladimir Rodskoy. Note on Streebog constants origin. Preprint, 2015. URL: http://tk26.ru/en/ISO_IEC/streebog/streebog_constants_eng.pdf.
37. Markku-Juhani O. Saarinen. Beyond modes: Building a secure record protocol from a cryptographic sponge permutation. In J. Benaloh, editor, *CT-RSA 2014*, volume 8366 of *LNCS*, pages 270–285. Springer, 2014. doi:10.1007/978-3-319-04852-9_14.
38. Markku-Juhani O. Saarinen. StriBob: Authenticated encryption from GOST R 34.11-2012 LPS permutation. In *CTCrypt '14, 05-06 June 2014, Moscow, Russia. Preproceedings.*, pages 170–182, June 2014. URL: https://eprint.iacr.org/2014/271.
39. Markku-Juhani O. Saarinen. The STRIBOBr1 authenticated encryption algorithm. CAESAR, 1st Round Candidate, March 2014. URL: http://www.stribob.com.
40. Markku-Juhani O. Saarinen. StriBob: Authenticated encryption from GOST R 34.11-2012 LPS permutation. *Mathematical Aspects of Cryptography*, 6(2):67–78, 2015. (Abstract In Russian). URL: http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=mvk&paperid=146&option_lang=eng.
41. Markku-Juhani O. Saarinen and Billy B. Brumley. Lighter, Faster, and Constant-Time: WHIRLBOB, the Whirlpool variant of STRIBOB. IACR ePrint 2014/501, June 2014. URL: http://eprint.iacr.org/2014/501.
42. Yu Sasaki, Lei Wang, Shuang Wu, and Wenling Wu. Investigating fundamental security requirements on Whirlpool: Improved preimage and collision attacks. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 562–379. Springer, 2012. doi:10.1007/978-3-642-34961-4_34.
43. Yu Sasaki and Kan Yasuda. How to incorporate associated data in sponge-based authenticated encryption. In Kaisa Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*, pages 353–370. Springer, 2015. doi:10.1007/978-3-319-16715-2_19.
44. Vasily Shishkin, Denis Dygin, Ivan Lavrikov, Grigory Marshalko, Vladimir Rudskoy, and Dmitry Trifonov. Low-weight and hi-end: Draft Russian Encryption Standard. In *CTCrypt '14, 05-06 June 2014, Moscow, Russia. Preproceedings.*, pages 183–188, June 2014.
45. Michael Weiß, Benedikt Heinz, and Frederic Stumpf. A cache timing attack on AES in virtualization environments. In Angelos D. Keromytis, editor, *FC 2012*, volume 7397 of *LNCS*, pages 314–328. Springer, 2013. doi:10.1007/978-3-642-32946-3_23.

Document version 20150630112200.