Misuse-resistant MGM2 mode

Liliya Akhmetzyanova, Evgeny Alekseev, Alexandra Babueva,
Andrey Bozhko and Stanislav Smyshlyaev

CryptoPro LLC, Russia
{lah, alekseev, babueva, bozhko, svs}@cryptopro.ru

Abstract

We introduce a modification of the Russian standardized AEAD MGM mode – an MGM2 mode, for which a nonce is not encrypted anymore before using it as an initial counter value. For the new mode we provide security bounds regarding security notions in the nonce-misuse setting (MRAE-integrity and CPA-resilience). The obtained bounds are even better than the bounds obtained for the original MGM mode regarding standard security notions.

Keywords: MGM, AEAD mode, security notion, security bounds, nonce-misuse, misuse-resistant

1 Introduction

Authenticated Encryption with Associated Data (AEAD) schemes, which aim at providing both integrity and confidentiality of data, are recently considered to be among the most widely used cryptographic schemes. Therefore, the security of such schemes is crucial. Security analysis of AEAD-schemes is usually carried out in the provable security paradigm regarding standard notions, introduced in [6], they are IND-CCA and IND-CPA for confidentiality and INT-CTXT for integrity.

One of the examples of such schemes is an MGM block cipher mode of operation, that was adopted in Russia as a standard AEAD-mode [13]. The MGM plaintext encryption procedure is quite similar to encryption in the CTR2 [18] mode. The main element of the MGM authentication procedure is a multilinear function with secret coefficients produced in the same way as the secret masking blocks used for plaintext encryption. Integrity and confidentiality of MGM were analysed in [1] regarding standard security notions. Since MGM was not developed with provable security in mind, security proofs turned out to be cumbersome and, hence, difficult to verify.

Even though analysis of AEAD-schemes in the standard models is mandatory and enough for use in many applications, some environments require other unusual cryptographic properties, e.g. leakage resilience [5], RUP («Release of Unverified Plaintext») security [3], KDM («Key Dependent Message») security [10], misuse-resistance [19], etc. In the current paper we focus on misuse-resistance or nonce misuse property [19]. A nonce (number used only once) is an input to encryption or decryption algorithms of AEAD-schemes that has to be unique (within a fixed key), but in some applications such requirement is hard to obtain, not to mention implementation faults. Misuse-resistant schemes aim to ensure the best possible security when faulty nonce is provided.

Security notions for misuse-resistant authenticated encryption were originally proposed by Rogaway and Shrimpton in [19] and further developed in [4]. Strong variant of misuse-resistant notions, called MRAE («Misuse-Resistant AE»), was introduced in [19]. This notion is the extensions of the IND-CCA and INT-CTXT notions by allowing an adversary to repeat nonces in all of its' queries. The MRAE notion is similar to a DAE notion [19] («Deterministic AE») where confidentiality is formalised as follows: ciphertext of each *new* query (not only new nonce) has to be indistinguishable from a random string. Providing such confidentiality is rather strong, and trying to achieve it seems to lead to loss in performance. All MRAE-secure modes, known to the authors, demand sufficiently larger amount of block cipher calls [16] or lose online property [15, 22]. For the reasons above, weak notions for confidentiality called CPA-res and CCA-res («Chosen Plaintext/Ciphertext Attack-resilience») were introduced in [4], where the confidentiality should be achieved only for messages that were encrypted correctly using unique nonces.

In nonce-misuse setting the MGM mode is obviously insecure in the MRAE model regarding confidentiality since counter-based encryption is actually used. MRAE-integrity of the MGM mode was analysed in [17]: the birthday type attack was proposed. However, no lower bounds for MGM were proven. So, there is a «hope» to provide non-trivial security bounds for MGM in the MRAE-integrity and CCA-res models.

Motivating by expectation that the security proof for the MGM mode in non-standard models will be even more complex, than in standard ones, we introduce modification of MGM mode – MGM2. The main difference between two modes lies in the way how secret masking blocks and secret coefficients of the multilinear function are produced – for the MGM2 mode this process is carried out in the CTR [18] style (without preliminary nonce encryption). Note, that the main cryptographic core of the construction, namely multilinear function, is not changed. We provide the security bounds for MGM2 in the MRAE-integrity and CPA-res models that turned out to be even better, than the bounds for the original MGM mode in the standard models. The corresponding security proofs are relatively short and, we hope, easier to verify. Among other advantages, the design of the MGM2 mode also allows to transparently integrate internal re-keying without a master key [2] (in the same way as for CTR-ACPKM [14] done) to achive new security properties like leakage-resilience and increase key lifetime.

2 Preliminaries

By $\{0,1\}^s$ we denote the set of s-component bit strings and by $\{0,1\}^*$ we denote the set of all bit strings of finite length including the empty string. Let $|a|$ be the bit length of the string $a \in \{0,1\}^*$. For a bit string a we denote by $|a|_n = \lceil |a|/n \rceil$ the length of the string a in n-bit blocks. By $\{0,1\}^{\leqslant s}$ we denote the set of bit strings which length is less or equal to s.

For a string $a \in \{0,1\}^*$ and a positive integer $l \leqslant |a|$ let $\mathrm{msb}_\ell(a)$ be the string, consisting of the leftmost l bits of a. For nonnegative integers l and i let $\mathrm{str}_l(i)$ be l-bit representation of i with the least significant bit on the right. For bit strings $a \in \{0,1\}^n$ and $b \in \{0,1\}^n$ we denote by $a \otimes b$ a string which is the result of their multiplication in $GF(2^n)$ (here strings encode polynomials in the standard way). If the value s is chosen from a set S uniformly at random, then we denote $s \xleftarrow{\mathcal{U}} S$. We define a function $\mathsf{Set1}_r \colon \{0,1\}^n \to \{0,1\}^n$, $\mathsf{Set1}_r(x) = x$ or $(\overbrace{0\ldots0}^{r}1\overbrace{0\ldots0}^{n-r-1})$, $0 \leqslant r < n$.

For any set S, define $Perm(S)$ as the set of all bijective mappings on S (permutations on S), and $Func(S)$ as the set of all mappings from S to S. A block cipher E (or just a cipher) with a block size n and a key size k is the permutation family $\big(E_K \in Perm(\{0,1\}^n) \mid K \in \{0,1\}^k\big)$, where K is a key.

3 Security models

This section introduces models for an adversary that may repeat nonces in its queries.

We define security model using the notion of «experiment» or «game» played between a challenger and an adversary. The adversary and challenger are modelled using consistent interactive probabilistic algorithms. The challenger simulates the functioning of the analysed cryptographic scheme for the adversary and may provide him access to one or more oracles (for details see [8]).

We describe challengers and adversaries using pseudocodes with the following notations. If a variable x gets a value val then we denote $x \leftarrow val$. Similarly, if a variable x gets the value of a variable y then we denote $x \leftarrow y$. If the variable x gets the result of a probabilistic algorithm $\mathcal{A}$ we denote $x \xleftarrow{\$} \mathcal{A}$. If we need to emphasize that $\mathcal{A}$ is deterministic than we denote it by $x \leftarrow \mathcal{A}$. The event when $\mathcal{A}$ returned value val as a result is denoted by $\mathcal{A} \rightarrow val$ (or $\mathcal{A} \xrightarrow{\$} val$ if $\mathcal{A}$ is probabilistic).

Firstly, we introduce the general definition of an AEAD-scheme.

Definition 1. *Let* $\boldsymbol{K}$ *be a set of keys,* $\boldsymbol{P}$ *be a set of plaintexts,* $\boldsymbol{A}$ *be a set of associated data,* $\boldsymbol{C}$ *be a set of ciphertexts, and* $\boldsymbol{T}$ *be a set of tags. An AEAD-scheme with nonce is a set of algorithms* $\Pi = \{\Pi.\mathsf{Gen},\ \Pi.\mathsf{Enc},\ \Pi.\mathsf{Dec}\}$*, where*

- $\Pi.\mathsf{Gen}() \xrightarrow{\$} K$: *A probabilistic key generation algorithm outputting a key* $K \in \boldsymbol{K}$*.*
- $\Pi.\mathsf{Enc}(K, N, A, P) \rightarrow (C, T)$: *A deterministic algorithm of authenticated encryption taking a key* $K \in \boldsymbol{K}$*, a nonce* $N \in \boldsymbol{N}$*, associated data* $A \in \boldsymbol{A}$*, a plaintext* $P \in \boldsymbol{P}$*. An output of the algorithm is a ciphertext* $C \in \boldsymbol{C}$ *and a tag* $T \in \boldsymbol{T}$*.*
- $\Pi.\mathsf{Dec}(K, N, A, C, T) \rightarrow P$: *A deterministic algorithm of authenticated decryption taking a key* $K \in \boldsymbol{K}$*, a nonce* $N \in \boldsymbol{N}$*, associated data* $A \in \boldsymbol{A}$*, a ciphertext* $C \in \boldsymbol{C}$ *and a tag* $T \in \boldsymbol{T}$*. An output of the algorithm is a plaintext* $P \in \boldsymbol{P}$ *or error symbol* $\perp$*.*

Let define a MRAE-int («Misuse-Resistant Authenticated Encryption - integrity») security notion for integrity (the integrity part of MRAE [19]).

Definition 2 (MRAE-int)**.** *For an AEAD-scheme* Π *the advantage of a* MRAE-int-*adversary* $\mathcal{A}$ *is defined as follows:*

$$\mathsf{Adv}_{\Pi}^{\text{MRAE-int}}(\mathcal{A}) = \Pr\big[\mathbf{Exp}_{\Pi}^{\text{MRAE-int}}(\mathcal{A}) \rightarrow 1\big],$$

where experiment $\mathbf{Exp}_{\Pi}^{\text{MRAE-int}}$ *is defined below:*

```
$\mathbf{Exp}_{\Pi}^{\text{MRAE-int}}(\mathcal{A})$
$K \xleftarrow{\$} \Pi.\mathsf{Gen}(\,)$
$sent \leftarrow \emptyset$
$win \leftarrow \mathsf{false}$
$\mathcal{A}^{Encrypt,Decrypt}(\,)$
return $win$
```

```
Oracle $Encrypt(N, A, P)$
$(C, T) \leftarrow \Pi.\mathsf{Enc}(K, N, A, P)$
$sent \leftarrow sent \cup \{(N, A, C, T)\}$
return $(C, T)$
```

```
Oracle $Decrypt(N, A, C, T)$
$P \leftarrow \Pi.\mathsf{Dec}(K, N, A, C, T)$
if $(P \neq \perp) \wedge ((N, A, C, T) \notin sent)$:
    $win \leftarrow \mathsf{true}$
return $P$
```

Let introduce the CPA-res («Chosen Plaintext Attack - resilience») security notion for confidentiality, defined in [4].

Definition 3 (CPA-res)**.** *For an AEAD-scheme* Π *with the tag length* s *the advantage of a* CPA-res-*adversary* $\mathcal{A}$ *is defined as follows:*

$$\mathsf{Adv}_{\Pi}^{\text{CPA-res}}(\mathcal{A}) = \Pr\big[\mathbf{Exp}_{\Pi}^{\text{CPA-res-1}}(\mathcal{A}) \rightarrow 1\big] - \Pr\big[\mathbf{Exp}_{\Pi}^{\text{CPA-res-0}}(\mathcal{A}) \rightarrow 1\big],$$

where experiments $\mathbf{Exp}^{\text{CPA-res-}b}$, $b \in \{0, 1\}$, *are defined below:*

```
$\mathbf{Exp}_{\Pi}^{\text{CPA-res-}b}(\mathcal{A})$
$K \xleftarrow{\$} \Pi.\mathsf{Gen}(\,)$
$\mathcal{L}_1, \mathcal{L}_2 \leftarrow \emptyset$
$b \xleftarrow{\$} \mathcal{A}^{O_1,O_2}(\,)$
return $b$
```

```
Oracle $O_1(N, A, P)$
if $N \in \mathcal{L}_1 \cup \mathcal{L}_2$:
  return $\perp$
if $b = 1$:
  $(C, T) \leftarrow \Pi.\mathsf{Enc}(K, N, A, P)$
else :
  $C \,\|\, T \xleftarrow{\mathcal{U}} \{0,1\}^{|P|+s}$
$\mathcal{L}_1 \leftarrow \mathcal{L}_1 \cup \{N\}$
return $(C, T)$
```

```
Oracle $O_2(N, A, P)$
if $N \in \mathcal{L}_1$:
  return $\perp$
$(C, T) \leftarrow \Pi.\mathsf{Enc}(K, N, A, P)$
if $N \notin \mathcal{L}_2$:
  $\mathcal{L}_2 \leftarrow \mathcal{L}_2 \cup \{N\}$
return $(C, T)$
```

In [4] the CCA-res («Chosen Ciphertext Attack - resilience») security notion is also defined. This notion differs from CPA-res in that an adversary is provided with additional access to a decryption oracle. By the technique similar to one described in [20] it is easy to show that MRAE-int-security and CPA-res-security jointly imply CCA-res-security. Therefore, further we consider the CPA-res security notion only.

4 MGM2 mode

```
$\mathsf{MGM2.Gen}()$
$K \xleftarrow{\mathcal{U}} \{0,1\}^k$
return $K$

$\mathsf{MGM2.Enc}(K, N, A, P)$
$h \leftarrow |A|_n, q \leftarrow |P|_n$
$\ell \leftarrow h + q + 1$
............ Encryption ............
for $i = 1 \dots q$ do:
  $\Gamma_i \leftarrow \mathsf{E}_K(N\|00\|\mathrm{str}_{n-r-2}(i-1))$
$C \leftarrow P \oplus \mathrm{msb}_{|P|}(\Gamma_1 \,\|\, \dots \,\|\, \Gamma_q)$
............. Padding .............
$a \leftarrow n|A|_n - |A|$
$c \leftarrow n|C|_n - |C|$
$len \leftarrow \mathrm{str}_{n/2}(|A|) \,\|\, \mathrm{str}_{n/2}(|C|)$
$M_1\| \dots \|M_l \leftarrow A\|0^a\|C\|0^c\|len$
.......... Tag generation ..........
for $i = 1 \dots \ell$ do:
  $H_i \leftarrow \mathsf{E}_K(N\|01\|\mathrm{str}_{n-r-2}(i-1))$
$\tau \leftarrow \mathsf{Set1}_r\left(\bigoplus_{i=1}^{l} M_i \otimes H_i\right)$
$T \leftarrow \mathrm{msb}_s(\mathsf{E}_K(\tau))$
return $(C, T)$
```

```
$\mathsf{MGM2.Dec}(K, N, A, C, T)$
$h \leftarrow |A|_n, q \leftarrow |C|_n$
$\ell \leftarrow h + q + 1$
............. Padding .............
$a \leftarrow n|A|_n - |A|$
$c \leftarrow n|C|_n - |C|$
$len \leftarrow \mathrm{str}_{n/2}(|A|) \,\|\, \mathrm{str}_{n/2}(|C|)$
$M_1\| \dots \|M_l \leftarrow A\|0^a\|C\|0^c\|len$
......... Tag verification .........
for $i = 1 \dots \ell$ do:
  $H_i \leftarrow \mathsf{E}_K(N\|01\|\mathrm{str}_{n-r-2}(i-1))$
$\tau \leftarrow \mathsf{Set1}_r\left(\bigoplus_{i=1}^{l} M_i \otimes H_i\right)$
$T' \leftarrow \mathrm{msb}_s(\mathsf{E}_K(\tau))$
if $T' \neq T$: return $\perp$
............ Decryption ............
for $i = 1 \dots q$ do:
  $\Gamma_i \leftarrow \mathsf{E}_K(N\|00\|\mathrm{str}_{n-r-2}(i-1))$
$P \leftarrow C \oplus \mathrm{msb}_{|C|}(\Gamma_1 \,\|\, \dots \,\|\, \Gamma_q)$
return $P$
```

Figure 1: AEAD mode MGM2

In this section we describe a new AEAD mode called MGM2 which is a slight modification of the MGM mode. By MGM2[E, r, s] we will denote the parametrized MGM2 mode with a block cipher E (with block size n and key size k), a nonce length r, $\frac{n}{2} \leqslant r \leqslant \frac{3n}{4}$ and a tag length s, $1 \leqslant s \leqslant n$.

For $\mathsf{MGM2}[\mathsf{E}, r, s]$ the corresponding sets are as follows: $\boldsymbol{K} = \{0,1\}^k$, $\boldsymbol{N} = \{0,1\}^r$, $\boldsymbol{A} = \boldsymbol{P} = \boldsymbol{C} = \{0,1\}^{\leqslant n(2^{n-r-2}-1)}$, $\boldsymbol{T} = \{0,1\}^s$. Moreover, the following condition should be satisfied: $0 < |A| + |P| \leqslant n(2^{n-r-2}-1)$. The key generation, encryption and decryption algorithms are defined in Figure 1.

Difference from MGM. The main difference of the new $\mathsf{MGM2}$ mode from the original MGM mode is in the modification of the way to produce the mask values for encryption (Γ_i), the coefficients of the multilinear function (H_i), and the tag values T. In $\mathsf{MGM2}$ block cipher inputs, used to generate values for different use cases (we have three use cases: Γ_i, H_i, T), are separated by fixing the certain bits of inputs. Such a modification allows to obtain better security bounds, since the collision among block cipher inputs may occur only among values τ.

5 Security analysis

The security of block cipher modes of operation is commonly analyzed under assumption that the underlying block cipher is PRP-CPA-secure (see [7]), i.e. E_K for a random key is computationally indistinguishable from a random permutation. We follow this approach and provide security bounds directly for the mode with a random permutation.

We write $\mathsf{MGM2}[Perm(n), r, s]$ for $\mathsf{MGM2}$ that uses a random permutation π as E_K and we write $\mathsf{MGM2}[Func(n), r, s]$ for $\mathsf{MGM2}$ that uses a random function ρ.

5.1 Integrity

Theorem 1. *For any* MRAE-int*-adversary $\mathcal{A}$, making at most Q_E queries to the Encrypt oracle and at most Q_D queries to the Decrypt oracle, where the total block-length of associated data in all queries is at most σ_A and the total block-length of plaintexts and ciphertexts in all queries is at most σ_P,*

$$\mathsf{Adv}^{\text{MRAE-int}}_{\mathsf{MGM2}[Perm(n),r,s]}(\mathcal{A}) \leqslant \left(\frac{Q(Q-1)}{2^n} + \frac{Q_D}{2^s}\right)\left(1 - \frac{\sigma-1}{2^n}\right)^{-\sigma/2}, \tag{1}$$

where $Q = Q_E + Q_D$ and $\sigma = 2\sigma_P + \sigma_A + 2Q$.

Note that for $n \geqslant 128$ and $\sigma \leqslant 2^{n/2}$, the bound (1) can be converted as follows:

$$\mathsf{Adv}^{\text{MRAE-int}}_{\mathsf{MGM2}[Perm(n),r,s]}(\mathcal{A}) \leqslant 1.7\left(\frac{Q(Q-1)}{2^n} + \frac{Q_D}{2^s}\right). \tag{2}$$

Thus, the $\mathsf{MGM2}$ mode provides integrity beyond birthday bound. Note that for the original MGM mode, if total amount of processed data achieves $2^{n/2}$, the bound, presented in [1], becomes trivial. This result also allows to use $\mathsf{MGM2}$ as a MAC function (and even as a PRF, see further) by fixing N with the constant value. Further we provide proof of the Theorem 1.

Proof. The proof is carried out in two steps. In the first step we introduce an auxiliary MAC-scheme with nonce called $\mathsf{MGM2\text{-}MAC}[r, s]$ and estimate its security (see Section 5.1.1).

In the second step we show that the UF-CMA-security of the $\mathsf{MGM2\text{-}MAC}[r, s]$ scheme tightly implies the MRAE-int-security of the $\mathsf{MGM2}[Func(n), r, s]$ scheme (see Section 5.1.2).

The security bound for $\mathsf{MGM2}[Perm(n), r, s]$ is obtained using Bernstein's result [9], Theorem 2.3. Due to that theorem for any distinguisher $\mathcal{D}^f$ with oracle $f\colon \{0,1\}^n \to \{0,1\}^n$, making at most q queries, the following inequality holds:

$$\Pr[\mathcal{D}^\pi \to 1] \leqslant \Pr[\mathcal{D}^\rho \to 1] \cdot \left(1 - \frac{q-1}{2^n}\right)^{-q/2},$$

where $\pi \xleftarrow{\mathcal{U}} Perm(n)$ and $\rho \xleftarrow{\mathcal{U}} Func(n)$.

If we let $\mathcal{D}$ be the algorithm $\mathbf{Exp}^{\text{MRAE-int}}_{\mathsf{MGM2}[\cdot,r,s]}(\mathcal{A})$, where it makes queries to the oracle instead of calling the underlying function (block cipher), we obtain the target bound. □

5.1.1 Security of MGM2-MAC

We introduce an auxiliary MAC-scheme with nonce called $\mathsf{MGM2\text{-}MAC}[r, s]$ based on the scheme $\mathsf{MGM2}[Func(n), r, s]$. Usually MAC-scheme is defined as a set of algorithms $\mathsf{MAC} = \{\mathsf{MAC.Gen}, \mathsf{MAC.Tag}, \mathsf{MAC.Verify}\}$, for $\mathsf{MGM2\text{-}MAC}[r, s]$ these algorithms are defined in Figure 2. This scheme is defined for the message set $\{M = M_1\|\ldots\|M_\ell\colon M_i \in \{0,1\}^n,\ M_\ell \neq 0^n,\ 1 \leqslant \ell \leqslant 2^{n-r-2}\}$ (the message length is divisible by n, the last block is non-zero).

```
MGM2-MAC.Gen()
$\rho, \rho' \xleftarrow{\mathcal{U}} Func(n)$
$K \leftarrow (\rho, \rho')$
return $K$
```

```
MGM2-MAC.Tag($K, N, M$)
$\tau \leftarrow \mathsf{PreTag}(\rho', N, M)$
$T \leftarrow \mathrm{msb}_s(\rho(\tau))$
return $T$
```

```
PreTag($\rho', N, M$)
$l \leftarrow |M|_n$
for $i = 1 \ldots \ell$ do:
    $H_i \leftarrow \rho'(N\|01\|\mathrm{str}_{n-r-2}(i-1))$
$\tau \leftarrow \mathsf{Set1}_r\left(\bigoplus_{i=1}^{l}(M_i \otimes H_i)\right)$
return $\tau$
```

```
MGM2-MAC.Verify($K, N, M, T$)
$\tau \leftarrow \mathsf{PreTag}(\rho', N, M)$
$T' \leftarrow \mathrm{msb}_s(\rho(\tau))$
if $T' \neq T$: return false
return true
```

Figure 2: The scheme MGM2-MAC

Firstly, we introduce the standard PRF security notion (in nonce-misuse setting) for nonce-based MAC-schemes and obtain the PRF-security bound for the MGM2-MAC scheme.

Definition 4 (PRF). *For a MAC-scheme* MAC *the advantage of a* PRF*-adversary* $\mathcal{A}$ *is defined as follows:*

$$\mathsf{Adv}^{\text{PRF}}_{\mathsf{MAC}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\text{PRF}-1}_{\mathsf{MAC}}(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{PRF}-0}_{\mathsf{MAC}}(\mathcal{A}) \to 1\right],$$

where experiments $\mathbf{Exp}^{\text{PRF}-b}_{\mathsf{MAC}}(\mathcal{A}),\ b \in \{0,1\}$ *are defined below:*

```
$\mathbf{Exp}^{\text{PRF}-b}_{\mathsf{MAC}}(\mathcal{A})$
if $b = 1$:
    $K \xleftarrow{\$} \mathsf{MAC.Gen}(\,)$
$sent \leftarrow \emptyset$
$b' \xleftarrow{\$} \mathcal{A}^{Tag^b}(\,)$
return $b'$
```

```
Oracle $Tag^1(N, M)$
if $(N, M) \in sent$:
    return $\perp$
$T \leftarrow \mathsf{MAC.Tag}(K, N, M)$
$sent \leftarrow sent \cup \{(N, M)\}$
return $T$
```

```
Oracle $Tag^0(N, M)$
if $(N, M) \in sent$:
    return $\perp$
$T \xleftarrow{\mathcal{U}} \{0,1\}^s$
$sent \leftarrow sent \cup \{(N, M)\}$
return $T$
```

Lemma 1. *For any* PRF*-adversary $\mathcal{A}$, making at most Q queries to the Tag oracle:*

$$\mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{MGM2\text{-}MAC}[r,s]}(\mathcal{A}) \leqslant \frac{Q(Q-1)}{2^n}.$$

Proof. Let define auxiliary experiments $\mathbf{Exp}^0$ and $\mathbf{Exp}^1$ (see Figure 3), which differ from the experiment $\mathbf{Exp}^{\mathrm{PRF-1}}_{\mathsf{MGM2\text{-}MAC}[r,s]}$ as follows. During the setup phase, the tau set is additionally initialized to empty value, and the flag bad is set to false. During the experiment execution, the values τ are put in the tau set, and the flag bad is set to true iff collision among the τ values occurs. Also, in the $\mathbf{Exp}^0$ experiment the tag value is chosen from $\{0,1\}^s$ uniformly at random in the case of collision (see line in box).

```
$\mathbf{Exp}^b(\mathcal{A}),\ b \in \{0,1\}$
$(\rho, \rho') \xleftarrow{\$} \mathsf{MGM2\text{-}MAC.Gen}(\,)$
$bad \leftarrow \mathsf{false}$
$tau, sent \leftarrow \emptyset$
$b' \xleftarrow{\$} \mathcal{A}^{Tag^b}(\,)$
return $b'$
```

```
Oracle $Tag^b(N, M)$
if $(N, M) \in sent$:
    return $\bot$
$\tau \leftarrow \mathsf{PreTag}(\rho', N, M)$
$T \leftarrow \mathrm{msb}_s(\rho(\tau))$
if $\tau \in tau$:
    $bad \leftarrow \mathsf{true}$
    $\boxed{\textbf{if } b = 0\colon T \xleftarrow{\mathcal{U}} \{0,1\}^s}$
$tau \leftarrow tau \cup \{\tau\}$
$sent \leftarrow sent \cup \{(N, M)\}$
return $T$
```

Figure 3: Experiments $\mathbf{Exp}^0$ and $\mathbf{Exp}^1$

It is easy to see that $\mathbf{Exp}^1$ is exactly the $\mathbf{Exp}^{\mathrm{PRF-1}}_{\mathsf{MGM2\text{-}MAC}[r,s]}$ experiment. Moreover, for any $\mathcal{A}$ the value $\Pr\big[\mathbf{Exp}^0(\mathcal{A}) \Rightarrow 1\big]$ is exactly the value $\Pr\Big[\mathbf{Exp}^{\mathrm{PRF-0}}_{\mathsf{MGM2\text{-}MAC}[r,s]}(\mathcal{A}) \Rightarrow 1\Big]$. Indeed, in the $\mathbf{Exp}^0$ experiment all tag values T are produced according to the uniform distribution as in $\mathbf{Exp}^{\mathrm{PRF-0}}_{\mathsf{MGM2\text{-}MAC}[r,s]}$ for the following reasons. For the queries, whose corresponding τ value is new (not in the current set tau), the uniform random function ρ is applied to the new input and, therefore, returns uniform output. For the other queries the T value is directly sampled uniformly at random (see the line in box, Figure 3). Therefore,

$$\mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{MGM2\text{-}MAC}[r,s]}(\mathcal{A}) = \Pr\big[\mathbf{Exp}^1(\mathcal{A}) \Rightarrow 1\big] - \Pr\big[\mathbf{Exp}^0(\mathcal{A}) \Rightarrow 1\big].$$

Note that before the bad flag is set to true (denote this event as $bad = \mathsf{true}$) the $\mathbf{Exp}^0$ and $\mathbf{Exp}^1$ experiments are functioning identically, therefore (due to Lemma 2, [8]) the following inequality holds:

$$\Pr\big[\mathbf{Exp}^1(\mathcal{A}) \Rightarrow 1\big] - \Pr\big[\mathbf{Exp}^0(\mathcal{A}) \Rightarrow 1\big] \leqslant \Pr[bad = \mathsf{true}].$$

Let estimate $\Pr[bad = \mathsf{true}]$. Without loss of generality, we assume the adversary to be deterministic and making Q pairwise different queries (N_i, M^i), $i = 1, \ldots, Q$. Denote by $\widetilde{\rho}$ and $\widetilde{\rho}'$ the uniform random variables with sample space $Func(n)$. We will also use notation coll^i, $i = 2, \ldots, Q$, to denote the event that the bad flag is set to true during the first i queries processing. Thus,

$$\Pr[bad = \mathsf{true}] = \sum_{i=2}^{Q} \Pr\Big[\mathsf{coll}^i \cap \overline{\mathsf{coll}^{i-1}}\Big],$$

where the probability is defined by the random variables $\widetilde{\rho}$ and $\widetilde{\rho}'$. Let estimate the value $\Pr\left[\mathsf{coll}^i \cap \overline{\mathsf{coll}^{i-1}}\right]$ for any $i = 2, \ldots, Q$.

Note, that each i-th query – the pair (N_i, M^i), where $M^i = M^i_1\|...\|M^i_{l_i}$, $M^i_j \in \{0,1\}^n$ – is determined by the tag values $T_1, \ldots, T_{i-1}$ previously obtained from the oracle. Without loss of generality, we assume $l_1 = \ldots = l_i$. Indeed, otherwise we can pad the messages with zero blocks to the length $l := \max(l_1, \ldots, l_i)$. This does not change the tag value, and the padded messages will stay pairwise different because of $M^j_{l_j} \neq 0^n$. Therefore, the $T_1, \ldots, T_{i-1}$ values fully determine l and $(N_1, M^1), \ldots, (N_i, M^i)$.

For fixed N_j we denote by $\widetilde{H^j_k}$, $j = 1, \ldots, i$; $k = 1, \ldots, l$, the random variable $\widetilde{\rho}'(N_j\|01\|\mathrm{str}_{n-r-2}(k-1))$. Notice that $\Pr\left[\widetilde{H^j_k} = B\right] = \dfrac{1}{2^n}$ for any $B \in \{0,1\}^n$. Note that the random variables $\widetilde{H^j_k}$ and $\widetilde{H^t_k}$ for some $j \neq t$ and any k are dependent, namely $\Pr\left[\widetilde{H^j_k} = \widetilde{H^t_k}\right] = 1$, iff $N_k = N_j$.

For short we denote by $\widetilde{H^j}$ the random variable $(\widetilde{H^j_1}, \ldots, \widetilde{H^j_\ell})$. Also for set $H = (H_1, \ldots, H_\ell)$ and message $M = M_1\|\ldots\|M_\ell$ let $\tau(H, M)$ be the function $\mathsf{Set1}_r\left(\bigoplus\limits_{k=1}^{l} H_k \otimes M_k\right)$. So, we have

$$\Pr\left[\mathsf{coll}^i \cap \overline{\mathsf{coll}^{i-1}}\right] = \sum_{T_1,\ldots,T_{i-1}} \Pr\left[\mathsf{coll}^i \cap \overline{\mathsf{coll}^{i-1}} \cap \{\widetilde{T_j} = T_j\}_{j=1}^{i-1}\right],$$

where we write $\widetilde{T_j}$ for random variable $\mathrm{msb}_s(\widetilde{\rho}(\tau(\widetilde{H^j}, M^j)))$, and sum is taken over all $(T_1, \ldots, T_{i-1}) \in (\{0,1\}^s)^{i-1}$.

For fixed $(N_1, M^1), \ldots, (N_i, M^i)$ introduce the following conditions on set $H^1, \ldots, H^i$, $H^j := (H^j_1, \ldots, H^j_\ell)$, $j = 1, \ldots, i$:

Condition E_1: $\forall\ j, t,\ 1 \leqslant j < t \leqslant i-1:\ \tau(H^j, M^j) \neq \tau(H^t, M^t)$.

Condition E_2: $\exists\ j,\ 1 \leqslant j \leqslant i-1:\ \tau(H^i, M^i) = \tau(H^j, M^j)$.

For any fixed $T_1, \ldots, T_{i-1}$, and hence fixed $(N_1, M^1), \ldots, (N_i, M^i)$, the event $\mathsf{coll}^i \cap \overline{\mathsf{coll}^{i-1}}$ occurs iff random variables $\widetilde{H^1}, \ldots, \widetilde{H^i}$ take such values $H^1, \ldots, H^i$ that the conditions E_1 and E_2 are satisfied. For short we will denote the events that these conditions are satisfied by the same way, namely, by E_1 and E_2 correspondingly.

Note that fixing values H^j, $j = 1, \ldots, i$, leads to fixing values $\tau_j := \tau(H^j, M^j)$. Therefore,

$$\begin{aligned}\Pr\left[\mathsf{coll}^i \cap \overline{\mathsf{coll}^{i-1}}\right] &= \sum_{T_1,\ldots,T_{i-1}} \Pr\left[\mathsf{E}_1 \cap \mathsf{E}_2 \cap \{\widetilde{T_j} = T_j\}_{j=1}^{i-1}\right] = \\ &= \sum_{T_1,\ldots,T_{i-1}} \sum_{\substack{H^1,\ldots,H^i: \\ \mathsf{E}_1 \cap \mathsf{E}_2}} \Pr\left[\{\widetilde{H^j} = H^j\}_{j=1}^{i} \cap \{\mathrm{msb}_s(\widetilde{\rho}(\tau_j)) = T_j\}_{j=1}^{i-1}\right] = \\ &= \sum_{T_1,\ldots,T_{i-1}} \sum_{\substack{H^1,\ldots,H^i: \\ \mathsf{E}_1 \cap \mathsf{E}_2}} \Pr\left[\{\widetilde{H^j} = H^j\}_{j=1}^{i}\right] \cdot \Pr\left[\{\mathrm{msb}_s(\widetilde{\rho}(\tau_j)) = T_j\}_{j=1}^{i-1}\right].\end{aligned}$$

Here, sum is taken over all $H^1, \ldots, H^i$, $H^j \in (\{0,1\}^n)^l$, for which the E_1 and E_2 conditions are satisfied. The last transition is due to the fact that $\widetilde{\rho}$ and $\widetilde{H^j}$, $j = 1, \ldots, i$, are independent.

Consider the value $\Pr\left[\{\mathrm{msb}_s(\widetilde{\rho}(\tau_j)) = T_j\}_{j=1}^{i-1}\right]$. For any $T_1, \ldots, T_{i-1}$ and $H^1, \ldots, H^{i-1}$ for which the condition E_1 is satisfied, this probability is exactly the probability to sample function ρ, such that $i-1$ fixed inputs correspond to outputs with fixed s bits, i.e $\frac{1}{2^{s(i-1)}}$. Thus:

$$\Pr\left[\mathsf{coll}^i \cap \overline{\mathsf{coll}^{i-1}}\right] = \sum_{T_1,\ldots,T_{i-1}} \sum_{\substack{H^1,\ldots,H^i\colon \\ \mathsf{E}_1 \cap \mathsf{E}_2}} \Pr\left[\{\widetilde{H^j} = H^j\}_{j=1}^{i}\right] \cdot \frac{1}{2^{s(i-1)}} =$$

$$= \frac{1}{2^{s(i-1)}} \sum_{T_1,\ldots,T_{i-1}} \Pr[\mathsf{E}_1 \cap \mathsf{E}_2] \leqslant \frac{1}{2^{s(i-1)}} \sum_{T_1,\ldots,T_{i-1}} \Pr[\mathsf{E}_2].$$

Now consider $\Pr[\mathsf{E}_2]$ for any fixed $T_1, \ldots, T_{i-1}$, and, hence, any fixed $(N_1, M^1), \ldots, (N_i, M^i)$.

$$\Pr[\mathsf{E}_2] = \Pr\left[\exists\, j,\ 1 \leqslant j \leqslant i-1\colon\ \tau(\widetilde{H^i}, M^i) = \tau(\widetilde{H^j}, M^j)\right] =$$

$$= \Pr\left[\bigcup_{j=1}^{i-1} \left\{\tau(\widetilde{H^i}, M^i) = \tau(\widetilde{H^j}, M^j)\right\}\right] \leqslant \sum_{j=1}^{i-1} \Pr\left[\tau(\widetilde{H^i}, M^i) = \tau(\widetilde{H^j}, M^j)\right].$$

Let estimate $p := \Pr\left[\tau(\widetilde{H^i}, M^i) = \tau(\widetilde{H^j}, M^j)\right]$ for any $j = 1, \ldots, i-1$. We consider two cases:

1. $N_i \neq N_j$ (in this case $\widetilde{H^i_k}$ and $\widetilde{H^j_k}$ are independent).

2. $N_i = N_j$ (in this case $\widetilde{H^i_k}$ and $\widetilde{H^j_k}$ are dependent).

The first case: $p = \dfrac{\#\{H^i, H^j\colon \tau(H^i, M^i) = \tau(H^j, M^j)\}}{2^{2nl}}$.

$$\#\{H^i, H^j\colon \tau(H^i, M^i) = \tau(H^j, M^j)\} =$$

$$= \#\left\{H^i, H^j\colon \bigoplus_{k=1}^{l} H^i_k \otimes M^i_k = \bigoplus_{k=1}^{l} H^j_k \otimes M^j_k\right\} +$$

$$+ \#\left\{H^i, H^j\colon \bigoplus_{k=1}^{l} H^i_k \otimes M^i_k = \bigoplus_{k=1}^{l} H^j_k \otimes M^j_k \oplus \mathsf{Set1}_r(0^n)\right\}.$$

Since $M^i_{\ell_i} \neq 0^n$ for any i, the cardinality is $2 \cdot 2^{n(2l-1)}$. And, $p = \dfrac{2}{2^n}$.

The second case: $p = \dfrac{\#\{H^i\colon \tau(H^i, M^i) = \tau(H^i, M^j)\}}{2^{nl}}$.

$$\#\{H^i\colon \tau(H^i, M^i) = \tau(H^i, M^j)\} =$$

$$= \#\left\{H^i\colon \bigoplus_{k=1}^{l} H^i_k \otimes (M^i_k \oplus M^j_k) = 0^n\right\} +$$

$$+ \#\left\{H^i\colon \bigoplus_{k=1}^{l} H^i_k \otimes (M^i_k \oplus M^j_k) = \mathsf{Set1}_r(0^n)\right\}.$$

Since for the same nonce the messages M^i and M^j should be different, there exists k such that $M^i_k \oplus M^j_k \neq 0^n$. Therefore, the cardinality is $2 \cdot 2^{n(l-1)}$. And, $p = \dfrac{2}{2^n}$.

Summing up, we have:

$$\Pr[bad = \mathsf{true}] = \sum_{i=2}^{Q} \frac{1}{2^{s(i-1)}} \sum_{T_1,\ldots,T_{i-1}} \sum_{j=1}^{i-1} \frac{2}{2^n} = \sum_{i=2}^{Q} \frac{i-1}{2^{n-1}} = \frac{Q(Q-1)}{2^n}.$$

□

Now we introduce the standard UF-CMA security notion for nonce-based MAC-schemes and obtain the UF-CMA-security bound for the $\mathsf{MGM2\text{-}MAC}$ scheme.

Definition 5. *For a MAC-scheme* MAC *the advantage of a* UF-CMA-*adversary* $\mathcal{A}$ *is defined as follows:*

$$\mathsf{Adv}_{\mathsf{MAC}}^{\text{UF-CMA}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}_{\mathsf{MAC}}^{\text{UF-CMA}}(\mathcal{A}) \to 1\right],$$

where experiment $\mathbf{Exp}_{\mathsf{MAC}}^{\text{UF-CMA}}(\mathcal{A})$ *is defined below:*

```
$\mathbf{Exp}_{\mathsf{MAC}}^{\text{UF-CMA}}(\mathcal{A})$
    $K \xleftarrow{\$} \mathsf{MAC.Gen}(\,)$
    $sent \leftarrow \emptyset$
    $win \leftarrow \mathsf{false}$
    $\mathcal{A}^{Tag,Verify}(\,)$
    return $win$

Oracle $Tag(N, M)$
    if $(N, M) \in sent$:
        return $\perp$
    $T \leftarrow \mathsf{MAC.Tag}(K, N, M)$
    $sent \leftarrow sent \cup \{(N, M)\}$
    return $T$

Oracle $Verify(N, M, T)$
    $res \leftarrow \mathsf{MAC.Vf}(K, N, M, T)$
    if $res \wedge ((N, M) \notin sent)$:
        $win \leftarrow \mathsf{true}$
    return $res$
```

Using Proposition 7.3 [7] and Lemma 1 we obtain the following result.

Corollary 1. *For any* UF-CMA-*adversary* $\mathcal{A}$*, making at most* Q_T *queries to the* Tag *oracle and at most* Q_V *queries to the* $Verify$ *oracle:*

$$\mathsf{Adv}_{\mathsf{MGM2\text{-}MAC}[r,s]}^{\text{UF-CMA}}(\mathcal{A}) \leqslant \frac{Q(Q-1)}{2^n} + \frac{Q_V}{2^s},$$

where $Q = Q_T + Q_V$*.*

5.1.2 Security of MGM2 with random function

Lemma 2. *For any* MRAE-int-*adversary* $\mathcal{A}$*, making at most* Q_E *queries to the Encrypt oracle and at most* Q_D *queries to the Decrypt oracle, there exists a* UF-CMA-*adversary* $\mathcal{B}$*, making at most* Q_E *queries to the* Tag *oracle and at most* Q_D *queries to the* $Verify$ *oracle, such that*

$$\mathsf{Adv}_{\mathsf{MGM2}[Func(n),r,s]}^{\text{MRAE-int}}(\mathcal{A}) \leqslant \mathsf{Adv}_{\mathsf{MGM2\text{-}MAC}[r,s]}^{\text{UF-CMA}}(\mathcal{B})$$

Proof. Let construct an adversary $\mathcal{B}$, that uses the adversary $\mathcal{A}$ as a black box. The adversary $\mathcal{B}$ (see Figure 4) intercepts the queries of the adversary $\mathcal{A}$ and process them by itself using its own oracles. For encryption/decryption $\mathcal{B}$ implements lazy sampling for ρ''. For tag generation/tag verification the adversary $\mathcal{B}$ implements the padding procedure and send the appropriate query to its oracles.

Note that the adversary $\mathcal{B}$ simulates for $\mathcal{A}$ exactly the experiment $\mathbf{Exp}_{\mathsf{MGM2}[Func(n),r,s]}^{\text{MRAE-int}}$. Indeed, since for $\mathsf{MGM2}[Func(n), r, s]$ the inputs to the random function in case of 1) tag generation, 2) computing values H_i and 3) computing values Γ_i are different (because of fixed bits in inputs), using one random function is indistinguishable from using three independent random functions ρ, ρ', ρ'' for these three cases. Also, note that messages M, formed by $\mathcal{B}$, satisfy conditions for message set of $\mathsf{MGM2\text{-}MAC}[r, s]$.

If the adversary $\mathcal{A}$ forges, then the adversary $\mathcal{B}$ also forges in $\mathbf{Exp}_{\mathsf{MGM2\text{-}MAC}[r,s]}^{\text{UF-CMA}}$. Indeed, if $\mathcal{A}$ makes non-trivial valid query (N, A, C, T) to the *Decrypt* oracle, then the adversary makes $\mathcal{B}$ corresponding non-trivial query $(N, M = A\|0^a\|C\|0^c\|len, T)$ to the $Verify$ oracle. □

```
$\mathcal{B}_{\mathcal{A}}^{Tag,Verify}$
$\rho'' \xleftarrow{\mathcal{U}} Func(n)$    // lazy sampling
return $\mathcal{A}^{SEncrypt,SDecrypt}(\,)$

$SEncrypt(N, A, P)$
$h \leftarrow |A|_n, q \leftarrow |P|_n$
........... Encryption ...........
for $i = 1 \dots q$ do:
    $\Gamma_i \leftarrow \rho''(N \| 00 \| \mathrm{str}_{n-r-2}(i-1))$
$C \leftarrow P \oplus \mathrm{msb}_{|P|}(\Gamma_1 \| \dots \| \Gamma_q)$
.............Padding.............
$a \leftarrow n|A|_n - |A|$
$c \leftarrow n|C|_n - |C|$
$len \leftarrow \mathrm{str}_{n/2}(|A|) \| \mathrm{str}_{n/2}(|C|)$
$M \leftarrow A \| 0^a \| C \| 0^c \| len$
......... Tag Genetation .........
$T \leftarrow Tag(N, M)$
return $(C, T)$
```

```
Oracle $SDecrypt(N, A, C, T)$
$h \leftarrow |A|_n, q \leftarrow |C|_n$
.............Padding.............
$a \leftarrow n|A|_n - |A|$
$c \leftarrow n|C|_n - |C|$
$len \leftarrow \mathrm{str}_{n/2}(|A|) \| \mathrm{str}_{n/2}(|C|)$
$M \leftarrow A \| 0^a \| C \| 0^c \| len$
......... Tag Verification .........
if $Verify(N, M, T) = \mathsf{false}$:
    return $\bot$
........... Decryption ...........
for $i = 1 \dots q$ do:
    $\Gamma_i \leftarrow \rho''(N \| 00 \| \mathrm{str}_{n-r-2}(i-1))$
$P \leftarrow C \oplus \mathrm{msb}_{|C|}(\Gamma_1 \| \dots \| \Gamma_q)$
return $P$
```

Figure 4: Adversary $\mathcal{B}$

5.2 Confidentiality

Theorem 2. *For any* CPA-res-*adversary $\mathcal{A}$, making at most Q_1 queries to the O_1 oracle and at most Q_2 queries to the O_2 oracle, where the total block-length of associated data in all queries is at most σ_A and the total block-length of plaintext and ciphertexts in all queries is at most σ_P,*

$$\mathsf{Adv}^{\text{CPA-res}}_{\mathsf{MGM2}[Perm(n),r,s]}(\mathcal{A}) \leqslant \frac{\sigma^2}{2^n} + \frac{Q(Q-1)}{2^{n-1}}, \tag{3}$$

where $Q = Q_1 + Q_2$ and $\sigma = 2\sigma_P + \sigma_A + 2Q$.

Proof. Firstly, we apply PRP-PRF switching lemma [12] to replace $Perm(n)$ by $Func(n)$ (this gives us the term $\frac{\sigma^2}{2^n}$ in the bound), and then we obtain the CPA-res-security bound for $\mathsf{MGM2}[Func(n), r, s]$.

The security bound for $\mathsf{MGM2}[Func(n), r, s]$ is obtained in the same way as in the proof of Theorem 1. Indeed, ciphertexts C, received from the O_1 oracle, are absolutely indistinguishable from uniform random strings since the inputs to the uniform random function ρ used to produce Γ_i are unique. The indistinguishability of the tags T, received from the O_1 oracle, from uniform random strings is estimated by constructing two PRF-adversaries for $\mathsf{MGM2\text{-}MAC}$ that uses CPA-res-adversary as a black box. Therefore, $\mathsf{Adv}^{\text{CPA-res}}_{\mathsf{MGM2}[Func(n),r,s]}(\mathcal{A}) \leqslant \frac{Q(Q-1)}{2^{n-1}}$. □

6 Conclusion

In the current paper we introduce the modification of the MGM mode — the $\mathsf{MGM2}$ mode. For this mode we obtain the security bounds for non-standard notions MRAE-int and CPA-res, allowing the adversary to repeat nonces. In comparison with the original mode, the security proof appears to be rather simple and short.

In the future work we are going to develop a SIV-construction (see [15]) of the MGM2 mode to achieve MRAE-conf-security. Also we are going to incorporate re-keying mechanisms in the MGM2 mode to achive new security properties like leakage-resilience and increase key lifetime.

References

[1] Akhmetzyanova L., Alekseev E., Karpunin G., Nozdrunov V. *Security of Multilinear Galois Mode (MGM).*, IACR Cryptology ePrint Archive 2019, p. 123, 2019.

[2] Akhmetzyanova L., Alekseev E., Smyshlyaev S., Oshkin I. (2020) *On Internal Re-keying.* In: van der Merwe T., Mitchell C., Mehrnezhad M. (eds) Security Standardisation Research. SSR 2020. Lecture Notes in Computer Science, vol 12529. Springer, Cham. https://doi.org/10.1007/978-3-030-64357-7_2

[3] Andreeva E., Bogdanov A., Luykx A., Mennink B., Mouha N., Yasuda K. (2014) *How to Securely Release Unverified Plaintext in Authenticated Encryption.* In: Sarkar P., Iwata T. (eds) Advances in Cryptology – ASIACRYPT 2014. ASIACRYPT 2014. Lecture Notes in Computer Science, vol 8873. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-45611-8_6

[4] Ashur T., Dunkelman O., Luykx A. *Boosting authenticated encryption robustness with minimal modifications* //Annual International Cryptology Conference. – Springer, Cham, 2017. – C. 3-33.

[5] Davide Bellizia and Olivier Bronchain and Gaëtan Cassiers and Vincent Grosso and Chun Guo and Charles Momin and Olivier Pereira and Thomas Peters and François-Xavier Standaert, *Mode-Level vs. Implementation-Level Physical Security in Symmetric Cryptography: A Practical Guide Through the Leakage-Resistance Jungle*, Cryptology ePrint Archive, Report 2020/211, 2020, https://eprint.iacr.org/2020/211

[6] Bellare M., Namprempre C. *Authenticated encryption: Relations among notions and analysis of the generic composition paradigm* //International Conference on the Theory and Application of Cryptology and Information Security. – Springer, Berlin, Heidelberg, 2000. – C. 531-545.

[7] Bellare M., Rogaway P. *Introduction to modern cryptography* //Ucsd Cse. – 2005. – T. 207. – C. 207.

[8] Bellare M., Rogaway P. *The Security of Triple Encryption and a Framework for Code- Based Game-Playing Proofs* // LNCS, Advances in Cryptology - EUROCRYPT 2006, 4004, ed. Vaudenay S., Springer, Berlin, Heidelberg, 2006.

[9] Bernstein, D.J.: *Stronger Security Bounds for Permutations* (2005), http://cr.yp.to/papers.html (accessed on May 31, 2012)

[10] John Black, Phillip Rogaway, and Thomas Shrimpton. 2002.*Encryption-Scheme Security in the Presence of Key-Dependent Messages.* In Revised Papers from the 9th Annual International Workshop on Selected Areas in Cryptography (SAC '02). Springer-Verlag, Berlin, Heidelberg, 62–75.

[11] CAESAR competion. `https://competitions.cr.yp.to/caesar.html`

[12] D. Chang and M. Nandi, *A Short Proof of the PRP/PRF Switching Lemma* // IACR ePrint Archive, 2008, Report 2008/078, https://eprint.iacr.org/2008/078.

[13] Federal Agency on Technical Regulating and Metrology, *Information technology. Cryptographic data security. Authenticated encryption block cipher operation modes*, R 1323565.1.026-2019, 2019.

[14] Federal Agency on Technical Regulating and Metrology, *Information technology. Cryptographic data security. Cryptographic algorithms accompanying the use of block ciphers*, R 1323565.1.017—2018, 2018.

[15] Gueron S., Lindell Y. *GCM-SIV: full nonce misuse-resistant authenticated encryption at under one cycle per byte* //Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. – 2015. – C. 109-119.

[16] Hoang V.T., Krovetz T., Rogaway P. (2015) *Robust Authenticated-Encryption AEZ and the Problem That It Solves.* In: Oswald E., Fischlin M. (eds) Advances in Cryptology – EUROCRYPT 2015. EUROCRYPT 2015. Lecture Notes in Computer Science, vol 9056. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-46800-5_2

[17] Kurochkin A., Fomin D. *MGM Beyond the Birthday Bound* // 8th Workshop on Current Trends in Cryptology (CTCrypt 2019).

[18] Rogaway P. (2004) *Nonce-Based Symmetric Encryption.* In: Roy B., Meier W. (eds) Fast Software Encryption. FSE 2004. Lecture Notes in Computer Science, vol 3017. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-25937-4_22

[19] Rogaway P., Shrimpton T. *A provable-security treatment of the key-wrap problem* //Annual International Conference on the Theory and Applications of Cryptographic Techniques. – Springer, Berlin, Heidelberg, 2006. – C. 373-390.

[20] Shrimpton T.: *A Characterization of Authenticated-Encryption as a Form of Chosen-Ciphertext Security.* In: Cryptology ePrint Archive, Report 2004/272 (2004).

[21] Smyshlyaev, S., Nozdrunov, V., Shishkin, V., and E. Smyshlyaeva *Multilinear Galois Mode (MGM)* // 2019, <https://tools.ietf.org/html/draft-smyshlyaev-mgm-17>

[22] Shrimpton T., Terashima R. S. *A modular framework for building variable-input-length tweakable ciphers* //International Conference on the Theory and Application of Cryptology and Information Security. – Springer, Berlin, Heidelberg, 2013. – C. 405-423.