Optimal Forgeries Against Polynomial-Based MACs and GCM

Atul Luykx[1] and Bart Preneel[2]

[1] Visa Research, Palo Alto, USA
aluykx@visa.com
[2] COSIC KU Leuven and iMinds, Leuven, Belgium
bart.preneel@esat.kuleuven.be

Abstract. Polynomial-based authentication algorithms, such as GCM and Poly1305, have seen widespread adoption in practice. Due to their importance, a significant amount of attention has been given to understanding and improving both proofs and attacks against such schemes. At EUROCRYPT 2005, Bernstein published the best known analysis of the schemes when instantiated with PRPs, thereby establishing the most lenient limits on the amount of data the schemes can process per key. A long line of work, initiated by Handschuh and Preneel at CRYPTO 2008, finds the best known attacks, advancing our understanding of the fragility of the schemes. Yet surprisingly, no known attacks perform as well as the predicted worst-case attacks allowed by Bernstein's analysis, nor has there been any advancement in proofs improving Bernstein's bounds, and the gap between attacks and analysis is significant. We settle the issue by finding a novel attack against polynomial-based authentication algorithms using PRPs, and combine it with new analysis, to show that Bernstein's bound, and our attacks, are optimal.

Keywords: forgery, Wegman-Carter, authenticator, MAC, GCM, universal hash, polynomial

1 Introduction

Polynomial-based universal hash functions [dB93,Tay93,BJKS93] are simple and fast. They map inputs to polynomials, which are then evaluated on keys to produce output. When used to provide data authenticity as Message Authentication Code (MAC) algorithms or in Authenticated Encryption (AE) schemes, they often take the form of Wegman-Carter (WC) authenticators [WC81], which add the polynomial output to randomly generated values.

Part of the appeal of such polynomial-based WC authenticators is that if the polynomial keys and random values are generated independently and uniformly for each message, then information-theoretic security is achieved, as initially explored by Gilbert, MacWilliams, and Sloane [GMS74], following pioneering work by Simmons as described in [Sim91]. However, in the interest of speed and practicality, tweaks were introduced to WC authenticators, seemingly not affecting security.

Wegman and Carter [WC81] introduced one of the first such tweaks[3], by holding polynomial keys constant across messages, which maintained security as long as the polynomial outputs are still added to fresh random values each time. Further work then instantiated the random values via a pseudorandom number generator [Bra82], pseudorandom function (PRF), and then pseudorandom permutation (PRP) outputs [Sho96], the latter being dubbed Wegman-Carter-Shoup (WCS) authenticators by Bernstein [Ber05b]. Uniqueness of the PRF and PRP outputs is guaranteed using a nonce. With m the message and n the nonce, the resulting constructions take the form $(n, m) \mapsto \pi(n) + \rho(m)$, with π the PRF or PRP, and ρ the universal hash function.

The switch to using PRFs and PRPs means that information-theoretic is replaced by complexity-theoretic security. Furthermore, switching to PRPs in WCS authenticators results in security bound degradation, impacting the amount of data that can be processed per key (as, for example, exploited by the Sweet32 attacks [BL16]). Naïve analysis uses the fact that PRPs are indistinguishable from PRFs up to the birthday bound, however this imposes stringent limits. Shoup [Sho96], and then Bernstein [Ber05b] improve this analysis significantly using advanced techniques, yet do not remove the birthday bound limit. Regardless, despite the data limits, the use of PRPs enables practical and fast instantiations of MAC and AE algorithms, such as Poly1305-AES [Ber05c] and GCM [MV04a,MV04b], the latter of which has seen widespread adoption in practice [VM06,SMC08,IS09].

As a result of the increased significance of WCS authenticators schemes likes GCM, more recent work has focused on trying to understand their fragility when deployed in the real-world. The history of attacks against WC and WCS authenticators consists of work exploring the consequences of fixing the polynomial key across all messages — once the polynomial key is known, all security is lost.

Joux [Jou] and Handschuh and Preneel [HP08] exhibit attacks which recover the polynomial key the moment a nonce is repeated. Ferguson [Fer05] explores attacks when tags are too short, further improved by Mattson and Westerlund [MW16]. A long line of work initiated by Handschuh and Preneel [HP08], illustrates how to efficiently exploit verification attempts to eliminate false keys, by systematically narrowing the set of potential polynomial keys and searching for so-called "weak" keys [Saa12,PC15,ABBT15,ZW17,ZTG13].

However, interestingly, in the case of polynomial-based WCS authenticators, none of the nonce-respecting attacks match the success of the predicted worst-case attacks by Bernstein [Ber05b]. Furthermore, the gap in success between the predicted worst-case and best-known attacks grows quadratically in the number of queries made to the authenticator. Naturally, one is led to question whether Bernstein's analysis is in fact the best one can do, or whether there actually is an attack, forcing us to abide by the data limits.

[3] Strictly speaking, Wegman and Carter did not tweak the constructions pioneered by Simmons, as the connection between the two works was made only later by Stinson [Sti91].

1.1 Contributions

We exhibit novel nonce-respecting attacks against polynomial-based WCS authenticators (Sect. 3), and show how they naturally arise from a new, simplified proof (Sect. 4). We prove that both our attack and Bernstein's bound [Ber05b] are optimal, by showing they match (Sect. 5).

Unlike other birthday bound attacks, our attacks work by establishing quadratically many polynomial systems of equations from the tagging queries. It applies to polynomial-based WCS authenticators such as Poly1305-AES, as well as GCM and the variant SGCM [Saa11]. We achieve optimality in a chosen-plaintext setting, however the attacks can be mounted passively, using just known plaintext for MACs and ciphertext for AE schemes.

1.2 Related Work

Our introduction provides only a narrow view of the history of universal hash functions, targeted to ones based on polynomials. Bernstein [Ber05c] provides a genealogy of polynomial-based universal hash functions and Wegman-Carter authenticators, and both Procter and Cid [PC15,PC13] and Abdelraheem et al. [ABBT15] provide detailed overviews of the past attacks against polynomial-based Wegman-Carter MACs and GCM.

Zhu, Tan, and Gong [ZTG13] and Ferguson [Fer05] have pointed out that non-96-bit nonce GCM suffers from birthday bound attacks which lead to immediate recovery of the polynomial key. Such attacks use the fact that the nonce is processed by the universal hash function before being used, resulting in block cipher call collisions. These attacks are not applicable to the most widely deployed version of GCM, which uses 96 bit nonces, nor to polynomial-based WCS authenticators in general.

Iwata, Ohashi, and Minematsu [IOM12] identify and correct issues with GCM's original analysis [MV04a]. Niwa, Ohashi, Minematsu, and Iwata find further improvements in GCM's bounds [NOMI15]. Their proofs do not improve over Bernstein's analysis [Ber05b].

New constructions using universal hash functions like EWCDM [CS16] achieve full security [MN17] in the nonce-respecting setting, and maintain security during nonce-misuse.

McGrew and Fluhrer [MF05] and Black and Cochran [BC09] explore how easy it is to find multiple forgeries once a single forgery has been performed.

A long line of research seeks attacks and proofs of constructions which match each other, such as the generic attack by Preneel and van Oorschot [PvO99], tight analysis for CBC-MAC [BPR05,Pie06], keyed sponges and truncated CBC [GPT15], and HMAC [GPR14], and new attacks for PMAC [LPSY16,GPR16].

2 Preliminaries

2.1 Basic Definitions and Notation

The notation used throughout the paper is summarized in App. C. Unless specified otherwise, all sets are assumed to be finite. Vectors are denoted $\boldsymbol{x} \in \mathsf{X}^q$, with corresponding components $(x_1, x_2, \ldots, x_q)$. Given a set X, $\mathsf{X}^{\leq \ell}$ denotes the set of non-empty sequences of elements of X with length not greater than ℓ.

A *random function* $\rho : \mathsf{M} \to \mathsf{T}$ is a random variable distributed over the set of all functions from M to T. A *uniformly distributed random permutation* (URP) $\varphi : \mathsf{N} \to \mathsf{N}$ is a random variable distributed over the set of all permutations on N, where N is assumed to be finite. When we write $\varphi : \mathsf{N} \to \mathsf{T}$ is a URP, we implicitly assume that $\mathsf{N} = \mathsf{T}$.

The symbol $\mathbb{P}$ denotes a probability measure, and $\mathbb{E}$ expected value.

We make the following simplifications when discussing the algorithms. We analyze block cipher-based constructions by replacing each block cipher call with a URP call. This commonly used technique allows us to focus on the constructions' security without worrying about the underlying block cipher's quality. See for example [Ber05b]. Furthermore, although our analysis uses information-theoretic adversaries, the attacks we describe are efficient, but require large storage.

We also implicitly include key generation as part of the oracles. For example, consider a construction $E : \mathsf{K} \times \mathsf{M} \to \mathsf{T}$, where E is stateless and deterministic, and K is its "key" input. In the settings we consider, E-queries are only actually made to $E(k, \cdot)$, where the key input is fixed to some random variable k chosen uniformly at random from K. Hence, rather than each time talking of $E(k, \cdot)$, we simplify notation by considering the random function $\rho(m) \stackrel{\text{def}}{=} E(k, m)$, with the uniform random variable k implicitly part of ρ's description.

2.2 Polynomial-based WCS Authenticators

Although not necessary, for simplicity we fix tags to lie in a commutative group. The following definition is from Bernstein [Ber05b].

Definition 2.1 (WCS Authenticator). *Let* T *be a commutative group with operation* $+$*. Let* $\pi : \mathsf{N} \to \mathsf{T}$ *be a URP, and* $\rho : \mathsf{M} \to \mathsf{T}$ *a random function. The* Wegman-Carter-Shoup (WCS) authenticator *maps elements* $(n, m) \in \mathsf{N} \times \mathsf{M}$ *to* $\pi(n) + \rho(m)$*.*

We take the following definition from Procter and Cid [PC15]

Definition 2.2 (Polynomial-Based Universal Hash). *Let* X *be a field and* ℓ *a positive integer. Given* $\boldsymbol{x} = (x_1, x_2, \ldots, x_l) \in \mathsf{X}^{\leq \ell}$*, define the polynomial* $p_{\boldsymbol{x}}(\alpha)$ *by*

$$p_{\boldsymbol{x}}(\alpha) \stackrel{\text{def}}{=} \sum_{i=1}^{l} x_i \cdot \alpha^i \,. \tag{1}$$

Then the polynomial-based universal hash function $\rho : \mathsf{X}^{\leq \ell} \to \mathsf{X}$ *is the random function* $\rho(\boldsymbol{x}) \stackrel{\text{def}}{=} p_{\boldsymbol{x}}(\kappa)$*, where* κ *is a uniform random variable over* X*, and* $\boldsymbol{x} \in \mathsf{X}^{\leq \ell}$*.*

We say that the input messages $\mathsf{X}^{\leq}$ to the polynomial-based universal hash consist of *blocks*, with the *block length* of the messages being at most ℓ.

When a WCS authenticator uses a polynomial-based universal hash function, we call the resulting construction a *polynomial-based WCS authenticator*.

Let $\gamma : \mathsf{N} \times \mathsf{M} \to \mathsf{T}$ be a WCS authenticator. An adversary $\mathbf{A}$ interacting with γ is said to be *nonce-respecting* if it never repeats N-input to γ. Furthermore, the *verification oracle* associated to γ, $V : \mathsf{N} \times \mathsf{M} \times \mathsf{T} \to \{0,1\}$, is defined as

$$V(n,m,t) = \begin{cases} 1 & \text{if } \gamma(n,m) = t \\ 0 & \text{otherwise} \end{cases} . \tag{2}$$

Nonce-respecting adversaries may repeat nonce-input to V.

Definition 2.3 (Authenticity Advantage). *Let* $\mathbf{A}$ *be a nonce-respecting adversary interacting with WCS authenticator* $\gamma : \mathsf{N} \times \mathsf{M} \to \mathsf{T}$ *and associated verification oracle* V*. Then* $\mathbf{A}$*'s* authenticity advantage*, denoted* $\mathsf{Auth}_\gamma(\mathbf{A})$*, is the probability that* $\mathbf{A}$ *makes a* V*-query* (n^*, m^*, t^*) *resulting in* V *outputting* 1 *and* $\gamma(n^*, m^*) = t^*$ *was not a previous query-response from* γ*.*

In our analysis we will also need the following definition.

Definition 2.4 (Single-Forgery Advantage). *Let* $\mathbf{A}$ *be a nonce-respecting adversary interacting with WCS authenticator* $\gamma : \mathsf{N} \times \mathsf{M} \to \mathsf{T}$*, resulting in queries* $\gamma(n_i, m_i) = t_i$ *for* $i = 1, \ldots, q$*. Say that* $\mathbf{A}$ *outputs* (n^*, m^*, t^*) *after its interaction. Then* $\mathbf{A}$*'s* single-forgery advantage *is*

$$\mathsf{sAuth}_\gamma(\mathbf{A}) \stackrel{\text{def}}{=} \mathbb{P}\left[\gamma(n^*, m^*) = t^*, (n^*, m^*, t^*) \neq (n_i, m_i, t_i)\ \forall i\right] . \tag{3}$$

The maximum over all adversaries making at most q *queries is denoted* $\mathsf{sAuth}_\gamma(q)$*.*

Bernstein connects Auth and sAuth as follows.

Theorem 2.1 ([Ber05a]). *Let* $\mathbf{A}$ *be an authenticity adversary making at most* q γ *and* v *verification queries, then*

$$\mathsf{Auth}_\gamma(\mathbf{A}) \leq v \cdot \mathsf{sAuth}_\gamma(q) . \tag{4}$$

Bellare, Goldreich, and Mityagin prove a similar result for different constructions [BGM04].

2.3 GCM

We present those details of GCM [MV04a,MV04b] necessary to describe our attacks. GCM takes nonce, associated data, and plaintext input. It operates by

first encrypting the plaintext using CTR mode [Nat80] into a ciphertext c. Then it processes the ciphertext and associated data using a WCS authenticator into a tag.

GCM only uses one key, namely a block cipher key; as explained before, we view the keyed block cipher as a URP π over the set of 128-bit strings, hence the block cipher key is implicit in our description. An authentication key L is computed as the output of π under the all-zero string, which we denote 0: $L \stackrel{\text{def}}{=} \pi(0)$.

GCM's WCS authenticator views the set of 128-bit strings as a finite field with 2^{128} elements. Once the ciphertext c has been computed using CTR mode, its length is encoded in a 64-bit string and the ciphertext is padded with zeros to have length a multiple of 128 bits. The associated data is processed in the same way. Let $a_1, a_2, \ldots, a_l$ and $c_1, c_2, \ldots, c_{l'}$ denote the padded associated data and ciphertext, respectively, where the length of all blocks a_i and c_i is 128 bits. Let x_0 denote the concatenation of the encoded lengths of the associated data and ciphertext. Then, if $\boldsymbol{x} = (x_0, a_l, a_{l-1}, \ldots, a_1, c_{l'}, \ldots, c_1)$, GCM computes its tag as

$$p_{\boldsymbol{x}}(L) + \pi(n) \quad \text{with } L = \pi(0)\,, \tag{5}$$

where n is a value deduced from the nonce.

All π-input in GCM can be derived from the nonce and L, and no two π-inputs are the same, unless some unlikely event happens, in which case GCM loses all security [Jou,Fer05,ZTG13]. In more detail, the nonce is converted into distinct counters for CTR mode, as well as an additional, distinct input, which is used for the URP input in GCM's WCS authenticator, denoted n in 5. In 96-bit nonce GCM, n is equal to the nonce concatenated with a string consisting of 31 zeroes, followed by a 1, and the counters used in CTR mode increment the last 32 bits of n.

In our attacks and analysis below we mostly focus on plain WCS authenticators, however everything translates nearly verbatim over to GCM's WCS authenticator.

3 Key Recovery Attacks

Most of the previously published attacks aim to recover the polynomial key of the WCS authenticator in order to be able to construct arbitrary forgeries. All known key recovery attacks focus either on reducing the set of candidate keys $\mathcal{T}$, which contains the actual key, or, equivalently, increasing $\mathcal{T}$'s complement $\mathcal{F}$, the set of "false" keys. The former can be achieved through nonce misuse [Jou,HP08], which allows one to obtain a polynomial for which the key is a root, thereby reducing $\mathcal{T}$ to the set of all roots of the polynomial. Although nonce misuse attacks are important to understand the fragility of the schemes, we focus on attacks which stay in the nonce-respecting model.

In contrast, the nonce-respecting attacks reduce $\mathcal{F}$ via repeated verification attempts [HP08,PC15,ABBT15]. Their goal is to construct a *forgery polynomial*

which evaluates to zero on the key. Then the forgery polynomial is combined with a previous tagging query into a verification attempt in such a way that if the verification attempt fails, then one knows that the key is not one of the roots of the forgery polynomial. If the forgery polynomial has degree ℓ, then at most ℓ faulty keys can be removed for each verification attempt, resulting in a success probability of at most

$$\frac{1}{|\mathsf{T}| - v\ell}\,, \tag{6}$$

where v is the number of verification attempts.

Our attacks differ from the previous nonce-respecting attacks in two ways: they do not require verification attempts in order to increase $\mathcal{F}$, and $\mathcal{F}$ increases quadratically as a function of the number of tagging queries, q, giving a success probability of roughly

$$\frac{1}{|\mathsf{T}| - q^2}\,. \tag{7}$$

We describe chosen-plaintext attacks which perfectly match the bounds for both polynomial-based WCS MACs and GCM. The attacks can also be applied passively, where adversaries do not have chosen-plaintext control. Success then depends in a non-trivial way on the message distribution, which in turn depends on the application in consideration; we leave further detailed analysis of the known-plaintext attacks for future work. In Sect. 5 we show that our chosen-plaintext attacks are optimal.

3.1 WCS Authenticator Attacks

Constructing the False-Key Set. Let $\gamma(n, m) = \pi(n) + \rho(m)$ be a polynomial-based WCS authenticator, with π a URP and ρ a polynomial-based universal hash function. Say that we somehow know that the queries $\gamma(n_i, m_i) = t_i$ for $i = 1, \ldots, q$ were made. This means

$$\pi(n_i) + \rho(m_i) = t_i \quad \text{or} \quad \pi(n_i) = t_i - \rho(m_i)\,, \quad \text{for } i = 1, \ldots, q\,. \tag{8}$$

Since π is a permutation, this means

$$t_i - \rho(m_i) \neq t_j - \rho(m_j)\,, \quad \text{for } i \neq j\,. \tag{9}$$

In particular, we know that the real key κ does *not* satisfy the polynomial equations

$$\rho(m_i) - \rho(m_j) + t_j - t_i = 0\,, \quad \text{for } i \neq j\,. \tag{10}$$

Therefore, each query to γ might allow us to increase the set of false keys. In fact, the jth query to γ gives an additional $j - 1$ equations which can be used to discard keys.

Known-plaintext Attack. Given (n_i, m_i, t_i) for $i = 1, \ldots, q$, perform the following:

1. Construct
$$\mathcal{F} \stackrel{\text{def}}{=} \left\{ k \;\middle|\; p_{m_i}(k) - p_{m_j}(k) + t_j - t_i = 0, i \neq j \right\} . \tag{11}$$
2. Pick any $k^* \notin \mathcal{F}$, output k^*.

Analysis of the known-plaintext attack is complicated by the choice of distribution for the messages m_i. We focus instead on analyzing the chosen-plaintext attack below.

Chosen-plaintext Attack. Choose q distinct messages of length one block, m_1, $m_2, \ldots, m_q$, and q nonces $n_1, n_2, \ldots, n_q$. For example, one could pick $m_i = n_i = i$, for some encoding of i. Then conclude with the known-plaintext attack described above. The resulting false-key set is
$$\mathcal{F} = \left\{ \frac{t_i - t_j}{m_i - m_j}, i \neq j \right\} . \tag{12}$$

The following proposition establishes the expected size of $\mathcal{F}$ for this attack. In Sect. 5 we connect the expected size of $\mathcal{F}$ with the success of key recovery attacks and forgeries.

Proposition 3.1. *Let $N = |\mathsf{T}|$, and say that $q \leq \sqrt{N-3}$, then*
$$\mathbb{E}\left(|\mathcal{F}|\right) \geq \frac{q(q-1)}{4} , \tag{13}$$
where $\mathcal{F}$ is from 12.

Proof. Let κ denote the real key, then
$$\mathcal{F} = \left\{ \frac{\pi(n_i) - \pi(n_j) + \kappa m_i - \kappa m_j}{m_i - m_j}, i \neq j \right\} \tag{14}$$
$$= \left\{ \frac{\pi(n_i) - \pi(n_j) + \kappa(m_i - m_j)}{m_i - m_j}, i \neq j \right\} \tag{15}$$
$$= \left\{ \frac{\pi(n_i) - \pi(n_j)}{m_i - m_j} + 1, i \neq j \right\} . \tag{16}$$
Let $S = \{(\pi(n_i) - \pi(n_j))/(m_i - m_j), i \neq j\}$, so that $|S| = |\mathcal{F}|$.

By Markov's inequality,
$$\mathbb{E}\left(|S|\right) \geq \mathbb{P}\left[|S| \geq \frac{q(q-1)}{2} \right] \cdot \frac{q(q-1)}{2} , \tag{17}$$
and $|S| \geq q(q-1)/2$ only if none of the $(\pi(n_i) - \pi(n_j))/(m_i - m_j)$ collide. By applying a union bound we know that the probability there is such a collision is at most $q(q-1)/(2(N-3))$, hence
$$\mathbb{P}\left[|S| \geq \frac{q(q-1)}{2} \right] \geq 1 - \frac{q(q-1)}{2(N-3)} . \tag{18}$$

If $q \leq \sqrt{N-3}$, then

$$1 - \frac{q(q-1)}{2(N-3)} \geq \frac{1}{2}\,, \tag{19}$$

and we have our desired bound. □

3.2 GCM Attacks

With a known-plaintext attack against GCM it is possible to increase $\mathcal{F}$ without resorting to verification attempts or polynomial equations. Since we know that the authentication key is computed as $\pi(0)$, and all inputs to π are distinct, each URP output from CTR mode reduces the set of valid keys, which you can compute easily if you know the plaintext. However, such an attack still requires known plaintext, potentially making it more difficult to implement in practice.

In contrast, if we apply our WCS authenticator attacks described above to GCM, by replacing messages with ciphertexts, then we arrive at an attack which potentially only requires ciphertext. In a passive setting, the steps are identical: create a false-key set $\mathcal{F}$ as in Eq. 11, except the polynomials are replaced by GCM's, from 5.

The optimal chosen-plaintext attack changes slightly for GCM, since we need to deal with the encoded lengths of the ciphertexts in the polynomials of Eq. 5. Instead of choosing q distinct plaintexts m_i, we now set all plaintexts to be the all-zero string of length one block. This results in polynomials

$$xL + c_i L^2\,, \tag{20}$$

where x is the encoding of the length of a one-block length ciphertext, and the c_i are the ciphertexts, all distinct from each other. The resulting false-key set is as follows:

$$\left\{ \sqrt{\frac{t_i - t_j}{c_i - c_j}}, i \neq j \right\}. \tag{21}$$

Since the square root is bijective in finite fields of characteristic two, we have that the above set contains the same number of elements as

$$\left\{ \frac{t_i - t_j}{c_i - c_j}, i \neq j \right\}, \tag{22}$$

and the analysis made for WCS authenticators holds with little modification.

4 Bounding Authenticity with Key Recovery

4.1 Bernstein's Analysis

Bernstein analyzes a generalization of Wegman-Carter and WCS MACs, namely those of the form $(n, m) \mapsto \rho(m) + \varphi(n)$, where $\rho : \mathsf{M} \to \mathsf{T}$ and $\varphi : \mathsf{N} \to \mathsf{T}$ are independent random functions. Wegman-Carter authenticators fix φ to be

a uniformly distributed random function, and WCS authenticators fix φ to be a URP. As part of his analysis, Bernstein uses *differential probability* [Ber05b], more commonly known as ϵ-almost (XOR) universal, given by

$$\Delta_\rho \stackrel{\text{def}}{=} \max_{\substack{m \neq m' \\ t \in \mathsf{T}}} \mathbb{P}\left[\rho(m) = \rho(m') + t\right] . \tag{23}$$

Various papers [dB93,Tay93,BJKS93] establish that for a polynomial-based universal hash function $\rho : \mathsf{M} \to \mathsf{T}$, $\Delta_\rho \leq \ell / |\mathsf{T}|$, where $\mathsf{M} = \mathsf{T}^{\leq \ell}$.

Bernstein also introduces the concept of *interpolation probabilities* of a random function φ, which is the probability that $\varphi(x_i) = y_i$ for some values $x_1, \ldots, x_q$ and $y_1, \ldots, y_q$. Bernstein establishes that $\rho(m) + \varphi(n)$ is secure if ρ's differential and φ's interpolation probabilities are small. Ultimately when applied to polynomial-based WCS authenticators, we get the following.

Theorem 4.1. *Let $\gamma : \mathsf{N} \times \mathsf{M} \to \mathsf{T}$ be a polynomial-based WCS authenticator with $\mathsf{M} = \mathsf{T}^{\leq \ell}$ and let $\mathbf{A}$ be a nonce-respecting adversary against γ making at most q γ and v verification queries, then*

$$\mathsf{Auth}_\gamma(\mathbf{A}) \leq v \cdot \frac{\ell}{|\mathsf{T}|} \cdot \left(1 - \frac{q}{|\mathsf{T}|}\right)^{-\frac{q+1}{2}} . \tag{24}$$

4.2 Reshaping Authenticity Advantage

Although Bernstein's analysis is general and applies to more than just polynomial-based WCS MACs, a targeted analysis will elucidate the gap between currently known attacks and the bound given by Bernstein.

Whereas Bernstein proves bounds for $\varphi(n) + \rho(m)$ in terms of φ's interpolation and ρ's differential probability, we instead rework the bounds to φ's unpredictability (Sec 4.3) and key recovery against ρ (Sec. 4.4), the latter only applying to polynomial-based MACs. The concepts introduced in this section will allow us to prove that the CPA attacks introduced in Sec. 3 are in fact optimal.

Instrumental to our analysis is the fact that an adversary's single-forgery advantage can be split in two, according to whether its attempted forgery (n^*, m^*, t^*) uses a nonce n^* that was never used before, or not. We let $\mathsf{sAuth}^{\text{new}}_\gamma(\mathbf{A})$ denote the probability that $\mathbf{A}$ forges and uses a new nonce, and $\mathsf{sAuth}^{\text{old}}_\gamma(\mathbf{A})$ the probability that $\mathbf{A}$ forges and uses an old nonce. By basic probability theory,

$$\mathsf{sAuth}_\gamma(\mathbf{A}) \leq \max\left\{\mathsf{sAuth}^{\text{new}}_\gamma, \mathsf{sAuth}^{\text{old}}_\gamma\right\} . \tag{25}$$

Letting KR denote polynomial key recovery advantage (see Def. 4.2), we establish the following result.

Corollary 4.1. *Let $\gamma : (n, m) \mapsto \rho(m) + \pi(n)$ be a polynomial-based WCS authenticator with $\rho : \mathsf{M} \to \mathsf{T}$ a random function, and $\pi : \mathsf{N} \to \mathsf{T}$ an independent*

URP. Let $\mathbf{A}$ *be an authenticity adversary against* γ *making at most* q *queries of length at most* ℓ*. Then*

$$\mathsf{Auth}_{\gamma}(\mathbf{A}) \leq v \cdot \max\left\{\ell \cdot \mathsf{KR}_{\gamma}(q)\,,\ \frac{1}{|\mathsf{T}| - q}\right\}. \tag{26}$$

The proof can be found in App. A, which relies on results developed in the next sections.

4.3 Unpredictability

We show how any attempted forgery using a new nonce against a WCS authenticator has low success probability. This means if authenticity adversaries want to achieve significant advantage, then they must re-use nonces during forgeries. We state the result more generally than for only polynomial-based WCS authenticators.

Definition 4.1 (Unpredictability). *Let* $\mathbf{A}$ *be an adversary interacting with random function* $\varphi : \mathsf{X} \to \mathsf{Y}$*. Say that* $\mathbf{A}$ *produces the sequence* $\boldsymbol{x} \in \mathsf{X}^q$ *and* φ *responds with outputs* $\boldsymbol{y} \in \mathsf{Y}^q$*. Let* (x^*, y^*) *be* $\mathbf{A}$*'s output, then* $\mathbf{A}$*'s* unpredictability *advantage against* φ *is*

$$\mathsf{Unpred}_{\varphi}(\mathbf{A}) \stackrel{\text{def}}{=} \mathbb{P}\left[\varphi(x^*) = y^*, x^* \neq x_i, i = 1, \ldots, q\right], \tag{27}$$

where the probability is taken over the randomness of $\mathbf{A}$ *and* φ*.*

Let $\gamma : (n, m) \mapsto \rho(m) + \pi(n)$ be any Wegman-Carter-style MAC using random functions $\rho : \mathsf{M} \to \mathsf{T}$ and $\varphi : \mathsf{N} \to \mathsf{T}$ which are independent of each other. Let $\mathbf{A}$ be an authenticity adversary against γ. We construct an unpredictability adversary $\mathbf{B}\,\langle\mathbf{A}\rangle$ against φ as follows.

1. $\mathbf{B}$ runs $\mathbf{A}$.
2. $\mathbf{B}$ simulates ρ using its own randomness; call it ρ'.
3. Every γ-query made by $\mathbf{A}$ is reconstructed by $\mathbf{B}$ using ρ' and the φ-oracle $\mathbf{B}$ interacts with. Concretely, every $\gamma(n, m)$ made by $\mathbf{A}$ gets forwarded as $\varphi(n)$, and $\mathbf{B}$ returns $\varphi(n) + \rho'(m)$.
4. $\mathbf{B}$ receives $\mathbf{A}$'s final output, (n^*, m^*, t^*), and finally outputs $(n^*, t^* - \rho'(m^*))$.

Proposition 4.1.

$$\mathsf{sAuth}^{new}_{\gamma}(\mathbf{A}) \leq \mathsf{Unpred}_{\varphi}(\mathbf{B}\,\langle\mathbf{A}\rangle)\,. \tag{28}$$

Proof. First note that $\mathbf{B}$ perfectly reconstructs $\mathbf{A}$'s authenticity game since ρ' is independent of φ. Then, if $\mathbf{A}$ wins its authenticity game, $\gamma(n^*, m^*) = t^*$, or in other words, $\varphi(n^*) + \rho(m^*) = t^*$. In particular, $\varphi(n^*) = t^* - \rho(m^*)$. If n^* has never been queried to φ before, $t^* - \rho(m^*)$ would correctly predict φ's output on an unknown input, hence $\mathbf{B}\langle\mathbf{A}\rangle$ would win its unpredictability game. □

Lemma 4.1. *Let* $\pi : \mathsf{N} \to \mathsf{T}$ *be a URP and* $\mathbf{B}$ *an adversary making at most* q *queries, then*

$$\mathsf{Unpred}_{\pi}(\mathbf{B}) \leq \frac{1}{|\mathsf{T}| - q}\,. \tag{29}$$

4.4 Bounding Forgeries with Key Recovery

Having set aside adversaries which use new nonces for forgeries, we can focus on those that re-use nonces. This section applies only to polynomial-based WCS authenticators.

Definition 4.2 (Polynomial Key Recovery). *Let* $\mathbf{A}$ *be a nonce-respecting adversary interacting with polynomial-based WCS authenticator* γ *using URP* π *and polynomial-based universal hash* ρ*, with* κ *denoting the random variable representing the key underlying* ρ*. Say that* $\mathbf{A}$ *outputs an element* $k^* \in \mathsf{K}$*, then* $\mathbf{A}$*'s* polynomial key recovery advantage *against* γ *is*

$$\mathsf{KR}_\gamma(\mathbf{A}) \stackrel{\text{def}}{=} \mathbb{P}\left[k^* = \kappa\right], \tag{30}$$

where the randomness is taken over $\mathbf{A}$ *and* γ*. We let* $\mathsf{KR}_\gamma(q)$ *denote the maximum of* $\mathsf{KR}_\gamma(\mathbf{A})$ *over all adversaries* $\mathbf{A}$ *making at most* q *queries.*

Forgeries can be used to recover authentication keys. We construct a polynomial key recovery adversary $\mathbf{C}\langle\mathbf{A}\rangle$ against γ.

1. $\mathbf{C}$ runs $\mathbf{A}$.
2. Every (n, m) query by $\mathbf{A}$ gets forwarded to $\mathbf{C}$'s oracle, and $\mathbf{C}$ returns the output $\gamma(n, m)$ to $\mathbf{A}$.
3. When $\mathbf{A}$ outputs (n^*, m^*, t^*), then $\mathbf{C}$ checks to see if $n^* = n_i$ for some previous query $\gamma(n_i, m_i) = t_i$. If this is not the case, then $\mathbf{C}$ aborts. Otherwise $\mathbf{C}$ computes the roots of the polynomial[4] $p_{m^*}(\alpha) - p_{m_i}(\alpha) - t^* + t_i = 0$, and chooses a key uniformly at random from the set of roots.

Proposition 4.2. *Let* $\mathbf{A}$ *be an adversary making queries of length at most* ℓ*. The probability that* $\mathbf{A}$ *wins its authenticity game and outputs* (n^*, m^*, t^*) *where* $n^* = n_i$ *for some previous query* (n_i, m_i) *to* γ*, is bounded above by*

$$\ell \cdot \mathsf{KR}_\gamma(\mathbf{C}\langle\mathbf{A}\rangle). \tag{31}$$

Proof. If $\mathbf{A}$ wins with $n^* = n_i$, then

$$\gamma(n^*, m^*) = \gamma(n_i, m^*) = \varphi(n_i) + \rho(m^*) = t^*, \tag{32}$$

and

$$\gamma(n_i, m_i) = \varphi(n_i) + \rho(m_i) = t_i, \tag{33}$$

therefore $\rho(m^*) - \rho(m_i) - t^* + t_i = 0$. We know that the key used by ρ is in the set of roots of the polynomial $p_{m^*}(\alpha) - p_{m_i}(\alpha) - t^* + t_i$, which has size at most $\max\{|m^*|, |m_i|\}$. Picking an element uniformly at random from this set, we have that $\mathbf{C}$ wins with probability at least $1/\max\{|m^*|, |m_i|\}$. □

[4] Finding roots of polynomials over a finite field is computationally efficient using Berlekamp's algorithm [Ber70] or the Cantor-Zassenhaus algorithm [CZ81]

5 Using Key Recovery to Mount Forgeries

The previous section discussed how to convert authenticity attacks into key recovery attacks to reshape the upper bounds on forgery attacks. Here we discuss the opposite, namely how to use key recovery adversaries to mount forgeries. This will allow us to not only show that the analysis of Sect. 4 is tight, but also that the attacks of Sect. 3 are optimal, using Bernstein's analysis.

5.1 Key-Set Recovery

The obvious way to convert a key recovery attack into an authenticity attack is to run the key recovery adversary and use the output of the key recovery adversary to mount a forgery. We explain this formally in App. B. However, this method constructs authenticity adversaries which are about as successful as key recovery adversaries.

In contrast, as seen in Sect. 4.4, Prop. 4.2, authenticity adversaries might improve over key recovery adversaries by up to a factor of ℓ. Intuitively, given a key recovery adversary, one could try to do this by taking the candidate key k^* output by the key recovery adversary, and finding a polynomial of degree ℓ which contains k^* as a root, and then construct a forgery using this polynomial. The problem with this approach is that most of the roots of the polynomial chosen by the resulting authenticity adversary could be useless, as they could, for example, lie in some false-key set determined by the key recovery adversary. Without any further information about the key recovery adversary it does not seem possible to improve the authenticity adversary.

However, if we instead look at *key-set recovery adversaries*, we can improve our chances of constructing forgeries. We will show that key-set recovery and key-recovery adversaries are in fact very similar, allowing us to prove tight bounds on the connection between key-recovery and forgeries.

Definition 5.1 (Polynomial Key-Set Recovery). *Let* $\mathbf{A}$ *be a nonce-respecting adversary interacting with polynomial-based WCS authenticator* γ *using URP* π *and polynomial-based universal hash* ρ*, with* κ *denoting the random variable representing the key underlying* ρ*. Say that* $\mathbf{A}$ *outputs a set* $K^* \subset \mathsf{K}$*, and let* 1_{K^*} *denote the random variable which equals one if* $\kappa \in K^*$ *and zero otherwise. Then* $\mathbf{A}$*'s* polynomial key-set recovery advantage *against* γ *is*

$$\mathsf{KS}_\gamma(\mathbf{A}) \stackrel{\text{def}}{=} \mathbb{E}\left(\frac{1_{K^*}}{|K^*|}\right), \tag{34}$$

where the randomness is taken over $\mathbf{A}$ *and* γ*. We let* $\mathsf{KS}_\gamma(q)$ *denote the maximum of* $\mathsf{KS}_\gamma(\mathbf{A})$ *taken over all adversaries making at most* q *queries.*

Given a key-set recovery adversary $\mathbf{C}$, we construct single-forgery adversary $\mathbf{A}\langle\mathbf{C}\rangle$ as follows:

1. $\mathbf{A}$ runs $\mathbf{C}$, and responds to any $\mathbf{C}$-query (n, m) with $\gamma(n, m)$.

2. When $\mathbf{C}$ outputs its candidate set K^*, $\mathbf{A}$ picks ℓ elements uniformly at random from K^* and constructs a polynomial p_{m^*} with those elements as roots.
3. $\mathbf{A}$ picks any previous query $\gamma(n, m) = t$ made by $\mathbf{C}$, adds m^* to m componentwise to get $m' = (m_1 + m_1^*, m_2 + m_2^*, \ldots)$, and submits the forgery attempt (n, m', t).

The following proposition shows that one can construct much better forgeries using key-set recovery adversaries.

Proposition 5.1.

$$\ell \cdot \mathsf{KS}_\gamma(\mathbf{C}) \leq \mathsf{sAuth}_\gamma^{old}(\mathbf{A}\langle\mathbf{C}\rangle)\,. \tag{35}$$

Proof. Let L denote the ℓ elements that $\mathbf{A}$ picks from $\mathcal{T}$. Adversary $\mathbf{A}$ wins if $\kappa \in L$, since then $p_{m^*}(\kappa) = 0$ and so $p_{m+m^*}(\kappa) + \pi(n) = t$.

$$\mathbb{P}\left[\kappa \in L\right] = \sum_n \mathbb{P}\left[\kappa \in L \;\middle|\; |\mathcal{T}| = n, \kappa \in \mathcal{T}\right] \mathbb{P}\left[\kappa \in \mathcal{T}, |\mathcal{T}| = n\right] \tag{36}$$

$$= \sum_n \frac{\ell}{n} \cdot \mathbb{P}\left[\kappa \in \mathcal{T}, |\mathcal{T}| = n\right] \tag{37}$$

$$= \ell \cdot \mathbb{E}\left(\frac{1_{\mathcal{T}}}{|\mathcal{T}|}\right) = \ell \cdot \mathsf{KS}_\gamma(\mathbf{C})\,. \tag{38}$$

□

Furthermore, there is little real difference between key-recovery and key-set recovery advantage.

Proposition 5.2.

$$\mathsf{KS}_\gamma(q) = \mathsf{KR}_\gamma(q)\,. \tag{39}$$

Proof. If the output set size of a key-set recovery adversary is always one, then key-set recovery advantage is identical to key-recovery advantage. Since any key-recovery adversary can be converted into a key-set recovery adversary with output set size one, we have that $\mathsf{KR}_\gamma(q) \leq \mathsf{KS}_\gamma(q)$.

Given a key-set recovery adversary $\mathbf{C}$, we convert it into a key-recovery adversary $\mathbf{C}'$ by picking a candidate key k^* uniformly at random from the output set $\mathcal{T}$. Then

$$\mathsf{KR}_\gamma(\mathbf{C}') = \mathbb{P}\left[\kappa = k^*\right] \tag{40}$$

$$= \sum_n \mathbb{P}\left[\kappa = k^* \;\middle|\; \kappa \in \mathcal{T}, |\mathcal{T}| = n\right] \mathbb{P}\left[\kappa \in \mathcal{T}, |\mathcal{T}| = n\right] \tag{41}$$

$$= \sum_n \frac{1}{n} \mathbb{P}\left[\kappa \in \mathcal{T}, |\mathcal{T}| = n\right] = \mathsf{KS}_\gamma(\mathbf{C})\,. \tag{42}$$

□

Prop. 4.2, Prop. 5.1, and Prop. 5.2 establish the following result, confirming that the analysis of Sect. 4.4 is tight.

Corollary 5.1.

$$\ell \cdot \mathsf{KR}_\gamma(q) = \mathsf{sAuth}_\gamma^{old}(q)\,. \tag{43}$$

5.2 Attack Success Probability and Optimality

Our chosen-plaintext attack only uses messages of length one block, which is reflected in the fact that $|\mathcal{F}|$ only grows as a function of q. Intuitively one would expect to be able to increase $\mathcal{F}$ as well by taking advantage of longer messages and the fact that polynomials of higher degree have more roots. However, here we show that this is impossible.

The success probability of the key recovery attacks from Sect. 3 is given as follows, which results from the observation that the real key cannot be in $\mathcal{F}$ by definition.

Proposition 5.3. *Let* $\mathbf{A}$ *denote the chosen-plaintext attack from Sect. 3, then*

$$\mathsf{KR}_\gamma(\mathbf{A}) \geq \frac{1}{|\mathsf{T}| - \mathbb{E}\left(|\mathcal{F}|\right)}\,. \tag{44}$$

Combining this result with Bernstein's result, we have the following.

Theorem 5.1. *Let* $\mathcal{F}$ *be defined as in Sect. 3, then*

$$\mathbb{E}\left(|\mathcal{F}|\right) \leq \frac{q(q+1)}{2}\,. \tag{45}$$

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

$$\ell \cdot \frac{1}{|\mathsf{T}| - \mathbb{E}\left(|\mathcal{F}|\right)} \leq \frac{\ell}{|\mathsf{T}|} \cdot \left(1 - \frac{q}{|\mathsf{T}|}\right)^{-\frac{q+1}{2}}\,. \tag{46}$$

Letting x denote $\mathbb{E}(|\mathcal{F}|)$ and $N = |\mathsf{T}|$, we have

$$\frac{1}{N - x} \leq \frac{1}{N}\left(1 - \frac{q}{N}\right)^{-\frac{q+1}{2}} \tag{47}$$

$$x \leq N\left[1 - \left(1 - \frac{q}{N}\right)^{\frac{q+1}{2}}\right]\,. \tag{48}$$

Applying Bernoulli's inequality, namely that $(1+x)^r \geq 1 + rx$, we have that

$$\left(1 - \frac{q}{N}\right)^{\frac{q+1}{2}} \geq 1 - \frac{q+1}{2} \cdot \frac{q}{N}\,, \tag{49}$$

hence

$$x \leq \frac{q(q+1)}{2}\,. \tag{50}$$

□

6 Conclusions, Limitations, and Open Problems

Using new analysis and attacks we have shown that, without further restrictions on the adversaries, Bernstein's analysis is in fact optimal. We can therefore conclude that the data limits imposed by Bernstein's bounds are necessary.

Our attacks illustrate for the first time how to maximally take advantage of tagging queries without needing verification queries in order to attack WCS authenticators. However, there are limitations on the applicability of the attacks.

As implied by the introduction, our attacks only work against polynomial-based WCS authenticators when they re-use the polynomial key, and is therefore not applicable to, for example, SNOW 3G [3GP17] or Poly1305 as used in NaCl [Ber09,Ber09].

The attacks work best when tags are not truncated, since the underlying PRP behaves more like a PRF with increased truncation [GG16,HWKS98]. However, as pointed out by Ferguson [Fer05] and Mattsson and Westerlund [MW16], one must take care when truncating tags in WCS authenticators. In some cases standards mandate that tags not be truncated [VM06,SMC08,IS09].

The attacks are not directly applicable to constructions which do not follow the WCS authenticator structure of mapping (n, m) to $\pi(n) + \rho(m)$. A few different constructions are discussed by Bernstein [Ber05c] and Handschuh and Preneel [HP08]. In particular, if a PRF instead of a PRP is used to hide the polynomial output, or if multiple PRP calls are XORed together as with CWC [KVW04] and GCM/2^+ [AY12], then the attacks are not applicable; it remains an open problem whether the analyses of the latter constructions are tight.

WCS authenticators can also be instantiated using non-polynomial-based universal hash functions, [BHK$^+$99,HK97,EPR99,Joh97,KYS05,Kro06,BHK$^+$99]. We expect that similar attacks are applicable to these functions.

As shown by Luykx, Mennink, and Paterson [LMP17], the attacks' success probability will not improve in the multi-key setting.

Finally, although our attacks show that one should abide by Bernstein's bounds, implementing the attacks seems to require a large amount of storage to achieve significant success probability. It is unclear whether there is a compact way of representing the set of false keys. Alternatively, if one were able to prove lower bounds on the storage requirements for any attacker, one could possibly afford to use keys beyond the data limits recommended by Bernstein's analysis, assuming adversaries have bounded storage capabilities.

Acknowledgments. The authors would like to thank Guy Barwell, Dan Bernstein, Bart Mennink, Scott Fluhrer, and the anonymous reviewers for their comments.

A Proof of Cor. 4.1

We re-use the notation and definitions from Sec. 4.3 and Sec. 4.4.

Corollary. Let $\gamma : (n, m) \mapsto \rho(m) + \pi(n)$ be a polynomial-based WCS authenticator with $\rho : \mathsf{M} \to \mathsf{T}$ a random function, and $\pi : \mathsf{N} \to \mathsf{T}$ an independent URP. Let $\mathbf{A}$ be an authenticity adversary against γ making at most q queries of length at most ℓ. Then $\mathbf{A}$'s advantage against γ is bounded by

$$v \cdot \max\left\{\ell \cdot \mathsf{KR}_\gamma(\mathbf{C}\langle\mathbf{A}\rangle), \frac{1}{|\mathsf{T}| - q}\right\}. \tag{51}$$

Proof. We restrict our attention to single-forgery adversaries, and use Thm. 2.1 to generalize to any authenticity adversary.

Let E denote the event that n^* does not equal a previous query to φ. By Prop. 4.1, the probability that $\mathbf{A}$ wins and E occurs is bounded above by the probability that $\mathbf{B}\langle\mathbf{A}\rangle$ wins, which is at most $1/(|\mathsf{T}| - q)$ by Lem. 4.1. By Prop. 4.2, the probability that $\mathbf{A}$ wins and E does not occur is bounded above by ℓ times the probability that $\mathbf{C}\langle\mathbf{A}\rangle$ wins. □

B From Key Recovery to Forgeries

Let $\mathbf{C}$ be a polynomial authenticator key recovery adversary against γ, then we construct an authenticity adversary $\mathbf{A}\langle\mathbf{C}\rangle$ against γ as follows:

1. $\mathbf{A}$ runs $\mathbf{C}$.
2. Every (n, m) query by $\mathbf{C}$ gets forwarded to $\mathbf{A}$'s oracle, and $\mathbf{A}$ returns the output $\gamma(n, m)$ to $\mathbf{C}$.
3. When $\mathbf{C}$ outputs k^*, $\mathbf{A}$ uses it to compute $y^* = \gamma(n_1, m_1) - p_{m_1}(k^*)$, where (n_1, m_1) is the first query made by $\mathbf{C}$. Then $\mathbf{A}$ picks a message m^*, and attempts the forgery $(n_1, m^*, y^* + p_{m^*}(k^*))$.

Proposition B.1.

$$\mathsf{KR}_\gamma(\mathbf{C}) \leq \mathsf{Auth}_\gamma(\mathbf{A}\langle\mathbf{C}\rangle). \tag{52}$$

Proof. If $\mathbf{C}$ wins its game, then $k^* = k$, the key used by the polynomial hash. Then we have

$$\begin{aligned}
\gamma(n_1, m^*) &= \pi(n_1) + p_{m^*}(k) && (53)\\
&= \gamma(n_1, m_1) - p_{m_1}(k) + p_{m^*}(k) && (54)\\
&= \gamma(n_1, m_1) - p_{m_1}(k^*) + p_{m^*}(k^*), && (55)
\end{aligned}$$

which is exactly the tag submitted by $\mathbf{A}$. □

C Notation

Table 1. List of notation.

Symbol	Description
Quantities	
v	number of verification queries
q	number of tagging queries
ℓ	maximum message length
N	size of T
Random Variables	
φ	random function
π	URP
γ	authenticator
ρ	polynomial-based universal hash
κ	key of a polynomial hash
Sets	
$x \in \mathsf{X}$	domain, block
$y \in \mathsf{Y}$	range
$k \in \mathsf{K}$	Key set
$n \in \mathsf{N}$	Nonce set
$m \in \mathsf{M}$	Message space
$t \in \mathsf{T}$	Tag space
$\mathcal{F}$	Faulty keys output by attacks
$\mathcal{T}$	Complement of $\mathcal{F}$, i.e. $\mathsf{K} \setminus \mathcal{F}$
Adversaries	
$\mathbf{A}$	Adversary (generic or authenticity)
$\mathbf{B}$	Unpredictability adversary
$\mathbf{C}$	Key recovery adversary
Miscellaneous	
$\boldsymbol{x}$	vector of elements
$p_m(k)$	polynomial defined by m evaluated at k

References

3GP17. Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2; Document 2: SNOW 3G specification, 2017. `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2396`.

ABBT15. Mohamed Ahmed Abdelraheem, Peter Beelen, Andrey Bogdanov, and Elmar Tischhauser. Twisted Polynomials and Forgery Attacks on GCM. *IACR Cryptology ePrint Archive*, 2015:1224, 2015.

AY12. Kazumaro Aoki and Kan Yasuda. The Security and Performance of "GCM" when Short Multiplications Are Used Instead. In Miroslaw Kutylowski and Moti Yung, editors, *Information Security and Cryptology - 8th International Conference, Inscrypt 2012, Beijing, China, November 28-30, 2012, Revised Selected Papers*, volume 7763 of *Lecture Notes in Computer Science*, pages 225–245. Springer, 2012.

BC09. John Black and Martin Cochran. MAC reforgeability. In Orr Dunkelman, editor, *Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers*, volume 5665 of *Lecture Notes in Computer Science*, pages 345–362. Springer, 2009.

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

Ber05a. Daniel J Bernstein. Stronger security bounds for permutations. `http://cr.yp.to/papers.html#permutations`, 2005. Date accessed 9 April, 2015.

Ber05b. Daniel J. Bernstein. Stronger Security Bounds for Wegman-Carter-Shoup Authenticators. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 164–180. Springer, 2005.

Ber05c. Daniel J. Bernstein. The Poly1305-AES Message-Authentication Code. In Henri Gilbert and Helena Handschuh, editors, *Fast Software Encryption: 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Revised Selected Papers*, volume 3557 of *Lecture Notes in Computer Science*, pages 32–49. Springer, 2005.

Ber09. Daniel J Bernstein. Cryptography in NaCl. `http://cr.yp.to/papers.html#naclcrypto`, 2009. Date accessed 14 September, 2017.

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 Wiener [Wie99], pages 216–233.

BJKS93. Jürgen Bierbrauer, Thomas Johansson, Gregory Kabatianskii, and Ben J. M. Smeets. On families of hash functions via geometric codes and concatenation. In Stinson [Sti94], pages 331–342.

BL16. Karthikeyan Bhargavan and Gaëtan Leurent. On the practical (in-)security of 64-bit block ciphers: Collision attacks on HTTP over TLS and openvpn. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 456–467. ACM, 2016.

BPR05. Mihir Bellare, Krzysztof Pietrzak, and Phillip Rogaway. Improved security analyses for CBC macs. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 527–545. Springer, 2005.

Bra82. Gilles Brassard. On computationally secure authentication tags requiring short secret shared keys. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology: Proceedings of CRYPTO '82,*

Santa Barbara, California, USA, August 23-25, 1982., pages 79–86. Plenum Press, New York, 1982.

CS16. Benoît Cogliati and Yannick Seurin. *EWCDM: An Efficient, Beyond-Birthday Secure, Nonce-Misuse Resistant MAC*, pages 121–149. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.

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*, 2:65–72, 1993.

EPR99. Mark Etzel, Sarvar Patel, and Zulfikar Ramzan. SQUARE HASH: fast message authenication via optimized universal hash functions. In Wiener [Wie99], pages 234–251.

Fer05. Niels Ferguson. Authentication weaknesses in GCM. *Comments submitted to NIST Modes of Operation Process*, 2005.

GG16. Shoni Gilboa and Shay Gueron. The advantage of truncated permutations. *CoRR*, abs/1610.02518, 2016.

GMS74. E. N. Gilbert, F. J. MacWilliams, and N. J. A. Sloane. Codes Which Detect Deception. *Bell System Technical Journal*, 53(3):405–424, 1974.

GPR14. Peter Gazi, Krzysztof Pietrzak, and Michal Rybár. The exact prf-security of NMAC and HMAC. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 113–130. Springer, 2014.

GPR16. Peter Gazi, Krzysztof Pietrzak, and Michal Rybár. The exact security of PMAC. *IACR Trans. Symmetric Cryptol.*, 2016(2):145–161, 2016.

GPT15. Peter Gazi, Krzysztof Pietrzak, and Stefano Tessaro. The exact PRF security of truncation: Tight bounds for keyed sponges and truncated CBC. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 368–387. Springer, 2015.

HK97. Shai Halevi and Hugo Krawczyk. MMH: Software Message Authentication in the Gbit/Second Rates. In Eli Biham, editor, *Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings*, volume 1267 of *Lecture Notes in Computer Science*, pages 172–189. Springer, 1997.

HP08. Helena Handschuh and Bart Preneel. Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms. In David Wagner, editor, *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*, pages 144–161. Springer, 2008.

HWKS98. Chris Hall, David A. Wagner, John Kelsey, and Bruce Schneier. Building PRFs from PRPs. In Hugo Krawczyk, editor, *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, volume 1462 of *Lecture Notes in Computer Science*, pages 370–389. Springer, 1998.

IOM12. Tetsu Iwata, Keisuke Ohashi, and Kazuhiko Minematsu. Breaking and Repairing GCM Security Proofs. In Reihaneh Safavi-Naini and Ran Canetti,

editors, *Advances in Cryptology CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 31–49. Springer Berlin Heidelberg, 2012.

IS09. Kevin Igoe and Jerome Solinas. AES Galois Counter Mode for the Secure Shell Transport Layer Protocol. RFC 5647, August 2009.

Joh97. Thomas Johansson. Bucket hashing with a small key size. In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, volume 1233 of *Lecture Notes in Computer Science*, pages 149–162. Springer, 1997.

Jou. Antoine Joux. Comments On The Draft GCM Specification – Authentication Failures in NIST version of GCM. `http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/comments/800-38_Series-Drafts/GCM/Joux_comments.pdf`.

Kro06. Ted Krovetz. Message authentication on 64-bit architectures. In Eli Biham and Amr M. Youssef, editors, *Selected Areas in Cryptography, 13th International Workshop, SAC 2006, Montreal, Canada, August 17-18, 2006 Revised Selected Papers*, volume 4356 of *Lecture Notes in Computer Science*, pages 327–341. Springer, 2006.

KVW04. Tadayoshi Kohno, John Viega, and Doug Whiting. CWC: A high-performance conventional authenticated encryption mode. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*, pages 408–426. Springer, 2004.

KYS05. Jens-Peter Kaps, Kaan Yüksel, and Berk Sunar. Energy scalable universal hashing. *IEEE Trans. Computers*, 54(12):1484–1495, 2005.

LMP17. Atul Luykx, Bart Mennink, and Kenneth G. Paterson. Analyzing multi-key security degradation. Cryptology ePrint Archive, Report 2017/435, 2017. `http://eprint.iacr.org/2017/435`.

LPSY16. Atul Luykx, Bart Preneel, Alan Szepieniec, and Kan Yasuda. On the influence of message length in pmac's security bounds. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 596–621. Springer, 2016.

MF05. David A. McGrew and Scott R. Fluhrer. Multiple forgery attacks against message authentication codes. Cryptology ePrint Archive, Report 2005/161, 2005. `http://eprint.iacr.org/2005/161`.

MN17. Bart Mennink and Samuel Neves. Encrypted davies-meyer and its dual: Towards optimal security using mirror theory. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 556–583. Springer, 2017.

MV04a. David A. McGrew and John Viega. The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In Anne Canteaut and Kapalee Viswanathan, editors, *INDOCRYPT*, volume 3348 of *Lecture Notes in Computer Science*, pages 343–355. Springer, 2004.

MV04b. David A. McGrew and John Viega. The Security and Performance of the Galois/Counter Mode of Operation (Full Version). *IACR Cryptology ePrint Archive*, 2004:193, 2004.

MW16. John Mattsson and Magnus Westerlund. Authentication key recovery on galois/counter mode (GCM). In David Pointcheval, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *Progress in Cryptology - AFRICACRYPT 2016 - 8th International Conference on Cryptology in Africa, Fes, Morocco, April 13-15, 2016, Proceedings*, volume 9646 of *Lecture Notes in Computer Science*, pages 127–143. Springer, 2016.

Nat80. National Institute of Standards and Technology. DES Modes of Operation. FIPS 81, December 1980.

NOMI15. Yuichi Niwa, Keisuke Ohashi, Kazuhiko Minematsu, and Tetsu Iwata. GCM Security Bounds Reconsidered. In Gregor Leander, editor, *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, volume 9054 of *Lecture Notes in Computer Science*, pages 385–407. Springer, 2015.

PC13. Gordon Procter and Carlos Cid. On Weak Keys and Forgery Attacks Against Polynomial-Based MAC Schemes. In Shiho Moriai, editor, *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*, volume 8424 of *Lecture Notes in Computer Science*, pages 287–304. Springer, 2013.

PC15. Gordon Procter and Carlos Cid. On Weak Keys and Forgery Attacks Against Polynomial-Based MAC Schemes. *J. Cryptology*, 28(4):769–795, 2015.

Pie06. Krzysztof Pietrzak. A tight bound for EMAC. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 168–179. Springer, 2006.

PvO99. Bart Preneel and Paul C. van Oorschot. On the security of iterated message authentication codes. *IEEE Trans. Information Theory*, 45(1):188–199, 1999.

Saa11. Markku-Juhani O. Saarinen. SGCM: The Sophie Germain Counter Mode. Cryptology ePrint Archive, Report 2011/326, 2011. `http://eprint.iacr.org/2011/326`.

Saa12. Markku-Juhani Olavi Saarinen. Cycling Attacks on GCM, GHASH and Other Polynomial MACs and Hashes. In Anne Canteaut, editor, *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, volume 7549 of *Lecture Notes in Computer Science*, pages 216–225. Springer, 2012.

Sho96. Victor Shoup. On Fast and Provably Secure Message Authentication Based on Universal Hashing. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 313–328. Springer, 1996.

Sim91. G.J. Simmons. A Survey of Information Authentication. In G.J. Simmons, editor, *Contemporary Cryptology: The Science of Information Integrity*, pages 381–419. IEEE Press, 1991.

SMC08. Joseph A. Salowey, Dr. David A. McGrew, and Abhijit Choudhury. AES Galois Counter Mode (GCM) Cipher Suites for TLS. RFC 5288, August 2008.

Sti91. Douglas R. Stinson. Universal hashing and authentication codes. In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 74–85. Springer, 1991.

Sti94. Douglas R. Stinson, editor. *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*. Springer, 1994.

Tay93. Richard Taylor. An integrity check value algorithm for stream ciphers. In Stinson [Sti94], pages 40–48.

VM06. John Viega and Dr. David A. McGrew. The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH. RFC 4543, May 2006.

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.

Wie99. Michael J. Wiener, editor. *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*. Springer, 1999.

ZTG13. Bo Zhu, Yin Tan, and Guang Gong. *Revisiting MAC Forgeries, Weak Keys and Provable Security of Galois/Counter Mode of Operation*, pages 20–38. Springer International Publishing, Cham, 2013.

ZW17. Kaiyan Zheng and Peng Wang. A uniform class of weak keys for universal hash functions. Cryptology ePrint Archive, Report 2017/436, 2017. `http://eprint.iacr.org/2017/436`.