Bernstein Bound on WCS is Tight

Repairing Luykx-Preneel Optimal Forgeries

Mridul Nandi

Indian Statistical Institute, Kolkata
mridul.nandi@gmail.com

Abstract. In Eurocrypt 2018, Luykx and Preneel described hash-key-recovery and forgery attacks against polynomial hash based Wegman-Carter-Shoup (WCS) authenticators. Their attacks require $2^{n/2}$ message-tag pairs and recover hash-key with probability about 1.34×2^{-n} where n is the bit-size of the hash-key. Bernstein in Eurocrypt 2005 had provided an upper bound (known as Bernstein bound) of the maximum forgery advantages. The bound says that all adversaries making $O(2^{n/2})$ queries of WCS can have maximum forgery advantage $O(2^{-n})$. So, Luykx and Preneel essentially analyze WCS in a range of query complexities where WCS is known to be perfectly secure. Here we revisit the bound and found that WCS remains secure against all adversaries making $q \ll \sqrt{n} \times 2^{n/2}$ queries. So it would be meaningful to analyze adversaries with beyond birthday bound complexities.

In this paper, we show that *the Bernstein bound is tight* by describing two attacks (one in the *"chosen-plaintext model"* and other in the *"known-plaintext model"*) which **recover the hash-key (hence forges) with probability at least $\frac{1}{2}$ based on $\sqrt{n} \times 2^{n/2}$ message-tag pairs**. We also extend the forgery adversary to the Galois Counter Mode (or GCM). More precisely, we **recover the hash-key of GCM with probability at least $\frac{1}{2}$ based on only $\sqrt{\frac{n}{\ell}} \times 2^{n/2}$ encryption queries**, where ℓ is the number of blocks present in encryption queries.

Keywords: WCS authenticator, GCM, polynomial hash, universal hash, AXU, key-recovery, forgery.

1 Introduction

Wegman-Carter Authentication. In 1974 ([GMS74]), Gilbert, MacWilliams and Sloane considered a coding problem which is essentially an one-time authentication protocol (a fresh key is required for every authentication). Their solutions required a key which is as large as the message to be authenticated. Later in 1981, Wegman and Carter [WC81] proposed a simple authentication protocol based on an almost strongly universal$_2$ hash function which was described in their early work in [CW79]. The hash-key size is the order of logarithm of message length (which is further reduced by some constant factor due to Stinson [Sti94]). The hash-key can be the same for every authentication, but it needs a fresh constant sized random key (used to mask the hash-output). More precisely, let κ be a hash-key of an n-bit hash function ρ_κ and $R_1, R_2, \ldots$ be a stream of secret n-bit keys. Given a message m and its unique message number $\mathcal{n}$ (also known as a nonce), the Wegman-Carter (WC) authenticator computes $R_{\mathcal{n}} \oplus \rho_\kappa(m)$ as a tag.

Almost Xor-Universal or AXU Hash. In [Kra94] Krawczyk had shown that almost strong universal$_2$ property can be relaxed to a weaker hash (named as AXU or almost-xor universal hash by Rogaway in [Rog95]). The polynomial hashing [dB93,BJKS94,Tay94], division hashing [KR87,Rab81] are such examples of AXU hash functions which were first introduced in a slightly different context. Afterwards, many AXU hash functions have been proposed for instantiating Wegman-Carter authentication [Sho96,HK97,Ber05a,BHK$^+$99,MV04]. A comprehensive survey of universal hash functions can be found in [Ber07,Nan14]. Among all known examples, the polynomial hashing is very popular as it requires hash-key of constant size and, both key generation and hash computation are very fast.

Wegman-Carter-Shoup or WCS Authenticator. To get rid of onetime masking in Wegman-Carter authenticator, Brassard (in [Bra83]) proposed to use a pseudorandom number generator which generates the keys $R_1, R_2, \ldots,$ from a short master key K. However, in some application, message number can come in arbitrary order and so a direct efficient computation of $R_{\mathcal{n}}$ is much desired (it is alternatively known as pseudorandom function or PRF). Brassard pointed out that the Blum-Blum-Shub pseudorandom number generator [BBS86] outputs can be computed directly. As blockciphers are more efficient, Shoup ([Sho96]) considered the following variant of WC authentication:

$$\mathsf{WCS}_{K,\kappa}(\mathcal{n}_1, m) := e_K(\mathcal{n}) \oplus \rho_\kappa(m)$$

where e_K is a keyed blockcipher modeled as a pseudorandom permutation (PRP). This was named as WCS authenticator by Bernstein in [Ber05b].

The use of PRPs enables practical and fast instantiations of WCS authenticators. The WCS authentication mechanism implicitly or explicitly has been used in different algorithms, such as Poly1305-AES [Ber05a] and Galois Counter Mode or GCM [MV04,AY12]. GCM was adopted in practice, e.g. [MV06,JTC11,SCM08]. GCM and its randomized variants, called RGCM [BT16], are used in TLS 1.2 and TLS 1.3.

1.1 Known Security Analysis of WCS prior to Luykx-Preneel Eurocrypt 2018

Hash-Key Recovery Attacks of WCS. Forgery and key-recovery are the two meaningful security notions for an authenticator. Whenever we recover hash-key, the security is completely lost as any message can be forged. Security of WCS relies on the nonce which should not repeat over different executions [Jou,HP08]. Most of the previously published nonce respecting attacks aim to recover the polynomial key [ABBT15,PC15,Saa12,ZTG13] based on multiple verification attempts. The total number of message blocks in all verification attempts should be about 2^n to achieve some significant advantage.

Provable Security Analysis of WCS. The WC authenticator based on polynomial hashing has maximum forgery or authenticity advantage $\frac{v\ell}{2^n}$ against all adversaries who make at most q authentication queries and v verification queries consisting of at most ℓ blocks. By applying the standard PRP-PRF switching lemma, WCS (which is based on a random permutation π) has an authenticity advantage at most $\frac{v\ell}{2^n} + \frac{(v+q)^2}{2^n}$. So the bound becomes useless as q approaches $2^{n/2}$ (birthday complexity). Shoup proved that the advantage is at most $\frac{v\ell}{2^n}$ for all $q < 2^{\frac{n-\log \ell}{2}}$ [Sho96]. So, when $\ell = 2^{10}$, $n = 128$, the above bound says that the authenticity advantage is at most $v\ell/2^{128}$, whenever $q \leq 2^{59}$. This is clearly better than the classical bound. However, the application of Shoup's bound would be limited if we allow large ℓ.

Bernstein Bound. Finally, Bernstein [Ber05b] provided an improved bound for WCS which is valid for wider range of q. The maximum authenticity advantage is shown to be bounded above by

$$\mathsf{B}(q, v) := v \cdot \epsilon \cdot (1 - \frac{q}{2^n})^{\frac{-(q+1)}{2}} \tag{1}$$

for all q, where ρ_κ is an ϵ-AXU hash function. Thus, when $q = O(2^{n/2})$, the maximum success probability is $O(v \cdot \epsilon)$ which is clearly negligible for all reasonable choices of v and ϵ. For example, the forgery advantage against 128-bit WCS based on polynomial hashing is at most (1) $1.7v\ell \times 2^{-128}$ when $q \leq 2^{64}$, and (2) $3000v\ell \times 2^{-128}$ when $q = 2^{66}$ (so WCS remains secure even if we go beyond birthday bound query complexity).

1.2 Understanding the result due to Luykx and Preneel in [LP18]

False-key or True-key set. All known key-recovery attacks focus on reducing the set of candidate keys, denoted $\mathscr{T}$, which contains the actual key. But the set of candidate keys, also called true-key set, is constructed from verification attempts. Recently, a true-key set (equivalently false-key set which is simply the complement of the true-key set) is constructed from authentication queries only. After observing some authentication outputs of a WCS based on a blockcipher e_K, some choices for the key can be eliminated using the fact that outputs of the blockcipher are distinct. More precisely, we can construct the following false-key set $\mathscr{F}$ based on a transcript $\tau := ((n_1, m_1, t_1), \ldots, (n_q, m_q, t_q))$ where $t_i = e_K(n_i) \oplus \rho_\kappa(m_i)$:

$$\mathscr{F} := \{x \ : \ t_i \oplus \rho_x(m_i) = t_j \oplus \rho_x(m_j), \text{ for some } i \neq j\}. \tag{2}$$

It is easy to see that the hash-key $\kappa \notin \mathscr{F}$, since otherwise, there would exist $i \neq j$, $e_K(n_i) = e_K(n_j)$, which is a contradiction. So, a random guess of a key from outside the false-key set would be a correct guess with probability at least

$\frac{1}{2^n-\mathbb{E}(|\mathscr{F}|)}$. This simple but useful observation was made in [LP18]. We also use this idea in our analysis.

– Lower bound on the expected size of false-key set.
Based on the above discussion, one natural approach would be to maximize the false-key set to obtain higher key-recovery advantage. This has been considered in [LP18]. Proposition 3.1 of [LP18] states that

$$\mathbb{E}(|\mathscr{F}|) \geq \tfrac{q(q-1)}{4}, \text{ for all } q < \sqrt{2^n - 3}.$$

In other words, expected size of the false-key set grows quadratically. They have stated the following in section 3 of [LP18].

> "We describe chosen-plaintext attacks which perfectly match the bounds for both polynomial-based WCS MACs and GCM."

Issue 1: The Luykx-Preneel attack is no better than random guessing. Their attack can eliminate about one fourth keys. In other words, there are still three-fourth candidate keys are left. So, the key-recovery advantage $\mathsf{KR}(q)$ is about $\frac{1.34}{2^n}$ (1.34 times more than a random guess attack without making any query). Naturally, as the key-recovery advantage is extremely negligible, claiming such an algorithm as an attack is definitely under question.

– Upper bound on the expected size of false-key set.
Now we discuss the other claim of [LP18]. They claimed that (Theorem 5.1 of [LP18]) the size of the false-key set cannot be more than $q(q+1)/2$ after observing q responses of polynomial-based WCS. In other words, irrespective of the length of queries ℓ, the upper bound of the size of the false-key set is independent of ℓ. At a first glance this seem to be counter-intuitive as the number of roots of a polynomial corresponding to a pair of query-responses can be as large as ℓ. So, at best one may expect the size of the false-key set can be $\binom{q}{2}\ell$. But, on the other extreme there may not be a single root for may pairs of queries. On the average, the number of roots for every pair of messages turns out to be in the order of q^2, independent of ℓ. We investigate the proof of Theorem 5.1 of [LP18] and in the very first line they have mentioned that

> "Using Thm. 4.1, Cor. 5.1, and Prop. 5.3, we have..."

However, the Cor 5.1 is stated for all $q \leq M_\gamma$ (a parameter defined in Eq. 41 of [LP18]). They have not studied how big M_γ can be. We provide an estimation which allows us to choose M_γ such that $\ell\binom{M_\gamma}{2} = 2^n - \ell$. With this bound, the Theorem 5.1 can be restated as

$$\mathbb{E}(|\mathscr{F}|) \leq \frac{q(q+1)}{2} \quad \text{for all } q < \frac{2^{n/2}}{\sqrt{\ell}}. \tag{3}$$

By combining Proposition 3.1 and a corrected version of Theorem 5.1 as just mentioned, we can conclude that

$$\mathbb{E}(|\mathscr{F}|) = \Theta(q^2), \quad \text{for all } q < \frac{2^{n/2}}{\sqrt{\ell}}.$$

In other words, authors have found a tight estimate of expected size of the false-key set in a certain range of q.

Issue 2: Usefulness of an upper bound of the false-key set: The lower bound of the expected false-key set immediately leads to a lower bound of key-recovery advantage. However, an upper bound of the expected false-key set does not lead to an upper bound of key-recovery advantage. This is mainly due to the fact, the key-recovery advantage based on q authentication responses can be shown as

$$\mathsf{KR}(q) = \mathbb{E}(\frac{1}{2^n - |\mathscr{F}|}) \geq \frac{1}{2^n - \mathbb{E}(|\mathscr{F}|)}.$$

The inequality follows from the Jensen inequality. So an upper bound of $\mathbb{E}(|\mathscr{F}|)$ does not give any implication on $\mathsf{KR}(q)$. Moreover, dealing the expression $\mathbb{E}(1/(2^n - |\mathscr{F}|))$ directly is much harder. So the usefulness of an upper bound of the expected size of false-key set is not clear to us (other than understanding tightness of size of the false-key set which could be of an independent interest).

1.3 Our Contributions

In this paper, we resolve the optimality issue of the Bernstein bound. We first provide a tight alternative expression of the Berstein bound. In particular, we observe that $\mathsf{B}(q, v) = \Theta(v \cdot \epsilon \cdot e^{\frac{q^2}{2^{n+1}}})$. So WCS is secure against all adversaries with $q \ll \sqrt{n} \times 2^{n/2}$ queries. An adversary must make about $\sqrt{n} \times 2^{n/2}$ queries to obtain some significant advantage. In this paper we describe three attacks to recover the hash key and analyze their success probabilities.

1. The first two attacks (in the known-plaintext and the chosen-plaintext models) are against WCS based on a polynomial hash; they also work for other hashes satisfying certain regular property. Our attacks are also based a false-key (equivalently a true-key set) as described in the Luykx-Preneel attack. Unlike the Luykx-Preneel attack, we however choose message randomly in case of chosen-plaintext model. The query complexity of our attacks is also beyond the birthday complexity. In particular, these attacks require $\sqrt{n2^n}$ authentication queries. So the bound due to Bernstein is tight (even in the known-plaintext model) when $q \approx \sqrt{n2^n}$.
2. We also extend these attacks to the authentication algorithm of GCM which utilizes the ciphertext of GCM encryption to reduce the complexity of encryption queries. In particular, if each encryption query contains ℓ blocks, then this attack requires $\sqrt{\frac{n}{\ell} \times 2^n}$ encryption queries to recover the hash key used in GCM authentication. We have proved that our forgery is optimum by proving a tight upper bound on the maximum forgery advantage.
3. We also provide a simple proof on the tightness of the false-key set which works for all q. In particular, we show that the expected size of the false-key set is at most $q(q-1)/2^n$.

2 Preliminaries

Notations. We write $\mathsf{X} \leftarrow_{\$} \mathcal{X}$ to denote that the random variable X is sampled uniformly (and independently from all other random variables defined so far) from the set $\mathcal{X}$. Let $(a)_b := a(a-1)\cdots(a-b+1)$ for two positive integers $b \le a$. A tuple $(x_1, \ldots, x_q)$ is simply denoted as x^q. We call x^q *coordinate-wise distinct* if x_i's are distinct. We write the set $\{1, 2, \ldots, m\}$ as $[m]$ for a positive integer m. We use standard asymptotic notations such as $o(\cdot)$, $O(\cdot)$, $\Theta(\cdot)$ and $\Omega(\cdot)$ notations. For real functions $f(x), g(x)$, we write $f = O(g)$ (equivalently $g = \Omega(f)$) if there is some positive constant C such that $f(x) \le Cg(x)$ for all x. If both $f = O(g)$ and $g = O(f)$ hold then we write $f = \Theta(g)$. We write $f(x) = o(g(x))$ if $\lim\limits_{x \to \infty} \frac{f(x)}{g(x)} = 0$.

Jensen Inequality. We write $\mathbb{E}(\mathsf{X})$ to denote the expectation of a real valued random variable X. A twice differentiable function f is called convex if for all x (from the domain of f), $f''(x) > 0$. For example, (1) $1/x$ is a convex function over the set of all positive real numbers and (2) $\frac{1}{N-x}$ is convex over the set of all positive real number less than N. For every convex function f and a real valued random variable X, $\mathbb{E}(f(\mathsf{X})) \ge f(\mathbb{E}(\mathsf{X}))$ (Jensen Inequality). In particular, for all positive random variable X,

$$\mathbb{E}(\frac{1}{\mathsf{X}}) \ge \frac{1}{\mathbb{E}(\mathsf{X})} \tag{4}$$

and for all positive random variable $\mathsf{Y} < N$,

$$\mathbb{E}(\frac{1}{N-\mathsf{Y}}) \ge \frac{1}{N-\mathbb{E}(\mathsf{Y})} \tag{5}$$

Lemma 1. *Let $0 < \epsilon \le \sqrt{2} - 1$. Then, for all positive real $x \le \epsilon$,*

$$e^{-(1+\epsilon)x} \le 1 - x.$$

Proof. It is well known (from calculus) that $e^{-x} \le 1 - x + \frac{x^2}{2}$ for all real x. Let $\eta = 1 + \epsilon < \sqrt{2}$. So

$$\begin{aligned} e^{-(1+\epsilon)x} &\le 1 - (1+\epsilon)x + \frac{\eta^2 x^2}{2} \\ &\le 1 - (1+\epsilon)x + x^2 \\ &= 1 - x - x(\epsilon - x) \le 1 - x \qquad \Box \end{aligned}$$

We also know that $1 - x \le e^{-x}$. So, the above result informally says that $1-x$ and e^{-x} are "almost" the same whenever x is a small positive real number.

2.1 Security Definitions

Pseudorandom Permutation Advantage. Let $\mathsf{Perm}_{\mathfrak{B}}$ be the set of all permutations over $\mathfrak{B}$. A blockcipher over a block set $\mathfrak{B}$ is a function $e : \mathcal{K} \times \mathfrak{B} \to \mathfrak{B}$

such that for all key $k \in \mathscr{K}$, $e(k, \cdot) \in \mathsf{Perm}_{\mathfrak{B}}$. So, a blockcipher is a keyed family of permutations. A uniform random permutation or URP is denoted as π, where $\pi \leftarrow\!\!\$\ \mathsf{Perm}_{\mathfrak{B}}$. The pseudorandom permutation advantage of a distinguisher $\mathscr{A}$ against a blockcipher e is defined as

$$\mathsf{Adv}_e^{\mathrm{prp}}(\mathscr{A}) := \Big| \Pr_{K \leftarrow\!\!\$\ \mathscr{K}}(\mathscr{A}^{e_K} \text{ returns } 1) - \Pr_{\pi}(\mathscr{A}^{\pi} \text{ returns } 1)\Big|.$$

Let $\mathbb{A}(q,t)$ denote the set of all adversaries which runs in time at most t and make at most q queries to either a blockcipher or a random permutation. We write $\mathsf{Adv}^{\mathrm{prp}}(q,t) = \max\limits_{\mathscr{A} \in \mathbb{A}(q,t)} \mathsf{Adv}_e^{\mathrm{prp}}(\mathscr{A})$.

Authenticator. A nonce based authenticator with nonce space $\mathscr{N}$, key space $\mathscr{K}$, message space $\mathscr{M}$ and tag space $\mathfrak{B}$ is a function $\gamma : \mathscr{K} \times \mathscr{N} \times \mathscr{M} \to \mathfrak{B}$. We also write $\gamma(k, \cdot, \cdot)$ as $\gamma_k(\cdot, \cdot)$ and hence a nonce based authenticator can be viewed as a keyed family of functions. We say that (n, m, t) is *valid* for γ_k (or for a key k when γ is understood) if $\gamma_k(n, m) = t$. We define a verifier $\mathsf{Ver}_{\gamma_k} : \mathscr{N} \times \mathscr{M} \times \mathfrak{B} \to \{0, 1\}$ as

$$\mathsf{Ver}_{\gamma_k}(n, m, t) = \begin{cases} 1 & \text{if } (n, m, t) \text{ is valid for } \gamma_k, \\ 0 & \text{otherwise.} \end{cases}$$

We also simply write Ver_k instead of Ver_{γ_k}.

An adversary $\mathscr{A}$ against a nonce based authenticator makes authentication queries to γ_K and verification queries to Ver_K for a secretly sampled $K \leftarrow\!\!\$\ \mathscr{K}$. An adversary is called

- *nonce-respecting* if nonces in all authentication queries are distinct,
- *single-forgery* (or *multiple-forgery*) if it submits only one (or more than one) verification query,
- *key-recovery* if it finally returns an element from key space.

In this paper we only consider nonce-respecting algorithm. We also assume that $\mathscr{A}$ does not submit a verification query (n, m, t) to Ver_{γ_K} for which (n, m) has already been previously queried to the authentication oracle. Let $\mathbb{A}(q, v, t)$ denote the set of all such nonce-respecting algorithms which runs in time t and make at most q queries to an authenticator and at most v queries to its corresponding verifier. In this paper our main focus on analyzing the information-theoretic adversaries (which can run in unbounded time). So we write $\mathbb{A}(q, v) = \cup_{t<\infty} \mathbb{A}(q, v, t)$.

View of an Adversary. An adversary $\mathscr{A} \in \mathbb{A}(q, v)$ makes queries $(n_1, m_1), \ldots,$ (n_q, m_q) to an authenticator γ_K adaptively and obtain responses $t_1, \ldots, t_q$ respectively. It also makes $(n'_1, m'_1, t'_1), \ldots, (n'_v, m'_v, t'_v)$ to verifier Ver_K and obtain responses $b_1, \ldots, b_v \in \{0, 1\}$ respectively. The authentication and verification queries can be interleaved and adaptive. Note that all n_i's are distinct as we consider only nonce-respecting adversary, however, n_i''s are not necessarily distinct and can match with n_j values. We also assume that both q and v are fixed and hence non-random. We call the tuple

$$\big((n_1, m_1, t_1), \ldots, (n_q, m_q, t_q), (n_1', m_1', t_1', b_1), \ldots, (n_v', m_v', t_v', b_v)\big)$$

view and denote it as $\mathsf{view}(\mathscr{A}^{\gamma_K, \mathsf{Ver}_K})$ (which is a random variable induced by the randomness of $\mathscr{A}$ and the key of γ). Let

$$\mathscr{V} = (\mathscr{N} \times \mathscr{M} \times \mathfrak{B})^q \times (\mathscr{N} \times \mathscr{M} \times \mathfrak{B} \times \{0,1\})^v$$

be the set of all possible views. We say that a view $\tau \in \mathscr{V}$ is *realizable* if

$$\Pr_{\mathscr{A}, K}(\mathsf{view}(\mathscr{A}^{\gamma_K}) = \tau) > 0.$$

Authenticity Advantage. Following the notation of the view of an adversary as denoted above, we define the *authenticity advantage* of $\mathscr{A}$ as

$$\mathsf{Auth}_\gamma(\mathscr{A}) := \Pr(\exists i, b_i = 1).$$

In words, it is the probability that $\mathscr{A}$ submits a valid verification query which has not been obtained through a previous authentication query. In this paper, we are interested in the following maximum advantages for some families of adversaries:

$$\mathsf{Auth}_\gamma(q, v, t) = \max_{\mathscr{A} \in \mathbb{A}(q,v,t)} \mathsf{Auth}(\mathscr{A}), \quad \mathsf{Auth}_\gamma(q, v) = \max_{\mathscr{A} \in \mathbb{A}(q,v)} \mathsf{Auth}(\mathscr{A}).$$

So $\mathsf{Auth}_\gamma(q, v)$ is the maximum advantage for all information theoretic adversaries with the limitation that it can make at most q authentication queries and v verification queries. It is shown in [BGM04,Ber05a] that

$$\mathsf{Auth}_\gamma(q, v) \leq v \cdot \mathsf{Auth}_\gamma(q, 1). \tag{6}$$

Key-recovery Advantage. A full-key-recovery algorithm $\mathscr{A}$ is an adversary interacting with γ_K and Ver_K and finally it aims to recover the key K. Once the key K is recovered, the full system is broken and so one can forge as many times as it wishes. For some authenticators, we can do the forgeries when a partial key is recovered. Let $\mathscr{K} = \mathscr{K}' \times \mathscr{H}$ for some sets $\mathscr{K}'$ and $\mathscr{H}$. We call $\mathscr{H}$ hash-key space. Let $K = (K', H) \leftarrow_{\$} \mathscr{K}' \times \mathscr{H}$.

Definition 1 (key-recovery advantage). *A hash-key recovery algorithm (or we simply say that a key-recovery algorithm) $\mathscr{A}$ is an adversary interacting with γ_K and Ver_K and finally it returns $\mathbf{h}$, an element from $\mathscr{H}$. We define* key-recovery advantage *of $\mathscr{A}$ against γ as*

$$\mathsf{KR}_\gamma(\mathscr{A}) \ := \ \Pr(\mathscr{A}^{\gamma_K, \mathsf{Ver}_K} \Rightarrow \mathbf{h} \ \wedge \ \mathbf{h} = H).$$

The above probability is computed under randomness of $\mathscr{A}$ and $K = (K', H)$.

Similar to the maximum authenticity advantages, we define

$$\mathsf{KR}_\gamma(q,v,t) = \max_{\mathscr{A}\in\mathbb{A}(q,v,t)} \mathsf{KR}(\mathscr{A}), \quad \mathsf{KR}_\gamma(q,v) = \max_{\mathscr{A}\in\mathbb{A}(q,v)} \mathsf{KR}(\mathscr{A}).$$

When $v = 0$, we simply write $\mathsf{KR}_\gamma(q,t)$ and $\mathsf{KR}_\gamma(q)$. A relationship between key-recovery advantage and authenticity advantage is the following which can be proved easily $\mathsf{KR}_\gamma(q) \leq \mathsf{Auth}_\gamma(q,1)$.

Authenticated Encryption. In addition to nonce and message, an authenticated encryption γ' takes associated data and returns a ciphertext-tag pair. A verification algorithm $\mathsf{Ver}_{\gamma'}$ takes a tuple of nonce, associated data, ciphertext and tag, and determines whether it is valid (i.e. there is a message corresponding to this ciphertext and tag) or not. A forgery adversary $\mathscr{A}$ submits a fresh tuple (not obtained through encryption queries) of nonce, associated data, ciphertext and tag. Similar to authenticity advantage of an authenticator, authenticity of an adversary $\mathscr{A}$, denoted $\mathsf{Auth}_{\gamma'}(\mathscr{A})$ is the probability that it submits a fresh valid tuple.

Almost XOR Universal and Δ-Universal Hash Function. Let $\rho : \mathscr{K} \times \mathscr{M} \to \mathscr{B}$, for some additive commutative group $\mathscr{B}$. We denote the subtraction operation in the group as "$-$". We call ρ ϵ-ΔU (ϵ-Δ-universal) if for all $x \neq x' \in \mathscr{M}$ and $\delta \in \mathfrak{B}$,

$$\Pr(\rho_\kappa(x) - \rho_\kappa(x') = \delta) \leq \epsilon.$$

Here, the probability is taken under the uniform distribution $\kappa \leftarrow_\$ \mathscr{K}$. Note that $\epsilon \geq 1/N$ (since, for any fixed x, x', $\sum_\delta \Pr(\rho_\kappa(x) - \rho_\kappa(x') = \delta) = 1$). When $\mathscr{B} = \{0,1\}^b$ for some positive integer b and the addition is "$\oplus$" (bit-wise XOR operation), we call ρ ϵ-almost-xor-universal or ϵ-AXU hash function.

3 Known Analysis of WCS

We describe a real and an idealized version of WCS.

Definition 2 (WCS authenticator). *Let e_K be a blockcipher over a commutative group $\mathfrak{B}$ of size N with a key space $\mathscr{K}'$ and $\rho_\kappa : \mathscr{M} \to \mathfrak{B}$ is a keyed hash function with a key space $\mathscr{K}$. On an input $(n, M) \in \mathfrak{B} \times \mathscr{M}$, we define the output of WCS as*

$$\mathsf{WCS}_{K,\kappa}(n, M) = e_K(n) + \rho_\kappa(M). \tag{7}$$

Here, the pair (K, κ), called secret key, is sampled uniformly from $\mathscr{K}' \times \mathscr{K}$.

An idealized version of WCS is based on a uniform random permutation $\pi \leftarrow_\$ \mathsf{Perm}_{\mathscr{B}}$ (replacing the blockcipher e) and it is defined as

$$\mathsf{iWCS}_{\pi,\kappa}(n, m) = \pi(n) + \rho_\kappa(M) \tag{8}$$

where the hash key $\kappa \leftarrow_\$ \mathscr{K}$ (and independent of the random permutation).

WCS is a nonce based authenticator in which $\mathscr{n}$ is the nonce and M is a message. The most popular choice of $\mathfrak{B}$ is $\{0,1\}^n$ for some positive integer n and the blockcipher is AES [DR05,Pub01] (in which $n = 128$). The WCS and the ideal-WCS authenticators are computationally indistinguishable provided the underlying blockcipher e is a pseudorandom permutation. More formally, one can easily verify the following relations by using standard hybrid reduction;

$$\mathsf{Auth}_{\mathsf{WCS}}(q,v,t) \le \mathsf{Auth}_{\mathsf{iWCS}}(q,v) + \mathsf{Adv}_e^{\mathrm{prp}}(q+v, t+t'), \tag{9}$$

$$\mathsf{KR}_{\mathsf{WCS}}(q,v,t) \le \mathsf{KR}_{\mathsf{iWCS}}(q,v) + \mathsf{Adv}_e^{\mathrm{prp}}(q+v, t+t') \tag{10}$$

where t' is the time to compute $q+v$ executions of hash functions ρ_κ.

Polynomial Hash. Polynomial hash is a popular candidate for the keyed hash function in WCS (also used in the tag computation of GCM [MV04]). Here we assume that $\mathfrak{B}$ is a finite field of size N. Given any message $M := (m_1, \ldots, m_d) \in \mathfrak{B}^d$ and a hash key $\kappa \in \mathscr{K} = \mathfrak{B}$, we define the polynomial hash output as

$$\mathsf{Poly}_M(\kappa) := m_d \cdot \kappa + m_{d-1} \cdot \kappa^2 + \cdots + m_1 \cdot \kappa^d. \tag{11}$$

There are many variations of the above definition. Note that it is not an AXU hash function over variable-length messages (as appending zero blocks will not change the hash value). To incorporate variable length message, we sometimes preprocess the message before we run the polynomial hash. One such example is to pad a block which encodes the length of the message. One can simply prepend the constant block 1 to the message. These can be easily shown to be $\frac{\ell}{N}$-AXU over the padded message space $\mathscr{M} = \cup_{i=1}^{\le \ell} \mathfrak{B}^i$. In this paper we ignore the padding details and for simplicity, we work only on the padded messages. Whenever we use the polynomial hash in the WCS authenticator, we call its hash-key κ the polynomial-key.

Nonce Misuse. The input $\mathscr{n}$ is called nonce which should not repeat over different executions. Joux [Jou] and Handschuh and Preneel [HP08] exhibit attacks which recover the polynomial key the moment a nonce is repeated. For any two messages $M \neq M' \in \mathfrak{B}^d$,

$$\mathsf{WCS}_{K,\kappa}(\mathscr{n}, M) - \mathsf{WCS}_{K,\kappa}(\mathscr{n}, M') = \mathsf{Poly}_M(\kappa) - \mathsf{Poly}_{M'}(\kappa)$$

which is a nonzero polynomial in κ of degree at most d. By solving roots of the polynomial (which can be done efficiently by Berlekamps algorithm [Ber70] or the Cantor-Zassenhaus algorithm [CZ81]), we can recover the polynomial key. So it is an essential for a WCS authenticator to keep the nonce unique.

3.1 Shoup and Bernstein Bound on WCS

Let iWCS (we simply call it ideal-WCS) be based on a URP and an ϵ-AXU hash function ρ. When we replace the outputs of URP by uniform random values, Wegman and Carter had shown that (in [WC81]) the forgery advantage os less than $v\epsilon$ (independent of the number of authentication queries). So by applying the classical PRP-PRF switching lemma, we obtain

$$\mathsf{Auth}_{\mathsf{iWCS}}(q,v) \leq v \cdot \epsilon + \frac{(q+v)^2}{2N}. \tag{12}$$

So the classical bound is useless as q approaches $\sqrt{N}$ or as v approaches to ϵ^{-1}. In [Sho96] Shoup provided an alternative bound (which is improved and valid in a certain range of q). In particular, he proved

$$\mathsf{Auth}_{\mathsf{iWCS}}(q,v) \leq v \cdot \epsilon \cdot (1 - \frac{q^2\epsilon}{2})^{-1}. \tag{13}$$

The above bound is a form of multiplicative (instead of additive form of the classical bounds). Thus, the above bound is simplified as

$$\mathsf{Auth}_{\mathsf{iWCS}}(q,v) \leq 2\epsilon_{ver}(v) := 2v \cdot \epsilon, \quad \forall q \leq \sqrt{\epsilon^{-1}}. \tag{14}$$

So the ideal-WCS is secure up to $q \leq \sqrt{\epsilon^{-1}}$ queries. When $\epsilon = 1/N$, it says that authentication advantage is less $2v \cdot \epsilon$ for all $q \leq \sqrt{N}$. In other words, ideal-WCS is secure against birthday complexity adversaries. However, when the hash function is polynomial hash, Shoup's bound says that the ideal-WCS is secure up to $q \leq \sqrt{N/\ell}$. For example, when we authenticate messages of sizes about 2^{24} bytes (i.e. $\ell = 2^{20}$) using AES-based ideal-WCS, we can ensure security up to $q = 2^{54}$ queries. Like the classical bound, it also does not provide guarantees for long-term keys. Bernstein proved the following stronger bound for WCS.

Theorem 1 (Bernstein Bound([Ber05b])). *For all q and v*

$$\mathsf{Auth}_{\mathsf{iWCS}}(q,v) \leq B(q,v) := v \cdot \epsilon \cdot (1 - \frac{q}{N})^{-\frac{q+1}{2}}. \tag{15}$$

As a simple corollary (recovering the hash-key implies forgery), for all $v \geq 1$ we have

$$\mathsf{KR}_{\mathsf{iWCS}}(q,v) \leq \mathsf{B}(q,v), \quad \mathsf{KR}_{\mathsf{iWCS}}(q,0) \leq \mathsf{B}(q,1). \tag{16}$$

The key-recovery bound was not presented in [Ber05b], but it is a simple straightforward corollary from the fact that recovering hash-key implies forgery.

3.2 Interpretation of the Bernstein Bound

We now provide the interpretation of the bound which is crucial for understanding the optimality of ideal-WCS. As $1 - x \leq e^{-x}$, we have

$$\mathsf{B}(q,1) \geq \epsilon \cdot e^{\frac{q(q+1)}{2N}}.$$

Obviously, the Bernstein bound becomes more than one when $q(q+1)/2 \geq N \ln N$ (note that $\epsilon \geq N^{-1}$). So we assume that $q(q+1)/2 \leq N \ln N$. We denote $n = \log_2 N$. By Lemma 1, we have

$$\begin{aligned}\mathsf{B}(q,1) &\leq \epsilon \cdot e^{\frac{q(q+1)}{2N}(1+\frac{q}{N})} \\ &\leq \epsilon \cdot e^{\frac{q(q+1)}{2N}} \times e^{\frac{q \ln N}{N}} \\ &\leq \epsilon \cdot e^{\frac{q(q+1)}{2N}} \times (1 + \sqrt{\frac{2\ln^3 N}{N}}) \\ &\leq \epsilon \cdot e^{\frac{q(q+1)}{2N}} \times (1 + \frac{2n^{1.5}}{2^{n/2}}) = \epsilon \cdot e^{\frac{q(q+1)}{2N}} \times (1 + \mathsf{negl}(n))\end{aligned}$$

where $\mathsf{negl}(n) = \frac{2n^{1.5}}{2^{n/2}}$. Thus, $\mathsf{B}(q,v) = \Theta(v \cdot \epsilon \cdot e^{\frac{q(q+1)}{2N}})$. Let us introduce another parameter δ, called the tolerance level. We would now solve for q and v satisfying $\mathsf{B}(q,v) = \delta$ (or the inequality $\mathsf{B}(q,v) \geq \delta$) for any fixed constant δ. In other words, we want to get a lower bound of q and v to achieve at least δ authenticity advantage.

1. **Case 1** When $v \cdot \epsilon = \delta$ and $q \geq 1$ we have $\mathsf{B}(q,\ell) \geq \delta$. In other words, one needs to have sufficient verification attempts (and only one authentication query suffices) to have some significant advantage. We would like to note that even when $q = O(\sqrt{N})$, $\mathsf{B}(q,v) = \Theta(v \cdot \epsilon)$. So the advantages remain same up to some constant factor for all values of $q = O(\sqrt{N})$. In other words, we can not exploit the number of authentication queries within the birthday-bound complexity.
2. **Case 2**. $v \cdot \epsilon < \delta$. Let us assume that $v\epsilon/\delta = N^{\beta}$ for some positive real β. In this case one can easily verify that $q = \Omega(\sqrt{\delta N \log N})$ to achieve at least δ advantage. In other words, if $q = o(\sqrt{N \log N})$ and $v = o(\epsilon^{-1})$ then $\mathsf{B}(q,v) = o(1)$.

Tightness of the bound for the Case 1. We have seen that when $q = O(\sqrt{N})$, we have $\mathsf{Auth}_{\gamma}(q,v) = O(v \cdot \epsilon)$. In fact, it can be easily seen to be tight (explained below) when the hash function is the polynomial hash function $\mathsf{Poly}_M(\kappa)$.

Key Guess Forgery/Key-Recovery. Suppose WCS is based on the polynomial hash. Given a tag t of a known nonce-message pair (n, M) with $M \in \mathfrak{B}^{\ell}$, a simple guess attack works as follows. It selects a subset $\mathfrak{B}_1 \subseteq \mathfrak{B}$ of size ℓ and defines a message $M' \in \mathscr{M}$ and t' such that the following identity as a polynomial in x holds:

$$\mathsf{Poly}_{M'}(x) - t' = \mathsf{Poly}_M(x) - t + \prod_{\alpha \in \mathfrak{B}_1} (x - \alpha).$$

If $\kappa \in \mathfrak{B}_1$ then it is easy to verify that t' is the tag for the nonce-message pair (n, M'). The success probability of the forging attack is exactly ℓ/N. If the forgery is allowed to make v forging attempts, it first chooses v disjoint subsets $\mathfrak{B}_1, \ldots, \mathfrak{B}_v \subseteq \mathfrak{B}$, each of size ℓ. It then performs the above attack for each set $\mathfrak{B}_i$. The success probability of this forgery is exactly $v\ell/N$. The same attack was used to eliminate false keys systematically narrowing the set of potential polynomial keys and searching for weak keys.

Remark 1. The tightness of multiple-forgery advantage for WCS based on the polynomial hash can be extended similarly to all those hash functions ρ for which there exist $v+1$ distinct messages $M_1, \ldots, M_v, M$ and $c_1, \ldots, c_v \in \mathfrak{B}$ such that

$$\Pr(\rho_\kappa(M_i) = \rho_\kappa(M) + c_i,\ \forall i) = v\epsilon_\ell.$$

Why the Bernstein bound is better than the classical birthday bound? One may think the Bernstein bound is very close to the classical birthday bound of the form $q^2/2^n$ and they differ by simply logarithmic factor. However, these two bound are quite different in terms of the data or query limit in the usage of algorithms. We illustrate the difference through an example. Let $n = 128$, and the maximum advantage we can allow is 2^{-32}. Suppose a construction C has maximum forgery advantage $\frac{q^2}{n2^n}$ (a beyond birthday bound with logarithmic factor). Then we must have the constraint $q \leq 2^{51.5}$. Whereas, WCS can be used for at most 2^{64} queries. In other words, Bernstein bound actually provide much better life time of key than the classical birthday bound.

4 False-Key/True-Key Set: A tool for Key-Recovery and Forgery

Our main goal of the paper is to obtain hash-key-recovery attacks against WCS and GCM. Note that we do not recover the blockcipher key. So key-recovery advantage of whats follows would mean the probability to recover the hash-key only.

Query System and Transcript. A key-recovery (with no verification attempt) or a single forgery adversary has two components. The first component $\mathbf{Q}$, called **query system**, is same for both key-recovery and forgery. It makes queries to $\mathsf{WCS}_{K,\kappa}$ adaptively and obtains responses. Let $(\mathcal{n}_1, M_1), \ldots, (\mathcal{n}_q, M_q)$ be authentication queries with distinct $\mathcal{n}_i$ (i.e., the query system is nonce-respecting) and let t_i denote the response of ith query. Let $\tau := \tau(\mathbf{Q}) = ((\mathcal{n}_1, M_1, t_1), \ldots, (\mathcal{n}_q, M_q, t_q))$ denote the transcript.

Based on the transcript, a second component of forgery returns a fresh $(\mathcal{n}, M, t)$ (not in the transcript). If $\mathcal{n} \neq \mathcal{n}_i$ for all i then the forgery of WCS is essentially reduced to a forgery of the URP (in particular, forging the value of $\pi(\mathcal{n})$). Hence, the forgery advantage in that case is at most $1/(N-q)$. The most interesting case arises when $\mathcal{n} = \mathcal{n}_i$ for some i. Similarly, the second component of a key-recovery adversary returns an element $k \in \mathscr{K}$ (key space of the random function) based on the transcript τ obtained by the query system.

Definition 3 (False-key set [LP18]). *With each* $\tau = ((\mathscr{n}_1, M_1, t_1), \ldots, (\mathscr{n}_q, M_q, t_q))$, *we associate a set*

$$\mathscr{F}_\tau = \{x \in \mathscr{K} \mid \exists i \neq j,\ \rho_x(M_i) - \rho_x(M_j) + t_j - t_i = 0, M_i \neq M_j\}$$

and we call it the **false-key set***.*

Note that $\Pr(\kappa \in \mathscr{F}_\tau) = 0$ and so the term false-key set is justified. In other words, the true key κ can be any one of the elements from $\mathscr{T} := \mathscr{K} \backslash \mathscr{F}_\tau$, called the *true-key set.* Given a query system $\mathbf{Q}$, let us consider the key-recovery adversary which simply returns a random key $\mathbf{k}$ from the true-key set. Let us denote the key-recovery adversary as $\mathbf{Q}_{TK}$. The following useful bound is established in [LP18].

Lemma 2 ([LP18]). *Following the notation as described above we have*

$$\mathit{KR}_{\mathsf{WCS}}(\mathbf{Q}_{TK}) \geq \frac{1}{N - \mathbb{E}(|\mathscr{F}_{\tau(\mathbf{Q})}|)}. \tag{17}$$

Proof. Given a transcript τ, the probability that $\mathbf{k} = \kappa$ is exactly $\frac{1}{N - |\mathscr{F}_\tau|}$. Then,

$$\begin{aligned}\mathsf{KR}_{\mathsf{WCS}}(\mathbf{Q}_{TK}) &= \sum_\tau \Pr(\mathbf{k} = \kappa \mid \tau) \times \Pr(\tau) \\ &= \sum_\tau \frac{1}{N - |\mathscr{F}_\tau|} \Pr(\tau) \\ &= \mathbb{E}(\frac{1}{N - |\mathscr{F}_\tau|}). \end{aligned} \tag{18}$$

Here the expectation is taken under the randomness of the transcript. A transcript depends on the randomness of π, κ and the random coins of the query system. Note that the function $f(x) = \frac{1}{N-x}$ is convex in the interval $(0, N)$ and so by using Jensen inequality, we have $\mathsf{KR}_{\mathsf{WCS}}(\mathbf{Q}_{TK}) \geq \frac{1}{N - \mathbb{E}(|\mathscr{F}_{\tau(\mathbf{Q})}|)}$. □

In [LP18], it was also shown that $\mathbb{E}(|\mathscr{F}_{\tau(\mathbf{Q})}| \leq q(q+1)/2$ for all $q < M_\gamma$ where

$$M_\gamma = \max\{q : \min_{m^q, t^q} |\mathscr{T}_\tau| \geq \ell\}$$

where τ denotes the transcript $((m_1, t_1), \ldots, (m_q, t_q))$ (ignoring nonce values as these are redundant). A straight forward estimation of M_γ is $2^{n/2}/\sqrt{\ell}$. Here we give a very simple proof of the above bound for all q.

Lemma 3. *For all* q, $\mathbb{E}(|\mathscr{F}_{\tau(\mathbf{Q})}| \leq q(q+1)/2$.

Proof. We define an indicator random variable I_x which takes value 1 if and only if there exists $i \neq j$ such that $\rho_x(M_i) - \rho_x(M_j) + t_j - t_i = 0$. We observe that $|\mathscr{F}_\tau| = \sum_{x \in \mathscr{K}} \mathsf{I}_x$.

Let us denote $\pi(n_i)$ as V_i. Note that for all i, $t_i = \mathsf{V}_i + \rho_\kappa(M_i)$. Now, $\mathbb{E}(|\mathscr{F}_\tau|) = \sum_{x \in \mathscr{K}} \mathbb{E}(\mathsf{I}_x)$. We write $p_x = \mathbb{E}(\mathsf{I}_x)$ which is nothing but the probability that there exists $i \neq j$ such that $\mathsf{V}_i - \mathsf{V}_j = \rho_x(M_i) - \rho_\kappa(M_i) + \rho_x(M_i) + \rho_\kappa(M_i)$. By using the union bound we have $p_x \leq \binom{q}{2}/(N-1)$. So

$$\begin{aligned}\mathbb{E}(|\mathscr{F}_\tau|) &\leq \frac{Nq(q-1)}{2(N-1)} \\ &\leq \frac{q(q-1)}{2} + \frac{q(q-1)}{2(N-1)}\end{aligned}$$

We can clearly assume that $q < N$ and so by using simple inequality the lemma follows. □

True-key Set. Instead of the false-key set we focus on the true key set. The set $\mathscr{T}_\tau := \mathscr{K} \setminus \mathscr{F}_\tau$ is called the true-key set. In terms of the true-key set, we can write $\mathsf{KR}_{\mathsf{WCS}}(\mathbf{Q}_{TK}) = \mathbb{E}(\frac{1}{|\mathscr{T}_{\tau(\mathbf{Q})}|})$. Let $\pi(n_i) = \mathsf{V}_i$ and $a_{i,x} := a_{i,x}(\kappa) := \rho_\kappa(M_i) - \rho_x(M_i)$. We can equivalently define the true-key set as

$$\begin{aligned}\mathscr{T}_\tau &= \{x \in \mathscr{K} \mid t_1 - \rho_x(M_1), \ldots, t_q - \rho_x(M_k) \text{ are distinct}\} \\ &= \{x \in \mathscr{K} \mid \mathsf{V}_1 + a_{1,x}, \ldots, \mathsf{V}_q + a_{q,x} \text{ are distinct}\}. \qquad (19)\end{aligned}$$

Now we define an indicator random variable I_x as follows:

$$\mathsf{I}_x = \begin{cases} 1, \text{if } \mathsf{V}_1 + a_{1,x}, \ldots, \mathsf{V}_q + a_{q,x} \text{ are distinct} \\ 0, \text{otherwise} \end{cases}$$

Let p_x denote the probability that $\mathsf{V}_1 + a_{1,x}, \ldots, \mathsf{V}_q + a_{q,x}$ are distinct. So,

$$\mathbb{E}(|\mathscr{T}_\tau|) = \sum_x \mathbb{E}(\mathsf{I}_x) = \sum_{x \in \mathscr{K}} p_x.$$

When we want to minimize the expected value of the size of the true-key set, we need to upper bound the probability p_x for all x. We use this idea while we analyze our key-recovery attacks.

5 Key-Recovery Security Attacks of WCS

5.1 A Chosen-Plaintext Key-Recovery Attack

In this section we provide a chosen-plaintext attack against any WCS based on any blockcipher and a keyed hash function which satisfies a reasonable assumption, called differential regular. This property is satisfied by the polynomial hash. A function $f : \mathscr{M} \to \mathfrak{B}$ is called *regular* if $X \leftarrow_{\$} \mathscr{M} \Rightarrow f(X) \leftarrow_{\$} \mathfrak{B}$. Now we define a special type of keyed hash functions.

Definition 4. *A keyed hash function $\rho_\kappa : \mathscr{K} \to \mathfrak{B}$ is called* differential regular *if for all distinct $x, k \in \mathscr{K}$, the function mapping $M \in \mathscr{M}$ to $\rho_k(M) - \rho_x(M)$ is regular.*

The polynomial hash is clearly differential regular. For example, when the message space is $\mathfrak{B}$ and $\kappa \neq x$, the function mapping $m \in \mathfrak{B}$ to $\rho_\kappa(m) - \rho_x(m) = m(\kappa - x)$ is regular.

Theorem 2. *Suppose* WCS *is based on a blockcipher and a keyed differential regular hash function* ρ*. Then,*

$$\mathit{KR}_{\mathsf{WCS}}(q) \geq \frac{1}{1 + N' e^{-\frac{q(q-1)}{2N}}} \tag{20}$$

where $N' = |\mathscr{K}|$ *(size of the hash-key space). In particular, when* $q(q-1) = 2N \log N'$ *we have* $\mathit{KR}_{\mathsf{WCS}}(q, \ell) \geq 1/2$*.*

Interpretation of the result. When $N' = N$ (key size is same as the block size), we can achieve 0.5 key-recovery advantage after making roughly $\sqrt{2N \log N}$ authentication queries . If $N' = N^c$ for some $c > 1$ (the hash-key size is larger than the block size) we need roughly $\sqrt{2cN \log N}$ (which is a constant multiple of the number queries required for hash-key space of size N) authentication queries.

Proof. Suppose $\mathsf{WCS} := \mathsf{WCS}_{K,\kappa}$ is the WCS authenticator based on a blockcipher e_K and a keyed differential regular hash function ρ_κ. We describe our key-recovery attack[1] $\mathscr{A}$ as follows:

1. Choose q messages $M_1, \ldots, M_q \leftarrow_{\$} \mathscr{M}$ and make authentication queries (n_i, M_i), $i \in [q]$ for distinct nonces n_i's.
2. Let $t_1, \ldots, t_q$ be the corresponding responses.
3. Construct the true-key set
$$\mathscr{T}_\tau = \{k \mid (t_i - \rho_k(M_i))'s \text{ are distinct}\}.$$
4. Return a key $\mathbf{k} \leftarrow_{\$} \mathscr{T}_\tau$.

Here, $\tau = ((n_1, M_1, t_1), \ldots, (n_q, M_q, t_q))$ is the transcript of the adversary $\mathscr{A}$. We also note that $\Pr(\kappa \in \mathscr{T}_\tau) = 1$ and so we have seen that $\mathsf{KR}_{\mathsf{WCS}}(\mathscr{A}) = \mathbb{E}(\frac{1}{|\mathscr{T}_\tau|})$. Here the expectation is taken under randomness of transcript. The randomness of a transcript depends on the randomness of K, κ and the messages M_i. By using Jensen inequality, we have

$$\mathsf{KR}_{\mathsf{WCS}}(\mathscr{A}) \;\geq\; \frac{1}{\mathbb{E}(|\mathscr{T}_\tau|)}.$$

We will now provide an upper bound of $\mathbb{E}(|\mathscr{T}_\tau|)$. In fact, we will provide an upper bound on the conditional expectation after conditioning the blockcipher key K and hash-key κ. Note that $t_i = e_K(n_i) + \rho_\kappa(M_i)$ and hence the true-key set is the set of all x for which $R_{i,x} := e_K(n_i) + \rho_\kappa(M_i) - \rho_x(M_i)$ are distinct for all $i \in [q]$.

[1] We note that the similar attack is considered in [LP18] where the messages are fixed and distinct. However in their attacks the analysis is done for $q \leq 2^{n/2}$ whereas, we analyze for all q.

Claim. Given K and κ, the conditional distributions of $R_{i,x}$'s are uniform and independent over $\mathfrak{B}$, whenever $x \neq \kappa$.

Proof of the Claim. Once we fix K and κ, for every $x \neq \kappa$, $\rho_\kappa(M_i) - \rho_x(M_i)$ is uniformly distributed (as ρ is differentially regular). So $e_K(\mathscr{n}_i) + \rho_\kappa(M_i) - \rho_x(M_i)$'s are also uniformly and independently distributed since $e_K(\mathscr{n}_i)$'s are some constants nd M_i's are independently sampled.

Now we write $|\mathscr{T}_\tau| = \sum_x \mathsf{I}_x$ where I_x is the indicator random variable which takes values 1 if and only if $R_{i,x}$ are distinct for all i. Note that $R_{i,x}$ are distinct for all i has probability exactly $\prod_{i=1}^{q-1}(1 - \frac{i}{N})$ (same as the birthday paradox bound). As $1 - x \leq e^{-x}$ for all x, we have $\mathbb{E}(\mathsf{I}_x) = \Pr(\mathsf{I}_x = 1) \leq e^{-\frac{q(q-1)}{N}}$. So,

$$\begin{aligned}\mathbb{E}(|\mathscr{T}| \mid K, \kappa) &= 1 + \sum_{x \neq \kappa} \mathbb{E}(\mathsf{I}_x) \\ &\leq 1 + (N' - 1)e^{-\frac{q(q-1)}{2N}}.\end{aligned}$$

This bound is true for all K and κ and hence $\mathbb{E}(|\mathscr{T}|) \leq 1 + (N' - 1)e^{-\frac{q(q-1)}{2N}}$. This completes the proof. □

5.2 Known-Plaintext Attack

Now we show a known-plaintext attack for polynomial-based hash in which we do not assume *any randomness of messages*. So our previous analysis does not work in this case. We first describe a combinatorial result which would be used in our known plaintext key-recovery advantage analysis.

Lemma 4. *Let* $\mathsf{V}_1, \ldots, \mathsf{V}_q$ *be a uniform without replacement sample from* $\mathfrak{B}$ *and* $a_1, \ldots, a_q \in \mathfrak{B}$ *be some distinct elements, for some* $q \leq N/6$. *Then,*

$$p_x := \Pr(\mathsf{V}_1 + a_1, \ldots, \mathsf{V}_q + a_q \textit{ are distinct}) \leq e^{-q^2/4N}.$$

Proof. For $1 \leq \alpha \leq q$, let h_α denote the number of tuples $v^\alpha = (v_1, \ldots, v_\alpha)$ such that $v_1 + a_1, \ldots, v_\alpha + a_\alpha$ are distinct. Clearly, $h_1 = N$. Now we establish some recurrence relation between $h_{\alpha+1}$ and h_α. We also abuse the term h_α to represent the set of solutions $v^\alpha = (v_1, \ldots, v_\alpha)$ such that $v_1 + a_1, \ldots, v_\alpha + a_\alpha$ are distinct.

Given any solution v^α (among the h_α solutions), we want to estimate the number of ways we can choose $v_{\alpha+1}$. Note that

$$v_{\alpha+1} \notin \{v_1, \ldots, v_\alpha\} \cup \{v_1 + a_1 - a_{\alpha+1}, \ldots, v_\alpha + a_\alpha - a_{\alpha+1}\}.$$

Let $S_\alpha := \{v_1 + a_1 - a_{\alpha+1}, \ldots, v_\alpha + a_\alpha - a_{\alpha+1}\}$. As v^α is one solution from h_α, the size of the set S_α is exactly α. Note that if $v_i = v_j + a_j - a_\alpha$ then j must be different from i as a_i's are distinct. For any $i \neq j \leq \alpha$, we denote $h'_\alpha(i, j)$ be the number of v^α such that $v_1 + a_1, \ldots, v_\alpha + a_\alpha$ are distinct and $v_i + a_i = v_j + a_j$ (once again we abuse this term to represent the set of solutions). So by the principle of inclusion and exclusion, we write

$$h_{\alpha+1} = (N-2\alpha)h_\alpha + \sum_{i\neq j} h'_\alpha(i,j).$$

Claim. For all $i \neq j \leq \alpha$, $h'_\alpha(i,j) \leq \frac{h_\alpha}{N-2\alpha}$.

Proof of claim. Let us assume $i = \alpha$ and $j = \alpha - 1$. The proof for the other cases will be similar. Any solution for $h'_\alpha(\alpha, \alpha-1)$ is a solution for $h_{\alpha-1}$ and $v_\alpha = v_{\alpha-1} + a_{\alpha-1} - a_\alpha$. However, all solutions corresponding to h_α satisfy the solution corresponding to $h_{\alpha-1}$ and v_α is not a member of a set of size at most 2α. So the claim follows.

Now, we have

$$h_{\alpha+1} \leq h_\alpha(N-2\alpha) + \alpha(\alpha-1)h_\alpha/(N-2\alpha).$$

In other words,

$$\frac{h_{\alpha+1}}{h_\alpha} \leq (N-2\alpha) + \frac{\alpha(\alpha-1)}{N-2\alpha} = \frac{N^2 - 4\alpha N + 5\alpha^2 - \alpha}{N-2\alpha}.$$

Now we simplify the upper bound as follows.

$$\begin{aligned}\frac{N^2 - 4\alpha N + 5\alpha^2 - \alpha}{N-2\alpha} &= (N-\alpha)\frac{N^2 - 4\alpha N + 5\alpha^2 - \alpha}{N^2 - 3\alpha N + 2\alpha^2} \\ &= (N-\alpha)(1 - \frac{\alpha N + \alpha - 3\alpha^2}{N^2 - 3\alpha N + 2\alpha^2}) \\ &\leq (N-\alpha)(1 - \frac{\alpha N + \alpha - 3\alpha^2}{N^2}) \\ &\leq (N-\alpha)(1 - \frac{\alpha}{2N})\end{aligned}$$

provided $\alpha(N+1) - 3\alpha^2 \geq \alpha N/2$, equivalently $(N+2) \geq 6\alpha$. So for all $\alpha \leq q \leq N/6$ we have

$$\frac{h_{\alpha+1}}{h_\alpha} \leq (N-\alpha)(1 - \frac{\alpha}{2N}) \leq (N-\alpha)e^{-\frac{\alpha}{2N}}.$$

By multiplying the ratio for all $1 \leq \alpha \leq q-1$ and the fact that $h_1 = N$, we have $h_q \leq (N)_q e^{-q^2/4N}$. The lemma follows from the definition that $p_x = \frac{h_q}{(N)_q}$. □

Now we consider the key-recovery adversary considered in [LP18]. However, they considered transcripts with $\sqrt{N}$ queries and were able to show a key-recovery advantage about $1.3/N$. However, we analyze it for all queries q and the key-recovery advantage can reach to $1/2$ for $q = O(\sqrt{N \log N})$.

Theorem 3. *Suppose $m_1, \ldots, m_q \in \mathfrak{B}$ be distinct messages and $\mathcal{n}_1, \ldots, \mathcal{n}_q$ be distinct nonces. Let $t_i = \mathsf{WCS}_{\pi,\kappa}(\mathcal{n}_i, m_i)$ where ρ_κ is the polynomial hash. Then, there is an algorithm $\mathscr{A}$ which recovers the hash-key κ with probability at least*

$$\frac{1}{1 + (N-1)e^{-\frac{q^2}{4N}}}.$$

So when $q = \sqrt{4N \log N}$, the key-recovery advantage is at least $\frac{1}{2}$.

Proof. We denote $\pi(\mathcal{n}_i) = \mathsf{V}_i$. So $\mathsf{V}_1, \ldots, \mathsf{V}_q$ forms a without replacement random sample from $\mathfrak{B}$. We write $t_i = \mathsf{V}_i + \rho_\kappa(m_i) = \mathsf{V}_i + \kappa \cdot m_i$. As before we define the true-key set as

$$\mathscr{T} := \{x \in \mathfrak{B} \mid t_1 - x \cdot m_1, \ldots, t_q - x \cdot m_q \text{ are distinct}\}.$$

Clearly $\kappa \in \mathscr{T}$. Let us fix $x \neq \kappa$ and denote $a_i = (\kappa - x) \cdot m_i$. Note that a_i's are distinct. So given a hash-key κ, we write the size of true-key set $|\mathscr{T}|$ as the sum of the indicator random variables as follows: $|\mathscr{T}| = 1 + \sum_{x \neq \kappa} \mathsf{I}_x$ where I_x takes value 1 if and only if $\mathsf{V}_1 + a_1, \ldots, \mathsf{V}_q + a_q$ are distinct. So,

$$\begin{aligned} \mathbb{E}(|\mathscr{T}| \mid \kappa) &= 1 + \sum_{x \neq \kappa} \mathbb{E}(\mathsf{I}_x) \\ &= 1 + \sum_{x \neq \kappa} p_x \end{aligned}$$

where

$$p_x := \Pr(\mathsf{V}_1 + a_1, \ldots, \mathsf{V}_q + a_q \text{ are distinct}).$$

By Lemma 4, we know that $p_x \leq e^{-\frac{q^2}{4N}}$ and hence $\mathbb{E}(|\mathscr{T}| \mid \kappa) \leq 1 + (N-1)e^{-\frac{q^2}{4N}}$. This is true for all hash-keys κ and hence we have $\mathbb{E}(|\mathscr{T}|) \leq 1 + (N-1)e^{-\frac{q^2}{4N}}$. This completes the proof. □

6 Key-Recovery Security Analysis of GCM

Definition of GCM. We briefly describe how GCM works. We refer the reader to see [MV04] for details. Here $\mathfrak{B} = \{0,1\}^n$ (with $n = 128$) Let e_K be a blockcipher as before. We derive hash-key as $\kappa = e_K(0^n)$. Given a message $(m_1, \ldots, m_\ell) \in \mathfrak{B}^\ell$ and a nonce $\mathcal{n} \in \{0,1\}^{b-s}$ for some s, we define the ciphertext as

$$c_i = \mathsf{V}'_i \oplus m_i, \; i \in [\ell], \mathsf{V}'_i = e_K(\mathcal{n} \| \langle i+1 \rangle)$$

where $\langle i \rangle$ represents s-bit encoding of the integer i. Finally, the tag is computed as xor of $\mathsf{V} := e_K(\mathcal{n} \| \langle 1 \rangle)$ and the output of the polynomial hash of the associated data and the ciphertext with length encoding. So, $t = \mathsf{V} \oplus c_0 \kappa \oplus c_1 \kappa^2 \oplus \cdots$ where c_0 is the block which encodes the length of message (same as the ciphertext) and the associated data.

In other words, the tag is computed as a WCS authentication over the ciphertext with the hash-key derived from the blockcipher. So, one can have a similar key-recovery attack as stated in Theorem 2 which requires roughly $\sqrt{n} \times 2^{n/2}$ authentication queries. More precisely, after making 2^{68} authentication queries with the first message block random we can recover $e_K(0)$ with probability at least 1/2. Note that the ciphertext blocks are uniformly distributed as it is an

XOR of message blocks and some blockcipher outputs independent of the message blocks. Now we show a more efficient algorithm $\mathscr{B}$ which utilize the length of messages as described below.

1. Choose q messages $M_1, \ldots, M_q \leftarrow_{\$} \mathfrak{B}^{\ell}$ and fix some associated data $A_i = A$. Make authentication queries $(\mathcal{n}_i, M_i, A)$, $i \in [q]$ for distinct nonces $\mathcal{n}_i$'s.
2. Let $(C_1, t_1), \ldots, (C_q, t_q)$ be the corresponding responses.
3. Let $M_i = m_{i,1} \| \cdots m_{i,\ell}$ and $C_i = c_{i,1} \| \cdots c_{i,\ell}$ where $\mathcal{n}_{i,j}, c_{i,j} \in \mathfrak{B}$. Construct a set
$$\mathscr{V}' = \{\mathsf{V}'_{i,j} := m_{i,j} \oplus c_{i,j} \mid i \in [q], j \in [\ell]\}$$
4. Construct the true-key set
$$\mathscr{T} = \{k \in \mathfrak{B} \mid t_i \oplus \rho_k(A, C_i) \notin \mathscr{V}' \ \forall i \in [q]\}.$$
5. Return a key $\mathbf{k} \leftarrow_{\$} \mathscr{T}$.

Remark 2. One may incorporate the relation that $t_i \oplus \rho_k(A, C_i)$'s are distinct while defining the true-key set. We can gain some complexity up to some small constant factor. For the sake of simplicity of the analysis and the attack, we keep the basic simple attack algorithm as described above.

Theorem 4. *Let $N = 2^n$ where n is the block size of the blockcipher used in GCM.*

$$\mathit{KR}_{\mathsf{GCM}}(q, \ell) \geq \frac{1}{1 + Ne^{-\frac{\ell q^2}{N}}} \tag{21}$$

In particular, when $\ell q^2 = N \log N$ we have $\mathit{KR}_{\mathsf{GCM}}(q, \ell) \geq 1/2$.

For example, when $n = 128, \ell = 2^{15}$ we now need $q = 2^{60}$ encryption queries to recover $\kappa = e_K(0)$. Once we recover κ, we can forge as many times as required. Moreover, one can define a universal forgery (for any chosen message and associated data but not the nonce).

Proof. From the permutation nature of the blockcipher, it is easy to see that $e_K(0) \in \mathscr{T}$ as defined in the algorithm. So, as before

$$\mathsf{KR}_{\mathsf{GCM}}(\mathscr{A}) \ \geq \ \frac{1}{\mathbb{E}(|\mathscr{T}|)}.$$

We will now provide an upper bound of $\mathbb{E}(|\mathscr{T}_{\tau}|)$. In fact, we will provide an upper bound of the conditional expectation after conditioning the blockcipher key K (so that all blockcipher outputs are fixed). Since message blocks are uniformly distributed, the ciphertext blocks are also uniformly distributed (due to one-time padding). This proves that after conditioning the blockcipher key K,

$$R_{1,x} := t_1 \oplus \rho_x(A, C_1), \ldots, R_{q,x} := t_q \oplus \rho_x(A, C_q) \leftarrow_{\$} \mathfrak{B}.$$

Now, we define an indicator random variable I_x to be one if $R_{i,x} \notin \mathscr{V}'$ for all $i \in [q]$ and 0 otherwise. So, from the definition of $\mathscr{T}$, it is easy to see that

$$|\mathscr{T}| = 1 + \sum_{x \neq \kappa} \mathsf{I}_x.$$

Condition a blockcipher key K (and hence the hash-key $\kappa = e_K(0^n)$ is fixed), and fix some $x \neq \kappa$. Now,

$$\begin{aligned}\mathbb{E}(\mathsf{I}_x \mid K) &= \Pr(\mathsf{I}_x = 1 \mid K)\\ &= \prod_{i=1}^{q}(\frac{N - \ell q}{N})\\ &\leq e^{-\frac{\ell q^2}{N}}.\end{aligned}$$

When $x = \kappa$, clearly, $\mathsf{I}_x = 1$. So,

$$\begin{aligned}\mathbb{E}(|\mathscr{T}| \mid K) &= 1 + \sum_{x \neq \kappa} \mathbb{E}(\mathsf{I}_x)\\ &\leq 1 + Ne^{-\frac{\ell q^2}{N}}.\end{aligned}$$

This bound is true for all blockcipher keys K and hence $\mathbb{E}(|\mathscr{T}|) \leq 1 + Ne^{-\frac{\ell q^2}{N}}$. This completes the proof. □

We show that when $\ell q^2 = \sqrt{2NlogN}$, we achieve some significant forgery advantage. Bernstein proved an upper bound of the forgery advantage for WCS. A similar proof is also applicable for GCM. In particular, we show that forgery advantage of GCM for single forging attempt is at most $\frac{\ell}{N} \cdot O(e^{\frac{4\ell q^2}{N}})$. So when we consider v forging attempts, the maximum forging advantage is at most $v \cdot \frac{\ell}{N} \cdot O(e^{\frac{4\sigma q}{N}})$. So our forgery algorithm (which is induced from the key-recovery algorithm) is also optimum for GCM. We denote the maximum forging advantage as $\mathsf{Auth}_{\mathsf{GCM}}(q, v, \sigma, \ell)$ where σ denotes the total number of blocks present in message and associated data in all q encryption queries, and ℓ denotes the number of blocks present in associated data and message or ciphertext for the largest query among all encryption and verification attempts. A similar result has been stated in Appendix C of [IOM12a] (full version of [IOM12b]).

Theorem 5. *Let* GCM *be based on the ideal n-bit random permutation π. Then, for all q, v and ℓ,*

$$\mathsf{Auth}_{\mathsf{GCM}}(q, v, \sigma, \ell) = v \cdot \frac{\ell}{N} \cdot O(e^{\frac{4\sigma q}{N}}) \tag{22}$$

Proof. We use x^q to denote a q tuple $(x_1, \ldots, x_q)$. For positive integers $r \leq m$, we write $(m)_r := m(m-1)\cdots(m-r+1)$. Bernstein proved an upper bound of the interpolation probability of a random permutation π as described below. Let $\delta_N(q) = (1 - (q-1)/N)^{-q/2}$.

Theorem 4.2 in [Ber05b] showed that for all $0 < r \le N$,

$$\frac{1}{(N)_r} \le \frac{\delta_N(r)}{N^r} = \frac{(1-\frac{r-1}{N})^{-\frac{r}{2}}}{N^r}. \tag{23}$$

Note that for any r distinct inputs $x_1, \ldots, x_r$ and outputs $y_1, \ldots, y_r$ the probability that $\pi(x_1) = y_1, \ldots, \pi(x_r) = y_r$ is exactly $\frac{1}{(N)_r}$. We use this result to prove our result.

Without loss of generality we assume that $\mathscr{A}$ is deterministic and the nonce in the forging attempt is one of the nonce in the encryption queries (since otherwise the bound can be shown to be smaller that what we claimed). We also assume that adversary makes single forging attempt (i.e. $v = 1$). Let $\mathscr{A}$ make queries (n_i, m_i, a_i) and obtain response (c_i, t_i) where $m_i = (m_i[1], \ldots, m_i[\ell_i])$, $a_i = (a_i[1], \ldots, a_i[\ell'_i])$ and $c_i = (c_i[1], \ldots, c_i[\ell_i])$ and let $\sigma = \sum_{i=1}^{q}(\ell_i + \ell'_i)$ (total number of blocks in all queries). We call $(n^q, m^q, a^q, c^q, t^q)$ transcript (for encryption queries).

Let (n^*, a^*, c^*, t^*) denote the forging attempt where c^* contains ℓ^* blocks. According to our simplification, let $n^* = n_i$ for some i. So c^q, t^q determine the whole transcript including the forging attempt. Let us write $z_i = m_i \oplus c_i$. It is also easy to see that t^q, z^q also determine the transcript.

Let F denote the forgery event, $n^* = n_i$ and $d = t^* \oplus t_i$. Moreover, for every k (a candidate of hash key), we set $y_i(k) = t_i \oplus \rho_k(a_i \| c_i)$. Now, $\Pr(F) = \Pr(\rho_\kappa(a_i \| c_i) \oplus \rho_\kappa(a^* \| m^*) = d)$. This can be written as the following sum

$$\begin{aligned}\Pr(F) &= \sum_{t^q, z^q} \Pr(\rho_\kappa(a_i \| c_i) \oplus \rho_\kappa(a^* \| c^*) = d \ \wedge\ \mathscr{A} \text{ obtains } z^q, t^q) \\ &= \sum_{t^q, z^q} \Pr(\rho_\kappa(a_i \| c_i) \oplus \rho_\kappa(a^* \| c^*) = d \ \wedge\ E(\kappa))\end{aligned}$$

where the sum is taken over all t^q and all those z^q for which all blocks of z_i's are distinct. The event $E(\kappa)$ denotes that $\pi(n_1 \| \langle 1 \rangle) = y_1(\kappa), \ldots, \pi(n_q \| \langle 1 \rangle) = y_q(\kappa)$ and $\pi(n_i \| \langle j \rangle) = z_i[j]$ for all $1 \le i \le q$, $1 \le j \le \ell_i$.

Now conditioning on any $\pi(0) := \kappa = k$ such that $\rho_\kappa(a_i \| c_i) \oplus \rho_\kappa(a^* \| c^*) = d$ (there are at most $\max\{\ell_i + \ell'_i, \ell^* + \ell^{'*}\} + 1 \le \ell$ choices of k), the conditional probability is reduced to $\Pr(E(k))$ which should be $\frac{1}{(N-1)_{q+\sigma}}$ (note that $\pi(0)$ is conditioned and the event $E(k)$ defines $q + \sigma$ many inputs-outputs of π). So,

$$\begin{aligned}\Pr(F) &= \sum_{t^q, z^q} \Pr(\rho_\kappa(a_i \| c_i) \oplus \rho_\kappa(a^* \| c^*) = d \ \wedge\ E(\kappa)) \\ &= \sum_{t^q, z^q} \Pr(\rho_\kappa(a_i \| c_i) \oplus \rho_\kappa(a^* \| c^*) = d) \times \Pr(E(\kappa) \mid \rho_\kappa(a_i \| c_i) \oplus \rho_\kappa(a^* \| c^*) = d) \\ &\le \sum_{t^q, z^q} \frac{\ell}{N} \cdot \frac{1}{(N-1)_{q+\sigma}} \\ &= \frac{\ell \cdot (N)_\sigma \cdot N^q}{(N)_{q+\sigma+1}}\end{aligned}$$

Note that in the above sum, we vary all distinct values of z blocks and so there are $(N)_\sigma$ such choices of z. Now it remains to simplify the bound.

$$\begin{aligned}
\Pr(F) &\leq \frac{\ell \cdot (N)_\sigma \cdot N^q}{(N)_{q+\sigma+1}} \\
&= \frac{\ell \cdot N^q}{(N-\sigma)_{q+1}} \\
&\leq_{(a)} \frac{\ell \cdot N^q}{(N-\sigma)^{q+1}} \delta_{N-\sigma}(q+1) \\
&= \frac{\ell}{N} \times (1 - \frac{\sigma}{N})^{-(q+1)} \times (1 - \frac{q}{N-\sigma})^{-(q+1)/2}.
\end{aligned}$$

The inequality (a) follows from Eq. 23 with N as $N-\sigma$. This provides the forgery bound for GCM (without using the privacy bound for GCM). For the values of q, ℓ and σ of our interest, we can assume that $\sigma \leq N/2$ and $1 - x = \Theta(e^{-x})$ (Lemma 1). So we can rewrite the upper bound of the forgery advantage of GCM as

$$\frac{\ell}{N} \cdot O(e^{\frac{\sigma(q+1)+q(q+1)}{N}}) = \frac{\ell}{N} \cdot O(e^{\frac{(\sigma+q)(q+1)}{N}}) = \frac{\ell}{N} \cdot O(e^{\frac{4\sigma q}{N}}).$$

The proof for v forging attempts simply follows by multiplying above bound by v. □

Remark 3. The above bound says that, as long as $q\sigma = o(N \log N)$, the forgery advantage is negligible and hence we need $q\sigma$ to be in the order of $N \log N$ to get non-negligible advantage. Along with our forgery adversary on GCM, we have shown the above forgery bound of GCM is indeed tight.

7 Conclusion

In this paper we describe key-recover attacks on WCS and GCM. The query complexity of the attack match with the Bernstein bound and hence we prove the tightness of Bernstein bound. Although the query complexity of our attacks are optimal, a straightforward implementation would require $O(N)$ memory and time complexity. Very recently Leurent and Sibleyras [LS18] demonstrated attacks for WCS. They have described a method to recover hash key of WCS (and counter mode encryption) with $O(2^{2n/3})$ query and time complexity. However, the success probability analysis of their attack is heuristic. It would be an interesting problem to see whether our concrete analysis can be adapted to their attacks.

Acknowledgments. The author would like to thank Anirban Ghatak, Eik List, Subhamoy Maitra, Bart Mennink and anonymous reviewers for their useful comments. The author would also like to thank Atul Luykx for the initial discussion of the paper. This work is supported by R. C. Bose Center for Cryptology and Security.

References

ABBT15. Mohamed Ahmed Abdelraheem, Peter Beelen, Andrey Bogdanov, and Elmar Tischhauser. Twisted polynomials and forgery attacks on gcm. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 762–786. Springer, 2015.

AY12. Kazumaro Aoki and Kan Yasuda. The security and performance of gcm when short multiplications are used instead. In *International Conference on Information Security and Cryptology*, pages 225–245. Springer, 2012.

BBS86. Lenore Blum, Manuel Blum, and Mike Shub. A simple unpredictable pseudorandom number generator. *SIAM Journal on computing*, 15(2):364–383, 1986.

Ber70. Elwyn R Berlekamp. Factoring polynomials over large finite fields. *Mathematics of computation*, 24(111):713–735, 1970.

Ber05a. Daniel J. Bernstein. The poly1305-aes message-authentication code. In *Fast Software Encryption, FSE 2005*, pages 32–49, 2005.

Ber05b. Daniel J Bernstein. Stronger security bounds for wegman-carter-shoup authenticators. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 164–180. Springer, 2005.

Ber07. Daniel J Bernstein. Polynomial evaluation and message authentication. *URL: http://cr. yp. to/papers. html# pema. ID b1ef3f2d385a926123e 1517392e20f8c. Citations in this document*, 2, 2007.

BGM04. Mihir Bellare, Oded Goldreich, and Anton Mityagin. The power of verification queries in message authentication and authenticated encryption. *IACR Cryptology ePrint Archive*, 2004:309, 2004.

BHK$^+$99. John Black, Shai Halevi, Hugo Krawczyk, Ted Krovetz, and Phillip Rogaway. Umac: Fast and secure message authentication. In *Annual International Cryptology Conference*, pages 216–233. Springer, 1999.

BJKS94. Jürgen Bierbrauer, Thomas Johansson, Gregory Kabatianskii, and Ben Smeets. On families of hash functions via geometric codes and concatenation. pages 331–342, 1994. URL: `http://cr.yp.to/bib/entries.html#1994/bierbrauer`.

Bra83. Gilles Brassard. On computationally secure authentication tags requiring short secret shared keys. In *Advances in Cryptology*, pages 79–86. Springer, 1983.

BT16. Mihir Bellare and Björn Tackmann. The multi-user security of authenticated encryption: Aes-gcm in tls 1.3. In *Annual Cryptology Conference*, pages 247–276. Springer, 2016.

CW79. Larry Carter and Mark N. Wegman. Universal classes of hash functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.

CZ81. David G Cantor and Hans Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, 36(154):587–592, 1981.

dB93. Bert den Boer. A simple and key-economical unconditional authentication scheme. *Journal of Computer Security*, 2:65–71, 1993. URL: `http://cr.yp.to/bib/entries.html#1993/denboer`.

DR05. Joan Daemen and Vincent Rijmen. Rijndael/aes. In *Encyclopedia of Cryptography and Security*, pages 520–524. Springer, 2005.

GMS74. Edgar N Gilbert, F Jessie MacWilliams, and Neil JA Sloane. Codes which detect deception. *Bell Labs Technical Journal*, 53(3):405–424, 1974.

HK97. Shai Halevi and Hugo Krawczyk. Mmh: Software message authentication in the gbit/second rates. In *International Workshop on Fast Software Encryption*, pages 172–189. Springer, 1997.

HP08. Helena Handschuh and Bart Preneel. Key-recovery attacks on universal hash function based mac algorithms. In *Annual International Cryptology Conference*, pages 144–161. Springer, 2008.

IOM12a. Tetsu Iwata, Keisuke Ohashi, and Kazuhiko Minematsu. Breaking and repairing gcm security proofs. 2012.

IOM12b. Tetsu Iwata, Keisuke Ohashi, and Kazuhiko Minematsu. Breaking and repairing gcm security proofs. In *Proceedings of the 32nd Annual Cryptology Conference on Advances in Cryptology—CRYPTO 2012-Volume 7417*, pages 31–49. Springer-Verlag New York, Inc., 2012.

Jou. Antoine Joux. Comments on the draft gcm specification–authentication failures in nist version of gcm.

JTC11. JTC1. ISO/IEC 9797-1:2011 information technology - security techniques - message authentication codes (macs) - part 1: Mechanisms using a block cipher. 2011.

KR87. Richard M. Karp and Michael O. Rabin. Efficient randomized pattern-matching algorithms. *IBM Journal of Research and Development*, 31:249–260, 1987. URL: `http://cr.yp.to/bib/entries.html#1987/karp`.

Kra94. Hugo Krawczyk. Lfsr-based hashing and authentication. In *Annual International Cryptology Conference*, pages 129–139. Springer, 1994.

LP18. Atul Luykx and Bart Preneel. Optimal forgeries against polynomial-based macs and gcm. In *EUROCRYPT 2018*, pages xxx–xxx, 2018.

LS18. Gaëtan Leurent and Ferdinand Sibleyras. The missing difference problem, and its applications to counter mode encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 745–770. Springer, 2018.

MV04. David A McGrew and John Viega. The security and performance of the galois/counter mode (gcm) of operation. In *International Conference on Cryptology in India*, pages 343–355. Springer, 2004.

MV06. David McGrew and John Viega. The use of galois message authentication code (gmac) in ipsec esp and ah. Technical report, May, 2006.

Nan14. Mridul Nandi. On the minimum number of multiplications necessary for universal hash functions. In *International Workshop on Fast Software Encryption*, pages 489–508. Springer, 2014.

PC15. Gordon Procter and Carlos Cid. On weak keys and forgery attacks against polynomial-based mac schemes. *Journal of Cryptology*, 28(4):769–795, 2015.

Pub01. NIST FIPS Pub. 197: Advanced encryption standard (aes). *Federal information processing standards publication*, 197(441):0311, 2001.

Rab81. Michael O. Rabin. Fingerprinting by random polynomials, 1981. URL: `http://cr.yp.to/bib/entries.html#1981/rabin`. Note: Harvard Aiken Computational Laboratory TR-15-81.

Rog95. Phillip Rogaway. Bucket hashing and its application to fast message authentication. In *Annual International Cryptology Conference*, pages 29–42. Springer, 1995.

Saa12. Markku-Juhani Olavi Saarinen. Cycling attacks on gcm, ghash and other polynomial macs and hashes. In *Fast Software Encryption*, pages 216–225. Springer, 2012.

SCM08. Joseph Salowey, Abhijit Choudhury, and David McGrew. Aes galois counter mode (gcm) cipher suites for tls. Technical report, August, 2008.

Sho96. Victor Shoup. On fast and provably secure message authentication based on universal hashing. In *Annual International Cryptology Conference*, pages 313–328. Springer, 1996.

Sti94. Douglas R. Stinson. Universal hashing and authentication codes. *Designs, Codes and Cryptography*, 4(3):369–380, 1994.

Tay94. Richard Taylor. An integrity check value algorithm for stream ciphers. pages 40–48, 1994. URL: `http://cr.yp.to/bib/entries.html#1994/taylor`.

WC81. Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.

ZTG13. Bo Zhu, Yin Tan, and Guang Gong. Revisiting mac forgeries, weak keys and provable security of galois/counter mode of operation. In *International Conference on Cryptology and Network Security*, pages 20–38. Springer, 2013.