Increasing the Lifetime of Symmetric Keys for the GCM Mode by Internal Re-keying

Liliya R. Ahmetzyanova, Evgeny K. Alekseev, 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 approaches to increase the security of block cipher operation modes based on re-keying, putting the focus on so-called *internal re-keying without master key* — re-keying during each separate message processing with no additional keys required. For extending the GCM base mode we provide an internal re-keying technique called ACPKM. This technique does not require additional secret parameters or complicated transformations — for key updating only the base encryption function is used. We quantify the security of the re-keyed GCM-ACPKM mode, respecting standard security notions with nonce-respecting adversaries, as a function of the security of a used primitive. We claim that the obtained proof framework can be reused to provide security bounds for other re-keyed modes without a master key. We also show that the ACPKM internal re-keying technique increases security, essentially extending the lifetime of a key with a minor loss in performance. We also consider the composition of internal and external re-keying and compare key lifetime limitations for the base and re-keyed GCM modes in TLS 1.3.

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 subject to birthday attack [3]), or from resisting certain specific cryptographic attacks on the used block cipher (e.g. differential [10] or linear cryptanalysis [19]), including side-channel attacks [28,11,12] (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 real vulnerabilities (see, e.g., [8,28]). Thus, when the total length of a plaintext processed with the same key reaches threshold values, 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 the interruption of payload transmissions, additional service-based data in the channel, the use of 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 [23], HKDF [26]). The session key should be updated each time a given amount of whole messages is processed. Another mechanism, called key meshing [25], 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. A new session key should be derived each time a given amount of whole messages is processed (e.g. $2^{24.5}$ records in TLS 1.3 for a certain safety margin [26]).

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. A relation was also obtained to bound the security of the key diversified mode of operation, separately analyzing the re-keying technique (the PRG notion of distinguishing a key sequence obtained according to the re-keying technique from a random key sequence) and the base mode of operation. Such 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 [7].

Key Meshing. Another mechanism to increase the key lifetime was presented for the first time in [25] and is called «CryptoPro Key Meshing» (CPKM). This solution assumes that each message is processed starting from the initially negotiated key, which is transformed each time a given relatively small amount of data has been processed. Such a transformation does not require any additional secret values and uses the initial key directly for data processing.

An operating disadvantages of CPKM is the usage of the decryption function. This doubles the code size for some block cipher modes and can reduce the performance. Another disadvantage is that the probability of trivial-breaking the derived key is nonzero.

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].

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, and discuss their features, advantages and disadvantages. We propose a new advanced CPKM (ACPKM) re-keying technique based on the internal re-keying approach for increasing the lifetime of keys used in the GCM mode. This technique respects the mode features and does not have the disadvantages of CPKM — only the encryption function is used and the probability of trivial-breaking the derived keys is zero. We obtain the security bounds for the re-keyed GCM-ACPKM mode in the Privacy and Authenticity notions with a nonce-respecting adversary. These bounds show that using the ACPKM re-keying technique significantly increases the lifetime of a key. We also claim that the obtained proof framework (the accompanying IKM task) is useful to obtain security bounds for other re-keyed modes of operation which do not use additional secret values (without master key).

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.

We also discuss the relationship between the internal and external re-keying approaches. We show that internal re-keying can be treated not as an alternative of the 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 lifetime of a key is strictly bounded [28]. Using the example of TLS 1.3 we show that the composition of these approaches essentially increases the key lifetime more than 5000 times.

1.3 Organization

The rest of this paper is organized as follows. Section 2 is dedicated to preliminaries. In Section 3 we recall some background on block ciphers and the associated security notions. We also review a «mode of operation» term, particularly the GCM mode of operation, and known security bounds. In Section 4 we describe internal and external re-keying approaches and introduce an ACPKM re-keying technique for the GCM mode. In Section 5 we provide a theorem on the security of the internally re-keyed GCM-ACPKM mode and then analyze its cryptographic and operational properties. In Section 6 we consider the composition of external and internal re-keying approaches and show by the example of TLS 1.3 that this hybrid technique allows to significantly increase the key lifetime. In Section 7 we point to several open problems which are associated with considered schemes. Finally, in the Appendix we prove the main theorems.

2 Preliminaries

By $\{0,1\}^n$ we denote the set of n-component bit strings. Let ε be the empty string and 0^n be the string, consisting of n zeros. For bit strings A and B we denote by $A\|B$ their concatenation. We denote by $M_{(i)}$ the i-th bit, $i \in \{0,\ldots,n-1\}$, of the string $M = M_{(0)}\|\ldots\|M_{(n-1)} \in \{0,1\}^n$. For positive integers m, n and the bit string $M \in \{0,1\}^{mn}$ we denote by $M[i]$, $0 \leqslant i \leqslant m-1$, the n-bit string $M_{(i\cdot n)}\|M_{(i\cdot n+1)}\|\ldots\|M_{(i\cdot n+n-1)}$ and call it the i-th block of the string M. Thus $M = M[0]\|M[1]\|\ldots\|M[m-1]$. Let $|M|$ be the bit length of the string M, and $|M|_n = \lceil |M|/n \rceil$ be its length in n-bit blocks.

For a bit string M and a positive integer $l \leqslant |M|$ let $\mathrm{msb}_l(M)$ $(\mathrm{lsb}_l(M))$ be the string, consisting of the leftmost (rightmost) l bits of M. For 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 a nonnegative integer l and a bit string $M \in \{0,1\}^l$ let $\mathrm{int}(M)$ be an integer i such that $\mathrm{str}_l(i) = M$.

For any set A, define $Perm(A)$ as the set of all bijective mappings on A (permutations on A), and $Func(A)$ as the set of all mappings from A to A. 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. If the value s is chosen in a set S uniformly at random, then we denote $s \in_{\mathcal{U}} S$.

We model an adversary using a probabilistic algorithm that has access to one or more oracles. Denote by $\mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2,\ldots}$ an adversary $\mathcal{A}$ that interacts with oracles $\mathcal{O}_1,\mathcal{O}_2,\ldots$ by making queries. Notation $\mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2,\ldots} \Rightarrow 1$ means that an algorithm $\mathcal{A}$, after interacting with oracles $\mathcal{O}_1,\mathcal{O}_2,\ldots$, 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 Modes of Operation and Security Notions

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

Standard security notions for block ciphers are PRP-CPA («Pseudo Random Permutation under Chosen Plaintext Attack») and PRF («Pseudo Random Function») (see, e.g., [3]).

For a cipher E with parameters n and k define

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

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

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

In the case of the block cipher with no attacks known, the values $\mathbf{Adv}_E^{\text{PRF}}(\mathcal{A})$ and $\mathbf{Adv}_E^{\text{PRP-CPA}}(\mathcal{A})$ are estimated, 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 exhaustive key search or linear cryptanalysis (see, e.g., [5]). So for such a cipher E we assume that for any adversary $\mathcal{A}$ with the time complexity at most t, making at most q queries,

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

Modes of Operation. the above-mentioned cryptographic challenges can be solved with the use of «block cipher modes of operation». In the current paper we consider the GCM authenticated encryption with associated data (AEAD) mode which is defined according to [20].

Denote by $\text{GCM}_{\mathcal{P},\tau}$ the GCM mode that uses the family $\mathcal{P}$ of permutations on $\{0,1\}^n$ and the positive integer $\tau \leqslant n$, denoted to a tag size, as parameters. Note, that GCM construction requires blockciphers with block size $n \geqslant 128$.

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

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

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 $X = X_1 \| \ldots \| X_m = A \| 0^a \| B \| 0^b \| \text{str}_{n/2}(|A|) \| \text{str}_{n/2}(|B|)$. If $A = \varepsilon$ then $X = X_1 \| \ldots \| X_m = B \| 0^b \| \text{str}_n(|B|)$.

Let $\pi : \{0,1\}^n \times \mathbb{N} \to \{0,1\}^n$ be the encoding function which takes the input (I, i), $I \in \{0,1\}^n$, $i \in \mathbb{N}$, and outputs the string

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

Authenticated Encryption in the GCM Mode. A processed message for authenticated encryption in the $\mathrm{GCM}_{\mathcal{P},\tau}$ 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, $0 \leqslant |M| \leqslant n(2^{32} - 2)$. For a permutation $P \in \mathcal{P}$ the result of GCM encryption is (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:

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

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

Here $H = P(0^n)$, $I_i = \pi(I, i)$, $1 \leqslant i \leqslant |M|_n$, where $I = IV\|\mathrm{str}_{n-96}(1)$, if $|IV| = 96$, or $I = \mathrm{GHASH}_H(\varepsilon, IV)$, otherwise. The nonces IV are different for different messages processed with the same permutation P.

Authenticated Decryption in the GCM Mode. An input message of authenticated decryption in the $\mathrm{GCM}_{\mathcal{P},\tau}$ 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. For a permutation $P \in \mathcal{P}$ the result of GCM decryption is the 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.

3.1 Security Notions for the GCM Mode

Following [20] and [16], standard security notions for the AEAD modes are Privacy and Authenticity. Consider them for the $\mathrm{GCM}_{E,\tau}$ mode, where E is the cipher with parameters n and k.

Privacy. We consider an adversary $\mathcal{A}$ that has access to an encryption oracle GCM-$\mathcal{E}$ 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 GCM encryption of (IV, A, M) for permutation E_K.

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

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

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 that choose IV unique for each query.

Authenticity. We consider an adversary $\mathcal{A}$ that has access to an encryption oracle GCM-$\mathcal{E}$ and a decryption oracle GCM-$\mathcal{D}$. 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 GCM decryption of (IV, A, C, T) for permutation E_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). 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 $\mathrm{GCM}_{E,\tau}$ mode define

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

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

Below we consider known results on the security of the GCM mode that are obtained in [20] for the first time and then repaired in [16].

Theorem 1. *[16] Let E and τ be the parameters of GCM. Then for any adversary $\mathcal{A}$ with at most time complexity 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*

$$\mathbf{Adv}^{Priv}_{GCM_{E,\tau}}(\mathcal{A}) \leqslant \mathbf{Adv}^{PRP\text{-}CPA}_{E}(\mathcal{A}') + \frac{(\sigma+q+1)^2}{2^{n+1}} + \frac{2^{22}q(\sigma+q)(l_{IV}+1)}{2^n},$$

where $\mathcal{A}'$ makes at most $\sigma + q + 1$ queries. Furthermore, the time complexity of $\mathcal{A}'$ is $t + cn\sigma_A$, 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. *[16] Assume that the nonce length is restricted to 96 bits. Then,*

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

Theorem 2. *[16] Let E and τ be the parameters of GCM. Then for any adversary $\mathcal{A}$ with at most time complexity 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 l_A blocks, there exists an adversary $\mathcal{A}'$ such that*

$$\mathbf{Adv}^{Auth}_{GCM_{E,\tau}}(\mathcal{A}) \leqslant \mathbf{Adv}^{PRP\text{-}CPA}_{E}(\mathcal{A}') + \frac{(\sigma+q+q'+1)^2}{2^{n+1}} + \\ + \frac{2^{22}(q+q'+1)(\sigma+q)(l_{IV}+1)}{2^n} + \frac{q'(l_A+1)}{2^\tau},$$

where $\mathcal{A}'$ makes at most $\sigma + q + q' + 1$ queries. Furthermore, the time complexity of $\mathcal{A}'$ is $t + cn\sigma_A$, 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. *[16] Assume that the nonce length is restricted to 96 bits. Then,*

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

4 Extending Block Cipher Modes by Re-keying

Re-keying is an approach, which is widely used for increasing the security of cryptographic schemes and protocols. 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 accompanying key update techniques (with *master key* and without it). 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.

In the current paper we put the focus on the internal re-keying approach without master key since its properties were not considered carefully. The known proof frameworks (e.g. [1]) cannot be applied to this construction, and a new approach is required. We present an ACPKM internal re-keying technique (without master key) for the GCM mode and demonstrate the proof framework by providing the security bounds for an internally re-keyed GCM-ACPKM mode.

4.1 External Re-keying

The main concept of this approach is as follows. A key, derived according to certain key update technique, is intended to process the fixed amount of separate messages after which the key should be updated. Using of 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 amount. In order to satisfy a requirement on the key lifetime limitation one should fix the maximal message length. If this requirement is restrictive enough (e.g, in the case of side-channel attacks) it leads to some problems. Thus, the amount of messages processed with the same key is proportional to their maximal length, and long message processing requires

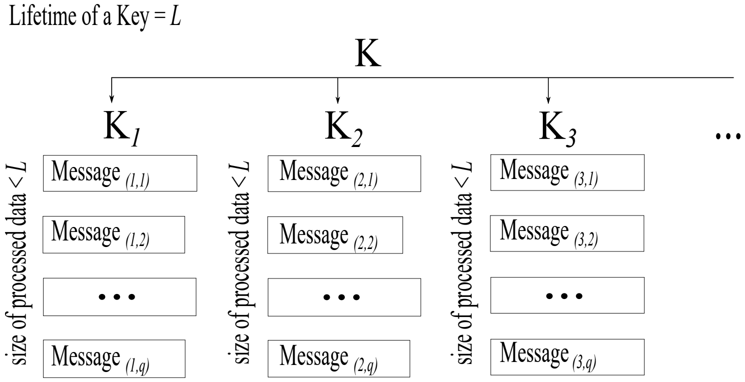

Fig. 1. External re-keying. Here q is a number of messages processed with one of keys K_i derived from the initial key K. The lifetime L of a key K_i (can be measured in blocks) defines the total length of data processed with this key.

additional fragmentation. Such a fragmentation can lead to frequent re-using a random number generator for generating IV (e.g., in the case of data processing in the CBC or CFB modes), that significantly affects the performance.

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

4.2 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 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 message blocks processed using the same key which we will call a «section key». Obviously a section size is bounded by the key lifetime that 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 as it affects the operating properties and limits the amount of messages: the larger the section size, the faster message processing (see Section 5.2), but the smaller the section size, the greater the amount of separate processed messages.

Security analysis of all internally re-keyed modes leads to the analysis of the abstract modes where section keys are chosen independently at random. For some mode of operation (CTR, CBC, OFB) the security of corresponding modes with

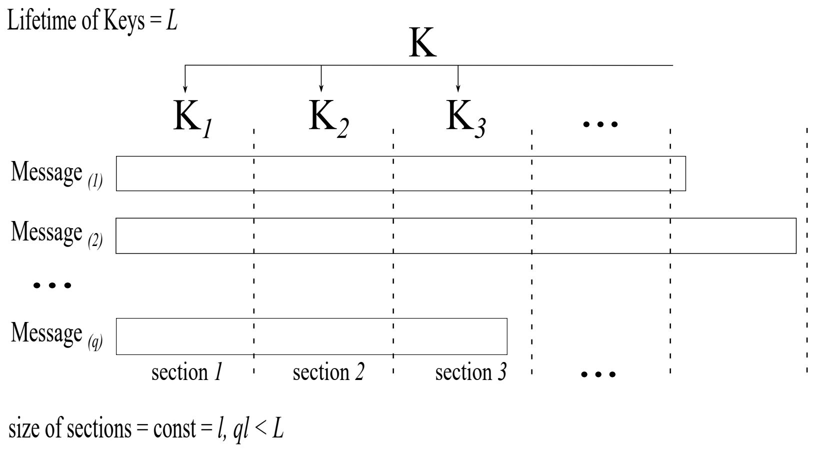

Fig. 2. Internal re-keying. Here q is a number of processed messages and each message is processed starting from the first derived key K_1. This key is changed each time a data section of fixed length l has been processed. The lifetime L of a key K_i (can be measured in blocks) defines the total length of data processed with this key.

random keys can be easily analyzed, using the technique of hybrid argument. 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 that 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 that can be processed with one key.

4.3 Key Update Techniques

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

We separate key update techniques with master key and without it. The first one has the following property: a shared initial key is never used directly for the encryption but is used for subkey derivation. Using of 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 used components:

- for external re-keying they are the key update technique and the base mode of operation [1];
- for internal re-keying they are the key update technique and the abstract mode with random section keys.

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

The second technique directly uses the initial key as the first key for data processing, and each next key is computed from previous one. It seems to be mostly useful in the case when the total amount of data for an established 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'll compute transformed keys only when they are needed. As distinct from the first technique we cannot consider the concrete key update function in isolation 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 show the following example.

Consider the CBC-MAC mode providing message authenticity. We give a rough specification of CBC-MAC: for the input message $M = M[1]\| \dots \|M[l]$, $l = |M|_n$ the authentication tag T is computed as follows:

$$T = E_K(E_K(\dots E_K(E_K(M[1]) \oplus M[2]) \dots) \oplus M[l]).$$

CBC-MAC is known to be provably secure below the birthday bound when applied to prefix free message space [4].

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:

$$K_0 = K, \qquad K_{i+1} = E_{K_i}(C_0) \oplus E_{K_i}(C_1), \quad i = 0, 1, \dots,$$

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 $K_1 = E_K(C_0) \oplus E_K(C_1)$ is $R_1\|R_2$. The known next section key allows to trivially forges long (more than section) messages. Similar attack can be applied to the OMAC mode (see [14,15,22]).

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 initialize vector. Therefore, to be convinced that the proposed internal key update function is «good» we should provide the security proof for each re-keyed mode of operation.

4.4 Internally Re-keyed GCM-ACPKM mode

In the current paper we consider certain internal re-keying technique, called ACPKM, particularly for the GCM mode. 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 [16,24] suggest that restricting GCM to 96-bit nonces is recommended from the provable security perspective as well: there is no the additional term $\frac{2^{22}q(\sigma+q)(l_{IV}+1)}{2^n}$, respected to the probability of nonce collision, in the security bound. Moreover, this security bound was shown to be tight in [24]: the proposed attack states that the constant 2^{22} cannot be made smaller than $2^{19.74}$.

Firstly, define the auxiliary function $\varphi_i : \{0,1\}^n \to \{0,1\}^n$, $\varphi_i(X) = X'$, $0 \leqslant i < n$, where $X'_{(i)} = 1$ and $X'_{(j)} = X_{(j)}$, for all $j \in \{0,\dots,n-1\}\backslash\{i\}$. This function sets the i-th bit of string to 1.

Key updating for the GCM encryption mode is as follows:

$$K_0 = K, \quad K_{i+1} = \mathrm{ACPKM}(K_i) = E_{K_i}(\varphi_{n-32}(D_1))\| \dots \|E_{K_i}(\varphi_{n-32}(D_{k/n})),$$

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

We denote by GCM-ACPKM$_{E,\tau,l}$ the 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 reseted for each new section. There is a certain reason for that: in order to protect against key-collision attack (see [9]), we should provide different input blocks for encryption under different keys. The key for computing values $E_K(I)$ and $H = E_K(0^n)$ is not updated and is equal to the initial key (the reasons for that are discussed in Appendix C). 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 $I_i = \pi(IV\|\mathrm{str}_{n-96}(1), i)$, $1 \leqslant i \leqslant 2^{31}-2$. This property is provided by the restriction on the plaintext length and using of the function φ_{n-32}. 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 the more reason for considering the 96-bit nonces.

5 Analysis of the GCM-ACPKM Encryption Mode

This section contains the main results on the security of the internally re-keyed GCM-ACPKM mode. The obtained results allow to claim that this mode is

```
GCTR-ACPKM$_{E,l}(K, IV, X)$
$I = IV \| \mathrm{str}_{n-96}(1)$
$K_0 = K$
for $j = 1$ to $\lceil |X|_n / l \rceil - 1$ do
        $K_j = \mathrm{ACPKM}(K_{j-1})$
for $i = 1$ to $|X|_n$ do
        $j = \lceil (i-1)/l \rceil$
        $I_i = \pi(I, i)$
        $G_i = E_{K_j}(I_i)$
$Y = X \oplus \mathrm{msb}_{|X|}\left(G_1 \| \ldots \| G_{|X|_n}\right)$
return $Y$
```

```
GCM-ACPKM$_{E,\tau,l}$.Encrypt$(K, IV, A, M)$
$I = IV \| \mathrm{str}_{n-96}(1)$
Ciphertext computation:
    $C = \text{GCTR-ACPKM}_{E,l}(K, IV, M)$
Tag computation:
    $H = E_K(0^n)$
    $T = \mathrm{msb}_\tau\left(\mathrm{GHASH}_H(A, C) \oplus E_K(I)\right)$
return $(C, T)$
```

```
GCM-ACPKM$_{E,\tau,l}$.Decrypt$(K, IV, A, C, T)$
$I = IV \| \mathrm{str}_{n-96}(1)$
Plaintext computation:
    $M = \text{GCTR-ACPKM}_{E,l}(K, IV, C)$
Tag verification:
    $H = E_K(0^n)$
    $T' = \mathrm{msb}_\tau\left(\mathrm{GHASH}_H(A, C) \oplus E_K(I)\right)$
    if $T = T'$ then
        return $M$
    else return $\perp$
```

Fig. 3. Authenticated encryption and decryption in the GCM-ACPKM Mode.

secure if the base block cipher is secure and 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.

The main element of the proof is an IKM$_{m,l}$ (Indistinguishable Key Meshing) task with parameters $m, l \in \mathbb{N}$ for the family $\mathcal{P}$ of permutations on $\{0,1\}^n$.

Definition 1. *An IKM$_{m,l}$ task, where $m, l \in \mathbb{N}$: $ml \leqslant 2^{31} - 2$, for the family $\mathcal{P}$ of permutations on $\{0,1\}^n$ is the following decisional task. A nonce-respecting adversary $\mathcal{A}$ has access to an oracle KM (Key Meshing) or to an oracle RK (Random Key). Initially both oracles choose a permutation $P \in_{\mathcal{U}} \mathcal{P}$.*

The initial query of the adversary $\mathcal{A}$ is a value $j \in \{0, 1, \ldots, m-1\}$. This query defines what section of encrypted blocks should be returned by the oracles. In response both oracles return $H = P(0^n)$ and K', where $K' \in_{\mathcal{U}} \{0,1\}^k$ in the case of the oracle RK and $K' = P(\varphi_{n-32}(D_1)) \| \ldots \| P(\varphi_{n-32}(D_s))$, $s = k/n$, in the case of the oracle KM.

The following queries of the adversary $\mathcal{A}$ are the values IV, $|IV| = 96$, which are unique for each query. On the query IV both oracles return the string

$$P(I) \| P(I_{j \cdot l+1}) \| \ldots \| P(I_{j \cdot l+l}),$$

where $I_i = \pi(IV \| \mathrm{str}_{n-96}(1), i)$, $j \cdot l + 1 \leqslant i \leqslant j \cdot l + l$.

We define

$$\mathbf{Adv}_{\mathcal{P}}^{IKM_{m,l}}(\mathcal{A}) = \Pr\left[P \in_{\mathcal{U}} \mathcal{P} : \mathcal{A}^{KM} \Rightarrow 1\right] - \\ - \Pr\left[F \in_{\mathcal{U}} \mathcal{P}, K' \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}^{RK} \Rightarrow 1\right],$$

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

Remark 1. In the $\mathrm{IKM}_{m,l}$ task, where $\mathcal{P}$ is a cipher E, both oracles choose a key $K \in \{0,1\}^k$ and set $P = E_K$ before starting the work.

This task is introduced for analyzing the re-keyed modes without master key. Informally, the task is dedicated to answer the following question: can the adversary distinguish the next key from random one, having the result of previous section processing?

Lemma 1. *Let $m, l \in \mathbb{N}$: $ml \leqslant 2^{31} - 2$, be the parameters of the $IKM_{m,l}$ task for a cipher E. Then for any adversary $\mathcal{A}$ with at most time complexity t that makes at most q queries, there exists an adversary $\mathcal{A}'$ such that*

$$\mathbf{Adv}_E^{IKM_{m,l}}(\mathcal{A}) \leqslant 2 \cdot \mathbf{Adv}_E^{PRP\text{-}CPA}(\mathcal{A}') + \frac{2s(ql+q+1)+s^2-s}{2^{n+1}},$$

where $\mathcal{A}'$ makes at most $ql + q + s + 1$ queries, $s = k/n$. Furthermore, the time complexity of $\mathcal{A}'$ is at most $t + cn(ql + q + s + 1)$, where c is a constant that depends only on the model of computation and the method of encoding.

The proof can be found in Appendix B.

The Privacy and Authenticity notions for the GCM-ACPKM mode are defined in the same way as for the base GCM mode except for the encryption and decryption oracles. Let GCM-KM-$\mathcal{E}$ and GCM-KM-$\mathcal{D}$ be encryption and decription oracles for the re-keyed mode, then for cipher E with parameters n and k, tag size τ and section size l

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

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

Consider the main result of the paper.

Theorem 3. *Let E, τ and l be the parameters of GCM-ACPKM mode. Then for any $\mathcal{A}$ with at most time complexity t that makes at most q encryption queries, where the maximal plaintext length is at most ml blocks ($m \in \mathbb{N}$ such that $ml \leqslant 2^{31}-2$), the nonce length restricted to 96 bits, there exists an adversary $\mathcal{A}'$ such that*

$$\mathbf{Adv}^{Priv}_{GCM\text{-}ACPKM_{E,\tau,l}}(\mathcal{A}) \leqslant 3m \cdot \mathbf{Adv}_E^{PRP\text{-}CPA}(\mathcal{A}') + m \cdot \frac{(ql+q+1)^2}{2^{n+1}} + \\ + m \cdot \frac{2s(ql+q+1)+s^2-s}{2^{n+1}},$$

where $\mathcal{A}'$ makes at most $ql + q + s + 1$ queries, $s = k/n$. Furthermore, the time complexity of $\mathcal{A}'$ is $t + cn\sigma_A$, 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 2. Note that the re-keyed mode is secure if the value $s = k/n$ is rather small. For the common block ciphers (AES-256 and AES-128) this condition is satisfied: $s \in \{1, 2\}$.

The full proof can be found in Appendix B.

Proof. (sketch) The proof framework is essentially based on the task of breaking the abstract GCM-RK mode that totally coincides with the GCM-ACPKM mode but section keys are chosen uniformly at random. This task can be reduced to the task of breaking the target GCM-ACPKM mode by a standard hybrid argument (see, e.g., [1]), if the adversary cannot distinguish a section key from a random one, having the result of previous section processing (the $\mathrm{IKM}_{m,l}$ task). Note that for the GCM-ACPKM mode with nonce length restricted to 96 bits the probability of the event that blocks of the next section key appear among blocks of previous section processing is zero. In other words, a table defining a cipher function is divided into two parts of the same size: the first part is used for the encryption procedure, and k/n blocks are chosen from the second one for the next section key.

Theorem 4. *Let E, τ and l be the parameters of GCM-ACPKM mode. Then for any $\mathcal{A}$ with at most time complexity t that makes at most q encryption queries and q' decryption queries, where the maximal plaintext length is at most ml blocks ($m \in \mathbb{N}$ such that $ml \leqslant 2^{31} - 2$), the nonce length restricted to 96 bits, the maximal summary length of plaintext or ciphertext and associated data in query is at most l_A blocks, there exists an adversary $\mathcal{A}'$ such that*

$$\mathbf{Adv}^{Auth}_{GCM\text{-}ACPKM_{E,\tau}}(\mathcal{A}) \leqslant 3m \cdot \mathbf{Adv}^{PRP\text{-}CPA}_{E}(\mathcal{A}') + m \cdot \frac{(ql+q+q'+1)^2}{2^{n+1}} + \\ + m \cdot \frac{2s(ql+q+q'+1)+s^2-s}{2^{n+1}} + \frac{q'(l_A+1)}{2^{\tau}},$$

where $\mathcal{A}'$ makes at most $ql+q+q'+s+1$ queries. Furthermore, the time complexity of $\mathcal{A}'$ is $t + cn\sigma_A$, 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.

Proof. (sketch) The first two terms in the bound are obtained by the same way as for the Privacy notion, considering reduction of the abstract GCM-RK mode to the target GCM-ACPKM mode. The only difference is the necessity for taking into account q' decryption queries. The last term $\frac{q'(l_A+1)}{2^{\tau}}$ that is respected to the forgery probability totally coincides with the associated term for the base GCM mode. The values $E_K(I)$ and $H = E_K(0^n)$ are not changed therefore this probability is obtained by the same way as described in [16]. The reasons for the proposed re-keying technique are in the fact, that changing of H actually degrades the security estimation (see Appendix C for more details).

5.1 Security

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

Note that the bounds in Section 3 for the base GCM mode are obtained in the term of the total plaintext length σ. At the same time bounds for the GCM-ACPKM mode are expressed in the term of the maximal plaintext length since for such internally re-keyed mode the number of sections which plaintext can consist of essentially influences on the security estimation. In order to compare the security of the base and internally re-keyed GCM modes we assume that the maximal plaintext length is at most ml blocks and $\sigma \leqslant qml$.

The σ model is a good way for further research and deserves a separate paper (see, e.g., [14,15]). Our results on qml are suitable for a large set of (typical) cases with small dispersion of lengths.

We assume that for the used cipher E the inequalities (1) hold and the obtained bounds for the GCM and GCM-ACPKM modes are tight. We also assume that $2^k \gg 2^n$.

If $qml + q + 1 < 2^{n/2}$ and $t \ll 2^k$ then for any adversary $\mathcal{A}$ with the time complexity at most t that makes at most q queries where the maximal plaintext length is at most $ml \leqslant 2^{31} - 2$ blocks and the nonce length is restricted to 96 bits

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

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

These relations indicate that the security of the GCM-ACPKM mode is almost quadratically in m improved compared to the security of the base GCM mode for the Privacy notion.

For the same reasons for any adversary $\mathcal{A}$ with the time complexity at most t that makes at most q encryption queries and q' decryption queries where the maximal plaintext length is at most $ml \leqslant 2^{31} - 2$ blocks, the nonce length is restricted to 96 bits and the maximal summary length of plaintext or ciphertext and associated data in query is at most l_A,

$$\mathbf{Adv}^{\text{Auth}}_{\text{GCM}_{E,\tau}}(\mathcal{A}) \approx \frac{(qml + q + q' + 1)^2}{2^n} + \frac{q'(l_A + 1)}{2^\tau},$$

$$\mathbf{Adv}^{\text{Auth}}_{\text{GCM-ACPKM}_{E,\tau,l}}(\mathcal{A}) \approx \frac{m(ql + q + q' + 1)^2}{2^n} + \frac{q'(l_A + 1)}{2^\tau}.$$

The authenticity security of the GCM-ACPKM mode is essentialy improved compared to the security of the base GCM mode in the most typical cases when the length of associated data is much less than the plaintext length and $\tau = n$ (e.g. a header in TLS).

We focus on limiting key exposure, not increasing authenticity bounds: to fight against adversary having collected enough data (e.g. by side-channels) and having spent a year of calculations — after that year we would be much more afraid (e.g. for TLS, IPsec) of compromise of those messages, not forgeries. Therefore we state that the ACPKM technique increases the overall security of the GCM AEAD mode.

5.2 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 mode with the following ciphers: hardware-supported AES-256 and AES-128 (using OpenSSL source [29]). 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 amount of separate processed messages, l is the section size) should be no greater than the lifetime of a key.

6 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 amount of separate messages. There is no technique, which is more preferable because the choice of technique can depend on protocol features. For example, for protocols which allow out-of-order delivery and lost records (e.g., [30,31]), 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., [32,33]), internal re-keying is better suited. In this section we consider the composition of external and internal re-keying techniques (see Figure 4) since these techniques negate each other's disadvantages.

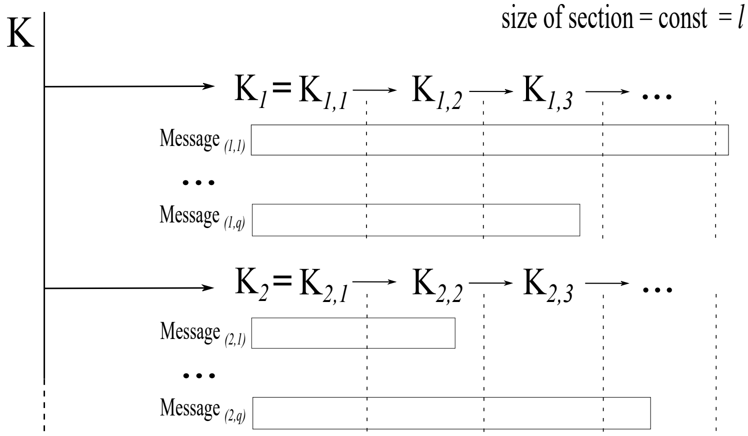
K
size of section = const = l
K1 = K1,1 → K1,2 → K1,3 → ...
Message (1,1)
...
Message (1,q)
K2 = K2,1 → K2,2 → K2,3 → ...
Message (2,1)
...
Message (2,q)

Fig. 4. Composition of Internal and External Re-keying. Here $K_1, K_2, \ldots$ are diversified from a master key K. Then each K_i is used for processing q messages in the internally re-keyed mode with the section size l.

Consider the security bounds for GCM, GCM-ACPKM, key diversified GCM ($\overline{\text{GCM}}$) and key diversified GCM-ACPKM ($\overline{\text{GCM-ACPKM}}$) for the Privacy notion. 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 5. *[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 the time complexity at most t that makes at most Nq encryption queries, where the maximal plaintext length is at most M blocks, there exist adversaries $\mathcal{A}'$ and $\mathcal{A}''$ such that*

$$\mathbf{Adv}^{Priv}_{\overline{\mathcal{SE}}^q}(\mathcal{A}) \leqslant 2 \cdot \mathbf{Adv}^{PRG}_{\mathcal{G},N}(\mathcal{A}') + N \cdot \mathbf{Adv}^{Priv}_{\mathcal{SE}}(\mathcal{A}''),$$

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.

If we assume the approximations considered in Section 5.1 for the adversary $\mathcal{A}$ that makes at most Q queries where the maximal plaintext length is at most M blocks, then we get the approximations, presented in Table 3.

$\mathcal{SE}$	$\mathbf{Adv}^{\text{Priv}}_{\mathcal{SE}}(\mathcal{A})$
$\text{GCM}_{E,\tau}$	$\approx \dfrac{(QM+Q+1)^2}{2^n}$
$\overline{\text{GCM}_{E,\tau}}^q$	$\approx \dfrac{Q}{q} \cdot \dfrac{(qM+q+1)^2}{2^n}$
$\text{GCM-ACPKM}_{E,\tau,l}$	$\approx \dfrac{M}{l} \cdot \dfrac{(Ql+Q+1)^2}{2^n}$
$\overline{\text{GCM-ACPKM}_{E,\tau,l}}^q$	$\approx \dfrac{QM}{ql} \cdot \dfrac{(ql+q+1)^2}{2^n}$

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 maximal plaintext length in blocks, q (subkey lifetime) and l (section size) are parameters of the external and internal re-keying techniques.

To compare the GCM modifications fix a safety margin δ of security which allows to process $Q = 2^{20}$ messages with length at most $M = 2^{20}$ blocks in the base GCM mode. Thus, the total amount of data processed with initial key is 2^{40} blocks. Let set the optimal parameters for the re-keyed modes: $q = 2^5$ messages per subkey for external re-keying and section size $l = 2^5$ blocks for internal re-keying. According to the approximate security bounds presented in the Table 3 we can securely increase the amount of messages $c_1 = 32766$ times by external re-keying, the message length $c_2 = 30812$ times by internal re-keying and the total amount of data $c = 1007744964 \approx 10^9$ times by both techniques. The values c_1, c_2, c are obtained due to the following relations:

- for the key diversified $\overline{\text{GCM}_{E,\tau}}^q$ mode

$$\frac{c_1 Q}{q} \cdot \frac{(qM+q+1)^2}{2^n} = \frac{(QM+Q+1)^2}{2^n} = \delta \Longrightarrow c_1 = \left(\frac{QM+Q+1}{qM+q+1}\right)^2 \cdot \frac{q}{Q};$$

- for the internally re-keyed $\text{GCM-ACPKM}_{E,\tau,l}$ mode

$$\frac{c_2 M}{l} \cdot \frac{(Ql+Q+1)^2}{2^n} = \frac{(QM+Q+1)^2}{2^n} = \delta \Longrightarrow c_2 = \left(\frac{QM+Q+1}{Ql+Q+1}\right)^2 \cdot \frac{l}{M};$$

- for the key diversified internally re-keyed $\overline{\text{GCM-ACPKM}_{E,\tau,l}}^q$ mode

$$\frac{cQM}{ql} \cdot \frac{(ql+q+1)^2}{2^n} = \frac{(QM+Q+1)^2}{2^n} = \delta \Longrightarrow c = \left(\frac{QM+Q+1}{ql+q+1}\right)^2 \cdot \frac{ql}{QM}.$$

Now consider the AES-GCM mode and AES-$\overline{\text{GCM-ACPKM}}_{E,\tau,l}^{\,q}$ with parameters $n = 128$, $q = 2^{15}$ and $l = 2^{6}$. Let compare key lifetime limitations for these modes in TLS 1.3 protocol [26] where record size is at most 2^{10} blocks or 2^{14} B. Technically, AES-$\overline{\text{GCM-ACPKM}}$ in TLS 1.3 assumes that the initial key should be diversified after every gigabyte and every subkey should be internally updated after every kilobyte. The results are presented in Table 4, where the first column contains certain success probabilities for a privacy attack, the second column is taken from Table 1, [18] and the third column is computed according to the last relation.

δ	Max Records	
	GCM	$\overline{\text{GCM-ACPKM}_l}^{\,q}$
2^{-60}	$2^{24.5}$	$2^{36.9}$
2^{-50}	$2^{29.5}$	$2^{46.9}$
2^{-40}	$2^{34.5}$	$2^{56.9}$
2^{-30}	$2^{39.5}$	$2^{66.9}$
2^{-20}	$2^{44.5}$	$2^{76.9}$

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

7 Open Problems

We see the following interesting open problems:

1. For the CBC, CFB, OFB encryption modes extended by the ACPKM technique the probability of trivial breaking the next section key is negligible but not zero: IV is chosen uniformly at random and can simply coincide with an open input block used in the re-keying technique. As a result the adversary has an opportunity to reveal blocks of the next key. Therefore it is interesting to modify the proposed internal re-keying technique respecting the considered modes features and to analyze its security properties.
2. Another interesting 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 [21]), 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.

8 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 notion is beyond the birthday bound since it is quadratically (in a term of the amount of sections) increased compared to the base mode. The authenticity security of GCM-ACPKM is also improved. Therefore we stress that the overall security of GCM is drastically increased by the ACPKM re-keying technique with only a minor loss in performance.

The proof framework proposed in this paper is useful to obtain security bounds for other re-keyed modes of operation which do not use additional secret values (without master key).

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.

Acknowledgement. The authors are very grateful to Nikolay P. Varnovsky, Ekaterina S. Smyshlyaeva, and Lolita A. Sonina for their valuable comments and suggestions concerning the text of the article.

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. 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.
4. 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.
5. Bellare, M. , Rogaway, P. Introduction to modern cryptography, 2005.
URL: http://cseweb.ucsd.edu/~mihir/cse207/classnotes.html.

6. Bellare, M. Practice-Oriented Provable-Security. Modern Cryptology in Theory and Practice, P. 1-15, 1999.
7. 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.
8. 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.
9. Biham, E. How to Forge DES-Encrypted Messages in 2^{28} Steps. Technion Computer Science Department Technical Report CS0884, 1996.
10. Biham, E., Shamir, A. Differential Cryptanalysis of DES-like Cryptosystems // Journal of Cryptology. V. 537. P. 2-21. 1990.
11. 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.
12. 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.
13. Chang, D., Nandi, M. A Short Proof of the PRP/PRF Switching Lemma. IACR Cryptology ePrint Archive, 2008:078, 2008.
14. Iwata, T., Kurosawa, K. OMAC: One-Key CBC MAC. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 129–153. Springer, Heidelberg, 2003.
15. 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.
16. Iwata,T., Ohashi, K., Minematsu K. Breaking and Repairing GCM Security Proofs. CRYPTO 2012, LNCS, vol. 7417, pp. 31-49. Springer, Heidelberg, 2012.
17. Rogaway, P. Nonce-Based Symmetric Encryption. The 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004.
18. 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
19. 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.
20. 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.
21. Micali, S., Reyzin, L. Physically Observable Cryptography (extended abstract). TCC 2004, LNCS, vol. 2951, pp. 278–296.
22. 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.
23. Chen, L. NIST Special Publication 800-108. Recommendation for Key Derivation Using Pseudorandom Functions (Revised). 2009.
24. 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.
25. 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.

26. Rescorla, E., RTFM, Inc. The Transport Layer Security (TLS) Protocol Version 1.3, draft-ietf-tls-tls13-20, April 28, 2017.
27. 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.
28. Ramsay C., Lohuis J. «TEMPEST attacks against AES. Covertly stealing keys for € 200», https://www.fox-it.com, 2017.
29. URL: https://www.openssl.org/
30. Rescorla E. and Modadugu N. «Datagram Transport Layer Security Version 1.2», RFC 6347, DOI 10.17487/RFC6347, January 2012,
31. Kent S. «IP Encapsulating Security Payload (ESP)», RFC 4303, DOI 10.17487/RFC4303, December 2005,
32. Dierks T. and Rescorla E. «The Transport Layer Security (TLS) Protocol Version 1.2», RFC 5246, DOI 10.17487/RFC5246, August 2008,
33. Ylonen T. and Lonvick C., Ed., «The Secure Shell (SSH) Transport Layer Protocol», RFC 4253, DOI 10.17487/RFC4253, January 2006,

A Additional notations

The Privacy and Authenticity notions for the GCM-RK mode (with section random keys) are defined in the same way as for the GCM-ACPKM mode except for the encryption and decryption oracles. :

$$\mathbf{Adv}^{\mathrm{Priv}}_{\text{GCM-RK}_{E,\tau,l}}(\mathcal{A}) = \Pr\left[K_0, \ldots \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}^{\text{GCM-RK-}\mathcal{E}} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\$} \Rightarrow 1\right],$$

$$\mathbf{Adv}^{\mathrm{Auth}}_{\text{GCM-RK}_{E,\tau,l}}(\mathcal{A}) = \Pr\left[K_0, \ldots \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}^{\text{GCM-RK-}\mathcal{E},\text{GCM-RK-}\mathcal{D}} \text{ forges}\right].$$

Here GCM-RK-$\mathcal{E}$ and GCM-RK-$\mathcal{D}$ are encryption and decryption oracles which process adversary's queries according to the GCM-RK mode.

B Proof of Theorem 3

Proof of Lemma 1

Proof. Firstly, we consider an information-theoretic lemma on which Lemma 1 relies.

Lemma 2. *Let $m, l \in \mathbb{N}$: $ml \leqslant 2^{31} - 2$, be the parameters of the $IKM_{m,l}$ task for a set of all permutations $Perm(\{0,1\}^n)$. Then for any adversary $\mathcal{A}$ which makes at most q queries,*

$$\mathbf{Adv}^{IKM_{m,l}}_{Perm(\{0,1\}^n)}(\mathcal{A}) \leqslant \frac{2s(ql+q+1)+s^2-s}{2^{n+1}},$$

where $s = k/n$.

Proof. We denote by $\overline{D}$ the set $\{\varphi_{n-32}(D_1), \varphi_{n-32}(D_2), \ldots, \varphi_{n-32}(D_s)\}$.

We denote by $\mathcal{N}$ all information that the adversary obtains during attack. For any allowed adversary's queries j, $IV^1, \ldots, IV^q$ the value $\mathcal{N}$ is determined by the following values:

– a string $K' \in \{0,1\}^k$ (we assume that $\{K'\}$ is a set that consists of $s = k/n$ blocks of the string K').
– a set $\{P(IV)\} = \bigcup_{i=1}^{q}\{P(I^i), P(I^i_{jl+1}), \ldots, P(I^i_{jl+l})\} \cup P(0^n)$, where $I^i = IV^i \| \mathrm{str}_{n-96}(1)$, such that $|\{P(IV)\} \cap \{P(\overline{D})\}| = 0$.

Note that if the value $\mathcal{N} = (\{P(IV)\}, \{K'\})$ is fixed, then

$$\Pr\left[\mathcal{A}^{KM} \Rightarrow 1 | \mathcal{N}\right] = \Pr\left[\mathcal{A}^{RK} \Rightarrow 1 | \mathcal{N}\right].$$

We denote this probability by $\Pr\left[\mathcal{A} \Rightarrow 1|\mathcal{N}\right]$. By the law of total probability we have

$$\begin{aligned}\mathbf{Adv}^{\mathrm{IKM}_{m,l}}_{Perm(\{0,1\}^n)}(\mathcal{A}) &= \Pr\left[P \in_{\mathcal{U}} Perm(\{0,1\}^n) : \mathcal{A}^{KM} \Rightarrow 1\right] - \\ &- \Pr\left[P \in_{\mathcal{U}} Perm(\{0,1\}^n), K' \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}^{RK} \Rightarrow 1\right] = \\ &= \sum_{\mathcal{N}} \Pr\left[\mathcal{A} \Rightarrow 1|\mathcal{N}\right] \cdot \Pr\left[\mathcal{N}|KM\right] - \sum_{\mathcal{N}} \Pr\left[\mathcal{A} \Rightarrow 1|\mathcal{N}\right] \cdot \Pr\left[\mathcal{N}|RK\right] = \\ &= \sum_{\mathcal{N}} \underbrace{\Pr\left[\mathcal{A} \Rightarrow 1|\mathcal{N}\right]}_{\leqslant 1} \cdot \left(\Pr\left[\mathcal{N}|KM\right] - \Pr\left[\mathcal{N}|RK\right]\right) \leqslant \\ &\leqslant \sum_{\mathcal{N} : \Pr[\mathcal{N}|KM] - \Pr[\mathcal{N}|RK] > 0} \left(\Pr\left[\mathcal{N}|KM\right] - \Pr\left[\mathcal{N}|RK\right]\right)\end{aligned}$$

If the adversary interacts with the RK oracle then all components of the value $\mathcal{N}$ are chosen independently of each other, therefore:

$$\Pr\left[\mathcal{N}|RK\right] = \frac{(2^n - (ql + q + 1))!}{2^n!} \cdot \frac{1}{2^{sn}}.$$

Consider the probability $\Pr\left[\mathcal{N}|KM\right]$. Note that

$$\Pr\left[\mathcal{N}|KM\right] = \begin{cases} \dfrac{(2^n - (ql + q + s + 1))!}{2^n!}, & \text{if } |\{P(IV)\} \cap \{K'\}| = 0, \\ 0, & \text{otherwise.}\end{cases}$$

Note that if $\Pr\left[\mathcal{N}|KM\right] > 0$ then $\Pr\left[\mathcal{N}|KM\right] > \Pr\left[\mathcal{N}|RK\right]$. Indeed,

$$\Pr\left[\mathcal{N}|KM\right] = \Pr\left[\mathcal{N}|RK\right] \cdot \frac{2^{sn} \cdot (2^n - (ql + q + s + 1))!}{(2^n - (ql + q + 1))!} \geqslant \Pr\left[\mathcal{N}|RK\right].$$

Therefore,

$$\begin{aligned}\mathbf{Adv}^{\mathrm{IKM}_{m,l}}_{Perm(\{0,1\}^n)}(\mathcal{A}) &= \sum_{\mathcal{N} : \Pr[\mathcal{N}|KM] > 0} \left(\Pr\left[\mathcal{N}|KM\right] - \Pr\left[\mathcal{N}|RK\right]\right) \leqslant \\ &\leqslant |\{\mathcal{N} : \Pr\left[\mathcal{N}|KM\right] > 0\}| \cdot \left(\Pr\left[\mathcal{N}|KM\right] - \Pr\left[\mathcal{N}|RK\right]\right).\end{aligned}$$

Finally, we obtain the following inequalities:

$$\begin{aligned}
\mathbf{Adv}_{Perm(\{0,1\}^n)}^{\mathrm{IKM}_{m,l}}(\mathcal{A}) \leqslant \frac{2^n!}{(2^n-(ql+q+s+1))!} \cdot \Big[\frac{(2^n-(ql+q+s+1))!}{2^n!} - \\
- \frac{(2^n-(ql+q+1))!}{2^{sn}\cdot 2^n!}\Big] = 1 - \prod_{i=0}^{s-1}\left(1 - \frac{(ql+q+1)+i}{2^n}\right) \leqslant \\
\leqslant \sum_{i=0}^{s-1} \frac{(ql+q+1)+i}{2^n} = \frac{2s(ql+q+1)+s^2-s}{2^{n+1}}.
\end{aligned}$$

□

Now consider an adversary $\mathcal{A}'$ in the PRP-CPA$_{m,l}$ notion for a cipher E. He intercepts all queries of the adversary $\mathcal{A}$ in the IKM$_{m,l}$ task for E. Receiving from $\mathcal{A}$ the first query $j \in \{0,1,\ldots,m-1\}$, the adversary $\mathcal{A}'$ remembers the value j, chooses a bit $b \in_{\mathcal{U}} \{0,1\}$ and returns K' obtained according to the ACPKM technique using his oracle, if $b=1$, and $K' \in_{\mathcal{U}} \{0,1\}^k$, if $b=0$.

Next queries of the adversary $\mathcal{A}$ are processed by making the queries 0^n, I^i and I^i_k, $1 \leqslant i \leqslant q$, $jl+1 \leqslant k \leqslant jl+l$, to the oracle of $\mathcal{A}'$.

Suppose that the adversary $\mathcal{A}$ returns a bit a as a result. The adversary $\mathcal{A}'$ returns 1, if $a=b$, and 0, otherwise.

Using the Lemma 2 for the advantage of the adversary $\mathcal{A}'$ we have

$$\begin{aligned}
&\mathbf{Adv}_E^{\text{PRP-CPA}}(\mathcal{A}') = \\
&\quad = \Pr\left[K \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}'^{E_K} \Rightarrow 1\right] - \Pr\left[P \in_{\mathcal{U}} Perm(\{0,1\}^n) : \mathcal{A}'^{P} \Rightarrow 1\right] \\
&\qquad = 1/2 \cdot \mathbf{Adv}_E^{\mathrm{IKM}_{m,l}}(\mathcal{A}) - 1/2 \cdot \mathbf{Adv}_{Perm(\{0,1\}^n)}^{\mathrm{IKM}_{m,l}}(\mathcal{A}) \geqslant \\
&\qquad\quad = 1/2 \cdot \mathbf{Adv}_E^{\mathrm{IKM}_{m,l}}(\mathcal{A}) - 1/2 \cdot \left(\frac{2s(ql+q+1)+s^2-s}{2^{n+1}}\right).
\end{aligned}$$

□

Proof of Theorem 3

Proof. Firstly, obtain the lower security bound of the GCM-RK mode.

Lemma 3. *Let E, τ and l be the parameters of the GCM-RK mode. Then for any adversary $\mathcal{A}$ with at most time complexity t that makes at most q encryption queries, where the maximal plaintext length is at most ml blocks ($m \in \mathbb{N}$ such that $ml \leqslant 2^{31}-2$) and the nonce length restricted to 96 bits, there exists an adversary $\mathcal{A}'$ such that*

$$\mathbf{Adv}_{\textit{GCM-RK}_{E,\tau,l}}^{\textit{Priv}}(\mathcal{A}) \leqslant m \cdot \mathbf{Adv}_E^{\textit{PRP-CPA}}(\mathcal{A}') + m \cdot \frac{(ql+q+1)^2}{2^{n+1}},$$

where $\mathcal{A}'$ makes at most $ql+q+1$ queries. Furthermore, the time complexity of $\mathcal{A}'$ is $t + cn\sigma_A$, 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.

Proof. Define a set of the hybrid experiments $\{Hybrid_{\mathcal{A},j}\}$ for $j \in \{0, 1, \ldots, m\}$. In the experiment $Hybrid_{\mathcal{A},j}$ the oracle in the Privacy notion is replaced by the oracle that operates in the following way:

- The oracle chooses j keys $K_0, K_1, \ldots, K_{j-1} \in_{\mathcal{U}} \{0,1\}^k$ independently of each other;
- In response to a query (IV, A, M) the oracle returns a pair (C, T), where $C = M \oplus \mathrm{msb}_{|M|}(C^{[0]} \| \ldots \| C^{[j-1]} \| C')$, here

  $$C^{[i]} = (E_{K_i}(I_{i \cdot l+1}) \| \ldots \| E_{K_i}(I_{i \cdot l+l})),$$

  for $0 \leqslant i \leqslant j-1$, $I_k = \pi(IV \| \mathrm{str}_{n-96}(1), k)$, $i \cdot l + 1 \leqslant k \leqslant i \cdot l + l$, and $C' \in_{\mathcal{U}} \{0,1\}^{(m-j)ln}$. The authentication tag

  $$T = \mathrm{msb}_\tau(Z \oplus \mathrm{GHASH}_H(A, C)),$$

  where $Z, H \in_{\mathcal{U}} \{0,1\}^n$ if $j = 0$, and $Z = E_{K_0}(I_1)$, $H = E_{K_0}(0^n)$, otherwise.

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

Note that for the adversary $\mathcal{A}$ the oracle in the experiment $Hybrid_{\mathcal{A},0}$ totally coincides with the oracle \$, and the oracle in the experiment $Hybrid_{\mathcal{A},m}$ coincides with the oracle GCM-RK-$\mathcal{E}$, i.e. the following inequalities hold:

$$\Pr\left[Hybrid_{\mathcal{A},0} \Rightarrow 1\right] = \Pr\left[\mathcal{A}^{\$} \Rightarrow 1\right],$$

$$\Pr\left[Hybrid_{\mathcal{A},m} \Rightarrow 1\right] = \Pr\left[K_0, \ldots, K_{m-1} \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}^{\text{GCM-RK-}\mathcal{E}} \Rightarrow 1\right].$$

Construct an adversary $\mathcal{A}'$ for the PRF notion which uses A as a black box.

The adversary $\mathcal{A}'$ chooses $j \in_{\mathcal{U}} \{0, \ldots, m-1\}$ and j keys $K_0, \ldots, K_{j-1} \in_{\mathcal{U}} \{0,1\}^k$. After receiving a query (IV, A, M) from $\mathcal{A}$ the adversary $\mathcal{A}'$ processes this query as in the $Hybrid_{\mathcal{A},j}$ experiment but the encrypted blocks for masking the j-th section are obtained by making queries to the oracles F or E_K. Note that the adversary $\mathcal{A}'$ makes at most $ql + q + 1$ queries. The adversary $\mathcal{A}'$ returns as a result what the adversary $\mathcal{A}$ returns.

Note that

$$\Pr\left[F \in_{\mathcal{U}} Func(\{0,1\}^n) : \mathcal{A}'^F = 1\right] = \frac{1}{m} \sum_{j=0}^{m-1} \Pr\left[Hybrid_{\mathcal{A},j} \Rightarrow 1\right],$$

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

Then for the advantage of the adversary $\mathcal{A}'$

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

From the PRP/PRF switching lemma [13] we have

$$\mathbf{Adv}_{\text{GCM-RK}_{E,\tau,l}}^{\mathrm{Priv}}(\mathcal{A}) \leqslant m \cdot \mathbf{Adv}_E^{\text{PRP-CPA}}(\mathcal{A}') + m \cdot \frac{(ql+q+1)^2}{2^{n+1}}.$$

□

Now consider the Privacy notion for the GCM-ACPKM mode. Determine a set of the hybrid experiments $\{Hybrid_{\mathcal{A},j}\}$ for an adversary $\mathcal{A}$, where $j \in \{0,1,\ldots,m\}$. In the experiment $Hybrid_{\mathcal{A},j}$ the oracle is replaced in the following way. In response to the query (IV, A, M) the output (C, T) is constructed as follows: the first j sections of the message M are processed with the random and independent keys $K_0, \ldots, K_{j-1}$, the key for processing the j-th section is generated at random too, but keys for the next sections are produced from previous one according to the ACPKM technique. The result of the experiment $Hybrid_{\mathcal{A},j}$ is 1, if the result of the adversary A is equal to 1.

```
GCM-Hyb-$\mathcal{E}(j)$
Initialization:
    $K_0, \ldots, K_j \in_{\mathcal{U}} \{0,1\}^k$
    for $i = j+1$ to $m-1$ do
        $K_i = \mathrm{ACPKM}(K_{i-1})$
GCM-Hyb-$\mathcal{E}(IV, A, M)$
$I = IV \| \mathrm{str}_{n-96}(1)$
Ciphertext computation:
    for $i = 1$ to $|M|_n$ do
        $j = \lceil (i-1)/l \rceil$
        $I_i = \pi(I, i)$
        $G_i = E_{K_j}(I_i)$
    $Y = X \oplus \mathrm{msb}_{|M|}\left(G_1 \| \ldots \| G_{|M|_n}\right)$
Tag computation:
    $H = E_{K_0}(0^n)$
    $T = \mathrm{msb}_\tau\left(\mathrm{GHASH}_H(A, C) \oplus E_{K_0}(I)\right)$
return $(C, T)$
```

```
$Hybrid_{\mathcal{A},j}$
GCM-Hyb-$\mathcal{E}(j)$
$a \Leftarrow \mathcal{A}^{\text{GCM-Hyb-}\mathcal{E}}$
return $a$
```

Fig. 5. Hybrid experiments for the GCM-ACPKM mode.

Note that

$$\Pr\left[Hybrid_{\mathcal{A},0} \Rightarrow 1\right] = \Pr\left[K \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}^{\text{GCM-KM-}\mathcal{E}} \Rightarrow 1\right],$$

$$\Pr\left[Hybrid_{\mathcal{A},m} \Rightarrow 1\right] = \Pr\left[K_0, \ldots, K_{m-1} \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}^{\text{GCM-RK-}\mathcal{E}} \Rightarrow 1\right],$$

Construct the adversary $\mathcal{A}'$. At the beginning he chooses $j \in_{\mathcal{U}} \{0, \ldots, m-1\}$, then he makes a query j to his oracle, receiving the key K' in response.

Then $\mathcal{A}'$ chooses j keys $K_0, \ldots, K_{j-1} \in_{\mathcal{U}} \{0,1\}^k$ independently. Intercepting the query (IV, A, M) from $\mathcal{A}$, the adversary $\mathcal{A}'$ makes the query IV to his oracle. In response he receives the section of encrypted blocks which are generated using IV and some secret key K (used by this oracle). Note that the returned section of encrypted blocks is appropriate to encrypt the j-th section of processed message M and then to compute a tag T.

The adversary $\mathcal{A}$ processes the first j sections of the message M using the keys $K_0, \ldots, K_{j-1}$ and IV, which is obtained from the adversary $\mathcal{A}'$ previously. He processes the $j+1$-th section with a key $K_{j+1} = K'$, and the next sections are processed with the keys $K_{j+2}, \ldots, K_{m-1}$ such that $K_i = \text{ACPKM}(K_{i-1})$. The adversary $\mathcal{A}'$ returns as a result what the adversary $\mathcal{A}$ returns.

Note that

$$\Pr\left[K \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}'^{KM} \Rightarrow 1\right] = \frac{1}{m} \sum_{j=0}^{m-1} \Pr\left[Hybrid_{\mathcal{A},j} \Rightarrow 1\right],$$

$$\Pr\left[K \in_{\mathcal{U}} \{0,1\}^k, K' \in_{\mathcal{U}} \{0,1\}^k : \mathcal{A}'^{RK} \Rightarrow 1 | j\right] = \frac{1}{m} \sum_{j=0}^{m-1} \Pr\left[Hybrid_{\mathcal{A},j+1} \Rightarrow 1\right].$$

This, for the advantage $\mathbf{Adv}_E^{\text{IKM}_{m,l}}(\mathcal{A}')$ we have

$$\begin{aligned}\mathbf{Adv}_E^{\text{IKM}_{m,l}}(\mathcal{A}') &= \frac{1}{m} \sum_{j=0}^{m-1} \left(\Pr\left[Hybrid_{\mathcal{A},j} \Rightarrow 1\right] - \Pr\left[Hybrid_{\mathcal{A},j+1} \Rightarrow 1\right]\right) = \\ &= \frac{1}{m} \cdot \left(\Pr\left[Hybrid_{\mathcal{A},0} \Rightarrow 1\right] - \Pr\left[Hybrid_{\mathcal{A},m} \Rightarrow 1\right]\right) = \\ &= \frac{1}{m} \cdot \left(\mathbf{Adv}^{\text{Priv}}_{\text{GCM-ACPKM}_{E,\tau,l}}(\mathcal{A}) - \mathbf{Adv}^{\text{Priv}}_{\text{GCM-RK}_{E,\tau,l}}(\mathcal{A})\right).\end{aligned}$$

Using the Lemma 1 and the Lemma 3 we obtained the target estimation. □

C Proof of Theorem 4

Proof. For the proposed GCM-ACPKM mode the proof of security in the Authenticity model is the same as described in [16]. Indeed, the ACPKM technique influences on the plaintext encryption only and does not change the tag computation. Therefore the term $m \cdot \frac{2s(ql+q+q'+1)+s^2-s}{2^{n+1}}$ respected to $\text{IKM}_{m,l}$ task for $Perm(\{0,1\}^n)$ in the proposed estimation is obtained as for Privacy but considering the decryption queries. Following [16], we do not take into account the decryption of ciphertexts since the returned plaintext has no effect on success probability of the adversary and we account for only q' encryptions $E_K(I)$ needed for tag computation. Thus, a task to estimate security of

GCM-ACPKM$_{E,\tau,l}$ for the Authenticity notion is replaced by the same task for GCM-RK$_{E,\tau,l}$ mode. From the technique of hybrid argument we have a term $m \cdot \frac{(ql+q+q'+s+1)^2}{2^{n+1}}$ for the GCM-RK$_{E,\tau,l}$ where E is replaced by m random functions $F_0, \ldots, F_{m-1} \in_{\mathcal{U}} Func(\{0,1\}^n)$ and $H = F_0(0^n)$.

Consider the forgery probability. According to [16] this term is determined by the probability of the event

$$T \oplus T' = \mathrm{msb}_\tau(\mathrm{GHASH}_\mathrm{H}(A, C) \oplus \mathrm{GHASH}_\mathrm{H}(A', C')),$$

where (IV, A, C, T) is a decryption query and (C', T') was previously returned to the adversary from the encryption oracle for a query (IV, A', M'). Note that IV for both queries is the same. The probability of this event is over the random choice of $H \in_{\mathcal{U}} \{0,1\}^n$ and depends on a number of solutions for the specified above equation. Therefore the term respected to the forgery probability over the random choice of H is $\frac{q'(l_A+1)}{2^\tau}$, where l_A is the maximal summary length of plaintext or ciphertext and associated data in query. □

Now consider the modified GCM-ACPKM mode were H is changed for every new section of plaintext: for a query (IV, A, M) tag

$$T = E_{K_0}(IV \| \mathrm{str}_{n-96}(1)) \oplus \mathrm{GHASH}_{H_0,\ldots,H_{m-1}}(A, C),$$

where C is a ciphertext of M in the GCM-ACPKM mode, $H_i = E_{K_i}(0^n)$ for $0 \leqslant i \leqslant m-1$ and $\mathrm{GHASH}_{H_0,\ldots,H_{m-1}}(A, C)$ is computed as follows. For an integer $0 \leqslant r \leqslant m$ and a ciphertext $C = C^{[0]} \| \ldots \| C^{[r-1]}$, $|C^{[i]}| = ln$, $0 \leqslant i \leqslant r-2$ and $0 \leqslant |C^{[r-1]}| \leqslant ln$ we have

$$\begin{cases} \mathrm{GHASH}_{H_0}(A, C^{[0]}) = Y_0; \\ \mathrm{GHASH}_{H_1}(Y_0, C^{[1]}) = Y_1; \\ \ldots \\ \mathrm{GHASH}_{H_{r-2}}(Y_{r-3}, C^{[r-2]}) = Y_{r-2}; \\ \mathrm{msb}_\tau(\mathrm{GHASH}_{H_{r-1}}(Y_{r-2}, C^{[r-1]}) = T. \end{cases}$$

Note that the H_0 value for the associated data A is constant since the security in the Privacy notion must not depend on length of A. A number of solutions for an equation

$$T \oplus T' = \mathrm{msb}_\tau(\mathrm{GHASH}_{H_0,\ldots,H_{m-1}}(A, C) \oplus \mathrm{GHASH}_{H_0,\ldots,H_{m-1}}(A', C')),$$

is $2^{n-\tau}2^{n(r-1)}(l_a + l + 1)(l+2)^{r-1}$, where l_a is the maximal length of associated data in query. Thus, the forgery probability over the random choice of $H_0, \ldots, H_{m-1}$ is at most $\frac{(l_a+l+1)(l+2)^{m-1}}{2^\tau}$ which is worse than initial one.