Scientific Research Publishing

Journal of Computer and Communications, 2020, 8, 214-230
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Modified Vanstone's Construction of lightweight MAC for Vehicular On-Board IT Systems

István Vajda

Department of Informatics, Technical University of Budapest, Budapest, Hungary
Email: vajda@hit.bme.hu

How to cite this paper: Vajda, I. (2020) Modified Vanstone's Construction of lightweight MAC for Vehicular on-Board IT Systems. *Journal of Computer and Communications*, **8**, 214-230.
https://doi.org/10.4236/jcc.2020.812019

Received: November 30, 2020
Accepted: December 27, 2020
Published: December 30, 2020

Abstract

We propose a lightweight construction, a modification of Vanstone's MAC construction, for the message authentication of the communication between Electronic Control Units (ECUs) in distributed car control systems. The proposed approach can solve the task of error control and authentication in unified algorithmic technology, called MAC (Message Authentication Code) with ECC (Error Correction Code). We follow a provable approach in the design of the cryptographic primitive, where we quantify the security measures in the parameters of the system. Provable security approaches are missing in the field of secure in-vehicle communication.

Keywords

Secure in-Vehicle Communication, Lightweight Authentication, MAC with Error Correction Capability, Provable Security

1. Introduction

Many mechanical systems in automobiles have been replaced by electronic systems. There are simpler tasks like wiper control and also safety-critical functions like anti-lock brake controls and airbags. The safety-critical functions are typically time-critical. There are standard communication protocols for the communication between embedded systems in automobiles. The CAN (Controller Area Network) protocol [1] was initially developed for the automotive industry. Nowadays, CAN is used in many other real-time applications, in industrial and home automation and medical equipment. Further widespread used protocols are CAN FD, LIN, FlexRay, and Automotive Ethernet. Motivated by the low-cost nature of automotive components, the Electronic Control Units (ECUs) have

considerably less computational resources and data rates than standard computers.

The CAN bus by design is not secure, and it describes the communication protocol [1]. The behavior of connected devices can be manipulated and compromised which may lead to the capability of modifying the behavior of the vehicle [2]. A malicious adversary might try to inject messages or to change messages unnoticed [3]-[10]. A malicious attacker must not have the ability to take the control of critical systems inside the vehicle via modification or fabrication of messages transmitted on the bus. Therefore secure authentication of forwarded messages has to be guaranteed. The other group of attacks aims at privacy, such as privacy of speech communication over microphone signals, location privacy (GPS data), id privacy (the identity of vehicle occupants) [11].

Several cryptography-based solutions have been proposed to prevent attacks. Message Authentication Codes (MACs) are based on symmetric cryptography, allowing fast and efficient computation, especially on ECUs with limited computational power. Paper [12] proposed the use of MACs for authentication proposes in CAN. In work [13] the authors did the same for FlexRay protocol, where they proposed the TESLA protocol. While TESLA supports sender authentication, it does not authenticate the messages. Recall, TESLA protocols are a family of related lightweight authentication protocols, designed to provide authenticated broadcast capabilities. However, the time delays introduced by the TESLA (by its very idea of construction) contradict the strict latency requirement in automotive applications.

Paper [14] proposes the key-hashed message authentication code HMAC to hash the messages for message authentication. However, by its computational complexity running it on resource-restricted ECU processor is not a viable solution.

Protocol LiBrA-CAN ([8]) authenticates senders at the receiving ECUs via Mixed Message Authentication Codes (M-MACs). Keys are pre-shared and assigned to groups of ECUs. There are no key exchanges.

Protocol CANAuth ([14]) allows broadcast authentication and keys are assigned for message groups. CANAuth also requires pre-shared keys. There are no key exchanges.

Paper ([15]) proposes a security framework for vehicular systems (VeCure). They divide the ECUs into two categories the low-trust group and the high-trust group. All the ECUs in the latter group share a (long-term) secret symmetric key for message authentication using a keyed SHA-3 hash function. Their main idea is to use offline (pre-) computations to minimize the online message processing delay.

Protocol LeiA [4] assumes tamper-resistant memory on ECUs that stores shared lifetime keys. Session keys are used for keying the MAC algorithm for message authentication. The session keys are generated from the long-term key using the same MAC algorithm and are changed periodically. The idea is that if the session key becomes compromised, the attacker can compute valid MACs

only until the epoch changes (under the assumption that long-term keys are safe). No concrete algorithm for MAC is mentioned, they assume a MAC with EU-CMA security. (In contrast, we construct a MAC with such type security.)

In approach LASAN ([10]) a complete arsenal of public-key cryptography and hardware support is assumed (like secure timestamps, public key certificates, hardware security module (HSM) at a trusted ECU and tamperproof key storages at each ECU). The initial key exchange between the ECUs and the security module is based on a public-key algorithm. One of their ideas is to run these computational expensive tasks off-line when the vehicle is not in use. During this off-line phase (called ECU authentication) the security module authenticates to all ECUs and all ECUs authenticate to the module, furthermore, a symmetric ECU key is established for all ECU. In the online phase, cryptographic computations are fully controlled by the security module. For encryption of messages transmitted between ECUs, unique keystreams are generated and sent to the ECUs by the module on request of ECUs. Their other idea (called stream authorization) eliminates the need for explicit authentication directly between pairs of ECUs. A drawback of security solutions based on HSM is that they are costly and require additional physical area and power.

Note, even public key technology cannot eliminate the exposure of private keys to compromising attacks on ECUs as the private part of the key pair needs such protection. The approach in [16] tries to minimize this problem by avoiding long-term keys on ECUs. They assume that a special physical source of entropy (PUF, Physically Unclonable Function) is deployed on each ECU and also on a key server. (A PUF can generate unique random elements and can reproduce it when it is queried with the same challenge value.)

The key server plays the role of a simplified public key directory. Each time when the engine is ignited a registration phase is run and each ECU generates a new public-private key pair using the PUF. The public keys of all ECUs are sent to the key server and stored on the server. The private part is erased. Note, this communication phase is unprotected, and the public key is transmitted without any cryptographic protection (authentication), so a timely adversary may attack this transmission. This is a serious vulnerability as this public key is used in the subsequent phase as an authenticator tag (see subsequently).

The session key establishment between two ECUs goes as follows. The ECU regenerates the same public/private key pair by using the PUF. The ECU initiating the key establishment and the server generate an auxiliary symmetric key, by applying public-key algorithm ECC DH on input the public key of the server and the private key of the ECU (note, already we are in the online phase of computation and a public key algorithm is used). The public key of the ECU is encrypted with the auxiliary key. The server decrypts the plaintext (*i.e.* the public key) and compares it to the registered one (the decryption key is computed from the secret key of the server and the public key of the ECU). The server sends the list of ECUs public keys encrypted with the public key of the requesting ECU. Now, the initiating ECU can establish a session key with the wanted peer ECU

by sending random elements encrypted with the party's public key (note, another public key algorithm is used). Once the session key is established the ECUs erase the private key.

In sum, this approach efficiently decreases the time the private key of ECUs is exposed. Note, however, the private key is implicitly stored in the PUF, therefore an adversary accessing the PUF can regenerate the key. There is an unprotected phase of communication and several different algorithms have to be run to establish a single session key between two ECUs. Furthermore, public key algorithms are run in the online phase.

For computational complexity reasons, most of the proposals are based on symmetric-key primitives. The purpose of the choice is to reduce latency in time-critical real-time communication. Typically, related primitives include the AES encryption algorithm for privacy protection, the AES-based Cipher-based Message Authentication Code (CMAC) and keyed hash functions (HMAC (e.g. [14]), SHA-3 (e.g. [15]) for message authentication. Proposals have also been made to use public-key cryptography (e.g. [10] [16]).

One of the fundamental problems is secure key management for the automotive environment. The proposed solutions show a wide variety. It is a difficult task to strike a balance between cost, latency requirement, and security. At one end pre-programmed, lifetime group keys (e.g. [8] [14] [15]), at the other adapted public key management (e.g. [16]). This important problem is open, and there is no final solution yet.

The current ECU architecture can only be loaded with limited computational overhead. In contrast, cryptographic algorithms are typically computationally intensive. Separate special hardware components are needed to resolve this contradiction within the recent architecture. Examples of related suggestions are Secure Hardware Extension (SHE) [17], security module [10], key server [16], and tamperproof key storage [4].

In this work, we present a construction for a symmetric-key lightweight MAC algorithm with a formal proof of security. Our proved claims quantify the corresponding security notions in variables of system dimensions. This provides the advantage of finding trade-offs between dimensions-related complexity and the level of security. The approach is inherently able to handle the tasks of authentication and error control together in a unified computational approach.

The full-fledged realization remains also provably secure under the reasonable assumption that AES in CTR mode generates a pseudorandom stream. (Informally, a pseudorandom stream cannot be distinguished from the ideally random one for an efficient adversary except a negligible probability.) This implies security even against quantum attackers: when the one-time stream is realized with AES-CTR mode keystream it remains quantum secure when the key length of the underlying AES is doubled [18]. Recall, post-quantum security may be a decisive advantage even within 10 years.

As for the complexity of our construction, one-time pads, MAC keys can be generated independently from the (next) message(s) between peer ECUs (in

sync with the peer). It follows that the MAC generator polynomials can also be precomputed. Therefore, the associated computational load can be eliminated from the online computational complexity of MAC. It follows the on-line computation complexity of MAC is essentially equivalent to a CRC computation over the field the MAC is constructed.

A further advantage of the construction is that generation and verification of the MAC can be started as the first message elements arrive, *i.e.* there is no need to wait for the full message to arrive. Indeed, the division by the precomputed generator polynomial provides this possibility.

Instead of error detection and automatic retransmissions, it is better to have an option of error correction capability. The optional use means the following: the error control code can run in error detection mode, and when an erroneous packet is detected there are two options, retransmission or error correction. Until the computational complexity of error correction is well below the complexity of the recomputation of cryptographic primitives (typical in case of a few errors), the built-in option improves the latency properties. Consider the following related example.

The organization of the paper is as follows. In Section 2 we give related works. In Section 3 we present our constructions. Conclusions are given in Section 4.

2. Related Works

An approach for the integration of MAC and error detection can be found in [19]. In some sense, this is the closest work to our approach. Briefly, they process a message to be forwarded the following way. First MAC is computed (they use CMAC) and truncated to a small size. The truncated MAC is appended to the message and CRC is computed for the extended message. In the last step, the truncated MAC is removed and the message is forwarded together with the CRC. Intuitively, this keyed CRC implicitly will incorporate "some information" from the MAC. No analysis (not to mention formal proof) is presented about quantifiable security properties.

Our main result is a lightweight MAC construction with formal proof of security. Additional value (and option) that error control can naturally be integrated into the MAC construction. In contrast to the above-cited publications (except publication [19]) we use the attribute lightweight directly for the complexity of the MAC and not for the lightening of the surrounding key management. (Of course, our symmetric key construction can be used with any secure key management, in particular, with a secure session key protocol.)

Our solution envisions both privacy and authentication issues. We are considering symmetric key cryptographic primitives. Lightweight MAC authentication is combined with CTR mode encryption.

The idea of the construction comes from a classic result of Krawczyk [20]. Briefly and informally, the two parties (transmitter and receiver) choose an ECC code from a large set of such codes using a symmetric secret key. Assuming the

adversary cannot guess the current key except a negligible probability, she will not know the algorithm for the computation of the MAC.

Error control capability can be integrated completely into the MAC construction. Vanstone's [21] construction is essentially such an approach, they take randomly (and secretly) an e-error correcting code from a set of codes with parameters (*N, K, e*) over GF(*q*). The receiver knowing the (symmetric) key can identify the actual code can run the error correction algorithm and remove the channel errors in the received packet. As the next step, the receiver verifies the MAC on the error-free packet. The elements of the root set of the (MAC/ECC) generator polynomial are in fixed algebraic relation (according to the definition of the ECC code).

Our algorithm is a modification of Vanstone's approach for improved characteristics. We divide the set of roots of the keyed MAC generator polynomial into two disjoint subsets. In one of the subsets, the roots are taken randomly from the whole set of roots, in the other subset the roots follow the algebraic relation dictated by the ECC code. By choosing the roots randomly we can increase the size of the set of the keyed MAC generator polynomials and we can set a finer trade-off between secrecy and error-control capabilities.

The published security constructions for automotive applications almost exclusively miss formal proofs of security. We provide formal security proof for our MAC construction. One step of the construction uses a random pad ([20]). The security proof refers to the hybrid MAC protocol with an ideal random pad. The security guarantee is statistical, meaning that the best strategy of an adversary is guessing the actual key. Concretely, we provide upper bounds on this probability of guessing in the parameters of the algorithm. In a full-fledged realization for random pads, we can use the keystream generated in CTR mode by the AES algorithm. We can claim quantum security is extended for the full realization. Indeed, the attacks by using quantum Grover's algorithm against symmetric ciphers can be blocked effectively by doubling the key size [19]. Resistance to replay attacks is included in the construction. Furthermore, the construction allows offline precomputations: the computation of MAC can be started as soon as the first message symbols are accessible (*i.e.* without waiting for the arrival of the whole message).

3. Constructions

The basic idea of the MAC construction from linear codes comes from the classic paper of Krawczyk [20]. A keyed CRC generator polynomial was used and the MAC was the CRC plus a random pad. Vanstone [21] noticed that an error correction capability is inherent in this construction. In construction [21] they propose to use linear error-correcting code (ECC) codes in Krawczyk's construction. They choose a code randomly from a family of codes (Reed-Solomon or BCH codes) with a chosen parameter triplet (N, K, t). They choose randomly a generator polynomial $F_k(x) = g_k(x)$, where $g_k(x)$, is the generator po-

lynomial of a randomly chosen ECC code. They generate systematic cyclic code C with generator polynomial $F_k(x)$. The systematic codeword $c=(m,u)$ of code C consists of the message (m) and the MAC (u), *i.e.* polynomial $u(x)$ is the polynomial remainder obtained by dividing polynomial $m(x)x^z$ by MAC generator polynomial $F_k(x)$, $z=\deg F_k(x)$.

The idea of our modified construction and corresponding improvement is as follows. We also start from systematic cyclic ECC code over $GF(q)$, where we also will consider binary ($q=2$) and non-binary ($q=2^m, m>1$) codes. We compose generator polynomial $F_k(x)$ from two-component polynomials. Assume $g(x)$ is a generator polynomial of the ECC. We choose also another polynomial $f_k(x)$ that is keyed with symmetric MAC-key k. The only restriction on $f_k(x)$ is that it is chosen relative prime to $g(x)$. Disjoint but otherwise randomly chosen root set is chosen for $f_k(x)$. We generate systematic cyclic code C with generator polynomial $F_k(x)=f_k(x)g(x)$. Note, a codeword $c\in C$ is also an ECC codeword. Indeed, property $F_k(x)\,|\,c(x)$ implies that $g(x)\,|\,c(x)$. Consequently, parity segment u (of systematic codeword $c=(m,u)$) is a symmetric key MAC with error correction capability. By choosing the roots of $f_k(x)$ randomly we can increase the size of the set of the keyed MAC generator polynomials and we can set a finer trade-off between secrecy and error control capabilities.

Telling otherwise, we have increased the variability of MAC generator polynomials within the construct. Note, the set of roots of Vanstone's polynomial $g_k(x)$ must meet rigorous algebraic constraint (by the rules of the code construction), in contrast, the roots of component polynomial $f_k(x)$ are chosen at random from the set of all roots (except the roots of $g(x)$). Note, polynomial $f_k(x)$ does not require further algebraic properties, only the number of such polynomials is important.

We adapt the definitions of security measures from [20] [21].

3.1. Constructions Based on Reed-Solomon Codes

Consider a (standard) cyclic Reed-Solomon (RS) code $C(N, K)$ over $GF(q)$. Let the generator polynomial be defined by

$$g(x)=(x-\propto)\left(x-\propto^2\right)\cdots\left(x-\propto^{N-K}\right),$$

where $\propto\in GF(q)$ has order N, $N\,|\,(q-1)$. The minimum code distance is known to be $d=N-K+1$. We will denote by a polynomial $a(x)$ the $j-1$ -degree polynomial with coefficients $a=(a_{j-1},\cdots,a_0), a\in\left[GF(q)\right]^j$.

Let $m\in\left[GF(q)\right]^s$ denote the message to be authenticated by an authenticator tag $MAC_k(m)$ with ECC capabilities. The parameters of the authenticator tag are as follows: $MAC_k(m)\in\left[GF(q)\right]^z$, where $s+z=N, z>N-K$, furthermore k stands for a one-time secret symmetric key. Let $v=z-(N-K)$.

For a message m we generate authenticator tag $MAC_k(m)$ with the following algorithm:

Construction 1

1) We choose a fresh MAC-generator polynomial $G_k(x) = f_k(x)g(x)$, where v-degree monic polynomial $f_k(x)$ is defined by choosing it's set of roots $\{\beta_1, \cdots, \beta_v\}$ such that $\beta_i, i = 1, \cdots, v$ are different randomly chosen elements from the set $\{GF(q) \setminus 0\} \setminus \{\alpha, \cdots, \propto^{N-K}\}$.

2) We compute pre-MAC value $MAC'_k(m)$ as a polynomial remainder using the MAC-generator polynomial

$$MAC'_k(m)(x) = -m(x) \cdot x^z \bmod G_k(x)$$

3) We add random pad $r \in [GF(q)]^z$:

$$MAC_k(m) = MAC'_k(m) + r \qquad \square$$

Steps 2-3) in Construction 1 follow [20]. (Without applying Step 3), by seeing the MAC value for messages with low Hamming weight an adversary might be able to reconstruct the MAC-generator polynomial.) Step 3) is stream encryption of the MAC segment. In a realization, such keystream could be generated by using the AES algorithm in counter mode. Note, because of Step 3) encryption is inherent in the construction, and privacy protection of the message can be achieved by "rolling the keystream over" both the message and the MAC parts of the packet.

In the analysis, we assume ideal one-time-key k and ideal random pad r. In other words, we analyze a hybrid MAC protocol ideal in these keys. The best strategy for an adversary in an attack against the hybrid MAC protocol is guessing. Essentially, in the analysis, we give upper bounds on this probability of guessing. In an application, this probability should be set negligible by appropriate selection of the parameters of the construction. In a (provably secure) realization the ideal keystream is substituted by a pseudorandom stream. For example, assuming that AES in CTR mode produces a pseudorandom stream, a full-fledged MAC protocol with AES becomes a secure realization.

Following publications [20] [21], the security of MAC constructs are measured with the following two characteristics: ε-balanced property and e-error-forgery-resistance.

***Definition* 1** [21]: The *ε-balanced property* of MAC $MAC'_k(m)$ means the following requirement: for all $m \neq 0, m \in [GF(q)]^s$, $u \in [GF(q)]^z$

$$P(MAC'_k(m) = u) \leq \varepsilon \qquad \square$$

The intuition behind this definition is as follows. For an arbitrary but fixed, extended message (m, u) we want to know the number of all MAC keys (alternatively all MAC generator polynomials $f_k(x)$ that match the extended message, *i.e.* such that equality $MAC'_k(m) = u$ holds. Accordingly, probability $P(MAC'_k(m) = u)$ is the probability that a randomly chosen key k matches. It follows that the probability that an adversary can reproduce MAC u by randomly choosing a key is bounded by ε.

Proof-technically, the ε-balanced property is used as a pre-calculation for the probability of e-error-forgery-resistance.

Recall, when the code distance of the underlying ECC code is $2e+1$, the code

can correct (symbol) errors with Hamming weight $\le e$. If (m,t) is a correct message-tag pair and (m',t') is another message–tag pair such that

$$dist\big((m,t),(m',t')\big) \le e$$

then "noisy" pair (m',t') is called an *equivalent acceptable pair* (due to the possibility of successful error correction).

***Definition* 2** [21]: $MAC_k(m)$ provides *e-error-forgery-resistance* with probability ϵ, if given any message-tag pair within distance e of a correct pair (m,t), no adversary succeeds by finding a pair $(m',t'), m \neq m', d(m,m') > e$ such that the latter pair is an acceptable pair with probability larger than ε. □

Note, forging an acceptable pair such that it is equivalent to a pair eavesdropped by a MIM adversary can be trivial simply by "adding limited noise". In contrast, Definition 2 requires that the forged (and potentially noisy) pair must have a message part outside the e-radius of the message part of the correct pair underlying the eavesdropped pair (*i.e.* a forged authenticator tag must correspond to a new message).

Properties of Construction 1

***Claim* 1:** Construction 1 is ϵ-balanced with

$$\varepsilon \le \left(\frac{1}{q/N-1}\right)^{v} \tag{1}$$

Example 1: Some typical evaluation results for Construction 1 are shown in Table 1.

***Proof of Claim* 1:** For a given message m and tag candidate u, the number of different polynomials $G_k(x)$ such that $G_k(x)$ divides polynomial $(m \| u)(x)$, $\deg C(x) \le N-1$ is at most $\binom{K-1}{v}$. To see this, note that the number of the different roots of $C(x)$ in $GF(q)$ is at most $\deg C(x)$ and among them, we have also the fixed $N-K$ roots of ECC generator polynomial $g(x)$. Therefore, polynomials of type $f_k(x)$ can take their roots from a set with size at most $N-1-(N-K)$.

Since by Construction 1 the number of all possible MAC-generator polynomials $G_k(x)$ is $\binom{q-1-(N-K)}{v}$, we get upper bound

$$\varepsilon \le \frac{\binom{K-1}{v}}{\binom{q-1-(N-K)}{v}} = \frac{(K-1)(K-2)\cdots(K-v)}{\big(q-\big((N-K)+1\big)\big)\cdots\big(q-\big((N-K)+v\big)\big)}$$

$$\le \frac{K^{v}}{\big(q-\big((N-K)+v\big)\big)^{v}} \le \frac{1}{\big(q/K-\big((N-K)+v\big)/K\big)^{v}}$$

In the special but practical case when the length of the message is at least the length of the parity segment (of the ECC), *i.e.* when inequality $(N-K)+v \le K$ holds, we arrive at the simple upper bound

Table 1. Evaluation examples for construction 1.

symbol size (byte)	q	N (symbols)	message length (bits)	v	MAC size (bits)	$\varepsilon \leq$
4	2^{32}	$2^8 \pm 1$	$\sim 2^{13}$	3	96	2^{-72}
3	2^{24}	$2^8 \pm 1$	$\sim 24 \cdot 2^8$	4	96	2^{-64}
2	2^{16}	$2^8 \pm 1$	$\sim 2^{12}$	6	96	2^{-48}
2	2^{16}	$2^8 \pm 1$	$\sim 2^{12}$	7	112	2^{-56}

$$\varepsilon \leq \frac{1}{\left(\frac{q}{K}-1\right)^{v}}. \qquad \square$$

***Claim* 2:** Construction 1 is e-error-forgery-resistant with probability ε.

***Proof of Claim* 2:** Assume the adversary sees a transmitted codeword that may contain transmission errors. The adversary wants to find out the actual MAC-generator polynomial (or equivalently the actual MAC-key) to fabricate a valid MAC for a fake message. The adversary first decodes the received word by using the decoding algorithm of the ECC. There are two cases.

Case 1: Assume the adversary can decode successfully the transmitted codeword $C(x)$. The actual MAC-generator polynomial will be in the set of all matching potential MAC-generator polynomials. Accordingly, the probability of successful guessing of the actual generator polynomial is bounded by probability ε.

Case 2: Assume now that the number of errors is larger than e. Note also that if the number of errors is larger than e, then the ECC-decoder will decode some message $m^*, m^* \in \left[GF(q)\right]^{s+v}$ that is not a MAC-extended message (generated by some possible MAC-generator polynomial), in general. Even, if it happens to be a MAC-extended message, *i.e.* dividable by polynomials of type $f_k(x)$ for some key k, the polynomial with the actual key k will not be in this set in general. It follows that the probability of successful guessing of the actual MAC-generator polynomial is bounded by probability ε (typically much smaller). $\square$

A version of Construction 1

Note, upper bound (1) on probability ε for Construction 1 is meaningful only if N is a proper divisor of $q-1$. The numerical results in Example 1 correspond to such parameter settings.

We can eliminate such dependence on parameter settings by choosing the fields of roots for generator polynomials $g(x)$ and $f_k(x)$ independently. The idea is to choose a sufficiently large extension field $GF(q')$ of field $GF(q)$ for the generation of polynomial $f_k(x)$, to boost the magnitude of ratio q'/K.

For the simplicity of the presentation, we formalize the idea for the specific case of $q' = q^2$. Minimal polynomials of the elements of field $GF(q^2)$ over field $GF(q)$ have degree 1 or 2 (divisors of 2). Minimal polynomials with degree 1 are those with roots in $GF(q)$, *i.e.* those in set $A = \{x - \gamma : \gamma \in GF(q)\}$. Minimal polynomials with degree 2 are in set

$$B=\left\{\left(x-\beta\right)\left(x-\beta^{q}\right):\beta\in GF\left(q^{2}\right)\setminus GF\left(q\right)\right\}.$$

The size of set B is $\left(q^{2}-q\right)/2$. Recall, minimal polynomials are irreducible. We will choose the factors of $f_{k}\left(x\right)$ randomly from set B. We modify Construction 1 in its first step as shown below:

Construction 2

1) We choose a fresh MAC-generator polynomial $G_{k}\left(x\right)=f_{k}\left(x\right)g\left(x\right)$, where v-degree polynomial $f_{k}\left(x\right)$ is defined as the product of $v/2$ different randomly chosen elements from set B. □

Properties of Construction 2

***Claim* 3:** Construction 2 is ε-balanced with

$$\varepsilon\le\frac{1}{\left(q-2\right)^{v/2}} \tag{2}$$

***Proof of Claim* 3:** The number of different polynomials $f_{k}\left(x\right)$ is $\binom{\left(q^{2}-q\right)/2}{v/2}$. For a given message m and tag candidate u, the number of different polynomials $G_{k}\left(x\right)$ such that $G_{k}\left(x\right)$ divides polynomial $\left(m\|u\right)\left(x\right)$ is at most $\binom{K'/2}{v/2}$, where $K'=K-1$. Indeed, the maximum is reached if $\left(m\|u\right)\left(x\right)=F\left(x\right)g\left(x\right)$, where K'-degree polynomial $F\left(x\right)$ is a product of different elements from set B. Whence for the probability ε we get the following upper bound

$$\begin{aligned}\varepsilon&\le\frac{\binom{K'/2}{v/2}}{\binom{\left(q^{2}-q\right)/2}{v/2}}\le\frac{\frac{K'}{2}\left(\frac{K'}{2}-1\right)\cdots\left(\frac{K'}{2}-\frac{v}{2}+1\right)}{\frac{q^{2}-q}{2}\left(\frac{q^{2}-q}{2}-1\right)\cdots\left(\frac{q^{2}-q}{2}-\frac{v}{2}+1\right)}\\&\le\frac{K'\left(K'-2\right)\cdots\left(K'-v+2\right)}{\left(q^{2}-q\right)\left(q^{2}-q-2\right)\cdots\left(q^{2}-q-v+2\right)}\\&\le\frac{q\left(q-2\right)\cdots\left(q-v+2\right)}{\left(q^{2}-q\right)\left(q^{2}-q-2\right)\cdots\left(q^{2}-q-v+2\right)}\\&\le\frac{q^{v/2}}{\left(q^{2}-q\right)\left(q^{2}-q-2\right)\cdots\left(q^{2}-q-v+2\right)}\\&\le\frac{1}{\left(q-1\right)\left(q-1-\frac{2}{q}\right)\left(q-1-\frac{4}{q}\right)\cdots\left(q-1-\frac{v-2}{q}\right)}\\&\le\frac{1}{\left(q-1-\frac{v-2}{q}\right)^{v/2}}\le\frac{1}{\left(q-2\right)^{v/2}}\end{aligned}$$

where in the last step we used inequality $1+\frac{v-2}{q}\le 2$. □

***Claim* 4:** Construction 1 is e-error-forgery-resistant with probability ε.

The proof logic of Claim 4 is similar to the proof Claim 2.

Example 2: Some typical evaluation results for Construction 2 are shown in Table 2.

3.2. Construction Based on BCH Codes

We recall the definition of the generator polynomial of a BCH code over $GF(q)$. For block-length $N = q^n - 1$, for some n the generator polynomial $g(x)$ of an e-error correcting BCH code over $GF(q)$ is constructed by the following algorithm:

1) Choose a prime polynomial of degree n (with root $\propto$) and construct $GF(q^n)$.

2) Find $f_j(x)$, the minimal polynomial of $\propto^j$ for $j = 1, \cdots, 2e$.

3) $g(x) = LCM\left[f_1(x), \cdots, f_{2e}(x)\right]$.

Accordingly, $N - K = \deg(g(x)) \le 2en$, where K is the message length. As the most important practical case is the binary ($q = 2$), subsequently we assume binary BCH code. (The construction can be applied to the general value of q straightforwardly).

We will choose the keyed-polynomial $f_k(x)$ to be the product of different irreducible polynomials randomly chosen from the set of all irreducible polynomials with a given degree (degree n) over $GF(q)$. Recall, here we have a big and a small field, $GF(Q), Q = q^n$ and $GF(q)$, respectively. The big field is the field of roots for the generator polynomial, the small one is the field of the coefficients of codewords.

We also recall that the number of all binary monic irreducible polynomials of degree n over a field with q elements is given by

$$I_2(n) = \frac{1}{n}\sum_{i|n}\mu(d)2^{n/i}$$

Table 2. Evaluation examples for construction 2.

Case $N = q - 1$ (primitive ECC code-length)						
symbol size (byte)	q	N(symbols)	message length (bits)	v	MAC size (bits)	$\varepsilon \le$
4	2^{32}	$2^{32} - 1$	$32 \cdot N$	4	128	2^{-64}
2	2^{16}	$2^{16} - 1$	$16 \cdot N$	6	96	2^{-48}
2	2^{16}	$2^{16} - 1$	$16 \cdot N$	8	128	2^{-64}
1	2^8	$2^8 - 1$	$8 \cdot N$	14	112	2^{-56}

Case $N/q - 1$						
symbol size (byte)	q	N(symbols)	message length (bits)	v	MAC size (bits)	$\varepsilon \le$
4	2^{32}	$2^{16} \pm 1$	$\sim 2^{21}$	4	128	2^{-96}
2	2^{16}	$2^8 \pm 1$	$\sim 2^{12}$	6	96	2^{-72}
2	2^{16}	$2^8 \pm 1$	$\sim 2^{12}$	8	128	2^{-96}

where μ is the Möbius function and the sum is over all positive integers (i) that divide n. A well known good estimation on the number of such polynomials is $\frac{2^n}{n}$.

The first step of Construction 3 is shown below:

Construction 3

The algorithm for the generation of the authenticator tag of length z for a message m with length $\le N-z$ over $GF(2)$, when the ECC-code is BCH is the same as in Construction 1 except Step 1 that is as follows:

1) We choose a fresh MAC-generator polynomial $G_k(x)=f_k(x)g(x)$ with degree z, where component polynomial $f_k(x)$ with degree $\nu=\nu' n$, $\nu=z-(N-K)$ is defined by randomly choosing a set of number ν' different irreducible polynomial from the set of all irreducible polynomials over $GF(2)$ with degree n. □

Properties of Construction 3

***Claim* 5:** The BCH-construction is ε-balanced with

$$-\log_2(\varepsilon)\le(\nu'-1)n\,?\left[(\nu'-1)\log_2(n)+\log_2\left((\nu'-1)!\right)+1\right]$$

Example 3: Some typical evaluation results for Construction 3 are shown in Table 3.

***Proof of Claim* 5:** For a given message m and tag candidate u, the number of different polynomials $G_k(x)$ such that $G_k(x)$ divides polynomial $(m\|u)(x)$ is at most $\frac{K'}{\nu}$, where $K'=K-1$. Indeed, the maximum is attained if polynomial $(m\|u)(x)$ is a product of different n-degree irreducible polynomials over $GF(2)$. As we know the number of all possible keyed-polynomials $f_k(x)$ we get

$$\varepsilon\le\frac{\frac{K'}{\nu}}{\binom{I_2(n)}{\nu'}}$$

Using the approximation cited above on the number of irreducible polynomials over $GF(2)$ we can show that probability ε can be bounded by a quantity that decreases exponentially in parameter n. This means that by choosing the big field $GF(2^n)$ large enough we can set arbitrarily small probability ε. With approximation $I_2(n)\sim 2^n/n$ and straightforward algebra, we get the claim. □

***Claim* 6**: The above BCH-code based MAC-construction is e-error-forgery-resistant with probability ϵ.

Table 3. Evaluation examples for construction 3.

Q	N(bits)	message length (bits)	ν'	MAC size (bits)	$\varepsilon\le$
2^{16}	$2^{16}-1$	~2^{16}	6	96	2^{-52}
2^{24}	$2^{24}-1$	~2^{24}	6	144	2^{-98}

The proof logic of Claim 6 is similar to the proof Claim 2.

We can consider a construction (a parameter setting) better if probability ϵ is lower assuming equal sizes of MAC (in bits). In other words, we measure this quality with the ratio

$$-\log_2(\varepsilon)/\text{MAC size}$$

Using this measure according to the evaluation examples we get the following order between the constructions. The best is Construction 2 for settings of $N|(q-1)$ (0.75), the second is Construction 3 and Construction 1 (0.61 and 0.604, respectively), and the third is Construction 2 for $N=q-1$ (primitive length code). Equivalently, Construction 2 provides the largest effective (MAC) key size.

4. Conclusions

Our main goal was to show, that provably secure MAC with lightened computational complexity and adjustable parameters can be generated with concrete proofs of security.

The reader might ask: Why this great complication with proofs when we could simply take say half of a "strong" 128-bit AES-based CBC-MAC (and use it as an element in a lightweight authentication approach as several papers do)? Such truncation of the transmitted MAC is an ad-hoc practice, there are no guarantees just guesses and expectations (even if the initial, full-length MAC would be provably secure). In contrast to similar ad-hoc approaches in the field of Vehicular On-Board IT Systems we present formal proofs for security guarantees. We provide practical bounds on the applied security measure where the formulae depend on the parameters of the constructions. Naturally, the important security measure in the case of a MAC construction is the resistance to forgery attacks.

When the one-time stream is realized with AES-CTR mode keystream it remains even quantum secure when the key length of the underlying AES is doubled [18]. Post-quantum security may be a decisive advantage even in near future.

A significant portion of computations (one-time pads, MAC keys, MAC generator polynomials) can be carried out off-line by precomputations. The on-line computation complexity of MAC is essentially equivalent to a CRC computation over the field that the MAC is constructed. Generation and verification of the MAC can be started as the first message elements arrive, *i.e.* there is no need to wait for the full message to arrive.

Instead of error detection and automatic retransmissions, it is better to have an option of error correction capability. The error control code can run in error detection mode, and when an erroneous packet is detected there are two options, retransmission or error correction. The built-in option improves the latency properties.

We presented a set of versions of the construction with an adjustable set of

parameters as well as formulae for the evaluation of their security performance. This provides the advantage that in concrete applications with corresponding concrete values for size/security parameters the designer can find a matching construction from our set.

We presented a set of numerical evaluation examples (**Tables 1-3**) showing that our constructions meet the expectations with respect the dimensions and security for the intended field of application. An implementation can essentially use the well-established and computationally optimized tools error control technology, e.g. the complexity of GF(q) operations (multiplication, inversion) can be drastically decreased by using look-up tables (LUTs) [22] or light-weight Galois Field (GF) processor in case of HW-supported solutions [23].

Acknowledgements

The presented work was carried out within the SETIT Project (2018-1.2.1-NKP-2018-00004), which has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the 2018-1.2.1-NKP funding scheme.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] BOSCH (1991) CAN Specifications. Version 2.0, BOSCH, Gerlingen.

[2] Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., *et al.* (2010) Experimental Security Analysis of a Modern Automobile. *Proceedings of the* 2010 *IEEE Symposium on Security and Privacy*, Berkeley, 16-19 May 2010, 447-462. https://doi.org/10.1109/SP.2010.34

[3] Palanca, A., Evenchick, E., Maggi, F. and Zanero, S. (2017) A Stealth, Selective, Link-Layer Denial-of-Service Attack Against Automotive Networks. *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Bonn, 6-7 July 2017, 185-206. https://doi.org/10.1007/978-3-319-60876-1_9

[4] Radu, A.I. and Garcia, F.D. (2016) LeiA: A Lightweight Authentication Protocol for CAN. *Proceedings of the* 21*st European Symposium on Research in Computer Security*, Heraklion, 26-30 September 2016, 283-300. https://doi.org/10.1007/978-3-319-45741-3_15

[5] Wolf, M., Weimerskirch, A. and Paar, C. (2006) Secure In-Vehicle Communication, In: Lemke, K., Paar, C. and Wolf, M., Eds., *Embedded Security in Cars*, Springer Berlin, Heidelberg, 95-109. https://doi.org/10.1007/3-540-28428-1_6

[6] Nilsson, D.K. and Larson, U. (2009) A Defense-in-Depth Approach to Securing the Wireless Vehicle Infrastructure. *Journal of Networks*, **4**, 552-564.

[7] Glas, B., Guajardo, J., Hacioglu, H., Ihle, M., Wehefritz, K. and Yavuz, A. (2012) Signal-Based Automotive Communication Security and Its Interplay with Safety Requirements. *Proceedings of the Embedded Security in Cars Conference*, Berlin, 28-29 November 2012, 93-109.

[8] Groza, B., Murvay, S., van Herrewege, A. and Verbauwhede, I. (2012) LiBrA-CAN: A Lightweight Broadcast Authentication Protocol for Controller Area Networks. *Proceedings of the* 2012 *International Conference on Cryptology and Network Security*, Darmstadt, 12-14 December, 185-200. https://doi.org/10.1007/978-3-642-35404-5_15

[9] Zou, Q., *et al.* (2017) The Study of Secure CAN Communication for Automotive Applications. *SAE World Congress Experience* 2017, Cobo Center, 4-6 April 2017. https://doi.org/10.4271/2017-01-1658

[10] Mundhenk, P., Paverd, A., Mrowca, A., Steinhorst, S., Lukasiewycz, M., Fahmy, S.A., *et al.* (2017) Security in Automotive Networks: Lightweight Authentication and Authorization. *ACM Transactions on Design Automation of Electronic Systems*, **22**, 1-25. https://doi.org/10.1145/2960407

[11] Huynh Le, V., den Hartog, J. and Zannone, N. (2018) Security and Privacy for Innovative Automotive Applications: A Survey. *Computer Communications*. **132**, 17-41. https://doi.org/10.1016/j.comcom.2018.09.010

[12] Lin, C.W., Zhu, Q., Phung, C. and Sangiovanni-Vincentelli, A. (2013) Security-Aware Mapping for CAN-Based Real-Time Distributed Automotive Systems. *2013 IEEE/ACM International Conference on Computer-Aided Design*, San Jose, 18-21 Nov. 2013, 115-121. https://doi.org/10.1109/ICCAD.2013.6691106

[13] Han, G., Zeng, H.B., Li, Y.P. and Dou, W.H. (2014) SAFE: Security-Aware FlexRay Scheduling Engine. 2014 *Design, Automation & Test in Europe Conference & Exhibition*, Dresden, 24-28 March 2014, 1-4. https://doi.org/10.7873/DATE.2014.021

[14] Van Herrewege, A., Singelee, A.D. and Verbauwhede, I. (2011) CANAuth—A Simple, Backward Compatible Broadcast Authentication Protocol for CAN Bus. *Proceedings of the ECRYPT Workshop on Lightweight Cryptography*, Louvain-la-Neuve, 28-29 November 2011, 220-235.

[15] Wang, Q. and Sawhney, S. (2014) VeCure: A Practical Security Framework to Protect the CAN Bus of Vehicles. *Proceedings of the* 2014 *International Conference on the Internet of Things*, Cambridge, 6-8 October 2014, 13-18. https://doi.org/10.1109/IOT.2014.7030108

[16] Siddiqui, A.S., Gui, Y., Plusquellic, J. and Saqib, F. (2017) Secure Communication over CANBus. 2017 *IEEE* 60*th International Midwest Symposium on Circuits and Systems*, Boston, 6-9 August 2017, 1264-1267. https://doi.org/10.1109/MWSCAS.2017.8053160

[17] Escherich, R., Ledendecker, I., Schmal, C., Kuhls, B., Grothe, C. and Scharberth, F. (2009) SHE—Secure Hardware Extension Functional Specification. Hersteller-Initiative Software (HIS) AK Security.

[18] Bernstein, D.J. (2009) Cost Analysis of Hash Collisions: Will Quantum Computers Make SHARCS Obsolete? Manuscript. http://cr.yp.to/hash/collisioncost-20090823.pdf

[19] Zalman, R. and Mayer, A. (2014) A Secure but Still Safe and Low Cost Automotive Communication Technique. *Proceedings of the 51st Annual Design Automation Conference*, San Francisco, June, 2014, 1-5. https://doi.org/10.1145/2593069.2603850

[20] Krawczyk, H. (1994) LFSR-Based Hashing and Authentication. *Annual International Cryptology Conference*, Santa Barbara, 21-25 August 1994, 129-139. https://doi.org/10.1007/3-540-48658-5_15

[21] Lam, C.Y., Gong, G. and Vanstone, S. (2002) Message Authentication Code with Error Correction Capability. *International Conference on Information and Com-*

munications Security, Singapore, 9-12 December 2002, 354-366. https://doi.org/10.1007/3-540-36159-6_30

[22] Mahboob, A. and Ikram, N. (2005) Lookup Table Based Multiplication Technique for GF(2m) with Cryptographic Significance. *IEE Proceedings—Communications*, **152**, 965-974. http://dx.doi.org/10.1049/ip-com:20050022

[23] Chen, Y.J., Lu, S.S., Fu, C., Blaauw, D., Dreslinski, R. and Mudge, T. (2017) A Programmable Galois Field Processor for the Internet of Things. *Proceedings of the* 44*th Annual International Symposium on Computer Architecture*, Toronto, June 2017, 55-68. https://doi.org/10.1145/3079856.3080227