Lightweight CRC-based Message Authentication

Elena Dubrova[1], Mats Näslund[2], Göran Selander[2], and Fredrik Lindqvist[2]

[1] Royal Institute of Technology, Electrum 229, 164 40 Stockholm, Sweden
dubrova@kth.se
[2] Ericsson Research, Ericsson, Färögatan 6, 164 80 Stockholm, Sweden
{mats.naslund,goran.selander,fredrik.lindqvist}@ericsson.com

Abstract. Low-cost resource-constrained devices can allocate very limited resources for implementing security. At the same time, they still require some level of protection. In this paper, we present a lightweight message authentication scheme based on Cyclic Redundancy Check (CRC). The presented CRC inherits the implementation simplicity of the conventional CRC checksum except that the LFSR implementing its encoding and decoding is made re-programmable. Similarly to previously proposed cryptographic CRCs, it detects both random and malicious errors without increasing bandwidth. The main difference from previous approaches is that we use arbitrary instead of irreducible generator polynomials. This eliminates the need for irreducibility tests. We provide a detailed quantitative analysis of the achieved security as a function of message and CRC sizes. The results show that the presented scheme is particularly suitable for the authentication of short messages.

Keywords: Hash function; message authentication; data integrity protection; CRC; error-detection.

1 Introduction

A rapid growth of *Internet-of-Things* (IoT) applications that use cellular network infrastructure is expected in the coming years. Home appliances, meters, sensors, are vehicles will be accessible and controlled via local networks or the Internet, opening an entirely new range of services designed to be appealing to users. The number of wirelessly connected devices is expected to grow to 50 billions by the year 2020 [1].

Unfortunately, new technologies are appealing to the attackers as well. Attacks are becoming more frequent, more sophisticated, and more widespread. In 2014, the annual cost to the global economy from cybercrimes was more than $400 billion [2]. While it is difficult to hack into a household appliance which is not coupled to anything but a power plug, a connected appliance becomes a target for all attackers around the globe unless appropriate security mechanisms are implemented and utilized. Household appliances typically do not have the same level of protection as computer systems. A compromised device can potentially be used as an entry point for cyberattacks on other devices connected to the network. The first proven cyberattack involving "smart" household appliances has been already reported [3]. The attack surface of future IoT with billions of connected devices will be enormous.

In addition to a larger attack surface, the return value for performing an attack grows. The assets accessible via tomorrow's networks are expected to be much greater than the ones available today, increasing incentive for cyber criminals and underground economies. As processing power and connectivity become cheaper, the cost of performing an attack drops, making it easier for adversaries of all types to penetrate networks. Considering the possible consequences of an attack on safety-critical applications such as automotive, industrial control, military and health, the damage caused by an individual actor may not be limited to a business or reputation, but could have a severe impact on public safety, national economy, and/or national security.

Low-cost end-point IoT devices will require utmost efficiency in the use of communication, computing, storage and energy resources. A typical IoT device is likely to spend most of its "life" in a sleep mode. It will get activated at periodic intervals, transmit a small amount of data and then shut down again. To satisfy extreme limitations of resource-constrained devices, lightweight cryptographic methods are required. It seems also beneficial to combine coding and cryptographic techniques, e.g. error detection and data integrity protection, since this may potentially reduce bandwidth and implementation cost.

Clearly, data integrity protection can be implemented by using some n-bit message authentication code, e.g. keyed Hash Message Authentication Code (HMAC) [4] or Cipher Block Chaining Message Authentication Code (CBC-MAC) [5], *on the top* of an error detection code, e.g. n-bit Cyclic Redundancy Check (CRC) [6]. However, such approach expands the message by n bits and requires a separate encoding/decoding engine which is more complex than the CRC encoding/decoding engine. On the other hand, if we simply *replace* an n-bit CRC with an n-bit HMAC or CBC-MAC, then we cannot guarantee the detection of the same type of random errors as the CRC. For example, the detection of n-bit burst errors[3] cannot be guaranteed. This may have a negative impact on the reliability of communication link. Only if we make the conventional CRC cryptographically secure, we can assure a certain level of security without sacrificing reliability.

The latter motivated the development of cryptographically secure CRC checksums. The core idea is to make the CRC generator polynomial variable and secret. The CRC presented by Krawczyk [7] is based on random irreducible generator polynomials. The approach described in [8] uses a product of random irreducible polynomials. The CRC proposed in [9] uses generator polynomials of type $(1+x)p(x)$, where $p(x)$ is a random primitive polynomial. In all three cases, testing for irreducibility or primitivity is required, which is either time or memory consuming. Selecting a random irreducible polynomial of degree n requires either randomly selecting a polynomial of degree n ($O(n)$ time) and running a test for irreducibility ($\Omega(n^3)$ time [10]), or randomly selecting a polynomial of

[3] A *burst* error is an error affecting adjacent bits.

degree n from a database of irreducible polynomials of degree n (roughly $2^n/n$ space).

In this paper, we propose cryptographically secure CRC based on arbitrary random generator polynomials, with no requirements on irreducibility. We provide a detailed quantitative analysis of the achieved security as a function of message and CRC sizes. To our best knowledge, no security analysis for the general case of reducible polynomials has been made so far. This might be due the fact that the evaluation involves estimating the maximum number of reducible polynomials which can be constructed from any multiset of irreducible polynomials of a given size, which is a non-trivial task.

The paper is organized as follows. Section 2 summarizes basic notation and definitions used in the paper. Section 3 gives a background on CRC codes. Section 4 describes previous work. In Section 5, we introduce two new families of cryptographically secure CRC hash functions. Section 6 analyses error-detecting capabilities of hash families. In Section 7, we present the security analysis of hash families. Section 8 shows experimental results. Section 9 concludes the paper.

2 Preliminaries

In this section we describe properties of hash functions which are required for the proof of the main result.

Throughout the paper, we associate each binary string $L \in \{0,1\}^l$ representing an l-bit message with a polynomial $L(x)$ over the Galois Field of the order 2, $GF(2)$, so that the coefficients of $L(x)$ correspond to the bits of L. We use $deg(L(x))$ to denote the degree of the polynomial $L(x)$.

Definition 1. [7] *A family of hash functions H is $\oplus$-linear if, for all messages L_1 and L_2 and for all $h \in H$, we have*

$$h(L_1 \oplus L_2) = h(L_1) \oplus h(L_2),$$

where "$\oplus$" is the bitwise exclusive-OR (XOR).

Definition 2. [7] *A family of hash functions H is ϵ-balanced if, for any non-zero l-bit message L and for any l-bit string a,*

$$\forall h \in H, \Pr[h(L) = a] \leq \epsilon,$$

where the probability is taken over h chosen uniformly at random from the family H.

Definition 3. [7] *A family of hash functions is ϵ-opt-secure if, for any message L, no adversary can generate another message L' with a valid authentication tag with probability larger than ϵ when the tag is computed as $h(L')$ for a randomly chosen h.*

Theorem 1. **[7]** *A necessary and sufficient condition for a family H of hash functions to be ϵ-opt-secure is that*

$$\forall L_1 \neq L_2, \forall a \in \{0,1\}^l, \forall h \in H, \Pr[h(L_1) \oplus h(L_2) = a] \leq \epsilon.$$

Theorem 2. **[7]** *If H is $\oplus$-linear, then H is ϵ-opt-secure if and only if H is ϵ-balanced.*

3 Cyclic Redundancy Check

A Cyclic Redundancy Check (CRC) is widely used for protecting data communication or storage against random errors [6]. Many wireless communication standards use CRC. For example, IEEE 802.15.4 standard uses 16-bit CRC [11], LTE uses 24-bit CRC [12], and GSM uses 40-bit CRC [13].

To perform n-bit CRC encoding, a message polynomial, $L(x)$, is multiplied by x^n and then divided modulo a generator polynomial $g(x)$ of degree n. The coefficients of the resulting polynomial

$$r(x) = L(x) \cdot x^n \ mod \ g(x)$$

represent the check bits of the CRC. These check bits are added to $L(x) \cdot x^n$ to get the resulting CRC codeword $L(x) \cdot x^n + r(x)$.

The CRC decoding is usually done by dividing the received message polynomial modulo the generator polynomial $g(x)$ and comparing the coefficients of the resulting remainder to the received CRC check bits. A disagreement indicates an error. It is well-known that if an irreducible generator polynomial of degree n is used as a generator polynomial, then the resulting CRC detects all burst errors of length n or less [14].

The CRC encoding and decoding can be efficiently implemented using a Linear Feedback Shift Register (LFSR) [15] having $g(x)$ as its connection polynomial. There are many efficient techniques for speeding up the computation of CRC [16].

Traditional CRCs are good at detecting random errors. However, they are not suitable for detecting malicious errors. An adversary who knows the generator polynomial $g(x)$ may simply substitute the original message $L(x)$ by another message $L'(x)$, encode $L'(x)$ as usual into the codeword $L'(x) \cdot x^n + r(x)$, where $r(x) = L'(x) \cdot x^n \ mod \ g(x)$, and then submit the resulting codeword. The receiver will not be able to distinguish the codeword $L'(x) \cdot x^n + r(x)$ from the codeword received from a legitimate sender.

4 Related Work

Message Authentication Codes (MACs) have also been thoroughly investigated in the past, see Simmons for an excellent survey [17]. A *message authentication* algorithm accepts as input a secret key and a message to be authenticated and outputs an *authentication tag.* The tag protects both, message data integrity and message authenticity. It allows legitimate users, who possess the secret key, to detect any changes in the message content.

Security of several types of MACs, including HMAC [18], CBC-MAC [5] and XOR-MAC [19], have been quantitative analyzed.

Unconditionally secure message authentication codes were introduced by Gilbert et al. [20] and their theoretical basis was developed by Simmons [21].

Carter and Wegman [22] were first to show that hash functions can be combined with one-time pads to construct strong authentication algorithms. Their approach was further developed by Brassared [23], Desmedt [24] and Krawczyk [7].

Stinson [25] has introduced the notion of "almost strongly universal hash families" which helped to considerably reduce the key size of unconditionally secure MACs. For more details on universal hashing, see his fundamental paper [26]. Black et al. have shown that universal hash families can be applied to construct efficient computationally secure MACs, e.g. UMAC [27]. Computationally secure MACs are used in 3G wireless communication.

Various techniques for cryptographic checksums and MACs based on stream ciphers have been proposed, including Lai et. al. [28], Taylor [29], Johansson [30] and [31]. In these techniques, a new hash function from a hash family is produced for every message by using the pseudo-random generator of a stream cipher. In the scheme presented in this paper, as well as in the method of Krawczyk [7], the same hash function can be re-used for multiple messages. Only the random pad which is used for the encryption of the hash values needs to be updated for each message.

Rabin [32] was first to use CRCs in the cryptographic context for the fingerprinting of information. However, in his scheme the modular division by the generator polynomial is applied directly to a message, without shifting the message n bit positions left first. As a result, Rabin's scheme is non-secure for message authentication even if the fingerprint is encrypted using a perfect one-time pad [7]. For example, if some of the least significant bits of the message together with the corresponding bits of the encrypted authentication tag are flipped, the change will not go undetected by the fingerprint.

Krawczyk [7] has proved that the inclusion of the n-bit shift into Rabin's scheme [32] makes the scheme secure for message authentication provided that tag is encrypted using a one-time pad. He has shown that the probability of breaking the resulting an authentication scheme is $\epsilon \leq \frac{l+n}{2^{n-1}}$, where n is CRC length and l is message length.

In [8] Krawczyk's approach has been extended to the case when a product of k irreducible polynomials is used to generate the CRC. The collision probability of such an authentication scheme is $\epsilon \leq \frac{(l+n)^k}{2^{n-k}}$.

In [9] generator polynomials of type $(1+x)p(x)$, where $p(x)$ is a primitive polynomial, are used to generate the CRC. Such CRCs are able to detect all double-bit errors in a message, which is of importance for systems using Turbo codes, including LTE. The collision probability in this case is $\epsilon \leq \frac{l+n-1}{2^{n-2}}$.

Apart from CRC, other error detecting/correcting codes have also been proposed for message authentication. MACs based on BCH and Reed-Solomon error-correcting codes have been presented in [33]. Approximate MACs, intended to tolerate a small number of errors in a message, have been introduced in [34] and [35].

5 Two families of cryptographically secure CRC hash functions

In this section, we define two new families of cryptographically secure CRC-based hash functions.

Definition 4 (Family $HR_{l,n}$). *For any message L of binary length l and for each polynomial $g(x)$ of degree n over $GF(2)$, a hash function $h_g(L)$ is defined as the 0/1 coefficients of the polynomial*

$$h_g(L) = L(x) \cdot x^n \ mod \ g(x).$$

The $HR_{l,n}$ family consists of the set of all hash functions $h_g(L)$.

Since each polynomial of degree n over $GF(2)$ defines one member of the family $HR_{l,n}$ and there are 2^n polynomials of degree n over $GF(2)$, the size of the family $HR_{l,n}$ is 2^n.

The can be used for

To authenticate a message L using the hash function family $HR_{l,n}$, a sender computes the authentication tag t as described next, appends t to L, and transmits the message and the appended tag. A receiver authenticates a received message $L'(x)$ (potentially different from $L(x)$) by re-computing the tag for $L'(x)$ and comparing the received and the re-computed tags. A disagreement implies an error.

An authentication tag t is computed as

$$t = h_g(L) \oplus s,$$

where the selection of the generator polynomial $g(x)$ for the hash function $h_g(L)$ and the pad s is made pseudo-randomly from the set of all possible polynomials of degree n over $GF(2)$ and the set of all binary n-tuples, respectively, based on a shared secret known to sender and receiver. The shared secret can be established, for example, by public key techniques or symmetric techniques using the traditional methods [36].

The addition of the random pad s is required to prevent the injection of all-0 messages by an attacker. Without s, CRC of an all-0 message is 0 independently of $g(x)$. The reader familiar with e.g. the UIA2 MAC of the 3G standard will recognize this type of construction. In that case, s is generated by the SNOW3G stream cipher [37].

We also consider separately a special case of the Definition 4 when the generator polynomial has a non-zero constant term. This case is particularly interesting because, as we show in the next section, CRCs based on such polynomials detect the same type of burst errors as CRCs based on irreducible polynomials.

Definition 5 (Family $HRC_{l,n}$). *For any binary message L of length l and for each polynomial $q(x)$ of degree n over $GF(2)$ with a non-zero constant term, a hash function $h_q(L)$ is defined as the 0/1 coefficients of the polynomial*

$$h_q(L) = L(x) \cdot x^n \ mod\ q(x).$$

The $HRC_{l,n}$ family consists of the set of all hash functions $h_q(L)$.

Since each polynomial of degree n over $GF(2)$ with a non-zero constant term defines one member of the family $HRC_{l,n}$ and there are 2^{n-1} polynomials of degree n over $GF(2)$ which have a non-zero constant term, the size of the family $HRC_{l,n}$ is 2^{n-1}.

Similarly to the family $HR_{l,n}$, the authentication tag for the family $HRC_{l,n}$ is computed as

$$t = h_q(L) \oplus s,$$

where s is a pseudo-random pad of length n.

The computation of CRCs defined above is based on the same operation of polynomial modular division as the traditional CRCs except that, in our case, the generator polynomial is changed periodically to appear random to an adversary. In general, it is sufficient to update the generator polynomial at the beginning of each session and keep it fixed for all messages. The pad s, however, has to be changed for each message.

The implementation of CRC encoding and decoding in hardware is very simple and efficient. The operation of polynomial modular division in $GF(2)$ can be realized using an n-bit LFSR with taps defined by the generator polynomial of degree n. Since the same operation is used for traditional CRCs, there are many references regarding the implementation, e.g. [6,38,39]. A difference is that in the traditional CRC the generator polynomial is fixed and the LFSR implementing the encoding/decoding usually has the taps hardwired into the circuit. A cryptographic CRC requires an LFSR with re-programmable connections. Techniques for implementing such LFSRs are known because non-cryptographic CRC implementations which need to support different CRC standards with different generator polynomials also use re-programmable LFSRs [40].

It is important to point out that restricting arbitrary polynomials to arbitrary polynomials with non-zero constant terms does not complicate the implementation of CRC in any way. The only difference is that, for polynomials with non-zero constant terms, the LFSR connection corresponding to the constant-one term of the polynomial should be made fixed rather than programmable.

6 Analysis of Error-Detecting Capabilities

It is well-known that a CRC based on an irreducible generator polynomial of degree n detect all burst errors on length n or less [14]. Next, we show that a cryptographically secure CRC based on an arbitrary generator polynomial of degree n with a non-zero constant term detects the same type of errors.

Theorem 3. *A CRC based on an arbitrary generator polynomial of degree $n > 1$ with a non-zero constant term detects the same type of burst errors as a CRC based on an irreducible generator polynomial of degree n.*

Proof: Let L be an l-bit message and let the CRC check bits be computed according to the Definition 5 using the generator polynomial $q(x)$ of degree n. Any k-bit burst error e, $0 < k \leq n$, can be described by a polynomial of type

$$e(x) = x^j \cdot f(x) \tag{1}$$

where

$$f(x) = x^{k-i-1} + x^{k-i-2} + \ldots + x + 1,$$

for $i \in \{0, 1, \ldots, k-1\}$ and $j \in \{0, 1, \ldots, l+i\}$.

The error e is not detected by the CRC if and only if $e(x)$ is evenly divisible by the generator polynomial $q(x)$. Since $q(x)$ has a non-zero constant term and the polynomial x^j in (1) does not have a non-zero constant term (except for the case $j = 0$), $q(x)$ cannot evenly divide x^j. So, $e(x)$ is evenly divisible by $q(x)$ if and only if $f(x)$ is evenly divisible by $q(x)$. However, this is not possible since the degree of $f(x)$ is at least by 1 smaller than the degree of $q(x)$, n. Therefore, a CRC based on a generator polynomial of degree n with a non-zero constant term detects all burst errors on length n or less.

□

Theorem 3 shows that, from the point of view of correcting random burst errors, no advantages are lost if an irreducible polynomial is replaced by an arbitrary polynomial with a non-zero constant term.

7 Security Analysis

In this section, we analyze the security of the new families of hash functions. We assume a typical setting in which the sender and the receiver transmit messages over an unsecure channel where messages can be maliciously modified [36]. The sender and the receiver share a secret key which is unknown to the adversary.

7.1 Quantifying collision probability

It is assumed that an adversary breaks the authentication if, after observing the message L and the tag t, he/she can find L' and t' such that $L' \neq L$ and $t' = h(L') \oplus s$. It is also assumed that the adversary knows the family of hash functions, but not the particular polynomial $g(x)$ and the pad s which are used to generate the authentication tag t.

In order to quantify the probability that an adversary can succeed to break the authentication based on the hash functions family $HR_{l,n}$, we first introduce some definitions.

Let P be a multiset of irreducible polynomials over $GF(2)$. In a multiset, the same element may repeat more than once. By $mult(p)$ we denote the number of occurrences of a polynomial p in P. By $size(P)$ we denote be the sum of degrees of all elements of P. For example, for $P = \{x, x, x+1, x^2+x+1\}$, $size(P) = 5$.

Let $N(n;P)$ denote the number of distinct polynomials of degree n which can be constructed from the elements of P. $N(0;P)$ is defined to be 1 for any P. Since each polynomial has a unique factorization into irreducible polynomials, $N(n;P)$ can be computed by counting the number of distinct combinations of elements of P whose degrees sum up to n. We return to this problem in Section 7.2.

For a given n, let P_{max} be a multiset of irreducible polynomials such that

$$N(n;P_{max}) \geq N(n;P)$$

for any other multiset P with $size(P) = size(P_{max})$.

The following Theorem quantifies the probability that an adversary can succeed to break the authentication based on the hash functions family $HR_{l,n}$.

Theorem 4. *For any values of n and l, the family of hash functions $HR_{l,n}$ is ϵ_1-opt-secure for*

$$\epsilon_1 \leq \frac{N(n;P_{max})}{2^n}, \tag{2}$$

where $size(P_{max}) = n + l$.

Proof: A family of hash functions is ϵ-opt-secure if it is $\oplus$-linear and ϵ-balanced. The family of hash functions $HR_{l,n}$ is $\oplus$-linear because for all messages L_1 and L_2 and for all $h_g \in HR_{l,n}$, we have $h_g(L_1 \oplus L_2) = h_g(L_1) \oplus h(L_2)$.

To show that the family $HR_{l,n}$ is also ϵ-balanced, we observe that, on one hand, for any polynomial $g(x)$ of degree n over GF(2), any non-zero message L of length l and any string a of length n, $h_g(L) = a$ if and only if $L(x) \cdot x^n \ mod \ g(x) = a(x)$. On the other hand, $L(x) \cdot x^n \ mod \ g(x) = a(x)$ if and only if $g(x)$ divides $L(x) \cdot x^n - a(x)$.

Let $f(x) = L(x) \cdot x^n - a(x)$. Obviously, $f(x)$ is a non-zero polynomial of degree less than or equal to $l + n$, and $g(x)$ is a polynomial of degree n which divides $f(x)$. On one hand, there are at most $N(n;P_{max})$ hash functions in the family $HR_{l,n}$ that map L into a, because $N(n;P_{max})$ is the maximum number of distinct polynomials of degree n which can be constructed from the irreducible factors of any polynomial of degree $n + l$. On the other hand, the family $HR_{l,n}$ consists of 2^n elements (the number of polynomials of degree n over GF(2)). Therefore

$$\Pr[h_g(L) = a] \leq \frac{N(n;P_{max})}{2^n}.$$

□

In a similar way we can quantify the probability that an adversary can succeed to break the authentication based on the hash functions family $HRC_{l,n}$.

Let P^* be a multiset of irreducible polynomials with a non-zero constant term over $GF(2)$. For a given n, let P^*_{max} be a multiset of irreducible polynomials with a non-zero constant term such that

$$N(n; P^*_{max}) \geq N(n; P^*)$$

for any other multiset P^* with $size(P^*) = size(P^*_{max})$.

Theorem 5. *For any values of n and l, the family of hash functions $HRC_{l,n}$ is ϵ_2-opt-secure for*

$$\epsilon_2 \leq \frac{N(n; P^*_{max})}{2^{n-1}}, \tag{3}$$

where $size(P^*_{max}) = n + l$.

Proof: Similar to the proof of Theorem 4.

In the following sections, we show how to compute $N(n; P_{max})$ and $N(n; P^*_{max})$.

7.2 Number of polynomials which can be constructed from a given set of irreducible polynomials

Let I_i be the number of distinct irreducible polynomials of degree i over $GF(2)$. It is well-known how to compute I_i [41].

Let $p_{i,j}$ be jth irreducible polynomial of degree i, for all $j \in \{1, 2, ..., I_i\}$. Note that for our purpose we only need to enumerate all irreducible polynomials of a given degree. The order in which they are assigned the index j does not matter. So, whether we assign $p_{1,1} = x$ and $p_{1,2} = x + 1$ or vice versa does not change the presented results.

As we mentioned in the previous section, for a given n and a given multiset of irreducible polynomials P, the number of distinct polynomials of degree n which can be constructed from the elements of P, $N(n; P)$, can be computed by counting the number of distinct combinations of elements of P whose degrees sum up to n.

As an example, consider a multiset P with contains five copies of the polynomial $p_{1,1} = x$, five copies of the polynomial $p_{1,2} = x + 1$ and two copies of the polynomial $p_{2,1} = x^2+x+1$. Let $n = 5$. Then, the following set of 12 polynomials can be constructed from the elements of P:

$$\begin{gathered} x^5, x^4(x+1), x^3(x+1)^2, x^2(x+1)^3, x(x+1)^4, (x+1)^5 \\ x^3(x^2+x+1), x^2(x+1)(x^2+x+1), x(x+1)^2(x^2+x+1), (x+1)^3(x^2+x+1) \\ x(x^2+x+1)^2, (x+1)(x^2+x+1)^2. \end{gathered}$$

So, $N(5; P) = 12$.

Next, we show that $N(n; P)$ can be computed using a recurrence relation given by the following Lemma. Let $deg(p)$ denote the degree of a polynomial p. It is obvious that elements p with $mult(p) > \lfloor \frac{n}{deg(p)} \rfloor$ do not contribute to the new polynomials of degree n. For this reason the index m in the Lemma is limited by $\lfloor \frac{n}{deg(p)} \rfloor$.

Lemma 1. *For any multiset of irreducible polynomials P, any irreducible polynomial $p \notin P$ of degree $deg(p) \leq n$, and any m such that $1 \leq m \leq \lfloor \frac{n}{deg(p)} \rfloor$, it holds that*

$$N(n; P \cup \{p^m\}) = \sum_{i=0}^{m} N(n - i \cdot deg(p); P),$$

where $\{p^m\}$ denotes a multiset containing m elements p.

Proof: By induction on m.
1. Base case: $m = 1$. We need to prove that

$$N(n; P \cup \{p\}) = N(n; P) + N(n - deg(p); P).$$

By subtracting $N(n; P)$ from both sides we get

$$N(n; P \cup \{p\}) - N(n; P) = N(n - deg(p); P).$$

The left-hand side is the difference between the number of distinct polynomials of degree n which can be constructed from the elements of $P \cup \{p\}$ and the number of distinct polynomials of degree n which can be constructed from the elements of P. This difference is equal to the number of distinct polynomials of degree n which contain p as a factor with the multiplicity exactly one. Removing factor p from each of such polynomials yield all possible distinct polynomials of degree $n - deg(p)$ which can be constructed from the elements of P, i.e. the right-hand side $N(n - deg(p); P)$.
2. Inductive step: Assume the statement holds for m. Next we prove that it holds for $m + 1$, i.e. that

$$\begin{aligned} N(n; P \cup \{p^{m+1}\}) &= \sum_{i=0}^{m+1} N(n - i \cdot deg(p); P) \\ &= N(n; P \cup \{p^m\}) + N(n - (m+1) \cdot deg(p); P) \end{aligned}$$

where $1 \leq m + 1 \leq \lfloor \frac{n}{deg(p)} \rfloor$.

By subtracting $N(n; P \cup \{p^m\})$ from both sides we get

$$N(n; P \cup \{p^{m+1}\}) - N(n; P \cup \{p^m\}) = N(n - (m+1) \cdot deg(p); P).$$

The left-hand side is the difference between the number of distinct polynomials of degree n which can be constructed from the elements of $P \cup \{p^{m+1}\}$ and the number of distinct polynomials of degree n which can be constructed from the elements of $P \cup \{p^m\}$. The former accounts for factorizations which contain p with multiplicity from 0 to $mult(p) + 1$. The latter accounts for all factorizations which contain p with multiplicity from 0 to $mult(p)$. Therefore, the difference is equal to the number of distinct polynomials of degree n which contain p as a factor with the multiplicity exactly $mult(p) + 1$. Removing the factor p with the multiplicity $mult(p) + 1$ from each of such polynomials yield all possible distinct polynomials of degree $n - (mult(p) + 1) \cdot deg(p)$ which can be constructed from the elements of P, i.e. the right-hand side $N(n - (mult(p) + 1) \cdot deg(p); P)$.

□

Next, we derive a general formula for $N(n;P)$. In the derivations below we denote by P_d a multiset of irreducible polynomials in which the maximum degree of elements is d. To unify the notation, we allow multiplicity of elements of P to be 0. In this way, any P_d can be uniquely represented by the vector of multiplicities of its elements

$$(m_{1,1},\dots,m_{1,I_1},m_{2,1},\dots,m_{2,I_2},\dots,m_{d,1},\dots,m_{d,I_d}),$$

where $m_{i,j} = mult(p_{i,j})$ for all $i \in \{1,2,\dots,d\}$ and $j \in \{1,2,....,I_i\}$.

There are two irreducible polynomials of degree 1. It is easy to see that

$$N(n;P_1) = \begin{cases} min(m_{1,1},n) + min(m_{1,2},n) - n + 1, & \text{if } m_{1,1} + m_{1,2} \geq n \\ 0, & \text{otherwise} \end{cases}$$

There is only one irreducible polynomial of degree 2. From Lemma 1, we can conclude that

$$N(n;P_2) = N(n;P_1)+N(n-2;P_1)+N(n-4;P_1)+\ldots+N(n-2\cdot min(m_{2,1},\lfloor\frac{n}{2}\rfloor);P_1)$$

or

$$N(n;P_2) = \sum_{i_{2,1}=0}^{min(m_{2,1},\lfloor\frac{n}{2}\rfloor)} N(n-2i_{2,1};P_1)$$

It is straightforward to extend the derivations above to the following result.

Theorem 6. *For $d = 1$*

$$N(n;P_1) = \begin{cases} min(m_{1,1},n) + min(m_{1,2},n) - n + 1, & \textit{if } m_{1,1} + m_{1,2} \geq n \\ 0, & \textit{otherwise} \end{cases}$$

and for $d > 1$

$N(n;P_d) =$

$$\sum_{i_{d,1}=0}^{A_{d,1}} \sum_{i_{d,2}=0}^{A_{d,2}} \cdots \sum_{i_{d,I_d}=0}^{A_{d,I_d}} \cdots \sum_{i_{2,1}=0}^{A_{2,1}} N(n - \sum_{h=2}^{d}\sum_{j=1}^{I_h} i_{h,j}; P_1) \tag{4}$$

where

$$\begin{aligned} A_{d,1} &= min(\lfloor\tfrac{n}{d}\rfloor, m_{d,1}) \\ A_{d,2} &= min(\lfloor\tfrac{n-d\cdot i_{d,1}}{d}\rfloor, m_{d,2}) \\ &\cdots \\ A_{d,I_d} &= min(\lfloor\frac{n-d\sum_{j=1}^{I_d-1} i_{d,j}}{d}\rfloor, m_{d,I_d}) \\ &\cdots \\ A_{2,1} &= min(\lfloor\tfrac{n-S(d:3)}{2}\rfloor, m_{2,1}); \end{aligned}$$

where $S(d:i) = \sum_{r=i}^{d} \left(r \cdot \sum_{j=1}^{I_r} i_{r,j} \right)$.

All the results derived above also apply to the case of P^* being a multiset of irreducible polynomials with non-zero constant term except that, in Theorem 6, $N(n; P_1^*)$ reduces to

$$N(n; P_1^*) = \begin{cases} 1, \text{ if } m_{1,2} \geq n \\ 0, \text{ otherwise.} \end{cases}$$

7.3 Computing maximum $N(n;P)$

Theorem 6 shows us how to compute $N(n;P)$ for a given n and P. However, we do not know how to select a multiset of irreducible polynomials P which maximizes $N(n;P)$ for a given n and $size(P)$, i.e. how to get P_{max}. In this section we derive some properties of P_{max} which help us to compute it.

Property 1. For any $n > 0$, there exist P_{max} such that an irreducible polynomial p_i with $deg(p_i) = i$ is contained in P_{max} only if each irreducible polynomial p_j with $deg(p_j) = j$, $1 \leq j < i$, is contained in P_{max} at least once.

Proof: Suppose that $p_i \in P_{max}$ and $p_j \notin P_{max}$ fore some $j < i$. Then we can replace P_{max} by P' such that

$$P' = (P_{max} - \{p_i\}^{mult(p_i)}) \cup \{p_j\}^{mult(p_i)} \cup \{p_{i-j}\}^{mult(p_i)}$$

where p_{i-j} is any irreducible polynomial of degree $i-j$. Obviously, $size(P') = size(P_{max})$. Furthermore, for any polynomial of degree n constructed from the elements of P_{max} which contains p_i^k as a factor, we can replace p_i^k by $p_j^k \cdot p_{i-j}^k$, for any $1 \leq k \leq mult(p_i)$. Since $p_j^k \notin P_{max}$, this implies that $N(n;P') \geq N(n;P_{max})$.

□

For P_{max} satisfying the condition of Property 1, we can derive a rough upper bound on the maximum degree of polynomials contained in P_{max} by computing the smallest integer d satisfying

$$size(P_{max}) \leq I_1 + 2I_2 + 3I_3 + \ldots + dI_d. \tag{5}$$

We can reduce the search space for P_{max} by first deriving an upper bound on d using (5) and then removing from the consideration multisets P in which do not satisfying the condition of Property 1. We also can take into account that the order of elements of the same degree in a multiset does not matter.

Property 2. For any two multisets P and P' with $size(P) = size(P')$ which are equivalent up to a permutation of elements of the same degree, $N(n;P) = N(n;P')$.

As an example, suppose that $n = 2$ and $size(P) = 4$. From $4 \leq 2 + 2 \cdot 1$ we get $d = 2$. There are four possible candidates into P_{max} defined by the following vectors of multiplicities $(m_{1,1}, m_{1,2}, m_{2,1})$:

$$(2, 2, 0), (2, 0, 1), (0, 2, 1), (1, 1, 1).$$

Recall that elements p with $mult(p) > \lfloor \frac{n}{deg(p)} \rfloor$ do not contribute to new constructions of polynomials of degree n, therefore vectors (4,0,0), (0,4,0), (3,1,0), (1,3,0), and (0,0,2) are not included in the list.

By applying Properties 1 and 2, we can reduce the set of candidates into P_{max} to two:

$$(2, 2, 0), (1, 1, 1).$$

Now by using Theorem 6 we can compute $N(2; P_1) = 3$ for $P_1 = \{p_{1,1}, p_{1,1}, p_{1,2}, p_{1,2}\}$ and $N(2; P_2) = 2$ for $P_2 = \{p_{1,1}, p_{1,2}, p_{2,1}\}$. We can see that $P_{max} = P_1$.

Finally, in order to compute $N(n; P)$ for large n and $size(P)$, Lemma 1 can be used to decompose the problem into two smaller sub-problems. The decomposition can be recursively applied until the problem size is sufficiently reduced.

8 Experimental results

Using the approach described above, we computed $N(n; P_{max})$ and $N(n; P^*_{max})$ for CRC lengths $n = 16, 32, 48$ and 64 bits and message lengths $l = 16, 32, 64, 128$ and 256 bits. The resulting upper bounds ϵ_1 and ϵ_2 on collision probabilities, computed using equations (2) and (3), are shown in Table 1 in the logarithmic form $-\log_2(\epsilon_i)$. The 7th column shows the upper bound ϵ_3 on collision probability of the cryptographically secure CRC of Krawczyk [7]. Columns 4, 6 and 8 show the fraction $\frac{-\log_2(\epsilon_i)}{n}$ reflecting the efficiency of ϵ_i with respect to the optimum collision probability $1/2^n$, for $i \in \{1, 2, 3\}$.

We can see from the table that the case of polynomials with a non-zero constant terms (column 5) has a smaller collision probability compared to the case of arbitrary polynomials (column 3). For example, for $n = 32$ and $m = 256$, $\epsilon_2 = 1/2^{9.65}$ while $\epsilon_1 = 1/2^{8.96}$, so $\epsilon_2 < \epsilon_1$. Since the former case is also preferable from the point of view of correcting random burst errors (see Theorem 3), the family $HRC_{l,n}$ seems more useful than the family $HR_{l,n}$.

We can also see that the presented method is particularly suitable for the authentication of short messages. As the message size grows, its the collision probability grows much sharper compared to the collision probability of Krawczyk's CRC [7]. Short messages (a few bytes to a few tens of bytes) are expected to be dominant in IoT applications. Since the presented method provides some level of protection almost for free, it method might be quite useful for low-cost resource-constrained IoT devices which can allocate very limited resources for implementing security.

In the current standard message formats, two separate fields are used for the protection against random and malicious errors. These fields may be located on different layers, e.g. CRC can be at the media access control layer while message

CRC	Message	Collision probability for different generator polynomials					
length	length	Arbitrary		With non-0 const.		Irreducible [7]	
n, bits	l, bits	$-\log_2(\epsilon_1)$	$\frac{-\log_2(\epsilon_1)}{n}$	$-\log_2(\epsilon_2)$	$\frac{-\log_2(\epsilon_2)}{n}$	$-\log_2(\epsilon_3)$	$\frac{-\log_2(\epsilon_3)}{n}$
16	16	7.91	0.49	8.54	0.53	10.00	0.63
16	32	6.06	0.38	6.66	0.42	9.42	0.59
16	64	4.62	0.29	4.83	0.30	8.68	0.54
16	128	3.08	0.19	3.55	0.22	7.83	0.49
16	256	2.20	0.14	2.58	0.16	6.91	0.43
32	16	22.06	0.69	22.66	0.71	25.42	0.79
32	32	18.16	0.57	18.82	0.59	25.00	0.78
32	64	14.53	0.45	15.33	0.48	24.42	0.76
32	128	11.46	0.36	12.23	0.38	23.68	0.74
32	256	8.96	0.28	9.65	0.30	22.83	0.71
48	16	36.95	0.77	37.45	0.78	41.00	0.85
48	32	31.90	0.66	32.74	0.68	40.68	0.85
48	64	26.70	0.56	27.56	0.57	40.19	0.84
48	128	21.97	0.46	22.87	0.48	39.54	0.82
48	256	17.84	0.37	18.71	0.39	38.75	0.81
64	16	52.28	0.82	52.83	0.83	56.68	0.89
64	32	46.53	0.73	47.33	0.74	56.42	0.88
64	64	39.82	0.62	40.77	0.64	56.00	0.88
64	128	33.65	0.53	34.62	0.54	55.42	0.87
64	256	27.95	0.44	28.98	0.45	54.68	0.85

Table 1. Comparison of collision probabilities for three types of CRC generator polynomials: (1) an arbitrary polynomial (column 3), (2) a polynomial with a non-zero constant term (column 5), (3) an irreducible polynomial (column 7).

authentication code can be at the application layer. A good strategy might be to combine these two fields into one at the media access control layer and use the presented method for the protection against both types of errors.

9 Conclusion

In this paper, we introduced two new families of cryptographically secure hash functions based on CRCs. Similarly to previously proposed cryptographically secure CRC-based hash families, the presented ones enable combining the detection of random and malicious errors without increasing bandwidth. They detect the same type of burst errors as cryptographically non-secure CRCs based on irreducible generator polynomials. They retain most of the encoding and decoding implementation simplicity of cryptographically non-secure CRCs except that the LFSR implementing the division modulo generator polynomial needs to have programmable feedback connections.

The main advantage of the proposed CRCs is that the irreducibility testing, which is either time or memory consuming, can be omitted. It takes only $O(n)$ time to generate a random polynomial of degree n. In contract, its takes $\Omega(n^3)$ time to generate a random irreducible polynomial of degree n.

However, using arbitrary polynomials as generator polynomials for the CRC gives an adversary a higher chance of braking authentication. We provide a detailed quantitative analysis of the achieved security as a function of message and CRC sizes and show that the presented authentication scheme might be useful for low-cost resource-constrained devices.

References

1. Ericsson, “More that 50 billions connected devices,” 2012. www.ericsson.com/res/docs/whitepapers/wp-50-billions.pdf.
2. Center for Strategic and International Studies, “Net losses: Estimating the global cost of cybercrime,” June 2014. https://www.mcafee.com/mx/resources/reports/rp-economic-impact-cybercrime2.pdf.
3. Proofpoint, “Proofpoint uncovers Internet of Things (IoT) cyberattack,” January 2014. https://www.proofpoint.com/us/ proofpoint-uncovers-internet-of-things-iot-cyberattack.
4. M. Bellare, R. Canetti, and H. Krawczyk, “Keying hash functions for message authentication,” in *Advances in Cryptology - CRYPTO 96* (N. Koblitz, ed.), vol. 1109 of *Lecture Notes in Computer Science*, pp. 1–15, Springer Berlin Heidelberg, 1996.
5. M. Bellare, J. Kilian, and P. Rogaway, “The security of cipher block chaining,” in *Advances in Cryptology CRYPTO 94* (Y. Desmedt, ed.), vol. 839 of *Lecture Notes in Computer Science*, pp. 341–358, Springer Berlin Heidelberg, 1994.
6. T.-B. Pei and C. Zukowski, “High-speed parallel CRC circuits in VLSI,” *IEEE Transactions on Communications*, vol. 40, pp. 653 –657, Apr. 1992.
7. H. Krawczyk, “LFSR-based hashing and authentication,” in *Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO ’94, (London, UK, UK), pp. 129–139, Springer-Verlag, 1994.

8. E. Dubrova, M. Naslund, G. Selander, and F. Lindqvist, "Cryptographically secure crc for lightweight message authentication," Tech. Rep. 2015/035, January 2015. Cryptology ePrint Archive.
9. E. Dubrova, M. Naslund, and G. Selander, "CRC-based message authentication for 5G mobile technology," in *Proceedings of 1st IEEE International Workshop on 5G Security*, August 2015.
10. S. Gao and D. Panario, "Tests and constructions of irreducible polynomials over finite fields," in *Foundations of Computational Mathematics* (F. Cucker and M. Shub, eds.), pp. 346–361, Springer Berlin Heidelberg, 1997.
11. IEEE Std 802.15.4-2011, "IEEE standard for local and metropolitan area networks - part 15.4: Low-rate wireless personal area networks (LR-WPANs)," 2011. standards.ieee.org/getieee802/download/802.15.4-2011.pdf.
12. 3GPP TS 36.212, "3gpp technical specifications 36.212, multiplexing and channel coding (release 8)," 2008. http://www.qtc.jp/3GPP/Specs/36212-830.pdf.
13. ETSI TS 100 909, "Digital cellular telecommunications system (Phase 2+); Channel coding," 2005. http://www.etsi.org/deliver/etsi_ts/100900_100999/100909/08.09.00_60/ts_100909v080900p.pdf.
14. W. Peterson and D. Brown, "Cyclic codes for error detection," *Proceedings of the IRE*, vol. 49, pp. 228 –235, Jan. 1961.
15. S. Golomb, *Shift Register Sequences*. Aegean Park Press, 1982.
16. E. Stavinov, "A practical parallel CRC generation method," *Feature Article*, pp. 38–45, Jan. 2010.
17. G. Simmons, "A survey of information authentication," *Proceedings of the IEEE*, vol. 76, pp. 603–620, May 1988.
18. M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '96, (London, UK), pp. 1–15, Springer-Verlag, 1996.
19. M. Bellare, R. Guérin, and P. Rogaway, "Xor macs: New methods for message authentication using finite pseudorandom functions," in *Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '95, (London, UK, UK), pp. 15–28, Springer-Verlag, 1995.
20. E. N. Gilbert, F. J. MacWilliams, and N. J. A. Sloane, "Codes which detect deception," *Bell System Technical Journal*, vol. 53, no. 3, pp. 405–424, 1974.
21. G. J. Simmons, "Authentication theory/coding theory," in *Proceedings of CRYPTO 84 on Advances in Cryptology*, (New York, NY, USA), pp. 411–431, Springer-Verlag New York, Inc., 1985.
22. M. N. Wegman and J. Carter, "New hash functions and their use in authentication and set equality," *Journal of Computer and System Sciences*, vol. 22, no. 3, pp. 265 – 279, 1981.
23. G. Brassard, "On computationally secure authentication tags requiring short secret shared keys," in *Advances in Cryptology* (D. Chaum, R. Rivest, and A. Sherman, eds.), pp. 79–86, Springer US, 1983.
24. Y. Desmedt, "Unconditionally secure authentication schemes and practical and theoretical consequences," in *Advances in Cryptology CRYPTO'85 Proceedings* (H. C. Williams, ed.), vol. 218 of *Lecture Notes in Computer Science*, pp. 42–55, Springer Berlin Heidelberg, 1986.
25. D. R. Stinson, "Universal hashing and authentication codes," *Des. Codes Cryptography*, vol. 4, pp. 369–380, Oct. 1994.

26. D. R. Stinson, "On the connections between universal hashing, combinatorial designs and error-correcting codes," in *In Proc. Congressus Numerantium 114*, pp. 7–27, 1996.
27. J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway, "Umac: Fast and secure message authentication," in *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, (London, UK, UK), pp. 216–233, Springer-Verlag, 1999.
28. X. Lai, R. Rueppel, and J. Woollven, "A fast cryptographic checksum algorithm based on stream ciphers," in *Advances in Cryptology AUSCRYPT '92* (J. Seberry and Y. Zheng, eds.), vol. 718 of *Lecture Notes in Computer Science*, pp. 339–348, Springer Berlin Heidelberg, 1993.
29. R. Taylor, "An integrity check value algorithm for stream ciphers," in *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '93, (London, UK, UK), pp. 40–48, Springer-Verlag, 1994.
30. T. Johansson, "A shift register construction of unconditionally secure authentication codes," *Designs, Codes and Cryptography*, vol. 4, no. 1, pp. 69–81, 1994.
31. M. Agren, M. Hell, T. Johansson, and W. Meier, "Grain-128a: A new version of grain-128 with optional authentication," *Int. J. Wire. Mob. Comput.*, vol. 5, pp. 48–59, Dec. 2011.
32. M. Rabin, "Fingerprinting by random polynomials," Tech. Rep. TR-15-81, Center for Research in Computing Technology, Harvard Univ., Cambridge, Mass, 1981.
33. C. C. Y. Lam, G. Gong, and S. A. Vanstone, "Message authentication codes with error correcting capabilities," in *Proceedings of the 4th International Conference on Information and Communications Security*, ICICS '02, (London, UK, UK), pp. 354–366, Springer-Verlag, 2002.
34. R. Ge, G. Arce, and G. Di Crescenzo, "Approximate message authentication codes for n-ary alphabets," *Information Forensics and Security, IEEE Transactions on*, vol. 1, pp. 56–67, March 2006.
35. O. Ur-Rehman, N. Zivic, S. Tabatabaei, and C. Ruland, "Error correcting and weighted noise tolerant message authentication codes," in *Signal Processing and Communication Systems (ICSPCS), 2011 5th International Conference on*, pp. 1–8, Dec 2011.
36. D. Stinson, *Cryptography Theorey and Practice.* Chapman & Hall/CRC, 3rd edition, 2006.
37. ETSI SAGE 3GPP.
38. G. Griffiths and G. C. Stones, "The tea-leaf reader algorithm: an efficient implementation of CRC-16 and CRC-32," *Commun. ACM*, vol. 30, pp. 617–620, July 1987.
39. M. Braun, J. Friedrich, T. Grun, and J. Lembert, "Parallel CRC computation in FPGAs," in *Field-Programmable Logic Smart Applications, New Paradigms and Compilers* (R. Hartenstein and M. Glesner, eds.), vol. 1142 of *Lecture Notes in Computer Science*, pp. 156–165, Springer Berlin / Heidelberg, 1996.
40. J. Birch, L. G. Christensen, and M. Skov, "A programmable 800 mbit/s crc check/-generator unit for lans and mans," *Comput. Netw. ISDN Syst.*, vol. 24, pp. 109–118, Apr. 1992.
41. R. Lidl and H. Niederreiter, *Introduction to Finite Fields and their Applications.* Cambridge Univ. Press, 1994.