Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 1916 – 1929
© TÜBİTAK
doi:10.3906/elk-1311-54

A new security relation between information rate and state size of a keystream generator

Orhun KARA[1], **İmran ERGÜLER**[1], **Emin ANARIM**[2,*]
[1]TÜBİTAK - BİLGEM - UEKAE
National Research Institute of Electronics and Cryptology,
Gebze, Kocaeli, Turkey
[2]Department of Electrical-Electronics Engineering, Boğaziçi University, Bebek, İstanbul, Turkey

Received: 06.11.2013 • **Accepted/Published Online:** 01.07.2014 • **Final Version:** 23.03.2016

Abstract: Wireless communication in near field applications is becoming widespread. Most of the devices such as sensor networks or RFID applications are operated in constraint environments and some of these prevalent technologies require security applications. As one conclusion, the design and analysis of lightweight cryptographic algorithms has been one of the favorite research subjects over the last decade. We have seen that mostly lightweight block ciphers have been designed as symmetric encryption algorithms. The main reason is that stream ciphers are supposed to have large internal states due to the strict requirement related to their resistance against tradeoff attacks (time–memory–data tradeoff (TMDT)). In this work, we introduce a new stream cipher encryption mode by making use of error correcting codes, constituting a new tradeoff between information rate of the employed code and the internal state size of the keystream generator. This tradeoff enables us to decrease the state size without sacrificing the security against TMDT attacks.

The classical stream cipher encryption relies on deterministic keystream generation both at transmission and at receiver sides. On the other hand, we propose a noisy and nondeterministic keystream production, which we call the noisy keystream encryption (NKE). The receiver does not need the noise sequence to decrypt the ciphertext. However, it is a difficult problem for an attacker to recover the keystream sequence under the known plaintext scenario. We show that this gives a significant advantage in resisting attacks that require the keystream perfectly without any error. Particularly we prove that adding noise improves the security level in terms of internal state size against TMDT-type attacks.

Key words: Encryption, error-correcting codes, stream cipher, TMDT time–memory–data tradeoff attacks, keystream, internal state size

1. Introduction

Vernam's one time pad provides perfect secrecy in terms of ciphertext only attacks. That is, an attacker has no advantage in getting information about the plaintext for a given corresponding ciphertext as long as the key is used only once and produced randomly (having entropy more than that of the plaintext). However, due to its cumbersome key management, deterministic keystream generators are mostly in use today. Receivers require the keystreams used in the encryption processes so as to decrypt the received ciphertexts perfectly in both schemes. We propose a new keystream encryption model in this work. The receiver does not need to

[*] Correspondence: anarim@boun.edu.tr

Some part of this paper and its extended abstract were presented at The State of the Art of Stream Ciphers Workshop (SASC 2008) and at ICACM 2012, respectively.

produce the same noisy keystream as the transmitter produced perfectly to recover the plaintext in the new model. This model makes use of error correcting codes and the keystream is produced by adding some noise from a certain set, which we call noisy keystream encryption (NKE). The receiver can successfully recover the encoded plaintext from the noisy keystream even though the noise sequence is not known to him.

Note that this approach is prominently different from the conventional stream cipher model in which a deterministic keystream can easily be extracted from a known plaintext/ciphertext pair. On the other hand, the noisy deterministic keystream itself cannot be recovered by anyone who cannot produce the deterministic keystream. Therefore, the proposed model enhances security dramatically against error nontolerant attacks, which produce either wrong output or no output when an error occurred in their keystream data, particularly against time–memory–data tradeoff (TMDT) attacks.

TMDT attacks are used to invert one-way functions. One obvious way of inverting one instance is to exhaustively search all the possible inputs. The other straightforward way is to store all input–output pairs in a memory and find the inversion of an output by using the memory without any search. Hellman described how to invert a special type of one-way functions deduced from DES encryptions [1]. Inverting each instance relates to recovering a DES key in the chosen plaintext scenario, introducing a tradeoff between memory and time complexities. Then Babbage [2] and Golic [3] independently introduced the tradeoff attack between data and memory to recover one of the internal states for stream ciphers. Biryukov and Shamir improved the Hellman attack by making use of more data in the stream cipher encryption [4]. All these attacks impose a security criterion on stream ciphers: the state size must be not less than twice as large as the key size.

In fact, resistance to TMDT attacks has recently been considered an important security measure in stream cipher design. For a stream cipher, the ideal case in terms of hardware area cost (HAC) efficiency is that the internal state size is equal to the size of the key. Note, however, that it is impossible to attain the ideal case for the classical models due to their vulnerability to TMDT attacks. An appreciated countermeasure to this threat is to pick the internal state size of a keystream generator not less than twice as large as the size of the key initiating the internal state of the keystream. Although stream ciphers are historically accepted as low-cost confidentiality solutions for small hardware needs, this criterion would change this standing by imposing modern stream cipher designs to have larger internal states. Eventually, such an enlargement in internal states implies a significant increase in HAC. For instance, one of the GSM encryptors, A5/1, designed in the mid-80's, covers about 750 gate equivalents (GE) [5], but it is highly vulnerable to TMDT attacks, e.g., in [6] Biryukov et al. presented a TMDT attack on A5/1 such that the key is recovered in 1 s with only 2 min of the conversation on a PC. However, two modern low-cost stream ciphers designed under the TMDT security criteria, Grain [7], requires 1300 [8]. Meanwhile, on the block cipher side, recently proposed lightweight ciphers such as PRESENT [9], KTANTAN [10], and PRINTcipher [11] need about 1000 GE, 460 GE, and 400 GE, respectively.

Area constraint may negate some advantages of stream ciphers over block ciphers. Therefore, we consider this issue in the remaining part of the work and show that at a cost of increasing the communication overhead, it is possible to decrease the HAC threshold of stream ciphers without undermining the security level. Furthermore, we construct a tradeoff between HAC and information rate as the ratio between the internal state size and the key size tending the minimum value $\frac{2}{2-R}$, where R is the information rate. In addition, we show that it is possible to achieve the ideal case for the NKE model asymptotically. Thanks to this tradeoff, efficient design solutions can be provided for extremely resource constrained environments in disparate systems, e.g., wireless sensor networks and RFID tags. Indeed, in these systems, even addition of a small number of gates might cause a cipher model to be unusable, and this is the main reason why HAC reduction is crucial.

The organization of this paper is as follows. In the next section, we give a brief update on related work. Section 3 introduces the building blocks used in the proposed NKE model and analyzes its security improvement against error nontolerant attacks. After a short recall of TMDT attacks, in Section 4, we point out that HAC reduction for stream ciphers is achievable using the proposed model. Additionally, we show how the minimum initial vector (IV) size for the NKE model changes. We conclude this paper finally in Section 5.

2. Related work

Intentionally the use of random noise in a cryptographic primitive has been introduced previously in different studies. McEliece presented a public-key crypto scheme that uses linear block coding such that the encryption process involves multiplication of the message with a transformed generator matrix and the modulo 2 addition of a random error vector $\mathbf{e}$ [12]. The security mechanism of the system utilizes the NP-hard decoding problem of an arbitrary linear code. Moreover, in [13], Hopper and Blum introduced the human–computer protocol relying on the learning parity with noise (LPN) as a problem of decoding random linear codes. Furthermore, Aumasson et al. introduced the TCHo, which is a public key cryptosystem whose security is built on the difficulty of the problem of finding a multiple of a given polynomial with a very small number of coefficients [14]. For this system, the encryption procedure results in a noisy message and by only using the key, i.e. the sparse multiple of the feedback polynomial, the message part can be extracted from the noisy part. Furthermore, [15, 16, 17] have presented an approach for stream ciphers that basically embeds a random bit sequence into the ciphertext depending on the value of a pseudo-random sequence. Firstly, messages are encoded and then XOR'ed with a keystream sequence. Then a random vector is concatenated to the result, augmenting the ciphertext, and this value is multiplied with a secret permutation matrix $\mathbf{P}$, which is also known by the receiver. Last, a random noise sequence is summed with the result.

The idea of noisy keystream encryption was first presented by Kara and Erguler in ISCTurkey [18] and then in SASC [19]. The main motivation in adding noise to the keystream was enhancing the security of the existing keystream generator against certain attacks. Several models such as accumulation model, confusion model, or feedback model were introduced in this manner. Then the papers discuss and compare security enhancements of these models against some attacks such as guess and determine attacks, correlation attacks, free binary decision diagram attacks, and algebraic attacks besides the tradeoff attacks. As a concrete example, the improvement of the security of the GSM encryption scheme, A5/1, is assessed for each model separately against certain types of attacks mounted on A5/1 in the literature. The problem of improving tradeoff between information rate and the state size of a keystream generator is not studied in [18] or in [19].

An extended abstract related to tradeoff between information rate and state size was presented in [20]. This presentation contains only Theorem 1 without proof and the tradeoff between IV size and the information rate, which is given in Theorem 2, is missing. We give a complete set of statements including proofs and corollaries in this work.

In the next section, we describe the NKE model, which uses error correcting coding to significantly increase the security performance of the system against error nontolerant attacks.

3. The NKE model

3.1. Preliminaries

The encryption process of the proposed NKE model consists of three steps. Firstly, the plaintext sequence is chopped into the block of k-bits and the encryption algorithm proceeds by executing the following steps on individual blocks. The (n,k) linear block code encodes each block of plaintext block into n bit codeword. In the next step, the keystream generator outputs a keystream block with the same length of codeword as n-bits. Simultaneously, depending result of a true random number generator (TRNG), an n-bit error vector (noise block) is randomly picked from a predetermined error vector set. At the last step, the codeword, corresponding keystream bloc,k and the selected error vector are summed up to produce the ciphertext block. Nevertheless, the decryption procedure is split into two steps. In the first step, each ciphertext block is XOR'ed with respective keystream block and the result is decoded in the second step to capture the message block. In order to make the description of the model clear, the following notation is demonstrated in Table 1.

Table 1. Notation for the proposed NKE model.

(n,k)	(codeword, message) length
$E_{(n,k)}$	The encoder for the linear block code (n,k)
$D_{(n,k)}$	The decoder for the linear block code (n,k)
S_ϵ	The set of error patterns
t	Error correcting capability
$\mathbf{G}$	The generator matrix
$\mathbf{m}$	k-bit plaintext block, $\mathbf{m} = (m_1, \cdots, m_k)$
$\mathbf{v}$	n-bit codeword block, $\mathbf{v} = (v_1, v_2, \cdots, v_n)$
$\mathbf{z}$	n-bit keystream block, $\mathbf{z} = (z_1, \cdots, z_n)$
$\mathbf{e}$	n-bit noise block, $\mathbf{e} = (e_1, e_2, \cdots, e_n)$
$\hat{\mathbf{z}}$	n-bit noisy-keystream, $\hat{\mathbf{z}} = (\hat{z}_1, \cdots, \hat{z}_n)$
$\mathbf{c}$	n-bit ciphertext block, $\mathbf{c} = (c_1, c_2, \cdots, c_n)$
w	Hamming weight
$\oplus$	XOR operator

3.2. Encryption

The encryption algorithm of the NKE model works in the following order. In the first step, plaintext sequence M is split into k-bit blocks as $\mathbf{m_1}, \mathbf{m_2}, \cdots$. Then by using code encoding $E_{(n,k)}(\mathbf{m_i})$ takes message block $\mathbf{m_i}$ and outputs codeword $\mathbf{v_i}$. In the meantime, an error vector is chosen randomly from the set of error patterns as $S_\epsilon = \{\epsilon_1, \epsilon_2, \cdots, \epsilon_\Psi\} : \mathbf{e_i} \in S_\epsilon$ depending on output of the TRNG, where Ψ represents the number of error patterns. Then the codeword $\mathbf{v_i}$, corresponding keystream block $\mathbf{z_i}$, and the picked error vector are $\mathbf{e_i}$ XOR'ed to produce the ciphertext block $\mathbf{c_i}$ as depicted in the following equation:

$$\mathbf{c_i} = \mathbf{v_i} \oplus \mathbf{z_i} \oplus \mathbf{e_i}. \tag{1}$$

The enciphering process is expressed in matrix form as given below; it is also shown in Figure 1, where $\mathbf{g_0}, \mathbf{g_1}, \cdots, \mathbf{g_k}$ constitutes rows of the generator matrix $\mathbf{G}$ such that $\mathbf{v} = \mathbf{m} \cdot \mathbf{G}$.

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} \mathbf{g_1} \\ \mathbf{g_2} \\ \vdots \\ \mathbf{g_k} \end{bmatrix}^{\mathbf{T}} \cdot \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_k \end{bmatrix} \oplus \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \oplus \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}.$$

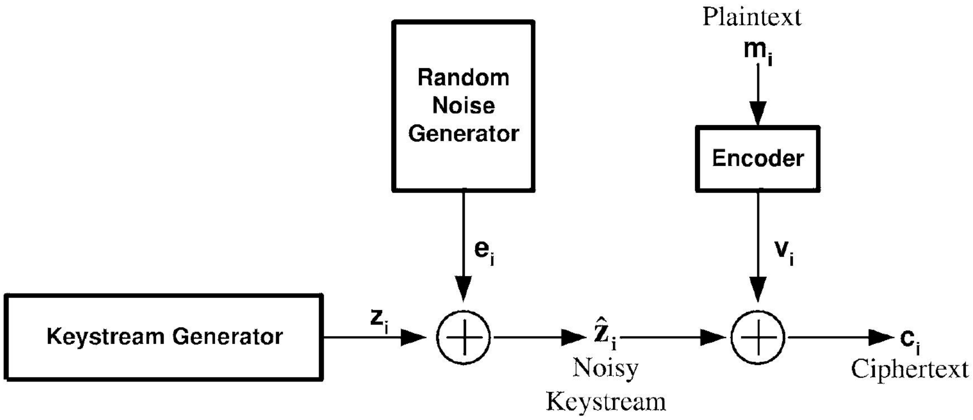

Figure 1. Noisy Keystream Encryption

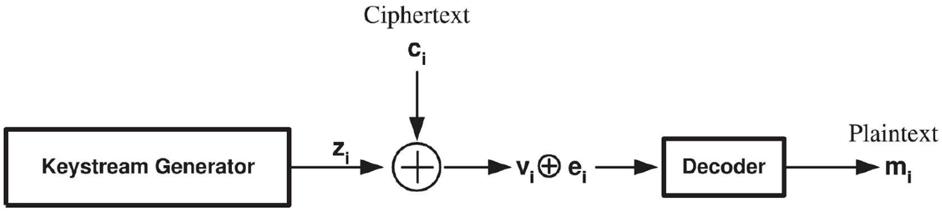

Figure 2. Noisy keystream decryption.

3.3. Decryption

In the decryption process of the NKE model, the ciphertext block $\mathbf{c_i}$ is firstly XOR'ed with the respective produced keystream block $\mathbf{z_i}$ and the noisy codeword is attained as $\mathbf{v_i} \oplus \mathbf{e_i}$. Since all transmitted error vectors can be corrected by the error correcting code, $D_{[n,k]}(\mathbf{v_i} \oplus \mathbf{e_i})$ results in the corresponding message block $\mathbf{m_i}$. This procedure is illustrated in Figure 2 and also expressed in the following equation:

$$\mathbf{m_i} = D_{[n,k]}(\mathbf{c_i} \oplus \mathbf{z_i}). \tag{2}$$

In a traditional noisy communication channel, correctness of the message is guaranteed by the linear block code if there are t or fewer errors in the received message, where t is the error correcting capability. In our case, however, all possible error patterns are known by both the sender and the receiver and a standard array should be constructed in order to maximize the number of error patterns as 2^{n-k}. In this standard array model, illustrated in Figure 3, the receiver perfectly captures the message as long as all elements of S_ϵ are assigned as coset leaders. Therefore, in the remaining parts of the study, we suppose that $|S_\epsilon| = 2^{n-k}$, i.e. 2^{n-k} different error patterns should be randomly selected in the error addition process of the sender.

3.4. Security of the NKE model for error nontolerant attacks

Before we elaborate on our security analysis, we will first state a set of reasonable definitions and assumptions. We define *an error nontolerant attack on a random one-way function as the attack that gives no output or a wrong and irrelevant output when an error occurs in its input. By irrelevant output, we mean an output that gives no advantages in obtaining the correct output.* Moreover, we suppose that the set of possible error patterns may not be kept secret and it is available to an attacker. Since error nontolerant attacks result in a wrong output even in the existence of a single bit error, a plausible method for an adversary may be realizing the following strategy. Assume that the attack requires V-bits of keystream sequence before adaptation of our technique. Notice that by means of our approach the number of candidates for an n-bit keystream block is 2^{n-k}, because one out of 2^{n-k} different error vectors has been added to this block. Therefore, there are now many possibilities for the V-bits of keystream sequence. The adversary should firstly compute all possibilities for this keystream sequence and then execute his classic attack approach. It is apparent that V indeed represents the consecutive keystream length to be used in each trial of the attack. Suppose τ denotes number of n-bit blocks in V-bits of keystream sequence as $\tau = \lfloor V/n \rfloor$. If $V \mod n \neq 0$, there may be also possibilities in the remaining $V \mod n$ bits. By assuming there are 2^l error vectors such that they differ in their first l bits for $l \leq n-k$, the total number of candidates for V-bits of keystream sequence can be expressed as

$$(2^{n-k})^{\tau} \cdot 2^{\min(l,n-k)}.$$

Let T denote time complexity of the classical attack before the NKE model is deployed, i.e. there is only one keystream sequence candidate to mount the attack. Note, however, that for the NKE model there are many possibilities for the same keystream sequence as $(2^{n-k})^{\tau} \cdot 2^{\min(l,n-k)}$. Hence, time complexity of the attack for our approach can be given as

$$\mathcal{T} = T \cdot (2^{n-k})^{\tau} \cdot 2^{\min(l,n-k)}. \tag{3}$$

In the next section, we review another type of error nontolerant attacks as TMDT attacks.

4. Hardware area cost reduction method

4.1. TMDT attacks

The main idea behind TMDT is distributing total attack complexity for a cipher into memory, time, and data complexities, so that an impractical attack may be applicable. This attack is more effective on stream ciphers compared to block ciphers, because precomputed tables are constructed related to keystream sequences rather than ciphertext sequences. In other words, the same tables can be reused for various ciphertext sequences produced by the same cipher.

The most interesting property of TMDT attacks is that they can be mounted on any cipher regardless of its internal working. Modern TMDT attacks are well known and considered in the design of stream ciphers for many years. In 1995, Babbage stated that as long as the state size of the stream cipher is less than twice the key size, the cipher is vulnerable to a time–memory tradeoff attack and its complexity will be less than $O(2^{\kappa})$, where κ is the length of the key [2]. It is equivalent to say that the internal state of a stream cipher should be more than twice the key length to resist such an attack. By conforming to this rule ensures[1] that the time–memory tradeoff attack would have complexity of at least $O(2^{\kappa})$. Nevertheless, such a security criterion forces cipher designs to have larger internal states, i.e. increase in HAC.

The idea of applying a time–memory tradeoff attack on stream ciphers was firstly addressed by Babbage [2] and Golic [3] independently, and hence known as the BG model. Following these studies, inspired by the original work of Hellman [1], Biryukov and Shamir introduced the use of multiple data to enhance the basic BG attack [4], named the BS model.

[1] It was demonstrated that the adoption of a state twice as large as the key is useless, unless IV size is as large as the key size (https://eprint.iacr.org/2005/090). The reason is obvious that to provide resistance against TMDT attacks, the internal state size of the cipher for any moment must be larger than twice the key size, i.e. it is ensured that the attack complexity is higher than exhaustive search. Hence, throughout this paper we suppose that this condition is satisfied by the related ciphers.

$$\begin{pmatrix} \mathbf{v_1} = 0 & \mathbf{v_2} & \mathbf{v_3} & \ldots & \mathbf{v_{2^k}} \\ \mathbf{e_2} & \mathbf{e_2} + \mathbf{v_2} & \mathbf{e_2} + \mathbf{v_3} & \ldots & \mathbf{e_2} + \mathbf{v_{2^k}} \\ \mathbf{e_3} & \mathbf{e_3} + \mathbf{v_2} & \mathbf{e_3} + \mathbf{v_3} & \ldots & \mathbf{e_3} + \mathbf{v_{2^k}} \\ \vdots & \vdots & \vdots & \vdots & \\ \mathbf{e_{2^{n-k}}} & \mathbf{e_{2^{n-k}}} + \mathbf{v_2} & \mathbf{e_{2^{n-k}}} + \mathbf{v_3} & \ldots & \mathbf{e_{2^{n-k}}} + \mathbf{v_{2^k}} \end{pmatrix}$$

Figure 3. NKE model standard array: The error vectors are assigned as coset leaders for an (n,k) linear code.

A TMDT attack is usually described by the following parameters: the search space N, time cost of precomputation stage P, the memory cost for storing tables M, time cost of the online phase T, and number of data samples used in the online phase D. According to the BG model, by using D different known keystreams of length $\log N$ and a precomputed table, which stores M randomly chosen possible internal states and their corresponding $\log N$ length outputs, it is likely that the internal state of the cipher is recovered. If $M = N/D$ is chosen, and $D \cdot M = N$ is satisfied, from the birthday paradox the probability of finding a match between the D keystreams and outputs of the table is high. Therefore, the tradeoff curve of the attack can be written as [2, 3]:

$$TM = N, T \leq D \text{ and } P = M.$$

For the BS model, in the precomputation stage t/D tables each of which contains m entries and encloses tm states are created, where $mt^2 = N$. The tables span only $tm \cdot \frac{t}{D} = N/D$ of all states. During the online stage, a match is searched between the data and the table entries. The tables include N/D states and D data samples are searched; hence it is likely that a match is obtained. The costs of the attack can be calculated as follows. There are t/D tables and size of each table is m. Therefore, the memory cost of the attack is $M = mt/D$. In addition, D data samples are searched within t/D tables. Since each trial costs time in t, online time complexity of the attack becomes $T = t \cdot t/D \cdot D = t^2$. In the light of these values, the BS model curve can be expressed as [4]:

$$M^2TD^2 = N^2, \text{with } P = N/D \text{ and } D^2 \leq T.$$

4.2. The lower bound for the state size of the NKE model

Since a TMDT attack is a type of error nontolerant attack, considering Eq. 3 one can realize that time complexity of the TMDT attack increases for the NKE model. As a consequence, minimum internal size requirement against TMDT attacks becomes less than 2κ and in the following theorem we give the minimum required state sizes for both BG and BS models.

Theorem 1 *For an (n,k) linear block code, $R = k/n$ is known as the information rate. Let κ be the length of the secret key. Then, according to BG and BS models, the lower bound for the state size of the NKE model is $\kappa \cdot (1 - \frac{R}{2})^{-1}$ and it is less than 2κ.*

Proof Let $K = 2^{\kappa}$ and $\mathcal{C}$ represent the attack complexity, i.e. $\mathcal{C} = \max(M, \mathcal{T}, D)$ and η represent state size of the cipher[2], and so $N = 2^{\eta}$. The constraint regarding the key size is $\mathcal{C} \geq K$. Below, we will derive the lower bounds of the state size for the BG and BS models successively.

[2] In general, complexity of the TMDT attack is determined by the largest of M, T and D. Moreover, it is usually assumed that the attacker has an unlimited amount of time in the precomputation phase. Thus, P is usually not considered in cost evaluation of the attack.

- For the BG model it is assumed that $DM = N$ in the case of $T = D$. The equation (3) can be also expressed as $\mathcal{T} = D \cdot (2^{n-k})^{\tau} \cdot 2^{\min(l,n-k)}$. Because $\tau = \lfloor logN/n \rfloor$, we can write

$$\mathcal{T} \approx D \cdot (2^{n-k})^{logN/n} = D \cdot N^{1-R}. \tag{4}$$

  It can be seen that $\mathcal{T} > D$, and so $\mathcal{C}^{BG} = \max(M, \mathcal{T})$. Let $M = N^{\alpha}$; then $D = N^{1-\alpha}$ is obtained and from (4) we calculate $\mathcal{T} = N^{2-\alpha-R}$. Hence, $\mathcal{C}^{BG} = \max(N^{\alpha}, N^{2-\alpha-R})$. The lower bound for the state size is determined by the minimum value of $\mathcal{C}^{BG}$. The minimum of $\mathcal{C}^{BG}$, denoted by $\mathcal{C}^{BG}_{min}$, is defined as

$$\mathcal{C}^{BG}_{min} \triangleq \min_{\alpha \in [0,1]} \mathcal{C}^{BG}(\alpha).$$

  One can obtain $\mathcal{C}^{BG}_{min}$ when $\alpha = 2 - \alpha - R$, i.e. $\alpha = 1 - \frac{R}{2}$ as $\mathcal{C}^{BG}_{min} = N^{1-\frac{R}{2}}$.

- The tradeoff curve of the BS model is $M^2TD^2 = N^2$ for $T \geq D^2$. Let $M = N^{\alpha}$, $D = N^{\beta}$; then $T = N^{2-2\alpha-2\beta}$. Because $\tau = \lfloor logN/n \rfloor$, we can rewrite (3) for the BS model as

$$\mathcal{T} \approx T \cdot (2^{n-k})^{logN/n} = N^{3-2\alpha-2\beta-R}. \tag{5}$$

  $\mathcal{C}^{BS} = \max(N^{\alpha}, N^{3-2\alpha-2\beta-R})$, since $T \geq D^2$. For the BS model $\mathcal{C}^{BS}_{min}$ is defined as

$$\mathcal{C}^{BS}_{min} \triangleq \min_{\alpha \in [0,1], \beta \in [0, \frac{1-\alpha}{2}]} \mathcal{C}^{BS}(\alpha, \beta).$$

  $\mathcal{C}^{BS}_{min}$ occurs for $\beta_{\max} = \frac{1-\alpha}{2}$ and $\alpha = 3 - 2\alpha - 2\beta - R$. For these chosen values $\alpha = 1 - \frac{R}{2}$ and $\mathcal{C}^{BS}_{min} = N^{1-\frac{R}{2}}$ are obtained.

Notice that $\mathcal{C}^{BG}_{min} = \mathcal{C}^{BS}_{min}$. Therefore, the bound of the state size for both the BG and BS models can be attained in the following steps:

$$\begin{aligned} &\mathcal{C}_{min} \geq K, \\ &N^{1-\frac{R}{2}} \geq K, \\ &2^{\eta \cdot (1-\frac{R}{2})} \geq 2^{\kappa}, \\ &\eta \geq \kappa \cdot (1 - \tfrac{R}{2})^{-1}. \end{aligned}$$

Thus, according to the BG and BS models, the minimum state size, η_{min}, is given as

$$\eta_{min} = \kappa \cdot (1 - \frac{R}{2})^{-1}. \tag{6}$$

Note that the minimum state size is less than 2κ, since R is always less than 1. In other words, $\kappa \cdot (1 - \frac{R}{2})^{-1} < 2\kappa$. □

Corollary 1 $\lim_{R \to 0} \eta_{min} = \kappa$.

Remark 1 *From Corollary 1 the lower bound for η_{min} is obtained when $R = 0$ and this is the ideal case, since the internal state size cannot be less than the key size.*

Remark 2 *When $R = 1$, i.e. there is no encoding, so η_{min} tends to the classical bound for the state size as 2κ.*

4.3. The minimum IV size for the NKE model

The adoption of a state twice the key size is useless, unless the internal state size is larger than the sum of the key and IV sizes (see https://eprint.iacr.org/2005/090). Additionally, size of IV has to be as large as the key. In this approach, the attacker tries to invert the function f as $f : \{keys\} \times \{IVs\} \rightarrow$keystream prefix that inputs key and IV pair and outputs $(\kappa + \upsilon)$-bit keystream prefix, where κ and υ denote the lengths of the key and IV, respectively. The attacker can benefit from multiple data samples produced by $f()$ with the same key but different IVs. Hence, a TMDT attack can be still mounted with BS model tradeoff curve $M^2TD^2 = N^2$, while N now spans all possible key and IV combinations as $N = 2^{\kappa+\upsilon}$. Notice that, similar to internal state size, for the NKE model the required IV size is changed and below we show that it is less than κ.

Theorem 2 *Let υ and κ be the size of the secret key and size of the initial vector IV. Then for the NKE model, the minimum IV size is $\kappa \cdot \frac{R}{2-R}$ and it is less than κ.*

Proof The proof is similar to the proof of Theorem 1. Since $N = 2^{\kappa+\upsilon}$ for this case, we replace η_{min} with $\kappa + \upsilon_{min}$ in (6) and obtain

$$\upsilon_{min} = \kappa \cdot \frac{R}{2-R}. \tag{7}$$

Note that the minimum IV size is less than κ, because R is always less than 1. In other words, $\upsilon_{min} < \kappa$. □

Corollary 2 $\lim_{R \to 0} \upsilon_{min} = 0$.

Remark 3 *From Corollary 2 the minimum required IV size is obtained when $R = 0$ and this is the ideal case, since no IV is used in the system.*

Remark 4 *When $R = 1$, i.e. there is no encoding, so υ_{min} tends to the classical bound for the state size as κ.*

4.4. Hardware complexity of the NKE

Noisy encryption may lead to additional hardware cost due to noise generation, storing error patterns, encoding, and decoding. However, the number of used flip-flops in the scheme, which is directly related to the state size, becomes mostly the dominant element. Hence, by Theorem 1, one can find a relation between HAC and the minimum state size given in (6).

In the crypto community, it is well known that the randomization of IV being used in stream ciphers is usually achieved with the help of a TRNG. In general, crypto devices such as smart cards or RFID tags contain TRNGs in their hardware block. Even if a hardware block does not encapsulate a TRNG as default, it is not costly to implement a new TRNG. For instance, the silicon area of a TRNG is of roughly 0.04 mm^2 whereas it is around 0.1 mm^2 for AES in a smart card micro controller implementation given in [21]. Throughput of a TRNG is also not an issue: new generation TRNG designs can run at a magnitude of hundreds of bits per second [22]. However, since a TRNG block is involved by default in many stream cipher based crypto systems, we do not consider the hardware cost of the TRNG in HAC calculation of the NKE model even though it has a negligible cost.

Another additional cost may arise from encoding and decoding. This cost calculation incorporates three components: (i) a lookup table for the error pattern set used in both the encoding and decoding processes, (ii) the computation of the parity bits, and (iii) the syndrome computation as realized to find the corresponding error pattern through the decoding process.

The lookup table is of size $(n-k)\times n$. Hence, it is supposed to occupy very low hardware area cost for small values of n and $n-k$, namely $2^{n-k}\cdot n$ bit ROM (read only memory). On the other hand, computing the parity bits or the syndrome bits consists of few XOR operations in general.

To give an idea we consider the encoding, the decoding, and the error pattern storage hardware costs of a $(10,6)$ code whose generator and parity check matrices are illustrated in Figure 4 as an example. Observe that calculating the parity bits in the encoding process requires 9 XORs whereas the computation of syndrome bits costs 13 XORs. On the other hand, the table that includes the error patterns occupies 4×10 bit ROM. To evaluate its storage cost, consider some similar structures such as: the 3×3 S-box of PRINTcipher costs 12 GE [11], the 4×4 S-Box of PRESENT costs between 22 GE and 28 GE [9, 23, 24, 25], and the 8×8 AES S-Box costs around 696 GE [26]. Hence it is expected that the set of the error patterns of the $(10,6)$ code costs around 35–55 GE. In [27], the analysis of the hardware implementations of different streams ciphers is given. According to this analysis, 1 GE is used for equivalent area that a two input NAND gate costs (6 transistors), a two input XOR gate typically occupies 2.33 GE (14 transistors), and a flip-flop requires 8 GE. Therefore, the encoding block of the NKE is estimated as about 56–76 (parity bit computation + lookup table) GE, while this is about 65–85 GE (syndrome computation + lookup table) for the decoding block. Indeed for a full duplex communication instead of occupying the lookup table twice, a single lookup table can be used in both encoding and decoding facilities. In this case, the total cost of the decoding and encoding will be about 91–111 GE (23 XORs + lookup table + a 2-to-1 MUX). These values are estimations. In fact, we observe similar numbers in the implementation results with FPGA Xilinx Spartan III. The synthesis estimates ten 4-input lookup tables for the error pattern set and 23 XORs for calculating the parity bits and the syndrome. In [27], it is assumed that one lookup table is about 6 GE. Thus, the total cost of the decoding and encoding in FPGA implementation will be about 116 GE. Let us remark that we can decrease the hardware area of the registers from 1280 GE to 915 GE by using the $(10,6)$ code by Theorem 1. Hence we gain 365 GE, which is higher than the encoding and/or the decoding costs.

Note that in low-cost devices, some data (e.g. key, ID) can be also stored in rewritable memory (EEPROM) or in ROM of the gadget like in ultralightweight RFID designs [28, 29, 30]. Since the error pattern set is static and not secret information it can be stored in ROM of the device and its cost can be excluded from the total cost. In any case, it does not take up too much hardware area even if it is deployed in hardware as discussed previously.

Table 2 illustrates the HAC values for the minimum internal state sizes of the NKE model by using some linear block codes. In order to compare the HAC for different designs, we give the gate equivalents of each example. In [27], hardware implementation analysis of different stream ciphers is given. The internal state size values are obtained for $\kappa = 80$ bits. Notice that for a classical stream cipher scenario the HAC will be about 1280 GE for only the internal state, whereas the HAC for the ideal case cipher design is 640 GE. To express reduction in state size we introduce the state size ratio—the minimum ratio of the internal state size over the key size—as $\eta_{ratio} = (1-\frac{R}{2})^{-1}\cdot\kappa)/\kappa = \frac{2}{2-R}$. Note that for $R=1$, which is the case of the classical models, we obtain $\eta_{ratio}=2$. As can be seen from Table 2 there is a tradeoff between η_{ratio} and communication overhead. That is, a reduction in the state size obviously requires a decrease in the information rate R, which results in

$$\mathbf{G} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Figure 4. The generator and parity check matrices of a $(10, 6)$ linear block code.

an increase in the communication overhead.

Table 2. Required HAC of the proposed NKE model and η_{ratio} for different linear block code types having ideally distributed error pattern sets.

Block Code	R	HAC	η_{ratio}
[12,8] Code	0.6667	960 GE	1.5
[10,6] Code	0.6	915 GE	1.43
[24,12] Code	0.5	853 GE	1.33
[16,5] Code	0.3125	758 GE	1.18
[31,6] Code	0.1935	704 GE	1.1
[511,19] Code	0.0372	652 GE	1.018

By considering Table 2, one of the principal questions for the stream cipher design philosophy can be raised: what is the minimum value of η_{ratio} for a secure stream cipher? We know that this ratio cannot be less than 1 due to the resistance against exhaustive search attacks. On the other hand, it cannot be less than 2 for the conventional keystream encryption, a criterion imposed by the TMDT attacks. We prove that the η_{ratio} tends to 1 securely, in spite of the TMDT attacks, as the information rate R gets closer to 0 in the proposed NKE model. However, the existence of a secure keystream encryption model for which the ratio is exactly 1 has remained an open question. We conjecture that the ratio is strictly greater than 1 for any model.

The cost of encoding and decoding is another issue that increases the HAC. It is clear that both the encoding and the decoding increase costs when the information rate decreases. For a small block length ($n < 16$) and for a moderate information rate ($R > 0.5$), we expect the HAC of the encoding and decoding to be negligible in comparison with the gain obtained from the decrease of the internal state size. Another remark that can be deduced from Theorem 1 is that we do not gain so much when R is too small. This can be seen from Table 2 also. For instance, we need roughly 960 GE for $R = 0.6667$ and 652 GE for $R = 0.0372$ for 80-bit security. A roughly 18 times decrease in the information rate will gain only 308 GE. However, we need 1280 GE when $R = 1$. Hence, we already obtain 320 GE gain when $R = 0.6667$. That is, the tradeoff is more beneficial when the information rate is not so small.

We consider the throughput of encoding and decoding of plaintext blocks as a subordinate issue since we concentrate on the lightweight applications of NKE and hence the most crucial efficiency parameters are the area and the power consumption. Nevertheless, it is worth to state that encoding does not generally hinder the throughput of a noisy keystream encryption. Encoding and keystream generation can be implemented in parallel. While an encoded plaintext block is being encrypted, the next plaintext block can be encoded simultaneously. Since the encoding procedure consists of a few XOR operations for each parity bit, it is not supposed to be the critical path in comparison with generating a sufficient number of keystream bits to encrypt one block of plaintext. On the other hand, stream cipher encryption is usually bit by bit or character by character encryption and hence we do not need to deploy several parallel implementations of encoding. Concisely, the latency of keystream generation is expected to be the latency of a noisy keystream encryption.

5. Discussion and conclusion

There was a common consensus in the early 90s that stream ciphers covered less area than block ciphers in general. However, the TMDT attacks on stream ciphers have reversed this opinion. Indeed, it has been seen that the most recent lightweight ciphers designed in the last decade are block ciphers such as PRESENT, KATAN, and PRINTcipher. On the other hand, it is surprisingly interesting to observe that the internal state sizes of blockciphers are not taken into consideration as a security criterion in the design philosophy as resistance to TMDT attacks and this is one of the main reasons why designers focus on block ciphers rather than stream ciphers for hardware constraint applications.

In this study, we proposed to use error correcting codes for a new stream cipher model—the NKE model—in which the keystream is generated by combining some deterministic and nondeterministic sequences. We showed that this model provides security enhancement against error nontolerant attacks and further analyzed the security of the model for TMDT attacks to show how a reduction in the minimum state size of a cipher can be achieved. We also noted that such a gain in hardware cost can make the NKE encryption system suitable for hardware constrained systems. Moreover, we linked HAC with the minimum state size of the cipher and give a tradeoff between HAC and information rate of the used block code.

References

[1] Hellman ME. A cryptanalytic time-memory trade-off. IEEE T Inform Theory 1980; 4: 401-406.

[2] Babbage S. A space/time tradeoff in exhaustive search attacks on stream ciphers. In: Proceedings of European Convention on Security and Detection; 16–18 May 1995; Brighton, England. London, England: IEE No:408. pp. 161-166.

[3] Golic J. Cryptanalysis of alleged A5 stream cipher. In: Proceedings of the 16th Annual International Conference on Theory and Application of Cryptographic Techniques – EUROCRYPT'97; 11–15 May 1997; Konstanz, Germany. New York, NY, USA: Springer-Verlag LNCS 1233. pp 239-255.

[4] Biryukov A, Shamir A. Cryptanalytic time/memory/data tradeoffs for stream ciphers. In: Proceedings of Advances in Cryptology – ASIACRYPT 2000; 3–7 December 2000; Kyoto, Japan. New York, NY, USA: Springer-Verlag LNCS 1976. pp 1-13.

[5] Batina L, Lano J, Mentens N, Ors SB, Preneel B, Verbauwhede I. Energy, Performance, Area Versus Security Trade-offs for Stream Ciphers. In: Proceedings of the State of the Art of Stream Ciphers SASC-2004, ECRYPT Workshop Record. pp. 302–310.

[6] Biryukov A, Shamir A, Wagner D. Real time cryptanalysis of A5/1 on a PC. In: Proceedings of the 7th International Workshop on Fast Software Encryption – FSE 2000. New York, NY, USA: Springer-Verlag LNCS 1978. pp. 1-18.

[7] Hell M, Johansson T, Meier W. Grain: a stream cipher for constrained environments. International Journal Of Wireless And Mobile Computing 2007; 2: 86-93.

[8] Good T, Benaissa M. Hardware results for selected stream cipher candidates. In: Proceedings of the State of the Art of Stream Ciphers, ECRYPT Workshop Record – SASC-2007; 31 January–1 February 2007; Bochum, Germany. pp. 191-204.

[9] Bogdanov A, Knudsen LR, Leander G, Paar C, Poschmann A, Robshaw MJ, Seurin Y, Vikkelsoe C. PRESENT: an ultra-lightweight block cipher. In: Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems – CHES'07. New York, NY, USA: Springer-Verlag LNCS 4727. pp 450-466.

[10] De Cannière C, Dunkelman O, Knežević M. KATAN and KTANTAN – A family of small and efficient hardware-oriented block ciphers. In: Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems – CHES 2009. New York, NY, USA: Springer-Verlag LNCS 5747. pp 272-288.

[11] Knudsen LR, Leander G, Poschmann A, Robshaw MJB. PRINTcipher: a block cipher for IC-printing. In: Proceedings of International Conference on Cryptographic Hardware and Embedded Systems – CHES 2010. New York, NY, USA: Springer-Verlag LNCS 6225. pp 16-32.

[12] McEliece RJ. A public key cryptosystem based on algebraic coding theory. DSN Progress Report 1978; 44: 114-116.

[13] Hopper NJ, Blum M. Secure human identification protocols. In: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology – ASIACRYPT 2001. New York, NY, USA: Springer-Verlag LNCS 2248. pp 52-66.

[14] Aumasson JP, Finiasz M, Meier W, Vaudenay S. TCHo: a hardware-oriented trapdoor cipher. In: Proceedings of the 12th Australasian Conference on Information Security and Privacy – ACISP'07. New York, NY, USA: Springer-Verlag LNCS 4586. pp 184-199.

[15] Mihaljevic MJ, Imai H. An approach for stream ciphers design based on joint computing over random and secret data. Computing 2009; 85: 153-168.

[16] Mihaljevic MJ, Imai H. A stream cipher design based on embedding of random bits. In: Proceedings of International Symposium on Information Theory and its Applications – ISITA2008; 7–10 December 2008, Auckland, New Zealand. pp. 1492-1502.

[17] Mihaljevic MJ, Imai H. A stream ciphering approach based on the wire-tap channel coding. In: Proceedings of 8th Central European Conference on Cryptography; 2–4 July 2008, Graz, Austria.

[18] Kara O, Erguler I. A new approach to keystream based cryptosystems. In: Proceedings of Information Security and Cryptology Conference ISCTurkey – 2007; Ankara, Turkey. pp. 163-170.

[19] Kara O, Erguler I. A new approach to keystream based cryptosystems. In: Proceedings of the State of the Art of Stream Ciphers. ECRYPT Workshop Record – SASC 2008; Lausanne, Switzerland. pp. 205-221.

[20] Kara O, Erguler I, Anarim E. A stream cipher model for hardware constraint environments. In: Proceedings of Extended Abstracts, International Conference on Applied and Computational Mathematics ICACM–2012, Ankara, Turkey: METU.

[21] Trichina E, Bucci M, Seta DD, Luzzi R. Supplemental cryptographic hardware for smart cards. IEEE Micro 2001; 6: 26-35.

[22] Tavas V, Demirkol AS, Özoguz S, Zeki A, Toker A. Integrated cross-coupled chaos oscillator applied to random number generation. IET Circuits, Devices & Systems 2009; 1: 1-11.

[23] Wu W, Zhang L. LBlock: a lightweight block cipher. In: Proceedings of Applied Cryptography and Network Security – ACNS 2011. New York, NY, USA: Springer-Verlag LNCS 6715. pp 327-344.

[24] Shibutani K, Isobe T, Hiwatari H, Mitsuda A, Akishita T, Shirai T. Piccolo: an ultra-lightweight blockcipher. In: Proceedings of the 13th International Conference on Cryptographic Hardware and Embedded Systems – CHES 2011. New York, NY, USA: Springer-Verlag LNCS 6917. pp 342-357.

[25] Guo J, Peyrin T, Poschmann A, Robshaw MJB. The LED block cipher. In: Proceedings of the 13th International Conference on Cryptographic Hardware and Embedded Systems – CHES 2001. New York, NY, USA: Springer-Verlag LNCS 6917. pp 326-341.

[26] Satoh A, Morioka S, Takano K, Munetoh S. A compact Rijndael hardware architecture with S-box optimization. In: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology – ASIACRYPT 2001. New York, NY, USA: Springer-Verlag LNCS 2248. pp 239-254.

[27] Good T, Chelton W, Benaissa M. Review of stream cipher candidates from a low resource hardware perspective. In: Proceedings of the State of the Art of Stream Ciphers, ECRYPT Workshop Record – SASC 2006; 2–3 February 2006; Leuven, Belgium.

[28] Chien HY. SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. IEEE T Depend Secure 2007; 4: 337-340.

[29] Peris-Lopez P, Castro JCH, Estévez-Tapiador JM, Ribagorda A. EMAP: an efficient mutual-authentication protocol for low-cost RFID tags. In: Proceedings of OTM Federated Conferences and Workshop; 29 October–3 November 2006; Montpelier, France. IS Workshop. pp. 352-361.

[30] Peris-Lopez P, Castro JCH, Estévez-Tapiador JM, Ribagorda A. M^2AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In: Proceedings of International Conference on Ubiquitous Intelligence and Computing UIC'06. New York, NY, USA: Springer-Verlag LNCS 4159. pp 912-923.