Cryptographically Secure CRC for Lightweight Message Authentication

Elena Dubrova*, Mats Näslund†, Göran Selander†, Fredrik Lindqvist†

*Royal Institute of Technology, 164 40 Stockholm, Sweden

dubrova@kth.se

†Ericsson Research, 164 80 Stockholm, Sweden

{mats.naslund,goran.selander,fredrik.lindqvist}@ericsson.com

Abstract

A simple and practical hashing scheme based on Cyclic Redundancy Check (CRC) is presented. Similarly to previously proposed cryptographically secure CRCs, the presented one detects both, random and malicious, errors without increasing bandwidth. However, we use a product of irreducible polynomials instead of a single irreducible polynomial for generating the CRC. This is an advantage since smaller irreducible polynomials are easier to compute. The price we pay is that the probability that two different messages map into the same CRC increases. We provide a detailed quantitative analysis of the achieved security as a function of message and CRC sizes. The presented method seems to be particularly attractive for the authentication of short messages.

Keywords

Hash function, message authentication, CRC, error-detection, LFSR, irreducible polynomial

I. Introduction

Cyclic Redundancy Checks (CRCs) are widely used for detecting random errors in data communication and storage [1]. Examples of common CRCs are:

- CRC-16-CDMA2000 is used in 3G mobile networks,
- CRC-CCITT is used in Bluetooth,
- CRC-32 is used in Ethernet, HDLC protocols,
- CRC-40-GSM is used in GSM control channel,

where CRC-n denotes a CRC with the the generator polynomial of degree n.

To perform the CRC encoding [2], the message polynomial, $M(x)$, is first multiplied by x^n, where n is the degree of the generator polynomial, $p(x)$. Then the result is divided modulo the generator polynomial $p(x)$. The coefficients of the result

$$r(x) = M(x) \cdot x^n \ mod \ p(x)$$

constitute the check bits of the CRC. These check bits are added to the message to form the CRC codeword:

$$M(x) \cdot x^n \oplus \; r(x)$$

where "$\oplus$" is the XOR operation.

The CRC decoding is typically done by dividing the received message modulo the generator polynomial $p(x)$ and comparing the coefficients of the resulting remainder with the received CRC check bits. A disagreement indicates an error. A CRC with the generator polynomial of degree n detects all burst[1] errors of length n or less [3]. The CRC encoding and decoding can be efficiently implemented using a Linear Feedback Shift Register (LFSR) [4] with $p(x)$ as a connection polynomial.

Traditional CRC techniques are suitable for detecting random errors. However, these techniques can easily be defeated by a malicious adversary. Since an adversary knows which generator polynomial is used by a CRC, he/she may easily craft an error so that the reminder after the division of the erroneous message by the generator polynomial is the same as the CRC check bits. For example, an adversary may add to the original message $M(x)$ an error $e(x)$ which is a multiple of the generator polynomial $p(x)$. Since $e(x) \; mod \; p(x) = 0$, such an error is not be detected by the CRC. Or, in another example, an adversary may replace the original message $M(x)$ by another message $M'(x)$, then encode $M'(x)$ as usual into the codeword $M'(x) \cdot x^n \oplus r(x)$, where $r(x) = M'(x) \cdot x^n \; mod \; p(x)$, and then submit it. The receiver will be unable to distinguish the codeword received from an adversary from a codeword received from a legitimate sender.

Using the CRC also as a basis for message authentication is very attractive since we save bandwidth and computational resources. Adding other authentication mechanism, e.g. HMAC [5], on the top of the traditional CRC requires message expansion and a separate encoding/decoding engine.

Using the CRC also as a basis for message authentication is very attractive. Obviously, authentication can be implemented by adding one of the existing authentication mechanisms, e.g. n-bit HMAC [5], on the top of the traditional n-bit CRC. However, such an approach requires message expansion by n bits and a separate encoding/decoding engine. On the other hand, if we simply *replace* an n-bit CRC with an n-bit HMAC, then the *guarantee* on the detection of all burst errors of length n or less is lost. Only in the case when we use a cryptographically secure n-bit CRC, we achieve security without sacrificing error-detection capabilities, with no loss of bandwidth, and with only an insignificant increase in computational resources.

Indeed, a type of cryptographically secure CRC was proposed by Krawczyk in [6]. It requires an irreducible polynomial of degree n to generate an authentication tag. The basic idea is to let the CRC polynomial be a secret which is known only to the sender and the receiver. This works well from the security point of view. However, a drawback is that it is not straightforward to find random irreducible polynomials. A given secret key of size, say 128 bits, should be efficiently turned into an irreducible polynomial of degree, say 32. Testing for irreducibility takes time and resources.

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

The key advantage of the cryptographically secure CRCs presented in this paper is that, instead of using a single irreducible polynomial of degree n, we use a product of k irreducible polynomials whose degrees sum up to n. Since the number of irreducible polynomials of degree n grows exponentially with n, it is considerably easier to compute smaller irreducible polynomials. Computation of irreducible polynomials can be done either by choosing a random polynomial and running a test for irreducibility (whose time complexity is $\Omega(n^3)$ bit operations [7]) or by keeping a database of irreducible polynomials. The most popular CRC size is $n = 32$ and the number of irreducible polynomials of degree 32 is 134.215.680 $\approx 2^{27}$. In contract, the number of irreducible polynomials of degree 16 is only 4080. Therefore, for many applications, a database of irreducible polynomials of degree 16 is acceptable while a database of irreducible polynomials of degree 32 is too large.

To the best of our knowledge, no cryptographic CRC based on reducible polynomials has been proposed so far. The reason for this might be that it is necessary to analyze the security level obtained. This task is quite straightforward if the CRC generator polynomial is irreducible. However, in the general case, it becomes rather formidable. Details of our analysis will be appreciated by reading the Section VI.

The paper is organized as follows. Section II describes previous work. In Section III, we briefly review properties of hash functions which are used in the proof of the main result. In Section IV, we introduce the new family of hash functions. Section V analyses error-detecting capabilities of the presented hash functions. In Section VI, we make the security analysis. Section VII concludes the paper.

II. Related Work

Cryptographic hash functions have been extensively studied in the past, see Preneel [8] for an excellent survey. Message Authentication Codes (MACs) have also been thoroughly investigated, see Simmons for a comprehensive review [9]. Security of several types of MACs have been formally evaluated, including HMAC [10], CBC-MAC [11] and XOR-MAC [12].

Unconditionally secure message authentication codes were pioneered by Gilbert et. al. [13] and the theoretical basis was laid by Simmons [14]. The idea of constructing authentication codes through hash functions belongs to Carter and Wegman [15]. They were first to show how to combine hash functions with one-time pads in order to construct strong and efficient authentication algorithms. Their approach was further investigated and refined by Brassed [16], Desmond [17] and Krawczyk [6].

Stinson introduced a formal definition of ”almost strongly universal hash families” [18] which made possible considerably reducing the key length of unconditionally secure MACs. For more details on universal hashing, see his influential paper [19]. Black et al. described how universal hash families can be applied to construct efficient computationally secure MACs, e.g. UMAC [20].

A number of methods for cryptographic checksums and MACs based on stream ciphers were proposed, including Lai et. al. [21], Taylor [22], Johansson [23] and [24]. In these methods, a new representative of a hash family is generated for each message by using the pseudo-random generator of a stream cipher. In our case, as well as in the case of [6], the same hash function can be re-used for multiple messages. Only the random pad which is used for encrypting the hash values has to be updated for each message.

Rabin [25] first proposed the use of CRCs in the cryptographic context, namely for fingerprinting information (where the fingerprint is kept secret). However, his construction does not shift the message by n bit positions before the polynomial division. For this reason, it is nonsecure for message authentication even if the fingerprint is encrypted using a perfect one-time pad. For example, if we complement some of the n least significant bits of the message as well as the corresponding bits in the encrypted authentication tag, such a modification will not be detected by the fingerprint [6].

Krawczyk [6] has shown that, if the multiplication by x^n factor is added to Rabin's scheme [25], then the scheme becomes secure for message authentication provided that the tag is encrypted using one-time pad. He has also presented another interesting family of hash functions employing Toeplitz hashing where the columns of the matrix are formed from the consecutive states on an LFSR [6]. Such a construction has a bit lower hashing and authentication strength as the construction based on a random matrix, but its implementation cost is considerably smaller.

Apart from using CRCs in cryptographic context, there is another line of work focusing on message authentication codes capable of detecting, correcting or tolerating random errors [26]–[29]. MACs based on BCH and Reed-Solomon error-correcting codes have been presented in [26]. Several classes of approximate MACs designed of tolerate a small number of errors in a message have been developed, including [28], [29]. The idea of re-using a cryptographic MAC for detecting random errors in resource-constrained applications, such as Wireless Sensor Networks (WSN), has been proposed in [27].

III. Preliminaries

In this section we briefly review properties of hash functions from [6] which are needed for the proof of the main result. The reader familiar with the terminology can skip this session.

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

Definition 1: **[6]** A family of hash functions H is $\oplus$-linear if, for all messages M_1 and M_2 for all $h \in H$, we have $h(M_1 \oplus M_2) = h(M_1) \oplus h(M_2)$.

Definition 2: **[6]** A family of hash functions H is ε-balanced if, for any non-zero m-bit message M and for any m-bit string z,

$$\forall h \in H, Pr[h(M) = z] \leq \varepsilon,$$

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

Definition 3: **[6]** A family of hash functions is ε-opt-secure if, for any message M, no adversary can generate another message M' with a valid authentication tag with probability larger than ε when the tag is computed as $h(M')$ for a randomly chosen h.

Theorem 1: **[6]** A necessary and sufficient condition for a family H of hash functions to be ε-opt-secure is that

$$\forall M_1 \neq M_2, \forall z \in \{0,1\}^m, \forall h \in H, Pr[h(M_1) \oplus h(M_2) = z] \leq \varepsilon.$$

Theorem 2: **[6]** If H is $\oplus$-linear, then H is ε-opt-secure if and only if H is ε-balanced.

IV. Definition of Cryptographically Secure CRC

Let $p(x)$ be a polynomial of degree $n > 1$ over $GF(2)$ of type

$$p(x) = p_1(x) \cdot p_2(x) \cdot \ldots \cdot p_k(x) \tag{1}$$

where $k > 1$ and each $p_i(x)$ is an irreducible polynomial of degree $deg(p_i) > 1$ with a non-zero constant term[2]. Furthermore, for each pair $(i,j) \in \{1,2,\ldots,k\}^2$ it holds that either $deg(p_i) \neq deg(p_j)$ or $deg(p_i) = deg(p_j)$.

In other words, the factorization of $p(x)$ into irreducible polynomial gives us either all factors or equal degree, or all factors or different degrees. The case of mixed degrees, i.e. $p(x) = (x^2+x+1)(x^3+x+1)(x^3+x^2+1)$, is not covered.

We define a family $H_{m,n}$ of cryptographically secure CRC hash functions as follows.

Definition 4: For any m-bit message M and for each $p(x)$ of type (1), a hash function $h_p(M)$ is defined as

$$h_p(M) = M(x) \cdot x^n \ mod \ p(x).$$

The family $H_{m,n}$ is defined to consist of the set of all such functions, i.e. each $p(x)$ defines one member of the family $H_{m,n}$.

The hash functions defined above can be used for secure authentication of messages as follows. In order to compute a message authentication tag, a generator polynomial $p(x)$ is drawn randomly from the set of all possible polynomials of degree n over $GF(2)$ of type (1) and an n-bit random pad s is selected. The authentication tag t is computed as

$$t = h_p(M) \oplus s.$$

The addition of the random pad s is required in order to convert the linear CRC operation in Definition 4 to an affine one. Without such a modification, CRC of an all-zero message would always be 0, independently of the generator polynomial $p(x)$, enabling an attacker to inject all-zero messages.

V. Analysis of Error-Detecting Capabilities

It is well-known that a CRC based on an irreducible generator polynomial of degree $n > 1$ detects all burst errors of length n or less [3]. Next, we show that a cryptographically secure CRC based on a generator polynomial $p(x)$ of type (1) detects the same type of errors as well.

[2] Note that in $GF(2)$ the exist only one irreducible polynomial which does not have a non-zero constant term, namely the polynomial x.

Theorem 3: A CRC based on a generator polynomial $p(x)$ of type (1) detects the same type of burst errors as a CRC based on an irreducible generator polynomial of degree equal to $deg(p(x))$.

Proof: A CRC based on any generator polynomial $p(x)$ detects all errors except those which are a multiple of $p(x)$. If $p(x)$ is of type (1), then all its factors are polynomials with a non-zero constant term.

Let M be an m-bit message and let the CRC check bits be computed according to the Definition 4. Any k-bit burst error, $0 < k \leq n$, can be described by a polynomial of type

$$b(x) = x^j \cdot a(x)$$

where

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

for $i \in \{0,1,\ldots,k-1\}$ and $j \in \{0,1,\ldots,m+i\}$. The polynomial $b(x)$ is a multiple of $p(x)$ if and only if all factors of $p(x)$ are factors of $b(x)$ as well.

Since the degree of $p(x)$ is larger than the degree of $a(x)$ by at least 1, $p(x) \neq a(x)$. Therefore, to be a multiple of $b(x)$, $p(x)$ must be of type $p(x) = a(x) \cdot c(x)$, where $c(x)$ is a polynomial with a non-zero constant term of degree at least 1.

However, since either all other factors of $b(x)$ except $a(x)$ have a zero constant term (case $j > 0$), or $b(x) = a(x)$ (case $j = 0$), $c(x)$ cannot be a factor of $b(x)$. Thus a CRC based on a generator polynomial $p(x)$ of degree n of type (1) can detect all burst errors on length n or less.

□

To summarize, by making the traditional CRC cryptographically secure we do not sacrifice its capabilities to detect burst errors, which are a dominant type of errors in data communication and storage [1]. Note, however, that two n-bit CRCs based on different generator polynomials of degree n may have different capabilities for detecting multiple errors in non-adjacent bits. As for any error-detecting code, these capabilities are related to the code distance (minimal Hamming distance between any pair of codewords). Two different generator polynomials of degree n may result in codes with different code distances [30]. This applies to any CRC which uses variable generator polynomials, including the presented CRC and the CRC of Krawczyk [6].

VI. Security Analysis

In this section, we turn to the main task of analyzing the security of the new family of hash functions. We assume a typical setting in which the sender and the receiver transmit messages over an unreliable channel where messages can be maliciously modified [31]. The sender and the receiver share a secret key which is unknown to the adversary.

It is assumed that an adversary breaks the authentication if, after observing the message M and the tag t, he/she can find M' and t' such that $M' \neq M$ and $t' = h_p(M') \oplus s$. It is also assumed that the adversary knows the family $H_{m,n}$ of hash functions, but not the particular hash function h_p and the encryption pad $s \in \{0,1\}^n$.

The analysis is carried out by analyzing the distribution of CRCs over all messages of a given length. Note that we make a worst-case analysis, i.e. we assume that an adversary will try to maximize his/her chances to produce a "faked" CRC by choosing messages that maximize the success probability. Thus the adversary's success probability depends on the maximum probability that two different messages has the same CRC since this means that an adversary can replace one message with another without being detected.

The following Theorem shows that no adversary can succeed in breaking the authentication based on the presented CRC with the probability larger than ε where ε is given by (2).

Theorem 4: For any values of n and m, the family of hash functions given by Definition 4 is ε-opt-secure where

$$\varepsilon \leq \frac{(m+n)^k}{2^{n-k}}. \tag{2}$$

Proof: By Theorem 2, if a family of hash function is $\oplus$-linear, then it is ε-opt-secure if and only if it is ε-balanced. Clearly, the family of hash functions specified by Definition 4 is $\oplus$-linear because the division modulo a polynomial is a linear operation. Next, we show that this family is also ε-balanced.

On one hand, we observe that for any polynomial $p(x)$ of degree n, any non-zero m-bit message M and any n-bit string z, $h_p(M) = z$ if and only if $M(x) \cdot x^n \; mod \; p(x) = z(x)$. On the other hand, $M(x) \cdot x^n \; mod \; p(x) = z(x)$ if and only if $p(x)$ evenly divides $M(x) \cdot x^n \oplus z(x)$.

Let $g(x) = M(x) \cdot x^n \oplus z(x)$. Obviously, $g(x)$ is a non-zero polynomial of degree not larger than $m+n$ and $p(x)$ is a polynomial of degree n which evenly divides $g(x)$.

Let $p(x)$ be of type (1). In the sequel, we use n_i to denote the degree of the polynomials p_i in (1), for all $i \in \{1,2,\ldots,k\}$, and I_{n_i} to denote the number of irreducible polynomials of degree n_i.

Case 1: Let $n_i \neq n_j$ for all $i, j \in \{1,2,\ldots,k\}$.

Because of the unique factorization property, $g(x)$ can contain no more than

$$\frac{m+n-\sum_{j=1, j\neq i}^{k} n_j}{n_i} = \frac{m+n_i}{n_i}$$

irreducible factors of degree n_i, for each $i \in \{1,2,\ldots,k\}$. Therefore, if $n_i \neq n_j$ for all $i, j \in \{1,2,\ldots,k\}$, the number of hash functions in the family $H_{m,n}$ which map M into z is less than:

$$\frac{m+n_1}{n_1} \cdot \frac{m+n_2}{n_2} \cdot \ldots \cdot \frac{m+n_k}{n_k} \leq \frac{(m+n)^k}{n_1 \cdot n_2 \cdot \ldots \cdot n_k}.$$

On the other hand, there are $I_{n_1} \cdot I_{n_2} \cdot \ldots I_{n_k}$ ways to construct $p(x)$. Since there are

$$\frac{2^{n-1}}{n} \leq I_n$$

irreducible polynomials of degree n, we get

$$\frac{2^{n-k}}{n_1 \cdot n_2 \cdot \ldots \cdot n_k} \leq I_{n_1} \cdot I_{n_2} \cdot \ldots I_{n_k}.$$

So, the collision probability is at most

$$Pr[p(x) \text{ divides } g(x)] \leq \frac{(m+n)^k}{2^{n-k}}.$$

Case 2: Let $n_i = n_j$ for all $i, j \in \{1, 2, \ldots, k\}$. We have $deg(p_i) = deg(p)/k$.

There are

$$\frac{k2^{n/k-1}}{n} \leq I_{n/k}$$

irreducible polynomials of degree n/k. Our solution space is therefore larger than

$$\begin{pmatrix} N_1 + k - 1 \\ k \end{pmatrix} = \frac{(N_1 + k - 1) \cdot (N_1 + k - 2) \cdot \ldots \cdot N_1}{k!} \geq \frac{N_1^k}{k!} \tag{3}$$

where $N_1 = \frac{k2^{n/k-1}}{n}$.

On the other hand, there are at most $N_2 = (m+n)/(n/k) = k(m+n)/n$ irreducible factors of $g(x)$ each of degree n/k. All distinct k-tuples of them produce a reducible polynomial of degree n. So, the number of hash functions in the family $H_{m,n}$ which map M into z is at most:

$$\begin{pmatrix} N_2 \\ k \end{pmatrix} = \frac{N_2 \cdot (N_2 - 1) \cdot \ldots \cdot (N_2 - k + 1)}{k!} \leq \frac{N_2^k}{k!}.$$

On the other hand, the number of elements in this family is bounded by (3). Thus the collision probability is at most

$$Pr[p(x) \text{ divides } g(x)] \leq \frac{(\frac{k(m+n)}{n})^k}{(\frac{k2^{n/k-1}}{n})^k} = \frac{(m+n)^k}{2^{n-k}}.$$

□

Next, we derive a tighter bound on the collision probability for the case when the generator polynomial is a product of two irreducible polynomials.

Lemma 1: If $k = 2$, then ε in Theorem 4 is given by

$$\begin{array}{ll} \varepsilon \leq \frac{(m+n)^2}{2^n} & \text{if } deg(p_1) \neq deg(p_2) \\ \varepsilon \leq \frac{2(m+n)^2 - n(m+n)}{n(2^{n-3} + 2^{n/2-2})} & \text{if } deg(p_1) = deg(p_2). \end{array}$$

Proof: Let $deg(p_1) = n_1$. Then $deg(p_2) = n - n_1$.

Suppose that $g(x)$ contains r irreducible factors of degree $n - n_1$. Then, because of the unique factorization property, $g(x)$ contains at most

$$\frac{(m+n) - r(n - n_1)}{n_1}$$

irreducible factors of degree n_1. There are

$$\frac{r((m+n) - r(n - n_1))}{n_1} = -\frac{n - n_1}{n_1} \cdot r^2 + \frac{m+n}{n_1} \cdot r$$

distinct pairs of these factors and each pair will produce a distinct reducible polynomial of degree n. To find what is the largest number of hash functions in the family $H_{m,n}$ which map M into z, we need to find which choice of r maximizes the value of the quadratic function

$$ar^2 + br = -\frac{n - n_1}{n_1} \cdot r^2 + \frac{m+n}{n_1} \cdot r = 0. \tag{4}$$

This can be done by finding the x-coordinate of the vertex (maximum point) of the parabola representing the function (4). It is known that x-coordinate of the vertex is located at the point $-b/2a$, which is in our case

$$r = \frac{m+n}{2(n-n_1)}.$$

So, the value of the quadratic function is maximized if $g(x)$ contains $\frac{m+n}{2(n-n_1)}$ irreducible polynomials of degree $n-n_1$ and

$$\frac{(m+n) - \frac{m+n}{2(n-n_1)} \cdot (n-n_1)}{n_1} = \frac{m+n}{2n_1}$$

irreducible polynomials of degree n_1.

Suppose that

$$\frac{m+n}{2(n-n_1)} \leq I_{n-n_1}$$

and

$$\frac{m+n}{2n_1} \leq I_{n_1}.$$

Then, if $n_1 \neq n/2$, the largest number of hash functions in the family $H_{m,n}$ which map M into z is given by

$$\frac{m+n}{2(n-n_1)} \cdot \frac{m+n}{2n_1} = \frac{(m+n)^2}{4n_1(n-n_1)}. \tag{5}$$

On the other hand, if $n_1 \neq n/2$, then, for any fixed n_1, there are $I_{n_1} \cdot I_{n-n_1}$ ways to construct $p(x)$. Since

$$\frac{2^{n_1-1}}{n_1} \cdot \frac{2^{n-n_1-1}}{n-n_1} \leq I_{n_1} \cdot I_{n-n_1}$$

$$\frac{2^{n-2}}{n_1(n-n_1)} \leq I_{n_1} \cdot I_{n-n_1}$$

the collision probability for the case of $n_1 \neq n/2$ is

$$Pr[p(x) \text{ divides } g(x)] \leq \frac{(m+n)^2}{2^n}.$$

If $\frac{m+n}{2(n-n_1)} \geq I_{n-n_1}$ (or $\frac{m+n}{2n_1} \geq I_{n_1}$) then, there are not enough distinct polynomials of degree $n-n_1$ (or n_1) to maximize the value of the quadratic function (4). So even less than (5) hash functions in the family $H_{m,n}$ which map M into z. Thus, the collision probability is even smaller than the upper bound derived above.

If $n_1 = n/2$, then $g(x)$ can contain up to $(m+n)/(n/2) = 2(m+n)/n = N$ irreducible factors of degree $n/2$. All distinct pairs of them produce a reducible polynomial of degree n. So, the largest number of hash functions in the family $H_{m,n}$ which map M into z is given by:

$$\left(\begin{array}{c} N \\ 2 \end{array} \right) = \frac{2(m+n)^2}{n^2} - \frac{m+n}{n}.$$

On the other hand, if $n_1 = n/2$, then, there are

$$\frac{I_{n/2}(I_{n/2}+1)}{2}$$

ways to construct $p(x)$. Since there are

$$\frac{2^{n/2-1}}{n} \leq I_{n/2}$$

n	m	Collision probability ε		
		$p(x) = p_1(x) \cdot p_2(x)$		$p(x)$ is irreducible [6]
		$deg(p_1) \neq deg(p_2)$	$deg(p_1) = deg(p_2)$	
16	16	$1/2^{6.00}$	$1/2^{6.43}$	$1/2^{10.00}$
16	32	$1/2^{4.83}$	$1/2^{5.10}$	$1/2^{9.42}$
16	64	$1/2^{3.36}$	$1/2^{3.52}$	$1/2^{8.68}$
16	128	$1/2^{1.66}$	$1/2^{1.75}$	$1/2^{7.83}$
32	32	$1/2^{20.00}$	$1/2^{21.42}$	$1/2^{25.00}$
32	64	$1/2^{18.83}$	$1/2^{20.09}$	$1/2^{24.42}$
32	128	$1/2^{17.36}$	$1/2^{18.51}$	$1/2^{23.68}$
32	256	$1/2^{15.66}$	$1/2^{16.74}$	$1/2^{22.83}$
32	512	$1/2^{13.83}$	$1/2^{14.87}$	$1/2^{21.91}$
32	1024	$1/2^{11.91}$	$1/2^{12.93}$	$1/2^{20.96}$
64	64	$1/2^{50.00}$	$1/2^{52.42}$	$1/2^{56.00}$
64	128	$1/2^{48.83}$	$1/2^{51.09}$	$1/2^{55.42}$
64	256	$1/2^{47.36}$	$1/2^{49.51}$	$1/2^{54.68}$
64	512	$1/2^{45.66}$	$1/2^{47.74}$	$1/2^{53.83}$
64	1024	$1/2^{43.83}$	$1/2^{45.87}$	$1/2^{52.91}$

TABLE I

COMPARISON OF THE COLLISION PROBABILITIES OF THE PRESENTED CRC AND WITH THE CRC OF KRAWCZYK [6].

irreducible polynomials of degree $n/2$, our solution space is therefore larger than

$$2^{n-3} + 2^{n/2-2}.$$

Thus for $n_1 = n/2$ the collision probability is

$$Pr[p(x) \text{ divides } g(x)] \leq \frac{2(m+n)^2 - n(m+n)}{n(2^{n-3} + 2^{n/2-2})}.$$

□

To better illustrate the derived bound, in Table I we show the collision probabilities for the case $k = 2$ for the CRCs $n = 16, 32, 64$ and the selected message lengths m. The 5th column shows the collision probability for Krawczyk's CRC [6], i.e. when a single irreducible polynomial is used as a generator polynomial, which is bounded by $\sigma \leq \frac{m+n}{2^{n-1}}$.

As we can see from Table 1, if a generator polynomial is a product of two irreducible polynomials, the collision probability is always higher than the collision probability of [6]. However, if the numbers in columns 3 and 4 meet the security requirements of a given application, the presented method might still be the preferred option because it can be implemented with less resources. It seems to be particularly attractive for short messages, for which the difference in security between the two approaches is smaller.

We can also see from the table that, for two equal-size polynomials (column 4), the collision probability is smaller compared to the collision probability for two unequal-size polynomials (column 3). Clearly, it is easier to generate equal-size polynomials. For example, we can maintain a database of polynomials of degree $n/2$, where n is the required CRC size, and choose $p_1(x)$ and $p_2(x)$ from this database at random. Therefore, we consider the case of two equal-size polynomials most appealing.

VII. Conclusion

We introduced a new family of hash functions based on CRC and analyzed the security level obtained. To our best knowledge, this is the first scheme using reducible polynomials in its construction. The method seems to be particularly attractive for short messages, for which the difference in security between the presented approach and the approach of [6] is smaller.

VIII. Acknowledgements

The first author was supported by the research grant No SM14-0016 from the Swedish Foundation for Strategic Research.

References

[1] T.-B. Pei and C. Zukowski, "High-speed parallel CRC circuits in VLSI," *IEEE Transactions on Communications*, vol. 40, pp. 653 –657, Apr. 1992.

[2] T. Ramabadran and S. Gaitonde, "A tutorial on CRC computations," *Micro, IEEE*, vol. 8, pp. 62 –75, Aug. 1988.

[3] W. Peterson and D. Brown, "Cyclic codes for error detection," *Proceedings of the IRE*, vol. 49, pp. 228 –235, Jan. 1961.

[4] S. Golomb, *Shift Register Sequences*. Aegean Park Press, 1982.

[5] 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.

[6] 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.

[7] 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.

[8] B. Preneel, "The state of cryptographic hash functions," in *Lectures on Data Security, Modern Cryptology in Theory and Practice, Summer School, Aarhus, Denmark, July 1998*, (London, UK), pp. 158–182, Springer-Verlag, 1999.

[9] G. Simmons, "A survey of information authentication," *Proceedings of the IEEE*, vol. 76, pp. 603–620, May 1988.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] D. R. Stinson, “Universal hashing and authentication codes,” *Des. Codes Cryptography*, vol. 4, pp. 369–380, Oct. 1994.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] T. Johansson, “A shift register construction of unconditionally secure authentication codes,” *Designs, Codes and Cryptography*, vol. 4, no. 1, pp. 69–81, 1994.

[24] 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.

[25] M. Rabin, “Fingerprinting by random polynomials,” Tech. Rep. TR-15-81, Center for Research in Computing Technology, Harvard Univ., Cambridge, Mass, 1981.

[26] 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.

[27] C. Karlof, N. Sastry, and D. Wagner, “Tinysec: A link layer security architecture for wireless sensor networks,” in *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys ’04, (New York, NY, USA), pp. 162–175, ACM, 2004.

[28] 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.

[29] 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.

[30] P. Koopman and T. Chakravarty, “Cyclic redundancy code (crc) polynomial selection for embedded networks,” in *Dependable Systems and Networks, 2004 International Conference on*, pp. 145–154, June 2004.

[31] D. Stinson, *Cryptography Theorey and Practice*. Chapman & Hall/CRC, 3rd edition, 2006.