GCM Security Bounds Reconsidered

Yuichi Niwa[1], Keisuke Ohashi[1], Kazuhiko Minematsu[2], and Tetsu Iwata[1(✉)]

[1] Nagoya University, Nagoya, Japan
{y_niwa,k_oohasi}@echo.nuee.nagoya-u.ac.jp, iwata@cse.nagoya-u.ac.jp
[2] NEC Corporation, Tokyo, Japan
k-minematsu@ah.jp.nec.com

Abstract. A constant of 2^{22} appears in the security bounds of the Galois/Counter Mode of Operation, GCM. In this paper, we first develop an algorithm to generate nonces that have a high counter-collision probability. We show concrete examples of nonces with the counter-collision probability of about $2^{20.75}/2^{128}$. This shows that the constant in the security bounds, 2^{22}, cannot be made smaller than $2^{19.74}$ if the proof relies on "the sum bound." We next show that it is possible to avoid using the sum bound, leading to improved security bounds of GCM. One of our improvements shows that the constant of 2^{22} can be reduced to 32.

Keywords: GCM · Provable security · Counter-collision · The sum bound

1 Introduction

The Galois/Counter Mode of Operation, GCM, is a widely deployed authenticated encryption scheme. It was designed by McGrew and Viega [18,19] in 2004, and has been adopted by NIST as the recommended blockcipher mode of operation in 2007 [7]. A large number of standards include GCM, e.g., it is included in TLS [29], ISO/IEC [11], NSA Suite B [22], and IEEE 802.1 [10]. A cryptographic competition on authenticated encryption schemes, called CAESAR, has been launched in 2013 [6], and it defines GCM as the benchmark algorithm of the competition. There are a large number of results studying the security of GCM. Ferguson showed a forgery attack against the use of short tags [8]. Joux showed a partial key recovery attack under the nonce-reuse setting [14]. Weak keys of GHASH, a polynomial hash function employed in GCM, was studied by Handschuh and Preneel [9], followed by Saarinen [28], Procter and Cid [24], and Bogdanov [5]. See also [1]. Other results related to GCM include [2,30,31], and Rogaway [26] presented a comprehensive survey on various aspects of GCM.

For the provable security aspect of GCM, the original proposal by McGrew and Viega [18,19] included proofs of the security. Later, Iwata, Ohashi, and Minematsu [12] pointed out a flaw in the proofs of [18,19] with counter examples that invalidate them. They also presented corrected proofs, but the security bounds are larger than the original ones, roughly by a factor of 2^{22}.

© International Association for Cryptologic Research 2015
G. Leander (Ed.): FSE 2015, LNCS 9054, pp. 385–407, 2015.
DOI: 10.1007/978-3-662-48116-5_19

The counter examples invalidate the proofs in [18,19], but they do not exclude the possibility that the original security bounds of [18,19] can still be proved, and in [12], an open question about the possibility of improving the security bounds of [12] was posed, which is the main question we consider in this paper. GCM relies its security on the use of a nonce, and the nonce determines the initial counter value. A collision on counter values, or a counter-collision, leads to an attack on GCM, and the counter-collision probability needs to be small. The crux of [12] is the development of a method to derive an upper bound on the counter-collision probability. [12] showed that the upper bound is obtained by solving a combinatorial problem involving arithmetic additions and xor's, and security bounds are derived by applying the sum bound to the counter-collision probability.

In this paper, we first develop an algorithm to generate nonces that have a high counter-collision probability. The problem is reduced to determining an equation that has as many solutions as possible, and the equation involves an arithmetic addition, finite field multiplications, and xor's. We show that it can be converted into a problem of solving a system of linear equations over GF(2), with a selection process of several constants in a greedy method. As a result, we obtain concrete examples of nonces that have a counter-collision probability of about $2^{20.75}/2^{128} = 2^{-107.25}$, and the results were verified by a program. With the same setting, the upper bound of [12] on the counter-collision probability is about $2^{22.75}/2^{128} = 2^{-105.25}$. This implies that, as long as we follow the proof strategy, in particular the use of the sum bound, the security bounds of [12] are tight within a factor of about 4.

A natural question is then whether it is possible to avoid using the sum bound in the proofs. We next answer this question positively, and we show that the avoidance indeed yields strong security bounds of GCM. We present two types of improvements. The first improvement reduces the constant, 2^{22}, appears in the security bounds in [12], to 32. The new security bounds improve the security bounds in [12] by a factor of 2^{17}, and they show that the security of GCM is actually close to what was originally claimed in [18,19]. Another improvement gives security bounds that are better than the first ones for long data. Specifically, if the average plaintext length to be authenticated and encrypted is longer than about 2 Gbytes, then the second improvement gives a stronger guarantee of security.

We note that the focus of this paper is the general case where a nonce of variable-length is used, while it is known that GCM has strong security bounds if the nonce length is fixed to 96 bits [12].

2 Preliminaries

We write $\{0,1\}^*$ for the set of all finite bit strings, and for an integer $\ell \geq 0$, we write $\{0,1\}^\ell$ for the set of all ℓ-bit strings. For $X \in \{0,1\}^*$, $|X|$ is its length in bits, and $|X|_\ell = \lceil |X|/\ell \rceil$ is its length in ℓ-bit blocks. We write ε for the empty string. For $X, Y \in \{0,1\}^*$, their concatenation is written as $X \,\|\, Y$,

(X, Y), or XY. The bit string of ℓ zeros is written as $0^\ell \in \{0,1\}^\ell$, and ℓ ones is written as $1^\ell \in \{0,1\}^\ell$. The prefix $\mathtt{0x}$ is used for the hexadecimal notation. For example, $\mathtt{0x28}$ is $00101000 \in \{0,1\}^8$. For $X \in \{0,1\}^*$ and an integer ℓ such that $|X| \geq \ell$, $\mathsf{msb}_\ell(X)$ denotes the most significant (the leftmost) ℓ bits of X, and $\mathsf{lsb}_\ell(X)$ denotes the least significant (the rightmost) ℓ bits of X. For $X \in \{0,1\}^*$ such that $|X| = j\ell$ for some integer $j \geq 1$, its partition into ℓ-bit blocks is written as $(X[1], \ldots, X[j]) \xleftarrow{\ell} X$, where $X[1], \ldots, X[j] \in \{0,1\}^\ell$ are unique bit strings that satisfy $X[1] \| \ldots \| X[x] = X$. For integers a and ℓ satisfying $0 \leq a \leq 2^\ell - 1$, we write $\mathsf{str}_\ell(a)$ for the ℓ-bit binary representation of a, i.e., if $a = \mathtt{a}_{\ell-1}2^{\ell-1} + \cdots + \mathtt{a}_1 2 + \mathtt{a}_0$ for $\mathtt{a}_{\ell-1}, \ldots, \mathtt{a}_1, \mathtt{a}_0 \in \{0,1\}$, then $\mathsf{str}_\ell(a) = \mathtt{a}_{\ell-1} \ldots \mathtt{a}_1 \mathtt{a}_0 \in \{0,1\}^\ell$. For $X = \mathtt{x}_{\ell-1} \ldots \mathtt{x}_1 \mathtt{x}_0 \in \{0,1\}^\ell$, let $\mathsf{int}(X)$ be the integer $\mathtt{x}_{\ell-1}2^{\ell-1} + \cdots + \mathtt{x}_1 2 + \mathtt{x}_0$. For a finite set $\mathcal{X}$, we write $\#\mathcal{X}$ for its cardinality, and $X \xleftarrow{\$} \mathcal{X}$ for a procedure of assigning X an element sampled uniformly at random from $\mathcal{X}$.

Throughout this paper, we fix a blockcipher $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$, where n is its block length in bits, which is fixed to $n = 128$, and $\mathcal{K}$ is a non-empty set of keys. The permutation specified by $K \in \mathcal{K}$ is written as E_K, and $C = E_K(M)$ denotes the ciphertext of a plaintext $M \in \{0,1\}^n$ under the key $K \in \mathcal{K}$. The set of n-bit strings, $\{0,1\}^n$, is also regarded as the finite field with 2^n elements which is written as $\mathrm{GF}(2^n)$. An n-bit string $\mathtt{a}_{n-1} \ldots \mathtt{a}_1 \mathtt{a}_0 \in \{0,1\}^n$ corresponds to a formal polynomial $a(x) = \mathtt{a}_{n-1} + \mathtt{a}_{n-2}x + \cdots + \mathtt{a}_1 x^{n-2} + \mathtt{a}_0 x^{n-1} \in \mathrm{GF}(2)[x]$. The irreducible polynomial used in GCM is $p(x) = 1 + x + x^2 + x^7 + x^{128}$, which is assumed to be the underlying polynomial throughout this paper.

3 Specification of GCM

We follow the description in [12], which follows the specification in [18,19] with minor notational changes. GCM takes two parameters: a blockcipher $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ and a tag length τ, where $64 \leq \tau \leq n$. If we use E and τ as parameters, then we write the corresponding GCM as $\mathrm{GCM}[E, \tau]$, and we write GCM-$\mathcal{E}$ for its encryption algorithm and GCM-$\mathcal{D}$ for its decryption algorithm. These algorithms are defined in Fig. 1. In GCM-$\mathcal{E}$ and GCM-$\mathcal{D}$, we use two subroutines defined in Fig. 2. The first one is the counter mode encryption, denoted by CTR, and the other one is the polynomial hash function over $\mathrm{GF}(2^n)$, denoted by GHASH. See Fig. 3 for the overall structure of GCM-$\mathcal{E}$, and Fig. 4 for the subroutines used therein.

The encryption algorithm, GCM-$\mathcal{E}$, takes a key $K \in \mathcal{K}$, a nonce $N \in \{0,1\}^*$, associated data $A \in \{0,1\}^*$, and a plaintext $M \in \{0,1\}^*$ as input, and returns a pair of a ciphertext $C \in \{0,1\}^*$ and a tag $T \in \{0,1\}^\tau$. We require $1 \leq |N| \leq 2^{n/2} - 1$, $0 \leq |A| \leq 2^{n/2} - 1$, and $0 \leq |M| \leq n(2^{32} - 2)$, and it holds that $|C| = |M|$. We write $(C, T) \leftarrow \text{GCM-}\mathcal{E}_K^{N,A}(M)$. The decryption algorithm, GCM-$\mathcal{D}$, takes a key $K \in \mathcal{K}$, a nonce $N \in \{0,1\}^*$, associated data $A \in \{0,1\}^*$, a ciphertext $C \in \{0,1\}^*$, and a tag $T \in \{0,1\}^\tau$ as input, and returns either a plaintext $M \in \{0,1\}^*$ or the distinguished invalid symbol denoted by $\perp$. We write $M \leftarrow \text{GCM-}\mathcal{D}_K^{N,A}(C, T)$ or $\perp \leftarrow \text{GCM-}\mathcal{D}_K^{N,A}(C, T)$.

Algorithm GCM-$\mathcal{E}_K^{N,A}(M)$

1. $L \leftarrow E_K(0^n)$
2. **if** $|N| = 96$ **then** $I[0] \leftarrow N \,\|\, 0^{31}1$
3. **else** $I[0] \leftarrow \mathsf{GHASH}_L(\varepsilon, N)$
4. $m \leftarrow |M|_n$
5. $S \leftarrow \mathsf{CTR}_K(I[0], m)$
6. $C \leftarrow M \oplus \mathsf{msb}_{|M|}(S)$
7. $\widetilde{T} \leftarrow E_K(I[0]) \oplus \mathsf{GHASH}_L(A, C)$
8. $T \leftarrow \mathsf{msb}_\tau(\widetilde{T})$
9. **return** (C, T)

Algorithm GCM-$\mathcal{D}_K^{N,A}(C, T)$

1. $L \leftarrow E_K(0^n)$
2. **if** $|N| = 96$ **then** $I[0] \leftarrow N \,\|\, 0^{31}1$
3. **else** $I[0] \leftarrow \mathsf{GHASH}_L(\varepsilon, N)$
4. $\widetilde{T}^* \leftarrow E_K(I[0]) \oplus \mathsf{GHASH}_L(A, C)$
5. $T^* \leftarrow \mathsf{msb}_\tau(\widetilde{T}^*)$
6. **if** $T \neq T^*$ **then return** $\perp$
7. $m \leftarrow |C|_n$
8. $S \leftarrow \mathsf{CTR}_K(I[0], m)$
9. $M \leftarrow C \oplus \mathsf{msb}_{|C|}(S)$
10. **return** M

Fig. 1. Definitions of GCM-$\mathcal{E}_K^{N,A}(M)$ and GCM-$\mathcal{D}_K^{N,A}(C, T)$

Algorithm $\mathsf{CTR}_K(I[0], m)$

1. **for** $j \leftarrow 1$ **to** m **do**
2. $\quad I[j] \leftarrow \mathsf{inc}(I[j-1])$
3. $\quad S[j] \leftarrow E_K(I[j])$
4. $S \leftarrow (S[1], S[2], \ldots, S[m])$
5. **return** S

Algorithm $\mathsf{GHASH}_L(A, C)$

1. $a \leftarrow n\lceil |A|_n - |A|$
2. $c \leftarrow n\lceil |C|_n - |C|$
3. $X \leftarrow A \,\|\, 0^a \,\|\, C \,\|\, 0^c \,\|\, \mathsf{str}_{n/2}(|A|) \,\|\, \mathsf{str}_{n/2}(|C|)$
4. $(X[1], \ldots, X[x]) \xleftarrow{n} X$
5. $Y \leftarrow 0^n$
6. **for** $j \leftarrow 1$ **to** x **do**
7. $\quad Y \leftarrow L \cdot (Y \oplus X[j])$
8. **return** Y

Fig. 2. Definitions of $\mathsf{CTR}_K(I[0], m)$ and $\mathsf{GHASH}_L(A, C)$

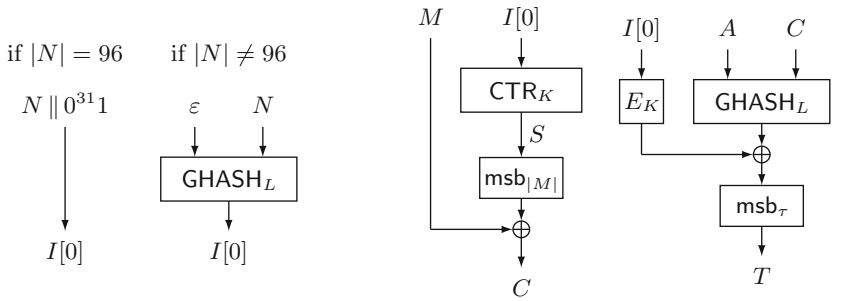

Fig. 3. Overall structure of $(C, T) \leftarrow$ GCM-$\mathcal{E}_K^{N,A}(M)$

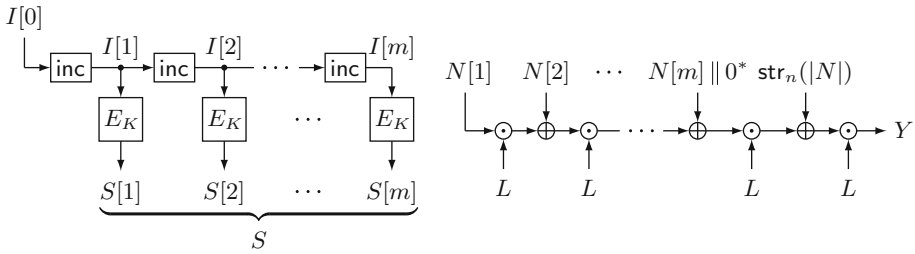

Fig. 4. Subroutines $S \leftarrow \mathsf{CTR}_K(I[0], m)$ and $Y \leftarrow \mathsf{GHASH}_L(A, C)$, where $(A, C) = (\varepsilon, N)$, $N = (N[1], \ldots, N[m])$, $|N[1]| = \cdots = |N[m-1]| = n$, and $1 \le |N[m]| \le n$

We use the increment function, denoted by inc, in the definition of CTR. It takes a bit string $X \in \{0,1\}^n$ as input, and we regard the least significant (the rightmost) 32 bits of X as a non-negative integer, and then increment the value by one modulo 2^{32}. That is, we have

$$\mathsf{inc}(X) = \mathsf{msb}_{n-32}(X) \,\|\, \mathsf{str}_{32}(\mathsf{int}(\mathsf{lsb}_{32}(X)) + 1 \bmod 2^{32}).$$

For $r \ge 0$, $\mathsf{inc}^r(X)$ means that we apply inc on X for r times, and $\mathsf{inc}^{-r}(X)$ means that we apply the inverse function of inc on X for r times. By convention, we let $\mathsf{inc}^0(X) = X$, and we thus have $I[j] = \mathsf{inc}^j(I[0])$ for $0 \le j \le m$ in the 2nd line in the definition of CTR. In the definition of GHASH, the multiplication in the 7th line is over $\mathrm{GF}(2^n)$. We note that when $|N| \ne 96$, we have $\mathsf{GHASH}_L(\varepsilon, N) = X[1] \cdot L^x \oplus \cdots \oplus X[x] \cdot L$, where $X = (X[1], \ldots, X[x]) = N \,\|\, 0^{n|N|_n - |N|} \,\|\, \mathsf{str}_n(|N|)$.

Let $\mathrm{Perm}(n)$ be the set of all permutations on $\{0,1\}^n$, and we call $P \xleftarrow{\$} \mathrm{Perm}(n)$ a random permutation. Let $\mathrm{GCM}[\mathrm{Perm}(n), \tau]$ be GCM where we use a random permutation P as the blockcipher E_K. We write GCM-$\mathcal{E}_P$ for its encryption algorithm and GCM-$\mathcal{D}_P$ for its decryption algorithm. Similarly, let $\mathrm{Rand}(n)$ be the set of all functions from $\{0,1\}^n$ to $\{0,1\}^n$, and we call $F \xleftarrow{\$} \mathrm{Rand}(n)$ a random function. Let $\mathrm{GCM}[\mathrm{Rand}(n), \tau]$ be GCM where we use F as E_K. We write GCM-$\mathcal{E}_F$ for its encryption algorithm and GCM-$\mathcal{D}_F$ for its decryption algorithm.

4 Security Definitions

An adversary is a probabilistic algorithm that has access to one or two oracles. We write $\mathcal{A}^{\mathcal{O}}$ for an adversary $\mathcal{A}$ that has access to an oracle $\mathcal{O}$, and $\mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2}$ for $\mathcal{A}$ that has access to two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$. Following [3,25], we consider privacy and authenticity of GCM.

A privacy adversary $\mathcal{A}$ has access to a GCM encryption oracle or a random-bits oracle. The GCM encryption oracle, which we write Enc_K, takes (N, A, M) as input and returns $(C, T) \leftarrow \text{GCM-}\mathcal{E}_K^{N,A}(M)$. The random-bits oracle, \$, takes (N, A, M) as input and returns $(C, T) \xleftarrow{\$} \{0,1\}^{|M|+\tau}$. The privacy advantage of

$\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{GCM}[E,\tau]}(\mathcal{A}) \stackrel{\mathrm{def}}{=} \Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathsf{Enc}_K(\cdot,\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\$(\cdot,\cdot,\cdot)} \Rightarrow 1\right],$$

where the first probability is defined over the randomness of $K \xleftarrow{\$} \mathcal{K}$ and $\mathcal{A}$, and the last one is over the randomness of \$ and $\mathcal{A}$. We assume that privacy adversaries are nonce-respecting: if $\mathcal{A}$ makes q queries and $N_1 \ldots, N_q$ are nonces used in the queries, then it holds that $N_i \neq N_j$ for $1 \leq i < j \leq q$.

An authenticity adversary $\mathcal{A}$ has access to two oracles, GCM encryption and decryption oracles. The GCM encryption oracle, Enc_K, is described as above. The GCM decryption oracle, Dec_K, takes (N, A, C, T) as input and returns $M \leftarrow \text{GCM-}\mathcal{D}_K^{N,A}(C,T)$ or $\perp \leftarrow \text{GCM-}\mathcal{D}_K^{N,A}(C,T)$. The authenticity advantage of $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{GCM}[E,\tau]}(\mathcal{A}) \stackrel{\mathrm{def}}{=} \Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathsf{Enc}_K(\cdot,\cdot,\cdot),\mathsf{Dec}_K(\cdot,\cdot,\cdot,\cdot)} \text{ forges}\right],$$

where the probability is defined over the randomness of $K \xleftarrow{\$} \mathcal{K}$ and $\mathcal{A}$. If $\mathcal{A}$ makes a query (N, A, M) to Enc_K and receives (C, T), then we assume that $\mathcal{A}$ does not subsequently make a query (N, A, C, T) to Dec_K. We also assume that $\mathcal{A}$ does not repeat a query to Dec_K. We define that $\mathcal{A}$ forges if at least one of the responses from Dec_K is not $\perp$. We assume that authenticity adversaries are nonce-respecting with respect to encryption queries. That is, assume that $\mathcal{A}$ makes q queries to Enc_K and q' queries to Dec_K, where $N_1, \ldots, N_q$ are the nonces used for Enc_K, and $N'_1, \ldots, N'_{q'}$ are the nonces for Dec_K. We assume that $N_i \neq N_j$ holds for $1 \leq i < j \leq q$, but $N_i = N'_j$ may hold for some $1 \leq i \leq q$ and $1 \leq j \leq q'$, and $N'_i = N'_j$ may also hold for some $1 \leq i < j \leq q'$.

5 GCM Security Bounds in [12] Need 881145

5.1 Review of Results in [12]

We first review results from [12]. Consider a privacy adversary $\mathcal{A}$, and suppose that $\mathcal{A}$ makes q queries $(N_1, A_1, M_1), \ldots, (N_q, A_q, M_q)$, where $|N_i|_n = n_i$ and $|M_i|_n = m_i$. Then the total plaintext length is $m_1 + \cdots + m_q$, and the maximum nonce length is $\max\{n_1, \ldots, n_q\}$. The following privacy result was proved.

Proposition 1 [12]. *Let* $\mathrm{Perm}(n)$ *and* τ *be the parameters of* GCM. *Then for any* $\mathcal{A}$ *that makes at most* q *queries, where the total plaintext length is at most* σ *blocks and the maximum nonce length is at most* ℓ_N *blocks,*

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{GCM}[\mathrm{Perm}(n),\tau]}(\mathcal{A}) \leq \frac{0.5(\sigma + q + 1)^2}{2^n} + \frac{2^{22} q(\sigma + q)(\ell_N + 1)}{2^n}. \tag{1}$$

Suppose that an authenticity adversary $\mathcal{A}$ makes q queries $(N_1, A_1, M_1), \ldots, (N_q, A_q, M_q)$ to Enc_K and q' queries $(N'_1, A'_1, C'_1, T'_1), \ldots, (N'_{q'}, A'_{q'}, C'_{q'}, T'_{q'})$ to Dec_K, where $|N_i|_n = n_i$, $|A_i|_n = a_i$, $|M_i|_n = m_i$, $|N'_i|_n = n'_i$, $|A'_i|_n = a'_i$, and

$|C'_i|_n = m'_i$. Then the total plaintext length is $m_1 + \cdots + m_q$, the maximum nonce length is $\max\{n_1, \ldots, n_q, n'_1, \ldots, n'_{q'}\}$, and the maximum input length is $\max\{a_1 + m_1, \ldots, a_q + m_q, a'_1 + m'_1, \ldots, a'_{q'} + m'_{q'}\}$. The following authenticity result was proved.

Proposition 2 [12]. *Let* $\mathrm{Perm}(n)$ *and* τ *be the parameters of* GCM. *Then for any* $\mathcal{A}$ *that makes at most* q *encryption queries and* q' *decryption queries, where the total plaintext length is at most* σ *blocks, the maximum nonce length is at most* ℓ_N *blocks, and the maximum input length is at most* ℓ_A *blocks,*

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{GCM}[\mathrm{Perm}(n),\tau]}(\mathcal{A}) \le \frac{0.5(\sigma + q + q' + 1)^2}{2^n} + \frac{2^{22}(q + q')(\sigma + q + 1)(\ell_N + 1)}{2^n} + \frac{q'(\ell_A + 1)}{2^\tau}. \quad (2)$$

We see that a non-small constant, 2^{22}, appears in (1) and (2). In what follows, we recall how the constant was introduced by reviewing the proof of Proposition 1. We first replace a random permutation P with a random function F. We have

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{GCM}[\mathrm{Perm}(n),\tau]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{priv}}_{\mathrm{GCM}[\mathrm{Rand}(n),\tau]}(\mathcal{A}) + \frac{0.5(\sigma + q + 1)^2}{2^n}$$

from the PRP/PRF switching lemma [4].

Now assume that $\mathcal{A}$ makes q queries, and for $1 \le i \le q$, let (N_i, A_i, M_i) be the i-th query, where $|M_i|_n = m_i$. Let the initial counter value, $I_i[0]$, be $I_i[0] \leftarrow \mathsf{GHASH}_L(\varepsilon, N_i)$ if $|N_i| \neq 96$, and $I_i[0] \leftarrow N_i \,\|\, 0^{31}1$ otherwise. We also let the counter value, $I_i[j]$, be $I_i[j] \leftarrow \mathsf{inc}^j(I_i[0])$ for $1 \le j \le m_i$. With this notation, we have the following list of counter values.

$$\begin{array}{c} I_1[0], I_1[1], \ldots, I_1[m_1] \\ I_2[0], I_2[1], \ldots, I_2[m_2] \\ \vdots \\ I_q[0], I_q[1], \ldots, I_q[m_q] \end{array} \quad (3)$$

At this point, we are ready to define a bad event. We say that the bad event occurs if we have at least one of the following events:

Case (A). $I_i[j] = 0^n$ holds for some (i, j) such that $1 \le i \le q$ and $0 \le j \le m_i$.
Case (B). $I_i[j] = I_{i'}[j']$ holds for some (i, j, i', j') such that $1 \le i' < i \le q$, $0 \le j' \le m_{i'}$, and $0 \le j \le m_i$.

As analyzed in detail in [13, Appendix D], the absence of the bad event implies that, each time $\mathcal{A}$ makes a query (N_i, A_i, M_i), $\mathcal{A}$ obtains a uniform random string of $|M_i| + \tau$ bits, which in turn implies that the adaptivity of $\mathcal{A}$ does not help and we may fix the q queries $(N_1, A_1, M_1), \ldots, (N_q, A_q, M_q)$ of $\mathcal{A}$. We evaluate the probability of the bad event based on the randomness of L. For simplicity, we write $\Pr_L[\mathsf{E}]$ for $\Pr[L \xleftarrow{\$} \{0,1\}^n : \mathsf{E}]$ for an event E. We have

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{GCM}[\mathrm{Rand}(n),\tau]}(\mathcal{A}) \le \Pr_L[\text{Case (A) holds}] + \Pr_L[\text{Case (B) holds}]. \quad (4)$$

The first probability is easy to evaluate and we have

$$\Pr_L[\text{Case (A) holds}] \le \sum_{1\le i\le q, 0\le j\le m_i} \Pr_L[I_i[j] = 0^n] \le \frac{(\sigma+q)(\ell_N+1)}{2^n}, \tag{5}$$

since $\mathsf{inc}^j(I_i[0]) = 0^n$ is a non-trivial equation in L of degree at most ℓ_N+1 over $\mathrm{GF}(2^n)$ if $|N_i| \neq 96$, and hence the probability is at most $(\ell_N+1)/2^n$, or we never have the event if $|N_i| = 96$.

The second probability can also be evaluated as the first one by using "the sum bound," and we obtain

$$\Pr_L[\text{Case (B) holds}] \le \sum_{1\le i'<i\le q, 0\le j'\le m_{i'}, 0\le j\le m_i} \Pr_L\left[I_i[j] = I_{i'}[j']\right]. \tag{6}$$

It remains to evaluate $\Pr_L\left[I_i[j] = I_{i'}[j']\right]$ for each (i, j, i', j'), and we have the following four cases to consider: $|N_i| = |N_{i'}| = 96$, $|N_i| \neq 96$ and $|N_{i'}| = 96$, $|N_i| = 96$ and $|N_{i'}| \neq 96$, and $|N_i|, |N_{i'}| \neq 96$.

The case $|N_i| = |N_{i'}| = 96$ is easy to analyze and we have $\Pr_L\left[I_i[j] = I_{i'}[j']\right] = 0$. If $|N_i| \neq 96$ and $|N_{i'}| = 96$, then we have $\Pr_L\left[I_i[j] = I_{i'}[j']\right] \le (\ell_N+1)/2^n$ since $\mathsf{inc}^j(I_i[0]) = \mathsf{inc}^{j'}(I_{i'}[0])$ is a non-trivial equation in L of degree at most ℓ_N+1 over $\mathrm{GF}(2^n)$. The analysis for the case $|N_i| = 96$ and $|N_{i'}| \neq 96$ is the same as the previous case. The analysis of the last case, $|N_i|, |N_{i'}| \neq 96$, is not simple, and we review the notation used in [12].

For $0 \le r \le 2^{32}-1$ and two distinct nonces N and N' which are not 96 bits, let the counter-collision, denoted by $\mathsf{Coll}_L(r, N, N')$, be the event

$$\mathsf{inc}^r(\mathsf{GHASH}_L(\varepsilon, N)) = \mathsf{GHASH}_L(\varepsilon, N'). \tag{7}$$

We say $\Pr_L[\mathsf{Coll}_L(r, N, N')]$ a counter-collision probability. Recall that $I_i[j] = I_{i'}[j']$ is equivalent to $\mathsf{inc}^j(I_i[0]) = \mathsf{inc}^{j'}(I_{i'}[0])$, where $I_i[0] \leftarrow \mathsf{GHASH}_L(\varepsilon, N_i)$ and $I_{i'}[0] \leftarrow \mathsf{GHASH}_L(\varepsilon, N_{i'})$, and this can be written as $\mathsf{Coll}_L(r, N, N')$ with $(r, N, N') = (j-j', N_i, N_{i'})$ if $j - j' \ge 0$, and $(r, N, N') = (j'-j, N_{i'}, N_i)$ otherwise.

Now define $\mathbb{Y}_r \subseteq \{0,1\}^{32}$, for $0 \le r \le 2^{32}-1$, as

$$\mathbb{Y}_r \stackrel{\text{def}}{=} \left\{\mathsf{str}_{32}(\mathsf{int}(Y) + r \bmod 2^{32}) \oplus Y \mid Y \in \{0,1\}^{32}\right\}, \tag{8}$$

and write its cardinality as $\alpha_r \stackrel{\text{def}}{=} \#\mathbb{Y}_r$. We let $\alpha_{\max} \stackrel{\text{def}}{=} \max\{\alpha_r \mid 0 \le r \le 2^{32}-1\}$. The following result was proved.

Proposition 3 [12]. *For any $0 \le r \le 2^{32}-1$ and two distinct nonces N and N' which are not 96 bits, it holds that $\Pr_L[\mathsf{Coll}_L(r, N, N')] \le \alpha_r(\ell_N+1)/2^n$, where $|N|_n, |N'|_n \le \ell_N$.*

$\mathbb{Y}_r$ can be used to replace the arithmetic addition by r in $\mathsf{inc}^r(X)$ with the xor of some constant. That is, we convert $\mathsf{inc}^r(X)$ into $X \oplus (0^{96} \,\|\, Y)$ for some

$Y \in \{0,1\}^{32}$, and as argued in [12], $\mathbb{Y}_r$ exhaustively covers all the possible constants, and it must be the case that $Y \in \mathbb{Y}_r$. Note that the constant is of the form $(0^{96} \,\|\, Y)$ and the most significant 96 bits can be fixed to 0^{96}, as inc has no effect on these bits. For simplicity, for any $Y \in \{0,1\}^{32}$, let $[\![Y]\!] = (0^{96} \,\|\, Y)$.

In [12], a recursive formula to compute the value of α_r was presented, and the value of $\alpha_{\max}$ was shown to be $\alpha_{\max} = 3524578$, where the equality holds when $r =$ `0x2aaaaaab`, `0xaaaaaaab`, `0x55555555`, and `0xd5555555`. We have $3524578 \le 2^{22}$, and this yields $\Pr_L\left[I_i[j] = I_{i'}[j']\right] \le 2^{22}(\ell_N + 1)/2^n$ for the last case, which is the source reason why we have this constant in (1) and (2).

A question is if we really need the constant, or if we can make it smaller.

5.2 Case $r =$ `0x55555555`

Our approach to the question is to derive the values of r, N, and N' where $\Pr_L[\mathsf{Coll}_L(r, N, N')]$ is large, or equivalently, the equation $\mathsf{Coll}_L(r, N, N')$ has as many solutions (in L) as possible. We now present our main result of this section.

Theorem 1. *There exist* $0 \le r \le 2^{32} - 1$ *and two distinct nonces* N *and* N' *such that* $|N| = |N'| = 128$ *and* $\Pr_L[\mathsf{Coll}_L(r, N, N')] \ge 1762290/2^n$.

Proof. Let $r =$ `0x55555555`, and let N and N' be the following values.

$$\begin{cases} N = \texttt{0x8d44009c dc550100 00000000 00000000} \\ N' = \texttt{0x5b6dbdd9 f3b151d9 d1bc4145 ecb396ef} \end{cases} \tag{9}$$

Then $\mathsf{Coll}_L(r, N, N')$ is equivalent to

$$\mathsf{inc}^r(U \cdot L^2 \oplus V \cdot L) = U' \cdot L^2 \oplus V \cdot L, \tag{10}$$

where $U = N$, $U' = N'$, and $V =$ `0x00000000 00000000 00000000 00000080`. Note that V is the hexadecimal form of $|N| = |N'| = 128$. Now $\mathbb{Y}_r$ consists of $\alpha_{\max}$ constants, and we can list all these constants by listing $\mathsf{str}_{32}(\mathsf{int}(Y) + r \bmod 2^{32}) \oplus Y$ for all $Y \in \{0,1\}^{32}$. Let $\mathbb{Y}_r = \{Y_1, \ldots, Y_{\alpha_{\max}}\}$ be the concrete representation of $\mathbb{Y}_r$. We can solve (in L) the equation $U \cdot L^2 \oplus V \cdot L \oplus [\![Y_\ell]\!] = U' \cdot L^2 \oplus V \cdot L$ for all $Y_\ell \in \mathbb{Y}_r$, which gives us $L = \left[(U \oplus U')^{-1} \cdot [\![Y_\ell]\!]\right]^{1/2}$, and see if this L satisfies (10). We find that 1762290 values of L satisfy (10), which was verified by using a program, and hence we have $\Pr_L[\mathsf{Coll}_L(r, N, N')] \ge 1762290/2^n$. □

With the same value of $r =$ `0x55555555`, the values of N and N' in the following list give the same probability.

$$\begin{cases} N = \texttt{0x215c004e 6e2a8080 00000000 00000000} \\ N' = \texttt{0xab48deec f9d8a8ec e8de20a2 f659cb77} \end{cases} \tag{11}$$

$$\begin{cases} N = \texttt{0x1bb000e9 9f71db00 00000000 00000000} \\ N' = \texttt{0xb0085245 fd3dc69e 9de41b1a 943d314f} \end{cases} \tag{12}$$

$$\begin{cases} N = \texttt{0x77500027 37154040 00000000 00000000} \\ N' = \texttt{0xd35a6f76 7cec5476 746f1051 7b2ce5bb} \end{cases} \tag{13}$$

Theorem 1 suggests that, for the particular value of $r = \texttt{0x55555555}$, there exist N and N' with $\Pr_L[\mathsf{Coll}_L(r, N, N')] \geq 1762290/2^n = 881145(\ell_N + 1)/2^n$, where $|N|_n = |N'|_n = \ell_N = 1$. Specifically, the result shows that the constant, $\alpha_{\max}$, in Proposition 3 for the case $r = \texttt{0x55555555}$ cannot be made smaller than 881145. Therefore, as long as we make use of the sum bound in (6) to derive the upper bound on $\Pr_L$[Case (B) holds], the constants in (1) and (2) cannot be made smaller than 881145. Since $3524578 \leq 2^{21.75}$ and $881145 \geq 2^{19.74}$, we may conclude that (1) and (2) are tight up to a constant factor of about 4 if we use the sum bound. We next present how we have derived the values of N and N' in (9).

5.3 Deriving N and N'

Recall that our goal is to derive r, N, and N' where $\mathsf{Coll}_L(r, N, N')$ defined in (7) has as many solutions in L as possible. We decided to focus on $r = \texttt{0x55555555}$ since this is one of the four values of r that is potential to have the maximum number of solutions. We also decided to focus on the case $|N| = |N'| = 128$, since even with this restricted length of nonces, we still have about 2^{256} possible search space of N and N'. With the setting, (7) is equivalent to

$$\mathsf{inc}^r(U \cdot L^2 \oplus V \cdot L) = U' \cdot L^2 \oplus V \cdot L, \tag{14}$$

where $r = \texttt{0x55555555}$ and $V = \texttt{0x00000000 00000000 00000000 00000080}$ are now fixed, and $U = N$ and $U' = N'$ are the variables we are searching for.

Converting $\mathsf{inc}^r(X)$ *into* $X \oplus [\![Y_\ell]\!]$. As mentioned in the proof of Theorem 1, $\mathbb{Y}_r$ consists of $\alpha_{\max}$ constants, and let $\mathbb{Y}_r = \{Y_1, \ldots, Y_{\alpha_{\max}}\}$ be the concrete representation of $\mathbb{Y}_r$. Now instead of directly considering (14), we consider the following simultaneous equation.

$$\begin{cases} \mathsf{inc}^r(U \cdot L^2 \oplus V \cdot L) = U \cdot L^2 \oplus V \cdot L \oplus [\![Y_\ell]\!] & (15) \\ (U \oplus U') \cdot L^2 = [\![Y_\ell]\!] & (16) \end{cases}$$

Equation (15) is the conversion of the arithmetic addition by r in the left hand side of (14) using some constant $Y_\ell \in \mathbb{Y}_r$, and then we obtain (16) by simplifying (14) after the conversion with $Y_\ell \in \mathbb{Y}_r$ used in (15) , where the term $V \cdot L$ cancels out. Note that the conversion of (15) is always possible, and (14) holds if and only if (16) holds, and hence (14) is equivalent to (15) and (16) holding for some $Y_\ell \in \mathbb{Y}_r$.

Deriving Conditions on X for $\mathsf{inc}^r(X) = X \oplus [\![Y_\ell]\!]$. Suppose that we fix some Y_ℓ from $\mathbb{Y}_r$, and convert $\mathsf{inc}^r(X)$ into $X \oplus [\![Y_\ell]\!]$. Now we observe that the equality of $\mathsf{inc}^r(X) = X \oplus [\![Y_\ell]\!]$ imposes restrictions on some bits of X. For instance, when $Y_\ell = \mathtt{0x55555555}$, then X must be of the form

$$X = \underbrace{* \cdots *}_{96 \text{ bits}} \underbrace{*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0}_{32 \text{ bits}}$$

in binary, where $*$ can be 0 or 1, i.e., if $X = \mathtt{x}_{127} \ldots \mathtt{x}_0$ is the binary representation of X, it must be the case that $\mathtt{x}_{30} = 0 \wedge \mathtt{x}_{28} = 0 \wedge \cdots \wedge \mathtt{x}_0 = 0$. When $Y_\ell = \mathtt{0xffffffff}$, then X must be of the form

$$X = \underbrace{* \cdots *}_{96 \text{ bits}} \underbrace{*00101010101010101010101010101}_{32 \text{ bits}}$$

in binary. Using $Y_\ell = \mathtt{0x55555555}$ fixes 16 bits of X, and $Y_\ell = \mathtt{0xffffffff}$ fixes 31 bits of X. The condition and the number of bits we have to fix depend on the value of Y_ℓ. We have to fix from 16 to 31 bits of X, and these are two extreme cases that have the minimum number and the maximum number of conditions. On average, around 20 bits are fixed. Let $\mathbb{C}(Y_\ell)$ be the set of conditions to replace $\mathsf{inc}^r(X)$ to $X \oplus [\![Y_\ell]\!]$. We represent $\mathbb{C}(Y_\ell)$ as a column vector

$$\mathbb{C}(Y_\ell) = \begin{bmatrix} \mathtt{x}_{127} \\ \vdots \\ \mathtt{x}_0 \end{bmatrix},$$

where $\mathtt{x}_i \in \{*, 0, 1\}$. Let $\mathbb{I}(Y_\ell)$ be the set of indices with $\mathtt{x}_i \neq *$, i.e., $\mathbb{I}(Y_\ell) = \{i \mid \mathtt{x}_i \neq *\}$. We note that $127, \ldots, 32$ are not in $\mathbb{I}(Y_\ell)$ as $\mathtt{x}_{127}, \ldots, \mathtt{x}_{32}$ are all $*$.

Given Y_ℓ, there are several approaches to write down $\mathbb{C}(Y_\ell)$. For instance, a possible approach is to follow the framework in [21], or to use the tool [15] developed in [16,17]. We present in [23] an algorithm that directly gives us the conditions.

Decomposition into Bits. Let us continue focusing on Y_ℓ from $\mathbb{Y}_r$ that we have fixed. We can solve (16) with respect to L, and we obtain $L = \left[(U \oplus U')^{-1} \cdot [\![Y_\ell]\!]\right]^{1/2} = \left[(U \oplus U')^{-1} \cdot [\![Y_\ell]\!]\right]^{2^{127}}$. Now we consider the argument, $U \cdot L^2 \oplus V \cdot L$, of inc^r of (15) . With this L, the argument becomes $U \cdot (U \oplus U')^{-1} \cdot [\![Y_\ell]\!] \oplus V \cdot \left[(U \oplus U')^{-1} \cdot [\![Y_\ell]\!]\right]^{2^{127}}$. At this point, instead of treating U and U' as variables, we let $W = (U \oplus U')^{-1}$ and regard U and W as variables. With this replacement, we have $L = \left[W \cdot [\![Y_\ell]\!]\right]^{2^{127}}$, and the argument becomes

$$U \cdot W \cdot [\![Y_\ell]\!] \oplus V \cdot W^{2^{127}} \cdot [\![Y_\ell]\!]^{2^{127}}. \tag{17}$$

It is well known that a multiplication by a constant and a squaring operation over $\mathrm{GF}(2^n)$ are linear operations in $\mathrm{GF}(2)$, e.g., see [8]. We make an observation that, if we decompose (17) into bits using $U = \mathtt{u}_{127} \ldots \mathtt{u}_0$ and $W = \mathtt{w}_{127} \ldots \mathtt{w}_0$

as variables, then each bit of the first term, $U \cdot W \cdot [\![Y_\ell]\!]$, can be represented by using $\mathtt{u}_{127}\mathtt{w}_{127}, \ldots, \mathtt{u}_{127}\mathtt{w}_0, \ldots, \mathtt{u}_0\mathtt{w}_{127}, \ldots, \mathtt{u}_0\mathtt{w}_0$, and the second term, $V \cdot W^{2^{127}} \cdot [\![Y_\ell]\!]^{2^{127}}$, can be represented by using $\mathtt{w}_{127}, \ldots, \mathtt{w}_0$. The first term consists of terms of the form $\mathtt{u}_i\mathtt{w}_j$, a total of $128 \times 128 = 16384$ variations, and we replace the term $\mathtt{u}_i\mathtt{w}_j$ with a monomial $\mathtt{s}_{128i+j}$. Let $\mathtt{z}_{127} \ldots \mathtt{z}_0$ be the decomposition of (17) into bits. Then we can represent $\mathtt{z}_i$ as a linear function of $\mathtt{s}_{16383}, \ldots, \mathtt{s}_0$ and $\mathtt{w}_{127}, \ldots, \mathtt{w}_0$. In other words, there is a linear function f_i that describes $\mathtt{z}_i$ as

$$\mathtt{z}_i = f_i(\mathtt{s}_{16383}, \ldots, \mathtt{s}_0, \mathtt{w}_{127}, \ldots, \mathtt{w}_0).$$

Let us define a binary row vector $\mathtt{row}_i$, which is associated to f_i, of length $16384+128$ that lists the coefficients of $\mathtt{s}_{16383}, \ldots, \mathtt{s}_0, \mathtt{w}_{127}, \ldots, \mathtt{w}_0$. We can collect them into a $128 \times (16384 + 128)$ binary matrix $\mathbb{M}$ to write

$$\begin{bmatrix} \mathtt{z}_{127} \\ \vdots \\ \mathtt{z}_0 \end{bmatrix} = \mathbb{M} \cdot \mathbb{S}, \text{where } \mathbb{M} = \begin{bmatrix} \mathtt{row}_{127} \\ \vdots \\ \mathtt{row}_0 \end{bmatrix} \text{ and } \mathbb{S} \stackrel{\text{def}}{=} \begin{bmatrix} \mathtt{s}_{16383} \\ \vdots \\ \mathtt{s}_0 \\ \mathtt{w}_{127} \\ \vdots \\ \mathtt{w}_0 \end{bmatrix}.$$

$\mathbb{S}$ is the column vector that consists of the variables we are searching for. We note that $\mathbb{M}$ depends on Y_ℓ, and we thus write $\mathbb{M}(Y_\ell)$ to describe the dependency.

Recall that $\mathtt{z}_{127} \ldots \mathtt{z}_0$ is the decomposition of (17) into bits. The equality of (15) holds if $\mathbb{C}(Y_\ell)$ is satisfied. In other words, we require

$$\mathtt{x}_i = f_i(\mathtt{s}_{16383}, \ldots, \mathtt{s}_0, \mathtt{w}_{127}, \ldots, \mathtt{w}_0)$$

holds for all $i \in \mathbb{I}(Y_\ell)$.

Deriving U and W. Let us still focus on Y_ℓ from $\mathbb{Y}_r$. For $\mathbb{C}(Y_\ell) = \left[\mathtt{x}_{127} \cdots \mathtt{x}_0\right]^{\mathsf{tr}}$, where $\mathtt{x}_i \in \{*, 0, 1\}$ and X^{tr} is the transposition of a row vector X, let $\widetilde{\mathbb{C}}(Y_\ell)$ be a column vector that is obtained from $\mathbb{C}(Y_\ell)$ by removing $*$. Suppose that $\widetilde{\mathbb{C}}(Y_\ell)$ consists of s elements, and let us represent it as $\widetilde{\mathbb{C}}(Y_\ell) = \left[\mathtt{x}_{i_1} \cdots \mathtt{x}_{i_s}\right]^{\mathsf{tr}}$. Note that we have $\mathbb{I}(Y_\ell) = \{i_1, \ldots, i_s\}$. Let $\widetilde{\mathbb{M}}(Y_\ell) = \left[\mathtt{row}_{i_1} \cdots \mathtt{row}_{i_s}\right]^{\mathsf{tr}}$ be a matrix that consists of the relevant s row vectors $\mathtt{row}_{i_1}, \ldots, \mathtt{row}_{i_s}$ of $\mathbb{M}(Y_\ell) = \left[\mathtt{row}_{127} \cdots \mathtt{row}_0\right]^{\mathsf{tr}}$. Now we can apply the Gaussian elimination to solve a system of linear equations

$$\widetilde{\mathbb{C}}(Y_\ell) = \widetilde{\mathbb{M}}(Y_\ell) \cdot \mathbb{S} \tag{18}$$

to derive $\mathtt{s}_{16383}, \ldots, \mathtt{s}_0, \mathtt{w}_{127}, \ldots, \mathtt{w}_0$, and if we can further derive $\mathtt{u}_{127}, \ldots, \mathtt{u}_0$ that are consistent with them, then this gives us U and W that have $L = \left[W \cdot [\![Y_\ell]\!]\right]^{2^{127}}$ as a solution to (15) and (16).

We next extend this to deal with multiple constants from $\mathbb{Y}_r$. Suppose that we choose j constants $Y_{\ell_1}, \ldots, Y_{\ell_j}$ from $\mathbb{Y}_r$. We combine the conditions of (18) into a single system of linear equations

$$\begin{bmatrix} \widetilde{\mathbb{C}}(Y_{\ell_1}) \\ \vdots \\ \widetilde{\mathbb{C}}(Y_{\ell_j}) \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbb{M}}(Y_{\ell_1}) \\ \vdots \\ \widetilde{\mathbb{M}}(Y_{\ell_j}) \end{bmatrix} \cdot \mathbb{S}. \tag{19}$$

If we can derive $\mathtt{s}_{16383}, \ldots, \mathtt{s}_0, \mathtt{w}_{127}, \ldots, \mathtt{w}_0$ and $\mathtt{u}_{127}, \ldots, \mathtt{u}_0$ that are consistent with them, then this gives us U and W that have $L_1 = \left[W \cdot [\![Y_{\ell_1}]\!]\right]^{2^{127}}, \ldots, L_j = \left[W \cdot [\![Y_{\ell_j}]\!]\right]^{2^{127}}$ as j solutions to (15) and (16) .

Our Algorithm. We are now ready to present our algorithm to derive U and W. It turns out that it is not possible to solve (19) if we use all the $\alpha_{\max}$ constants from $\mathbb{Y}_r$. Therefore, we need to choose some of the constants from $\mathbb{Y}_r$, and this turns out to be a non-trivial task. We follow a greedy method and our approach is to list $Y_1, \ldots, Y_{\alpha_{\max}}$ in the increasing order of the number of conditions $\#\mathbb{I}(Y_\ell)$. For the constants with the same number of conditions, we list them in the lexicographic order. Assume that $\mathbb{Y}_r = \{Y_1, \ldots, Y_{\alpha_{\max}}\}$ is listed with this order.

1. First, initialize $\widetilde{\mathbb{C}}$ as an empty binary column vector, and $\widetilde{\mathbb{M}}$ as a binary $0 \times (16384 + 128)$ matrix.
2. Next, execute Steps 3 and 4 for $i = 1$ to $\alpha_{\max}$.
3. Apply the Gaussian elimination to the following system of linear equations and see if it can be solved.

$$\begin{bmatrix} \widetilde{\mathbb{C}} \\ \widetilde{\mathbb{C}}(Y_i) \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbb{M}} \\ \widetilde{\mathbb{M}}(Y_i) \end{bmatrix} \cdot \mathbb{S} \tag{20}$$

4. If (20) has a solution, then let $\widetilde{\mathbb{C}} \leftarrow \begin{bmatrix} \widetilde{\mathbb{C}} \\ \widetilde{\mathbb{C}}(Y_i) \end{bmatrix}$ and $\widetilde{\mathbb{M}} \leftarrow \begin{bmatrix} \widetilde{\mathbb{M}} \\ \widetilde{\mathbb{M}}(Y_i) \end{bmatrix}$.
5. Finally, return $\widetilde{\mathbb{C}}$ and $\widetilde{\mathbb{M}}$.

Result. The execution of the algorithm gives us $\widetilde{\mathbb{M}}$ of the form presented in Fig. 5. The matrix is in the row echelon form where the lower left part of the elements are zeros.

We can arbitrarily fix $\mathtt{w}_{19}, \ldots, \mathtt{w}_0$, and then $\mathtt{w}_{57}, \ldots, \mathtt{w}_{20}$ are uniquely determined. We then arbitrarily fix $\mathtt{w}_{76}, \ldots, \mathtt{w}_{58}$, and then $\mathtt{w}_{127}, \ldots, \mathtt{w}_{77}$ are uniquely determined. At this point, all the bits of $W = \mathtt{w}_{127} \ldots \mathtt{w}_0$ are fixed, and we substitute them into $\mathtt{s}_{128i+j} = \mathtt{u}_i \mathtt{w}_j$ and see if we can determine $U = \mathtt{u}_{127} \ldots \mathtt{u}_0$.

It turns out that it is indeed possible if we let $\mathtt{w}_{76}, \ldots, \mathtt{w}_{58} \mathtt{w}_{19}, \ldots, \mathtt{w}_0 = 0^{39}$, which gives us $W =$ `0xa288088a 02a88000 00eff100 0e100000`, and $N = U$ and $N' = U' = U \oplus W^{-1}$ presented in (9), where the bits of U that can be fixed to any value are fixed to 0. Other results in (11), (12), and (13) are obtained with different values of $\mathtt{w}_{76}, \ldots, \mathtt{w}_{58} \mathtt{w}_{19}, \ldots, \mathtt{w}_0$, which are $0^{38}1$ for (11), $0^{37}10$ for (12), and $0^{37}11$ for (13).

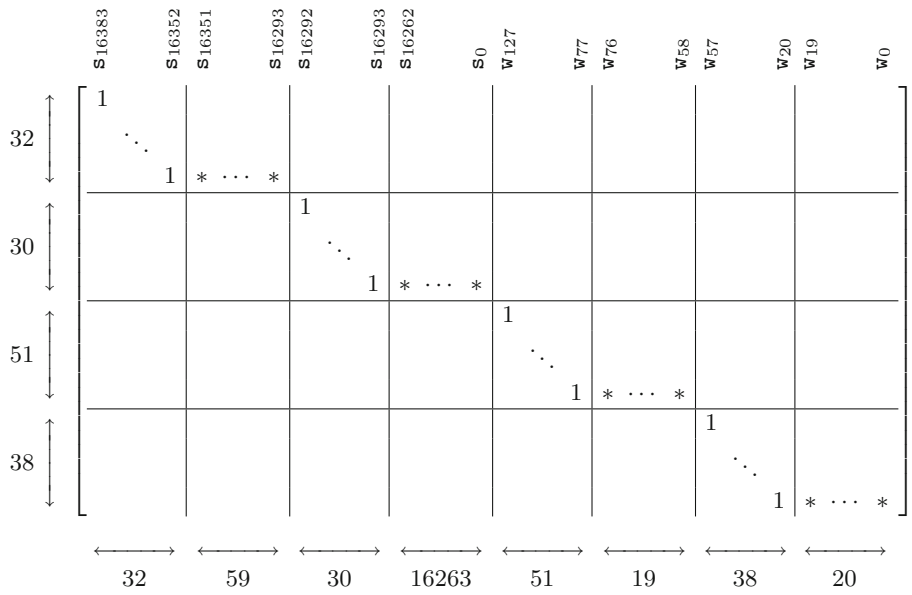

Fig. 5. The output $\widetilde{\mathbb{M}}$ of our algorithm

5.4 Applications to Other Values of r

The algorithm presented in the previous section can be naturally applied to other values of r. We present in Fig. 6 results of applying our algorithm on several values of r. The figure in $\#L$ shows the number of solutions (in L) that we can cover, and this suggests that we have identified N and N' such that $\Pr_L[\mathsf{Coll}_L(r, N, N')] \geq \#L/2^n$. The figure in $\#L/(\ell_N + 1)$ is normalized by dividing $\#L$ with the degree $(\ell_N + 1)$ of the polynomial, and we have $\ell_N = 1$ in our algorithm. The figure in α_r shows the value of α_r, and Proposition 3 states that we have $\Pr_L[\mathsf{Coll}_L(r, N, N')] \leq \alpha_r(\ell_N + 1)/2^n$ for any N and N'.

We see that, for these values of r, our algorithm gives N and N' such that the counter-collision probability is close to the upper bound in Proposition 3, and this suggests that Proposition 3 is tight up to a factor of about 4 to 16 depending on the value of r. However, there are other values of r where our algorithm does not work. We see that for $r =$ `0x2aaaaaab` and `0xd5555555`, it fails to give N and N' with a high counter-collision probability.

The existence of N and N' with a high counter-collision probability even for several values of r suggests that, if we rely on the sum bound in (6), the constants in security bounds in (1) and (2) cannot be significantly reduced. Now a natural question is whether it is possible to avoid using the sum bound, and if so, whether this leads to improved security bounds. In the next section, we answer these questions positively.

r	$\#L$	$\#L/(\ell_N+1)$	α_r
0x00000005	17	$2^{3.09}$	$2^{6.48}$
0x00000055	59	$2^{4.88}$	$2^{9.07}$
0x00000555	298	$2^{7.22}$	$2^{11.60}$
0x00005555	1930	$2^{9.91}$	$2^{14.09}$
0x00055555	13115	$2^{12.68}$	$2^{16.49}$
0x00555555	90134	$2^{15.46}$	$2^{18.77}$
0x05555555	667663	$2^{18.35}$	$2^{20.77}$
0x55555555	1762290	$2^{19.75}$	$2^{21.75}$
0x2aaaaaab	35	$2^{4.13}$	$2^{21.75}$
0xaaaaaaab	1762290	$2^{19.75}$	$2^{21.75}$
0xd5555555	35	$2^{4.13}$	$2^{21.75}$

Fig. 6. Summary of application of our algorithm to several values of r

6 Improving GCM Security Bounds

6.1 Avoiding the Sum Bound

For $0 \le r < r' \le 2^{32}-1$ and two distinct nonces N and N' which are not 96 bits, consider deriving the upper bound on $\Pr_L[\mathsf{Coll}_L(r,N,N') \vee \mathsf{Coll}_L(r',N,N')]$, i.e., $\Pr_L\left[\mathsf{inc}^r(I[0]) = I'[0] \vee \mathsf{inc}^{r'}(I[0]) = I'[0]\right]$, where $I[0] \leftarrow \mathsf{GHASH}_L(\varepsilon, N)$ and $I'[0] \leftarrow \mathsf{GHASH}_L(\varepsilon, N')$. The first step is to replace the arithmetic additions by r and r' with the xor of some constants $Y \in \mathbb{Y}_r$ and $Y' \in \mathbb{Y}_{r'}$. We obtain the following upper bound.

$$\Pr_L\left[I[0] \oplus [\![Y]\!] = I'[0] \text{ for some } Y \in \mathbb{Y}_r \vee I[0] \oplus [\![Y']\!] = I'[0] \text{ for some } Y' \in \mathbb{Y}_{r'}\right] \tag{21}$$

The proof in [12,13] relies on the sum bound, and (6) suggests the use of

$$\sum_{Y \in \mathbb{Y}_r} \Pr_L\left[I[0] \oplus [\![Y]\!] = I'[0]\right] + \sum_{Y' \in \mathbb{Y}_{r'}} \Pr_L\left[I[0] \oplus [\![Y']\!] = I'[0]\right]$$

as the upper bound on (21). We now present the following simple lemma.

Lemma 1. *Fix $0 \le r < r' \le 2^{32}-1$, and consider $Y \in \{0,1\}^{32}$ such that $Y \in \mathbb{Y}_r$ and $Y \in \mathbb{Y}_{r'}$. Then there does not exist $X \in \{0,1\}^n$ that satisfies $\mathsf{inc}^r(X) = X \oplus [\![Y]\!]$ and $\mathsf{inc}^{r'}(X) = X \oplus [\![Y]\!]$ simultaneously.*

Proof. Suppose for a contradiction that there exists $X \in \{0,1\}^n$ that satisfies both $\mathsf{inc}^r(X) = X \oplus [\![Y]\!]$ and $\mathsf{inc}^{r'}(X) = X \oplus [\![Y]\!]$. From $\mathsf{inc}^r(X) = \mathsf{inc}^{r'}(X)$, we have $\mathsf{inc}^{r'-r}(X) = X$. This is a contradiction as $r' - r \not\equiv 0 \bmod 2^{32}$, and hence $\mathsf{lsb}_{32}(\mathsf{inc}^{r'-r}(X))$ and $\mathsf{lsb}_{32}(X)$ cannot take the same value. □

It follows from Lemma 1 that

$$\sum_{Y \in \mathbb{Y}_r} \Pr_L\left[I[0] \oplus [\![Y]\!] = I'[0]\right] + \sum_{Y' \in \mathbb{Y}_{r'} \setminus \mathbb{Y}_r} \Pr_L\left[I[0] \oplus [\![Y']\!] = I'[0]\right] \tag{22}$$

is also an upper bound on (21). If the cardinality of $\mathbb{Y}_r \cap \mathbb{Y}_{r'}$ is small, then (22) does not seem to give us any improvement. However, it turns out that there is a non-obvious effect of considering the cardinality of $\mathbb{Y}_r \cap \mathbb{Y}_{r'}$, and (22) indeed gives us improved security bounds on GCM.

This observation motivates us to consider another upper bound on (21), which is

$$\sum_{Y \in \mathbb{Y}_r \cup \mathbb{Y}_{r'}} \Pr_L\left[I[0] \oplus [\![Y]\!] = I'[0]\right]. \tag{23}$$

In what follows, we present improved security bounds of GCM with (22) and (23).

6.2 Towards Improved Security Bounds

Consider an adversary $\mathcal{A}$ in the privacy game. As outlined in Sect. 5.1, we may focus on non-adaptive adversaries and consider the list of counter values in (3). The privacy advantage can be derived as (4), and $\Pr_L$ [Case (A) holds] is obtained as (5). We focus on $\Pr_L$ [Case (B) holds], i.e., we are interested in the probability of having a collision $I_i[j] = I_{i'}[j']$ for some (i, j, i', j'), where $1 \le i' < i \le q$, $0 \le j' \le m_{i'}$, and $0 \le j \le m_i$. For each $2 \le i \le q$, we have at most $(m_1 + 1) + (m_2 + 1) + \cdots + (m_{i-1} + 1) + (i - 1)m_i$ cases of (j, i', j') to consider. To see this, we observe that for $I_i[0]$, we need to consider

$$I_i[0] \in \{I_{i'}[0], I_{i'}[1], \ldots, I_{i'}[m_{i'}]\} \text{ for some } 1 \le i' < i, \tag{24}$$

and thus for $j = 0$, we have $(m_1 + 1) + (m_2 + 1) + \cdots + (m_{i-1} + 1)$ cases of (i', j') to consider. See Fig. 7 (left). For $I_i[1], I_i[2], \ldots, I_i[m_i]$, we consider

$$\begin{array}{c} I_i[1] \in \{I_1[0], I_2[0], \ldots, I_{i-1}[0]\}, \\ I_i[2] \in \{I_1[0], I_2[0], \ldots, I_{i-1}[0]\}, \\ \vdots \\ I_i[m_i] \in \{I_1[0], I_2[0], \ldots, I_{i-1}[0]\}, \end{array} \tag{25}$$

and we thus have $(i - 1)$ cases of (i', j') for each $1 \le j \le m_i$. See Fig. 7 (right). We note that we can exclude the cases $I_i[j] = I_{i'}[j']$ for $1 \le j \le m_i$, $1 \le i' < i$, and $1 \le j' \le m_{i'}$, as these cases are covered in (24) or in another case of (25).

So far, we have proceeded as was done in [12,13]. Now for $0 \le a \le b \le 2^{32} - 1$ and two distinct nonces N and N' which are not 96 bits, let $\mathsf{Coll}_L([a..b], N, N')$ denote the event

$$\mathsf{inc}^r(\mathsf{GHASH}_L(\varepsilon, N)) = \mathsf{GHASH}_L(\varepsilon, N') \text{ for some } a \le r \le b.$$

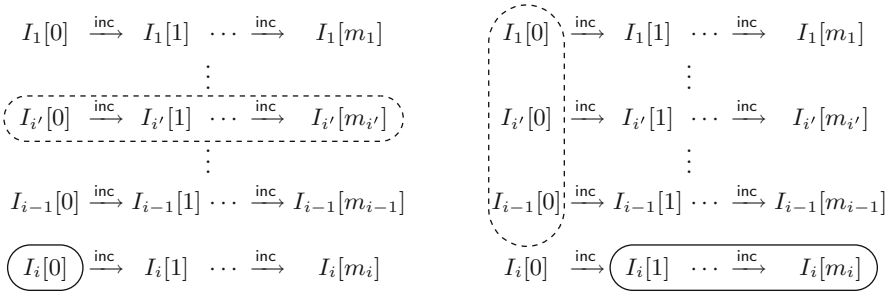

Fig. 7. Cases of (i', j') to consider for $j = 0$ (left) and for $1 \le j \le m_i$ (right)

We see that (24) is equivalent to $\mathsf{inc}^0(I_{i'}[0]) = I_i[0] \vee \mathsf{inc}^1(I_{i'}[0]) = I_i[0] \vee \cdots \vee \mathsf{inc}^{m_{i'}}(I_{i'}[0]) = I_i[0]$ for some $1 \le i' < i$, and the probability can be evaluated as

$$\sum_{1 \le i' < i} \Pr_L \left[\mathsf{Coll}_L([0..m_{i'}], N_{i'}, N_i)\right]. \tag{26}$$

With respect to (25), we rearrange them as $I_{i'}[0] \in \{I_i[1], I_i[2], \ldots, I_i[m_i]\}$ for some $1 \le i' < i$. We see that this is equivalent to $\mathsf{inc}^1(I_i[0]) = I_{i'}[0] \vee \mathsf{inc}^2(I_i[0]) = I_{i'}[0] \vee \cdots \vee \mathsf{inc}^{m_i}(I_i[0]) = I_{i'}[0]$ for some $1 \le i' < i$, and the upper bound on the probability can be evaluated as

$$\sum_{1 \le i' < i} \Pr_L \left[\mathsf{Coll}_L([1..m_i], N_i, N_{i'})\right] \le \sum_{1 \le i' < i} \Pr_L \left[\mathsf{Coll}_L([0..m_i], N_i, N_{i'})\right]. \tag{27}$$

6.3 Improving the Security Bounds with (22)

To apply (22) on (26) and (27), we define $\mathbb{W}_r \subseteq \{0,1\}^{32}$, for $0 \le r \le 2^{32} - 1$, as

$$\mathbb{W}_0 \stackrel{\text{def}}{=} \mathbb{Y}_0 \text{ and } \mathbb{W}_r \stackrel{\text{def}}{=} \mathbb{Y}_r \setminus (\mathbb{Y}_0 \cup \mathbb{Y}_1 \cup \cdots \cup \mathbb{Y}_{r-1}) \text{ for } r \ge 1.$$

We denote its cardinality as $w_r \stackrel{\text{def}}{=} \#\mathbb{W}_r$ and let $w_{\max} \stackrel{\text{def}}{=} \max\{w_r \mid 0 \le r \le 2^{32} - 1\}$. We show the following lemma.

Lemma 2. *For $0 \le m \le 2^{32} - 1$ and two distinct nonces N and N' which are not 96 bits, it holds that $\Pr_L[\mathsf{Coll}_L([0..m], N, N')] \le w_{\max}(m+1)(\ell_N + 1)/2^n$, where $|N|_n, |N'|_n \le \ell_N$.*

Proof. Recall that $\mathsf{Coll}_L([0..m], N, N')$ is the event $\mathsf{inc}^0(I[0]) = I'[0] \vee \mathsf{inc}^1(I[0]) = I'[0] \vee \cdots \vee \mathsf{inc}^m(I[0]) = I'[0]$, and the probability can be evaluated as

$$\sum_{0 \le r \le m} \sum_{Y \in \mathbb{Y}_r \setminus (\mathbb{Y}_0 \cup \mathbb{Y}_1 \cup \cdots \cup \mathbb{Y}_{r-1})} \Pr_L \left[I[0] \oplus [\![Y]\!] = I'[0]\right] \le \sum_{0 \le r \le m} \frac{w_{\max}(\ell_N + 1)}{2^n},$$

since $I[0] \oplus [\![Y]\!] = I'[0]$ is a non-trivial equation in L over $\mathrm{GF}(2^n)$ of degree at most $\ell_N + 1$. □

It follows that

$$(26)+(27) \le \sum_{1\le i'<i} \frac{w_{\max}(m_{i'}+1)(\ell_N+1)}{2^n} + \sum_{1\le i'<i} \frac{w_{\max}(m_i+1)(\ell_N+1)}{2^n}$$
$$\le \frac{w_{\max}(\ell_N+1)}{2^n}\left(\left(\sum_{1\le i'<i}(m_{i'}+1)\right)+(i-1)(m_i+1)\right),$$

and by taking the summation with respect to i, we obtain $\Pr_L[\text{Case (B) holds}] \le w_{\max}(q-1)(\sigma+q)(\ell_N+1)/2^n$, since

$$\sum_{2\le i\le q}\left(\left(\sum_{1\le i'<i}(m_{i'}+1)\right)+(i-1)(m_i+1)\right) \le (q-1)(\sigma+q).$$

From (5), $\Pr_L[\text{Case (A) holds}] + \Pr_L[\text{Case (B) holds}]$ is at most

$$\frac{(\sigma+q)(\ell_N+1)}{2^n} + \frac{w_{\max}(q-1)(\sigma+q)(\ell_N+1)}{2^n} \le \frac{w_{\max}q(\sigma+q)(\ell_N+1)}{2^n},$$

and it remains to evaluate the value of $w_{\max}$, which is shown in the lemma below.

Lemma 3. $w_{\max} \le 32$.

A proof is presented in Appendix A.

We are now ready to present the improved security bound based on (22).

Theorem 2. *With the same notation as in Proposition 1, we have*

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{GCM}[\mathrm{Perm}(n),\tau]}(\mathcal{A}) \le \frac{0.5(\sigma+q+1)^2}{2^n} + \frac{32q(\sigma+q)(\ell_N+1)}{2^n}. \tag{28}$$

We have focused on the privacy result, but the authenticity result can also be obtained as follows.

Theorem 3. *With the same notation as in Proposition 2, we have*

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{GCM}[\mathrm{Perm}(n),\tau]}(\mathcal{A}) \le \frac{0.5(\sigma+q+q'+1)^2}{2^n} + \frac{32(q+q')(\sigma+q+1)(\ell_N+1)}{2^n} + \frac{q'(\ell_A+1)}{2^\tau}. \tag{29}$$

Proofs follow the corresponding proofs in [13, Appendix D] for privacy and [13, Appendix E] for authenticity. For privacy, the difference is the analysis of Case (B) in [13, Appendix D], which is presented in this section, and for authenticity, the difference is the analysis of Case (B) and Case (D) in [13, Appendix E], where we can directly apply the analysis of this section.

6.4 Improving the Security Bounds with (23)

To apply (23) on (26) and (27), we define $\mathbb{Z}_r \subseteq \{0,1\}^{32}$, for $0 \le r \le 2^{32}-1$, as

$$\mathbb{Z}_r \stackrel{\text{def}}{=} \mathbb{Y}_0 \cup \mathbb{Y}_1 \cup \cdots \cup \mathbb{Y}_r,$$

and denote its cardinality as $z_r \stackrel{\text{def}}{=} \#\mathbb{Z}_r$. We also let $z_{\max} \stackrel{\text{def}}{=} \max\{z_r \mid 0 \le r \le 2^{32}-1\}$. We show the following lemma.

Lemma 4. *For $0 \le m \le 2^{32}-1$ and two distinct nonces N and N' which are not 96 bits, it holds that $\Pr_L[\mathsf{Coll}_L([0..m], N, N')] \le z_{\max}(\ell_N+1)/2^n$, where $|N|_n, |N'|_n \le \ell_N$.*

Proof. The upper bound on $\Pr_L[\mathsf{Coll}_L([0..m], N, N')]$ can be evaluated as

$$\sum_{Y \in \mathbb{Y}_0 \cup \mathbb{Y}_1 \cup \cdots \cup \mathbb{Y}_m} \Pr_L\left[I[0] \oplus [\![Y]\!] = I'[0]\right] \le \frac{z_{\max}(\ell_N+1)}{2^n},$$

since $I[0] \oplus [\![Y]\!] = I'[0]$ is a non-trivial equation of degree at most ℓ_N+1. □

It follows that

$$(26)+(27) \le 2\sum_{1 \le i' < i} \frac{z_{\max}(\ell_N+1)}{2^n} \le \frac{2(i-1)z_{\max}(\ell_N+1)}{2^n},$$

and by taking the summation with respect to i, we obtain $\Pr_L[\text{Case (B) holds}] \le z_{\max}q^2(\ell_N+1)/2^n$. We use (5) to have

$$\Pr_L[\text{Case (A) holds}] + \Pr_L[\text{Case (B) holds}] \le \frac{(\sigma+q)(\ell_N+1)}{2^n} + \frac{z_{\max}q^2(\ell_N+1)}{2^n},$$

and it remains to evaluate the value of $z_{\max}$, which is stated in the following lemma.

Lemma 5. $z_{\max} \le 2^{32}$.

We have $\mathbb{Z}_r \subseteq \{0,1\}^{32}$, and hence the lemma follows. We note that the analysis is tight, as $\mathsf{str}_{32}(r)$ is always included in $\mathbb{Y}_r$, and the union $\mathbb{Y}_0 \cup \mathbb{Y}_1 \cup \cdots \cup \mathbb{Y}_{2^{32}-1}$ covers $\{0,1\}^{32}$.

We have the following improved security bound based on (23).

Theorem 4. *With the same notation as in Proposition 1, we have*

$$\mathbf{Adv}^{\text{priv}}_{\text{GCM}[\text{Perm}(n),\tau]}(\mathcal{A}) \le \frac{0.5(\sigma+q+1)^2}{2^n} + \frac{(\sigma+q)(\ell_N+1)}{2^n} + \frac{2^{32}q^2(\ell_N+1)}{2^n}. \tag{30}$$

The authenticity theorem is given as follows.

Theorem 5. *With the same notation as in Proposition 2, we have*

$$\mathbf{Adv}^{\text{auth}}_{\text{GCM}[\text{Perm}(n),\tau]}(\mathcal{A}) \le \frac{0.5(\sigma+q+q'+1)^2}{2^n} + \frac{(\sigma+q+q')(\ell_N+1)}{2^n} + \frac{2^{32}q(q+q')(\ell_N+1)}{2^n} + \frac{q'(\ell_A+1)}{2^\tau}. \tag{31}$$

6.5 Discussions

We present a comparison of the three privacy bounds in (1), (28), and (30). We see that (28) is always smaller than (1), hence we focus on the comparison between (28) and (30). By simplifying (28) $\leq$ (30), we obtain

$$\left(32 - \frac{1}{q}\right)\left(\frac{\sigma}{q} + 1\right) \leq 2^{32}.$$

This suggests that if σ/q, the average block length of each query, is at most $2^{32}/32$ blocks, then (28) is smaller, where $2^{32}/32$ blocks amount to 2 Gbytes from $n = 128$. Similarly, for authenticity, (29) is always better than (2). By simplifying (29) $\leq$ (31), we obtain

$$\frac{\sigma}{q}\left(32 - \frac{1}{q+q'}\right) + \frac{1}{q} + 32 \leq 2^{32}.$$

As with the case of privacy, this suggests that if σ/q is at most $2^{32}/32$ blocks, which is about 2 Gbytes, then (29) gives a better bound than (31).

7 Conclusions

In this paper, we developed an algorithm to generate nonces that have a high counter-collision probability, and showed concrete examples of nonces as the results of our experiments. This implies that, if we use the sum bound in the security proof, then the security bounds of [12] are tight within a factor of about 4. We next showed that it is possible to avoid using the sum bound. We presented improved security bounds of GCM, and one of our security bounds suggests that the security of GCM is close to what was originally claimed by the designers in [18,19].

There are several interesting research directions. With respect to the generation of nonces, it would be interesting to extend our algorithm to handle nonces of different lengths. It would also be interesting to study the security of variants of GCM, including SGCM [27] and MGCM [20].

Acknowledgments. The authors received useful comments from participants of Dagstuhl Seminar 12031 (Symmetric Cryptography), ASK 2012 (Asian Workshop on Symmetric Key Cryptography), Early Symmetric Crypto (ESC) seminar 2013, and "Shin-Akarui-Angou-Benkyou-Kai." In particular, the authors thank Antoine Joux for motivating this work at Dagstuhl Seminar 12031. The work by Tetsu Iwata was supported in part by JSPS KAKENHI, Grant-in-Aid for Scientific Research (B), Grant Number 26280045, and was carried out in part while visiting Nanyang Technological University, Singapore.

A Proof of Lemma 3

Let x and c be integers such that $0 \le x \le 31$ and $0 \le c \le 2^x - 1$. Throughout the proof of Lemma 3, we abuse the notation and regard an integer $0 \le a \le 2^{32} - 1$ and its 32-bit binary representation, $\mathsf{str}_{32}(a)$, identically. For a 32-bit string $\mathsf{a}_{31} \ldots \mathsf{a}_0$, the i-th bit refers to a_i. We show the proof of Lemma 3 with the following two claims.

Claim. $2^x + c \in \mathbb{Y}_{2^x - c}$.

Proof. We have $2^x + c \in \mathbb{Y}_{2^x - c}$ if there exists $Y \in \{0,1\}^{32}$ that satisfies $Y + (2^x - c) = Y \oplus (2^x + c)$, which is equivalent to $2^x + c = (Y + (2^x - c)) \oplus Y$. Now let $Y \leftarrow \mathsf{str}_{32}(c)$. Then the right hand side is $(c + (2^x - c)) \oplus c$, which is equal to the left hand side from $0 \le c \le 2^x - 1$. Therefore, we have $2^x + c \in \mathbb{Y}_{2^x - c}$. □

Claim. $2^x + c \notin \mathbb{Y}_r$ for $0 \le r < 2^x - c$.

Proof. Let d be an integer such that $c < d \le 2^x$. We show that there does not exist $Y \in \{0,1\}^{32}$ that satisfies $2^x + c = (Y + 2^x - d) \oplus Y$, implying $2^x + c \notin \mathbb{Y}_{2^x - d}$. From $c < d \le 2^x$, we have $2^x + c = 2^x \oplus c$ and $2^x - d = 2^x - 1 - (d-1) = (2^x - 1) \oplus (d - 1)$.

We first consider the case $d - 1 = c$. We see that the 0-th bit of $2^x + c$ is different from the 0-th bit of $2^x - d$. Therefore, there does not exist Y that satisfies $2^x + c = (Y + 2^x - d) \oplus Y$.

We next consider the case $d - 1 > c$. Let $d' = d - 1$, and let $\mathsf{str}_{32}(c) = \mathsf{c}_{31} \ldots \mathsf{c}_0$ and $\mathsf{str}_{32}(d') = \mathsf{d}'_{31} \ldots \mathsf{d}'_0$ be the binary representations of c and d'. Define $\ell \stackrel{\text{def}}{=} \max\{i \mid \mathsf{d}'_i \ne \mathsf{c}_i\}$. Then we have $\mathsf{d}'_\ell = 1$ and $\mathsf{c}_\ell = 0$ from $d - 1 > c$. This implies that the ℓ-th bit of $2^x + c$ and the ℓ-th bit of $2^x - d$ are both 0. Now from $\mathsf{d}'_{\ell+1} = \mathsf{c}_{\ell+1}$ and the fact that the $(\ell+1)$-st bit of 2^x and the $(\ell+1)$-st bit of $2^x - 1$ are different, we necessary have that the $(\ell+1)$-st bit of $2^x + c$ and the $(\ell+1)$-st bit of $2^x - d$ are different. In order the equality of $2^x + c = (Y + 2^x - d) \oplus Y$ to hold, we must have a carry to the $(\ell+1)$-st bit in computing $Y + 2^x - d$. However, it is impossible to have the carry since the ℓ-th bit of $2^x - d$ is 0. Therefore, there does not exist Y that satisfies $2^x + c = (Y + 2^x - d) \oplus Y$. □

The two claims show $2^x + c \in \mathbb{W}_{2^x - c}$. Now any integer between 1 and $2^{32} - 1$ can be uniquely represented in the form of $2^x + c$ for some $0 \le x \le 31$ and $0 \le c \le 2^x - 1$. The uniqueness follows from the fact that, if $(x, c) \ne (x', c')$, then $2^x + c \ne 2^{x'} + c'$. We note that 0 cannot be represented in the form of $2^x + c$, which is an element of $\mathbb{Y}_0$, and is not included in $\mathbb{Y}_r$ for $r \ge 1$, since $0 = (Y + r) \oplus Y$ cannot hold for $r \ge 1$. This implies that $\mathbb{W}_r$ for $r \ge 1$ can be written as $\mathbb{W}_r = \{2^x + c \mid r = 2^x - c, 0 \le x \le 31, 0 \le c \le 2^x - 1\}$. We can specifically list the elements of $\mathbb{W}_r$ as

$$\mathbb{W}_r = \{2^{31} + (2^{31} - 2^x + c), 2^{30} + (2^{30} - 2^x + c), \ldots, 2^{x+1} + (2^{x+1} - 2^x + c), 2^x + c\},$$

where $x = \lceil \log_2 r \rceil$ and $c = r - 2^x$. This proves $\#\mathbb{W}_r = 32 - \lceil \log_2 r \rceil$, and hence we have $w_{\max} \le 32$. □

In [23], we present a small-scale example that supports our claims.

References

1. Abdelraheem, M.A., Beelen, P., Bogdanov, A., Tischhauser, E.: Twisted polynomials and forgery attacks on GCM. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 762–786. Springer, Heidelberg (2015)
2. Aoki, K., Yasuda, K.: The security and performance of "GCM" when short multiplications are used instead. In: Kutyłowski, M., Yung, M. (eds.) Inscrypt 2012. LNCS, vol. 7763, pp. 225–245. Springer, Heidelberg (2013)
3. Bellare, M., Namprempre, C.: Authenticated encryption: relations among notions and analysis of the generic composition paradigm. J. Cryptology **21**(4), 469–491 (2008)
4. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006)
5. Bogdanov, A.: Challenges and advances in authenticated encryption. Annual Workshop of TCCM-CACR (2014)
6. CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness. http://competitions.cr.yp.to/caesar.html
7. Dworkin, M.: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D (2007)
8. Ferguson, N.: Authentication Weaknesses in GCM. Public comments to NIST (2005). http://csrc.nist.gov/groups/ST/toolkit/BCM/comments.html
9. Handschuh, H., Preneel, B.: Key-recovery attacks on universal hash function based MAC algorithms. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 144–161. Springer, Heidelberg (2008)
10. IEEE Standard for Local and Metropolitan Area Networks Media Access Control (MAC) Security. IEEE Std 802.1AE-2006 (2006)
11. Information Technology – Security Techniques – Authenticated Encryption, ISO/IEC 19772:2009. International Standard ISO/IEC 19772 (2009)
12. Iwata, T., Ohashi, K., Minematsu, K.: Breaking and repairing GCM security proofs. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 31–49. Springer, Heidelberg (2012)
13. Iwata, T., Ohashi, K., Minematsu, K.: Breaking and Repairing GCM Security Proofs. Cryptology ePrint Archive, Report 2012/438 (2012). http://eprint.iacr.org/
14. Joux, A.: Authentication Failures in NIST version of GCM. Public comments to NIST (2006). http://csrc.nist.gov/groups/ST/toolkit/BCM/comments.html
15. Leurent, G.: ARX Toolkit. http://www.di.ens.fr/~leurent/arxtools.html
16. Leurent, G.: Analysis of differential attacks in ARX constructions. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 226–243. Springer, Heidelberg (2012)
17. Leurent, G.: Construction of differential characteristics in ARX designs application to Skein. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 241–258. Springer, Heidelberg (2013)
18. McGrew, D.A., Viega, J.: The security and performance 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)
19. McGrew, D.A., Viega, J.: The Security and Performance of the Galois/Counter Mode of Operation (Full Version). Cryptology ePrint Archive, Report 2004/193 (2004). http://eprint.iacr.org/

20. Meloni, N., Nègre, C., Hasan, M.A.: High performance GHASH and impacts of a class of unconventional bases. J. Cryptographic Eng. **1**(3), 201–218 (2011)
21. Mouha, N., Velichkov, V., De Cannière, C., Preneel, B.: The differential analysis of S-functions. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) SAC 2010. LNCS, vol. 6544, pp. 36–56. Springer, Heidelberg (2011)
22. National Security Agency, Internet Protocol Security (IPsec) Minimum Essential Interoperability Requirements, IPMEIR Version 1.0.0 Core (2010). http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml
23. Niwa, Y., Ohashi, K., Minematsu, K., Iwata, T.: GCM Security Bounds Reconsidered. Cryptology ePrint Archive, Report 2015/214 (2015). http://eprint.iacr.org/
24. Procter, G., Cid, C.: On weak keys and forgery attacks against polynomial-based MAC schemes. In: Moriai, S. (ed.) FSE 2013. LNCS, vol. 8424, pp. 287–304. Springer, Heidelberg (2014)
25. Rogaway, P.: Authenticated-encryption with associated-data. In: Atluri, V. (ed.) ACM Conference on Computer and Communications Security, CCS 2002. pp. 98–107. ACM (2002)
26. Rogaway, P.: Evaluation of Some Blockcipher Modes of Operation. Investigation Reports on Cryptographic Techniques in FY 2010 (2011). http://www.cryptrec.go.jp/english/
27. Saarinen, M.-J.O.: SGCM: The Sophie Germain Counter Mode. Cryptology ePrint Archive, Report 2011/326 (2011). http://eprint.iacr.org/
28. Saarinen, M.-J.O.: Cycling attacks on GCM, GHASH and other polynomial MACs and hashes. In: Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 216–225. Springer, Heidelberg (2012)
29. Salowey, J., Choudhury, A., McGrew, D.A.: AES Galois Counter Mode (GCM) Cipher Suites for TLS. IETF RFC 5288 (2008)
30. Yap, W., Yeo, S.L., Heng, S., Henricksen, M.: Security analysis of GCM for communication. Secur. Commun. Networks **7**(5), 854–864 (2014)
31. Zhu, B., Tan, Y., Gong, G.: Revisiting MAC forgeries, weak keys and provable security of Galois/Counter Mode of operation. In: Abdalla, M., Nita-Rotaru, C., Dahab, R. (eds.) CANS 2013. LNCS, vol. 8257, pp. 20–38. Springer, Heidelberg (2013)