Breaking and Repairing GCM Security Proofs*

Tetsu Iwata[1], Keisuke Ohashi[1], and Kazuhiko Minematsu[2]

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

Abstract. In this paper, we study the security proofs of GCM (Galois/Counter Mode of Operation). We first point out that a lemma, which is related to the upper bound on the probability of a counter collision, is invalid. Both the original privacy and authenticity proofs by the designers are based on the lemma. We further show that the observation can be translated into a distinguishing attack that invalidates the main part of the privacy proof. It turns out that the original security proofs of GCM contain a flaw, and hence the claimed security bounds are not justified. A very natural question is then whether the proofs can be repaired. We give an affirmative answer to the question by presenting new security bounds, both for privacy and authenticity. As a result, although the security bounds are larger than what were previously claimed, GCM maintains its provable security. We also show that, when the nonce length is restricted to 96 bits, GCM has better security bounds than a general case of variable length nonces.

Keywords: GCM, counter-example, distinguishing attack, proof of security.

1 Introduction

GCM (Galois/Counter Mode of Operation) is the authenticated encryption mode of blockciphers designed by McGrew and Viega [26,27]. The mode is based on the counter mode encryption and the polynomial hash function, and the designers presented proofs of security both for privacy and authenticity [26,27]. It was selected as the NIST recommended blockcipher mode in 2007 [15], and is widely used in practice, e.g., in [1,2,4,5,6,7,8,14,17,19,20,25,33,34].

The security of GCM has been extensively evaluated. Ferguson pointed out that a forgery is possible if the tag length is short [16]. Joux showed that a part of the secret key can be obtained if the nonce is reused [21]. Handschuh and Preneel discussed weak keys of GCM and presented generalizations of Joux's attack [18]. Saarinen pointed out that GCM has more weak keys than previously known, and used the weak keys for forgery attacks [32]. See also [31] for comprehensive discussions on various aspects on GCM.

Despite aforementioned attacks, it is widely considered that the provable security results of GCM are sound, in the sense that the previous attacks do not contradict the claimed security bounds by the designers, and that no flaw in the proofs has been identified. Some of these attacks show the tightness of the security bounds, and others are outside the security model (e.g., nonce reuse). Therefore, there is no attack that undermines the security bounds or their proofs.

GCM uses the counter mode encryption, and the initial counter value is derived from a nonce, where there are two different ways to generate the initial counter value depending on the length of the nonce. When the nonce length is 96 bits, the initial counter value is the nonce padded with a constant. When the nonce length is not 96 bits, the polynomial hash function is applied to the nonce to obtain the initial counter value. In order to prove the security of GCM, one has to show that the probability of a counter collision is small. McGrew and Viega presented a lemma showing the upper bound on the probability in [27], which is the basis for both the privacy and authenticity proofs.

* A preliminary version of this paper appears in the proceedings of CRYPTO 2012. This is the full version.

In this paper, we first point out that the claimed lemma cannot be valid; the probability of a counter collision is larger than claimed. We show concrete counter-examples of two distinct nonces that invalidate the lemma. It turns out that the original security proofs (both for privacy and authenticity) of GCM contain a flaw, and hence the claimed security bounds are not justified.

We next translate the above observation into a distinguishing attack. The attack is simple and efficient. However, from the practical perspective, the success probability of the attack is insignificantly small, and it does not contradict the security bounds by the designers. On the other hand, the success probability is large enough to invalidate the main part of the privacy proof. In more detail, there are three terms in the privacy bound of GCM. The first one comes from the difference between a random permutation and a random function, the second one is the main part of the privacy proof that bounds the distinguishing probability of ciphertexts of GCM based on a random function from random strings (of the same lengths as the ciphertexts), and the last one bounds the forgery probability. The success probability of our distinguishing attack is larger than the second term, invalidating the main part of the privacy proof. Consequently, the security of GCM is not supported by the proofs.

Then a very natural question is whether the proofs can be repaired, or more generally, whether the security of GCM can ever be proved. In order to answer the question, we first introduce a combinatorial problem of quantifying a cardinality of a certain set of bit strings. The problem belongs to one of the problems of counting the number of output differences with non-zero probability of S-functions [29], which presents tools to analyze ARX systems (e.g., see [23]). One possible approach to solve the problem is to follow [29] (or [22]). In this paper, we take another approach and present a solution to the problem by giving a recursive formula that quantifies the cardinality. Basing on the solution, we present new security bounds on GCM, both for privacy and authenticity.

As a result, although the security bounds are larger than what were previously claimed, we show that GCM maintains its provable security. We also present provable security results of GCM when the nonce length is restricted to 96 bits, in which case GCM has better security bounds than a general case.

2 Preliminaries

Let $\{0,1\}^*$ be the set of all bit strings, and for an integer $\ell \geq 0$, let $\{0,1\}^\ell$ be a set of ℓ-bit strings. For a bit string $X \in \{0,1\}^*$, $|X|$ is its length in bits, and $|X|_\ell = \lceil |X|/\ell \rceil$ is the length in ℓ-bit blocks. The empty string is denoted as ε. Let 0^ℓ and 1^ℓ denote the bit strings of ℓ zeros and ones, respectively. We use the prefix $\mathsf{0x}$ for the hexadecimal notation, e.g., $\mathsf{0x63}$ is $01100011 \in \{0,1\}^8$. We also write $(\mathsf{0x0})^\ell$ to mean $0^{4\ell}$. For a bit string X and an integer ℓ such that $|X| \geq \ell$, $\mathsf{msb}_\ell(X)$ is the most significant ℓ bits (the leftmost ℓ bits) of X, and $\mathsf{lsb}_\ell(X)$ is the least significant ℓ bits (the rightmost ℓ bits) of X. For $X, Y \in \{0,1\}^*$, we write $X \| Y$, (X, Y), or simply XY to denote their concatenation. For a bit string X whose length in bits is a multiple of ℓ, we write its partition into ℓ-bit strings as $(X[1], \ldots, X[x]) \xleftarrow{\ell} X$, where $X[1], \ldots, X[x] \in \{0,1\}^\ell$ are unique bit strings such that $X[1] \| \ldots \| X[x] = X$. For non-negative integers a and ℓ with $a \leq 2^\ell - 1$, let $\mathsf{str}_\ell(a)$ be its ℓ-bit binary representation, i.e., if $a = a_{\ell-1}2^{\ell-1} + \cdots + a_1 2 + a_0$ for $a_{\ell-1}, \ldots, a_1, a_0 \in \{0,1\}$, then $\mathsf{str}_\ell(a) = a_{\ell-1} \ldots a_1 a_0 \in \{0,1\}^\ell$. For a bit string $X = X_{\ell-1} \ldots X_1 X_0 \in \{0,1\}^\ell$, let $\mathsf{int}(X)$ be the integer $X_{\ell-1}2^{\ell-1} + \cdots + X_1 2 + X_0$. For a finite set $\mathcal{X}$, $\#\mathcal{X}$ denotes its cardinality, and $X \xleftarrow{\$} \mathcal{X}$ means the uniform sampling of an element from $\mathcal{X}$ and assigning it to X.

Throughout this paper, we fix a block length n and a blockcipher $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$, where $\mathcal{K}$ is a non-empty set of keys. Unless otherwise specified, we let $n = 128$. We write E_K for the permutation specified by $K \in \mathcal{K}$, and $C = E_K(M)$ for 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 $\mathrm{GF}(2^n)$,

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

$L \leftarrow E_K(0^n)$
if $|N| = 96$ then $I[0] \leftarrow N \,\|\, 0^{31}1$
else $I[0] \leftarrow \mathsf{GHASH}_L(\varepsilon, N)$
$m \leftarrow |M|_n$
$S \leftarrow \mathsf{CTR}_K(I[0], m)$
$C \leftarrow M \oplus \mathsf{msb}_{|M|}(S)$
$\bar{T} \leftarrow E_K(I[0]) \oplus \mathsf{GHASH}_L(A, C)$
$T \leftarrow \mathsf{msb}_\tau(\bar{T})$
return $(C, T)$
```

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

$L \leftarrow E_K(0^n)$
if $|N| = 96$ then $I[0] \leftarrow N \,\|\, 0^{31}1$
else $I[0] \leftarrow \mathsf{GHASH}_L(\varepsilon, N)$
$\bar{T}^* \leftarrow E_K(I[0]) \oplus \mathsf{GHASH}_L(A, C)$
$T^* \leftarrow \mathsf{msb}_\tau(\bar{T}^*)$
if $T \neq T^*$ then return $\perp$
$m \leftarrow |C|_n$
$S \leftarrow \mathsf{CTR}_K(I[0], m)$
$M \leftarrow C \oplus \mathsf{msb}_{|C|}(S)$
return $M$
```

Fig. 1. The encryption and decryption algorithms of GCM

the finite field with 2^n elements. An n-bit string $a_{n-1} \dots a_1 a_0 \in \{0,1\}^n$ corresponds to a formal polynomial $a(\mathtt{x}) = a_{n-1} + a_{n-2}\mathtt{x} + \cdots + a_1\mathtt{x}^{n-2} + a_0\mathtt{x}^{n-1} \in \mathrm{GF}(2)[\mathtt{x}]$. When $n = 128$, the irreducible polynomial used in GCM is $p(\mathtt{x}) = 1 + \mathtt{x} + \mathtt{x}^2 + \mathtt{x}^7 + \mathtt{x}^{128}$.

3 Specification of GCM

We follow [27,28] with some notational changes. GCM is parameterized by a blockcipher $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ and a tag length τ, where $64 \leq \tau \leq n$. We write GCM$[E, \tau]$ for GCM that uses E and τ as parameters. Let GCM-$\mathcal{E}$ be the encryption algorithm and GCM-$\mathcal{D}$ be the decryption algorithm, which are defined in Fig. 1. GCM uses the counter mode encryption and the polynomial hash function over GF(2^n) as its subroutines. They are denoted CTR and GHASH and are defined in Fig. 2. Figure 3 illustrates the overall structure of GCM-$\mathcal{E}$.

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 inputs, and returns a pair of a ciphertext $C \in \{0,1\}^*$ and a tag $T \in \{0,1\}^\tau$ as an output, where $1 \leq |N| \leq 2^{n/2} - 1$, $0 \leq |A| \leq 2^{n/2} - 1$, $0 \leq |M| \leq n(2^{32} - 2)$, and $|C| = |M|$. We write $(C, T) \leftarrow \text{GCM-}\mathcal{E}_K^{N,A}(M)$. 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 inputs, and returns either a plaintext $M \in \{0,1\}^*$ or the symbol $\perp$ indicating that the inputs are invalid. We write $M \leftarrow \text{GCM-}\mathcal{D}_K^{N,A}(C, T)$ or $\perp \leftarrow \text{GCM-}\mathcal{D}_K^{N,A}(C, T)$.

In the definition of CTR, for a bit string $X \in \{0,1\}^n$, $\mathsf{inc}(X)$ treats the least significant 32 bits (the rightmost 32 bits) of X as a non-negative integer, and increments this value modulo 2^{32}, i.e.,

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

For $r \geq 0$, we write $\mathsf{inc}^r(X)$ to denote the r times iterative applications of inc on X, and $\mathsf{inc}^{-r}(X)$ to denote the r times iterative applications of the inverse function of inc on X. We use the convention that $\mathsf{inc}^0(X) = X$. In the definition of GHASH, the multiplication in line 7 is over GF(2^n). We remark that, if $|N| \neq 96$, we have $\mathsf{GHASH}_L(\varepsilon, N) = X[1] \cdot L^x \oplus \cdots \oplus X[x] \cdot L$, where $X = (X[1], \dots, X[x]) = N \,\|\, 0^{n|N|_n - |N|} \,\|\, \mathsf{str}_n(|N|)$.

4 Security Definitions

An adversary is a probabilistic algorithm that has access to one or more oracles. Let $\mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \dots}$ denote an adversary $\mathcal{A}$ interacting with oracles $\mathcal{O}_1, \mathcal{O}_2, \dots$, and $\mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \dots} \Rightarrow 1$ denote the event that $\mathcal{A}$, after interacting with $\mathcal{O}_1, \mathcal{O}_2, \dots$, outputs 1. The resources of $\mathcal{A}$ are measured in terms of time and query complexities. The time complexity includes the description size of $\mathcal{A}$, and

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

for $j \leftarrow 1$ to $m$ do
    $I[j] \leftarrow \mathsf{inc}(I[j-1])$
    $S[j] \leftarrow E_K(I[j])$
$S \leftarrow (S[1], S[2], \ldots, S[m])$
return $S$
```

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

$a \leftarrow n|A|_n - |A|$
$c \leftarrow n|C|_n - |C|$
$X \leftarrow A \,\|\, 0^a \,\|\, C \,\|\, 0^c \,\|\, \mathsf{str}_{n/2}(|A|) \,\|\, \mathsf{str}_{n/2}(|C|)$
$(X[1], \ldots, X[x]) \overset{n}{\leftarrow} X$
$Y \leftarrow 0^n$
for $j \leftarrow 1$ to $x$ do
    $Y \leftarrow L \cdot (Y \oplus X[j])$
return $Y$
```

Fig. 2. Subroutines used in the encryption and decryption algorithms

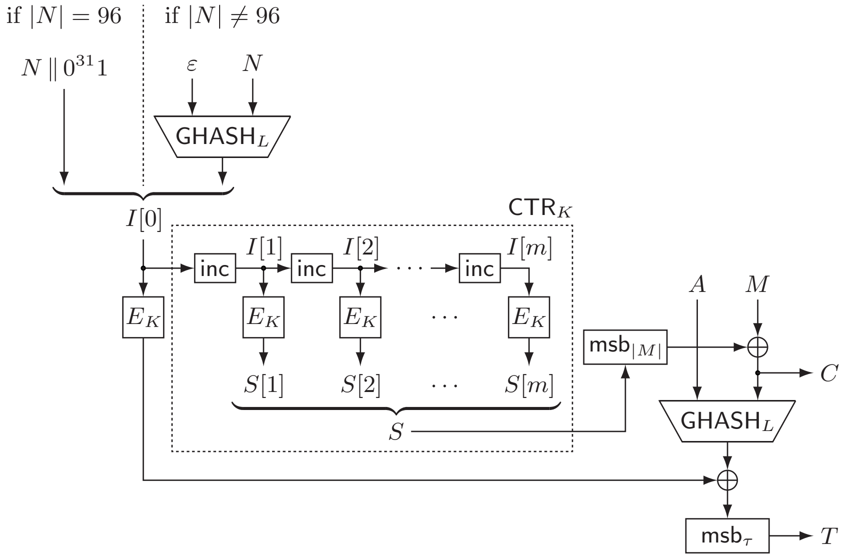

Fig. 3. The encryption algorithm of GCM

we fix a model of computation and a method of encoding. The query complexity includes the number of queries, the total length of queries, and the maximum length of queries, and a more precise definition is given in each theorem statement.

Following [10,30], we consider two security notions for GCM: privacy and authenticity. For privacy, we consider an adversary $\mathcal{A}$ that has access to a GCM encryption oracle or a random-bits oracle. The GCM encryption oracle takes (N, A, M) and returns $(C, T) \leftarrow \text{GCM-}\mathcal{E}_K^{N,A}(M)$. The random-bits oracle, \$, takes (N, A, M) and returns $(C, T) \overset{\$}{\leftarrow} \{0,1\}^{|M|+\tau}$. We define

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{GCM}[E,\tau]}(\mathcal{A}) \overset{\mathrm{def}}{=} \Pr[K \overset{\$}{\leftarrow} \mathcal{K} : \mathcal{A}^{\text{GCM-}\mathcal{E}_K} \Rightarrow 1] - \Pr[\mathcal{A}^{\$} \Rightarrow 1],$$

where the first probability is defined over the randomness of $\mathcal{A}$ and the choice of K, and the last is over the randomness of $\mathcal{A}$ and the random-bits oracle. We assume that $\mathcal{A}$ is nonce-respecting: $\mathcal{A}$ does not make two queries with the same nonce.

For authenticity, we consider an adversary $\mathcal{A}$ that has access to GCM encryption and decryption oracles. The GCM decryption oracle takes (N, A, C, T) and returns $M \leftarrow \text{GCM-}\mathcal{D}_K^{N,A}(C, T)$ or $\perp \leftarrow \text{GCM-}\mathcal{D}_K^{N,A}(C, T)$. We define

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{GCM}[E,\tau]}(\mathcal{A}) \overset{\mathrm{def}}{=} \Pr[K \overset{\$}{\leftarrow} \mathcal{K} : \mathcal{A}^{\text{GCM-}\mathcal{E}_K, \text{GCM-}\mathcal{D}_K} \text{ forges}],$$

where the probability is defined over the randomness of $\mathcal{A}$ and the choice of K, and the adversary forges if the GCM decryption oracle returns a bit string (other than $\perp$) for a query (N, A, C, T), but (C, T) was not previously returned to $\mathcal{A}$ from the encryption oracle for a query (N, A, M). As in the privacy notion, we assume that $\mathcal{A}$ is nonce-respecting: $\mathcal{A}$ does not make two queries to the encryption oracle with the same nonce. 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. Without loss of generality, we assume that $\mathcal{A}$ does not make trivial queries: if the encryption oracle returns (C, T) for a query (N, A, M), then $\mathcal{A}$ does not make a query (N, A, C, T) to the decryption oracle, and $\mathcal{A}$ does not repeat a query to the decryption oracle.

In [27], McGrew and Viega analyzed the security of GCM, and there are differences between the above security notions. In [27], for privacy, the adversary has access to both the encryption and decryption oracles, while we chose to follow a more standard notion [10,30] where the privacy adversary has access to the encryption oracle only. Another difference is the assumption about the nonce reuse. In [27], the adversary is not allowed to reuse a nonce within decryption queries (but nonces used in encryption queries can be used in decryption queries and vice-versa), while our adversary can reuse nonces within decryption queries.

For the blockcipher $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$, we consider the PRP notion [24]. Let $\mathrm{Perm}(n)$ be the set of all permutations on $\{0,1\}^n$. We say that P is a random permutation if $P \xleftarrow{\$} \mathrm{Perm}(n)$. We define

$$\mathbf{Adv}_E^{\mathrm{prp}}(\mathcal{A}) \overset{\mathrm{def}}{=} \Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{E_K} \Rightarrow 1] - \Pr[P \xleftarrow{\$} \mathrm{Perm}(n) : \mathcal{A}^{P} \Rightarrow 1],$$

where the probabilities are defined over the randomness of $\mathcal{A}$, and the choices of K and P, respectively. We write $\mathrm{GCM}[\mathrm{Perm}(n), \tau]$ for GCM that uses a random permutation P as a blockcipher E_K, and we write the corresponding encryption and decryption algorithms as GCM-$\mathcal{E}_P$ and GCM-$\mathcal{D}_P$, respectively.

We also consider GCM that uses a random function as E_K, which is naturally defined as the invertibility of E_K is irrelevant in the definition of GCM. Let $\mathrm{Rand}(n)$ be the set of all functions from $\{0,1\}^n$ to $\{0,1\}^n$. We say that F is a random function if $F \xleftarrow{\$} \mathrm{Rand}(n)$, and write $\mathrm{GCM}[\mathrm{Rand}(n), \tau]$ for GCM that uses F as E_K. We write the corresponding encryption and decryption algorithms as GCM-$\mathcal{E}_F$ and GCM-$\mathcal{D}_F$, respectively.

5 Breaking GCM Security Proofs

5.1 Review of [27, Lemma 3], [27, Theorem 1], and [27, Theorem 2]

In this section, we first review a lemma in [27] that was used to derive the provable security results on GCM. Consider $\mathrm{GCM}[\mathrm{Rand}(n), \tau]$, GCM with E_K being a random function F, and the privacy notion for it. Let (N_1, A_1, M_1) and (N_2, A_2, M_2) be two encryption queries, where $N_1 \neq N_2$ and $|N_1|, |N_2| \neq 96$. Let $I_1[0] \leftarrow \mathsf{GHASH}_L(\varepsilon, N_1)$, and $I_1[j] \leftarrow \mathsf{inc}(I_1[j-1])$ for $1 \le j \le m_1$, where $m_1 = |M_1|_n$. Similarly, let $I_2[0] \leftarrow \mathsf{GHASH}_L(\varepsilon, N_2)$, and $I_2[j] \leftarrow \mathsf{inc}(I_2[j-1])$ for $1 \le j \le m_2$, where $m_2 = |M_2|_n$. If we have

$$\{I_1[0], I_1[1], \ldots, I_1[m_1]\} \cap \{I_2[0], I_2[1], \ldots, I_2[m_2]\} = \emptyset$$

and $I_i[j] \neq 0^n$ for $i = 1, 2$ and $0 \le j \le m_i$, then the two masks $S_1 = (S_1[1], \ldots, S_1[m_1])$ and $S_2 = (S_2[1], \ldots, S_2[m_2])$, produced from the counter mode encryption based on F, are uniformly distributed over $(\{0,1\}^n)^{m_1}$ and $(\{0,1\}^n)^{m_2}$, respectively. Furthermore, $F(I_1[0])$ and $F(I_2[0])$ are uniform random n-bit strings, and hence the probability distribution of strings returned from the encryption oracle is identical to that from the random-bits oracle.

Therefore, in order to prove the security of GCM, one has to show that the probability of a counter collision, $I_1[j_1] = I_2[j_2]$ for some $0 \le j_1 \le m_1$ and $0 \le j_2 \le m_2$, is small. We see that the event is equivalent to $\mathsf{inc}^{j_1}(\mathsf{GHASH}_L(\varepsilon, N_1)) = \mathsf{inc}^{j_2}(\mathsf{GHASH}_L(\varepsilon, N_2))$. Let $\mathsf{Coll}_L(r, N_1, N_2)$ denote the event

$$\mathsf{inc}^r(\mathsf{GHASH}_L(\varepsilon, N_1)) \oplus \mathsf{GHASH}_L(\varepsilon, N_2) = 0^n.$$

We need to bound $\Pr[L \xleftarrow{\$} \{0,1\}^n : \mathsf{Coll}_L(r, N_1, N_2)]$, where $r = j_1 - j_2$, which we write $\Pr_L[\mathsf{Coll}_L(r, N_1, N_2)]$ for simplicity. Since $-m_2 \le r \le m_1$ and $0 \le m_1, m_2 \le 2^{32} - 2$, the range of r is $-(2^{32} - 2) \le r \le 2^{32} - 2$.

In [27, Lemma 3], McGrew and Viega showed the following lemma (notation has been adapted to this paper).

Lemma 1 ([27]). *For any $-(2^{32}-2) \le r \le 2^{32}-2$, N_1, and N_2 such that $N_1 \ne N_2$, $|N_1|, |N_2| \ne 96$, and $|N_1|_n, |N_2|_n \le \ell_N$, $\Pr_L[\mathsf{Coll}_L(r, N_1, N_2)] \le (\ell_N + 1)/2^n$.*

Based on the lemma, [27, Theorem 1] states that the privacy advantage of GCM[Perm(n), τ], GCM with E_K being a random permutation P, is at most

$$\frac{0.5(\sigma/n + 2q)^2}{2^n} + \frac{2q(\sigma/n + 2q)\lceil \ell_N/n + 1\rceil}{2^n} + \frac{q\lceil \ell/n + 1\rceil}{2^\tau}, \tag{1}$$

and [27, Theorem 2] states that the authenticity advantage is at most

$$\frac{0.5(\sigma/n + 2q)^2}{2^n} + \frac{2q(\sigma/n + 2q + 1)\lceil \ell_N/n + 1\rceil}{2^n} + \frac{q\lceil \ell/n + 1\rceil}{2^\tau}, \tag{2}$$

where q is the maximum number of queries (either encryption or decryption queries), σ is the total length in bits of the plaintexts (either in encryption queries or returned from the decryption oracle), ℓ_N is the maximum length in bits of nonces in queries (either encryption or decryption queries), and ℓ is the maximum value of $|A_j| + |C_j|$, where A_j and C_j are the associated data and ciphertext, respectively, in the j-th query (either in decryption queries or returned from the encryption oracle). Note that the definitions of privacy and authenticity advantages are slightly different from ours, as explained in Sect. 4.

It is easy to see that the lemma is correct when $r = 0$: $\mathsf{Coll}_L(0, N_1, N_2)$ is the event $\mathsf{inc}^0(\mathsf{GHASH}_L(\varepsilon, N_1)) \oplus \mathsf{GHASH}_L(\varepsilon, N_2) = 0^n$, and we see that the left hand side is a non-trivial polynomial in L of degree at most $\ell_N + 1$ over GF(2^n), and hence there are at most $\ell_N + 1$ values of L that satisfy the equality.

However, for $r \ne 0$, it is not clear if $\mathsf{inc}^r(\mathsf{GHASH}_L(\varepsilon, N_1))$ is a polynomial of degree at most $\ell_N + 1$ even if this is the case for $\mathsf{GHASH}_L(\varepsilon, N_1)$, and the analysis for this case is missing in the proof of [27, Lemma 3]. The lemma is crucial in that it is used in the proofs for both privacy and authenticity.

5.2 Invalidating [27, Lemma 3]

Let $r = 1$, $N_1 = (\mathtt{0x0})^{17} \,\|\, \mathtt{0x2} = 0^{68} \,\|\, 0010$, and $N_2 = (\mathtt{0x0})^{17} \,\|\, \mathtt{0x6} = 0^{68} \,\|\, 0110$, where $|N_1| = |N_2| = 72$. Then $\mathsf{Coll}_L(r, N_1, N_2)$ is equivalent to

$$\mathsf{inc}^1(U_1 \cdot L^2 \oplus V \cdot L) \oplus (U_2 \cdot L^2 \oplus V \cdot L) = 0^n, \tag{3}$$

where $U_1 = (\mathtt{0x0})^{17} \,\|\, \mathtt{0x2} \,\|\, (\mathtt{0x0})^{14}$, $U_2 = (\mathtt{0x0})^{17} \,\|\, \mathtt{0x6} \,\|\, (\mathtt{0x0})^{14}$, and $V = (\mathtt{0x0})^{30} \,\|\, \mathtt{0x48}$. In binary, $U_1 = 0^{68} \,\|\, 0010 \,\|\, 0^{56}$, $U_2 = 0^{68} \,\|\, 0110 \,\|\, 0^{56}$, and $V = 0^{120} \,\|\, 01001000$. Now [27, Lemma 3] states that $\Pr_L[\mathsf{Coll}_L(r, N_1, N_2)] \le 2/2^n$, i.e., (3) has at most two solutions. However, one can verify (with the help of some software, e.g., [3]) that (3) has 32 solutions, which are listed in Appendix A. In other words, $\Pr_L[\mathsf{Coll}_L(r, N_1, N_2)] \geq 32/2^n$ holds, and hence [27, Lemma 3] cannot be valid.

We present one more observation regarding the counter-example. Consider

$$\mathsf{inc}^2(U_1 \cdot L^2 \oplus V \cdot L) \oplus (U_2 \cdot L^2 \oplus V \cdot L) = 0^n, \tag{4}$$
$$\mathsf{inc}^4(U_1 \cdot L^2 \oplus V \cdot L) \oplus (U_2 \cdot L^2 \oplus V \cdot L) = 0^n, \tag{5}$$

where the values of U_1, U_2, and V are as above. Then one can verify that (4) has 31 solutions, and that (5) has 30 solutions, which are also listed in Appendix A. The 93 values of L are all distinct, and are also different from 0^n, which is a solution for $\mathsf{inc}^0(U_1 \cdot L^2 \oplus V \cdot L) \oplus (U_2 \cdot L^2 \oplus V \cdot L) = 0^n$. Therefore we have

$$\Pr_L\left[\bigvee_{r=0,1,2,4} \mathsf{Coll}_L(r, N_1, N_2)\right] \geq \frac{94}{2^n}. \tag{6}$$

We remark that we exclude the case $r = 3$ since $\mathsf{Coll}_L(3, N_1, N_2)$ has no solution. In Appendix A, we present other examples of (N_1, N_2) that satisfy (6) and also invalidate [27, Lemma 3].

We next show that the above observation is not merely spotting of a subtle error in the proofs of GCM. The observation can actually be translated into a distinguishing attack.

5.3 Distinguishing Attack

Consider GCM with E_K being a random function F, and the privacy notion for it. We remark that the analysis of this idealized version of GCM is essential since the main part of the privacy proof is the analysis of this case. Let N_1 and N_2 be as in Sect. 5.2, $A_1 = \varepsilon$, $M_1 = 0^{5n}$, $A_2 = \varepsilon$, and $M_2 = 0^n$.

Let $\mathcal{A}$ be an adversary that has access to an oracle $\mathcal{O}$ which is either the GCM encryption oracle or the random-bits oracle. $\mathcal{A}$ works as follows.

1. First, $\mathcal{A}$ makes two queries, (N_i, A_i, M_i) for $i = 1, 2$, and obtains $(C_i, T_i) \leftarrow \mathcal{O}(N_i, A_i, M_i)$.
2. Let $(C_1[1], \ldots, C_1[5]) \overset{n}{\leftarrow} C_1$ and output 1 if

$$C_1[1] = C_2 \text{ or } C_1[2] = C_2 \text{ or } C_1[3] = C_2 \text{ or } C_1[5] = C_2. \tag{7}$$

First, suppose that $\mathcal{O}$ is the GCM encryption oracle. If $\mathsf{Coll}_L(0, N_1, N_2) \vee \mathsf{Coll}_L(1, N_1, N_2) \vee \mathsf{Coll}_L(2, N_1, N_2) \vee \mathsf{Coll}_L(4, N_1, N_2)$, then we see that $\mathcal{A}$ outputs 1. Otherwise the probability distributions of $C_1[1]$, $C_1[2]$, $C_1[3]$, $C_1[5]$, and C_2 are exactly the same as those of returned by the random-bits oracle. In particular, (7) is satisfied with the same probability for the GCM encryption oracle and for the random-bits oracle. Therefore, we have

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{GCM}[\mathrm{Rand}(n),\tau]}(\mathcal{A}) = \Pr[\mathcal{A}^{\mathrm{GCM}\text{-}\mathcal{E}_F} \Rightarrow 1] - \Pr[\mathcal{A}^{\$} \Rightarrow 1] \geq \frac{94}{2^n}. \tag{8}$$

Now using the notation of (1) and (2), our adversary has the following query complexity: $q = 2$, $\sigma = 6n$, $\ell_N = 72$, and $\ell = 5n$. Then (1) is $50/2^n + 80/2^n + 12/2^\tau = 130/2^n + 12/2^\tau$. Therefore, the attack does not contradict the claimed privacy bound (1).

However, (1) allows the use of the GCM decryption oracle, and rounding up the details makes it sufficiently large so that our attack is tolerated in appearance. Now if we take a closer look at (1), the second term, $80/2^n$, is the main part of the privacy proof that bounds $\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{GCM}[\mathrm{Rand}(n),\tau]}(\mathcal{A})$, while the first term is from the application of the PRP/PRF switching lemma [11] and the last term bounds the forgery probability due to the use of the decryption oracle. Therefore, the above attack does invalidate the main part of the privacy proof, and we also see that it invalidates the second term of (2), which is $88/2^n$.

We have already shown that the claimed bound on $\Pr_L[\mathsf{Coll}_L(r, N_1, N_2)]$ is invalid, which implies that the claimed security bounds, (1) and (2), are not justified. Furthermore, the above attack invalidates the main part of the privacy proof. Therefore, at this point, it is fair to conclude that the security of GCM is not supported by the proofs. We note that, although our attack does not work with 96-bit nonces, this statement holds even in this case since (1) and (2) cover a general case including the case that the nonce length is restricted to 96 bits.

5.4 Remarks

Our attack is efficient. It uses two oracle calls, and the lengths of the queries are short. However, the success probability, although large enough to invalidate the second terms in (1) and (2), is insignificantly small in practice and it has a limited practical implication. We also note that many standards require or recommend using GCM with 96-bit nonces, in which case the attack does not work. Indeed, in many RFCs, such as RFC 4106 (IPsec) [34], 5647 (SSH) [20], 5288 (SSL) [33], the nonce length is fixed to 96 bits, and RFC 5084 [19] and 5116 [25] recommend 96-bit nonces. For IEEE standards, some strictly require 96-bit nonces (e.g. IEEE 802.1AE [5]) and some do not (e.g. IEEE P1619.1 [6]). There are cases where non-96-bit nonces are allowed, including NIST SP 800-38D [15], ISO/IEC 19772 [7], PKCS #11 [4], and most software libraries (e.g., Gladman's code [17], CRYPTO++ [14], Java SE [2], and BouncyCastle [1]). Finally, NSA Suite B Cryptography includes GCM with a strong recommendation (but not mandatory; see e.g. [8]) of using it with 96-bit nonces.

We emphasize that, even when non-96-bit nonces are allowed, our attack has a limited practical implication. However, it does undermine the provable security of GCM, making its provable security open. A very natural question is then whether the security of GCM can ever be proved. In the following sections, we present an affirmative answer to this question.

6 Towards Repairing the Proofs

6.1 Combinatorial Problem

To prove the security of GCM, the first step is to derive the upper bound on $\Pr_L[\mathsf{Coll}_L(r, N_1, N_2)]$. The obvious bound is $\Pr_L[\mathsf{Coll}_L(r, N_1, N_2)] \le (\ell_N+1)/2^{n-32}$, which can be shown by ignoring the least significant 32 bits. In this section, we derive a better upper bound on $\Pr_L[\mathsf{Coll}_L(r, N_1, N_2)]$, and we first introduce a combinatorial problem for this goal.

For $0 \le r \le 2^{32}-1$, let

$$\mathbb{Y}_r \stackrel{\text{def}}{=} \left\{\mathsf{str}_{32}(\mathsf{int}(Y) + r \bmod 2^{32}) \oplus Y \mid Y \in \{0,1\}^{32}\right\}. \tag{9}$$

We also let $\alpha_r \stackrel{\text{def}}{=} \#\mathbb{Y}_r$ and $\alpha_{\max} \stackrel{\text{def}}{=} \max\{\alpha_r \mid 0 \le r \le 2^{32}-1\}$. For given r, it is not hard to experimentally derive the value of α_r by exhaustively evaluating $\mathsf{str}_{32}(\mathsf{int}(Y) + r \bmod 2^{32}) \oplus Y$ for all $Y \in \{0,1\}^{32}$. For example, we have

$$\alpha_0 = 1, \alpha_1 = 32, \alpha_2 = 31, \alpha_3 = 61, \alpha_4 = 30, \alpha_5 = 89, \ldots$$

and the problem is to identify $\alpha_{\max}$.

6.2 Relation to the Security of GCM

We show that identifying α_r gives the upper bound on $\Pr_L[\mathsf{Coll}_L(r, N_1, N_2)]$.

Lemma 2. *For any* $0 \le r \le 2^{32} - 1$, N_1, *and* N_2 *such that* $N_1 \ne N_2$, $|N_1|, |N_2| \ne 96$, *and* $|N_1|_n, |N_2|_n \le \ell_N$, $\Pr_L[\mathsf{Coll}_L(r, N_1, N_2)] \le \alpha_r(\ell_N + 1)/2^n$.

Proof. Let $Y_1, \ldots, Y_{\alpha_r} \in \{0,1\}^{32}$ be the α_r elements of $\mathbb{Y}_r$. Let $\mathcal{Y}_1, \ldots, \mathcal{Y}_{\alpha_r} \subseteq \{0,1\}^{32}$ be the α_r disjoint subsets of $\{0,1\}^{32}$ such that

$$\mathcal{Y}_j = \left\{ Y \in \{0,1\}^{32} \mid \mathsf{str}_{32}(\mathsf{int}(Y) + r \bmod 2^{32}) \oplus Y = Y_j \right\}$$

for $1 \le j \le \alpha_r$, $\mathcal{Y}_1 \cup \cdots \cup \mathcal{Y}_{\alpha_r} = \{0,1\}^{32}$, and $\mathcal{Y}_j \cap \mathcal{Y}_{j'} = \emptyset$ for $1 \le j < j' \le \alpha_r$. Observe that if $Y \in \mathcal{Y}_j$, then $\mathsf{str}_{32}(\mathsf{int}(Y) + r \bmod 2^{32})$ can be replaced with $Y \oplus Y_j$.

For $1 \le j \le \alpha_r$, let D_j be the event $\mathsf{Coll}_L(r, N_1, N_2) \wedge \mathsf{lsb}_{32}(\mathsf{GHASH}_L(\varepsilon, N_1)) \in \mathcal{Y}_j$. Since $\mathsf{D}_1, \ldots, \mathsf{D}_{\alpha_r}$ are disjoint events, we have

$$\Pr_L[\mathsf{Coll}_L(r, N_1, N_2)] = \sum_{1 \le j \le \alpha_r} \Pr_L[\mathsf{D}_j]. \tag{10}$$

Recall that $\mathsf{Coll}_L(r, N_1, N_2)$ is the event $\mathsf{inc}^r(\mathsf{GHASH}_L(\varepsilon, N_1)) \oplus \mathsf{GHASH}_L(\varepsilon, N_2) = 0^n$, and since $\mathsf{lsb}_{32}(\mathsf{GHASH}_L(\varepsilon, N_1)) \in \mathcal{Y}_j$, $\mathsf{inc}^r(\mathsf{GHASH}_L(\varepsilon, N_1))$ can be replaced with $\mathsf{GHASH}_L(\varepsilon, N_1) \oplus (0^{n-32} \,\|\, Y_j)$, implying that the event D_j is equivalent to

$$\mathsf{GHASH}_L(\varepsilon, N_1) \oplus \mathsf{GHASH}_L(\varepsilon, N_2) \oplus (0^{n-32} \,\|\, Y_j) = 0^n \tag{11}$$

and $\mathsf{lsb}_{32}(\mathsf{GHASH}_L(\varepsilon, N_1)) \in \mathcal{Y}_j$. We see that (11) is a non-trivial equation in L of degree at most $\ell_N + 1$ over $\mathrm{GF}(2^n)$, and hence it has at most $\ell_N + 1$ solutions. From (10), we obtain the lemma. □

6.3 Deriving α_r and $\alpha_{\max}$

The problem introduced in Sect. 6.1 can be solved by exhaustively evaluating (9) for all $0 \le r \le 2^{32} - 1$, which is computationally costly. Another possible approach is to follow the framework in [29] (or to use tools in [22]).

Instead of following these approaches, in this section, we present a recursive formula to efficiently compute α_r. Let $r \ge 0$ be a given integer, ℓ be the number of runs of ones in the binary representation of r (e.g., if r is 00110101 in binary, then $\ell = 3$), and v_ℓ be an integer with $r \le 2^{v_\ell} - 1$. Suppose $\mathsf{str}_{v_\ell}(r) = 0^{s_\ell} 1^{t_\ell} \ldots 0^{s_1} 1^{t_1} 0^{s_0}$, where $s_{\ell-1}, \ldots, s_1 \ge 1$, $s_\ell, s_0 \ge 0$, $t_\ell, \ldots, t_1 \ge 1$, and $v_\ell = (s_\ell + \cdots + s_1 + s_0) + (t_\ell + \cdots + t_1)$.

Define

$$A_\ell \overset{\text{def}}{=} \# \left\{ \mathsf{str}_{v_\ell}(\mathsf{int}(Y) + r \bmod 2^{v_\ell}) \oplus Y \mid Y \in \{0,1\}^{v_\ell} \right\}.$$

Note that, when $v_\ell = 32$, α_r is equal to A_ℓ. The next proposition gives an efficient recursive formula to compute A_ℓ.

Proposition 1. *For any* $\ell \ge 1$,

$$A_\ell = \begin{cases} t_\ell A_{\ell-1} + B_{\ell-1} & \text{if } s_\ell = 0, \\ s_\ell B_\ell + A_{\ell-1} & \text{if } s_\ell \ge 1, \end{cases} \tag{12}$$

where $B_j = t_j A_{j-1} + B_{j-1}$ *for* $1 \le j \le \ell$, $A_j = s_j B_j + A_{j-1}$ *for* $1 \le j \le \ell - 1$, $A_0 = 1$, *and* $B_0 = 0$.

Table 1. List of (r, α_r) (left) and the relation between $r_{\max}$ and $\beta(r_{\max})$ (right)

r	α_r
0x00000001	32
0x00000003	61
0x00000005	89
0x0000000b	143
0x00000025	294
0x00000055	538
0x0000012b	1115
0x00000455	2113
0x00000955	4124
0x000024ab	8579
0x00005555	17389
0x00012aab	34702
0x00049555	69742
0x000aaaab	138117
0x00255555	262471
0x00955555	559000
0x02555555	1127959
0x0a555555	2116814

Range of $r_{\max}$	$\beta(r_{\max})$
0x00000001–0x00000002	2^5
0x00000003–0x00000004	2^6
0x00000005–0x0000000a	2^7
0x0000000b–0x00000024	2^8
0x00000025–0x00000054	2^9
0x00000055–0x0000012a	2^{10}
0x0000012b–0x00000454	2^{11}
0x00000455–0x00000954	2^{12}
0x00000955–0x000024aa	2^{13}
0x000024ab–0x00005554	2^{14}
0x00005555–0x00012aaa	2^{15}
0x00012aab–0x00049554	2^{16}
0x00049555–0x000aaaaa	2^{17}
0x000aaaab–0x00255554	2^{18}
0x00255555–0x00955554	2^{19}
0x00955555–0x02555554	2^{20}
0x02555555–0x0a555554	2^{21}
0x0a555555–0xffffffff	2^{22}

In Appendix B, we present an elementary proof of the proposition. Given Proposition 1, it is not hard to experimentally derive $\alpha_{\max}$ by directly evaluating (12). We have $\alpha_{\max} = 3524578$, where $\alpha_r = \alpha_{\max}$ holds when $r =$ 0x2aaaaaab, 0xaaaaaaab, 0x55555555, and 0xd5555555.

From Lemma 2 and since $3524578 \leq 2^{22}$, we obtain the following corollary.

Corollary 1. *For any* $0 \leq r \leq 2^{32} - 1$*,* N_1*, and* N_2 *such that* $N_1 \neq N_2$*,* $|N_1|, |N_2| \neq 96$*, and* $|N_1|_n, |N_2|_n \leq \ell_N$*,* $\Pr_L[\mathsf{Coll}_L(r, N_1, N_2)] \leq 2^{22}(\ell_N + 1)/2^n$*.*

If the upper bound of r is smaller than $2^{32} - 1$, then depending on the value, the constant, 2^{22}, in Corollary 1 can be replaced by a smaller constant. Specifically, there are 303 values of $1 \leq r \leq 2^{32} - 1$ that satisfy $\max\{\alpha_0, \ldots, \alpha_{r-1}\} < \alpha_r$, and a list of the 303 values of (r, α_r) can be used to obtain a smaller constant. The list can be found in Appendix C. Table 1 (left) is the excerpt from the list for better readability and usability. For each $i = 5, 6, \ldots, 22$, Table 1 (left) lists the minimum value of r, among the 303 values, such that $2^{i-1} < \alpha_r \leq 2^i$.

Table 1 (right) is obtained from Table 1 (left) and shows the relation between the range of $r_{\max}$ and the corresponding constant $\beta(r_{\max})$, where $r_{\max}$ is the upper bound of r and $\beta(r_{\max})$ is the upper bound of α_r. For example, if $r_{\max} = 2^{10} =$ 0x400, then it is within the range of 0x0000012b–0x00000454 and hence $\beta(r_{\max}) = 2^{11}$. See also Fig. 4 for a graph showing the relation between $r_{\max}$ and $\beta(r_{\max})$.

We have the following corollary.

Corollary 2. *For any* $0 \leq r \leq r_{\max}$*,* N_1*, and* N_2 *such that* $N_1 \neq N_2$*,* $|N_1|, |N_2| \neq 96$*, and* $|N_1|_n, |N_2|_n \leq \ell_N$*,* $\Pr_L[\mathsf{Coll}_L(r, N_1, N_2)] \leq \beta(r_{\max})(\ell_N + 1)/2^n$*.*

We note that for $r_{\max} = 0$, from $\alpha_0 = 1$, $\beta(r_{\max})$ is defined to be 1.

In Appendix C, we discuss that a more precise value of $\beta(r_{\max})$ can be obtained with the complete list of the 303 values of (r, α_r).

7 Repairing GCM Security Proofs

In this section, basing on the results of the previous section, we present new security bounds on GCM. We also present overviews of the proofs.

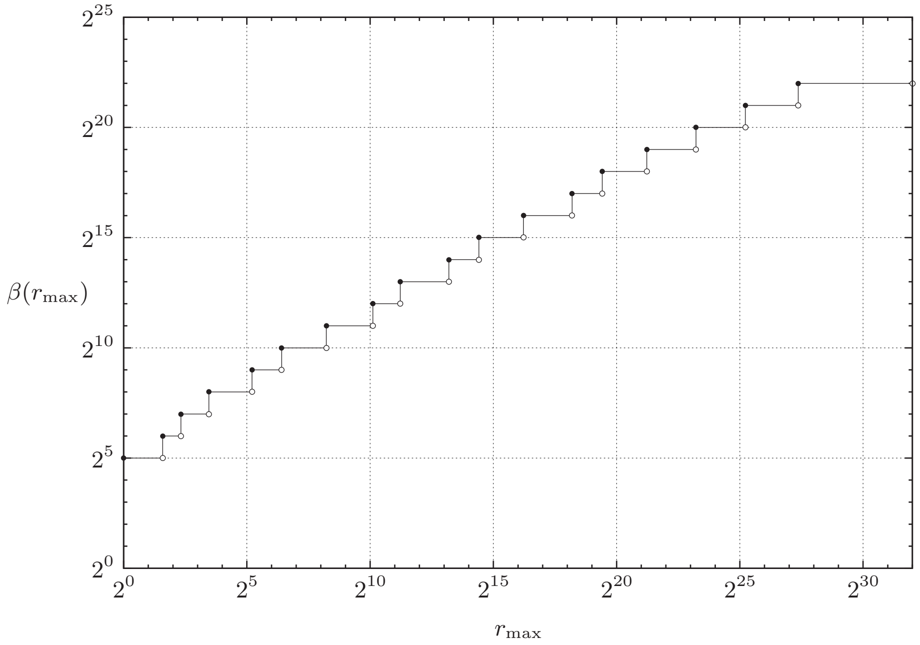

Fig. 4. Relation between $r_{\max}$ and $\beta(r_{\max})$

7.1 Privacy Result

In the privacy result, if $\mathcal{A}$ makes q queries $(N_1, A_1, M_1), \ldots, (N_q, A_q, M_q)$, then the total plaintext length is $m_1 + \cdots + m_q$, and the maximum nonce length is $\max\{n_1, \ldots, n_q\}$, where $|N_i|_n = n_i$ and $|M_i|_n = m_i$. We have the following information theoretic result.

Theorem 1. *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,*

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

Observe that the security bound is essentially the same as the original one (1), except that we have a constant, 2^{22}, in the second term, and we do not have a term that corresponds to the forgery probability.

We next present a corollary showing that GCM has a better security bound if the nonce length is restricted to 96 bits.

Corollary 3. *Assume that the nonce length is restricted to 96 bits. Then,*

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

Let E be a blockcipher secure in the sense of the PRP notion. Then the corresponding complexity theoretic results, where E is used in place of $\mathrm{Perm}(n)$, can be obtained by a standard argument (see e.g. [9]).

In the next section, we present an intuitive proof overview of Theorem 1. A complete proof is presented in Appendix D. A proof of Corollary 3 is obtained by modifying the proof of Theorem 1, and is omitted.

7.2 Proof Overview of Theorem 1

Suppose that $\mathcal{A}$ has access to the GCM encryption oracle. We first replace the random permutation P by a random function F. The difference between the two advantage functions is at most $(\sigma+q+1)^2/2^{n+1}$ from the PRP/PRF switching lemma [11]. Next, suppose that $\mathcal{A}$ has made $i-1$ queries $(N_1, A_1, M_1), \ldots, (N_{i-1}, A_{i-1}, M_{i-1})$, obtained $(C_1, T_1), \ldots, (C_{i-1}, T_{i-1})$, and is now making the i-th query (N_i, A_i, M_i). For $1 \le j \le i$, let $\mathcal{I}_j = \{I_j[0], I_j[1], \ldots, I_j[m_j]\}$ be a set of n-bit strings used as the inputs of F during the computation of (C_j, T_j) for the query (N_j, A_j, M_j) (other than 0^n used to generate L). Observe that, if $I_i[0], I_i[1], \ldots, I_i[m_i]$ are not previously used, then (C_i, T_i) is a random string of $|M_i| + \tau$ bits. That is, unless

$$\mathcal{I}_i \cap (\{0^n\} \cup \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_{i-1}) \neq \emptyset \tag{13}$$

holds for some $1 \le i \le q$, the distribution of the output of the GCM encryption oracle (based on the random function F) is identical to that of the random-bits oracle, and hence $\mathcal{A}$ is unable to distinguish between the two oracles. It can be shown that the probability of (13) is at most $2^{22} q(\sigma+q)(\ell_N+1)/2^n$ by using Corollary 1. The result is obtained by summing up the two above-mentioned probabilities.

7.3 Authenticity Result

If $\mathcal{A}$ makes q encryption queries $(N_1, A_1, M_1), \ldots, (N_q, A_q, M_q)$ and q' decryption queries $(N'_1, A'_1, C'_1, T'_1), \ldots, (N'_{q'}, A'_{q'}, C'_{q'}, T'_{q'})$, 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 $\max\{a_1 + m_1, \ldots, a_q + m_q, a'_1 + m'_1, \ldots, a'_{q'} + m'_{q'}\}$, 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 $|C'_i|_n = m'_i$. We have the following information theoretic result.

Theorem 2. *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,*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{GCM}[\mathrm{Perm}(n),\tau]}(\mathcal{A}) \le{}& \frac{0.5(\sigma+q+q'+1)^2}{2^n} \\ &+ \frac{2^{22}(q+q'+1)(\sigma+q)(\ell_N+1)}{2^n} + \frac{q'(\ell_A+1)}{2^\tau}.\end{aligned} \tag{14}$$

As in the privacy result, the bound is essentially the same as the original one (2), except that we have a constant, 2^{22}, in the second term. The next corollary shows that we have a better security bound if the nonce length is restricted to 96 bits.

Corollary 4. *Assume that the nonce length is restricted to 96 bits. Then,*

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{GCM}[\mathrm{Perm}(n),\tau]}(\mathcal{A}) \le \frac{0.5(\sigma+q+q'+1)^2}{2^n} + \frac{q'(\ell_A+1)}{2^\tau}. \tag{15}$$

The corresponding complexity theoretic results can be obtained based on the PRP notion of a blockcipher E by a standard argument (see e.g. [9]).

We present an intuitive proof overview of Theorem 2 in the next section, and a complete proof is presented in Appendix E. A proof of Corollary 4 can be obtained from the proof of Theorem 2, and is omitted.

7.4 Proof Overview of Theorem 2

We replace the random permutation P by a random function F. From the PRP/PRF switching lemma [11], we have a term $(\sigma+q+q'+1)^2/2^{n+1}$. We then consider the probability of a counter collision as in the privacy proof, but this time, we consider the counter values used for decryption queries as well. The probability can be shown to be at most $2^{22}(q+q'+1)(\sigma+q)(\ell_N+1)/2^n$ by using Corollary 1. Under the condition that there is no counter collision, the adversary is essentially asked to forge a message authentication code $(N,A,C) \rightarrow F(g(N)) \oplus \mathsf{GHASH}_L(A,C)$, where $g(N) = N \,\|\, 0^{31}1$ if $|N| = 96$, and $g(N) = \mathsf{GHASH}_L(\varepsilon, N)$ if $|N| \neq 96$. The probability can be shown to be at most $q'(\ell_A+1)/2^\tau$, and we obtain the theorem by summing up the three probabilities.

7.5 Better Security Bounds

Use of Corollary 2. Suppose that, either in privacy or authenticity notions, $\mathcal{A}$ makes q encryption queries $(N_1,A_1,M_1),\ldots,(N_q,A_q,M_q)$. Let ℓ_M be the maximum plaintext length, which is $\max\{m_1,\ldots,m_q\}$, where $|M_i|_n = m_i$. Theorem 1 and Theorem 2 assume $\ell_M = 2^{32}-2$ and use Corollary 1 to obtain the results. However, if ℓ_M is known to be smaller, then Corollary 2 can be used to obtain better bounds. Specifically, in Theorem 1 and Theorem 2, if $\ell_M \le r_{\max}$, then the constant becomes $\beta(r_{\max})$ instead of 2^{22}. For example, if ℓ_M is 2^{10}, then from Table 1 and by following the argument in Sect. 6.3, the constant becomes 2^{11}.

Use of [12, Theorem 2.3]. Using Bernstein's result [12, Theorem 2.3], we can further improve the authenticity bound (but not the privacy bound). For positive integer a, let $\delta_n(a) = (1-(a-1)/2^n)^{-a/2}$. With the same notation as in Theorem 2, the right hand side of (14) can be improved to the following bound.

$$\left[\frac{2^{22}(q+q'+1)(\sigma+q)(\ell_N+1)}{2^n} + \frac{q'(\ell_A+1)}{2^\tau}\right] \cdot \delta_n(\sigma+q+q'+1)$$

Note that when $a \ll 2^n$ we have $\delta_n(a) \approx (1+a^2/2^{n+1})$. It was shown that $\delta_n(a) \le 1.7$ when $a \le 2^{64}$ and $n \ge 128$ [13]. Hence, for example, if $1 < q' \le q \le \sigma$, $n = \tau = 128$, and $\sigma+q+q' < 2^{64}$, we obtain the bound $(17 \cdot 2^{22} q\sigma\ell_N + 4q'\ell_A)/2^{128}$.

See Appendix C for more discussions on the better security bounds.

8 Conclusion

In this paper, we studied the security proofs of GCM. We first pointed out that the proofs contain a flaw, and translated the observation into a distinguishing attack that invalidates the main part of the privacy proof. We then showed that the proofs can be repaired by presenting new privacy and authenticity bounds. The bounds are larger than what were previously claimed in a general case of variable length nonces, but they are smaller when the nonce length is restricted to 96 bits. Many standards require or recommend using GCM with 96-bit nonces for efficiency reasons. Our results suggest that restricting GCM to 96-bit nonces is recommended from the provable security perspective as well. This follows [31], where GCM with 96-bit nonces is recommended as the use of variable length nonces increases the proof complexity and the proofs are infrequently verified.

We remark that since our attack only invalidates the second terms of (1) and (2), it does not exclude a possibility that the original security bounds, (1) and (2), can still be proved, and it would be interesting to see if our security bounds can be improved.

Acknowledgments. The authors would like to thank Masato Aikawa for discussions at the initial phase of this work, Yasushi Osaki and Xiangyu Quan for helping searching the counter-examples given in Sect. 5.1 and in Appendix A, Nicky Mouha for verifying the values of r that give $\alpha_r = \alpha_{\max}$ presented in Sect. 6.3, and participants of Dagstuhl Seminar 12031, Symmetric Cryptography, and the anonymous CRYPTO 2012 reviewers for helpful comments. The work by Tetsu Iwata was supported in part by MEXT KAKENHI, Grant-in-Aid for Young Scientists (A), 22680001.

References

1. Bouncy Castle, `http://www.bouncycastle.org/` (accessed on May 26, 2012)
2. Java Platform, Standard Edition 7, `http://docs.oracle.com/javase/7/docs/` (accessed on May 26, 2012)
3. Risa/Asir, `http://www.math.kobe-u.ac.jp/Asir/asir.html` (accessed on May 26, 2012)
4. PKCS #11 v2.20: Cryptographic Token Interface Standard. PKCS #11 v2.20 (2004), `http://www.rsa.com/rsalabs/node.asp?id=2133` (accessed on May 31, 2012)
5. IEEE Standard for Local and Metropolitan Area Networks Media Access Control (MAC) Security. IEEE Std 802.1AE-2006 (2006)
6. IEEE Standard for Authenticated Encryption with Length Expansion for Storage Devices. IEEE Std 1619.1-2007 (2007)
7. Information Technology — Security Techniques — Authenticated Encryption, ISO/IEC 19772:2009. International Standard ISO/IEC 19772 (2009)
8. 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`
9. Bellare, M., Kilian, J., Rogaway, P.: The Security of the Cipher Block Chaining Message Authentication Code. J. Comput. Syst. Sci. 61(3), 362–399 (2000)
10. Bellare, M., Namprempre, C.: Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In: Okamoto, T. (ed.) ASIACRYPT. Lecture Notes in Computer Science, vol. 1976, pp. 531–545. Springer (2000)
11. Bellare, M., Rogaway, P.: The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In: Vaudenay, S. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 4004, pp. 409–426. Springer (2006)
12. Bernstein, D.J.: Stronger Security Bounds for Permutations (2005), `http://cr.yp.to/papers.html` (accessed on May 31, 2012)
13. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC Security Bound from PRP-Advantage (2005), `http://fastcrypto.org/umac/umac_security.pdf` (accessed on May 31, 2012)
14. Dai, W.: Crypto++ Library, `http://www.cryptopp.com/` (accessed on May 26, 2012)
15. Dworkin, M.: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D (2007)
16. Ferguson, N.: Authentication Weaknesses in GCM. Public Comments to NIST (2005), `http://csrc.nist.gov/groups/ST/toolkit/BCM/comments.html`
17. Gladman, B.: `http://www.gladman.me.uk/` (accessed on May 26, 2012)
18. Handschuh, H., Preneel, B.: Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms. In: Wagner, D. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 5157, pp. 144–161. Springer (2008)
19. Housley, R.: Using AES-CCM and AES-GCM Authenticated Encryption in the Cryptographic Message Syntax (CMS). IETF RFC 5084 (2007)
20. Igoe, K.M., Solinas, J.A.: AES Galois Counter Mode for the Secure Shell Transport Layer Protocol. IETF RFC 5647 (2009)
21. Joux, A.: Authentication Failures in NIST version of GCM. Public Comments to NIST (2006), `http://csrc.nist.gov/groups/ST/toolkit/BCM/comments.html`
22. Leurent, G.: ARXtools: A Toolkit for ARX Analysis. In: The Third SHA-3 Candidate Conference (2012), `http://csrc.nist.gov/groups/ST/hash/sha-3/Round3/March2012/index.html`
23. Leurent, G., Thomsen, S.S.: Practical Near-Collisions on the Compression Function of BMW. In: Joux, A. (ed.) FSE. Lecture Notes in Computer Science, vol. 6733, pp. 238–251. Springer (2011)
24. Luby, M., Rackoff, C.: How to Construct Pseudorandom Permutations from Pseudorandom Functions. SIAM J. Comput. 17(2), 373–386 (1988)
25. McGrew, D.A.: An Interface and Algorithms for Authenticated Encryption. IETF RFC 5116 (2008)
26. McGrew, D.A., Viega, J.: The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT. Lecture Notes in Computer Science, vol. 3348, pp. 343–355. Springer (2004)

Table 2. List of solutions of (3)

`0x7f6db6d2db6db6db6db6db6492492492`	`0x7f6db6dadb6db6db6db6db6492492492`
`0x81b6db776db6db6db6db6dadb6db6db6`	`0x81b6db676db6db6db6db6dadb6db6db6`
`0xbe00003c0000000000000003ffffffff`	`0xbe00001c0000000000000003ffffffff`
`0xc16db6aadb6db6db6db6db1b6db6db6d`	`0xc16db6eadb6db6db6db6db1b6db6db6d`
`0x3fb6db876db6db6db6db6d5249249249`	`0x3fb6db076db6db6db6db6d5249249249`
`0x000001dc000000000000001c00000000`	`0x000000dc000000000000001c00000000`
`0x7f6db56adb6db6db6db6d8e492492492`	`0x7f6db76adb6db6db6db6d8e492492492`
`0x81b6dc076db6db6db6db6aadb6db6db6`	`0x81b6d8076db6db6db6db6aadb6db6db6`
`0xbe000edc00000000000000e3ffffffff`	`0xbe0006dc00000000000000e3ffffffff`
`0xc16dab6adb6db6db6db6c71b6db6db6d`	`0xc16dbb6adb6db6db6db6c71b6db6db6d`
`0x3fb6e0076db6db6db6db555249249249`	`0x3fb6c0076db6db6db6db555249249249`
`0x000076dc000000000000071c00000000`	`0x000036dc000000000000071c00000000`
`0x7f6d5b6adb6db6db6db638e492492492`	`0x7f6ddb6adb6db6db6db638e492492492`
`0x81b700076db6db6db6daaaadb6db6db6`	`0x81b600076db6db6db6daaaadb6db6db6`
`0xbe03b6dc00000000000038e3ffffffff`	`0xbe01b6dc00000000000038e3ffffffff`
`0xc16adb6adb6db6db6db1c71b6db6db6d`	`0x00000004000000000000000000000000`

27. 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/`

28. McGrew, D.A., Viega, J.: The Galois/Counter Mode of Operation (GCM). Submission to NIST (2005), `http://csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html`

29. Mouha, N., Velichkov, V., De Cannière, C., Preneel, B.: The Differential Analysis of S-Functions. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 6544, pp. 36–56. Springer (2011)

30. Rogaway, P.: Authenticated-Encryption with Associated-Data. In: Atluri, V. (ed.) ACM Conference on Computer and Communications Security. pp. 98–107. ACM (2002)

31. Rogaway, P.: Evaluation of Some Blockcipher Modes of Operation. Investigation Reports on Cryptographic Techniques in FY 2010 (2011), `http://www.cryptrec.go.jp/english/` (accessed on May 31, 2012)

32. Saarinen, M.J.O.: Cycling Attacks on GCM, GHASH and Other Polynomial MACs and Hashes. Preproceedings of FSE 2012 (2012), `http://fse2012.inria.fr/` (accessed on March 17, 2012)

33. Salowey, J., Choudhury, A., McGrew, D.A.: AES Galois Counter Mode (GCM) Cipher Suites for TLS. IETF RFC 5288 (2008)

34. Viega, J., McGrew, D.A.: The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP). IETF RFC 4106 (2005)

A Solutions of (3), (4), and (5), and Examples of (N_1, N_2) Satisfying (6)

In Table 2, Table 3, and Table 4, we show a list of values of L that satisfy (3), (4), and (5), respectively. We see that the 93 values from these lists are all distinct, and are different from 0^n.

The counter-example presented in Sect. 5.2 was found by experimentally searching over the values of U_1, U_2, and V. We started by searching over random U_1, U_2, and V, and found that the values of the form $U_1 = 0^{8i} \,\|\, X \,\|\, 0^{n-8-8i}$, $U_2 = 0^{8i} \,\|\, Y \,\|\, 0^{n-8-8i}$, and $V \in \{0,1\}^n$ have many examples that satisfy (6), where $X, Y \in \{0,1\}^8$, $0 \le i \le 15$, and $\mathsf{int}(V) = 8j$ for some $i + 1 \le j \le 16$ but $j \ne 12$. The examples include the following values:

$$(U_1, U_2, V) = \begin{cases} ((\mathtt{0x0})^{15} \,\|\, \mathtt{0x2} \,\|\, (\mathtt{0x0})^{16}, (\mathtt{0x0})^{15} \,\|\, \mathtt{0x6} \,\|\, (\mathtt{0x0})^{16}, (\mathtt{0x0})^{30} \,\|\, \mathtt{0x70}) \\ ((\mathtt{0x0})^{17} \,\|\, \mathtt{0x2} \,\|\, (\mathtt{0x0})^{14}, (\mathtt{0x0})^{17} \,\|\, \mathtt{0x6} \,\|\, (\mathtt{0x0})^{14}, (\mathtt{0x0})^{30} \,\|\, \mathtt{0x70}) \\ ((\mathtt{0x0})^{17} \,\|\, \mathtt{0x4} \,\|\, (\mathtt{0x0})^{14}, (\mathtt{0x0})^{17} \,\|\, \mathtt{0xc} \,\|\, (\mathtt{0x0})^{14}, (\mathtt{0x0})^{30} \,\|\, \mathtt{0x48}) \end{cases}$$

These values are equivalent to $(0^{60} \,\|\, 0010 \,\|\, 0^{64}, 0^{60} \,\|\, 0110 \,\|\, 0^{64}, 0^{120} \,\|\, 01110000)$, $(0^{68} \,\|\, 0010 \,\|\, 0^{56}, 0^{68} \,\|\, 0110 \,\|\, 0^{56}, 0^{120} \,\|\, 01110000)$, and $(0^{68} \,\|\, 0100 \,\|\, 0^{56}, 0^{68} \,\|\, 1100 \,\|\, 0^{56}, 0^{120} \,\|\, 01001000)$ in binary. N_1 and N_2 are the most significant $\mathsf{int}(V)$ bits of U_1 and U_2, respectively.

Table 3. List of solutions of (4)

0x7f6db6d6db6db6db6db6db6492492492	0x7f6db6dedb6db6db6db6db6492492492
0x81b6db736db6db6db6db6dadb6db6db6	0x81b6db636db6db6db6db6dadb6db6db6
0xbe000038000000000000003fffffffff	0xbe000018000000000000003fffffffff
0xc16db6aedb6db6db6db6db1b6db6db6d	0xc16db6eedb6db6db6db6db1b6db6db6d
0x3fb6db836db6db6db6db6d5249249249	0x3fb6db036db6db6db6db6d5249249249
0x000001d800000000000001c000000000	0x000000d800000000000001c000000000
0x7f6db56edb6db6db6db6d8e492492492	0x7f6db76edb6db6db6db6d8e492492492
0x81b6dc036db6db6db6db6aadb6db6db6	0x81b6d8036db6db6db6db6aadb6db6db6
0xbe000ed80000000000000e3fffffffff	0xbe0006d80000000000000e3fffffffff
0xc16dab6edb6db6db6db6c71b6db6db6d	0xc16dbb6edb6db6db6db6c71b6db6db6d
0x3fb6e0036db6db6db6db555249249249	0x3fb6c0036db6db6db6db555249249249
0x000076d800000000000071c000000000	0x000036d800000000000071c000000000
0x7f6d5b6edb6db6db6db638e492492492	0x7f6ddb6edb6db6db6db638e492492492
0x81b700036db6db6db6daaaadb6db6db6	0x81b600036db6db6db6daaaadb6db6db6
0xbe03b6d80000000000038e3fffffffff	0xbe01b6d80000000000038e3fffffffff
0xc16adb6edb6db6db6db1c71b6db6db6d	

Table 4. List of solutions of (5)

0xbe076db80000000000071c7fffffffff	0xc16c000edb6db6db6db5555b6db6db6d
0xc16e000edb6db6db6db5555b6db6db6d	0xfedbb6d5b6db6db6db6c71c924924924
0xfedab6d5b6db6db6db6c71c924924924	0x00006db8000000000000e38000000000
0x0000edb8000000000000e38000000000	0x7f6d800edb6db6db6db6aaa492492492
0x7f6dc00edb6db6db6db6aaa492492492	0x40db76d5b6db6db6db6d8e36db6db6db
0x40db56d5b6db6db6db6d8e36db6db6db	0xbe000db80000000000001c7fffffffff
0xbe001db80000000000001c7fffffffff	0xc16db00edb6db6db6db6d55b6db6db6d
0xc16db80edb6db6db6db6d55b6db6db6d	0xfedb6ed5b6db6db6db6db1c924924924
0xfedb6ad5b6db6db6db6db1c924924924	0x000001b8000000000000038000000000
0x000003b8000000000000038000000000	0x7f6db60edb6db6db6db6daa492492492
0x7f6db70edb6db6db6db6daa492492492	0x40db6dd5b6db6db6db6db636db6db6db
0x40db6d55b6db6db6db6db636db6db6db	0xbe000038000000000000007fffffffff
0xbe000078000000000000007fffffffff	0xc16db6cedb6db6db6db6db5b6db6db6d
0xc16db6eedb6db6db6db6db5b6db6db6d	0xfedb6db5b6db6db6db6db6c924924924
0xfedb6da5b6db6db6db6db6c924924924	0x00000008000000000000000000000000

B Proof of Proposition 1

With the same notation in Sect. 6, let $u_\ell = (s_\ell + \cdots + s_1) + (t_\ell + \cdots + t_1)$, and r' be an integer satisfying $\mathsf{str}_{u_\ell}(r') = 0^{s_\ell}1^{t_\ell} \ldots 0^{s_1}1^{t_1}$, i.e., r' is the integer defined by taking the most significant u_ℓ bits of r. Then we see that $A_\ell = \#\{\mathsf{str}_{u_\ell}(\mathsf{int}(Y) + r' \bmod 2^{u_\ell}) \oplus Y \mid Y \in \{0,1\}^{u_\ell}\}$ since the least significant s_0 bits of r does not affect A_ℓ. In what follows, without loss of generality, we make the assumption that $s_0 = 0$ (and hence $u_\ell = v_\ell$).

We first show the following lemma.

Lemma 3. *If $\ell = 1$, $s_1 = 0$, and $\mathsf{str}_{v_1}(r) = 1^{t_1}$, then $A_1 = t_1$.*

Proof. If $Y = 0^{t_1}$, then $\mathsf{str}_{t_1}(\mathsf{int}(Y) + r \bmod 2^{t_1}) \oplus Y = 1^{t_1}$. If $Y = (Y' \,\|\, 10^{j-1})$ for some $1 \le j \le t_1$ and $Y' \in \{0,1\}^{t_1-j}$, then $\mathsf{str}_{t_1}(\mathsf{int}(Y) + r \bmod 2^{t_1}) \oplus Y = 0^{t_1-j}1^j$. Therefore, $\mathsf{str}_{t_1}(\mathsf{int}(Y) + r \bmod 2^{t_1}) \oplus Y$ takes t_1 values when Y ranges over $Y \in \{0,1\}^{t_1}$. □

By a similar argument, we obtain the following lemma.

Lemma 4. *If $\mathsf{str}_{j+t_1}(r) = (r' \,\|\, 1^{t_1})$ for some $j \ge 1$ and $r' \in \{0,1\}^j$, then for any fixed $Y' \in \{0,1\}^j$, $\mathsf{str}_{j+t_1}(\mathsf{int}(Y) + r \bmod 2^{j+t_1}) \oplus Y$ takes t_1 values when Y ranges over $Y \in \{0,1\}^{j+t_1}$, where $Y = (Y' \,\|\, Y'')$, $Y'' \in \{0,1\}^{t_1}$, and $\mathsf{int}(0^{t_1-1}1) \le \mathsf{int}(Y'') \le \mathsf{int}(1^{t_1})$.*

Proof. We see that the least significant t_1 bits of $\mathsf{str}_{j+t_1}(\mathsf{int}(Y) + r \bmod 2^{j+t_1}) \oplus Y$ is of the form $\mathsf{str}_{t_1}(\mathsf{int}(Y'') + \mathsf{int}(1^{t_1}) \bmod 2^{t_1}) \oplus Y''$. It takes t_1 values when Y'' ranges over $\mathsf{int}(0^{t_1-1}1) \leq \mathsf{int}(Y'') \leq \mathsf{int}(1^{t_1})$, while the remaining bits are fixed to a constant. □

We next show the following lemma.

Lemma 5. *If $\ell = 1$, $s_1 \geq 1$, and $\mathsf{str}_{v_1}(r) = 0^{s_1}1^{t_1}$, then $A_1 = s_1t_1 + 1$.*

Proof. If $Y = (Y' \,\|\, 0^{t_1})$ for some $Y' \in \{0,1\}^{s_1}$, then $\mathsf{str}_{v_1}(\mathsf{int}(Y) + r \bmod 2^{v_1}) \oplus Y = 0^{s_1}1^{t_1}$.

If $Y = (Y' \,\|\, 01^{j-1} \,\|\, Y'')$ for some $1 \leq j \leq s_1$, $Y' \in \{0,1\}^{s_1-j}$, and $Y'' \in \{0,1\}^{t_1}$ such that $\mathsf{int}(0^{t_1-1}1) \leq \mathsf{int}(Y'') \leq \mathsf{int}(1^{t_1})$, then $\mathsf{str}_{v_1}(\mathsf{int}(Y) + r \bmod 2^{v_1}) \oplus Y$ can be written as

$$0^{s_1-j}1^j \,\|\, \mathsf{str}_{t_1}(\mathsf{int}(Y'') + \mathsf{int}(1^{t_1}) \bmod 2^{t_1}) \oplus Y''. \tag{16}$$

From Lemma 4, (16) takes s_1t_1 different values when j and Y'' ranges over $1 \leq j \leq s_1$ and $\mathsf{int}(0^{t_1-1}1) \leq \mathsf{int}(Y'') \leq \mathsf{int}(1^{t_1})$. Furthermore, the s_1t_1 values are different from $0^{s_1}1^{t_1}$.

If $Y = (1^{s_1} \,\|\, Y'')$ for some $Y'' \in \{0,1\}^{t_1}$ such that $\mathsf{int}(0^{t_1-1}1) \leq \mathsf{int}(Y'') \leq \mathsf{int}(1^{t_1})$, then $\mathsf{str}_{v_1}(\mathsf{int}(Y) + r \bmod 2^{v_1}) \oplus Y$ is

$$1^{s_1} \,\|\, \mathsf{str}_{t_1}(\mathsf{int}(Y'') + \mathsf{int}(1^{t_1}) \bmod 2^{t_1}) \oplus Y''.$$

It takes t_1 values from Lemma 4 if Y'' ranges over $\mathsf{int}(0^{t_1-1}1) \leq \mathsf{int}(Y'') \leq \mathsf{int}(1^{t_1})$, however, all these values are contained in the values appeared in the analysis of the second case of $Y = (01^{s_1-1} \,\|\, Y'')$.

We have covered all the values of $Y \in \{0,1\}^{v_1}$, and this completes the proof of the lemma. □

Lemma 6. *If $\ell = 2$, $s_2 \geq 1$, and $\mathsf{str}_{v_2}(r) = 0^{s_2}1^{t_2}0^{s_1}1^{t_1}$, then $A_2 = s_2(t_2 + t_2s_1t_1 + t_1) + s_1t_1 + 1$.*

Proof. We write $Y = (Y' \,\|\, Y'')$ for $Y' \in \{0,1\}^{s_2+t_2}$ and $Y'' \in \{0,1\}^{s_1+t_1}$. First, by a similar analysis to Lemma 5, $\mathsf{str}_{v_1}(\mathsf{int}(Y'') + r \bmod 2^{v_1}) \oplus Y''$ takes $(s_1t_1 + 1)$ values when Y'' ranges over $\mathsf{int}(0^{s_1+t_1}) \leq \mathsf{int}(Y'') \leq \mathsf{int}(1^{s_1}0^{t_1})$. We next fix any Y'' that satisfies $\mathsf{int}(0^{s_1+t_1}) \leq \mathsf{int}(Y'') \leq \mathsf{int}(1^{s_1}0^{t_1})$. Then, again by a similar analysis to Lemma 5, $\mathsf{str}_{v_2}(\mathsf{int}(Y) + r \bmod 2^{v_2}) \oplus Y$ takes (s_2t_2+1) values when Y' ranges over $Y' \in \{0,1\}^{s_2+t_2}$. Overall, $\mathsf{str}_{v_2}(\mathsf{int}(Y)+r \bmod 2^{v_2}) \oplus Y$ takes $(s_2t_2 + 1)(s_1t_1 + 1)$ values when Y ranges over $Y = (Y' \,\|\, Y'') \in \{0,1\}^{v_2}$, where $Y' \in \{0,1\}^{s_2+t_2}$, $Y'' \in \{0,1\}^{s_1+t_1}$, and $\mathsf{int}(0^{s_1+t_1}) \leq \mathsf{int}(Y'') \leq \mathsf{int}(1^{s_1}0^{t_1})$.

We next consider the case where Y ranges over $Y = (Y' \,\|\, Y'') \in \{0,1\}^{v_2}$, $Y' \in \{0,1\}^{s_2+t_2}$, $Y'' \in \{0,1\}^{s_1+t_1}$, and $\mathsf{int}(1^{s_1}0^{t_1-1}1) \leq \mathsf{int}(Y'') \leq \mathsf{int}(1^{s_1+t_1})$. Observe that

$$\mathsf{str}_{v_1}(\mathsf{int}(Y'') + r \bmod 2^{v_1}) \oplus Y'' = (0^{s_1} \,\|\, Y'''),$$

where $Y''' \in \{0,1\}^{t_1}$, and from Lemma 4, Y''' takes t_1 values when Y' is fixed and Y'' ranges over $\mathsf{int}(1^{s_1}0^{t_1-1}1) \leq \mathsf{int}(Y'') \leq \mathsf{int}(1^{s_1+t_1})$. As for the most significant $s_2 + t_2$ bits, we have a carry coming from the least significant $s_1 + t_1$ bits, and hence we need to consider

$$\mathsf{str}_{s_2+t_2}(\mathsf{int}(Y') + \mathsf{int}(0^{s_2}1^{t_2}) + \mathsf{int}(0^{s_2+t_2-1}1) \bmod 2^{s_2+t_2}) \oplus Y'. \tag{17}$$

Since $\mathsf{int}(0^{s_2}1^{t_2}) + \mathsf{int}(0^{s_2+t_2-1}1) = \mathsf{int}(0^{s_2-1}10^{t_2})$, (17) is

$$\mathsf{str}_{s_2+t_2}(\mathsf{int}(Y') + \mathsf{int}(0^{s_2-1}10^{t_2}) \bmod 2^{s_2+t_2}) \oplus Y'. \tag{18}$$

We see that the least significant t_2 bits of (18) is fixed to 0^{t_2}. If $s_2 = 1$, then we use Lemma 3 to see that (18) takes $1 = s_2$ value. If $s_2 \geq 2$, then we use Lemma 5 with $(s_1, t_1) = (s_2 - 1, 1)$ and we see that (18) takes $(s_2 - 1) \times 1 + 1 = s_2$ different values.

Overall, we have s_2t_2 different values, and hence we have the lemma by simplifying $(s_2t_2 + 1)(s_1t_1 + 1) + s_2t_2$. □

If we consider v_2+1 bits instead of v_2 bits by replacing s_2 with s_2+1, then by following a similar argument to the proof of Lemma 6, we see that

$$\#\left\{\mathsf{str}_{v_2+1}(\mathsf{int}(Y)+r \bmod 2^{v_2+1}) \oplus Y \mid Y=(0 \,\|\, Y'), Y' \in \{0,1\}^{v_2}\right\},$$

for $\mathsf{str}_{v_2+1}(r)=0^{s_2+1}1^{t_2}0^{s_1}1^{t_1}$, is $(s_2+1)(t_2+t_2s_1t_1+t_1)+s_1t_1+1$. The difference to the value of A_2 in Lemma 6, $t_2+t_2s_1t_1+t_1$, represents the number of $Y \in \{0,1\}^{v_2}$ such that the most significant bit of $\mathsf{str}_{v_2+1}(\mathsf{int}(Y)+r \bmod 2^{v_2+1}) \oplus (0 \,\|\, Y)$ is 1 and hence such Y has a carry to the (v_2+1)-st bit.

Lemma 7. *If $\ell=2$, $s_2=0$, and $\mathsf{str}_{v_2}(r)=1^{t_2}0^{s_1}1^{t_1}$, then $A_2=t_2(s_1t_1+1)+t_1$.*

Proof. As in the proof of Lemma 6, we write $Y=(Y' \,\|\, Y'')$ for $Y' \in \{0,1\}^{t_2}$ and $Y'' \in \{0,1\}^{s_1+t_1}$. First, $\mathsf{str}_{v_1}(\mathsf{int}(Y'')+r \bmod 2^{v_1}) \oplus Y''$ takes (s_1t_1+1) values when Y'' ranges over $\mathsf{int}(0^{s_1+t_1}) \le \mathsf{int}(Y'') \le \mathsf{int}(1^{s_1}0^{t_1})$, and for any fixed Y'' within the range, from Lemma 3, $\mathsf{str}_{v_2}(\mathsf{int}(Y)+r \bmod 2^{v_2}) \oplus Y$ takes t_2 values when Y' ranges over $Y' \in \{0,1\}^{t_2}$. Overall, $\mathsf{str}_{v_2}(\mathsf{int}(Y)+r \bmod 2^{v_2}) \oplus Y$ takes $t_2(s_1t_1+1)$ values when Y ranges over $Y=(Y' \,\|\, Y'') \in \{0,1\}^{v_2}$, where $Y' \in \{0,1\}^{t_2}$, $Y'' \in \{0,1\}^{s_1+t_1}$, and $\mathsf{int}(0^{s_1+t_1}) \le \mathsf{int}(Y'') \le \mathsf{int}(1^{s_1}0^{t_1})$.

We next consider the case where Y ranges over $Y=(Y' \,\|\, Y'') \in \{0,1\}^{v_2}$, $Y' \in \{0,1\}^{t_2}$, $Y'' \in \{0,1\}^{s_1+t_1}$, and $\mathsf{int}(1^{s_1}0^{t_1-1}1) \le \mathsf{int}(Y'') \le \mathsf{int}(1^{s_1+t_1})$. Then we see that $\mathsf{str}_{v_1}(\mathsf{int}(Y'')+r \bmod 2^{v_1}) \oplus Y''=(0^{s_1} \,\|\, Y''')$, where $Y''' \in \{0,1\}^{t_1}$, and from Lemma 4, Y''' takes t_1 values when Y' is fixed and Y'' ranges over $\mathsf{int}(1^{s_1}0^{t_1-1}1) \le \mathsf{int}(Y'') \le \mathsf{int}(1^{s_1+t_1})$. As for the most significant t_2 bits, we have a carry coming from the least significant s_1+t_1 bits. This fixes $\mathsf{str}_{t_2}(\mathsf{int}(Y')+\mathsf{int}(1^{t_2})+\mathsf{int}(0^{t_2-1}1) \bmod 2^{t_2}) \oplus Y'$ to 0^{t_2}, implying that we have t_1 different values in this case, and hence we have the lemma. □

We are now ready to present the proof of Proposition 1.

Proof (of Proposition 1). The proof is based on an induction on ℓ. First, if $\ell=1$, then we obtain Proposition 1 from Lemma 3 and Lemma 4.

Suppose that Proposition 1 holds for $\ell=j\ge 1$. If $s_j \ge 1$, the number of $Y \in \{0,1\}^{v_j}$ such that the most significant bit of $\mathsf{str}_{v_j+1}(\mathsf{int}(Y)+r \bmod 2^{v_j+1}) \oplus (0 \,\|\, Y)$ equals 1 is obtained by the difference between A_j with s_j being replaced with s_j+1 and A_j, which is

$$(s_j+1)B_j+A_{j-1}-A_j=B_j. \tag{19}$$

Now we show that Proposition 1 holds for $\ell=j+1$. We divide $Y \in \{0,1\}^{v_{j+1}}$ into $Y=(Y' \,\|\, Y'')$, where $Y' \in \{0,1\}^{s_{j+1}+t_{j+1}}$ and $Y'' \in \{0,1\}^{v_j}$.

Suppose $s_{j+1} \ge 1$. Then from the induction hypothesis and by following the same argument to the proof of Lemma 6, A_{j+1} is the sum of $(s_{j+1}t_{j+1}+1)A_j$ (for Y ranges over $Y' \in \{0,1\}^{s_{j+1}+t_{j+1}}$ and $\mathsf{int}(0^{v_j}) \le \mathsf{int}(Y'') \le \mathsf{int}(1^{s_j}0^{t_j}\dots 1^{s_1}0^{t_1})$) and $s_{j+1}B_j$ (for Y ranges over $Y' \in \{0,1\}^{s_{j+1}+t_{j+1}}$ and $\mathsf{int}(1^{s_j}0^{t_j}\dots 1^{s_1}0^{t_1-1}1) \le \mathsf{int}(Y'') \le \mathsf{int}(1^{v_j})$), where the latter value is obtained using (19). Therefore,

$$A_{j+1}=(s_{j+1}t_{j+1}+1)A_j+s_{j+1}B_j=s_{j+1}(t_{j+1}A_j+B_j)+A_j=s_{j+1}B_{j+1}+A_j$$

as desired.

On the other hand, if $s_{j+1}=0$, then by following the same argument to the proof of Lemma 7, we see that A_{j+1} is the sum of $t_{j+1}A_j$ (for Y ranges over $Y' \in \{0,1\}^{s_{j+1}+t_{j+1}}$ and $\mathsf{int}(0^{v_j}) \le \mathsf{int}(Y'') \le \mathsf{int}(1^{s_j}0^{t_j}\dots 1^{s_1}0^{t_1})$) and B_j (for Y ranges over $Y' \in \{0,1\}^{s_{j+1}+t_{j+1}}$ and $\mathsf{int}(1^{s_j}0^{t_j}\dots 1^{s_1}0^{t_1-1}1) \le \mathsf{int}(Y'') \le \mathsf{int}(1^{v_j})$). Therefore,

$$A_{j+1}=t_{j+1}A_j+B_j,$$

and we thus obtain the proposition. □

Table 5. List of (r, α_r) that satisfies $\max\{\alpha_0, \ldots, \alpha_{r-1}\} < \alpha_r$

r	α_r	r	α_r	r	α_r	r	α_r
0x00000001	32	0x000008ab	3263	0x000092ab	20061	0x0004aa55	70910
0x00000003	61	0x0000092b	3628	0x000094ab	20354	0x0004aaab	79295
0x00000005	89	0x0000094b	3689	0x0000952b	20403	0x00052aab	81947
0x00000009	115	0x00000953	3751	0x0000954b	20453	0x00054aab	82334
0x0000000b	143	0x00000955	4124	0x00009553	20754	0x000552ab	82391
0x00000013	194	0x00000a55	4271	0x00009555	22811	0x000554ab	82403
0x00000015	221	0x00000a95	4289	0x0000a555	23602	0x0005552b	82430
0x00000023	241	0x00000aab	4804	0x0000a955	23717	0x0005554b	82607
0x00000025	294	0x00001155	5025	0x0000aa55	23731	0x00055553	83819
0x0000002b	346	0x00001253	5071	0x0000aaab	26539	0x00055555	92126
0x00000045	361	0x00001255	5584	0x00011555	27479	0x0008aaab	94278
0x00000049	386	0x00001295	5661	0x00012553	27766	0x00092aab	104627
0x0000004b	463	0x000012ab	6351	0x00012555	30519	0x00094aab	106137
0x00000053	487	0x000014ab	6577	0x00012955	30962	0x000952ab	106358
0x00000055	538	0x0000152b	6612	0x00012a55	31023	0x000954ab	106395
0x0000008b	570	0x0000154b	6631	0x00012aab	34702	0x0009552b	106433
0x00000093	643	0x00001553	6729	0x00014aab	35893	0x0009554b	106662
0x00000095	717	0x00001555	7396	0x000152ab	36067	0x00095553	108227
0x000000a5	739	0x000022ab	7720	0x000154ab	36094	0x00095555	118953
0x000000ab	837	0x000024ab	8579	0x0001552b	36109	0x000a5555	122867
0x00000115	880	0x0000252b	8707	0x0001554b	36187	0x000a9555	123438
0x00000125	973	0x0000254b	8744	0x00015553	36718	0x000aa555	123521
0x0000012b	1115	0x00002553	8875	0x00015555	40357	0x000aa955	123531
0x0000014b	1158	0x00002555	9755	0x00022aab	41645	0x000aaaab	138117
0x00000153	1181	0x00002955	10099	0x00024aab	46242	0x00115555	140612
0x00000155	1299	0x00002a55	10148	0x000252ab	46913	0x00125553	141927
0x0000022b	1367	0x00002aab	11357	0x000254ab	47013	0x00125555	155993
0x0000024b	1523	0x00004555	11826	0x0002552b	47042	0x00129555	158237
0x00000253	1568	0x00004953	11951	0x0002554b	47145	0x0012a555	158564
0x00000255	1727	0x00004955	13139	0x00025553	47837	0x0012a955	158609
0x00000295	1788	0x00004a55	13329	0x00025555	52578	0x0012aaab	177343
0x000002ab	2013	0x00004a95	13346	0x00029555	54361	0x0014aaab	183066
0x00000455	2113	0x00004aab	14941	0x0002a555	54621	0x00152aab	183901
0x00000495	2347	0x000052ab	15464	0x0002a955	54658	0x00154aab	184023
0x000004a5	2368	0x000054ab	15541	0x0002aaab	61118	0x001552ab	184042
0x000004ab	2672	0x0000552b	15557	0x00045555	62825	0x001554ab	184053
0x0000052b	2769	0x0000554b	15592	0x00049553	63453	0x0015552b	184111
0x0000054b	2789	0x00005553	15821	0x00049555	69742	0x0015554b	184506
0x00000553	2832	0x00005555	17389	0x0004a555	70751	0x00155553	187213
0x00000555	3113	0x00008aab	18059	0x0004a955	70897	0x00155555	205767

C List of (r, α_r) and Applications

We present a list of (r, α_r) that satisfies $\max\{\alpha_0, \ldots, \alpha_{r-1}\} < \alpha_r$ in Table 5 and in Table 6. The list has 303 values, and it corresponds to Table 1 (left). The list of the 303 values can be used to obtain a more precise value of $\beta(r_{\max})$ appears in Corollary 2.

Let $r_{\max}$ be the upper bound of r as in Corollary 2, and let r_1 and r_2 be two consecutive values of r in the list. If $r_1 \leq r_{\max} < r_2$, then we have $\beta(r_{\max}) = \alpha_{r_1}$, i.e., for any $0 \leq r \leq r_{\max}$, we have $\Pr_L[\mathsf{Coll}_L(r, N_1, N_2)] \leq \beta(r_{\max})(\ell_N + 1)/2^n = \alpha_{r_1}(\ell_N + 1)/2^n$.

As noted in Sect. 6.3, for $r_{\max} = 0$, we have $\beta(r_{\max}) = 1$ from $\alpha_0 = 1$, and we also note that for $r_{\max} \geq$ `0x2aaaaaab`, we have $\beta(r_{\max}) = 3524578$. We present a graph in Fig. 5 showing the relation between $r_{\max}$ and $\beta(r_{\max})$ obtained from Table 5 and Table 6.

Table 6. List of (r, α_r) that satisfies $\max\{\alpha_0, \ldots, \alpha_{r-1}\} < \alpha_r$ (Cont'd)

r	α_r	r	α_r	r	α_r	r	α_r
0x0022aaab	208207	0x0095554b	501241	0x02aa5555	1162651	0x0aaaaaab	2374727
0x0024aaab	230887	0x00955553	508595	0x02aa9555	1162664	0x12aaaaab	2646170
0x00252aab	234196	0x00955555	559000	0x02aaaaab	1299901	0x14aaaaab	2685773
0x00254aab	234679	0x00a55555	575491	0x04955555	1367551	0x152aaaab	2691551
0x002552ab	234751	0x00a95555	577897	0x04a55555	1386393	0x154aaaab	2692394
0x002554ab	234772	0x00aa5555	578248	0x04a95555	1389142	0x1552aaab	2692517
0x0025552b	234847	0x00aa9555	578299	0x04aa5555	1389543	0x1554aaab	2692535
0x0025554b	235351	0x00aaa555	578305	0x04aa9555	1389601	0x15552aab	2692538
0x00255553	238804	0x00aaaaab	646568	0x04aaa555	1389606	0x15554aab	2692541
0x00255555	262471	0x01255555	704636	0x04aaaaab	1553633	0x155552ab	2692559
0x00295555	270744	0x01295555	714561	0x052aaaab	1590652	0x155554ab	2692682
0x002a5555	271951	0x012a5555	716009	0x054aaaab	1596053	0x1555552b	2693525
0x002a9555	272127	0x012a9555	716220	0x0552aaab	1596841	0x1555554b	2699303
0x002aa555	272152	0x012aa555	716249	0x0554aaab	1596956	0x15555553	2738906
0x002aaaab	304281	0x012aaaab	800799	0x05552aab	1596973	0x15555555	3010349
0x00455555	305645	0x014aaaab	823301	0x05554aab	1596977	0x25555555	3131742
0x00495553	308224	0x0152aaab	826584	0x055552ab	1596988	0x29555555	3149453
0x00495555	338771	0x0154aaab	827063	0x055554ab	1597061	0x2a555555	3152037
0x004a5555	343604	0x01552aab	827133	0x0555552b	1597561	0x2a955555	3152414
0x004a9555	344309	0x01554aab	827144	0x0555554b	1600988	0x2aa55555	3152469
0x004aa555	344411	0x015552ab	827151	0x05555553	1624477	0x2aa95555	3152477
0x004aa955	344420	0x015554ab	827189	0x05555555	1785473	0x2aaa5555	3152478
0x004aaaab	385084	0x0155552b	827448	0x092aaaab	1826673	0x2aaaaaab	3524578
0x0052aaab	396873	0x0155554b	829223	0x094aaaab	1851382		
0x0054aaab	398593	0x01555553	841389	0x0952aaab	1854987		
0x00552aab	398844	0x01555555	924776	0x0954aaab	1855513		
0x00554aab	398881	0x024aaaab	992659	0x09552aab	1855590		
0x005552ab	398889	0x0252aaab	1006511	0x09554aab	1855603		
0x005554ab	398908	0x0254aaab	1008532	0x095552ab	1855617		
0x0055552b	399033	0x02552aab	1008827	0x095554ab	1855702		
0x0055554b	399889	0x02554aab	1008871	0x0955552b	1856283		
0x00555553	405756	0x025552ab	1008884	0x0955554b	1860265		
0x00555555	445969	0x025554ab	1008931	0x09555553	1887558		
0x0092aaab	491816	0x0255552b	1009247	0x09555555	2074627		
0x0094aaab	498793	0x0255554b	1011412	0x0a555555	2116814		
0x00952aab	499811	0x02555553	1026251	0x0a955555	2122969		
0x00954aab	499960	0x02555555	1127959	0x0aa55555	2123867		
0x009552ab	499985	0x02955555	1157603	0x0aa95555	2123998		
0x009554ab	500011	0x02a55555	1161928	0x0aaa5555	2124017		
0x0095552b	500168	0x02a95555	1162559	0x0aaa9555	2124019		

Better Privacy Result. As discussed in Sect. 7.5, Theorem 1 can be improved by using Corollary 2, and Table 5 and Table 6 give a more precise value of $\beta(r_{\max})$ than the value obtained from Table 1.

We summarize our privacy result by using the following notation. Suppose that $\mathcal{A}$ makes q queries $(N_1, A_1, M_1), \ldots, (N_q, A_q, M_q)$, where $|N_i|_n = n_i$, $|A_i|_n = a_i$, and $|M_i|_n = m_i$. Then the total plaintext length is $m_1 + \cdots + m_q$, the maximum plaintext length is $\max\{m_1, \ldots, m_q\}$, the maximum nonce length is $\max\{n_1, \ldots, n_q\}$, and the total input length is $(a_1 + m_1) + \cdots + (a_q + m_q)$.

We state our result in the complexity theoretic form.

Corollary 5. *Let $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ 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, the maximum plaintext length is at most ℓ_M blocks, the maximum nonce length is at most ℓ_N blocks, and the*

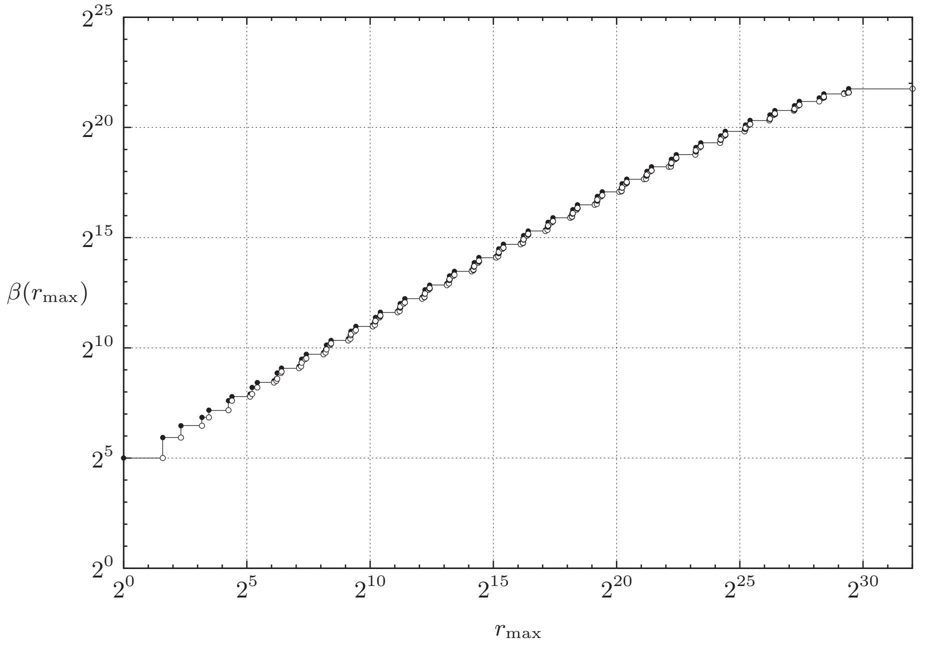

Fig. 5. Relation between $r_{\max}$ and $\beta(r_{\max})$ based on Table 5 and Table 6

total input length is at most σ_A blocks, there exists an adversary $\mathcal{A}'$ such that

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{GCM}[E,\tau]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{A}') + \frac{0.5(\sigma+q+1)^2}{2^n} + \frac{\beta(r_{\max})q(\sigma+q)(\ell_N+1)}{2^n},$$

where $\mathcal{A}'$ makes at most $\sigma+q+1$ queries, $r_{\max}$ is any value satisfying $\ell_M \leq r_{\max}$, and $\beta(r_{\max})$ is obtained from Table 5 and Table 6. Furthermore, the time complexity of $\mathcal{A}'$ is $\mathrm{Time}(\mathcal{A}) + cn\sigma_A$*, where* $\mathrm{Time}(\mathcal{A})$ *is the time complexity of $\mathcal{A}$ and c is a constant that depends only on the model of computation and the method of encoding.*

The corresponding information theoretic result of Corollary 5 improves Theorem 1, but we remark that the observation cannot be used to improve Corollary 3.

Better Authenticity Result. Theorem 2 can be improved by using Corollary 2, and [12, Theorem 2.3] can be used for Theorem 2 and Corollary 4.

Suppose that $\mathcal{A}$ makes q encryption queries $(N_1, A_1, M_1), \ldots, (N_q, A_q, M_q)$ and q' decryption queries $(N'_1, A'_1, C'_1, T'_1), \ldots, (N'_{q'}, A'_{q'}, C'_{q'}, T'_{q'})$, 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 plaintext length is $\max\{m_1, \ldots, m_q\}$, the maximum nonce length is $\max\{n_1, \ldots, n_q, n'_1, \ldots, n'_{q'}\}$, the total input length is $(a_1 + m_1) + \cdots + (a_q + m_q) + (a'_1 + m'_1) + \cdots + (a'_{q'} + m'_{q'})$, and the maximum input length is $\max\{a_1 + m_1, \ldots, a_q + m_q, a'_1 + m'_1, \ldots, a'_{q'} + m'_{q'}\}$.

We first apply Corollary 2 to Theorem 2. We have the following complexity theoretic result.

Corollary 6. *Let $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ 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 plaintext length is at most ℓ_M blocks, the maximum nonce length is at most ℓ_N blocks, the total input length is at most σ_A blocks, and the maximum* *input length is at most ℓ_A blocks, there exists an adversary $\mathcal{A}'$ such that*

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{GCM}[\mathrm{Perm}(n),\tau]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{A}') + \frac{0.5(\sigma+q+q'+1)^2}{2^n} + \frac{\beta(r_{\max})(q+q'+1)(\sigma+q)(\ell_N+1)}{2^n} + \frac{q'(\ell_A+1)}{2^\tau}, \qquad (20)$$

where $\mathcal{A}'$ makes at most $\sigma+q+q'+1$ queries, $r_{\max}$ is any value satisfying $\ell_M \leq r_{\max}$, and $\beta(r_{\max})$ is obtained from Table 5 and Table 6. Furthermore, the time complexity of $\mathcal{A}'$ is $\mathrm{Time}(\mathcal{A})+cn\sigma_A$*, where* $\mathrm{Time}(\mathcal{A})$ *is the time complexity of $\mathcal{A}$ and c is a constant that depends only on the model of computation and the method of encoding.*

The corresponding information theoretic result of Corollary 6 improves Theorem 2.

Next, if we use [12, Theorem 2.3], then instead of (20), we obtain

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{GCM}[E,\tau]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{A}') + \left[\frac{\beta(r_{\max})(q+q'+1)(\sigma+q)(\ell_N+1)}{2^n} + \frac{q'(\ell_A+1)}{2^\tau}\right] \cdot \delta_n(\sigma+q+q'+1). \qquad (21)$$

Recall that $\delta_n(a) = (1-(a-1)/2^n)^{-a/2}$. Furthermore, if the nonce length is restricted to 96 bits, then (20) becomes

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{GCM}[E,\tau]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{A}') + \left[\frac{q'(\ell_A+1)}{2^\tau}\right] \cdot \delta_n(\sigma+q+q'+1). \qquad (22)$$

The right hand side of (21) is generally smaller than that of (20) for most parameters. However, we do not know if this holds for all parameters. We also note that the last term of (22) is, for most parameters, smaller than the right hand side of (15).

D Proof of Theorem 1

Without loss of generality, we assume that $\mathcal{A}$ is deterministic and makes exactly q queries. Recall that $\mathrm{GCM}[\mathrm{Perm}(n),\tau]$ is GCM that is based on a random permutation $P \overset{\$}{\leftarrow} \mathrm{Perm}(n)$ and $\mathrm{GCM}[\mathrm{Rand}(n),\tau]$ is GCM based on a random function $F \overset{\$}{\leftarrow} \mathrm{Rand}(n)$. We follow the game playing proof technique in [11]. From the PRP/PRF switching lemma [11], have

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{GCM}[\mathrm{Perm}(n),\tau]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{priv}}_{\mathrm{GCM}[\mathrm{Rand}(n),\tau]}(\mathcal{A}) + \frac{(\sigma+q+1)^2}{2^{n+1}} \qquad (23)$$

as we need one call to P or F for L, and at most m_i+1 calls for the i-th query (N_i, A_i, M_i), and we have $\sum_{1\leq i\leq q}(m_i+1) \leq \sigma+q$.

We next define two games, Game G_0 and Game G_1, in Fig. 6 to derive the upper bound on $\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{GCM}[\mathrm{Rand}(n),\tau]}(\mathcal{A})$. In Fig. 6, Game G_1 includes the boxed statements and Game G_0 does not. Observe that Game G_0 simulates the random-bits oracle, and that Game G_1 simulates the GCM encryption oracle using the lazy sampling of F, where F is regarded as an array, and the array $F(X)$ is initially undefined for all $X \in \{0,1\}^n$. We maintain a set $\mathcal{I}_F$ in order to keep the record of domain points that have already been used. The set is updated when $F(X) \leftarrow Y$ is executed for some X and Y. We also maintain a flag, bad, which is initialized to false. We see that Game G_0 and Game G_1 are identical until the flag gets set, and hence, from the fundamental lemma of game playing [11], we have

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{GCM}[\mathrm{Rand}(n),\tau]}(\mathcal{A}) \leq \Pr[\mathcal{A}^{G_0} \text{ sets } \mathsf{bad}]. \qquad (24)$$

```
Game $G_0$   $\boxed{\textbf{Game } G_1}$
initialization
  $L \xleftarrow{\$} \{0,1\}^n$, $F(0^n) \leftarrow L$

if $\mathcal{A}$ makes a query $(N_i, A_i, M_i)$
  if $|N_i| = 96$ then $I_i[0] \leftarrow N_i \,\|\, 0^{31}1$
  else $I_i[0] \leftarrow \mathsf{GHASH}_L(\varepsilon, N_i)$
  $S_i[0] \xleftarrow{\$} \{0,1\}^n$
  if $I_i[0] \in \mathcal{I}_F$ then $\mathsf{bad} \leftarrow \mathsf{true}$, $\boxed{S_i[0] \leftarrow F(I_i[0])}$
  $F(I_i[0]) \leftarrow S_i[0]$
  for $j \leftarrow 1$ to $m_i$ do
      $I_i[j] \leftarrow \mathsf{inc}(I_i[j-1])$
      $S_i[j] \xleftarrow{\$} \{0,1\}^n$
      if $I_i[j] \in \mathcal{I}_F$ then $\mathsf{bad} \leftarrow \mathsf{true}$, $\boxed{S_i[j] \leftarrow F(I_i[j])}$
      $F(I_i[j]) \leftarrow S_i[j]$
  $C_i \leftarrow M_i \oplus \mathsf{msb}_{|M_i|}(S_i[1], \ldots, S_i[m_i])$
  $T_i \leftarrow \mathsf{msb}_\tau(S_i[0] \oplus \mathsf{GHASH}_L(A_i, C_i))$
  return $(C_i, T_i)$
```

Fig. 6. Game G_0 and Game G_1 for the proof of Theorem 1

We next evaluate $\Pr[\mathcal{A}^{G_0}$ sets $\mathsf{bad}]$. Since Game G_0 always returns a random string of $|M_i|+\tau$ bits for the i-the query (N_i, A_i, M_i), we can modify it to directly choose $(C_i, T_i) \xleftarrow{\$} \{0,1\}^{|M_i|+\tau}$ and return it to $\mathcal{A}$, which fixes the q queries made by $\mathcal{A}$, and then consider the probability that the bad flag gets set based on the randomness of L. Let $(N_1, A_1, M_1), \ldots, (N_q, A_q, M_q)$ be the q queries. The bad flag gets set in the following two cases.

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 $0 \le j \le m_i$.

We see that the event in Case (A) is equivalent to $\mathsf{inc}^j(I_i[0]) = 0^n$, and it is easy to see that the probability of this event is at most $(\sigma+q)(\ell_N+1)/2^n$, since if $|N_i| \ne 96$, then $\mathsf{inc}^j(I_i[0]) = 0^n$ is a non-trivial equation in L of degree at most $\ell_N + 1$ over GF(2^n), and otherwise the probability is obviously zero. We therefore have

$$\Pr[\mathcal{A}^{G_0} \text{ sets } \mathsf{bad} \text{ with Case (A)}] \le \frac{(\sigma+q)(\ell_N+1)}{2^n}. \tag{25}$$

For the event in Case (B), which is equivalent to $\mathsf{inc}^j(I_i[0]) = \mathsf{inc}^{j'}(I_{i'}[0])$, we see that for each $1 \le i \le q$, we have at most $(m_1+1) + (m_2+1) + \cdots + (m_{i-1}+1) + (i-1)(m_i+1)$ values of (j, i', j') to consider. We next evaluate $\Pr[\mathsf{inc}^j(I_i[0]) = \mathsf{inc}^{j'}(I_{i'}[0])]$ for each (i, j, i', j').

- If $|N_i| = |N_{i'}| = 96$, then $\Pr[\mathsf{inc}^j(I_i[0]) = \mathsf{inc}^{j'}(I_{i'}[0])] = 0$.
- If $|N_i| \ne 96$ and $|N_{i'}| = 96$, then $\Pr[\mathsf{inc}^j(I_i[0]) = \mathsf{inc}^{j'}(I_{i'}[0])] \le (\ell_N+1)/2^n$ since $\mathsf{inc}^j(I_i[0]) = \mathsf{inc}^{j'}(I_{i'}[0])$ is a non-trivial equation in L of degree at most $\ell_N + 1$ over GF(2^n). The same observation holds for the case $|N_i| = 96$ and $|N_{i'}| \ne 96$.
- If $|N_i|, |N_{i'}| \ne 96$, then $\Pr[\mathsf{inc}^j(I_i[0]) = \mathsf{inc}^{j'}(I_{i'}[0])] \le 2^{22}(\ell_N+1)/2^n$ by applying Corollary 1 with $(r, N_1, N_2) = (j - j', N_i, N_{i'})$ if $j - j' \ge 0$, and $(r, N_1, N_2) = (j' - j, N_{i'}, N_i)$ otherwise.

From these analyses, we have $\Pr[\mathsf{inc}^j(I_i[0]) = \mathsf{inc}^{j'}(I_{i'}[0])] \le 2^{22}(\ell_N+1)/2^n$ for any case. Now it is easy to verify

$$\sum_{1 \le i \le q} (m_1+1) + (m_2+1) + \cdots + (m_{i-1}+1) + (i-1)(m_i+1) = (q-1) \sum_{1 \le i \le q} (m_i+1),$$

which is at most $(q-1)(\sigma+q)$. We therefore have

$$\Pr[\mathcal{A}^{G_0} \text{ sets } \mathsf{bad} \text{ with Case (B)}] \le \frac{2^{22}(q-1)(\sigma+q)(\ell_N+1)}{2^n}, \tag{26}$$

and we obtain the claimed bound from (23), (24), (25), and (26). □

We remark that the proof of Corollary 3 is obtained from $\Pr[\mathcal{A}^{G_0} \text{ sets } \mathsf{bad}] = 0$.

E Proof of Theorem 2

We evaluate $\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{GCM}[\mathrm{Perm}(n),\tau]}(\mathcal{A})$ following the game playing proof technique in [11]. Without loss of generality, we assume that $\mathcal{A}$ is deterministic and makes exactly q encryption queries and q' decryption queries. We also assume that the decryption oracle, if $\mathcal{A}$ succeeds in forgery, returns a bit 0 instead of the plaintext since the returned value has no effect on the success probability of $\mathcal{A}$. Then from the PRP/PRF switching lemma [11], we have

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{GCM}[\mathrm{Perm}(n),\tau]}(\mathcal{A}) \le \mathbf{Adv}^{\mathrm{auth}}_{\mathrm{GCM}[\mathrm{Rand}(n),\tau]}(\mathcal{A}) + \frac{(\sigma+q+q'+1)^2}{2^{n+1}}, \tag{27}$$

as we need one call to P or F for L, at most m_i+1 calls for the i-th encryption query (N_i, A_i, M_i), and at most one call for the i-th decryption query (N'_i, A'_i, C'_i, T'_i), and we have $\sum_{1\le i\le q}(m_i+1) \le \sigma+q$.

In order to evaluate $\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{GCM}[\mathrm{Rand}(n),\tau]}(\mathcal{A})$, we define two games, Game G_0 and Game G_1, in Fig. 7, where Game G_1 includes the boxed statements and G_0 does not. Game G_1 simulates the GCM encryption oracle in lines 2–22 and the decryption oracle in lines 23–34 using the lazy sampling of F. In Fig. 7, we maintain a set $\mathcal{N}$ to keep the record of nonces that have already been used. The set is initially empty, and is updated when $\mathcal{A}$ makes a query. Observe that if the nonce N_i of the i-th encryption query satisfies $N_i \in \mathcal{N}$ (lines 15–22), then the corresponding $I_i[0]$ (used in line 16) and $S_i[0]$ (used in line 21) are already defined. Similarly, if the nonce N'_i of the i-th decryption query satisfies $N'_i \in \mathcal{N}$ (lines 32–34), then the corresponding $S'_i[0]$ (used in line 32) is already defined. In Fig. 7, we also maintain a set $\mathcal{I}'_F$ which keeps the domain points of F that have been used only for decryption queries. That is, for the i-th decryption query (N'_i, A'_i, C'_i, T'_i), the set $\mathcal{I}'_F$ is defined to be

$$\{I'_1[0], \dots, I'_{i-1}[0]\} \setminus \{0^n, I_1[0], \dots, I_1[m_1], \dots, I_j[0], \dots, I_j[m_j]\}$$

assuming that $\mathcal{A}$ has made j encryption queries before making the i-th decryption query. The set $\mathcal{I}'_F$ is used in line 26. Intuitively, for the j-th and i-th nonces N'_j and N'_i used only for decryption queries, where $j < i$ and $N'_j \ne N'_i$, the set is maintained to reuse the previous $S'_j[0]$ as $S'_i[0]$ if $I'_j[0] = I'_i[0]$ holds. We also maintain bad_1 and bad_2 flags.

Game G_0 is obtained from Game G_1 by removing the boxed statements, and we see that Game G_0 and Game G_1 are identical until one of the flags gets set. Therefore, from the fundamental lemma of game playing [11], we have

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{GCM}[\mathrm{Rand}(n),\tau]}(\mathcal{A}) \le \Pr[\mathcal{A}^{G_0} \text{ sets } \mathsf{bad}_1 \text{ or } \mathsf{bad}_2]. \tag{28}$$

We next evaluate $\Pr[\mathcal{A}^{G_0} \text{ sets } \mathsf{bad}_1 \text{ or } \mathsf{bad}_2]$. Observe that Game G_0 always returns a random string of $|M_i|+\tau$ bits for the i-th encryption query (N_i, A_i, M_i), or the symbol $\perp$ for the i-th decryption query (N'_i, A'_i, C'_i, T'_i). We then modify Game G_0 to choose $(C_i, T_i) \xleftarrow{\$} \{0,1\}^{|M_i|+\tau}$ and return it to $\mathcal{A}$ for the i-th encryption query, and return $\perp$ for the i-th decryption query. This fixes all queries, and we let the q encryption queries be $(N_1, A_1, M_1), \dots, (N_q, A_q, M_q)$, and the q' decryption queries be $(N'_1, A'_1, C'_1, T'_1), \dots, (N'_{q'}, A'_{q'}, C'_{q'}, T'_{q'})$.

```
**Game $G_0$** $\boxed{\textbf{Game } G_1}$
**initialization**
$L \xleftarrow{\$} \{0,1\}^n$, $F(0^n) \leftarrow L$

**if $\mathcal{A}$ makes a query $(N_i, A_i, M_i)$ such that $N_i \notin \mathcal{N}$**
**if** $|N_i| = 96$ **then** $I_i[0] \leftarrow N_i \,\|\, 0^{31}1$
**else** $I_i[0] \leftarrow \mathsf{GHASH}_L(\varepsilon, N_i)$
$S_i[0] \xleftarrow{\$} \{0,1\}^n$
**if** $I_i[0] \in \mathcal{I}_F$ **then** $\mathsf{bad}_1 \leftarrow \mathsf{true}$, $\boxed{S_i[0] \leftarrow F(I_i[0])}$
$F(I_i[0]) \leftarrow S_i[0]$
**for** $j \leftarrow 1$ **to** $m_i$ **do**
    $I_i[j] \leftarrow \mathsf{inc}(I_i[j-1])$
    $S_i[j] \xleftarrow{\$} \{0,1\}^n$
    **if** $I_i[j] \in \mathcal{I}_F$ **then** $\mathsf{bad}_1 \leftarrow \mathsf{true}$, $\boxed{S_i[j] \leftarrow F(I_i[j])}$
    $F(I_i[j]) \leftarrow S_i[j]$
$C_i \leftarrow M_i \oplus \mathsf{msb}_{|M_i|}(S_i[1], \ldots, S_i[m_i])$
$T_i \leftarrow \mathsf{msb}_\tau(S_i[0] \oplus \mathsf{GHASH}_L(A_i, C_i))$
**return** $(C_i, T_i)$

**if $\mathcal{A}$ makes a query $(N_i, A_i, M_i)$ such that $N_i \in \mathcal{N}$**
**for** $j \leftarrow 1$ **to** $m_i$ **do**
    $I_i[j] \leftarrow \mathsf{inc}(I_i[j-1])$
    $S_i[j] \xleftarrow{\$} \{0,1\}^n$
    **if** $I_i[j] \in \mathcal{I}_F$ **then** $\mathsf{bad}_1 \leftarrow \mathsf{true}$, $\boxed{S_i[j] \leftarrow F(I_i[j])}$
    $F(I_i[j]) \leftarrow S_i[j]$
$C_i \leftarrow M_i \oplus \mathsf{msb}_{|M_i|}(S_i[1], \ldots, S_i[m_i])$
$T_i \leftarrow \mathsf{msb}_\tau(S_i[0] \oplus \mathsf{GHASH}_L(A_i, C_i))$
**return** $(C_i, T_i)$

**if $\mathcal{A}$ makes a query $(N'_i, A'_i, C'_i, T'_i)$ such that $N'_i \notin \mathcal{N}$**
**if** $|N'_i| = 96$ **then** $I'_i[0] \leftarrow N'_i \,\|\, 0^{31}1$
**else** $I'_i[0] \leftarrow \mathsf{GHASH}_L(\varepsilon, N'_i)$
$S'_i[0] \xleftarrow{\$} \{0,1\}^n$
**if** $I'_i[0] \in \mathcal{I}'_F$ **then** $S'_i[0] \leftarrow F(I'_i[0])$
**else if** $I'_i[0] \in \mathcal{I}_F$ **then** $\mathsf{bad}_1 \leftarrow \mathsf{true}$, $\boxed{S'_i[0] \leftarrow F(I'_i[0])}$
$F(I'_i[0]) \leftarrow S'_i[0]$
$T^*_i \leftarrow \mathsf{msb}_\tau(S'_i[0] \oplus \mathsf{GHASH}_L(A'_i, C'_i))$
**if** $T'_i = T^*_i$ **then** $\mathsf{bad}_2 \leftarrow \mathsf{true}$, $\boxed{\textbf{return } 0}$
**return** $\perp$

**if $\mathcal{A}$ makes a query $(N'_i, A'_i, C'_i, T'_i)$ such that $N'_i \in \mathcal{N}$**
$T^*_i \leftarrow \mathsf{msb}_\tau(S'_i[0] \oplus \mathsf{GHASH}_L(A'_i, C'_i))$
**if** $T'_i = T^*_i$ **then** $\mathsf{bad}_2 \leftarrow \mathsf{true}$, $\boxed{\textbf{return } 0}$
**return** $\perp$
```

Fig. 7. Game G_0 and Game G_1 for the proof of Theorem 2

We evaluate $\Pr[\mathcal{A}^{G_0}$ sets $\mathsf{bad}_1]$. The q' decryption queries can be divided into two sets. $\mathcal{Q}_1$ is a set of decryption queries where the corresponding nonces are also used in encryption queries, and $\mathcal{Q}_2$ is a set of decryption queries where the nonces are used only for decryption queries. To evaluate $\Pr[\mathcal{A}^{G_0}$ sets $\mathsf{bad}_1]$, we consider the encryption queries first (which effectively include the decryption queries in $\mathcal{Q}_1$), followed by the decryption queries in $\mathcal{Q}_2$. Then we have the following four cases to consider.

Case (A). $I_i[j] = 0^n$ holds for some (i, j) such that $1 \leq i \leq q$ and $0 \leq j \leq m_i$.
Case (B). $I_i[j] = I_{i'}[j']$ holds for some (i, j, i', j') such that $1 \leq i' < i \leq q$, $0 \leq j' \leq m_{i'}$, and $0 \leq j \leq m_i$.
Case (C). $I'_i[0] = 0^n$ holds for some i such that $1 \leq i \leq q'$ and $(N'_i, A'_i, C'_i, T'_i) \in \mathcal{Q}_2$.
Case (D). $I'_i[0] = I_{i'}[j']$ holds for some (i, i', j') such that $1 \leq i \leq q'$, $(N'_i, A'_i, C'_i, T'_i) \in \mathcal{Q}_2$, $1 \leq i' \leq q$, and $0 \leq j' \leq m_{i'}$.

The analyses for Case (A) and Case (B) are the same as those for (25) and (26), respectively, and we have

$$\Pr[\mathcal{A}^{G_0} \text{ sets } \mathsf{bad}_1 \text{ with Case (A)}] \leq \frac{(\sigma + q)(\ell_N + 1)}{2^n} \tag{29}$$

and

$$\Pr[\mathcal{A}^{G_0} \text{ sets } \mathsf{bad}_1 \text{ with Case (B)}] \leq \frac{2^{22}(q-1)(\sigma + q)(\ell_N + 1)}{2^n}. \tag{30}$$

For Case (C), it is easy to verify that

$$\Pr[\mathcal{A}^{G_0} \text{ sets } \mathsf{bad}_1 \text{ with Case (C)}] \leq \frac{q'(\ell_N + 1)}{2^n}, \tag{31}$$

since for each $1 \leq i \leq q'$ with $(N'_i, A'_i, C'_i, T'_i) \in \mathcal{Q}_2$, we have $\Pr[I'_i[0] = 0^n] \leq (\ell_N + 1)/2^n$. It remains to evaluate the event in Case (D), which is equivalent to $I'_i[0] = \mathsf{inc}^{j'}(I_{i'}[0])$. Then by a similar argument to the analysis of (26), for each (i, i', j') with $(N'_i, A'_i, C'_i, T'_i) \in \mathcal{Q}_2$, we have $\Pr[I'_i[0] = \mathsf{inc}^{j'}(I_{i'}[0])] \leq 2^{22}(\ell_N + 1)/2^n$. For each $1 \leq i \leq q'$, we have at most $\sum_{1 \leq j \leq q}(m_j + 1) \leq \sigma + q$ values of (i', j') to consider. We therefore have

$$\Pr[\mathcal{A}^{G_0} \text{ sets } \mathsf{bad}_1 \text{ with Case (D)}] \leq \frac{2^{22}q'(\sigma + q)(\ell_N + 1)}{2^n}, \tag{32}$$

and obtain

$$\Pr[\mathcal{A}^{G_0} \text{ sets } \mathsf{bad}_1] \leq \frac{2^{22}(q + q' + 1)(\sigma + q)(\ell_N + 1)}{2^n} \tag{33}$$

from (29), (30), (31), and (32).

We next evaluate $\Pr[\mathcal{A}^{G_0} \text{ sets } \mathsf{bad}_2]$ by following the order of encryption/decryption queries made by $\mathcal{A}$. For each $1 \leq i \leq q'$, we consider two cases. If $N'_i \notin \{N_1, \ldots, N_j\}$, where $N_1, \ldots, N_j$ denote nonces used in encryption queries before making the i-th decryption query, then the event $T'_i = T^*_i$ is equivalent to $T'_i = \mathsf{msb}_\tau(S'_i[0] \oplus \mathsf{GHASH}_L(A'_i, C'_i))$, and since $S'_i[0]$ can be treated as a uniform random n-bit string (which might have been chosen for the previous decryption query), we have $\Pr[T'_i = T^*_i] = 1/2^\tau$ in this case. On the other hand, if $N'_i \in \{N_1, \ldots, N_j\}$, then there exists an encryption query, $(N_{j'}, A_{j'}, M_{j'})$, satisfying $N'_i = N_{j'}$. Let $T_{j'}$ and $C_{j'}$ be the corresponding tag and ciphertext. Then the event $T'_i = T^*_i$ is equivalent to

$$T'_i \oplus T_{j'} = \mathsf{msb}_\tau(\mathsf{GHASH}_L(A'_i, C'_i) \oplus \mathsf{GHASH}_L(A_{j'}, C_{j'})). \tag{34}$$

Now if $(A'_i, C'_i) = (A_{j'}, C_{j'})$, then we necessary have $T'_i \neq T_{j'}$, and hence (34) cannot hold. If $(A'_i, C'_i) \neq (A_{j'}, C_{j'})$, then for any $Y \in \{0,1\}^n$, $Y = \mathsf{GHASH}_L(A'_i, C'_i) \oplus \mathsf{GHASH}_L(A_{j'}, C_{j'})$ has at most $\ell_A + 1$ solutions, which implies that (34) has at most $2^{n-\tau}(\ell_A + 1)$ solutions. Therefore, we have $\Pr[T'_i = T^*_i] \leq (\ell_A + 1)/2^\tau$ in this case.

From the analyses above, for each $1 \leq i \leq q'$, we have $\Pr[T'_i = T^*_i] \leq (\ell_A + 1)/2^\tau$, and hence we obtain

$$\Pr[\mathcal{A}^{G_0} \text{ sets } \mathsf{bad}_2] \leq \frac{q'(\ell_A + 1)}{2^\tau}. \tag{35}$$

The claimed bound is obtained from (27), (28), (33), and (35). □

Finally, we remark that the proof of Corollary 4 is obtained from $\Pr[\mathcal{A}^{G_0} \text{ sets } \mathsf{bad}_1] = 0$.