Cryptogr. Commun. (2018) 10:383–399
DOI 10.1007/s12095-017-0227-8

Message authentication based on cryptographically secure CRC without polynomial irreducibility test

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

Received: 19 December 2016 / Accepted: 10 April 2017 / Published online: 29 April 2017

Abstract In this paper, we present a message authentication scheme based on cryptographically secure cyclic redundancy check (CRC). Similarly to previously proposed cryptographically secure CRCs, the presented one detects both random and malicious errors without increasing bandwidth. The main difference from previous approaches is that we use random 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 Message authentication · Hash function · CRC · LFSR

Mathematics Subject Classification (2010) 94A60 · 94A62

This article is part of the Topical Collection on Sequences and Their Applications.

Elena Dubrova is supported by the research grant No SM14-0016 from the Swedish Foundation for Strategic Research.

✉ Elena Dubrova
dubrova@kth.se

Mats Näslund
mats.naslund@ericsson.com

Göran Selander
goran.selander@ericsson.com

Fredrik Lindqvist
fredrik.lindqvist@ericsson.com

[1] Department of Electronics, Royal Institute of Technology, Stockholm, Sweden

[2] Ericsson Research, Ericsson AB, Stockholm, Sweden

1 Introduction

Today minimal or no security is typically provided to low-end low-cost wireless devices such as sensors or RFID tags in the conventional belief that the information they gather is of little concern to attackers [45]. However, case studies have shown that a compromised sensor can be used as a stepping stone to mount an attack on a wireless network [37]. For example, in the attack described in [37], wireless tire pressure sensors were hacked and used to access the automotive system.

Future wireless networks are expected to support security-critical services related to industrial automation, traffic safety, smart transport, smart grid, e-health, etc. The value of the information to which the low-end devices will have access via future wireless networks is expected to be much greater than the one today, hence the incentives for attackers will increase [17]. As processing power and connectivity become cheaper, the cost of performing an attack drops. 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 national security.

Many low-cost wireless devices work under severe resource constrains such as limited battery and computing power, little memory, and insufficient bandwidth. These devices must dedicate most of their available resources to executing core application functionality and have little resources left for implementing security. To satisfy their constrains, it might be necessary to reuse existing functions, e.g. by combining coding techniques (scrambling, checksums, forward error correction (FEC)) with cryptographic techniques (encryption, integrity protection). In particular, functional similarities between error detection and data integrity protection can be exploited to combine these functions in one.

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) [7], *on the top* of an error-detecting code, e.g. n-bit cyclic redundancy check (CRC). However, such an 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, we cannot guarantee the detection of the same type of random errors as the CRC. For example, the detection of n-bit burst errors cannot be guaranteed. This may have a negative impact on the reliability of communication links. Only if we make the conventional CRC *cryptographically secure*, can we assure a certain level of security without sacrificing reliability.

The latter motivated the development of cryptographically secure CRCs. The core idea is to make the CRC generator polynomial variable and secret. The CRC presented by Krawczyk [27] is based on irreducible generator polynomials. The approach described in [15] uses a product of irreducible polynomials. The CRC proposed in [14] uses generator polynomials of type $(1 + x)\,p(x)$, where $p(x)$ is a primitive polynomial. In all three cases, testing for irreducibility or primitivity is required, which is either time or memory consuming. Selecting an irreducible degree-n polynomial at random requires either selecting at random a degree-n polynomial ($O(n)$ time) and running a test for irreducibility ($\Omega(n^3)$ time[1] [21]), or selecting at random a degree-n polynomial from a database of irreducible

[1]There are asymptotically better algorithms for testing irreducibility, approaching $O(n^2)$ [12], but their usefulness for small n, relevant in the context of this paper, is unclear.

degree-n polynomials (roughly $2^n/n$ space). Note that the irreducibility test has to be done during key agreement, i.e. it incurs delay before the communication can start. Therefore, it is desirable to minimize the time spent on doing it as much as possible.

In this paper, we present a cryptographically secure CRC based on any randomly selected generator polynomial, with no requirements on irreducibility. We provide a detailed quantitative analysis of the achieved security as a function of message and CRC sizes. To the best of our knowledge, no security analysis for the general case of reducible polynomials has been made so far. This might be due to 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 gives a background on hash functions and describes the basics of CRC codes. In Section 3, we introduce two new families of cryptographically secure CRC hash functions. Section 4 analyzes error-detecting capabilities of hash families. In Section 5, we present the security analysis of hash families. Section 6 shows experimental results. Section 7 describes related work. Section 8 concludes the paper and discusses open problems.

2 Preliminaries

2.1 Notation

Throughout the paper, we associate each binary string $L \in \{0,1\}^l$ representing an l-bit binary 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 $\deg(L)$ to denote the degree of the polynomial $L(x)$.

We use $a \in_R A$ to indicate that the element a is taken uniformly at random from the set A. The term $\Pr_h$ is used to denote the probability of the event $E(h)$ where $h \in_R H$ and H is a set of hash functions, usually implicit.

2.2 Hash functions

In this section we describe properties of hash functions which are used in the sequel.

Definition 1 [27] An (l, n)-family of hash functions H is a set of functions h that map the set of binary strings of length l into the set of binary strings of length n.

The hash functions considered in this paper are linear relative to the exclusive-OR operation. This property simplifies their analysis.

Definition 2 [27] A family of hash functions H is $\oplus$-linear if, for all messages L_1 and L_2 and for all $h \in H$,

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

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

Another important property of some hash functions is the ability to map elements into their images in a balanced way.

Definition 3 [27] A family of hash functions H is ϵ-balanced if

$$\forall L \neq 0, \forall a \in \{0,1\}^n : \Pr_h[h(L) = a] \leq \epsilon.$$

2.3 Message authentication

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 is known that hash functions can be combined with one-time pads to construct strong authentication algorithms [46]. In this case, the secret key consists of the description of a particular hash function $h \in_R H$ drawn randomly from an (l, n)-family of hash functions H and a random pad $s \in_R \{0,1\}^n$.

In the definition below, it is assumed that the adversary knows the family of hash functions H, but not the particular value of h or the pad s. As mentioned in [27] the name "otp"-secure is intended to stress that importance of the one-time pad for the security of the authentication scheme.

Definition 4 [27] A family of hash functions H is ϵ-otp-secure if, for any message L, no adversary that has L and its hash tag $t = h(L) \oplus s$, where $h \in_R H$ and $s \in_R \{0,1\}^n$, can find $L' \neq L$ and $t' = h(L') \oplus s$ with probability larger than ϵ.

Note that the success probability of an adversary that can modify a single transmitted message remains ϵ even if the adversary has access to more than one pair of messages and tags, provided that truly random pads s are used for computing the tags and these pads are changed for every message [27]. If this holds, then the authentication tags look completely random and therefore leak no information on the value of h. If the adversary is able to modify k of the transmitted messages, then the success probability is at most $k\epsilon$ [27].

In most practical applications, pseudo-random pads generated from a secret seed shared by the communicating parties rather than truly random pads of the size of the hash output are used. In this case, the unconditional security of the authentication scheme in Definition 4 reduces to the security of the pseudo-random generator producing the pads and the computational power of the adversary is assumed to be bounded depending on the security model of the pseudo-random generator [27].

The following theorem characterizes ϵ-otp-secure families of hash functions.

Theorem 5 [27] *A necessary and sufficient condition for a family of hash functions H to be ϵ-otp-secure is that*

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

For linear families of hash functions, Theorem 5 implies the following result.

Theorem 6 [27] *If H is $\oplus$-linear, then H is ϵ-otp-secure if and only if H is ϵ-balanced.*

We use Theorem 6 as the main step in proving the security of the presented authentication scheme.

2.4 Cyclic redundancy check

A cyclic redundancy check (CRC) is widely used for protecting data communication or storage against random errors [34]. Many wireless communication standards use CRC.

For example, IEEE 802.15.4 standard uses 16-bit CRC [24], LTE uses 24, 16 and 8-bit CRCs [1], and GSM uses 40-bit CRC [19].

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

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

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 [36]. 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 [34].

The CRC encoding and decoding can be efficiently implemented using a linear feedback shift register (LFSR) [23] having $g(x)$ as its connection polynomial. There are many efficient techniques for speeding up the computation of CRC [32, 33, 38].

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 $r(x) = L'(x) \cdot x^n \bmod g(x)$, 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.

3 Two families of cryptographically secure CRC hash functions

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

Definition 7 (Family H_R) For any binary message L of length l and any polynomial $g(x)$ of degree n over $GF(2)$, a hash function $h_g(L)$ is defined as the coefficients of the polynomial

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

The (l, n)-family H_R is a set of all hash functions h_g, $H_R = \{h_g : \{0,1\}^l \to \{0,1\}^n\}$.

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

To authenticate a message L using the hash function family H_R, a sender computes the authentication tag t as

$$t = h_g(L) \oplus s, \tag{1}$$

where $h_g \in_R H_R$ and $s \in_R \{0,1\}^n$, appends t to L, and transmits the message and the appended tag. A receiver authenticates a received message L' (potentially different from L) by re-computing the tag for L' and comparing the received and the re-computed tags. A disagreement implies an error.

Note that the modification of the linear hash function to the affine one is necessary to prevent an attacker from injecting all-0 messages. Without such a modification, the hash value of an all-0 message would always be 0, independently of the polynomial $g(x)$. The reader familiar with e.g. the UIA2 MAC of the 3G standard will recognize this type of construction. In that case, the encryption pad s is generated by the SNOW3G stream cipher [18].

We also consider separately a special case of the Definition 7 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 8 (Family H_{RC}) For any binary message L of length l and for any 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 coefficients of the polynomial

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

The (l, n)-family H_{RC} is a set of all hash functions h_q, $H_{RC} = \{h_q : \{0,1\}^l \rightarrow \{0,1\}^n\}$.

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

Similarly to the family H_R, the authentication tag for the family H_{RC} is computed as

$$t = h_q(L) \oplus s, \tag{2}$$

where $h_q \in_R H_{RC}$ and $s \in_R \{0,1\}^n$.

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 has to be changed to appear random to an adversary. Therefore, an LFSR implementing encoding and decoding for the cryptographic CRC needs re-programmable connections. Techniques for implementing re-programmable LFSRs are known [9]. Re-programmable LFSRs are used, for example, in applications which support multiple CRC standards.

Note that restricting generator polynomials to polynomials with non-zero constant terms does not complicate the implementation of CRC encoding and decoding 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 is made fixed rather than programmable.

4 Analysis of error-detecting capabilities

It is well-known that a CRC based on an irreducible generator polynomial of degree n detects all burst errors on length n or less [34]. Next, we show that a cryptographically secure CRC based on a reducible generator polynomial of degree n with a non-zero constant term detects the same type of errors.

Theorem 9 *A CRC based on a reducible 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 8 using a reducible degree-n generator polynomial $q(x)$ with a non-zero constant term. Any k-bit burst error e, $0 < k \leq n$, can be described by a polynomial of type

$$e(x) = x^j \cdot f(x) \tag{3}$$

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 divisible by the generator polynomial $q(x)$. Clearly $gcd(x^j, q(x)) = 1$. So, $e(x)$ is divisible by $q(x)$ if and only if $f(x)$ is divisible by $q(x)$. However, this is not possible since $\deg(f) < n$. Therefore, a CRC based on a reducible degree-n generator polynomial with a non-zero constant term detects all burst errors on length n or less. □

Theorem 9 shows that, from the point of view of correcting burst errors, no advantage is lost if an irreducible polynomial is replaced by a reducible polynomial with a non-zero constant term.

5 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 [43]. The sender and the receiver share a secret key which is unknown to the adversary. In our case, the key consists of the description of a particular generator polynomial $g(x)$ (respectively, $q(x)$) drawn randomly from the set of all possible degree-n polynomials (all possible degree-n polynomials with a non-zero constant term) over $GF(2)$ and a random pad $s \in_R \{0,1\}^n$.

In order to prove the security of the hash function families H_R and H_{RC} for implementation for the message authentication schemes (1) and (2), respectively, in this section we show that these families are ϵ-otp-secure with ϵ being exponentially small in the length of the hash value. By Definition 4, if a hash family is ϵ-otp-secure, then the success probability for an adversary to modify a message is at most ϵ. Throughout the paper, when we say "attack success probability", we mean the probability that an adversary can successfully modify a message according to the scenario described by Definition 4.

In the following section we quantify ϵ for the hash function families H_R and H_{RC}.

5.1 Quantifying attack success probability

An adversary can successfully replace a message and a tag pair (L, t) by another pair (L', h'), $L' \neq L$, only if for the hash function $h \in_R H$ and pad $s \in_R \{0,1\}^n$ used by the communicating parties it holds that $t = h(L) \oplus s$ and $t' = h(L') \oplus s$, or equivalently $t \oplus t' = h(L) \oplus h(L')$ [27]. Thus, the success probability of the adversary is bounded by

$$\max_{L,L',a} \Pr_h[h(L) \oplus h(L') = a]$$

where $a = t \oplus t'$. By Theorem 6, for linear hash functions the above condition can be simplified to

$$\max_{L,a} \Pr_h[h(L) = a]$$

for all $L \neq 0$. Let us analyze how this probability can be maximized for the hash function families H_R and H_{RC}.

By Definition 7, for the hash function family H_R, the success probability is proportional to the number of degree-n polynomials, $g(x)$, that divide the polynomial $L(x) \cdot x^n - a(x)$. So, in order to find ϵ for H_R, we need to estimate the maximum number of distinct degree-n polynomials that can be constructed from the irreducible factors of a degree-$(n + l)$ polynomial.

Similarly, by Definition 8, for the hash function family H_{RC}, the success probability is proportional to the number of degree-n polynomials with a non-zero constant term, $q(x)$, that divide the polynomial $L(x) \cdot x^n - a(x)$. So, in this case we need to find the maximum number of distinct degree-n polynomials with a non-zero constant term that can be constructed from the irreducible factors of a degree-$(n+l)$ polynomial. Note that all irreducible factors of a polynomial with a non-zero constant term have a non-zero constant term.

We start with the case of the hash function family H_R. 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 $P = \{x, x, x+1, x^2+x+1\}$, $mult(x) = 2$ and $size(P) = 5$.

Let $N(n; P)$ denote the number of distinct degree-n polynomials over $GF(2)$ which can be constructed by multiplying a subset of the elements of P. For example, if $P = \{x, x, x+1, x^2+x+1\}$ and $n = 2$, we can construct x^2, $x(x+1)$ and x^2+x+1, so $N(2; P) = 3$. $N(0; P)$ 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 address this problem in Section 5.2.

For a given n, we want to find a multiset of irreducible polynomials P_{max} such that

$$N(n; P_{max}) = \max_{\forall P: size(P) = size(P_{max})} N(n; P).$$

If P_{max} is known, we can quantify the attack success probability for the hash function family H_R as follows.

Theorem 10 *For any values of l and n, the (l, n)-family of hash functions H_R is ϵ_1-otp-secure for*

$$\epsilon_1 \leq \frac{N(n; P_{max})}{2^n}, \tag{4}$$

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

Proof A family of hash functions is ϵ-otp-secure if it is $\oplus$-linear and ϵ-balanced. The family of hash functions H_R is $\oplus$-linear because for all messages L_1 and L_2 and for all $h_g \in H_R$, we have $h_g(L_1 \oplus L_2) = h_g(L_1) \oplus h(L_2)$.

To show that the family H_R is also ϵ-balanced, we observe that, on one hand, for any degree-n polynomial $g(x)$ 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 $L(x) \cdot x^n \bmod g(x) = a(x)$. On the other hand, $L(x) \cdot x^n \bmod g(x) = a(x)$ 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 H_R that map L into a, because $N(n; P_{max})$ is the maximum number of distinct degree-n polynomials which can be constructed from the irreducible factors of any degree-$(n + l)$ polynomial. On the other hand, the family H_R consists of 2^n elements (the number of degree-n polynomials over $GF(2)$). Therefore

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

□

In a similar way we can quantify the attack success probability for the hash function family H_{RC}.

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 11 *For any values of l and n, the (l, n)-family of hash functions H_{RC} is ϵ_2-otp-secure for*

$$\epsilon_2 \leq \frac{N(n;\, P^*_{max})}{2^{n-1}}, \tag{5}$$

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

Proof Similar to the proof of Theorem 10.

In the following subsections we show how to compute $N(n;\, P)$ and $N(n;\, P^*)$. □

5.2 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 [30].

Let $p_{i,j}$ be the 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 is not significant. 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 degree-n polynomials which can be constructed by multiplying a subset of 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 which 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 12 distinct polynomials can be constructed from P:

$$x^5, x^4(x+1), x^3(x+1)^2, x^2(x+1)^3, x(x+1)^4, (x+1)^5$$
$$x^3(x^2+x+1), x^2(x+1)(x^2+x+1), x(x+1)^2(x^2+x+1), (x+1)^3(x^2+x+1)$$
$$x(x^2+x+1)^2, (x+1)(x^2+x+1)^2.$$

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

Next, we show that $N(n;\, P)$ can be computed using a recurrence relation given by the following Lemma. 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 12 *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 degree-n polynomials which can be constructed from the elements of $P \cup \{p\}$ and the number of distinct degree-n polynomials which can be constructed from the elements of P. This difference is equal to the number of distinct degree-n polynomials which contain p as a factor with the multiplicity exactly one. Removing factor p from each of such polynomials yields 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 $2 \leq m + 1 \leq \lfloor \frac{n}{\deg(p)} \rfloor$.

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 degree-n polynomials which can be constructed from the elements of $P \cup \{p^{m+1}\}$ and the number of distinct degree-n polynomials 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 degree-$n$polynomials which contain p as a factor with the multiplicity exactly $mult(p) + 1$. Removing the r 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 $N(n - (mult(p) + 1) \cdot \deg(p); P)$. □

Finally, 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 multiplicities 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

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

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

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

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

There is only one irreducible polynomial of degree 2. From Lemma 12, we can conclude that

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

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 13 *For* $d = 1$

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

and for $d > 1$

$$N(n;\, P_d) = \sum_{i_{d,1}=0}^{A_{d,1}} \sum_{i_{d,2}=0}^{A_{d,2}} \ldots \sum_{i_{d,I_d}=0}^{A_{d,I_d}} \ldots \sum_{i_{2,1}=0}^{A_{2,1}} N\left(n - \sum_{h=2}^{d} \sum_{j=1}^{I_h} i_{h,j};\, P_1\right) \tag{6}$$

where

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

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 13, $N(n;\, P_1^*)$ reduces to

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

Lastly, we would like to point out the relation between the problem we addressed in this section and restricted colored integer partitions.[2] The number $N(n;\, P)$ is equal to the number of colored partitions of the integer n into arbitrarily many parts such that the integer i may occur in $f(i)$ different colors ($f(i)$ corresponds to the number of polynomials of degree i in P) and the number of occurrences of the integer i with a color $c \in f(i)$ in the partition is at most $m(i, c)$ ($m(i, c)$ corresponds to the multiplicity of the polynomial with the degree i and color c in P). A lot of work has been done on k-colored partitions, in which parts may appear in k different colors, see for example [8], or a survey [3]. The

[2] An *integer partition* of a nonnegative integer n with k summands, or parts, is a way of writing n as a sum of k nonnegative integers, where the order of parts is not significant.

generalization of k-colored partitions in which at most j colors may appear for a given part size has been recently presented in [26]. However, we are not aware of any work addressing the specific case of this paper in which the integer i may occur in $f(i)$ different colors and the number of occurrences of the integer i with a color $c \in f(i)$ in the partition is at most $m(i, c)$. We only know the work of Eger [16] on S-restricted f-colored integer compositions (where the order of parts is significant) in which all parts lie within a subset S of nonnegative integers and each integer $i \in S$ may take on $f(i)$ different colors.

5.3 Computing $N(n; P_{max})$

Theorem 13 shows us how to compute $N(n; P)$ for a given n and P. Next we need to find a vector of multiplicities which maximizes $N(n; P)$ for a given n and $size(P)$. In this section, we derive some properties which allow us to guide and bound the search.

Property 14 *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 $p_j \notin P_{max}$ for 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 14, we can derive a rough upper bound on the maximum degree of polynomials contained in P_{max} by computing the smallest integer d satisfying

$$size(P_{max}) \leq I_1 + 2I_2 + 3I_3 + \ldots + dI_d. \tag{7}$$

We can reduce the search space for P_{max} by first deriving an upper bound on d using (7) and then removing from the consideration multisets P which do not satisfy the condition of Property 14. We also can take into account that the order of elements of the same degree in a multiset does not matter.

Property 15 *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 14 and 15, we can reduce the set of candidates into P_{max} to two:

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

Now by using Theorem 13 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 12 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.

6 Experimental results

Using the approach described above, we computed $N(n; P_{max})$ for CRC lengths $n = 32, 48, 64, 96$ and 128 bits and message lengths $l = 32, 64, 128$ and 256 bits. The resulting upper bounds ϵ_1 and ϵ_2 on attack success probabilities, computed using (4) and (5), are shown in Table 1 in the logarithmic form $-\log_2(\epsilon_i)$. The 7th column shows the upper bound ϵ_3 on attack success probability of the cryptographically secure CRC of Krawczyk [27], given by $\epsilon_3 \leq (n + l)/2^{n-1}$. 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 probability $1/2^n$, for $i \in \{1, 2, 3\}$.

As we can see, the case of random polynomials with a non-zero constant terms (column 5) has a smaller attack success probability compared to the case of random polynomials

Table 1 Comparison of attack success probabilities for three types of generator polynomials

CRC	Message	Attack success probability for different generator polynomials					
Length	Length	Random		Random with non-0 const.		Irreducible [27]	
n, bits	l, bits	$-\log_2(\epsilon_1)$	$\frac{-\log_2(\epsilon_1)}{n}$	$-\log_2(\epsilon_2)$	$\frac{-\log_2(\epsilon_2)}{n}$	$-\log_2(\epsilon_3)$	$\frac{-\log_2(\epsilon_3)}{n}$
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	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	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
96	32	76.52	0.80	77.29	0.81	88.00	0.92
96	64	68.16	0.71	69.21	0.72	87.68	0.91
96	128	59.03	0.61	60.08	0.63	87.19	0.91
96	256	50.23	0.52	49.96	0.52	86.54	0.90
128	32	107.46	0.84	108.23	0.85	119.68	0.93
128	64	97.65	0.76	98.62	0.77	119.42	0.93
128	128	86.03	0.67	87.06	0.68	119.00	0.93
128	256	74.69	0.58	75.90	0.59	118.42	0.93

(column 3). The former case is also preferable from the point of view of correcting burst errors. We can also see from the table that the presented method is particularly suitable for the authentication of short messages.

7 Related work

A lot of work has been done on message authentication codes in the past, see [39] for an excellent survey. Security of several types of MACs, including HMAC [5], CBC-MAC [7] and XOR-MAC [6], have been quantitatively analyzed.

Unconditionally secure message authentication codes were pioneered by Gilbert et al. [22] and their theoretical basis was developed by Simmons [40].

Wegman and Lawrence Carter [46] showed that hash functions can be combined with one-time pads to construct strong authentication algorithms. Their approach was further developed by Brassard [11], Desmedt [13] and Krawczyk [27].

Stinson [41] introduced the notion of "almost strongly universal hash families" which made possible to considerably reduce the key size of unconditionally secure MACs. For more details on universal hashing, see [42]. Black et al. showed that universal hash families can be used to construct efficient computationally secure MACs, e.g. UMAC [10].

Various techniques for cryptographic checksums and MACs based on stream ciphers have been proposed, including Lai et al. [28], Taylor [44], Johansson [25] and [2]. 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 [27], 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 [35] 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 [27]. 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 [27] proved that the inclusion of the n-bit shift into Rabin's scheme [35] makes the scheme secure for message authentication provided that tag is encrypted using a one-time pad. He showed that the probability of breaking the resulting authentication scheme is $\epsilon \leq \frac{l+n}{2^{n-1}}$, where n is CRC length and l is message length.

In [15] Krawczyk's approach was extended to the case when a product of k irreducible polynomials is used to generate the CRC. The attack success probability of such an authentication scheme is $\epsilon \leq \frac{(l+n)^k}{2^{n-k}}$.

In [14] 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 attack success probability in this case is $\epsilon \leq \frac{l+n-1}{2^{n-2}}$.

Krawczyk also developed another interesting family of hash functions based on Toeplitz hashing in which the columns of a matrix are formed by the consecutive states on an LFSR [27]. Such a method has a lower hashing and authentication strength compared to

the approach based on a random matrix, namely $\epsilon \leq \frac{l}{2^{n-1}}$, where n is CRC length and l is message length, however its implementation cost is much smaller.

Apart from CRC, other error detecting/correcting codes were also proposed for message authentication. MACs based on BCH and Reed-Solomon error-correcting codes were presented in [29].

8 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 re-programmable feedback connections. The main advantage of the presented CRCs over the previously proposed ones is that the irreducibility testing, which is either time or memory consuming, can be omitted.

However, using random polynomials as generator polynomials for the CRC gives an adversary a higher chance of braking authentication. We provided a detailed quantitative analysis of the achieved security as a function of message and CRC lengths and showed that the presented authentication scheme is particularly suitable for short messages. Short messages (a few bytes to a few tens of bytes) are expected to be dominant in machine-to-machine (M2M) communication. Since the presented method provides some level of integrity protection almost for free, it might be quite useful for resource-constrained M2M devices.

Note that in our attack scenario it is assumed that an adversary gets access to a message and its authentication tag. Other attack scenarios are also possible, for example, an adversary may have an access to a verification oracle as well. In this case any cryptographic CRC, including the presented one, is susceptible to Ferguson's attack [20, 31] which reveals the polynomial used for generating the CRC with the probability 2^{-n}, where n is the polynomial degree. The access to an oracle is a reasonable assumption, for example, in a multicast.[3] Therefore, we do not recommend the use of cryptographic CRCs with short generator polynomials.

In the current wireless standard message formats two separate fields are typically used for the protection against random and malicious errors. These fields may be located on different layers, e.g. in LTE the CRC is located at the physical (PHY) layer while the message authentication code is located at the packet data convergence protocol (PDCP) layer. A good strategy might be to combine these two fields into the one at the PHY layer and use the a cryptographic CRC for the protection against both types of errors. Future work involves investigating implications for security and coverage caused by such a merge.

[3]Multicast is a type of communication where the information is addressed to many destinations simultaneously.

Acknowledgements The authors are grateful to John Mattsson for brining to our attention the fact that cryptographic CRCs are susceptible to Ferguson's attack, to Kai-Uwe Schmidt for pointing out the connection of $N(n; P)$ to colored integer partitions, to Steffen Eger for helping us out to clarify the relation of our results to the previous work on restricted colored integer partitions, and to Daniel Katz for interesting discussions regarding expressing $N(n; P)$ in terms of generating functions and strategies for finding P_{max}.

The authors would also like to thank the anonymous reviewers for their constructive comments that greatly helped to improve the final version of the paper.

References

1. 3GPP TS 36.212: 3GPP technical specifications 36.212, multiplexing and channel coding (release 8). http://www.qtc.jp/3GPP/Specs/36212-830.pdf (2008)
2. Agren, M., Hell, M., Johansson, T., Meier, W.: Grain-128a: a new version of Grain-128 with optional authentication. Int. J. Wire. Mob. Comput. **5**(1), 48–59 (2011)
3. Andrews, G.E.: A Survey of Multipartitions Congruences and Identities, pp. 1–19. Springer, New York (2008)
4. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Koblitz, N. (ed.) Advances in Cryptology - CRYPTO '96, vol. 1109 of Lecture Notes in Computer Science, pp. 1–15. Springer, Berlin Heidelberg (1996)
5. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '96, pp. 1–15. Springer, London (1996)
6. Bellare, M., Guérin, R., Phillip, R.: XOR MACs: New methods for message authentication using finite pseudorandom functions, pp. 15–28. Springer, London (1995)
7. Bellare, M., Kilian, J., Rogaway, P.: The security of cipher block chaining. In: Desmedt, Y. (ed.) Advances in Cryptology — CRYPTO '94, Volume 839 of Lecture Notes in Computer Science, pp. 341–358. Springer, Berlin Heidelberg (1994)
8. Berndt, B.C.: Partition-theoretic interpretations of certain modular equations of Schröter, Russell, and Ramanujan. Ann. Comb. **11**(2), 115–125 (2007)
9. Birch, J., Christensen, L.G., Skov, M.: A programmable 800 Mbit/s CRC check/generator unit for LANs and MANs. Comput. Netw. ISDN Syst. **24**(2), 109–118 (1992)
10. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Phillip, R.: UMAC: Fast and secure message authentication. In: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '99, pp. 216–233. Springer, London (1999)
11. Brassard, G.: On computationally secure authentication tags requiring short secret shared keys. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology, pp. 79–86. Springer, US (1983)
12. Brent, R.P., Zimmermann, P.: Three ways to test irreducibility. http://maths-people.anu.edu.au/~brent/pd/MASCOS02t4.pdf (2008)
13. Desmedt, Y.: Unconditionally secure authentication schemes and practical and theoretical consequences. In: Williams, H.C. (ed.) Advances in Cryptology — CRYPTO'85 Proceedings, vol. 218 of Lecture Notes in Computer Science, pp. 42–55. Springer, Berlin Heidelberg (1986)
14. Dubrova, E., Naslund, M., Selander, G.: CRC-based message authentication for 5G mobile technology. In: Proceedings of 1st IEEE International Workshop on 5G Security (2015)
15. Dubrova, E., Naslund, M., Selander, G., Lindqvist, F.: Cryptographically secure CRC for lightweight message authentication. Technical Report 2015/035, Cryptology ePrint Archive (2015)
16. Eger, S.: Restricted weighted integer compositions and extended binomial coefficients. J. Integer Seq. **18**(1), 13.1.3 (2013)
17. Ericsson: 5G security. www.ericsson.com/res/docs/whitepapers/5G-security.pdf (2015)
18. ETSI SAGE 3GPP: Specification of the 3GPP confidentiality and integrity algorithms UEA2 & UIA2, document 2: SNOW 3G Specification (2006)
19. ETSI TS 100 909: Digital cellular telecommunications system (Phase 2+); Channel coding. http://www.etsi.org/deliver/etsi_ts/100900_100999/100909/08.09.00_60/ts_100909v080900p.pdf (2005)

20. Ferguson, N.: Authentication weaknesses in GCM. csrc.nist.gov/groups/ST/toolkit/BCM/documents/comments/CWC-GCM/Ferguson2.pdf (2005)
21. Gao, S., Panario, D.: Tests and constructions of irreducible polynomials over finite fields. In: Cucker, F., Shub, M. (eds.) Foundations of computational mathematics, pp. 346–361. Springer, Berlin Heidelberg (1997)
22. Gilbert, E.N., MacWilliams, F.J., Sloane, N.J.A.: Codes which detect deception. Bell Syst. Tech. J. **53**(3), 405–424 (1974)
23. Golomb, S.W.: Shift Register Sequences. Aegean Park Press (1982)
24. 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). standards.ieee.org/getieee802/download/802.15.4-2011.pdf (2011)
25. Johansson, T.: A shift register construction of unconditionally secure authentication codes. Des. Codes Crypt. **4**(1), 69–81 (1994)
26. Keith, W.J.: Restricted k-color partitions. Ramanujan J. **40**(1), 71–92 (2016)
27. Krawczyk, H.: LFSR-based hashing and authentication. In: Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '94, pp. 129–139. Springer, London (1994)
28. Lai, X., Rueppel, R.A., Woollven, J.: A fast cryptographic checksum algorithm based on stream ciphers. In: Seberry, J., Zheng, Y. (eds.) Advances in Cryptology — AUSCRYPT '92, Volume 718 of Lecture Notes in Computer Science, pp. 339–348. Springer, Berlin Heidelberg (1993)
29. Lam, C.C.Y., Gong, G., Vanstone, S.A.: Message authentication codes with error correcting capabilities. In: Proceedings of the 4th International Conference on Information and Communications Security, ICICS '02, pp. 354–366. Springer, London (2002)
30. Lidl, R., Niederreiter, H.: Introduction to Finite Fields and their Applications. Cambridge Univ Press (1994)
31. Mattsson, J., Westerlund, M.: Authentication key recovery on Galois/counter mode (GCM). Progress Cryptol. - AFRICACRYPT **2016**, 127–143 (2016)
32. McCluskey, J.: High speed calculation of cyclic redundancy codes. In: Proceedings of the 1999 ACM/SIGDA Seventh International Symposium on Field Programmable Gate Arrays, FPGA '99, pp. 250–256. ACM, New York (1999)
33. Pei, T.-B., Zukowski, C.: High-speed parallel CRC circuits in VLSI. IEEE Trans. Commun. **40**(4), 653–657 (1992)
34. Peterson, W.W., Brown, D.T.: Cyclic codes for error detection. Proc. IRE **49**(1), 228–235 (1961)
35. Rabin, M.: Fingerprinting by Random Polynomials. Technical Report TR-15-81, Center for Research in Computing Technology. Harvard University, Cambridge (1981)
36. Ramabadran, T.V., Gaitonde, S.S.: A tutorial on CRC computations. Micro IEEE **8**(4), 62–75 (1988)
37. Rouf, I., Miller, R., Mustafa, H., Taylor, T., Oh, S., Xu, W., Gruteser, M., Trappe, W., Seskar, I.: Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study. In: Proceedings of the 19th USENIX Conference on Security, pp. 21–21. Berkeley (2010)
38. Sarwate, D.V.: Computation of cyclic redundancy checks via table look-up. Commun. ACM **31**, 1008–1013 (1988)
39. Simmons, G.J.: A survey of information authentication. Proc. IEEE **76**(5), 603–620 (1988)
40. Simmons, G.J.: Authentication theory/coding theory. In: Proceedings of CRYPTO 84 on Advances in Cryptology, pp. 411–431. Springer, New York (1985)
41. Stinson, D.R.: Universal hashing and authentication codes. Univ. Codes Cryptogr. **4**(4), 369–380 (1994)
42. Stinson, D.R.: On the connections between universal hashing, combinatorial designs and error-correcting codes. Proc. Congressus Numerantium **114**, 7–27 (1996)
43. Stinson, D.: Cryptography Theory and Practice, 3rd edn. Chapman & hall/CRC (2006)
44. Taylor, R.: An integrity check value algorithm for stream ciphers. In: Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '93, pp. 40–48. Springer, London (1994)
45. Trappe, W., Howard, R., Moore, R.S.: Low-energy security: Limits and opportunities in the internet of things. IEEE Secur. Priv. **13**(1), 14–21 (2015)
46. Wegman, M.N., Lawrence Carter J.: New hash functions and their use in authentication and set equality. J. Comput. Syst. Sci. **22**(3), 265–279 (1981)