On internal re-keying

Evgeny K. Alekseev, Liliya R. Akhmetzyanova, Igor B. Oshkin, and Stanislav V. Smyshlyaev

CryptoPro LLC, Moscow, Russia
{lah,alekseev,oshkin,svs}@cryptopro.ru

Abstract. In this paper we introduce a classification of existing re-keying-based approaches to increase the security of block cipher operation modes. We introduce the concepts of external and internal re-keying putting the focus on the second one. Whereas the external re-keying approach is widely used and provides the mechanism of key usage control on a message stream processing level, the internal re-keying approach is the first known mechanism providing such a control on a single message processing level. These approaches can be applied completely independently. The internal re-keying approach was already applied to the CTR encryption mode and yielded the CTR-ACPKM mode. This mode is currently being standardized in ISO and in IETF/IRTF (CFRG).
In the current paper we apply the internal re-keying approach to the well-known GCM authenticated encryption mode. The main results of this paper are a new internally re-keyed GCM-ACPKM mode and its security bounds. The proposed mode is also passing through the last formal standardization stages in IETF (CFRG). We estimate the security of the GCM-ACPKM mode respecting standard security notions. We compare both security and performance of the GCM-ACPKM and GCM modes. The results show that changing GCM mode by integrating the ACPKM internal re-keying procedure increases security, significantly extending the lifetime of a key with a negligible loss in performance. Also we show how the re-keying approaches could increase the security of TLS 1.3 cipher suites.

Key words: re-keying, block cipher modes, AEAD, GCM, provable security

1 Introduction

One of the main problems related to secure functioning of any cryptosystem is the control of lifetimes of keys. Regarding symmetric keys the main concern is constraining the key exposure by limiting the maximal amount of data processed with one key. The restrictions can derive either from combinatorial properties of the used cipher modes of operation (e.g. most modes of operation are secure up to the birthday paradox bound [4]), or from resisting certain specific cryptographic attacks on the used block cipher, e.g. differential [12] or linear cryptanalysis [22]), including side-channel attacks [13,14,32] (in this case the restrictions are the most severe ones). The adversary's opportunity to obtain an

essential amount of data processed with the same key leads not only to theoretical but also to practical vulnerabilities (see, e.g., [10,32]). Thus, when the total length of data processed with the same key reaches a threshold value, certain procedures on encryption keys are needed. This leads to several operating limitations, e.g. processing overhead caused by the new keys generation and the impossibility of long message processing.

In the context of high-level protocols, the most obvious way to overcome the above-mentioned limitations is a regular session key renegotiation. However, such an operation assumes interrupting payload transmissions, sending additional service-based data in the channel, using random number generators and even public key cryptography. Frequent key renegotiation is undesirable, since this would drastically reduce the total performance.

Another way is to deterministically transform a previously negotiated key. One mechanism, and the most common one in practice, is a key diversification (e.g. key hierarchy [26] and HKDF [30]). As soon as a given amount of whole messages is processed, the session key should be updated. Another mechanism, called key meshing [29], assumes the key transformation during separate message processing, which starts with the same key each time.

1.1 Related Work

Key Diversification. A key diversification scheme treats a shared key as a master key, which is never used directly for data processing. As soon as a given amount of whole messages is processed, a new session key should be derived (e.g. $2^{24.5}$ records in TLS 1.3 for a certain safety margin [30]).

Key diversification was addressed by Abdalla and Bellare in [1] — a motivation was given, criteria for such mechanisms and concrete security bounds were obtained, and two schemes were proposed (parallel and serial ones). One of the main points of this work is that the «satisfactory» key diversification technique allows you to essentially increase the key lifetime as compared to a direct usage of a key for data processing. The obtained security bound of the key diversified mode of operation allows to separately analyze the re-keying technique and the base mode of operation. Such a clear separation of security analysis is the definitive advantage of this mechanism. Another feature of this approach is a forward security property, as discussed in [8].

Key Meshing. Another mechanism to increase the key lifetime was presented for the first time in [29] and is called «CryptoPro Key Meshing» (CPKM). This solution assumes that each message is processed starting from the initially negotiated key, which is transformed as soon as a given relatively small amount of data is processed. Such a transformation does not require any additional secret values and uses the initial key directly for data processing. The security of this mechanism had not been analyzed for a long time until the security bound for the re-keyed CTR encryption mode was obtained in [2].

An operating disadvantages of CPKM is the usage of the decryption function. This can double the code size for some block cipher modes and, consequently,

reduce the performance. Another disadvantage is that the probability of trivial-recovering the derived key is nonzero.

To negate the disadvantages mentioned above the new ACPKM (advanced CPKM) re-keying technique was proposed in CTCrypt 2018 for increasing the lifetime of keys used in CTR mode. This technique uses only the encryption function and the probability of trivial-recovering the derived keys is zero. The paper [3] contains the analysis of the internally re-keyed CTR-ACPKM mode for the standard IND-CPA notion. The obtained security bound shows that the usage of ACPKM increases the key lifetime compared to the base CTR mode.

The internally re-keyed CTR-ACPKM mode is passing through the last formal standardization stages in IETF (CFRG) [38] and is being standardized in ISO. This mechanism is also used in the TLS 1.2 cipher suites [39] which have been recently added by IANA in the TLS Cipher Suite Registry.

1.2 Our Contribution

In the current paper we introduce concepts of internal and external re-keying approaches — generalizations of key diversification and key meshing mechanisms. We discuss the advantages and disadvantages of both the internal and external re-keying approaches, the relationship between them and their application fields. We show that the internal re-keying approach can be treated not as an alternative of the external approach analyzed in [1] but rather as its powerful extension. It allows us to avoid such an operating problem as the message length limitation in the case when the key lifetime is rather strict [32]. Using the example of TLS 1.3 we show that the composition of these approaches essentially increases the key lifetime that allows to securely process the maximum possible amount of records (2^{64}).

In the current paper we integrate the ACPKM key update procedure into the well-known GCM authenticated encryption mode. The main results of this paper are a new internally re-keyed GCM-ACPKM mode and its security bounds respecting both Privacy and Authenticity notions. We show that the ACPKM re-keying improves not only privacy, that was already shown in [3] for the quite similar CTR encryption mode, but also authenticity.

The ACPKM technique is chosen with performance aspects in mind — the key transformation needs relatively small amount of encryption operations, which code is already initialized and presented in the cache. We compare the performance of the base GCM mode and the internally re-keyed GCM-ACPKM mode with different section sizes. We consider base block cipher AES-256 and AES-128 with hardware support. Slowdown due to using the ACPKM technique does not exceed 3% for any section size.

1.3 Organization

The rest of this paper is organized as follows. Section 2 is dedicated to preliminaries. In Section 3 we describe internal and external re-keying concepts and

discuss their properties and application fields. In Section 4 we recall the description of the GCM mode and introduce an ACPKM re-keying technique for this mode. In subsections we provide the main theorems on the security of the internally re-keyed GCM-ACPKM mode and then carry out the comparative analysis with the GCM mode basing on known security bounds and operational properties. In Section 5 we consider the practical significance of considered re-keying techniques and show by the example of TLS 1.3 that the hybrid re-keying technique allows to significantly increase the key lifetime. Finally, in the Appendix we prove the main theorems.

2 Preliminaries

By $\{0,1\}^u$ we denote the set of u-component bit strings and by $\{0,1\}^*$ we denote the set of all bit strings of finite length. Let ε be the empty string and 0^u be the string, consisting of u zeros. For bit strings U and V we denote by $U\|V$ their concatenation. We denote by $U_{(i)}$ the i-th bit, $i \in \{0,\dots,n-1\}$, of the string $U = U_{(0)}\|\dots\|U_{(n-1)} \in \{0,1\}^n$. Let $|U|$ be the bit length of the string U.

For a bit string U and a positive integer l such that $l \leqslant |U|$, $\mathrm{msb}_l(U)$ ($\mathrm{lsb}_l(U)$) denotes the leftmost (rightmost) l bits of U. For nonnegative integers l and $i < 2^l$, let $\mathrm{str}_l(i)$ be the l-bit binary representation of i with the least significant bit on the right. For a nonnegative integer l and a bit string $U \in \{0,1\}^l$, let $\mathrm{int}(U)$ be the integer i such that $\mathrm{str}_l(i) = U$.

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. Throughout this paper, we fix a blockcipher E with the block size $n = 128$. If the value s is chosen from a set S uniformly at random, then we denote $s \in_{\mathcal{U}} S$.

For a bit string U we denote by $U_i \in \{0,1\}^n$, $1 \leqslant i \leqslant \lceil |U|/n \rceil - 1$, and $U_{\lceil |U|/n \rceil} \in \{0,1\}^h$, $h \leqslant n$, such strings that $U = U_1\|U_2\|\dots\|U_{\lceil |U|/n \rceil}$ and call them «blocks» of the string U. We denote by $|U|_u = \lceil |U|/u \rceil$ the length of the string U in u-bit blocks.

We model an adversary using a probabilistic algorithm that has access to one or more oracles. Denote by $\mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2,\dots}$ an adversary $\mathcal{A}$ that interacts with oracles $\mathcal{O}_1, \mathcal{O}_2, \dots$ by making queries. Notation $\mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2,\dots} \Rightarrow 1$ means that the algorithm $\mathcal{A}$, after interacting with oracles $\mathcal{O}_1, \mathcal{O}_2, \dots$, outputs 1. The resources of $\mathcal{A}$ are measured in terms of 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.

3 Block Cipher Modes and Re-keying

A block cipher is a family of permutations, which on its own do not provide such application-level security properties as integrity, confidentiality or authenticity

(see, e.g., [6]). The cipher is usually used as a base function for constructing other schemes or protocols that solve the above-mentioned cryptographic challenges. The security of such constructions is usually proven under assumption that the block cipher is secure. In the paradigm of practice-oriented provable security (see [7]) we should quantify the security as a function of the used primitive security for given notions.

The above-mentioned cryptographic challenges can be solved with the use of «block cipher modes of operation». The modes define how to use the underlined block cipher to process messages which can consist of more than one block. Thus, a single key can be used for processing a large number of blocks. To achieve the sufficient security level this number should be limited. The main reasons for this are pointed out in Introduction.

Re-keying is an approach, which is widely used to overcome the above-mentioned limitation for block cipher modes of operation. The main idea behind this approach is as follows: the data is processed with a sequence of keys derived from an initial «truly» random key.

In this section we introduce the classifications of existing re-keying approaches (*internal* and *external*) and of accompanying key update techniques (with a *master key* and without a master key). These classifications are also described in [38]. Two out of four possible combinations were mentioned in Introduction: external re-keying with master key (key diversification) and internal re-keying without master key (key meshing). In this section we consider the common approaches and discuss their properties, advantages and disadvantages. We put the focus on the internal re-keying approach, since its properties were not considered carefully.

3.1 External Re-keying

The main concept of this approach is as follows: a key, derived according to a certain key update technique, is intended to process the fixed number of separate messages, after which the key should be updated. Using external re-keying jointly with the block cipher mode of operation does not change the mode internal structure, therefore we call this approach «external re-keying». The main idea behind it is presented in Fig. 1.

Doubtless advantage of external re-keying is the possibility to explicitly use the obtained security bounds for the base mode to quantify security of the corresponding externally re-keyed mode (see [1]).

External re-keying is proposed to be performed each time a given amount of messages is processed. However, the key lifetime is defined by the total length of the processed messages and not by their number. In order to satisfy a certain requirement on the key lifetime limitation, one should fix the maximal message length. If this requirement is restrictive enough (e.g, to resist side-channel attacks), it leads to some problems. Thus, long message processing requires additional fragmentation. Such a fragmentation can lead to frequent re-using a random number generator for generating new IVs (e.g., in the case of data processing in the CBC or CFB modes), that significantly affects the performance.

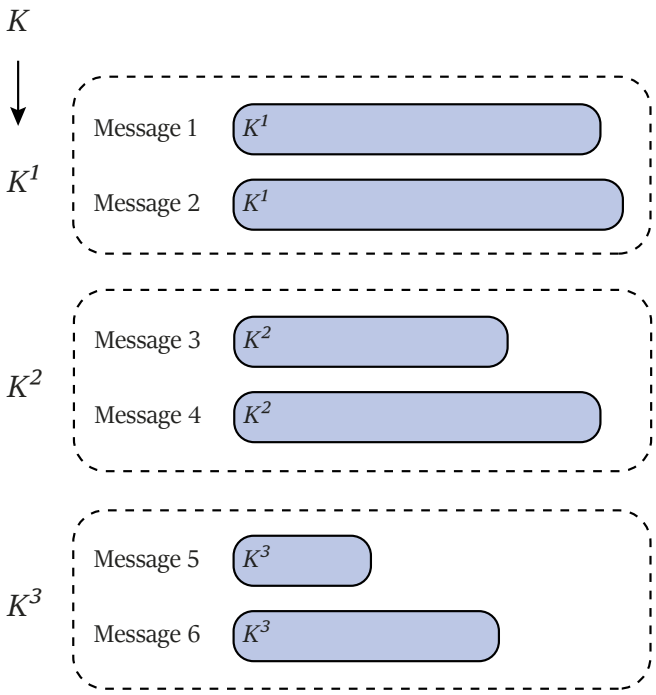

Fig. 1. Idea behind external re-keying. For simplicity, a case with only six messages is shown, and the key is changed after every two messages is processed. Here every two messages are processed under the corresponding key K^i that is produced from the initial key K.

External re-keying is recommended for the usage in protocols, which process quite small messages, since the maximum gain in increasing the key lifetime is achieved by increasing the number of messages.

3.2 Internal Re-keying

The internal re-keying approach modifies the base mode of operation in such a way that each message is processed starting from the same key, which is changed using the certain key update technique during processing of the current message. It is integrated into the base mode of operation and changes its internal structure, therefore we call it «internal re-keying». The main idea behind internal re-keying is presented in Fig. 2.

The concept of internal re-keying is inseparable from the concept of «section». A section is the string, which consists of all input cipher blocks processed using the same key, which we will call a «section key». In order to fully define a section for a certain mode of operation there is a need to determine what section key

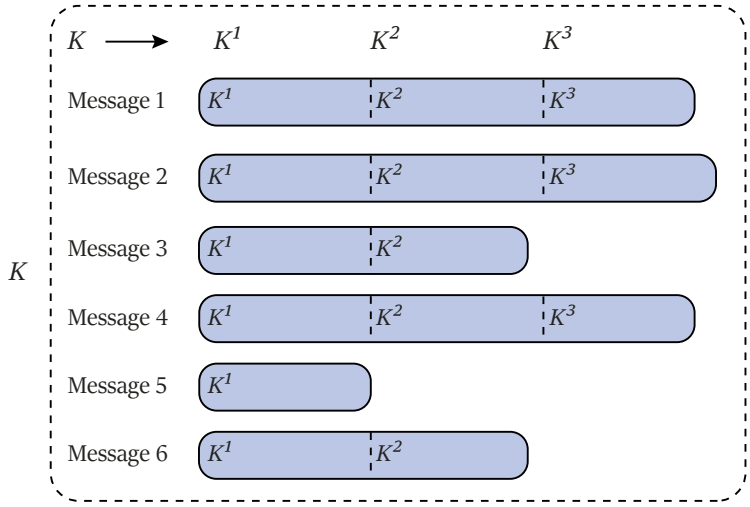

Fig. 2. Idea behind internal re-keying. For simplicity, a case with only six messages is shown. Here each message is processed starting from the first derived key K_1. This key is changed each time a data section of fixed length has been processed.

will be used to process a certain input block. Therefore, for correct processing we need to define the order on all input blocks for the cipher. For several simple encryption modes (CTR, CBC, OFB) the order can be defined trivially — in accordance with the case of consequent message processing. However, for the other modes of operation, particularly for AEAD modes, there are too many ways to define a section in common. Indeed, for such modes the blocks processing order for encryption can differ from the order for decryption, moreover, blocks for plaintext encryption and tag computation can be processed in parallel. So, we stress that internal re-keying should be determined in each specific case with respect to security and operational features of the mode.

Obviously a section size is bounded by the key lifetime, which depends on the combinatorial properties of the used operation mode or existing attacks on the base block cipher including side-channel attacks. A certain section size can be chosen optionally for different cases, because it affects the operating properties and limits the number of messages: the larger the section size, the faster message processing (see Section 4.5), but the smaller the section size, the greater the number of separate processed messages.

Security analysis of internally re-keyed modes leads to the analysis of the abstract modes where section keys are chosen independently at random. For standard encryption modes of operation the security of corresponding modes with random keys can be easily analyzed, using the technique of hybrid argu-

ment. To obtain security bounds for more complicated modes (AEAD, MAC types), where sections are not consistent, their base proof should be rethought.

Summing up the above-mentioned issues, we can conclude that internal re-keying should be treated as a technique, which produces a new set of the re-keyed modes of operation.

Internal re-keying mechanisms are recommended to be used in protocols, which process large single messages (e.g., CMS messages), since the maximum gain in increasing the key lifetime is achieved by increasing the length of a message, while it provides almost no increase in the number of messages, which can be processed with a single key.

3.3 Composition of Internal and External Re-keying

Both external re-keying and internal re-keying have their own advantages and disadvantages discussed above. For instance using external re-keying can essentially limit the message length, while in the case of internal re-keying the section size, which can be chosen the maximal possible for operational properties, limits the number of separate messages. There is no technique, which is more preferable, because the choice of technique can depend on certain protocol features. For example, for protocols, which allow out-of-order delivery and lost records (e.g., [34,35]), external re-keying is preferable to be used, but if a protocol assumes processing a significant amount of ordered records, which can be considered as a single data stream (e.g., [36,37]), internal re-keying is better suited.

In order to negate the mentioned disadvantages, the composition of external and internal re-keying approaches (see Figure 3) can be applied. It can be easily done due to the concepts of external and internal re-keying. Indeed, external re-keying controls key lifetime on the protocol level (a message stream) and internal re-keying controls key lifetime on the block cipher mode level (a single message). This allows to compose these techniques independently.

3.4 Key Update Techniques

In the previous subsections we discussed the approaches to data processing with a sequence of derived keys. The current subsection is dedicated to the several techniques of producing such keys.

We distinguish two key update techniques: with a master key and without a master key. The first one has the following property: a shared initial key is never used directly for data processing but is used only for subkey derivation. Using this technique in the internal and external ways allows to combine the arbitrary key update function with the arbitrary mode of operation and to bound security of the construction, separately analyzing the used components:

– for external re-keying — the key update technique and the base mode of operation [1];

– for internal re-keying — the key update technique and the abstract mode with random section keys.

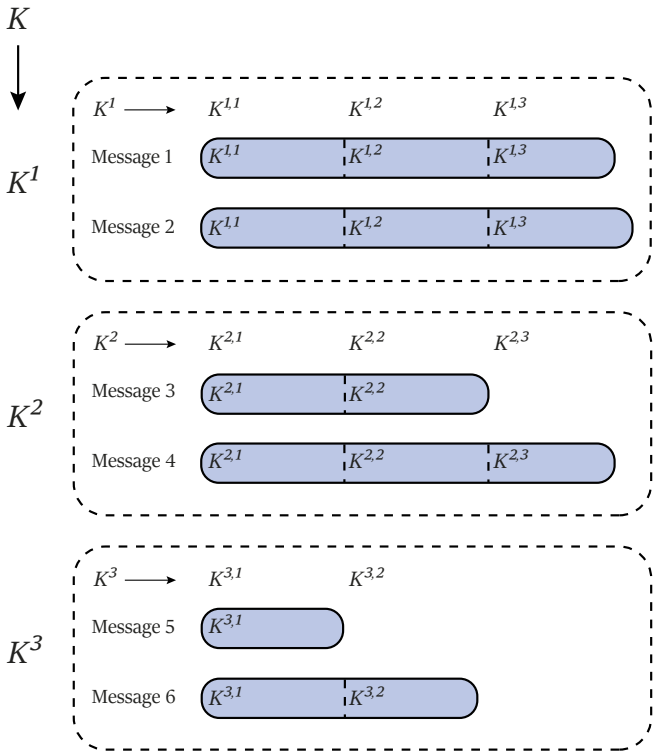

Fig. 3. Composition of Internal and External Re-keying. For simplicity, a case with only six messages is shown. Here K^1, K^2, K^3 are diversified from an initial key K. Then each K^i is used for processing two messages in the internally re-keyed mode.

Another advantage is the possibility to protect keys for some pieces of data even in the case when keys for the other pieces are compromised.

The second technique directly uses the initial key as the first key for data processing, and each next key is computed from the previous one. It seems to be mostly useful in the case when the total amount of data for an initial key is not known beforehand: we will not lose performance on useless operations if the data is rather short, and we will not lack security when it occurs to be large. We will derive new keys only when they are needed. As distinct from the first technique we cannot consider the concrete key update function separately from the mode of operation. In order to illustrate the importance of considering the

key update function and the mode of operation as a whole, we will show the following example.

Consider the CBC-MAC mode providing message authenticity. We give a rough specification of CBC-MAC: for the input message $M = M_1\|\ldots\|M_l$, $l = |M|_n$ the authentication tag T is computed as follows:

$$T = E_K(E_K(\ldots E_K(E_K(M_1) \oplus M_2)\ldots) \oplus M_l).$$

CBC-MAC is known to be provably secure up to the birthday paradox bound when applied to prefix free message space [5].

Suppose $k = n$ for the used block cipher and message length be at least 2 blocks. Let internally extend the base mode with the following key update function:

$$K^1 = K, \qquad K^{i+1} = E_{K^i}(C_0) \oplus E_{K^i}(C_1), \quad i = 2, \ldots,$$

where $K \in \{0,1\}^k$ is the initially shared key, C_0, $C_1 \in \{0,1\}^n$ are arbitrary different constants. Let the section size be at least 2 blocks.

Due to the message length limitations we cannot trivially find the results of the constants C_0, C_1 encryption. However, this technique does not increase the security of the base mode, because there is the attack, which allows to find out the key of the second section with probability 1 and $2 \cdot 2^{n/2}$ pairs (M, T) for chosen M, $|M|_n = 2$. The adversary requests authentication tags for $2^{n/2}$ messages $C_0\|R_1\|0^{n/2}$ and $2^{n/2}$ messages $C_1\|0^{n/2}\|R_2$, where R_1 and R_2 take all strings from $\{0,1\}^{n/2}$. Note that all messages are prefix-free. Obviously, there is the collision $T_1 = T_2$ with probability 1, where $T_1 = E_K(E_K(C_0) \oplus R_1\|0^{n/2})$ and $T_2 = E_K(E_K(C_1) \oplus 0^{n/2}\|R_2)$. Thus, the next section key $K^2 = E_K(C_0) \oplus E_K(C_1)$ is $R_1\|R_2$. The revealed next section key allows to trivially forge the tag for long (more than section) messages. The similar attack can be applied to the OMAC mode (see [16,17,25]).

We may conclude that the proposed key update function is «bad», but for such encryption modes as CBC, OFB, CFB the considered attack is not applicable because of using random initialization vector.

Therefore, to be convinced that the proposed key update function is «good», we should provide the security proof in both cases of external and internal re-keying.

4 GCM and GCM-ACPKM modes

In the current section we introduce an internally re-keyed authenticated encryption with associated data (AEAD) mode called GCM-ACPKM.

4.1 Description

Firstly recall the description of the GCM mode according to [23].

GCM. Denote by $\mathrm{GCM}_{E,\tau}$ the GCM mode that uses a blockcipher E with the block size $n = 128$ and the positive integer $64 \leqslant \tau \leqslant n$, denoted to a tag size, as parameters.

Before considering the GCM mode in details define the auxiliary functions. For bit strings A, B of arbitrary lengths and $H \in \{0,1\}^n$ we have the function

$$\mathrm{GHASH}_H(A,B) = \sum_{i=1}^{m} X_i \cdot H^{m+1-i},$$

where «$\sum$» and «$\cdot$» are addition and multiplication in $GF(2^n)$, and the string X is computed as follows. Let $a = n\cdot|A|_n - |A|$, $b = n\cdot|B|_n - |B|$, $m = |A|_n + |B|_n + 1$, then $X = X_1\| \ldots \|X_m = A\|0^a\|B\|0^b\|\mathrm{str}_{n/2}(|A|)\|\mathrm{str}_{n/2}(|B|)$. If $A = \varepsilon$ then $X = X_1\| \ldots \|X_m = B\|0^b\|\mathrm{str}_n(|B|)$.

Let $\mathrm{incr} : \{0,1\}^n \to \{0,1\}^n$ be the encoding function, which takes the input $I \in \{0,1\}^n$, and outputs the string

$$\mathrm{msb}_{n-32}(I)\|\mathrm{str}_{32}(\mathrm{int}(\mathrm{lsb}_{32}(I)) + 1 \bmod 2^{32}).$$

$\underline{\mathrm{GCTR}_E(K,I,X)}$

1: $I_0 = I$
2: for $i = 1$ to $|X|_n$ do
3: $\quad I_i = \mathrm{incr}(I_{i-1})$
4: $\quad G_i = E_K(I_i)$
5: $Y = X \oplus \mathrm{msb}_{|X|}\left(G_1\| \ldots \|G_{|X|_n}\right)$
6: **return** Y

$\underline{\mathrm{GCM}_{E,\tau}.\mathsf{Encrypt}(K,IV,A,M)}$

1: $I_0 = IV\|0^{31}1$
2: Ciphertext computation:
3: $\quad C = \mathrm{GCTR}_E(K, I_0, M)$
4: Tag computation:
5: $\quad H = E_K(0^n)$, $Z = E_K(I_0)$
6: $\quad T = \mathrm{msb}_\tau\left(\mathrm{GHASH}_H(A,C) \oplus Z\right)$
7: **return** (C,T)

$\underline{\mathrm{GCM}_{E,\tau}.\mathsf{Decrypt}(K,IV,A,C,T)}$

1: $I_0 = IV\|0^{31}1$
2: Plaintext computation:
3: $\quad M = \mathrm{GCTR}_E(K, I_0, C)$
4: Tag verification:
5: $\quad H = E_K(0^n)$, $Z = E_K(I_0)$
6: $\quad T' = \mathrm{msb}_\tau\left(\mathrm{GHASH}_H(A,C) \oplus Z\right)$
7: $\quad$ if $T = T'$ then
8: $\quad\quad$ **return** M
9: $\quad$ else **return** $\perp$

Fig. 4. Authenticated encryption and decryption in the GCM Mode for the nonce length restricted to 96 bits.

Authenticated Encryption in the GCM *Mode.* A processed message for authenticated encryption in the $\mathrm{GCM}_{E,\tau}$ mode is (IV, A, M), where IV is a nonce, $0 \leqslant |IV| \leqslant 2^{n/2} - 1$, A is an associated data, $0 \leqslant |A| \leqslant 2^{n/2} - 1$, and M is a

plaintext, $0 \leqslant |M| \leqslant n(2^{32}-2)$. The result of GCM encryption under a key K is a pair (C,T), where $C \in \{0,1\}^{|M|}$ is a ciphertext of M and $T \in \{0,1\}^{\tau}$ is an authentication tag, which are computed as follows:

$$C = M \oplus \mathrm{msb}_{|M|}(E_K(I_1)\| \ldots \|E_K(I_{|M|_n})),$$

$$T = \mathrm{msb}_{\tau}\left(E_K(I_0) \oplus \mathrm{GHASH}_H(A,C)\right).$$

Here $H = E_K(0^n)$, $I_i = \mathrm{incr}(I_{i-1})$, $1 \leqslant i \leqslant |M|_n$, where $I_0 = IV\|0^{31}1$, if $|IV| = 96$, or $I_0 = \mathrm{GHASH}_H(\varepsilon, IV)$, otherwise. The nonces IV are different for different messages processed with the same key K.

Authenticated Decryption in the GCM *Mode.* An input message of authenticated decryption in the $\mathrm{GCM}_{E,\tau}$ mode is (IV,A,C,T), where IV is a nonce, $0 \leqslant |IV| \leqslant 2^{n/2}-1$, A is an associated data, $0 \leqslant |A| \leqslant 2^{n/2}-1$, C is a ciphertext, $0 \leqslant |C| \leqslant n(2^{32}-2)$, and $T \in \{0,1\}^{\tau}$ is an authentication tag. The result of GCM decryption under a key K is a plaintext $M \in \{0,1\}^{|C|}$, if (C,T) is the result of GCM encryption of (IV,A,M), and $\perp$, if there are no plaintexts, satisfying this condition.

Now we introduce the internally re-keyed GCM-ACPKM mode.

GCM-ACPKM. In order to show an idea behind internal re-keying technique more clear, we consider the GCM mode with the nonce length restricted to 96 bits. Another reason for that is in the facts, that many standards require or recommend using GCM with 96-bit nonces for efficiency and the results obtained in [18,28] suggest that restricting GCM to 96-bit nonces is recommended from the provable security perspective as well: there is no the additional term $\frac{32q(\sigma+q)(l_{IV}+1)}{2^n}$, respected to the probability of nonce collision, in the security bound.

Firstly, define the auxiliary function $\varphi_i : \{0,1\}^n \to \{0,1\}$, $\varphi_i(X) = X_{(i)}$, $0 \leqslant i < n$. This function returns the i-th bit of string.

Key updating for the GCM encryption mode is as follows:

$$K^1 = K, \quad K^{i+1} = \mathrm{ACPKM}(K^i) = \mathrm{msb}_k(E_{K^i}(D_1)\| \ldots \|E_{K^i}(D_s)),$$

where $s = \lceil k/n \rceil$, $D_1, \ldots, D_s \in \{0,1\}^n$ are pairwise different arbitrary constants such that $\varphi_{n-32}(D_1) = \ldots = \varphi_{n-32}(D_s) = 1$ are pairwise different and K is an initially shared key.

We denote by $\mathrm{GCM\text{-}ACPKM}_{E,\tau,l}$ the $\mathrm{GCM}_{E,\tau}$ mode of operation that takes the key updating according to the ACPKM technique after each l processed blocks of the plaintext M (without consideration of the associated data A). The internal state (counter) of the mode is not reset for each new section. There is a certain reason for that: in order to protect against a key-collision attack (see [11]), we should provide different input blocks for encryption under different keys. The key for computing values $E_K(I_0)$ and $H = E_K(0^n)$ is not updated and is equal to the initial key. The plaintext length should be at most $2^{31}-2$ blocks.

The structure of the GCM-ACPKM mode with 96-bit nonces IV is such that blocks of the next key never appear in a set of blocks $E_K(I_i)$, where

$1 \leqslant i \leqslant 2^{31} - 2$. This property is provided by the restriction on the plaintext length and the constants $D_1, \ldots, D_s$. Note that the GCM-ACPKM mode with nonces of variable length has not this property and the probability of the trivial breaking the next section key is small but not zero. It is one more reason for considering the 96-bit nonces.

GCTR-ACPKM$_{E,l}(K, I, X)$

$I_0 = I$
$K^1 = K$
for $j = 2$ to $\lceil |X|_n / l \rceil - 1$ do
    $K^j = \text{ACPKM}(K^{j-1})$
for $i = 1$ to $|X|_n$ do
    $j = \lceil i/l \rceil$
    $I_i = \text{incr}(I_{i-1})$
    $G_i = E_{K^j}(I_i)$
$Y = X \oplus \text{msb}_{|X|}\left(G_1 \| \ldots \| G_{|X|_n}\right)$
return Y

GCM-ACPKM$_{E,\tau,l}$.Encrypt(K, IV, A, M)

$I_0 = IV \| 0^{31}1$
Ciphertext computation:
    $C = \text{GCTR-ACPKM}_{E,\tau,l}(K, I_0, M)$
Tag computation:
    $H = E_K(0^n)$, $Z = E_K(I_0)$
    $T = \text{msb}_\tau\left(\text{GHASH}_H(A, C) \oplus Z\right)$
return (C, T)

GCM-ACPKM$_{E,\tau,l}$.Decrypt(K, IV, A, C, T)

$I_0 = IV \| 0^{31}1$
Plaintext computation:
    $M = \text{GCTR-ACPKM}_{E,\tau,l}(K, I_0, C)$
Tag verification:
    $H = E_K(0^n)$, $Z = E_K(I_0)$
    $T' = \text{msb}_\tau\left(\text{GHASH}_H(A, C) \oplus Z\right)$
    if $T = T'$ then
        return M
    else **return** $\perp$

Fig. 5. Authenticated encryption and decryption in the GCM-ACPKM Mode.

4.2 Security Notions

Block cipher. Standard security notions for block ciphers are PRP-CPA («Pseudo Random Permutation under Chosen Plaintext Attack») and PRF («Pseudo Random Function») (see, e.g., [4]).

For a cipher E with parameters n and k define

$$\mathbf{Adv}_E^{\text{PRP-CPA}}(\mathcal{A}) = \Pr\left[K \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}^{E_K} \Rightarrow 1\right] - \\ - \Pr\left[P \in_{\mathcal{U}} Perm(\{0,1\}^n) : \mathcal{A}^P \Rightarrow 1\right],$$

where the probabilities are defined over the randomness of $\mathcal{A}$, and the choices of K and P.

The PRF notion is defined in the same way as PRP-CPA except for the random permutation $P \in_{\mathcal{U}} Perm(\{0,1\}^n)$, which is replaced by the random

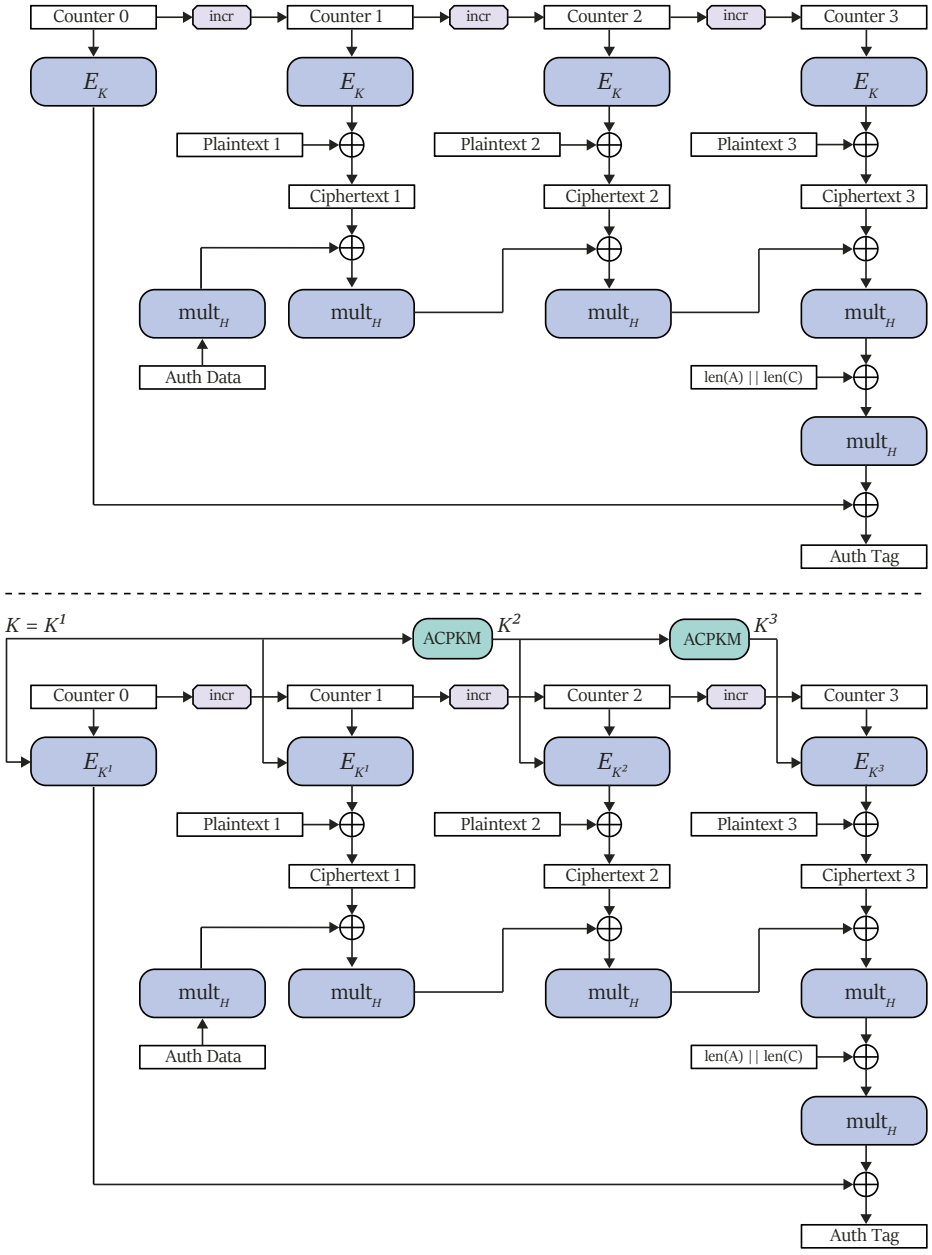

Fig. 6. The authenticated encryption operation for the GCM mode (top) and the GCM-ACPKM mode (bottom). For simplicity, a case with only a single block of additional authenticated data and three block of plaintext is shown. For GCM-ACPKM the section size is restricted to one block. Here E_K denotes the block cipher encryption using the key K, mult_H denotes multiplication in $GF(2^n)$ by the key H and incr denotes the counter increment function.

function $F \in_{\mathcal{U}} Func(\{0,1\}^n)$:

$$\mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{A}) = \Pr\left[K \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}^{E_K} \Rightarrow 1\right] - \\ - \Pr\left[F \in_{\mathcal{U}} Func(\{0,1\}^n) : \mathcal{A}^F \Rightarrow 1\right].$$

In the case of the block cipher with no attacks known, the values $\mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{A})$ and $\mathbf{Adv}_E^{\text{PRP-CPA}}(\mathcal{A})$ are bounded, considering the characteristics of general attacks. For the PRF notion it is the attack based on the birthday paradox, and for the PRP-CPA notion it is a brute force key search (e.g. random guessing). If the number of queries q exceeds the unicity distance of the block cipher E with block length n and key length k, i.e. as $qn \geqslant k$, then, assuming that one key trial spends at least a unit of computational resource t, we can suppose that for such a cipher the following inequality holds:

$$\mathbf{Adv}_E^{\text{PRP-CPA}}(\mathcal{A}) \leqslant \frac{t}{2^k}, \qquad \mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{A}) \leqslant \frac{t}{2^k} + \frac{q^2}{2^n}. \tag{1}$$

AEAD mode. Following [23] and [18], standard security notions for the AEAD modes are Privacy and Authenticity. Consider them for the abstract $\mathrm{AEAD}_{E,\tau}$ mode, where E is the underlined cipher with parameters n and k and τ is a tag size. For simplicity, below we consider the case where a ciphertext has the same length as a plaintext and an authentication tag of size τ can be treated separately from the ciphertext.

Privacy. We consider an adversary $\mathcal{A}$ that has access to an encryption oracle $\mathcal{E}$ or a random-bits oracle \$. Before starting the work the encryption oracle chooses a key $K \in_{\mathcal{U}} \{0,1\}^k$. The adversary makes queries (IV, A, M), where IV is a nonce, A is an associated data and M is a plaintext. The random-bits oracle in response returns (C, T), where $C\|T \in_{\mathcal{U}} \{0,1\}^{|M|+\tau}$. The encryption oracle returns (C, T), $C \in \{0,1\}^{|M|}$, $T \in \{0,1\}^\tau$, — the result of $\mathrm{AEAD}_{E,\tau}$ encryption of (IV, A, M) under the key K.

For the $\mathrm{AEAD}_{E,\tau}$ mode define

$$\mathbf{Adv}_{\mathrm{AEAD}_{E,\tau}}^{\mathrm{Priv}}(\mathcal{A}) = \Pr\left[K \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}^{\mathcal{E}} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\$} \Rightarrow 1\right],$$

where the probabilities are defined over the randomness of $\mathcal{A}$, the choices of K and randomness of the random-bits oracle, respectively. We consider a set of nonce-respecting adversaries, which choose IV unique for each query.

Authenticity. We consider an adversary $\mathcal{A}$ that has access to an encryption oracle $\mathcal{E}$ and a decryption oracle $\mathcal{D}$. Before starting the work both oracles choose a common key $K \in_{\mathcal{U}} \{0,1\}^k$. The adversary interacts with the encryption oracle in the same way as described in the Privacy notion. Additionally the adversary can make queries (IV, A, C, T) to the decryption oracle, where IV is a nonce, A is an associated data, C is a ciphertext and T is an authentication tag. Its returns the result of $\mathrm{AEAD}_{E,\tau}$ decryption of (IV, A, C, T) under the key K: $M \in \{0,1\}^{|C|}$ or $\perp$.

The adversary forges if the decryption oracle returns a bit string (other than $\perp$) for a query (IV, A, C, T), but (C, T) was not previously returned to $\mathcal{A}$ from the encryption oracle for a query (IV, A, M) with some M. As in the Privacy notion, we assume that $\mathcal{A}$ is nonce-respecting to encryption oracle. We remark that nonces used for the encryption queries can be used for decryption queries and vice-versa, and that the same nonce can be repeated for decryption queries.

For the $\mathrm{AEAD}_{E,\tau}$ mode define

$$\mathbf{Adv}^{\mathrm{Auth}}_{\mathrm{AEAD}_{E,\tau}}(\mathcal{A}) = \Pr\left[K \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}^{\mathcal{E},\mathcal{D}} \text{ forges}\right],$$

where the probability is defined over the randomness of $\mathcal{A}$ and the choice of K.

4.3 Security Bounds

GCM. Below we consider known results on the security of the GCM mode that are obtained in [23] for the first time and then repaired and improved in [18,27,28].

Theorem 1 ([28]). *Let E and τ be the parameters of* GCM. *Then for any adversary $\mathcal{A}$ with time complexity at most t that makes at most q queries, where the total plaintext length is at most σ blocks and the maximal nonce length is at most l_{IV} blocks, there exists an adversary $\mathcal{A}'$ such that*

$$\mathbf{Adv}^{\mathrm{Priv}}_{\mathrm{GCM}_{E,\tau}}(\mathcal{A}) \leqslant \mathbf{Adv}^{\text{PRP-CPA}}_{E}(\mathcal{A}') + \frac{(\sigma+q+1)^2}{2^{n+1}} + \frac{32q(\sigma+q)(l_{IV}+1)}{2^n},$$

where $\mathcal{A}'$ makes at most $\sigma + q + 1$ queries. Furthermore, the time complexity of $\mathcal{A}'$ is at most $t + cn\sigma_A$, where σ_A is the total input queries length, c is a constant that depends only on the model of computation and the method of encoding.

Corollary 1 ([18]). *Assume that the nonce length is restricted to 96 bits. Then,*

$$\mathbf{Adv}^{\mathrm{Priv}}_{\mathrm{GCM}_{E,\tau}}(\mathcal{A}) \leqslant \mathbf{Adv}^{\text{PRP-CPA}}_{E}(\mathcal{A}') + \frac{(\sigma+q+1)^2}{2^{n+1}}. \tag{2}$$

Theorem 2 ([28]). *Let E and τ be the parameters of* GCM. *Then for any adversary $\mathcal{A}$ with time complexity at most t that makes at most q encryption queries and q' decryption queries, where the total plaintext length is at most σ blocks, the maximal nonce length is at most l_{IV} blocks and the maximal summary length of plaintext or ciphertext and associated data in query is at most m_A blocks, there exists an adversary $\mathcal{A}'$ such that*

$$\mathbf{Adv}^{\mathrm{Auth}}_{\mathrm{GCM}_{E,\tau}}(\mathcal{A}) \leqslant \mathbf{Adv}^{\text{PRP-CPA}}_{E}(\mathcal{A}') + \\ + \left[\frac{32(q+q')(\sigma+q+1)(l_{IV}+1)}{2^n} + \frac{q'(m_A+1)}{2^\tau}\right] \cdot \delta_n(\sigma+q+q'+1),$$

where $\delta_n(x) := \dfrac{1}{(1-\frac{x-1}{2^n})^{x/2}}$, $\mathcal{A}'$ *makes at most* $\sigma + q + q' + 1$ *queries. Furthermore, the time complexity of* $\mathcal{A}'$ *is at most* $t + cn\sigma_A$, *where* σ_A *is the total queries length,* c *is a constant that depends only on the model of computation and the method of encoding.*

Corollary 2 ([27]). *Assume that the nonce length is restricted to 96 bits. If* $\sigma \leqslant 2^{n-1}$, *then,*

$$\mathbf{Adv}^{\text{Auth}}_{\text{GCM}_{E,\tau}}(\mathcal{A}) \leqslant \mathbf{Adv}^{\text{PRP-CPA}}_{E}(\mathcal{A}') + \left[\frac{q'(m_A+1)}{2^\tau}\right] \cdot \exp\left(\frac{4\sigma q}{2^n}\right). \quad (3)$$

GCM-ACPKM. Below we present the main results on the security of the internally re-keyed GCM-ACPKM mode. The obtained results allow to claim that this mode is secure if the base block cipher is secure and that the usage of the ACPKM internal re-keying technique increases security, essentially extending the lifetime of a key as compared to the base GCM mode.

Since the plaintext encryption for GCM-ACPKM is quite similar to the encryption for CTR-ACPKM the security bound for Privacy is obtained by the same way as described in [3]. Recall the main theorem about the security of CTR-ACPKM.

Theorem 3 ([3]). *Let* E *and* l *be the parameter of* CTR-ACPKM *mode. Then for any adversary* $\mathcal{A}$ *with time complexity at most* t *that makes queries, where the maximum message length is at most* m ($m \leqslant 2^{n/2-1}$) *blocks and the total message length is at most* σ *blocks, there exists an adversary* $\mathcal{A}'$ *such that*

$$\mathbf{Adv}^{\text{IND-CPNA}}_{\text{CTR-ACPKM}_{E,l}}(\mathcal{A}) \leqslant N \cdot \mathbf{Adv}^{\text{PRP-CPA}}_{E}(\mathcal{A}') + \\ + \frac{(\sigma_1+s)^2 + \ldots + (\sigma_{N-1}+s)^2 + (\sigma_N)^2}{2^{n+1}}$$

where $s = \lceil k/n \rceil$, $N = \lceil m/l \rceil$, σ_j *is the total data block length processed under the section key* K^j, $\sigma_1 + \ldots + \sigma_N = \sigma$. *The adversary* $\mathcal{A}'$ *makes at most* $\sigma_1 + s$ *queries. Furthermore, the time complexity of* $\mathcal{A}'$ *is at most* $t + cn(\sigma + ls)$, *where* c *is a constant that depends only on the model of computation and the method of encoding.*

Now we present the same theorem for GCM-ACPKM that shows the security bound for Privacy.

Theorem 4. *Let* E, τ *and* l *be the parameters of* GCM-ACPKM *mode. Then for any adversary* $\mathcal{A}$ *with time complexity at most* t *that makes at most* q *queries, where the maximal plaintext length is at most* $m \leqslant 2^{31} - 2$ *blocks and the total plaintext length is at most* σ *blocks, there exists an adversary* $\mathcal{A}'$ *such that*

$$\mathbf{Adv}^{\text{Priv}}_{\text{GCM-ACPKM}_{E,\tau,l}}(\mathcal{A}) \leqslant N \cdot \mathbf{Adv}^{\text{PRP-CPA}}_{E}(\mathcal{A}') + \frac{(\sigma_1+q+s+1)^2}{2^{n+1}} + \\ + \frac{(\sigma_2+s)^2 + \ldots + (\sigma_{N-1}+s)^2 + (\sigma_N)^2}{2^{n+1}}, \quad (4)$$

where $s = \lceil k/n \rceil$, $N = \lceil m/l \rceil$, σ_j *is the total data block length processed during plaintext encryption under the section key* K^j *and* $\sigma_1 + \ldots + \sigma_N = \sigma$. *The adversary* $\mathcal{A}'$ *makes at most* $\sigma_1 + q + s + 1$ *queries. Furthermore, the time complexity of* $\mathcal{A}'$ *is at most* $t + cn\sigma_A$, *where* σ_A *is the total input queries length,* c *is a constant that depends only on the model of computation and the method of encoding.*

Remark 1. Note that the re-keyed mode is secure if the value $s = \lceil k/n \rceil$ is rather small. For the common block ciphers (AES-256 and AES-128) this condition is satisfied: $s \in \{1, 2\}$.

Remark 2. Note that if $m \leqslant l$ (that is the case when the ACPKM mechanism is not applied, $\sigma_1 = \sigma$, $N = 1$) then the bound (4) totally coincides with the bound (2).

Remark 3. The bound for the internally re-keyed mode shows that the insecurity of the mode reaches minimum if $\sigma_1 = \ldots = \sigma_N$, i.e. if all messages are of the same length.

The proof can be found in Appendix B.

Now consider the security bound for Authenticity.

Theorem 5. *Let* E, τ *and* l *be the parameters of* GCM-ACPKM *mode. Then for any* $\mathcal{A}$ *with time complexity at most* t, *which makes at most* q *encryption queries and* q' *decryption queries, where the maximal summary length of plaintext or ciphertext and associated data in query is at most* m_A *blocks and the total plaintext length is at most* σ *blocks, there exists an adversary* $\mathcal{A}'$ *such that*

$$\mathbf{Adv}^{\text{Auth}}_{\text{GCM-ACPKM}_{E,\tau,l}}(\mathcal{A}) \leqslant \mathbf{Adv}^{\text{PRP-CPA}}_{E}(\mathcal{A}') + \left\lceil \frac{q'(m_A+1)}{2^\tau} \right\rceil \cdot \exp\left(\frac{4(\sigma_1 + s)q}{2^n} \right), \tag{5}$$

where $s = \lceil k/n \rceil$, σ_1 *is the total data block length processed during plaintext encryption under then section key* $K = K^1$, $\sigma_1 + s \leqslant 2^{n-1}$. *The adversary* $\mathcal{A}'$ *makes at most* $\sigma_1 + q + q' + s + 1$ *queries. Furthermore, the time complexity of* $\mathcal{A}'$ *is at most* $t + cn\sigma_A$, *where* σ_A *is the total input queries length,* c *is a constant, which depends only on the model of computation and the method of encoding.*

Remark 4. Note that if $m \leqslant l$ (that is the case when the ACPKM mechanism is not applied, $\sigma_1 = \sigma$, $N = 1$) then the bound (5) totally coincides with the bound (3).

The proof can be found in Appendix C.

4.4 Comparison of Bounds

Compare the security bounds of the GCM and GCM-ACPKM modes for a cipher E such that $s = \lceil k/n \rceil = 2$.

We assume that for the used cipher E the inequalities (1) hold. We also assume that $2^k \gg 2^n$. Note that the obtained bounds for the GCM mode are tight. For the Privacy notion it conventionally holds and for the Authenticity notions it follows from the recently obtained results [19,27].

Privacy. If $t \ll 2^k$ then for any adversary $\mathcal{A}$ with time complexity at most t that makes at most q queries, where the total plaintext length is at most σ blocks and the maximal plaintext length is at most $m \leqslant 2^{31} - 2$ blocks

$$\mathbf{Adv}^{\mathrm{Priv}}_{\mathrm{GCM}_{E,\tau}}(\mathcal{A}) \approx \frac{(\sigma+q)^2}{2^{n+1}},$$

$$\mathbf{Adv}^{\mathrm{Priv}}_{\text{GCM-ACPKM}_{E,\tau,l}}(\mathcal{A}) \approx \frac{(\sigma_1+q)^2+\sigma_2^2+\ldots+\sigma_{N-1}^2+\sigma_N^2}{2^{n+1}},$$

where $N = \lceil m/l \rceil$. Here we neglect the constants.

These relations indicate that the security of the GCM-ACPKM mode is improved compared to the security of the base GCM mode for the Privacy notion in the most typical cases due to $\sigma^2 \geqslant \sigma_1^2 + \ldots + \sigma_N^2$ for all $\sigma = \sigma_1 + \ldots + \sigma_N$.

Authenticity. For the same reasons for any adversary $\mathcal{A}$ with time complexity at most t that makes at most q encryption queries and q' decryption queries, where the total plaintext length is at most σ blocks and the maximal summary length of plaintext or ciphertext and associated data in query is at most m_A blocks,

$$\mathbf{Adv}^{\mathrm{Auth}}_{\mathrm{GCM}_{E,\tau}}(\mathcal{A}) \approx \frac{q' m_A}{2^{\tau}} \cdot \exp\left(\frac{4\sigma q}{2^n}\right),$$

$$\mathbf{Adv}^{\mathrm{Auth}}_{\text{GCM-ACPKM}_{E,\tau,l}}(\mathcal{A}) \approx \frac{q' m_A}{2^{\tau}} \cdot \exp\left(\frac{4\sigma_1 q}{2^n}\right).$$

The authenticity security of the GCM-ACPKM mode is also improved compared to the security of the base GCM mode for all typical cases since $\sigma_1 < \sigma$.

Remark 5. The paper [27] propose the attack that recovers the hash-key H of GCM with probability at least $\frac{1}{2}$ based on $\sqrt{n/m} \cdot 2^{n/2}$ encryption queries, where m is the number of blocks present in plaintext of encryption queries. In the case of GCM-ACPKM we need for now $\sqrt{n/l} \cdot 2^{n/2}$ encryption queries to recover the hash-key H using the same attack where l is the section size.

The considered bounds can be rewritten in the term of the q and m parameters using $\sigma \leqslant qm$ and $\sigma_i \leqslant ql$ for all i:

$$\mathbf{Adv}^{\mathrm{Priv}}_{\mathrm{GCM}_{E,\tau}}(\mathcal{A}) \approx \frac{(qm+q)^2}{2^{n+1}}, \quad \mathbf{Adv}^{\mathrm{Priv}}_{\text{GCM-ACPKM}_{E,\tau,l}}(\mathcal{A}) \approx \frac{m}{l} \cdot \frac{(ql+q)^2}{2^{n+1}},$$

$$\mathbf{Adv}^{\mathrm{Auth}}_{\text{GCM-ACPKM}_{E,\tau,l}}(\mathcal{A}) \approx \frac{q' m_A}{2^{\tau}} \cdot \exp\left(\frac{4lq^2}{2^n}\right).$$

Let fix a safety margin of privacy, which allows to process q messages with plaintext length exactly m blocks in the base $\mathrm{GCM}_{E,n}$ mode. Note that the case of equal length messages is practical: messages can be padded in purpose of achieving a length-hiding property. According to the approximate security bounds presented above the plaintext length can be (without loss in security)

increased by internal re-keying up to $\min\left(\frac{m+1}{l+1}\cdot m,\ 2^{31}-2\right)$. Herewith, if the length of an associated data (e.g. a header) is negligible compared to the maximal plaintext length, then the forgery probability for $q'=1$ is still at $\frac{1}{2^{\tau-32}}$ as long as $q \leqslant \frac{2^{n/2}-1}{\sqrt{l}}$.

4.5 Performance

As the GCM mode of operation is actively exploited in high-level protocols, the issue of efficiency of the extended GCM-ACPKM mode is highly important.

We analyze the correlation between efficiency of the internally re-keyed encryption mode and the section size l. The results are presented in the tables below, where the first row is the section size in kilobytes and the second one is the appropriate processing speed in megabytes per second. The last row shows loss of performance compared to the base mode (in percent). We measure the processing speed during the encryption of one long message in the GCM and GCM-ACPKM modes with the following ciphers: hardware-supported AES-256 and AES-128 (using OpenSSL source [33]). The computer with the following characteristics was used: Intel Core i5-6500 CPU 3.20GHz, L1 D-Cache 32 KB x 4, L1 I-Cache 32 KB x 4, L2 Cache 256 KB x 4.

Speed of the encryption process in the base GCM mode with the hardware-supported AES-256 cipher is 2690 MB/s and for the hardware-supported AES-128 cipher it is 3400 MB/s.

KB	64	128	256	512	1024	2048	4096
MB/s	2614.2	2628.2	2647.5	2661.6	2670.2	2680.1	2687.0
%	2.8	2.2	1.6	1.1	0.7	0.4	0.1

Table 1. The GCM-ACPKM mode with the AES-256 cipher (hardware support).

KB	64	128	256	512	1024	2048	4096
MB/s	3319.9	3330.9	3356.0	3370.3	3381.1	3390.9	3395.2
%	2.5	2.0	1.5	0.9	0.6	0.3	0.1

Table 2. The GCM-ACPKM mode with the AES-128 cipher (hardware support).

The section size can be varied depending on the different purposes. Obviously processing speed is proportional to the section size. However, when choosing this parameter, the following condition should be satisfied: the value ql (where q is the number of separate processed messages, l is the section size) should be no greater than the lifetime of a key.

5 Practical Significance

Consider the security bounds for GCM, for GCM-ACPKM, for key diversified GCM ($\overline{\text{GCM}}$) and for key diversified GCM-ACPKM ($\overline{\text{GCM-ACPKM}}$). The next theorem was originally formulated for the LOR-CPA notion in [1]. For convenience we convert it to the bound for the Privacy notion by the obvious reduction.

Theorem 6 ([1]). *Let $\mathcal{SE}$ be a base encryption scheme with key size k, $\mathcal{G}$ be a stateful generator with block size k and q be a subkey lifetime. Let $\overline{\mathcal{SE}}^q$ be the associated re-keyed encryption scheme. Then for any adversary $\mathcal{A}$ with time complexity at most t, which makes at most Q encryption queries, where the maximal plaintext length is at most m blocks, there exist adversaries $\mathcal{A}'$ and $\mathcal{A}''$ such that*

$$\mathbf{Adv}^{\text{Priv}}_{\overline{\mathcal{SE}}^q}(\mathcal{A}) \leqslant 2 \cdot \mathbf{Adv}^{\text{PRG}}_{\mathcal{G},N}(\mathcal{A}') + \left\lceil \frac{Q}{q} \right\rceil \cdot \mathbf{Adv}^{\text{Priv}}_{\mathcal{SE}}(\mathcal{A}''),$$

where $\mathcal{A}'$ makes at most q queries with the maximal plaintext length at most M blocks, and the time complexities of $\mathcal{A}'$ and $\mathcal{A}''$ are at most t.

Corollary 3. *The same bound can be applied for the Authenticity notion:*

$$\mathbf{Adv}^{\text{Auth}}_{\overline{\mathcal{SE}}^q}(\mathcal{A}) \leqslant 2 \cdot \mathbf{Adv}^{\text{PRG}}_{\mathcal{G},N}(\mathcal{A}') + \left\lceil \frac{Q}{q} \right\rceil \cdot \mathbf{Adv}^{\text{Auth}}_{\mathcal{SE}}(\mathcal{A}'').$$

If we assume the approximations considered in Section 4.4 for the adversary $\mathcal{A}$, which makes Q encryption queries, where all messages of the length m blocks, (thus, $\sigma < Qm$) and one decryption query, then we get the approximations, presented in Table 3.

Now consider the AES-GCM$_{E,n}$ and AES-$\overline{\text{GCM-ACPKM}_{E,n,l}}^q$ modes with parameters $n = 128$, $q = 2^6$ and $l = 2^6$. Let compare key lifetime limitations for these modes in TLS 1.3 protocol [30], where record size m is at most 2^{10} blocks or 2^{14} bytes. Technically, AES-$\overline{\text{GCM-ACPKM}}$ in TLS 1.3 assumes that the initial key should be diversified after every megabyte and every subkey should be internally updated after every kilobyte.

The comparison results are presented in Table 4, where the first column contains the number of processed record and the next columns contain the corresponding upper bounds for success probabilities of a privacy attack (δ_{priv}) and of a forgery (δ_{auth}). The success probabilities were calculated using the approximate security bounds presented in Table 3 and $\exp(4x/2^{128}) \leqslant 2$ for $x \leqslant 2^{126}$. Note that only in the case of GCM for $Q = 2^{64}$ this does not hold.

These results show that after processing by AES-GCM of maximum possible in TLS 1.3 number of records (2^{64}) both privacy and integrity will be totally corrupted. Herewith, the AES-$\overline{\text{GCM-ACPKM}}$ mode still remains secure up to 2^{-42} for privacy and 2^{-60} for integrity. Thus, using the TLS 1.3 KeyUpdate technique for key diversification together with the ACPKM technique for key meshing allows to drastically increase the key lifetime in TLS 1.3.

$\mathcal{SE}$	$\mathbf{Adv}^{\mathrm{Priv}}_{\mathcal{SE}}(\mathcal{A})$	$\mathbf{Adv}^{\mathrm{Auth}}_{\mathcal{SE}}(\mathcal{A})$
$\mathrm{GCM}_{E,\tau}$	$\frac{(Qm+Q)^2}{2^n}$	$\frac{m}{2^\tau}\cdot\exp\left(\frac{4mQ^2}{2^n}\right)$
$\overline{\mathrm{GCM}_{E,\tau}}^q$	$\frac{Q}{q}\cdot\frac{(qm+q)^2}{2^n}$	$\frac{Q}{q}\cdot\frac{m}{2^\tau}\cdot\exp\left(\frac{4mq^2}{2^n}\right)$
$\mathrm{GCM\text{-}ACPKM}_{E,\tau,l}$	$\frac{m}{l}\cdot\frac{(Ql+Q)^2}{2^n}$	$\frac{m}{2^\tau}\cdot\exp\left(\frac{4lQ^2}{2^n}\right)$
$\overline{\mathrm{GCM\text{-}ACPKM}_{E,\tau,l}}^q$	$\frac{Qm}{ql}\cdot\frac{(ql+q)^2}{2^n}$	$\frac{Q}{q}\cdot\frac{m}{2^\tau}\cdot\exp\left(\frac{4lq^2}{2^n}\right)$

Table 3. Approximate security bounds for the re-keyed GCM modes. Here Q is the number of queries to the encryption oracle, m is a number of blocks present in query, τ ia a tag size, q (subkey lifetime) and l (section size) are parameters of the external and internal re-keying techniques.

Max Records	GCM		$\overline{\text{GCM-ACPKM}}$	
	δ_{priv}	δ_{auth}	δ_{priv}	δ_{auth}
2^{34}	2^{-40}	2^{-117}	2^{-72}	2^{-89}
2^{44}	2^{-20}	2^{-117}	2^{-62}	2^{-79}
2^{54}	1	2^{-117}	2^{-52}	2^{-69}
2^{64}	1	1	2^{-42}	2^{-59}

Table 4. Key lifetime limitations in TLS 1.3 with record size $m = 2^{10}$ blocks (16 kilobytes) for AES-GCM and AES-$\overline{\text{GCM-ACPKM}}$ with parameters $n = 128$ bits, $q = 2^6$ records (1 megabyte), $l = 2^6$ blocks (1 kilobyte).

6 Conclusion

In this paper, we have introduced the clear classification of existing re-keying approaches and have discussed their advantages and disadvantages. We have proposed a new internally re-keyed GCM-ACPKM mode and have studied its security, respecting the standard notions. We have shown that the security for the Privacy and Authenticity notion is increased compared to the base mode. Therefore we are convinced that the overall security of GCM is drastically increased by the ACPKM re-keying technique with only a minor loss in performance.

Also we have considered the composition of internal and external re-keying approaches and have provided certain parameters leading to improvements in applications, particularly in TLS 1.3.

The most interesting open problem is to thoroughly analyze the security of the key update technique without master key in a side-channel security model (e.g. described in [24]), where an adversary has some additional information about section keys (e.g. some key bits). In the case of using the master key, keys are non-computable from each other and can be considered as independent. Therefore we cannot tie side-channel information obtained for different keys to break one of them.

Keys generated according to key update techniques without master key are related. However, key transformation considered in the current paper shuffle key bits such that the task to tie side-channel data for different sections seems to be computationally intractable. Therefore a problem of obtaining certain security bounds in the side-channel model is still interesting.

References

1. Abdalla, M., Bellare, M. Increasing the Lifetime of a Key: A Comparative Analysis of the Security of Re-keying Techniques. In Okamoto, T., ed.: Advances in Cryptology — ASIACRYPT '00. Volume 1976 of LNCS., Springer (December 3-7, 2000) 546–559.
2. Ahmetzyanova, L, Alekseev, E, Oshkin, I, Smyshlyaev, S, Sonina, L. On the properties of the CTR encryption mode of the Magma and Kuznyechik block ciphers with re-keying method based on CryptoPro Key Meshing. IACR Cryptology ePrint Archive, 2016:628, 2016.
3. Ahmetzyanova, L., Alekseev, E., Smyshlyaev, S. Security bound for CTR-ACPKM internally re-keyed encryption mode. IACR Cryptology ePrint Archive, 2018:950.
4. Bellare, M., Desai, A., Jokipii, E., Rogaway, P. A concrete security treatment of symmetric encryption. In Proceedings of 38th Annual Symposium on Foundations of Computer Science (FOCS '97), pages 394–403. IEEE, 1997.
5. Bellare, M., Pietrzak, K., and Rogaway, P. Improved security analyses for CBC MACs. In V. Shoup, editor, CRYPTO 2005, volume 3621 of LNCS, pages 527–545. Springer, Aug. 2005.
6. Bellare, M. , Rogaway, P. Introduction to modern cryptography, 2005. URL: http://cseweb.ucsd.edu/~mihir/cse207/classnotes.html.
7. Bellare, M. Practice-Oriented Provable-Security. Modern Cryptology in Theory and Practice, P. 1-15, 1999.
8. Bellare, M., Yee, B. Forward-Security in Private-Key Cryptography. In: Joye M. (eds) Topics in Cryptology – CT-RSA 2003. CT-RSA 2003. Lecture Notes in Computer Science, vol 2612. Springer, Berlin, Heidelberg, 2003.
9. Bernstein, D.J. Stronger Security Bounds for Permutations, 2005. URL: http://cr.yp.to/papers.html
10. Bhargavan, K., Leurent, G. «On the Practical (In-)Security of 64-bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN». IACR Cryptology ePrint Archive, 2016:798, 2016.
11. Biham, E. How to Forge DES-Encrypted Messages in 2^{28} Steps. Technion Computer Science Department Technical Report CS0884, 1996.
12. Biham, E., Shamir, A. Differential Cryptanalysis of DES-like Cryptosystems // Journal of Cryptology. V. 537. P. 2-21. 1990.

13. Biryukov, A., Khovratovich, D.: Two New Techniques of Side-Channel Cryptanalysis. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, p. 195–208. Springer, Heidelberg, 2007.
14. Bogdanov, A.: Improved Side-Channel Collision Attacks on AES. In: Adams, C., Miri, A., Wiener, M. (eds.) SAC 2007. LNCS, vol. 4876, pp. 84–95. Springer, Heidelberg, 2007.
15. Chang, D., Nandi, M. A Short Proof of the PRP/PRF Switching Lemma. IACR Cryptology ePrint Archive, 2008:078, 2008.
16. Iwata, T., Kurosawa, K. OMAC: One-Key CBC MAC. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 129–153. Springer, Heidelberg, 2003.
17. Iwata, T., Kurosawa, K. Stronger Security Bounds for OMAC, TMAC and XCBC. In proceedings of 4th International Conference on Cryptology in India, New Delhi, India, December 8-10, 2003.
18. Iwata,T., Ohashi, K., Minematsu K. Breaking and Repairing GCM Security Proofs. CRYPTO 2012, LNCS, vol. 7417, pp. 31-49. Springer, Heidelberg, 2012.
19. Luykx, A. and Preneel, B. Optimal forgeries against polynomial-based macs and gcm. In EUROCRYPT 2018, 2018.
20. Rogaway, P. Nonce-Based Symmetric Encryption. The 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004.
21. Luykx, A. and Paterson K. G. Limits on authenticated encryption use in TLS, 2015.
URL: http://www.isg.rhul.ac.uk/~kp/TLS-AEbounds.pdf
22. Matsui, M. Linear Cryptanalysis Method for DES Cipher // Advanced in Cryptology - EUROCRYPT'93. Lect. Notes in Comp. Sci., Springer, 1994. V. 765. P. 386-397.
23. McGrew, D.A., Viega, J. The security and perfomance of the Galois/Counter Mode (GCM) of operation. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 343-355. Springer, Heidelberg, 2004.
24. Micali, S., Reyzin, L. Physically Observable Cryptography (extended abstract). TCC 2004, LNCS, vol. 2951, pp. 278–296.
25. Mitchell C.J. On the security of XCBC, TMAC and OMAC. Technical Report RHUL-MA-2003-4, 19 August, 2003.
URL: http://www.rhul.ac.uk/mathematics/techreports.
26. Chen, L. NIST Special Publication 800-108. Recommendation for Key Derivation Using Pseudorandom Functions (Revised). 2009.
27. Nandi, M. Bernstein Bound on WCS is Tight Repairing Luykx-Preneel Optimal Forgeries. In: Shacham H., Boldyreva A. (eds) Advances in Cryptology – CRYPTO 2018. CRYPTO 2018. Lecture Notes in Computer Science, vol 10992.
28. Niwa, Y., Iwata, T., Ohashi, K., Minematsu, K. GCM Security Bounds Reconsidered. In: 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers.
29. Popov, V., Kurepkin, I., Leontiev, S. Additional cryptographic algorithms for use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 algorithms. RFC 4357. 2007.
30. Rescorla, E., RTFM, Inc. The Transport Layer Security (TLS) Protocol Version 1.3, RFC 8446, August 2018.
31. Rogaway, P. Authenticated-encryption with associated-data. Ninth ACM Conference on Computer and Communications Security (CCS-9). ACM Press, 2002. Proceedings version of this paper.
32. Ramsay C., Lohuis J. «TEMPEST attacks against AES. Covertly stealing keys for € 200», https://www.fox-it.com, 2017.

33. URL: https://www.openssl.org/
34. Rescorla E. and Modadugu N. «Datagram Transport Layer Security Version 1.2», RFC 6347, DOI 10.17487/RFC6347, January 2012,
35. Kent S. «IP Encapsulating Security Payload (ESP)», RFC 4303, DOI 10.17487/RFC4303, December 2005,
36. Dierks T. and Rescorla E. «The Transport Layer Security (TLS) Protocol Version 1.2», RFC 5246, DOI 10.17487/RFC5246, August 2008,
37. Ylonen T. and Lonvick C., Ed., «The Secure Shell (SSH) Transport Layer Protocol», RFC 4253, DOI 10.17487/RFC4253, January 2006,
38. Smyshlyaev, S. «Re-keying Mechanisms for Symmetric Keys», draft-irtf-cfrg-re-keying-13, March 25, 2019.
39. Smyshlyaev, S. «GOST Cipher Suites for Transport Layer Security (TLS) Protocol Version 1.2», draft-smyshlyaev-tls12-gost-suites-04, December 29, 2018.

A Additional notations

B Proof of Theorem 4

Proof. Define the hybrid experiments $Hybrid_j(\mathcal{A})$, $j = 0, 1, \ldots, N$. In the experiment $Hybrid_j(\mathcal{A})$ the oracle in the Privacy notion is replaced by the oracle, which operates in the following way:

- The oracle chooses key $K^{j+1} \in_{\mathcal{U}} \{0,1\}^k$;
- In response to a query (IV, A, M) the oracle returns a pair (C, T) which is calculated as follows.
  A ciphertext:
  $$C = M \oplus \mathrm{msb}_{|M|}(G' \| G^{j+1} \| \ldots \| G^N),$$
  where $G' \in_{\mathcal{U}} \{0,1\}^{nlj}$ and $G^i = E_{K^i}(I_{(i-1)l+1}) \| \ldots \| E_{K^i}(I_{il})$, $i = (j+1), \ldots, N$, is the concatenation of the appropriate l encrypted counter blocks under the K^i section key. Note that the $(j+1)$-th section is processed under the «truly» random K^{j+1} key and each next key is produced from previous one according to ACPKM.
  An authentication tag:
  $$T = \mathrm{msb}_\tau(Z \oplus \mathrm{GHASH}_H(A, C)),$$
  where $Z = E_{K^1}(IV \| \mathrm{str}_{n-96}(1))$, $H = E_{K^1}(0^n)$ if $j = 0$, and $Z, H \in_{\mathcal{U}} \{0,1\}^n$, otherwise.

The result of any experiment described above is what the adversary $\mathcal{A}$ returns as a result. Further we denote by $Hybrid_j(\mathcal{A}) \Rightarrow 1$ an event, which occurs if the result of the experiment $Hybrid_j(\mathcal{A})$ is 1.

Note that for the adversary $\mathcal{A}$ the oracle in the experiment $Hybrid_N(\mathcal{A})$ totally coincides with the oracle \$, and the oracle in the experiment $Hybrid_0(\mathcal{A})$ coincides with the oracle $\mathcal{E}$, i.e. the following equalities hold:

$$\Pr\left[Hybrid_N(\mathcal{A}) \Rightarrow 1\right] = \Pr\left[\mathcal{A}^{\$} \Rightarrow 1\right],$$

$$\Pr\left[Hybrid_0(\mathcal{A}) \Rightarrow 1\right] = \Pr\left[K \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}^{\mathcal{E}} \Rightarrow 1\right].$$

Construct a set of adversaries $\mathcal{A}'_j$, $j = 1, \ldots, N$, for the block cipher E in the PRF model, which uses $\mathcal{A}$ as a black box.

After receiving a query (IV, A, M) from $\mathcal{A}$ the adversary $\mathcal{A}'_j$ processes this query as in the $Hybrid_j(\mathcal{A})$ experiment but the encrypted blocks for masking the j-th section and blocks of the $(j+1)$-th section key are obtained by making queries to the oracles F or E_K provided by the PRF experiment. Note that $\mathcal{A}'_1$ makes at most $\sigma_1 + q + s + 1$ queries, $\mathcal{A}'_j$, $j = 2, \ldots, N-1$, makes at most $\sigma_j + s$ queries and $\mathcal{A}'_N$ makes at most σ_N queries. The adversary $\mathcal{A}'_j$ returns 1, if the adversary $\mathcal{A}$ returns 1, and returns 0, otherwise.

Note that

$$\Pr\left[K \in_{\mathcal{U}} \{0,1\}^k : (\mathcal{A}'_j)^{E_K} \Rightarrow 1\right] = \Pr\left[Hybrid_{j-1}(\mathcal{A}) \Rightarrow 1\right].$$

$$\Pr\left[F \in_{\mathcal{U}} Func(\{0,1\}^n) : (\mathcal{A}'_j)^{F} \Rightarrow 1\right] = \Pr\left[Hybrid_j(\mathcal{A}) \Rightarrow 1\right],$$

The last equality is proceeded from that the input blocks for producing the K^{j+1} section key and the input blocks for masking the j-th section and producing the Z and H values are different for the random function. Therefore, the K^{j+1} variable distribution is statistically indistinguishable from the uniform one.

Then for the advantages of the adversaries $\mathcal{A}'_j$

$$\begin{aligned}\sum_{j=1}^{N} \mathbf{Adv}_E^{\mathrm{PRF}}\left(\mathcal{A}'_j\right) = \sum_{j=1}^{N} \Big(&\Pr\left[K \in_{\mathcal{U}} \{0,1\}^k : (\mathcal{A}'_j)^{E_K} \Rightarrow 1\right] - \\ &- \Pr\left[F \in_{\mathcal{U}} Func(\{0,1\}^n) : (\mathcal{A}'_j)^{F} \Rightarrow 1\right]\Big) = \\ = \sum_{j=1}^{N} \Pr\left[Hybrid_{j-1}(\mathcal{A}) \Rightarrow 1\right] &- \sum_{j=1}^{N} \Pr\left[Hybrid_j(\mathcal{A}) \Rightarrow 1\right] = \\ = \Pr\left[Hybrid_0(\mathcal{A}) \Rightarrow 1\right] - \Pr\left[Hybrid_N(\mathcal{A}) \Rightarrow 1\right] &= \mathbf{Adv}^{\mathrm{Priv}}_{\text{GCM-ACPKM}_{E,\tau,l}}(\mathcal{A}).\end{aligned}$$

From the PRP/PRF switching lemma [15] for any block cipher E and any adversary $\mathcal{A}'$ making at most q queries we have

$$\mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{A}') \leqslant \mathbf{Adv}_E^{\text{PRP-CPA}}(\mathcal{A}') + \frac{q(q-1)}{2^{n+1}} \leqslant \mathbf{Adv}_E^{\text{PRP-CPA}}(\mathcal{A}') + \frac{q^2}{2^{n+1}}.$$

Thus,

$$\mathbf{Adv}^{\mathrm{Priv}}_{\text{GCM-ACPKM}_{E,\tau,l}}(\mathcal{A}) = \sum_{j=1}^{N} \mathbf{Adv}^{\mathrm{PRF}}_{E}\left(\mathcal{A}'_j\right) \leqslant$$

$$\leqslant \left(\mathbf{Adv}^{\text{PRP-CPA}}_{E}\left(\mathcal{A}'_1\right) + \frac{(\sigma_1 + q + s + 1)^2}{2^{n+1}}\right) +$$

$$+ \sum_{j=2}^{N-1} \left(\mathbf{Adv}^{\text{PRP-CPA}}_{E}\left(\mathcal{A}'_j\right) + \frac{(\sigma_j + s)^2}{2^{n+1}}\right) + \left(\mathbf{Adv}^{\text{PRP-CPA}}_{E}\left(\mathcal{A}'_N\right) + \frac{\sigma_N^2}{2^{n+1}}\right) \leqslant$$

$$\leqslant N \cdot \mathbf{Adv}^{\text{PRP-CPA}}_{E}\left(\mathcal{A}'\right) + \frac{(\sigma_1 + q + s + 1)^2}{2^{n+1}} +$$

$$+ \frac{(\sigma_2 + s)^2 + \ldots + (\sigma_{N-1} + s)^2 + \sigma_N^2}{2^{n+1}},$$

where $\mathcal{A}'$ is an adversary which makes at most $\sigma_1 + q + s + 1$ queries. The last relation is due to $\sigma_1 \geqslant \ldots \geqslant \sigma_N$ and $\mathbf{Adv}^{\text{PRP-CPA}}_{E}\left(\mathcal{A}''\right) \leqslant \mathbf{Adv}^{\text{PRP-CPA}}_{E}\left(\mathcal{A}'\right)$ for such adversaries $\mathcal{A}'$ and $\mathcal{A}''$ with the same computational resources that the queries number made by $\mathcal{A}''$ is less than the queries number made by $\mathcal{A}'$.

□

C Proof of Theorem 5

Proof. For the proposed GCM-ACPKM mode the proof of security in the Authenticity model is the same as for Theorem 5 [27]. Indeed, the ACPKM technique influences on the plaintext encryption only and does not change the tag computation: values $Z = E_K(I)$ and $H = E_K(0^n)$ are computed under the initial key $K = K^1$. Without loss of generality, we assume a key size k be multiple of a block size n, and $s = k/n$.

Consider the following modification of the Authenticity model: the adversary at the begining of the game additionally takes as input blocks $\pi(D_1), \ldots, \pi(D_s)$. Note that the advantage of the adversary in this game is not less then the same advantage in the initial game.

The proof is identical to one described in [27] except for the only modification in the definition of the event $E(\kappa)$ which in our case denotes that $\pi(IV_1 \| 0^{31}1) = y_1(\kappa), \ldots, \pi(IV_q \| 0^{31}1) = y_q(\kappa)$, $\pi(IV_i \| \mathrm{str}_{32}(j)) = z_i$ for all $1 \leqslant i \leqslant q$, $1 \leqslant j \leqslant l$, and $\pi(D_1) \| \ldots \| \pi(D_s) = K^2$. Thus, σ is replaced by $\sigma_1 + s$ that gives the required bound.