L. R. Ahmetzyanova, G. A. Karpunin, G. K. Sedov, Near birthday attack on “8 bits” AEAD mode,
Mat. Vopr. Kriptogr., 2019, Volume 10, Issue 2, 47–60

https://www.mathnet.ru/eng/mvk283

МАТЕМАТИЧЕСКИЕ ВОПРОСЫ КРИПТОГРАФИИ
2019 Т. 10 № 2 С. 47–60

УДК 519.719.2 DOI https://doi.org/10.4213/mvk283

Near birthday attack on “8 bits” AEAD mode

L. R. Ahmetzyanova, G. A. Karpunin, G. K. Sedov

Crypto-Pro LLC, Moscow, Russia

Получено 06.II.2018

Abstract. We describe an attack on the “8 bits” authentication encryption with associated data (AEAD) mode proposed during the AEAD standardization process. The “8 bits” mode is similar to the CCM mode except for several design features. We show that these distinctive features allow to construct a near birthday attack on “8 bits” mode. We also propose countermeasures to resist suggested attack.

Key words: “8 bits” mode, birthday attack, AEAD forgery

Атака методом «дней рождения» на алгоритм аутентифицированного шифрования «8 бит»

Л. Р. Ахметзянова, Г. А. Карпунин, Г. К. Седов

ООО «КРИПТО-ПРО», Москва, Россия

Аннотация. Представлена атака на режим аутентифицированного шифрования (AEAD-режим) «8 бит», который являлся одним из кандидатов на роль стандартизованного российского AEAD-режима. Режим «8 бит» отличается от режима CCM несколькими конструктивными особенностями. Показано, что эти особенности позволяют построить атаку на режим «8 бит» с трудоемкостью, близкой к трудоемкости атаки на основе парадокса дней рождения. Предложены способы противодействия построенному методу.

Ключевые слова: AEAD-режим «8 бит», атака на основе парадокса дней рождения, подделка имитовставки

Citation: *Mathematical Aspects of Cryptography*, 2019, v. 10, № 2, pp. 47–60 (Russian)

1. Introduction

Authenticated encryption schemes, which aim at providing both confidentiality and integrity of data, have gained renewed attention in the light of the recently commenced CAESAR competition [2].

The AEAD modes are the most widely spreaded subset of authenticated encryption schemes which allow to process associated data that should be authenticated but not encrypted.

The importance of the AEAD schemes development is explained by their exploitation simplicity: it is much easier to implement them properly than MAC and encryption schemes separately under random and independent keys.

Also when using the AEAD scheme we can reduce the key size, state size, and improve the data processing speed. Another advantage of such schemes is their transparent embedding into high-level schemes and protocols because there is no need of using additional diversifications for enough key material generation. For example, the use of such schemes is supposed to be mandatory for the Record protocol in TLS 1.3 [6].

The AEAD scheme called “8 bits” was proposed during the AEAD standardization process of the Russian Technical Committee for Standardization TC 26 and was presented at the seminar “Mathematical methods of cryptanalysis” at MSU.

This scheme is based on the standardized blockcipher modes of operation CTR and OMAC. The prototype of “8 bits” is the CCM mode [9] which is standardized in IEEE 802.11i [3]. The crucial difference between CCM and “8 bits” is the absence of additional tag encryption. This distinctive feature creates the possibility of a near birthday attack.

Although the birthday bound does not mean the total insecurity of the mode, there are some AEAD modes with no applicable attacks on the authentication with near birthday complexity [1, 9, 7]. Thus we claim that the presence of near birthday complexity attack should be considered as a flaw in the construction of AEAD scheme.

The current paper is organized as follows. In Section 2 we provide basic definitions and remind the reader of some notions, in Section 3 the definition of “8 bits” is provided, and in Section 4 we describe the above-mentioned attack.

At the end of the introduction we note that another attack on “8 bits” mode was independently developed by Sergey Aksenov.

2. Preliminaries and basic definitions

By V_n we denote the set of n-component bit strings. Also we consider V_n as a vector space over field $\mathbb{F}_2 = \{0, 1\}$. Let V^* be the set of all bit strings of finite length. For nonnegative integers l and $i < 2^l$ let $\mathrm{str}_l(i)$ be a l-bit representation of i with the least significant bit on the right. For a nonnegative integer l and a bit string $M \in V_l$ let $\mathrm{int}(M)$ be an integer i such that $\mathrm{str}_l(i) = M$.

For a bit string M and a positive integer $l \leqslant |M|$ let $\mathrm{msb}_l(M)$ ($\mathrm{lsb}_l(M)$) be the string consisting of the leftmost (rightmost) l bits of M.

For bit strings M and N by $M \| N$ we denote their concatenation. For the bit string M by M^n we denote the string M concatenated n times. Let $|M|$ be the bit length of the string M.

For any set S, define $Perm(S)$ as the set of all bijective mappings from S to S (permutations on S).

A block cipher is a mapping

$$E\colon V_k \times V_n \to V_n$$

such that for all $K \in V_k$ the mapping $E(K, \cdot)$ is a permutation on V_n. By n and k we denote the block size and the key size respectively. By $E_K(\cdot)$ we denote the mapping $E(K, \cdot)$.

We model an adversary $\mathcal{A}$ by means of an interactive probabilistic algorithm that has access to one or more oracles. The resources of $\mathcal{A}$ are defined by time and query complexities. For a fixed model of computation and a method of encoding the time complexity includes the description size of $\mathcal{A}$. The query complexity usually includes the number of queries and the maximal length of queries or the total legth of queries.

3. “8 bits” AEAD Mode

The “8 bits” mode is defined for the block size $n = 128$ bits. The additional parameter of the mode is a tag size $s \leqslant n$. This parameter should be fixed and known both by the sender and the receiver.

Both the sender and the receiver know the secret key K used for computing a tag and a ciphertext. By $P \in V^*$ and $A \in V^*$ we denote a plaintext and the associated data respectively. The length of both P and A must be less than 2^{64} bits. P also must have non-zero length. The pair of a plaintext and the associated data we will call “a message”. This mode uses also an initialization vector $IV \in V_{56}$. This vector must be unique for each new message.

3.1. OMAC Algorithm

The integrity and authenticity in "8 bits" is achieved using the OMAC (one-key message authentication code) algorithm. Let us remind the reader the main idea of the OMAC computation. The detailed description of OMAC may be found in [4].

The algorithm starts with the construction of MAC keys as follows:

$$\begin{aligned} R &= E_K(0^{128}), \\ K_1 = K_1(R) &= \begin{cases} R \ll 1, & \text{if} \quad \mathrm{msb}_1(R) = 0, \\ (R \ll 1) \oplus B_{128} & \text{otherwise}, \end{cases} \\ K_2 = K_2(R) &= \begin{cases} K_1 \ll 1, & \text{if} \quad \mathrm{msb}_1(R) = 0, \\ (K_1 \ll 1) \oplus B_{128} & \text{otherwise}, \end{cases} \end{aligned} \tag{1}$$

where $\ll$ is a left shift, $\oplus$ is a bit-wise XOR, $B_{128} = 0^{120}||10000111$.

Then the message $M \in V^*$ is divided into $t = \lceil |M|/n \rceil$ blocks $M_1, \ldots, M_{t-1} \in V_n$ and $M_t \in V_r$, $r \leqslant n$, that is

$$M = M_1 \| \ldots \| M_{t-1} \| M_t.$$

The OMAC algorithm is described by Pseudocode 1.

$OMAC^{(s)}(M = M_1 \| \ldots \| M_t, K)$

1: $C_0 = 0^n$

2: $C_i = E_K(C_{i-1} \oplus M_i), i = 1, \ldots, t-1$

3: **if** $|M_t| = n$ **then**

4: $\quad K^* = K_1, \;\; M^* = M_t$

5: **else**

6: $\quad K^* = K_2, \;\; M^* = M_t \| 1 \| 0^{n-|M_t|-1}$

7: $T = \mathrm{msb}_s(E_K(C_{t-1} \oplus M^* \oplus K^*))$

8: **return** T

Pseudocode 1: The OMAC algorithm

The mode structure is illustrated by Fig. In the "8 bits" description the computation of the tag of size s under the key K according to the OMAC algorithm is denoted by $\mathsf{MAC}_K^{(s)}(M)$.

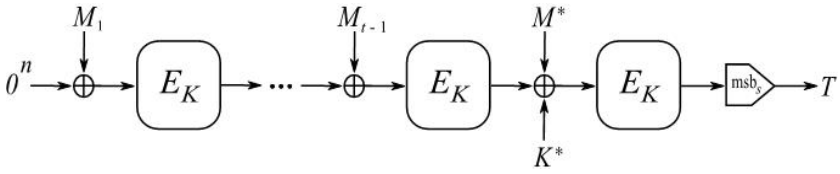

Fig. The OMAC algorithm which takes the message $M = M_1 \| M_2 \| \ldots \| M_t \in V^*$, $t = \lceil |M|/n \rceil$, and the key $K \in V_k$ as inputs and outputs the tag $T \in V_s$.

3.2. The “8 bits” authenticated encryption and decryption algorithms

Let t be the length of a plaintext P in blocks, i. e.

$$t = \left\lceil \frac{|P|}{128} \right\rceil,$$

$s \leqslant n$ be the tag size and

$$d = \left\lceil \frac{|P| + |A| + 72}{128} \right\rceil.$$

The “8 bits” authenticated encryption and decryption algorithms are described by Pseudocode 2 and Pseudocode 3 correspondingly. The probabilistic key generation algorithm $8\text{bits}.\mathsf{K}()$ outputs the key $K \xleftarrow{\mathcal{U}} V_k$.

$8\text{bits}^{(s)}.\mathcal{E}(K, IV, P, A)$

1: $S_i = 0^8 \| IV \| \text{str}_{64}(i), i = 1, \ldots, t$
2: $\Gamma = E_K(S_1) \| E_K(S_2) \| \ldots \| E_K(S_t)$
3: $C = P \oplus \text{msb}_{|P|}(\Gamma)$
4: **if** $|A| = 0$ **then**
5: $\quad F = 1^7 \| 0$
6: **else**
7: $\quad F = 1^7 \| 1$
8: $B = F \| \text{str}_8(s) \| IV \| A \| C \| 0^{128d - |C| - |A| - 72} \| \text{str}_{64}(|A|) \| \text{str}_{64}(|C|)$
9: $T = \mathsf{MAC}_K^{(s)}(B)$
10: **return** $C \| T$

Pseudocode 2: Authenticated Encryption Algorithm of the “8 bits” mode

$\underline{8\text{bits}^{(s)}.\mathcal{D}(K, IV, C \| T, A, s)}$

if $|A| = 0$ **then**
    $F = 1^7 \| 0$
else
    $F = 1^7 \| 1$
$B = F \| \text{str}_8(s) \| IV \| A \| C \| 0^{128d-|C|-|A|-72} \| \text{str}_{64}(|A|) \| \text{str}_{64}(|C|)$
$T' = \mathsf{MAC}_K^{(s)}(B)$
if $T \neq T'$ **then**
    return $\perp$
$S_i = 0^8 \| IV \| \text{str}_{64}(i), i = 1, \ldots, t$
$\Gamma = E_K(S_1) \| E_K(S_2) \| \ldots \| E_K(S_t)$
$P = C \oplus \text{msb}_{|C|}(\Gamma)$
return P

Pseudocode 3: Authenticated Decryption Algorithm of the “8 bits” mode

4. Attack

4.1. Adversary model

The standard model relevant for analyzing the AEAD security is the IND-CCA3 model [8] which allows to investigate the security of proposed scheme from the view point of both integrity and privacy. The paper [8] states that the scheme is IND-CCA3-secure iff it is IND-CPA-(privacy) and Auth-secure (integrity). In the current paper we consider the Auth model in detail and describe the attack in the framework of this model.

Definition. The advantage of the adversary $\mathcal{A}$ in the Auth model for the AEAD mode is defined as follows:

$$\mathsf{Adv}^{\text{Auth}}_{\mathsf{AEAD}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\text{Auth}}_{\mathsf{AEAD}}(\mathcal{A}) = 1\right],$$

where $\mathbf{Exp}^{\text{Auth}}_{\mathsf{AEAD}}(\mathcal{A})$ is described in the following way:

$\underline{\mathbf{Exp}^{\text{Auth}}_{\mathsf{AEAD}}(\mathcal{A})}$

$K \xleftarrow{\$} \mathsf{AEAD.K}()$
$sent \Leftarrow \varnothing$
$(IV', C' \| T', A') \Leftarrow \mathcal{A}^{\mathsf{Encrypt}}$
$P \Leftarrow \mathsf{AEAD}.\mathcal{D}(K, IV', C' \| T', A')$ **if** $(C' \| T', A') \notin sent$ **and** $P \neq \perp$ **then**
$win \Leftarrow 1$ **else**
$win \Leftarrow 0$ **end if**
return win

$\underline{\text{Oracle Encrypt}}$

$C \| T \xleftarrow{\$} \mathsf{AEAD}.\mathcal{E}(K, IV, P, A)$
$sent \Leftarrow sent \cup \{(A, C \| T)\}$
return $C \| T$

The Auth model allows the adversary to choose adaptively messages for encryption and receive their ciphertexts and tags. The adversary’s goal is to make the receiver to accept a “non-authentic” pair of ciphertext C and the associated data A. In the Auth model the “non-authentic” message means it was never transmitted by the sender (satisfied the condition $\notin sent$).

Now consider the “8bits” scheme with $s = 128$. Suppose that IV is generated with the use of a counter, i. e. for each new message the initialization vector IV' takes value $\mathrm{str}_{56}(\mathrm{int}_{56}(IV) + 1)$ if the previous initialization vector was IV. Let the first initialization vector be 0^{56}.

4.2. Attack details

The main idea behind the proposed attack is to exploit the information received from obtained ciphertexts to “break” integrity. In the model relevant for schemes providing integrity the adversary can observe only tags, while in the AEAD schemes case the adversary can get additional information from received ciphertexts. The proposed attack uses the possibility of getting enciphered counter values to find collision with the OMAC values. It results in the possibility of modifying undetectively the associated data length stored in the last block of the OMAC-processed string.

4.2.1. The first stage

Let us introduce a parameter l such that $6 \leqslant l < 55$. Then we make 2^l queries $(IV_1, P_1), \ldots, (IV_{2^l}, P_{2^l})$, $|P_i| = 2^{64} - 128$, with empty associated data to the Encrypt oracle. The Encrypt oracle returns the corresponding ciphertexts $C_1, C_2, \ldots, C_{2^l}$. Note that for the message P_i the corresponding initialization vectors are $IV_i = \mathrm{str}_{64}(i-1)$. By $\mathcal{IV}' = \{IV_i |\ i = 1, \ldots, 2^l\}$ we denote the set of all initialization vectors for the messages $P_1, P_2, \ldots, P_{2^l}$. Note that the messages length is the greatest possible length multiple of the block size. The number of 128-bit blocks in any of these messages is equal to $2^{57} - 1$.

Let $P_i[j]$ be the j-th 128-bit block of the message P_i and $S_i[j]$ be the string $0^8 \| IV \| \mathrm{str}_{64}(j)$ which is used for ciphering the block $P_i[j]$:

$$C_i[j] = P_i[j] \oplus E_K(S_i[j]).$$

Denote by

$$\mathcal{S} = \{S_i[j] |\ i = 1, \ldots, 2^l,\ j = 1, \ldots, 2^{57} - 1\}$$

the set of all strings $S_i[j]$.

Given the plaintext blocks $P_i[j]$ and the ciphertext blocks $C_i[j]$ at the end of this stage we get the keystream blocks

$$\Gamma_i[j] = E_K(S_i[j]) \quad \text{for all strings}$$

$\mathrm{S}_i[j]$ from $\mathcal{S}$.

4.2.2. The second stage

At the second stage we compute $2^{56} - 2^l$ values of ciphertexts and OMAC tags for all remained values of initialization vectors $\mathcal{IV}'' = V_{56} \setminus \mathcal{IV}'$. More accurately, we make $2^{56} - 2^l$ queries (IV, P, A) to the $\mathsf{Encrypt}$ oracle, where

$$IV \in \mathcal{IV}'', \quad P = 0^1 \in V_1, \quad A = 0^1 \in V_1.$$

For query (IV, P, A) the $\mathsf{Encrypt}$ oracle returns a pair $C \| T$, where C is the ciphertext and T is a tag. The tag T is computed by the $\mathsf{MAC}_K^{(128)}$ algorithm with an input B formatted as follows

$$B = \underbrace{F \| \mathrm{str}_8(128) \| IV \| A \| C \| 0^{54}}_{B_0} \| \underbrace{\mathrm{str}_{64}(1) \| \mathrm{str}_{64}(1)}_{B_1}.$$

Since B consists of only two blocks, the tag T equals $E_K(E_K(B_0) \oplus K_1 \oplus B_1)$.

Note that for any new IV the string B_0 is new and B_1 is constant and equal to $\mathrm{str}_{64}(1) \| \mathrm{str}_{64}(1)$. By

$$\mathcal{B} = \{(F \| \mathrm{str}_8(128) \| IV \| A \| C \| 0^{54}) |\ IV \in \mathcal{IV}''\}$$

we denote the set of all such blocks B_0.

Thus at the end of this stage we have the OMAC tags

$$E_K(E_K(B_0) \oplus K_1 \oplus B_1)$$

for all $2^{56} - 2^l$ blocks $B_0 \in \mathcal{B}$.

4.2.3. The third stage

In this subsection we estimate the probability p of getting collision between OMAC tags from the second stage and the one of keystream blocks $\Gamma_i[j]$ from the first stage.

More formally, we estimate the probability

$$p = \Pr_K \left[\{E_K(S)_{S\in\mathcal{S}}\} \cap \{E_K(E_K(B_0) \oplus K_1 \oplus B_1)\}_{B_0\in\mathcal{B}} \neq \varnothing\right]$$

under the following conditions :

$$\mathcal{S} \cap \mathcal{B} = \varnothing, \quad 0^{128} \notin \mathcal{S}, \quad 0^{128} \notin \mathcal{B},$$
$$|\mathcal{S}| = 2^l\,(2^{57} - 1), \quad |\mathcal{B}| = 2^{56} - 2^l,$$

the key $K_1 = K_1(E_K(0^{128}))$ is a function which depends only on the value $E_K(0^{128})$.

We estimate this probability in the ideal cipher model, where E_K is supposed to be a random permutation on V_{128}.

Then, by the technical Lemma from the Appendix section, we obtain

$$p \geqslant 1 - \exp\left\{\frac{(2^{56} - 2^l)\,(2^l\,(2^{57} - 1) - 1)}{2^{128}}\right\} =$$
$$= 1 - \exp\left\{-2^{l-15}\left(1 - \frac{1}{2^{56-l}}\right)\left(1 - \frac{1}{2^{57}} - \frac{1}{2^{57+l}}\right)\right\}.$$

The right-hand side increases monotonically on $6 \leqslant l < 55$, and if $l = 15$ then we have $p > 0.63 \approx 1 - e^{-1}$.

Therefore if we encrypt 2^{15} messages on the first stage then one of OMAC values collides with one of the keystream blocks with the probability $p > 0.63$.

4.2.4. Forging tag

Suppose that we have collision and

$$\mathsf{MAC}_K^{(128)}(B) = \Gamma_i[j] = E_K(0^8 \,\|\, IV_i \,\|\, \mathrm{str}_{64}(j)),$$

where

$$B = \underbrace{F \,\|\, \mathrm{str}_8(128) \,\|\, IV \,\|\, A \,\|\, C \,\|\, 0^{54}}_{B_0} \,\|\, \underbrace{\mathrm{str}_{64}(1) \,\|\, \mathrm{str}_{64}(1)}_{B_1}.$$

Consider pairs (C', A') of the ciphertexts $C' = 0^1 \in V_1$ and the associated data $A' = 0 \| C \| 0^u$ with $u = 0, \ldots, 53$. Note that the OMAC input for such pairs is equal to

$$B' = \underbrace{F \| \mathrm{str}_8(128) \| IV \| A' \| C' \| 0^{55-(u+2)}}_{B'_0} \| \underbrace{\mathrm{str}_{64}(u+2) \| \mathrm{str}_{64}(1)}_{B'_1}.$$

Note that $B'_0 = B_0$ and thus OMAC value $\mathsf{MAC}_K^{(128)}(B')$ is equal to

$$E_K(E_K(B'_0) \oplus K_1 \oplus B'_1) = E_K(E_K(B_0) \oplus K_1 \oplus B'_1).$$

Let us consider the set of strings

$$\widehat{\mathcal{B}} = \left\{ \widehat{B} \in V_{128} \,|\, \widehat{B} = \mathrm{str}_{64}(r) \| \mathrm{str}_{64}(1), r = 2, \ldots, 55 \right\}.$$

This set describes all possible values of B'_1 for pairs (C', A'). Note that for all $\widehat{B} \in \widehat{\mathcal{B}}$

$$\begin{aligned} E_K(B_0) \oplus K_1 \oplus \widehat{B} &= E_K(B_0) \oplus K_1 \oplus B_1 \oplus (B_1 \oplus \widehat{B}) = \\ &= 0^8 \| IV_i \| \mathrm{str}_{64}(j) \oplus \\ &\quad \oplus ((\mathrm{str}_{64}(1) \| \mathrm{str}_{64}(1)) \oplus (\mathrm{str}_{64}(r) \| \mathrm{str}_{64}(1))) = \\ &= 0^8 \| IV_i \| \mathrm{str}_{64}(j) \oplus ((\mathrm{str}_{64}(1) \oplus \mathrm{str}_{64}(r)) \| \mathrm{str}_{64}(0)) = \\ &= 0^8 \| IV_t \| \mathrm{str}_{64}(j), \end{aligned}$$

where IV_t can differ from IV_i only in the 6 least significant bits. Hence

$$0^8 \| IV_t \| \mathrm{str}_{64}(j) \in \mathcal{S},$$

and we know the corresponding ciphertext $E_K(0^8 \| IV_t \| \mathrm{str}_{64}(j))$ from the first stage.

Therefore we can forge the tag value for the pair (C', A') that corresponds to $\widehat{B}$ as follows

$$E_K(E_K(B_0) \oplus K_1 \oplus \widehat{B}) = E_K(0^8 \| IV_t \| \mathrm{str}_{64}(i)),$$

where $\widehat{B} = \mathrm{str}_{64}(r) \| \mathrm{str}_{64}(1)$. Therefore we get 54 forged tags with the probability $p > 0.63$.

4.3. Attack complexity

In the current section we estimate the complexity of the described attack.

At the first stage the adversary should form 2^{15} queries consisting of $2^{57}-1$ blocks and, hence, store near 2^{72} blocks in a sorted list. So the complexity of the first stage is near

$$72 \cdot 2^{72} \approx 2^{79}.$$

The second stage needs processing just one block for $(2^{56}-2^{15})$ remained messages and find the collision with the values from the stored sorted list. So the complexity of the second stage may be bounded by

$$72 \cdot 2^{56} \approx 2^{63}.$$

As it was proven in the previous section the success probability p is greater than 0.63.

Summarizing this section we claim that with the total (time and query) complexity of near 2^{79} and probability of $p > 0.63$ the adversary can forge tag for 54 “non-authentic” messages.

4.4. Provable security

This attack may be interpreted using provable security ideas. Consider “8 bits” in the Auth model supposing that the used block cipher is a family of all permutations. In this case the adversary complexity can be measured only in terms of the total length of queries, since the resulting “8 bits” mode becomes the information theoretic object which security does not depend on adversary’s time complexity (only on the query complexity).

The adversary constructed in the proposed attack makes 2^{56} encryption queries of total length of near 2^{72} blocks and then one decryption query of length no more than 1 block. So we can estimate the advantage in the Auth model as follows:

$$\mathsf{Adv}^{\mathrm{Auth}}_{8bits}(\mathcal{A}) > 0.63,$$

where $\mathcal{A}$ makes the above-mentioned amount of queries of the certain lengths.

The total length of all queries is slightly bigger than the birthday bound, but significantly lower than the trivial random guessing forgery attack complexity. For the original CCM mode the proven bound [5] is near the birthday attack complexity (for the total length of processed data). But there is no known attacks of such complexity on the CCM mode, hence the complexity of real attack has to be at least the birthday attack complexity .

5. Conclusion

In the current paper the near birthday attack for the “8 bits” mode was proposed. The described attack is based on the adversary’s possibility to obtain clear tag values and then to compare them with keystream blocks used for encryption. Note that the considered mode may be easily modified to resist this attack with a minor performance loss. The only modification is to additionally encrypt the tag value (as in the original CCM mode). Thus, under secure block cipher the adversary will not obtain information about tag values.

6. Acknowledgments

We thank Evgeny K. Alekseev for useful discussions and comments during this work.

References

[1] Bellare M., Rogaway P., Wagner D., “The EAX mode of operation”, FSE 2004, Lect. Notes Comput. Sci., **3017**, 2004, 389–407.

[2] “Competition for Authenticated Encryption: Security, Applicability, and Robustness. CAESAR.” (2014), http://competitions.cr.yp.to/caesar.html.

[3] “802.11-2016 — IEEE Standard for Information technology – Telecommunications and information exchange between systems. Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications” (14 Dec. 2016), https://ieeexplore.ieee.org/document/7786995/.

[4] *Information technology. Cryptographic protection of information. Block cipher modes of operation. GOST R 34.13-2015*, M.: STANDARTINFORM, 2016 (in Russian).

[5] Jonsson J., “On the security of CTR + CBC-MAC”, SAC 2002, Lect. Notes Comput. Sci., **2595**, 2003, 76–93.

[6] Rescorla E., “The transport layer security (TLS) protocol version 1.3”, RFC, **8446**: Internet Engineering Task Force (IETF), August 2018, 160 pp., https://tools.ietf.org/html/rfc8446

[7] Rogaway P., Shrimpton T., “A provable-security treatment of the key-wrap problem”, EUROCRYPT 2006, Lect. Notes Comput. Sci., **4004**, 2006, 373–390.

[8] Shrimpton T., “A characterization of authenticated-encryption as a form of chosen-ciphertext security”, *IACR ePrint Archive* (2004), Report 2004/272, 7 pp., https://eprint.iacr.org/2004/272.pdf

[9] Whiting D., Housley R., Ferguson N., “Counter with CBC-MAC (CCM)”, RFC, **3610**: Internet Engineering Task Force (IETF), September 2003, 26 pp., https://tools.ietf.org/html/rfc3610

Appendix

Lemma. *Let* $x_1, \ldots, x_s$ *and* $y_1, \ldots, y_t$ *be different non-zero vectors from* V_n *and* β *be an arbitrary vector from* V_n. *Consider a uniform distribution over* $Perm(V_n)$. *Then*

$$\Pr_{P \in Perm(V_n)} \left[\{P(x_m)\}_{m=1}^{s} \cap \{P(P(y_k) \oplus K_1(P(0^n)) \oplus \beta)\}_{k=1}^{t} \neq \varnothing \right] \geqslant$$

$$\geqslant 1 - e^{-t(s-1)/2^n}.$$

Proof. Let us estimate the probability of the opposite event

$$\Pr_{P \in Perm(V_n)} \left[\{P(x_m)\}_{m=1}^{s} \cap \{P(P(y_k) \oplus K_1(P(0^n)) \oplus \beta))\}_{k=1}^{t} = \varnothing \right] =$$

$$= \Pr_{P \in Perm(V_n)} \left[\{x_m\}_{m=1}^{s} \cap \{P(y_k) \oplus K_1(P(0^n)) \oplus \beta)\}_{k=1}^{t} = \varnothing \right] =$$

$$= \Pr_{P \in Perm(V_n)} \left[\{x_m \oplus K_1(P(0^n)) \oplus \beta\}_{m=1}^{s} \cap \{P(y_k)\}_{k=1}^{t} = \varnothing \right].$$

Since we have a uniform distribution over $Perm(V_n)$, we can calculate this probability by combinatorial methods:

$$\Pr_{P \in Perm(V_n)} \left[\{x_m \oplus K_1(P(0^n)) \oplus \beta\}_{m=1}^{s} \cap \{P(y_k)\}_{k=1}^{t} = \varnothing \right] =$$

$$= \frac{\#\{P \in Perm(V_n) \,|\, \{x_m \oplus K_1(P(0^n)) \oplus \beta\}_{m=1}^{s} \cap \{P(y_k)\}_{k=1}^{t} = \varnothing\}}{|Perm(V_n)|} =$$

$$= \frac{1}{2^n!} \sum_{\alpha \in V_n} \#\big\{P \in Perm(V_n) \,\big|\, P(0^n) \alpha \text{ and } \{x_m \oplus K_1(\alpha) \oplus \beta\}_{m=1}^{s} \cap \{P(y_k)\}_{k=1}^{t} = \varnothing\big\}. \quad (2)$$

Note that given a fixed α the permutations to be enumerated under the sum sign have the following properties: 1) the value of a permutation on zero vector 0^n is equal to α; 2) the values of a permutation on the vectors $y_1, \ldots, y_t$ may be arbitrary, but should not belong to the s-element set $S_\alpha = \{x_m \oplus K_1(\alpha) \oplus \beta\}_{m=1}^{s}$. There are two cases: $\alpha \in S_\alpha$ and $\alpha \notin S_\alpha$. Depending on the case the number of these permutations may be slightly different:

$$\#\{P \in Perm(V_n) \,|\, P(0^n) = \alpha \text{ and } S_\alpha \cap \{P(y_k)\}_{k=1}^{t} = \varnothing\} =$$

$$= \begin{cases} (2^n - (t+1))! \prod_{j=0}^{t-1} (2^n - s - j), & \text{if} \quad \alpha \in S_\alpha, \\ (2^n - (t+1))! \prod_{j=1}^{t} (2^n - s - j), & \text{if} \quad \alpha \notin S_\alpha. \end{cases} \quad (3)$$

We need an upper bound of the opposite event probability. So, taking into account formulas (2) and (3), we have the following inequalities:

$$\Pr_{P\in Perm(V_n)}\left[\{x_m \oplus K_1(P(0^n)) \oplus \beta\}_{m=1}^{s} \cap \{P(y_k)\}_{k=1}^{t} = \varnothing\right] \leqslant$$

$$\leqslant 2^n \cdot \frac{(2^n - s)\cdot(2^n - s - 1)\cdot \ldots \cdot (2^n - s - (t-1))\cdot(2^n - (t+1))!}{2^n!} =$$

$$= \frac{2^n - s}{2^n - 1} \cdot \frac{2^n - s - 1}{2^n - 2} \cdot \ldots \cdot \frac{2^n - s - (t-1)}{2^n - t} \leqslant$$

$$\leqslant \left(\frac{2^n - (s-1)}{2^n}\right)^t = \left(1 - \frac{s-1}{2^n}\right)^t = e^{t \ln\left(1 - \frac{s-1}{2^n}\right)} \leqslant$$

$$\leqslant e^{-t(s-1)/2^n}.$$

This estimate of the opposite event probability proves the lemma. □