Miodrag J. Mihaljević *

ON CERTAIN APPROACHES FOR ANALYSIS AND DESIGN OF CRYPTOGRAPHIC TECHNIQUES FOR SYMMETRIC ENCRYPTION AND KEY MANAGEMENT

Abstract. This chapter yields a review of certain mathematical approaches for analysis and design of the basic cryptographic elements for establishing information security in information-communication systems. The following two topics are addressed: selected issues on stream ciphers for encryption and key management based on broadcast encryption. Certain coding related issues for security evaluation and design of stream ciphers are considered. The discussed security evaluation techniques corresponding to decoding approaches include one-step and iterative decoding paradigms, and the addressed design issues involve homophonic coding for joint employment of pseudorandomness and randomness. Elements for cryptographic security evaluation and advanced design of the key managements based on broadcast encryption are pointed out.

Mathematics Subject Classification (2010): Primary: 94-02, 94A60; Secondary 11T71.

Keywords: cryptology, cryptography, cryptanalysis, pseudorandomness, randomness, coding theory.

**Mathematical Institute, Serbian Academy of Sciences and Arts, Kneza Mihaila 36, Belgrade*

Contents

1. Introduction

Establishing data security in information-communication systems or more generally, establishing cyber-security is one of the most important issues in order to avoid that information-communication technologies become misused with potentially catastrophic impacts. Cryptology is a mathematical discipline which provides basic methods and techniques for establishing elements of mechanisms for information and communications security. Employing cryptology we can develop a large number of different elements for achieving the security goals including the following main ons: (i) secrecy and privacy; (ii) integrity control; (iii) authenticity control (including the non-repudiation). For achieving the previous goals, cryptology deals with design and security evaluation of certain cryptographic primitives. The cryptographic primitives are mathematical algorithms which mainly (but not always) involve certain secret data for achieving the addressed goals. These secret data are called cryptographic keys or keys (for short). Among the main cryptographic primitives are the ones for encryption and key management. Cryptographic primitives for encryption are basic elements for the secrecy protection, and the ones for key management are main elements of the necessary "infrastructure" management the secret data employed for encryption and in a number of other cryptographic primitives. It is out of the scope of this chapter to serve as an introduction to cryptology and regarding this issue an interested reader is advised to check some of the related text books like [98] (which is available at http://cacr.uwaterloo.ca/hac/).

This chapter is devoted to the following two cryptographic primitives: stream ciphers for encryption and key management based on broadcast encryption. Selection of the addressed topics and contents of this chapter originate from the results reported in [1]–[62].

Encryption Based on Stream Ciphers. Stream ciphers play an important role in information security and they are a well recognized topic within cryptology. A stream cipher encrypts one individual character of a plaintext message at a time, using an encryption transformation which varies with time. Such a cipher is typically implemented by the use of a pseudorandom number generator or a keystream generator which expands a short secret key into a long running key sequence. A keystream generator is equivalent to a finite state machine that, based on a secret key, generates a keystream for controlling an encryption transformation. Design of highly efficient and secure stream ciphers is still an important challenge.

This chapter addresses certain coding related issues for security evaluation and design of stream ciphers. The discussed security evaluation techniques corresponding to decoding approaches include one-step and iterative decoding paradigms.

Key Management Based on Broadcast Encryption. In order to perform symmetric encryption/decryption the secret session key should be shared between the encryption and decryption entities. Broadcast encryption is a technique for distribution, via a public communication channel, secret session keys employing the pre-shared secret keys which provide that only selected parties can learn the secret session key. This chapter provides elements for cryptographic security evaluation and advanced design of the key managements based on broadcast encryption.

2. Decoding Based Approach for Security Evaluation of Certain Stream Ciphers

A number of the published keystream generators are based on binary linear feedback shift registers (LFSRs) assuming that parts of the secret key are used to load the LFSRs initial states (see [98], for example).

Note that a binary LFSR generate recurrence sequences over GF(2), and under certain assumption these sequences have the maximum possible period (for the given recurrence order) and good properties of pseudorandomnes.

The unpredictability request, which is one of the main cryptographic requests, implies that the linearity inherent in LFSRs should not be "visible" in the generator output. One general technique for destroying the linearity is to use several LFSRs which run in parallel, and to generate the keystream as a nonlinear function of the outputs of the component LFSRs. Particularly, suitable Boolean functions can be employed for realization of the nonlinear mapping. Such keystream generators are called nonlinear combination generators (see [98], for example).

Accordingly, an output sequence from nonlinear combination generator can be considered as follows:

$$y_i = f(x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(m)}), i = 1, 2, \ldots,$$

and assuming the binary case,

$$f(\cdot) : \{0,1\}^m \to \{0,1\},$$

$$x_i^{(j)} = \bigoplus_{l=1}^{L} \alpha_l^{(j)} \cdot x_{j-l}^{(j)}, \;\; \alpha_l^{(j)} \in \{0,1\}, \; l = 1, 2, \ldots, L, \; j = 1, 2, \ldots, m,$$

assuming that $\{x_i^{(j)}\}_i$, $j = 1, 2, \ldots, m$, are binary sequences.

This section yields a brief overview of a decoding based approach for security evaluation of the combination keystream generators.

Fast Correlation Attack. A central weakness of a nonlinear combination keystream generator has been demonstrated in [106]. Assuming certain nonlinear functions it is shown in [106] that it is possible to reconstruct independently initial states of the LFSRs, i.e. parts of the secret key (and accordingly the whole secret key as well) based on the correlation between the keystream generator output and the output of each of the LFSRs. The reported approach is based on exhaustive search through all possible nonzero initial states of each LFSR. A substantial improvement of the previous approach which yields nonexponential complexity with the LFSR length has been proposed in [97]. This approach is called fast correlation attack (FCA), and its extensions and refinements, as well as its analysis are presented in a number of papers including [29], [26], [73], [28] and [6].

The basic ideas of all reported FCAs include the following two main steps: (i) Transform the cryptographic problem into a suitable decoding one; (ii) Apply (devise) an appropriate decoding algorithm.

In the following, correlation means that the mod-2 sum of the LFSR output and the generator output can be considered as a realization of a binary random variable

taking values 0 and 1 with probabilities $1-p$ and p, respectively, with $p < 0.5$ (or $p \neq 0.5$). Consequently, the problem of the LFSR initial state reconstruction based on the keystream generator output sequence can be considered as the decoding problem of a punctured simplex code (defined by the feedback connections of the LFSR) after transmission over a binary symmetric channel (BSC) with crossover probability p uniquely determined by the correlation. More precisely, the fast correlation attack on a particular LFSR in a nonlinear combining generator given the segment of the generator output can be considered as follows: (i) The N-bit segment of the output sequence from the length-L LSFR is a codeword of an (N, L) punctured simplex code; (ii) The corresponding N-bit segment of the nonlinear combination generator output is the corresponding noisy codeword obtained through a BSC with crossover probability p; (iii) The problem of the LFSR initial state reconstruction, assuming its characteristic polynomial is known, is the problem of decoding the (n, k) punctured simplex code transmitted over a BSC with crossover probability p.

Two main classes of the reported FCAs are one-step decoding and iterative decoding based fast correlation attacks.

FCAs based in One-Step Decoding. Powerful approaches for FCAs realization based on one-step decoding have been reported in [29], [26], and further developed in a number of references including [73]. These techniques are based on a threshold decoding for reconstruction of all information bits under a hypotheses of certain B bits in conjunction with exhaustive search over all 2^B possibilities. The analysis of these algorithms include the results reported in [63] implying the high efficiency assuming an enough long sample for cryptanalysis.

FCAs based on Iterative Decoding. Certain approaches for FCAs based on iterative decoding which have performance invariant on the weight of the LFSR feedback polynomial have been reported in [29] (IDA) and [28]. These methods employ a number of moderate-weight parity checks available under assumption of certain exhaustive search in conjunction with a iterative decoding techniques. Four different iterative decoding techniques have been considered in [28] and their performance have been experimentally justified showing efficiency when only the short samples are available for cryptanalysis. The origins for the iterative based decoding FCAs reported in [30] include [32], [33] and [31].

Implications on Security Evaluation and Design of Stream Ciphers. The considered one-step decoding based FCAs appear as a powerful tool for security evaluation of certain stream ciphers assuming that enough long sample for cryptanalysis is available. The performance of these FCAs can be heavily degraded if only a short sample is available for cryptanalysis. In these scenarios, when only short samples are available, the iterative decoding based FCA considered in this section appears as a suitable alternative.

Accordingly, the security evaluation and the design guidelines should take into account considering both the one-step and iterative decoding based FCAs.

3. A Low-Complexity and High-Performance Algorithm for the Fast Correlation Attack

As an in-details illustration of the topic, this section provides a self-contained presentation of an algorithm for FCA which has the following two advantages over the related previously reported ones: (i) it is more powerful and (ii) it provides a high-speed software implementation, as well as a simple hardware one, suitable for (highly) parallel architectures. This chapter is mainly based on the results reported in [49], [46], [29], [26], [30], [28] and [6].

The discussed algorithm is a method for the fast correlation attack with significantly better performance in comparison with the previously reported methods, assuming a lower complexity and the same inputs. The algorithm is based on decoding procedures of the corresponding binary block code with novel constructions of the parity-checks, and employment of the following two decoding approaches: the a posterior probability based threshold decoding and the belief propagation based bit-flipping iterative decoding. These decoding procedures offer good trade-offs between the required sample length, overall complexity and performance. The discussed algorithm is compared with previously reported fast correlation attacks based on convolutional codes and turbo decoding: The underlying principles, performance and complexity are compared, and the gain obtained is pointed out.

3.1. Preliminaries. An important method for attack or security examination of certain stream ciphers based on nonlinear combination keystream generators composed of several linear feedback shift registers (LFSR's) (see [98], for example) are the basic correlation attack [106], and particularly the fast correlation attacks considered in a number of papers, including [97], [49], [46], [74], [87], [88] and [33]. Developing or improving techniques for realization of the fast correlation attack is still an important topic of cryptology.

The basic ideas of all reported fast correlation attacks include the following two main steps:

–Transform the cryptographic problem into a suitable decoding one;

–Apply (devise) an appropriate decoding algorithm.

There are two main approaches for realization of the fast correlation attack. The first one is based on decoding techniques for block codes (introduced in [97] and [109]), and the second one is based on decoding techniques for convolutional codes (proposed in [87] and [88]).

The main underlying ideas for a number of the fast correlation attacks based on linear binary block codes decoding is the iterative decoding principle introduced in [79]. For example, the fast correlation attacks reported in [97], [109], [49], [46] and [74], could be considered as variants of iterative decoding based on simple bit-flipping (BF) [79] or iterative extensions of *a posterior probability* (APP) decoding. Most of these methods (practically all except the method from [49]) are restricted on the LFSR feedback polynomials of low weight. Due to the established advantages of belief propagation (BP) based iterative decoding over iterative APP (see [32], for example), the application of BP based iterative decoding for realization of the fast correlation attack has been reported in [33]. The main goal of [33] was to report

the potential gain and its origins when BP based iterative decoding is employed instead of APP based decoding, assuming the same construction method of the parity-checks and the same overall structure of the algorithm for fast correlation attack. A comparison of the iterative decoding approaches based on simple, APP and BF based decodings for the fast correlation attack is reported in [46].

Alternative approaches for fast correlation attack based on the theory of convolutional codes are given in [87]-[88]. They can be applied to arbitrary LFSR feedback polynomials, in opposite to the previous methods, which mainly focus on feedback polynomials of low weight. The proposed algorithm transforms a part of the code $\mathbf{C}$ steaming from the LFSR sequence into a convolutional code, based on finding suitable parity check equations for $\mathbf{C}$. The approach considers a decoding algorithm that includes memory, but still has a low decoding complexity. With respect to the previous methods, this allows looser restrictions on the parity check equations that can be used, leading to many more equations. As the final decoding method, the Viterbi algorithm with memory orders of 10-15 was used. The results reported in [87] improve significantly the few previous results for high weight feedback polynomials, and are in many cases comparable with that corresponding to low weight feedback polynomials. Further developments of the idea for fast correlation attack based on decoding of certain convolutional codes are presented in [88] where new methods employing the techniques used for constructing and decoding turbo codes are proposed. The most powerful technique presented in [88] is based on the turbo decoding approach with M component convolutional codes and iterative APP decoding employing the BCJR algorithm [67].

Interests and the advances in developing algorithms for the fast correlation attack have raised a natural question of further improvements of the fast correlation attack, especially in the light of fast implementations.

The main goal of this section is to discuss the algorithm reported in [29] for the fast correlation attack suitable for a high-speed software implementation, as well as for a simple hardware one. Most previously reported algorithms can be considered as inappropriate ones for this goal assuming an LFSR feedback polynomial of arbitrary weight. Accordingly, the intention is to point out to an algorithm which employs $mod2$ additions and simple logical operations for processing, so that it is suitable for highly parallel architectures and high speed software or hardware implementations. Also, our goal is to point out to an algorithm which yields possibility for trade-offs between length of the required sample, overall complexity and performance.

In this section, a powerful algorithm for the fast correlation attack [29] is presented. The discussed algorithm is based on a novel method for constructing the parity-checks, motivated by the approach of [87] and [88], and two decoding approaches of the corresponding binary block code, APP threshold decoding and iterative decoding employing BP-like BF (see [79]). The construction of the parity-checks is based on searching for certain parity-check equations and theirs linear combinations employing the finite-state machine model of an LFSR with primitive characteristic polynomial. The expected numbers of parity-checks per parity bit

are derived, showing that a large number of appropriate parity-checks can be constructed. An analysis of the algorithm performance and complexity is presented. The novel algorithm is compared with recently proposed improved fast correlation attacks based on convolutional codes and turbo decoding. The underlying principles, performances and complexities are compared, and the gains obtained with the novel approach are pointed out. It is shown that assuming the same input, the novel algorithm yields better performance and lower complexity than the best algorithm reported before it.

This section is organized as follows. Subsection 2 presents preliminaries. Subsection 3 points out the main underlying results for the construction of a novel algorithm for the fast correlation attack. Complete specification of the proposed algorithm is given in subsection 4. Experimental analysis of the performance is presented in subsection 5, as well as a discussion of the complexity issue. Comparisons between the previously reported fast correlation attacks, and in [29] proposed algorithm are given in subsection 6. Finally, the main issues are summarized in subsection 7.

3.2. Decoding Concept for the Fast Correlation Attack. Recall that, the correlation means that the mod 2 sum of corresponding outputs of the LFSR and the generator can be considered as a realization of a binary random variable which takes value 0 and 1 with the probabilities $1-p$ and p, respectively, $p \neq 0.5$.

The fast correlation attack on a particular LFSR, with primitive feedback polynomial, in a nonlinear combining generator given the segment of the generator output can be considered as follows:

- The n-bit segment of the output sequence from the length-k LSFR is a codeword of an (n, k) punctured simplex code;
- The corresponding n-bit segment of the nonlinear combination generator output is the corresponding noisy codeword obtained through a BSC with crossover probability p;
- The problem of the LFSR initial state reconstruction, assuming known characteristic polynomial, is equivalent to the problem of decoding after transmission over a BSC with crossover probability p.

The decoding approach employed in this section is based on combination of a restricted exhaustive search over a set of hypotheses and a one-step or an iterative decoding technique. The exhaustive search is employed in order to provide a possibility for construction of suitable parity-check equations relevant for high performance of complete decoding. This approach could be considered as a particular combination of the minimum distance decoding and another decoding technique.

Recall that a parity-check equation which involves a smaller number of bits is more powerful than a higher weight one. Also note that performance associated with a set of the parity-checks depends on its cardinality as well as on the parity-check weight distribution. Finally, the overall complexity of a decoding procedure depends on the number and weights of the employed parity-checks. Accordingly, from performance and complexity point of views, a favorable situation corresponds to the availability of a large number of low-weight parity-checks.

In the following, x_n, $n = 1, 2, \ldots, N$, denotes an LFSR output sequence which is a codeword $\mathbf{x}$ of a binary (N, L) punctured simplex code $\mathbf{C}$ where N is codeword length and L is number of information bits. $\mathbf{x}_0 = [x_1, x_2, \ldots, x_L]$ is the vector of information bits identical to the LFSR initial state; $\{z_n\}$ denotes the degraded sequence $\{x_n\}$ after transmission over a BSC with crossover probability p. Accordingly, $z_n = x_n \oplus e_n$, $n = 1, 2, \ldots, N$, where the effect of the BSC with error probability p is modeled by an N-dimensional binary random variable $\mathbf{E}$ defined over $\{0, 1\}^N$ with independent coordinates E_n such that $\Pr(E_n = 1) = p$, $n = 1, 2, \ldots, N$, and e_n is a realization of E_n. Applying a codeword $\mathbf{x} = [x_n]_{n=1}^N \in \mathbf{C}$, to the input of the BSC, we obtain the random variable $\mathbf{Z} = \mathbf{E} \oplus \mathbf{x}$ as a received codeword at its output. Let $\mathbf{z} = [z_n]_{n=1}^N$ and $\mathbf{e} = [e_n]_{n=1}^N$ denote particular values of the random vector variables $\mathbf{Z}$ and $\mathbf{E}$, respectively.

3.3. Parity-Check Sets. This section points out novel sets of the parity-check equations relevant for construction of an algorithm for the fast correlation attack which will be proposed in the next section. Also, this section points out the expected cardinalities of these sets.

3.3.1. Preliminaries. An LFSR can be considered as a linear finite state machine. Recall that a linear finite state machine is a realization or an implementation of certain linear operator. Accordingly, a state of a length-L LFSR after t clocks is given by the following matrix-vector product over GF(2):

$$\mathbf{x}_t = \mathbf{A}^t \mathbf{x}_0, \quad t = 1, 2, \ldots,$$

where $\mathbf{x}_t$ is an L dimensional binary vector representing the LFSR state after t clocks, $\mathbf{x}_0$ is an L dimensional binary vector representing the initial LFSR state (in notation that it has index L at the top and index 1 at the bottom), and $\mathbf{A}^t$ is the t-th power over GF(2) of the state transition $L \times L$ binary matrix $\mathbf{A}$. Assuming the LFSR characteristic polynomial $f(u) = 1 + \sum_{i=1}^{L} b_i u^i$, the matrix $\mathbf{A}$ is given by:

$$\mathbf{A} = \begin{bmatrix} b_1 & b_2 & b_3 & \ldots & & b_L \\ 1 & 0 & 0 & \ldots & & 0 \\ 0 & 1 & 0 & \ldots & & . \\ . & . & . & \ldots & & . \\ 0 & & & \ldots & 1 & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \\ . \\ \mathbf{A}_L \end{bmatrix}, \tag{3.1}$$

where each $\mathbf{A}_i$, $i = 1, 2, \ldots, L$, represents a $1 \times L$ binary matrix (a row-vector).

Powers of the matrix $\mathbf{A}$ determine algebraic replica of the LFSR initial state bits, i.e. linear equations satisfied by the bits of the codewords from the dual code. Accordingly, they directly specify the parity-checks.

Since our approach assumes an exhaustive search, over the first B information bits, the parity checks are obtained:

–directly from the powers of the matrix $\mathbf{A}$ corresponding to an arbitrary subset of the first B bits of the LFSR initial state and no more than three bits from the remaining $L - B$ bits of the initial state and the bit of the LFSR output sequence;

–as the mod2 sum of any two parity checks determined from the powers of the matrix $\mathbf{A}$ when this sum includes an arbitrary number of the first B bits of the LFSR initial state, at most one bit from the remaining $L-B$ bits of the initial state, and the two bits of the LFSR output sequence.

–as the mod2 sum of any three parity checks determined by the powers of matrix $\mathbf{A}$ when this sum includes an arbitrary number of the first B bits of the LFSR initial state, no bit from the remaining $L-B$ bits of the initial state, and the three corresponding bits of the LFSR output sequence.

As previously in this section pointed out, a desirable situation is that corresponding to as many low-weight parity-checks as possible. Following this fact and due to the comparison purposes with recently reported improved fast correlation attacks [87]–[88], we focus our intention mainly to parity-checks of effective weight three (i.e. without considering the first B bits), but also employ some of parity-checks of effective weight four as well (also note that parity-checks of an arbitrary weight could be considered).

3.3.2. Methods for Construction and Specification of the Parity-Check Sets. This subsection presents two methods for obtaining appropriate sets of parity-checks. The developed methods are related to the *information bits* (Method A) and to the *parity bits* (Method B) of the underlying punctured simplex code.

Method A: Parity-check sets related to the *information bits* of the underlying punctured simplex codeword. Note that $x_{L+n} = \mathbf{A}_1^n \mathbf{x}_0$, $n = 1, 2, \ldots, N-L$, where $\mathbf{A}_1^n$ is the first row of the n-th power of the state transition matrix $\mathbf{A}$. Accordingly, the basic parity-check equations (defined on the noisy sequence) are given by:

$$c_{L+n} = z_{L+n} \oplus \mathbf{A}_1^n \mathbf{z}_0, \quad n = 1, 2, \ldots, N-L,$$

where $\mathbf{z}_0 = [z_1, z_2, \ldots, z_L]$.

Assuming that the first B information bits are known, appropriate parity-check equations for the i-th information bit, $i = B+1, B+2, \ldots, L$ can constructed according to the following definition.

Definition 3.1. The set Ω_i of parity-check equations associated with information bit-i is composed of:

- All parity-check equations corresponding to the vectors $\mathbf{A}_1^n$ such that each $\mathbf{A}_1^n$ has arbitrary values in the first B coordinates, has value one at the i-th coordinate, and has two ones in all other information bit coordinates;
- All parity-check equations obtained as the mod2 sum of two other basic parity-check equations, $(z_m \oplus \mathbf{A}_1^m \mathbf{z}_0) \oplus (z_n \oplus \mathbf{A}_1^n \mathbf{z}_0)$, where m and n have arbitrary values providing that the vector sum $\mathbf{A}_1^m \oplus \mathbf{A}_1^n$ has arbitrary values in the first B coordinates, value one at the i-th coordinate, and value zero in the all other coordinates.

Note that for given parameters N, L, and B, the sets Ω_i, $i = B+1, B+2, \ldots, L$, can be constructed in advance through a search procedure in a preprocessing phase, and later used for any particular application with these given parameters.

Method B: Parity-check sets related to the *parity bits* of the underlying punctured simplex codeword. First, an appropriate form of the parity check matrix of a punctured simplex code is pointed out. Then a method for constructing the parity checks is given and the parity checks to be employed by the algorithm are specified by Definition 3.2.

Recall, that the fast correlation attack has been modelled by the decoding of an (N, L) punctured simplex code used over a BSC. Accordingly, the following statement points out an appropriate form of the code parity-check matrix. This particular form has a one-to-one correspondence with the finite-state machine model of an LFSR with primitive characteristic polynomial.

Proposition 3.1. *The parity-check matrix* $\mathbf{H} = [\, \mathbf{P}^T, \mathbf{I}_{N-L} \,]$ *of a punctured simplex code* (N, L) *with corresponding polynomial* $f(u) = 1 + \sum_{i=1}^{L} b_i u^i$*, where the binary matrix* $\mathbf{P}$ *is the* $L \times (N-L)$ *matrix of parity checks,* $\mathbf{P}^T$ *is its transpose, and* $\mathbf{I}_{N-L}$ *is the identity matrix of dimension* $(N-L) \times (N-L)$*, is specified by the following:*

$$\mathbf{P}^T = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \cdot \\ \cdot \\ \mathbf{P}_{N-L} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1^{(1)} \\ \mathbf{A}_1^{(2)} \\ \cdot \\ \cdot \\ \mathbf{A}_1^{(N-L)} \end{bmatrix},$$

where the m*-th row of the matrix* $\mathbf{P}^T$*, is an* L*-dimensional row vector* $\mathbf{A}_1^{(m)}$ *equal to the first row of the* m*-th power,* $\mathbf{A}^m$*, of the matrix* $\mathbf{A}$ *given in* (3.1)*.*

The construction of the parity-checks is based on searching for certain linear combinations of rows in an appropriate form of the parity-check matrix given by Proposition 3.1. Accordingly, the preprocessing phase of the algorithm includes the construction of the parity-checks according to the following algorithm which generates a set of parity checks for each parity bit. Each parity check includes certain B information bits, and no more than $W+1$ other arbitrary check bits.

Note that $W+1$ is used here instead three to illustrate that a straightforward generalization is possible where not only the parity-checks of effective weight equal to three are considered.

Algorithm for the construction of the parity checks

- *Input:* The parity check matrix $\mathbf{H} = [\, \mathbf{P}^T, \mathbf{I}_{N-L} \,]$.
- *Processing Steps:* For each parity bit, generate a set of parity check equations employing the following procedure.
  - For $n = L+1, L+2, \ldots, N$ and each w, $1 \leqslant w \leqslant W$, proceed as follows:
    - Calculate the *mod*2-sum of the n-th row of the parity-check matrix $\mathbf{H} = [\mathbf{P}^T, \mathbf{I}_{N-L}]$ and any possible w other rows.
    - If the values at positions $i = B+1, B+2, \ldots, L$, are all zeros, where $B < L$, is a predetermined parameter, record the considered combination into the set Ω_n^*.
- *Output:* The sets of parity check equations Ω_n^*, $n = L+1, L+2, \ldots, N$.

Definition 3.2. The set Ω_n^* generated by the above algorithm is the set of all considered parity-check equations related to the n-th parity bit of codewords in the punctured (N, L) simplex code.

Note that each parity-check in Ω_n^* consists of α of the first B information bits with $0 < \alpha \leqslant B$, none of the remaining last $L - B$ information bits and at most $W + 1$ of the $N - L$ parity check bits, including bit-n.

3.3.3. Expected Cardinalities of the Parity-Check Sets.

Lemma 3.1. *In any set Ω_i, specified by Definition* 3.1*, $i = B + 1, B + 2, \ldots, L$, a tight approximation about the expected number $|\bar{\Omega}|$ of the parity-checks is given by the following:*

$$|\bar{\Omega}| = 2^{B-L}\left[(N-L)\binom{L-B-1}{2} + \binom{N-L}{2}\right].$$

Note that Lemma 3.1 motivates the construction of Ω_i given in Definition 3.1. For each type of check sums in Ω_i, that corresponding to minimum weight with non negligible contribution to $|\bar{\Omega}|$ is chosen.

As an illustration, note that for $N = 40000$, $L = 40$ and $B = 18, 19, 20, 21, 22$, Lemma 3.1 yields that the expected cardinality, $|\bar{\Omega}|$ is equal to 192, 384, 768, 1534, 3066, respectively.

Lemma 3.2. *In any set Ω_n^*, specified by Definition* 3.2*, $n = L + 1, L + 2, \ldots, N$, a tight approximation about the expected number $|\bar{\Omega^*}|$ of the parity-checks is given by the following:*

$$|\bar{\Omega^*}| = 2^{-L+B}\sum_{w=1}^{2}\binom{N-L-1}{w}.$$

As an illustration, note that for $L = 40$, and (N, B)=(1024,26), (4096,22), (8192,20), and (16384,18), Lemma 3.2 yields that the expected cardinalities, $|\bar{\Omega^*}|$ are equal to 29.5, 31.4, 31.7, and 31.9, respectively.

Note that Lemmas 3.1 and 3.2 show that Definitions 3.1 and 3.2 yield large numbers of the parity-checks relevant for an error-correction procedure.

Also, note that Lemmas 3.1 and 3.2 imply that the expected cardinalities of the parity-check sets specified by Definitions 3.1 and 3.2 do not depend on the LFSR characteristic polynomial, and particularly on its weight.

3.4. Algorithm for Fast Correlation Attack. The main underlying principles for construction of the novel fast correlation attack include the following:

- General concepts of linear block codes decoding, and particularly:
  - decoding of information bits only, employing an APP based threshold decoding;
  - iterative decoding of the parity bits employing a reduced complexity BP based iterative decoding.
- A novel method for constructing parity checks of a punctured simplex code based on linear finite state machine model of an LFSR (see [49]);

- The idea (implicitly given in [87]) of employing a partial (restricted) exhaustive search in order to enhance performance of the fast correlation attack. The developed algorithm assumes exhaustive search over the first B information bits in conjunction with appropriate decoding approaches.

According to these principles an algorithm for the fast correlation attack (based on a linear block code decoding approach) has been developed. The algorithm is based on the methods for constructing the appropriate parity-checks presented in the previous section, and its processing phase includes the following three techniques: (i) hypothesis testing, (ii) decoding of a punctured simplex code and (iii) correlation check. The algorithm employs two different decoding procedures in order to provide desired trade-offs between necessary length of the sample, i.e. the rate of underlying code, performance and overall complexity.

Algorithm for the Fast Correlation Attack

INPUT:

- values of the parameters N, L, B, and the threshold T;
- the noisy received bits $z_1, z_2, \ldots, z_N$;
- for each information bit i, $i = B+1, B+2, \ldots, L$, the set Ω_i of corresponding parity-check equations (constructed in the preprocessing phase based on Definition 3.1), and for each parity bit n, $n = L+1, L+2, \ldots, N^*$, $N^* \leqslant N$, the set Ω_n of corresponding parity-check equations (constructed in the preprocessing phase based on Definition 3.2).

PROCESSING STEPS:

(1) *setting the hypothesis*
From the set of all possible 2^B binary patterns, select a not previously considered pattern $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_B$, for the first B information bits. If no new pattern is available, go to the Output (b).

(2) *decoding*
Employ one of the following two decoding algorithms for estimating a candidate for the information bits (i.e. LFSR initial state):
- One-Step Decoding Algorithm (OSDA) using parity-checks specified by Definition 3.1;
- Iterative Decoding Algorithm (IDA) using parity-checks specified by Definition 3.2.

(3) *correlation check*
Check if the current estimation of the information bits (obtained from the decoding step) $\hat{\mathbf{x}}_0 = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_L]$, is the true one, according to the following:
For $\hat{\mathbf{x}}_0$, generate the corresponding sequence $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N$, and calculate $S = \sum_{n=1}^{N} \hat{x}_n \oplus z_n$.
If $S \leqslant T$ go to Output (a), otherwise go to Step 1.

OUTPUT:

(a) the considered vector $\hat{\mathbf{x}}_0$ of information bits is the true one;
(b) the true vector of information bits is not found.

The threshold scalar T is used for checking a hypothesis over all the information bits. For given N, L, B, p, the threshold T is calculated based on the method presented in [106].

The specifications of the employed decoding algorithms OSDA and IDA are given in the following.

3.4.1. One-Step Decoding Algorithm-OSDA. OSDA decodes the noisy received sequence $[z_1, z_2, \ldots, z_N]$ for the (N, L) truncated simplex code employing an APP threshold decoding and the sets Ω_i of parity-check equations, specified by Definition 3.1, $i = B+1, B+2, \ldots, L$ according to the following.

- *parity-checks calculation*
  For each information bit position i, $i = B+1, B+2, \ldots, L$, calculate the parity-check values employing the parity check equations from the set Ω_i.
- *error-correction*
  For each i, $i = B+1, B+2, \ldots, L$ do the following:
  - if the number of satisfied parity-check equations for the considered information bit is smaller than the threshold $T_1(i)$ set $\hat{x}_i = z_i \oplus 1$, otherwise set $\hat{x}_i = z_i$.

The algorithm employs a vector threshold $\mathbf{T}_1 = [T_1(i)]_{i=B+1}^{L}$ which contains values for the APP threshold decoding of certain information bits.

Elements of the threshold vector $\mathbf{T}_1$ are determined based on the posterior error probabilities computed by using the parity-checks specified by Definition 3.1. We assume that for each codeword bit, the parity-checks used are orthogonal on that bit, meaning that except for that bit, every other involved unknown bit appears in exactly one of the parity-checks. Finally, assuming as an appropriate approximation, that all the parity-check equations involve exactly two unknown bits beside the considered one, for any $i = B+1, B+2, \ldots, L$, the threshold $T_1(i)$ is equal to the smallest integer such that the following inequality holds:

$$\frac{p}{1-p}\left(\frac{1+(1-2p)^2}{1-(1-2p)^2}\right)^{|\Omega_i|-2T_1(i)} \leqslant 1,$$

where $|\Omega_i|$ denotes the number of parity-check equations related to the i-th information bit.

3.4.2. Iterative Decoding Algorithm-IDA. For a given $N^* \leqslant N$, IDA decodes the received sequence $[z_1, z_2, \ldots, z_{N^*}]$ for the (N^*, L) punctured simplex code employing a BP based bit-flipping (BP-BF) iterative decoding and the sets Ω_n^* of parity-check equations, specified by Definition 3.2, $n = L+1, L+2, \ldots, N^*$.

BP-BF based iterative decoding (see [79], for example) includes the following main difference in comparison with simple BF.

- For each bit n, and each combination of $|\Omega_n^*| - 1$ parity-checks out of the $|\Omega_n^*|$ parity checks associated with bit-n, make $|\Omega_n^*|$ estimate of the nth bit value associated with these combinations.

Accordingly, we employ the following iterative BP-BF based decoding algorithm.

- *Initialization*: $\hat{x}_n = z_n$ and $\hat{x}_{nm} = z_n$.

- *Iterative Processing*
  (1) *Step* 1:
  (a) For each n and for each $m \in \Omega_n^*$, evaluate:
  $\sigma_n(m) = \sum_{n' \in \omega(m)} \hat{x}_{n'm}\ [mod 2]$.
  (b) If all $\sigma_n(m) = 0$ go to Step 3 (a). If some maximum number of iterations (e.g. 30) is exceeded go to Step 3 (b).
  (2) *Step* 2: For each n, do the following:
  (a) If $\sum_{m}^{|\Omega_n^*|} \sigma_n(m) \geqslant |\Omega_n^*|/2$, then $\hat{x}_n = \hat{x}_n \oplus 1$.
  (b) If $\sum_{m'}^{|\Omega_n^* \setminus m|} \sigma_n(m') \geqslant |\Omega_n^* \setminus m|/2$, then $\hat{x}_{nm} = \hat{x}_{nm} \oplus 1$.
  If no complementation was performed go to Step 3 (b); otherwise go to Step 1.
  (3) *Step* 3:
  (a) $\hat{\mathbf{x}} = [\hat{x}_n]$ is the decoding result.
  (b) Algorithm halts and a warning is declared that a valid decoding is not reached.

3.5. Performance and Complexity.

3.5.1. Performance. The performance of the novel algorithm is experimentally considered when the LFSR characteristic polynomial is chosen as $1 + u + u^3 + u^5 + u^9 + u^{11} + u^{12} + u^{17} + u^{19} + u^{21} + u^{25} + u^{27} + u^{29} + u^{32} + u^{33} + u^{38} + u^{40}$ and $N = 40000$ (i.e. assuming the same example as was considered in [87]–[88]). Note that the proposed algorithm can be applied for values of L significantly longer than $L = 40$, but this value was employed in all numerical and experimental illustrations for comparison with previously reported results.

Results of the performance analysis are presented in Table 1. This table displays the error-rate of the LFSR initial state reconstruction as a function of the correlation noise p when the algorithm employs:

(i) OSDA with B=18, 19, 20, 21, 22,

(ii) IDA with $N^* = 4096$, $B = 22$, and at most 20 iterations.

Each error-rate given in the table is obtained by calculation over a corresponding, randomly selected, set of 1000 samples. Recall that "error-rate" indicates the fraction of trials for which we obtain incorrect decoding (and accordingly incorrect reconstruction of the secret key).

3.5.2. Complexity. Recall that the overall complexity assumes time and space complexity requirements. The complexity analysis yields that according to the structure of the proposed algorithm:

–The algorithm requires a space for the input. Space requirements for the decoding process are as follows: when OSDA is employed, decoding processing does not require memory; IDA requires a memory proportional to the parameter N^*;

–Time complexity is specified by the following corollaries.

Corollary 3.1. *Assuming that OSDA is employed, that* $|\Omega|$ *denotes the average cardinality of the parity-check sets* $|\Omega_i|$*, that* ω *denotes the average number of bits*

TABLE 1. Performance of the novel algorithm-experimental analysis: Error-rate of the LFSR initial state reconstruction, as a function of the correlation noise p when the LFSR length is $L = 40$, the characteristic polynomial weight is 17, and the length of the sequence available for processing is $N = 40000$ bits.

p	Error rate of LFSR initial state reconstruction					
	OSDA $B = 18$	OSDA $B = 19$	OSDA $B = 20$	OSDA $B = 21$	OSDA $B = 22$	IDA $B = 22$, $N^* = 4096$
0.25	0.009	0.000	0.000	0.000	0.000	0.000
0.26	0.015	0.000	0.000	0.000	0.000	0.000
0.27	0.024	0.000	0.000	0.000	0.000	0.000
0.28	0.081	0.000	0.000	0.000	0.000	0.000
0.29	0.159	0.006	0.000	0.000	0.000	0.000
0.30	0.254	0.023	0.000	0.000	0.000	0.000
0.31	0.384	0.041	0.002	0.000	0.000	0.000
0.32	0.569	0.098	0.002	0.000	0.000	0.000
0.33	0.696	0.226	0.020	0.000	0.000	0.000
0.34	0.838	0.356	0.053	0.001	0.000	0.000
0.35	0.915	0.542	0.114	0.002	0.001	0.000
0.36	0.955	0.743	0.225	0.019	0.022	0.000
0.37	0.983	0.865	0.450	0.080	0.062	0.001
0.38	0.990	0.932	0.652	0.210	0.208	0.023
0.39	0.997	0.980	0.850	0.445	0.399	0.052
0.40	1.000	0.988	0.935	0.663	0.651	0.267

in a parity-check, and that w denotes the weight of the LFSR characteristic polynomial, the implementation complexity of the proposed algorithm is proportional to $2^B[(L-B)|\Omega|\omega + (N-L)w]$ mod2 additions.

Corollary 3.2. *Assuming that IDA is employed, that $|\Omega^*|$ denotes the average cardinality of the parity-check sets $|\Omega_n|$, that ω^* denotes the average number of bits in a parity-check, that I denotes the number of iterations, and that w denotes the weight of the LFSR characteristic polynomial, the implementation complexity of the proposed algorithm is proportional to $2^B[I(N^*-L)|\Omega^*|(|\Omega^*|-1)\omega^* + (N-L)w]$ mod2 additions.*

Note also that from the structure of the proposed algorithms, it is readily seen that the proposed algorithms are suitable for fast software implementation, as well as for simple hardware implementation: the algorithms employ only simple arithmetic operations (*mod*2 addition) and simple logical operations.

Also, since the decoding process is mainly memoryless, note that a reduction of the time complexity specified by the previous corollaries can be obtained by an appropriate time-memory complexity trade-off.

Finally note that in the presented experiments, the decoding step has employed the underlying codeword lengths $N = 40000$ and $N^* = 4096$ for OSDA and IDA, respectively. This is an illustration that OSDA and IDA yield a trade-off between the length of the required sample (i.e. the code rate) and the decoding complexity.

3.6. Comparison of the Discussed Algorithm with Previously Reported Fast Correlation Attacks. This section presents a comparative analysis of the underlying principles, performance and complexity of recently proposed improved fast correlation attacks [88] and the novel algorithm, assuming the same input.

3.6.1. Comparison of the Underlying Principles. Comparison of the underlying principles employed in [87]–[88] and in the novel algorithm for the fast correlation attack can be summarized as follows.

- The approaches of [87]-[88] are based on decoding of convolutional codes and turbo codes with convolutional codes as the component codes constructed over the LFSR sequence. The novel approach is based on decoding punctured simplex block codes corresponding to the LFSR sequence.
- The algorithms [87]-[88] and the novel algorithm employ different parity-checks.

  The parity-checks employed in [88]-[87] are constructed by searching for these parity checks which include the following bits: currently considered bit, bits from a subset of B previous bits, and no more than two other bits.

  The parity-checks employed in the novel algorithm are constructed by searching for these parity checks which include the following bits:

  (i) currently considered information bit, bits from a subset of B first information bits, and two other information bits with the corresponding parity-bit, or two arbitrary parity bits only, or

  (ii) currently considered parity bit, bits from a subset of B first information bits, and no more than two other parity bits.

  Note that these different approaches in the parity-check constructions imply different number of parity-checks per bit, as well.
- The decoding techniques employed in [87]-[88] are Viterbi decoding, BCJR decodings, and MAP turbo decoding (see [67]). On the other hand the novel algorithm employs the following two low-complexity decoding techniques: (i) APP threshold decoding, and (ii) BP based BF iterative decoding.
- The fast correlation attacks from [87]-[88] implicitly include an exhaustive search over a set of dimension 2^B through employment of the Viterbi or BCJR decodings due to the trellis search. The novel algorithm employs an explicit search over all 2^B possible patterns corresponding to the first B information bits.
- A decoding process based on the Viterbi or BCJR algorithm requires a memory of dimension proportional to 2^B. On the other hand, OSDA does not require memory, and IDA requires a memory proportional to the parameter N^*.

3.6.2. Comparison of the Performance and Complexity. For the performance comparison of the novel and turbo based fast correlation attacks [88] the same inputs are employed and relevant parameters are selected so that the novel

TABLE 2. Comparison of the algorithms performance, assuming the same inputs, and lower complexity of the novel algorithm in comparison to the turbo algorithm [9]: Limit noise for which the algorithms yield, with probability close to 1, correct reconstruction of the initial LFSR state, when the LFSR characteristic polynomial is $1+u+u^3+u^5+u^9+u^{11}+u^{12}+u^{17}+u^{19}+u^{21}+u^{25}+u^{27}+u^{29}+u^{32}+u^{33}+u^{38}+u^{40}$, and the available sample is 40000 bits.

ALGORITHM	Limit Noise
turbo algorithm [88]: $B=15$, $M=2$	0.27
novel algorithm with OSDA: $B=19$	0.28
turbo algorithm [88]: $B=15$, $M=4$	0.29
novel algorithm with OSDA: $B=21$	0.33
turbo algorithm [88]: $B=15$, $M=16$	0.30
novel algorithm with OSDA: $B=22$	0.34
novel algorithm with IDA: $N^*=4096$, $B=22$	0.36

algorithm always has significantly lower overall implementation complexity than the algorithm [88].

According to [88], the time complexity of the turbo decoding is proportional to $2^B IMJm$ real multiplications where I denotes the number of the iterations, M the number of the component codes, J the number of processed bits, and m the number of employed parity-checks per bit. The time complexity of the novel algorithm is given in Corollaries 3.1 and 3.2.

Also note that the space complexity of the approach from [88] is proportional to 2^B due to employment of the BCJR algorithm. If OSDA is employed no space complexity is required, and if IDA is employed it is usually significantly smaller than 2^B due to its linear rather than exponential nature.

An illustrative performance comparison is presented in the Table 2. Note that, in each case, the complexity of the proposed algorithm could be considered as significantly lower than complexity of the turbo decoding [88] although the proposed algorithm assumes search over a much larger set of hypotheses, since: (i) [88] employs iterative processing with M component codes and (ii) the dominant arithmetic operation in the proposed algorithm is $mod2$ addition against real multiplication for the turbo based decoding of [88].

Finally, note that the actual time for performing the attack by the novel algorithm strongly depends on the implementation constraints so that a straightforward comparison is not appropriate. Also, the approaches of [87]-[88] can be modified to involve $mod2$ additions, but at the expense of performance degradation.

3.7. Concluding Notes. The considered algorithm for the fast correlation attack is based on decoding procedures of the corresponding binary block code with novel constructions of the parity-checks, independent of the LFSR feedback polynomial weight, and the following two decoding approaches are employed: an APP based threshold decoding and a BP based BF iterative decoding. The constructions of the

parity-checks are based on searching for certain parity-check equations and their linear combinations employing the finite-state machine model of an LFSR with primitive characteristic polynomial. The expected numbers of the parity-checks per parity bit have been derived, showing that a large number of appropriate parity-checks can be constructed.

The experimental consideration of the algorithm performance shows that the algorithm is a powerful one.

The overall implementation complexity has been specified. As dominant operations the algorithm employs *mod*2 additions and simple logical operations, so that it is very suitable for high-speed software implementation as well as for simple hardware implementation.

The algorithm offers good trade-offs between required sample length (i.e. rate of the underlying code), overall complexity and performance. The one-step threshold decoding approach yields high performance assuming long enough sample, and the iterative decoding approach can reach the same performance using a significantly shorter sample but at the expense of increased complexity.

The algorithm has been compared with recently reported improved fast correlation attacks based on convolutional codes and turbo decoding. The underlying principles, performance and complexity have been compared, and the essential gain obtained with the novel approach is pointed out. The developed algorithm has the following two main advantages over other previously reported ones:

(a) Assuming a lower overall complexity, and the same inputs, the algorithm yields significantly better performance.

(b) It is suitable for high-speed software implementation as well as for simple hardware implementation and highly parallel architectures.

4. Certain Approaches for Randomized Stream Ciphers

This section discusses design and security evaluation issues regarding a class of stream ciphers known as randomized stream ciphers. The security evaluation involves computational complexity as well as information theoretic ones. After certain introductory notes an approach for design of stream ciphers based on joint employment of pseudorandomness, randomness and dedicated coding, is in-details considered. As a generalization of the discussed approach, a generic framework for developing randomized stream ciphers is pointed out and elements of its security evaluation from information-theoretic and computational-complexity points of view are given. This section is mainly based on the results reported in [4], [51], [61] with origins in [7] and [9].

4.1. Introduction. Randomized symmetric key encryption as an alternative encryption paradigm has been reported in [103]. According to [103], the randomized encryption is a procedure which enciphers a message by randomly choosing a ciphertext from a set of ciphertexts corresponding to the message under the current encryption key, and the following is claimed, [103]: "At the cost of increasing the required bandwidth, randomized encryption procedures may achieve greater cryptographic security than their deterministic counterparts ...".

Stream ciphers are an important class of encryption techniques for providing data secrecy. Traditional stream ciphers do not include any randomness in generation of the outputting ciphertext: They are based on the deterministic operations which expand a short secret seed into a long pseudorandom sequence. This paper points out to a novel approach for design of stream ciphers based on a combination of the pseudo-randomness and randomness.

Usefulness of involvement pure randomness into a cryptographic primitive has been recognized in a number of reported designs and particularly in the following ones. McEliece public-key system [96], based on decoding complexity after a noisy channel, is the classical and a very illustrative example of the randomness involvement. In [103], a number of approaches for including randomness in the encryption techniques have been discussed mainly regarding block and stream ciphers.

In [70], a pseudorandom number generator based on the Learning from Parity with Noise (LPN) problem, derived from an older proposal of one-way function based on the hardness of decoding a random linear code, has been reported. (Informally note that the LPN problem can be considered as the problem of solving a system of linear equations corrupted by noise. or a problem of decoding a linear code). Recently a number of randomized symmetric key encryption techniques has been reported [81], [4], [51] [65] and [61].

In [81], a probabilistic private-key encryption scheme named LPN-C whose security can be reduced to the hardness of the LPN problem has been proposed and considered. Recently, in [65] a symmetric encryption scheme similar to the one reported in [81] is reported and its security and implementation complexity are analyzed. The symmetric encryption schemes reported in [81] and [65] appears as interesting and stimulating for further considerations (having in mind improvements as well) particularly because the security is related to the recognized hard (LPN) problem.

A different approach for achieving secrecy of communication has been reported in [108] assuming that the channel between the legitimate parties is with a lower noise in comparison with the channel via which a wire-tapper has access to the ciphertext. The method proposed in [108] does not require any secret: It is based on a specific coding scheme which provides a reliably communications within the legitimate parties and prevents, at the same time, the wire-tapper from learning the communication's contents. Wire-tap channel coding is based on assigning multiple codewords to the same information vector and from that point of view, it shares the same underlying idea employed in the homophonic coding, or homophonic substitution (see [86], for example). A basic cryptographic application of homophonic coding is to convert the plaintext into a sequence of completely random (equiprobable and independent) code letters. An approach to provide secrecy employing an error-correcting code, in a scenario similar to the wire-tap channel, has been reported in [105]. Under the assumption that an attacker is forced to wire-tap the communications via a channel with a noise, the following scheme for providing secrecy is proposed in [105]: To encrypt a bit, the sender randomly selects a bit sequence whose parity is equal to the message bit, choosing this sequence to be long

enough so that, due to the noise in the wire-tap channel, the attacker is unable to determine the parity of the codeword.

Also, the following results have been reported regarding security usefulness of pure randomness in cryptographic primitives. Effects of random noise and wire-tap channel coding regarding certain quantum stream ciphers have been considered in [7]. The trapdoor cipher TCHo has been proposed in [66] where the additive noise has been employed to mask a pseudorandom sequence generated by an LFSR with feedback polynomial, which has a low-weight multiple, used as the trapdoor. An approach for design of stream ciphers employing error-correction coding and certain additive noise degradation of the keystream has been reported in [89]. A message is encoded before the encryption so that the decoding, after mod 2 addition of the noiseless keystream sequence and the ciphertext, provides its correct recovery. Resistance of this approach against a number of general techniques for cryptanalysis, has been also considered in [89].

4.2. A Stream Cipher Based on Embedding Pseudorandomness and Randomness. This section yields and analyzes an approach for design of stream ciphers based on joint computing over random and secret data. Feasibility of encryption/decryption computation when the ciphertext involve pure random data is shown. The core element of the proposed approach for stream ciphering is a pseudorandom embedding of the random bits into the ciphertext and this embedding plays role of a homophonic encoding. The initial ciphertext with the embedded random bits is further on intentionally degraded by its exposure to a moderate noise which can be modelled as the binary symmetric channel effect. A security evaluation of the proposed approach implies that its security appears as a consequence of hardness of the LPN problem, as well. The developed design has potential of providing that complexity of recovering the secret key in the known plaintext attack scenario is close to the complexity of recovering the secret key via the exhaustive search, i.e. close to the maximal possible one for the given size of the secret key. The proposed approach can be considered as a trade-off between the increased security and decreased communications efficiency which in a number of scenarios appears as a suitable one.

4.2.1. Introduction. The discussed construction originates from a consideration of the possibilities for some novel approaches for inclusion of pure randomness into a stream cipher framework. The main goal of employment the pure randomness is to provide a supporting element for achieving the maximum possible security of a stream cipher, i.e. to make it as high as it can be for the given secret key dimension. Also, the involvement of the randomness is considered in a manner that provides a low-complexity implementation as well as a low communications overhead. As the result, this paper yields the following: (i) a proposal of stream ciphers class which involve pure randomness; (ii) a discussion of the impact of randomness on the security of the proposed class of stream ciphers and for a particular family of the class the security statement based on the LPN problem hardness; (iii) a discussion on the implementation complexity and the communications overhead of the proposed class of stream ciphers.

This subsection is organized as follows. Part 4.2.2 contains certain preliminaries. Part 4.2.3 yields the underlying ideas for the design and the framework of the proposed stream ciphers. Part 4.2 4 specifies the related encryption and decryption algorithms as well as a particular instantiation of the proposed stream ciphers. A preliminary security evaluation of the proposed stream ciphers framework is given in the part 4.2.5, and a formal security evaluation of a particular instantiation is given in the part 4.2.6. Part 4.2.7 yields a consideration of the implementations complexity and the communications overhead. Finally, some concluding notes are pointed out in the part 4.2.8.

4.2.2. Preliminaries. This section introduces certain notations and, as a background, yields a brief overview of the LPN problem.

Notations. This paper employs the following particular notations.

Drawing from a distribution. Given a finite set G and a probability distribution Δ on G, $g \leftarrow \Delta$ denotes the drawing of an element of G according to Δ. $g \leftarrow G$ denotes the random drawing of an element of G according to the uniform probability distribution.

Bernoulli distributions. Ber_η denotes the Bernoulli distribution with the parameter $\eta \in [0, 1/2]$, i.e. a bit $\nu \leftarrow \mathrm{Ber}_\eta$ is such that $\Pr[\nu{=}1] = \eta$ and $\Pr[\nu{=}0] = 1 - \eta$. Vectorial distribution $\mathrm{Ber}_{n,\eta}$ is defined as follows: An n-bit vector $\mathbf{v} \leftarrow \mathrm{Ber}_{n,\eta}$ is such that each bit ν of $\mathbf{v}$ is independently drawn according to Ber_η.

Oracles. $\mathcal{U}_n$ denote the oracle returning independent uniformly random n-bit strings. LPN oracle: For a fixed k-bit string $\mathbf{s}$, $\Pi_{s,\eta}$ will be the oracle returning independent $(k+1)$-bit strings according to the distribution (to which we will informally refer to as an LPN distribution):

$$\left\{\mathbf{a} \leftarrow \{0,1\}^k \,;\, \nu \leftarrow \mathrm{Ber}_\eta : (\mathbf{a}, \mathbf{a}\cdot\mathbf{s} \oplus \nu)\right\}$$

The LPN Problem. Informally, Learning from Parity with Noise (LPN) problem can be described as learning an unknown k-bit vector $\mathbf{s}$ given noisy versions of its scalar product $\mathbf{a}\cdot\mathbf{s}$ with randomly selected vectors $\mathbf{a}$.

In a formal manner, the LPN problem is the problem of retrieving $\mathbf{s}$ given access to the oracle $\Pi_{s,\eta}$. For a fixed value of k, we will say that an algorithm $\mathcal{A}(T, q, \delta)$-solves the LPN problem with noise parameter η if $\mathcal{A}$ runs in time at most T, makes at most q oracle queries, and

$$\Pr\left[\mathbf{s} \leftarrow \{0,1\}^k : \mathcal{A}^{\Pi_{s,\eta}}(1^k) = s\right] \geqslant \delta$$

By saying that the LPN problem is hard, we mean that any efficient adversary solves it with only negligible probability. There is a significant amount of literature dealing with the hardness of the LPN problem. It is closely related to the problem of decoding a random linear code and it is NP-hard.

It is NP-hard to even find a vector $\mathbf{x}$ satisfying more than half of the equations outputted by $\Pi_{s,\eta}$. The LPN average-case hardness has also been extensively investigated and one of the currently best algorithms for this case has been reported in [9].

TABLE 3. The framework of the main operations at the sender's and receiver's sides: "Embedding" assumes interleaving of the effective and random (dummy) bits and "splitting" assumes separation of the effective and dummy bits.

Sender :	Encode → Encrypt → Embedding & Additive Nose Degradation
Receiver :	Splitting → Decrypt → Decode (Decimation)

4.2.3. A Framework for the Stream Ciphers Design. This section yields underlying ideas for design of stream ciphers which involve pure randomness and the architecture of the proposed stream ciphers.

Underlying Ideas. The novel design assumes the following: (i) a source of pure randomness is available (for example, as an efficient hardware module); and (ii) a suitable error-correcting coding (ECC) techniques is available. The availability means that the implementation complexities of the source of randomness and ECC do not imply a heavy implementation overhead in suitable implementation scenarios.

The main design goal is the following one: Any method for cryptanalysis of a novel stream cipher scheme should have complexity close to the complexity of the exhaustive search. Particular origins for achieving the design goals include the results reported in [7], [9], [6] and [5], where certain issues regarding coding and randomness, complexity of the LPN problem, and generic time-memory-data trade-off method for recovering the secret key are considered.

The novel approach for design of stream ciphers is based on the following:

–employment of the pure randomness for the intentional data degradation;

–employment of a dedicated homophonic-like coding which involves pure randomness.

Note that in the considered scenario, the homophonic coding does not have the same role as in its traditional applications where the role is to provide randomness of the plaintext. Here, a homophonic coding is employed to provide additional confusion at the attacker's side.

So, the main underlying ideas of a framework for stream ciphers which involves pure randomness and provide low-complexity implementation include the following:

- Encoding/Decoding of the plaintext;
- Encryption/Decryption of the encoded plaintext/ciphertext;
- Homophonic encoding via embedding random bits and an intentional degradation of the codewords before transmission.

Accordingly, the framework of the main operations at the sender's and receiver's sides is given in Table 3.

Regarding the underlying design ideas given in this section and some of the previously reported ones, note the following. Certain randomized stream cipher

approaches based on the insertions of random bits are considered in [103] including the following relevant ones: (i) pseudo-random interspersing random bits after encryption, (ii) random interspersing random bits before encryption and pseudo-random encryption of the random control sequence. Note that these approaches are based on the random bits embedding but do not include neither employment of error-correction codes neither the additive degradation by random bits. On the other hand, independently of this paper, recently in [89], it has been proposed an approach for stream ciphers which includes error-correction coding and deliberate additive random degradation. The approach [89] is based on error-correction coding of the plaintext so that it can be correctly recovered when a randomized keystream is employed for encryption. Randomization of the keystream is performed via its degradation by randomly selected error patterns which are such that provide the decodability. Note that the approach from [89] does not include any embedding of the random bits in the employed processing. Finally, regarding a comparison with the approaches reported in [103] and [89], note that the underlying ideas of the design given in this section include joint employment of randomness via the embedding and the additive degradation, as well as employment of the dedicated error-correction coding, implying a noticeable conceptual difference between the proposed approach and the reported ones.

Components, Roles and Architecture. In comparison with a traditional stream cipher which performs "encoding & encryption", the structure of the proposing one has the following three additional components:

(1) a source of pure randomness called RAND-box;
(2) a component, which at the encryption side performs homophonic encoding of the ciphertext via embedding the random bits and at the decryption side performs "decoding" via the (corresponding) decimation which provides splitting of the embedded bits;
(3) a component at the encryption side which simulates a binary symmetric channel with controllable crossover probability.

Let's call ECC-box a box which encodes the plaintext in order to provide correction of the random errors. Note that, in the proposing stream cipher, ECC-box encodes the plaintext so that it can be recovered correctly after corruption due to the errors introduced intentionally in the ciphertext (in a general setting, certain noise in the public channel can be involved as well).

Block scheme of the considered stream cipher family is depicted in Fig. 1. The "white" boxes in Fig. 1 correspond to the boxes in a traditional stream cipher which performs "encoding+encryption" in order to perform reliable operation over a noisy communication channel, and the "gray" boxes are the additional ones.

The role of the employed homophonic encoding, implemented via the embedding of the random bits, is to provide a heavy masking of the keystream generator sequences so that they appear as very uncertain for a given ciphertext even when the plaintext is known.

Accordingly, the main features of the proposed stream ciphers framework are as follows.

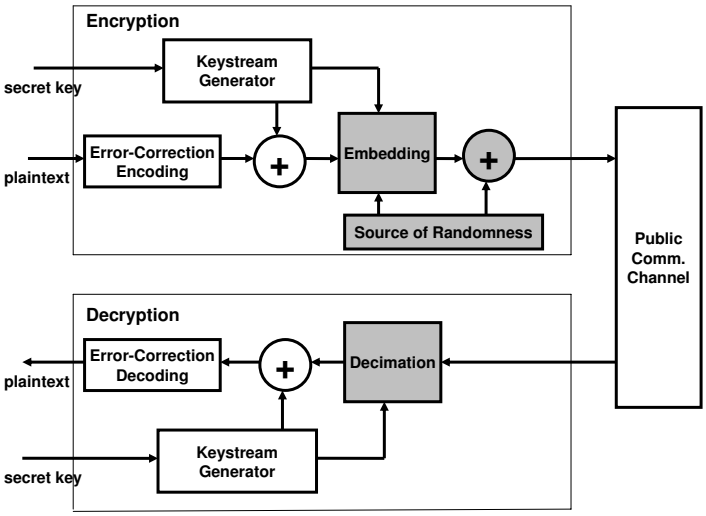

FIGURE 1. A framework of randomized stream ciphers.

- The resulting ciphertext consists of effective bits and dummy ones embedded in a manner controlled by the secret key.
- At the receiving part, the dummy bits are simply discarded and the effective bits are those which are employed for the deciphering. The decimation assumes splitting of the effective and dummy bits.
- The security is based on the impossibility of correct separation of effective bits from the dummy ones via the decimation of the available (embedded) sequence without the secret key.

4.2.4. Encryption&Decryption Algorithms and a Particular Instantiation. This section specifies the encryption and decryption algorithms in the proposed class of stream ciphers and, as an instantiation of the general framework, a particular family of the ciphers is defined.

Encryption and Decryption Algorithms

Encryption Algorithm

- *Input*: The message organized as a string of l-dimensional binary vectors $\{\mathbf{x}_t\}_t$, the secret key & non-secret initial vector which control the keystream generator, and the algorithm parameters m, n and η.
- *Encryption Steps.* For each t do the following.
  (1) Encode $\mathbf{x}_t \in \{0,1\}^l$ into the codeword $C(\mathbf{x}_t) \in \{0,1\}^m$ employing the selected ECC suitable for a binary symmetric channel with the crossover probability η.
  (2) Employing the output vector $\mathbf{y}_t \in \{0,1\}^m$ from the keystream generator compute $C(\mathbf{x}_t) \oplus \mathbf{y}_t$, where $\oplus$ denotes bit-by-bit $mod 2$ addition.

(3) Generate by the RAND-box a random vector $\vec{\rho_t} \leftarrow \{0,1\}^{n-m}$ and perform psudorandom embedding (controlled by the keystream generator) of the bits from the vectors $C(\mathbf{x}_t) \oplus \mathbf{y}_t$ and $\vec{\rho}$ as follows: $(C(\mathbf{x}_t) \oplus \mathbf{y}_t || \vec{\rho_t})\mathbf{P}_t$, where $\mathbf{P}_t$ is an $n \times n$ permutation matrix which corresponds to the considered embedding and $||$ denotes the concatenation.

(4) Generate by the RAND-box a random $\vec{\nu_t} \leftarrow \mathrm{Ber}_{n,\eta}$ and generate the ciphertext vector as follows:

$$\mathbf{z}_t = (C(\mathbf{x}_t) \oplus \mathbf{y}_t || \vec{\rho_t})\mathbf{P}_t \oplus \vec{\nu_t} \ . \tag{4.1}$$

- *Output*: The ciphertext $\{\mathbf{z}_t\}_t$.

Decryption Algorithm

- *Input*: The ciphertext organized as a string of n-dimensional binary vectors $\{\mathbf{z}_t\}_t$, the secret key & non-secret initial vector which control the keystream generator, and the algorithm parameters m, n and η.
- *Decryption Steps*
  For each t do the following.
  (1) Perform decimation of $\mathbf{z}_t$ corresponding to the embedding performed in the encryption step 3 as follows:

$$\mathbf{z}_t\mathbf{P}_t^{-1} = (C(\mathbf{x}_t) \oplus \mathbf{y}_t || \vec{\rho_t}) \oplus (\vec{\nu_t}\mathbf{P}_t^{-1}),$$
$$tcat_m(\mathbf{z}_t\mathbf{P}_t^{-1}) = C(\mathbf{x}_t) \oplus \mathbf{y}_t \oplus tcat_m(\vec{\nu_t}\mathbf{P}_t^{-1}),$$

  where $\mathbf{P}_t^{-1}$ denotes the inverse permutation of $\mathbf{P}_t$ (which is the transpose of $\mathbf{P}_t$), and $tcat_m(\cdot)$ denotes the truncating of the argument to the first m bits.

  (2) Employing the output vector $\mathbf{y}_t \in \{0,1\}^m$ from the keystream generator compute

$$tcat_m(\mathbf{z}_t\mathbf{P}_t^{-1}) \oplus \mathbf{y}_t = C(\mathbf{x}_t) \oplus tcat_m(\vec{\nu_t}\mathbf{P}_t^{-1})$$

  (3) Perform decoding $C^{-1}(\cdot)$ according to the employed ECC and recover $\mathbf{x}_t$ as follows:

$$\mathbf{x}_t = C^{-1}(C(\mathbf{x}_t) \oplus tcat_m(\vec{\nu_t}\mathbf{P}_t^{-1}))$$

- *Output*: The message in the form of the string $\{\mathbf{x}_t\}_t$.

Regarding the employed ECC we assume the following. It should be such that provides reliable decoding for the given parameter η and characteristics of the public communication channel. In the scenarios when the public communication channel is noiseless and the employed ECC is an $[m, l]$ binary block code with the decoding capability of correcting up to d errors, the lower bound on the probability of correct decoding $P(m, \eta)$ is determined by the following:

$$P(m,\eta) \geqslant \sum_{i=0}^{d} \binom{m}{i} \eta^i (1-\eta)^{m-i}.$$

Assuming that the probability of the acceptable decoding error is ϵ, the employed ECC $[m, l]$ should be such that $\epsilon \leqslant 1 - P(m, \eta)$. Finally note that in details discussion of suitable ECC selection is out of the scope of this paper.

An Equivalent Representation and a Particular Instantiation of the Proposed Stream Ciphers Family The following statement points out an equivalent analytical representation of the encryption algorithm given in the previous section which is suitable for specification of a particular and illustrative instantiation of the proposed family of stream ciphers.

Equivalent Representation

Proposition 4.1. *An equivalent analytical expression of the encryption specified by* (4.1) *is given by the following:*

$$\mathbf{z}_t = (C(\mathbf{x}_t)||\vec{\rho_t})\mathbf{P}_t \oplus (\mathbf{y}_t||\mathbf{0}^{n-m})\mathbf{P}_t \oplus \vec{\nu_t}, \tag{4.2}$$

where $\mathbf{0}^{n-m}$ *denotes the all zeros* $(n-m)$*-dimensional vector.*

Proof. We have

$$\begin{aligned}(C(\mathbf{x}_t) \oplus \mathbf{y}_t||\vec{\rho_t})\mathbf{P}_t \oplus \vec{\nu_t} &= (C(\mathbf{x}_t) \oplus \mathbf{y}_t||\mathbf{0}^{n-m})\mathbf{P}_t \oplus (\mathbf{0}^m||\vec{\rho_t})\mathbf{P}_t \oplus \vec{\nu_t} \\ &= (C(\mathbf{x}_t)||\mathbf{0}^{n-m})\mathbf{P}_t \oplus (\mathbf{y}_t||\mathbf{0}^{n-m})\mathbf{P}_t \oplus (\mathbf{0}^m||\vec{\rho_t})\mathbf{P}_t \oplus \vec{\nu_t} \\ &= (C(\mathbf{x}_t) \oplus \mathbf{0}^m||\mathbf{0}^{n-m} \oplus \vec{\rho_t})\mathbf{P}_t \oplus (\mathbf{y}_t||\mathbf{0}^{n-m})\mathbf{P}_t \oplus \vec{\nu_t},\end{aligned}$$

which implies the proposition statement. □

Particular Instantiation. According to the encryption and decryption algorithms and Proposition 4.1, an instantiation of the proposed stream cipher framework is specified by the following definition.

Definition 4.1. Let $\mathbf{S}$ be a secret $k \times n$ binary matrix, and $\mathbf{P}_0$ be a secret $n \times n$ secret permutation matrix. Let $\mathbf{a}_t$ be a k-dimensional random vector which is publicly available, $t = 1, 2, \ldots$. Finally, let $\mathbf{P}_t = f(\mathbf{a_t}, \mathbf{P}_{t-1})$, where $f(\cdot)$ is a suitably selected function. For $t = 1, 2, \ldots$, encryption of $\mathbf{x}_t$ into $\mathbf{z}_t$ is

$$\mathbf{z}_t = (C(\mathbf{x}_t)||\vec{\rho_t})\mathbf{P}_t \oplus \mathbf{a}_t \cdot \mathbf{S} \oplus \vec{\nu_t},$$

and accordingly, decryption of $\mathbf{z}_t$ into $\mathbf{x}_t$ is as follows:

$$\mathbf{x}_t = C^{-1}(tcat_m((\mathbf{z}_t \oplus \mathbf{a}_t \cdot \mathbf{S})\mathbf{P}_t^{-1}) \; . \tag{4.3}$$

4.2.5. A Preliminary Security Evaluation of the Proposed Framework. This section yields a preliminary and informal discussion on the security of the proposed stream ciphers framework which points out to the security origins.

The role of the employed homophonic encoding, implemented via the random bits embedding, is to provide a heavy masking of the keystream generator sequences so that they appear as very uncertain for a given ciphertext even when the plaintext is known.

The proposed paradigm for providing the security is based on the following: (i) impossibility of correct decimation i.e. splitting of the effective from the dummy bits of the ciphertext without the secret key; and (ii) availability of the noisy sample only, due to the employed additive noise degradation of the ciphertext before its transmission via a public communications channel.

The main role of the additive random degradation of the ciphertext is to introduce uncertainty into a sample available for cryptanalysis preventing a possibility

of mounting the generic time-memory trade-off approaches for cryptanalysis (see [84] and [69]) in order to employ a generic approach more efficient than the exhaustive search. When an error-free sample is available the time-memory (and time-memory-data) trade-off based attacks can be directly mounted in order to recover the secret key $\mathbf{K}$. On the other hand, when the sample for cryptanalysis is not error-free, the time-memory trade-off approach, in a general case, does not work.

The above arguments are a background for the security evaluation of the proposed framework and for a conjecture that the complexity of cryptanalysis is determined by the complexity of exhaustive secret keys search.

Note that the following are basic approaches for cryptanalysis of any stream cipher: (i) the generic key recovery attacks based on different search techniques (including the trade-off ones); (ii) the dedicated key recovery attacks based on particular weaknesses of the underlying structure; (iii) a number of different not key recovery oriented attacks (distinguishing attacks, ...).

In a known plaintext attack scenario, the goal of cryptanalysis is to recover the key $\mathbf{K}$. There are the following two basic approaches for achieving this goal:

–recovering $\mathbf{K}$ based on the given ciphertext $\{\mathbf{z}_t\}_t$,

–recovering certain pesudorandom sequences specified by $\mathbf{K}$ based on $\{\mathbf{z}_t\}_t$ and then recovering $\mathbf{K}$ based on these sequences.

For achieving any of these goals, an attacker faces the following two main problems:

- the inverse mapping without knowledge of the secret key in order to recover the considered pseudorandom sequences based on $\{\mathbf{z}_t\}_t$;
- impact of the noise sequence $\{\vec{\nu}_t\}_t$ to complexity of any generic technique for recovering the secret key beside the exhaustive search over the space of all possible keys which has complexity $O(2^K)$.

Hardness of the above problems is elaborated by the following.

Note that even in the case of a noiseless public communication channel, there are the following two problems at the attacker's side:

–removing the dummy bits from the ciphertext without knowledge of the secret key;

–uncertainty due to effect of the binary symmetric channel with crossover probability $p^* < 1/2$ which corrupts the data before theirs availability to the attacker.

The uncertainty at the attacker's side can be considered as a consequence of a noise corresponding to a channel with the bits insertion and complementing which corrupts the sample for cryptanalysis. Because, the legitimate parties share the secret key, they face a lower noise (corresponding only to the bits complementing) in comparison with the noise which an attacker faces.

Accordingly, security of the scheme appears as consequence of the employed wiretap channel like encoding which provides confusion of an attacker which faces much more heavy equivalent noise in comparison with the legitimate receiver because the attacker does not posses the employed secret key. This heavy noise implies that the attacker can not learn about the keystream generator output sequences.

Without knowledge about the employed secret, it is not possible to efficiently remove dummy bits and to learn about (noisy) sequences from the keystream generator. On the other hand, without reliable knowledge on the keystream generator output sequences, it is not possible to construct a more efficient approach for cryptanalysis than a hypothesis testing. Accordingly, the corruption of the output sequences by the noise $\{\vec{\nu}_t\}_t$ implies (as discussed above) that the time-memory trade-off hypotheses testing based attacks are not feasible because the entire system appears as a stochastic one which makes the algebraic approaches not feasible. So, the exhaustive search over the space of all possible keys appears as the only one option.

The above discussion implies that the security appears as a consequence of the uncertainty at the attacker's side which is jointly implied by: (i) pseudorandom homophonic encoding; (ii) effect of the intentional corruption of the data which are available only via a binary symmetric channel.

4.2.6. A Formal Security Evaluation of the Particular Instantiation. This section yields a formal security evaluation of the stream ciphers specified by Definition 4.1.

Security Evaluation Background. One of the security goals is the indistinguishability (IND): IND deals with the secrecy provided by the scheme in the following sense: An adversary must be unable to distinguish the encryption of two (chosen) plaintexts. This definition was introduced in the context of public-key encryption as a more practical equivalent to semantic security and recently employed for security evaluation of the schemes reported in [66] and [81]. Accordingly, and particularly following [81], this paper adopts IND as the security criterion for a formal security consideration. For the IND considerations we assume the following traditional approach. An adversary is considered as a pair of algorithms $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and they operate through two phases as follows.

- $\mathcal{A}_1$ is employed during the first phase and at the end of this phase, $\mathcal{A}_1$ outputs a pair of plaintexts $(\mathbf{x}_1, \mathbf{x}_2)$.
- One of the given plaintexts is selected with probability equal 1/2, then encrypted, and the obtained ciphertext is delivered to $\mathcal{A}_2$-this represents $\mathcal{A}$'s challenge. The success of $\mathcal{A}$ is determined according to correctness of decision whether $\mathbf{x}_1$ or $\mathbf{x}_2$ was encrypted.

The adversary $\mathcal{A}$ is classified according to the oracles (encryption and/or decryption) available in each of the phases. $\mathcal{A}$ is labelled as PX-CY, where P stands for the encryption oracle and C for the decryption oracle, and where $X, Y \in 0, 1, 2$ indicates when $\mathcal{A}$ is allowed to access the oracle:

–0: $\mathcal{A}$ never accesses the oracle;

–1: $\mathcal{A}$ can only access the oracle during phase 1, i.e. before seeing the challenge (also termed non-adaptive);

–2: $\mathcal{A}$ can access the oracle during phases 1 and 2 (also termed adaptive).

The following lemma states that the hardness of the LPN problem implies that the two oracles $\mathcal{U}_{k+1}$ and $\Pi_{s,\eta}$ are indistinguishable.

Lemma 4.1. [90, Lemma 1]. *Assume there exists an algorithm $\mathcal{M}$ making q oracle queries, running in time T, and such that*

$$\left|\Pr\left[\mathbf{s}\leftarrow\{0,1\}^k:\mathcal{M}^{\Pi_{s,\eta}}(1^k)=1\right]-\Pr\left[\mathcal{M}^{\mathcal{U}_{k+1}}(1^k)=1\right]\right|\geqslant\delta.$$

Then there is an algorithm $\mathcal{A}$ making $q'=O(q\cdot\delta^{-2}\log k)$ oracle queries, running in time $T'=O(T\cdot k\delta^{-2}\log k)$, and such that

$$\Pr\left[\mathbf{s}\leftarrow\{0,1\}^k:\mathcal{A}^{\Pi_{s,\eta}}(1^k)=\mathbf{s}\right]\geqslant\frac{\delta}{4}.$$

An Analysis of the Security. This section yields a method for reducing security evaluation of the stream ciphers specified by Definition 4.1 to the problem of distinguishing $\mathcal{U}_{k+1}$ and $\Pi_{s,\eta}$ and according to Lemma 4.1 further on to the LPN problem.

Theorem 4.1. *Assume there is an adversary $\mathcal{A}$, running in time T, and attacking the stream cipher specified by Definition* 1 *with parameters (l,m,k,n,η) in the sense of IND-P1-C0 with advantage δ by making at most q queries to the encryption oracle. Then there is an algorithm $\mathcal{M}$ making $O(q)$ oracle queries, running in time $O(T)$, and such that*

$$\left|\Pr\left[\mathbf{s}\leftarrow\{0,1\}^k:\mathcal{M}^{\Pi_{s,\eta}}(1^k)=1\right]-\Pr\left[\mathcal{M}^{\mathcal{U}_{k+1}}(1^k)=1\right]\right|\geqslant\frac{\delta}{n}.$$

Proof. The proof is an adaptation of the proof technique reported in [81, Theorem 1]. Particularly note that non-adaptive CPA-security (P1) implies adaptive CPA-security (P2), and so it is possible to restrict the consideration to adversaries accessing the encryption oracle only during the first phase of the attack i.e. before seeing the challenge ciphertext (see [81], for example). □

In the following, the same notation as in the previous sections is used, but the index t has been omitted for the simplicity.

The proof proceeds by a hybrid argument based on the following hybrid distributions on $\{0,1\}^{k+n}$. For $j\in[0,\ldots,n]$, let $\mathbf{S}'$ denotes a $k\times(n-j)$ binary matrix. We define the probability distribution $\mathcal{P}_{j,S',\eta}$ as

$$\left\{\mathbf{a}\leftarrow\{0,1\}^k\,;\,\mathbf{r}\leftarrow\{0,1\}^j\,;\,\vec{\nu}\leftarrow\mathrm{Ber}_{(n-j),\eta}:\mathbf{a}||\mathbf{r}||(\mathbf{a}\cdot\mathbf{S}'\oplus\vec{\nu})\right\}.$$

Accordingly, we obtain the vector $\mathbf{a}||\mathbf{b}$ such that the first j bits of $\mathbf{b}=\mathbf{r}||(\mathbf{a}\cdot\mathbf{S}'\oplus\vec{\nu})$ are uniformly random, whereas the last $(n-j)$ bits are distributed according to the $(n-j)$ independent LPN distributions related to the respective columns of $\mathbf{S}'$.

Note that $\mathcal{P}_{n,S',\eta}$ corresponds to $\mathcal{U}_{k+n}$.

The next step is specification of the following hybrid encryption oracles $\mathcal{E}_{j,S',\eta}$ associated with the secret matrix $\mathbf{S}'$ and noise parameter η:

- On input the l-bit plaintext $\mathbf{x}$, the encryption oracle performs a homophonic encoding and maps it to $[C(\mathbf{x})||\tilde{\rho}]\mathbf{P}$, draws a random $(k+n)$-bit vector $\mathbf{a}||\mathbf{b}$ distributed according to $\mathcal{P}_{j,S',\eta}$, and returns $(\mathbf{a},[C(\mathbf{x})||\tilde{\rho}]\mathbf{P}\oplus\mathbf{b})$.

Recall that $\mathcal{M}$ has access to an oracle and wants to distinguish whether this is $\mathcal{U}_{k+1}$ or $\Pi_{s,\eta}$. In order to achieve its goal, the distinguisher $\mathcal{M}$ acts performing the following steps.

(1) On input the security parameter 1^k, $\mathcal{M}$ draws a random $j \in [1, \ldots, n]$. If $j < n$, it also draws a random $k \times (n-j)$ binary matrix $\mathbf{S}'$. It then launches the first phase $\mathcal{A}_1$ of the adversary $\mathcal{A}$.
(2) Each time $\mathcal{A}_1$ asks for the encryption of some $\mathbf{x}$, $\mathcal{M}$ obtains a sample $(\mathbf{a}, z)$ from its oracle, and performs the following:
   - draws a random $(j-1)$-bit vector $\mathbf{r} \leftarrow \{0,1\}^{j-1}$;
   - draws a $(m-j)$-bit noise vector $\vec{\nu}$ distributed according $\mathrm{Ber}_{n-j,\eta}$;
   - forms the masking vector $\mathbf{b} = \mathbf{r}||z||(\mathbf{a} \cdot \mathbf{S}' \oplus \vec{\nu}$, and returns $(\mathbf{a}, [C(\mathbf{x})||\tilde{\rho}]\mathbf{P} \oplus \mathbf{b})$.
(3) –The adversary $\mathcal{A}_1$ returns two plaintexts $\mathbf{x}_1$ and $\mathbf{x}_2$.
   –The distinguisher $\mathcal{M}$ selects a uniformly random $\alpha \in 1, 2$ and returns to $\mathcal{A}_2$ the ciphertext corresponding to $\mathbf{x}_\alpha$ encrypted exactly as described just before.
   –If the answer of $\mathcal{A}_2$ is correct, then $\mathcal{M}$ returns 1, otherwise it returns 0.

It is straightforward to verify the following

- when $\mathcal{M}$'s oracle is $\mathcal{U}_{k+1}$, $\mathcal{M}$ simulates the encryption oracle $\mathcal{E}'_{j,S',\eta}$, and
- when $\mathcal{M}$'s oracle is $\Pi_{s,\eta}$, then $\mathcal{M}$ simulates the encryption oracle $\mathcal{E}'_{j-1,S'',\eta}$ where $\mathbf{S}'' = \mathbf{s}||\mathbf{S}'$ is the matrix obtained as the concatenation of $\mathbf{s}$ and $\mathbf{S}'$.

So, the advantage of the distinguisher can be expressed as follows:

$$\begin{aligned}
\mathrm{Adv} &= \left|\Pr\left[\mathbf{s} \leftarrow \{0,1\}^k : \mathcal{M}^{\Pi_{s,\eta}}(1^k) = 1\right] - \Pr\left[\mathcal{M}^{\mathcal{U}_{k+1}}(1^k) = 1\right]\right| \\
&= \frac{1}{n}\left|\sum_{j=0}^{n-1} \Pr\left[\mathcal{A}^{\mathcal{E}'_{j,S',\eta}} \text{ succeeds}\right] - \sum_{j=1}^{n} \Pr\left[\mathcal{A}^{\mathcal{E}'_{j,S',\eta}} \text{ succeeds}\right]\right| \\
&= \frac{1}{n}\left|\Pr\left[\mathcal{A}^{\mathcal{E}'_{0,S',\eta}} \text{ succeeds}\right] - \Pr\left[\mathcal{A}^{\mathcal{E}'_{n,S',\eta}} \text{ succeeds}\right]\right|
\end{aligned}$$

Note that the encryption oracle $\mathcal{E}'_{0,S',\eta}$ is exactly the real encryption oracle.

On the other hand the encryption oracle $\mathcal{E}'_{n,M',\eta}$ encrypts all plaintexts by blinding them with uniformly random vectors $\mathbf{b}$ so that in this case the adversary $\mathcal{A}$ cannot do better (or worse) than guessing at random and has the success probability of 1/2. Accordingly, $\left|\Pr\left[\mathcal{A}^{\mathcal{E}'_{0,S',\eta}} \text{ succeeds}\right] - \Pr\left[\mathcal{A}^{\mathcal{E}'_{n,S',\eta}} \text{ succeeds}\right]\right|$ is exactly the advantage of the adversary which is greater than δ by the hypothesis, implying the theorem statement.

4.2.7. Implementation Complexity and Communications Overhead. This section yields a brief discussion on the implementation complexity and the communication overhead of the proposed framework for stream ciphers and it points out the main issues only. An in details consideration of the implementation complexity and the communications overhead requires focusing on particular instantiations of the proposed class of stream ciphers and it is out of the scope of this paper.

Complexity. The implementation complexity, in comparison with a traditional stream ciphering scheme which includes the error-correction coding is mainly due to requirement for the source of randomness (RAND-box) because the embedding and decimation operations could be considered as low complexity ones.

The implementation complexity of the additional components depends on the overall implementation scenario and particularly whether it is software only or a hybrid one. Assuming availability of a suitable RAND-box the dominant implementation complexity overhead appears as a very low one.

Overhead. In order to achieve the main security goal, the proposed stream ciphering approach includes the following processing with impacts on the communications overhead: (i) error-correction encoding of the messages; (ii) a homophonic encoding via random bits embedding which performs expansion of the "initial ciphertext". Both of these issues imply the communications overhead: Assuming that the error-correction encoding and the embedding introduce the expansion for the factors α_1 and α_2, respectively, the related communications overhead is determined by the factor $\alpha_1 \cdot \alpha_2$.

Accordingly, the proposed stream ciphers framework includes certain trade-off between the security and the communications overhead which in a number of scenarios can be considered as very appropriate.

4.2.8. Concluding Notes. This section proposes an alternative approach for design of stream ciphers which involve pure randomness and provide low-complexity of the implementation. The proposed framework employs a dedicated homophonic coding and a deliberate noise which, assuming the appropriate code and noise level provides at the attacker's side increased confusion close to the limit determined by the secret key length. The employed homophonic encoding/decoding is based on pseudorandom embedding/decimation of random bits, and it is specific in the following sense: (i) its only purpose is to introduce additional uncertainty at the attackers side, and (ii) decoding complexities with and without the secret key are extremely different. Generic encryption/decryption algorithms are proposed, an equivalent interpretation, and a particular family of stream ciphers.

Security evaluation implies that the proposed stream ciphering provides high security which can be very close to the maximum one indicated by the employed secret key length. Consequently, under certain conditions, a straightforward exhaustive search over all possible secret keys appears as very close to the most efficient method of cryptanalysis. For a particular family of the proposed stream ciphers it is formally shown that the security appears as a consequence of hardness of the LPN problem.

In order to achieve the main security goal, the proposed stream ciphering approach includes the following two encoding schemes with impacts on the communications overhead: (i) error-correction encoding of the messages; (ii) dedicated homophonic encoding via the random bits embedding which performs expansion of the initial ciphertext. Both of these issues imply the communications overhead: Accordingly, the proposed stream ciphers framework includes certain trade-off between the security and the communications overhead which, in a number of scenarios, can be considered as appropriate.

4.3. A Generic Framework of Randomized Stream Ciphers and Its Security Evaluation.

Following the encryption approaches recently reported in [4] and [61],

this section considers and analyzes from security point of view a generic model of randomized stream ciphers.

The section yields an analysis of security of a model of randomized stream cipher based on joint employment of pseudorandomness, randomness and dedicated coding. The considered scheme sequentially encrypts l-bit plaintext vectors into n-bit ciphertext vectors employing a keystream generator seeded by k-bit secret key, $m-l$, $l < m < n$, balanced random bits where ones and zeros appear with the same probability equal to $1/2$, n biased random bits where ones appear with the probability $p < 1/2$, and two linear encoding schemes for dedicated homophonic and error correction encoding. The security analysis has been performed assuming the chosen plaintext attack. The information-theoretic security evaluation was focussed towards the posterior uncertainty on the secret key. The equivocation of the secret key has been derived and analyzed. The equivocation expression shows that it can be kept to a nonzero value assuming appropriate selection of the encryption parameters $m-l$, n and p, when the sample available for cryptanalysis is limited. The previous imply that the scheme has potential of providing residual uncertainty on the secret key under certain conditions. Also, the considered encryption scheme is analyzed from computational complexity security point of view. The performed evaluation of the secret key recovery implies that it is as hard as decoding of a random linear block code after a binary symmetric channel with the additive noise (cross-over probability) parameter ϵ equal to $\frac{1}{2}\big(1-(1-2p)^{(m-l)/2}\big)$. The analysis performed imply that the considered encryption paradigm provides a framework for design of provably secure stream ciphers which can provide low implementation complexity as well (noting that the implementation issues are out of the scope of this chapter).

4.3.1. Framework of Certain Randomized Stream Ciphers. We consider the randomized stream ciphers framework displayed in the following figure.

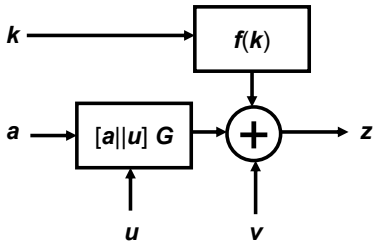

FIGURE 2. A generic randomized stream cipher encryption.

The analytical description of the considered encryption is as follows. Let:
–$\mathbf{a}^{(l)}$ is a known l-dimensional binary vector;
–$\mathbf{G}$ is a known $(m \times n)$-dimensional binary matrix such that $\mathbf{G} = \mathbf{G}_H\,\mathbf{G}_{ECC}$ where $\mathbf{G}_H$ is the matrix of a linear homophonic encoding, and $\mathbf{G}_{ECC}$ is the generator matrix of a linear error-correcting code (ECC) designed to correct errors over a

binary symmetric channel (b.s.c.) with the crossover probability p;
–$\mathbf{u}^{(m-l)}$ is a realization of $(m-l)$-dimensional binary random variable $\mathbf{U}^{(m-l)}$ such that $\Pr(\mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)}) = \frac{1}{2^{m-l}}$;
–$\mathbf{v}^{(n)}$ is a realization of n-dimensional binary random variable $\mathbf{V}^{(n)}$ such that $\Pr(\mathbf{V}^{(n)} = \mathbf{v}^{(n)}) = p^w(1-p)^{n-w}$, $p < 0.5$, and $w = Hwt(\mathbf{v}^{(n)})$, and $Hwt(\cdot)$ denotes the Hamming weight.

Accordingly, we have the following algebraic representation of the ciphertext:

$$\mathbf{z}^{(n)} = [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G} \oplus f^{(n)}(\mathbf{k}) \oplus \mathbf{v}^{(\mathbf{n})}.$$

The corresponding decryption process is as follows:

$$\mathbf{a}^{(l)} = tcat\{[ECC^{-1}(\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}))]\mathbf{G}_H^{-1}\},$$

where $tcat\{\cdot\}$ is the operator of truncation to the first l bits, $ECC^{-1}(\cdot)$ denotes the decoding operator of the ECC with the generator matrix $\mathbf{G}_{ECC}$, and $\mathbf{G}_H^{-1}$ is the inverse matrix of $\mathbf{G}_H$.

4.3.2. Encryption of a Sequence of Vectors. We consider encryption of a sequence of vectors at the time instances $t = 1, 2, \ldots, \tau$, employing the following notation:
–$\mathbf{a}_t^{(l)}$ is a known l-dimensional binary vector at the time instance t;
–$f_t^{(t)}(\mathbf{k})$ is the keystream generator output segment of length n generated at the time instance t;
–$\mathbf{u}_t^{(m-l)}$ is a realization of $(m-l)$-dimensional binary random variable $\mathbf{U}_t^{(m-l)}$ such that $\Pr(\mathbf{U}_t^{(m-l)} = \mathbf{u}_t^{(m-l)}) = 2^{-m+l}$, at the time instance t;
–$\mathbf{v}_t^{(n)}$ is a realization of n-dimensional binary random variable $\mathbf{V}_t^{(n)}$ at the time instance t such that $\Pr(\mathbf{V}_t^{(n)} = \mathbf{v}_t^{(n)}) = p^{w_t}(1-p)^{n-w_t}$, $p < 0.5$, and $w_t = Hwt(\mathbf{v}_t^{(n)})$ denotes the Hamming weight of the vector $\mathbf{v}_t^{(n)}$.

Accordingly, the ciphertext vectors $\mathbf{z}_t^{(n)}$, $t = 1, 2, \ldots, \tau$, are specified by the following:

$$\mathbf{z}_t^{(n)} = [\mathbf{a}_t^{(l)}||\mathbf{u}_t^{(m-l)}]\mathbf{G} \oplus f_t^{(n)}(\mathbf{k}) \oplus \mathbf{v}_t^{(n)}, \quad t = 1, 2, \ldots, \tau.$$

4.3.3. Information-Theoretic Security Evaluation of a Single Encryption: On the Equivocation. This section consider the uncertainty on the secret key when a corresponding keystream generator output segment is known. As the first, the posterior probability that certain key $\mathbf{k}$ has been involved into generation of a keystream segment $\mathbf{z}^{(n)}$ is given and finally the equivocation[1] (which specifies the posterior uncertainty) is derived.

Lemma 4.2. *We have*

$$\Pr(\mathbf{K} = \mathbf{k}|\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) = \frac{\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w}}{\sum_{\mathbf{k}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w}},$$

[1]see [104] or a textbook on information theory

where $\alpha_{\mathbf{k},\mathbf{z}}(w)$ *is the number of different vectors* $\mathbf{u}^{(m-l)}$ *which, for given* $\mathbf{k}$ *and* $\mathbf{z}^{(n)}$ *imply the same* $w = Hwt(\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G})$*, and* $\sum_{\mathbf{k}}(\cdot)$ *denotes summation over all possible keys, assuming that* $\mathbf{a}^{(l)}$ *is known.*

Proof. We have

$$\Pr(\mathbf{K}=\mathbf{k}|\mathbf{Z}^{(n)}=\mathbf{z}^{(n)}) = \frac{\Pr(\mathbf{Z}^{(n)}=\mathbf{z}^{(n)}|\mathbf{K}=\mathbf{k})\,\Pr(\mathbf{K}=\mathbf{k})}{\sum_{\mathbf{k}}\Pr(\mathbf{Z}^{(n)}=\mathbf{z}^{(n)}|\mathbf{K}=\mathbf{k})\,\Pr(\mathbf{K}=\mathbf{k})},$$

and when all the keys are equiprobable

$$\Pr(\mathbf{K}=\mathbf{k}|\mathbf{Z}^{(n)}=\mathbf{z}^{(n)}) = \frac{\Pr(\mathbf{Z}^{(n)}=\mathbf{z}^{(n)}|\mathbf{K}=\mathbf{k})}{\sum_{\mathbf{k}}\Pr(\mathbf{Z}^{(n)}=\mathbf{z}^{(n)}|\mathbf{K}=\mathbf{k})},$$

On the other hand we have the following

$$\begin{aligned}
\Pr(\mathbf{Z}^{(n)}=\mathbf{z}^{(n)}|\mathbf{K}=\mathbf{k}) &= \frac{\Pr(\mathbf{Z}^{(n)}=\mathbf{z}^{(n)},\mathbf{K}=\mathbf{k})}{\Pr(\mathbf{K}=\mathbf{k})} \\
&= \frac{1}{\Pr(\mathbf{K}=\mathbf{k})}\sum_{\mathbf{u}^{(m-l)}}\Pr(\mathbf{Z}^{(n)}=\mathbf{z}^{(n)},\mathbf{K}=\mathbf{k},\mathbf{U}^{(m-l)}=\mathbf{u}^{(m-l)}) \\
&= \frac{1}{\Pr(\mathbf{K}=\mathbf{k})}\sum_{\mathbf{u}^{(m-l)}}(\Pr(\mathbf{Z}^{(n)}=\mathbf{z}^{(n)}|\mathbf{K}=\mathbf{k},\mathbf{U}^{(m-l)}=\mathbf{u}^{(m-l)}) \\
&\qquad\qquad \cdot \Pr(\mathbf{U}^{(m-l)}=\mathbf{u}^{(m-l)}|\mathbf{K}=\mathbf{k})\,\Pr(\mathbf{K}=\mathbf{k})) \\
&= \sum_{\mathbf{u}^{(m-l)}}\Pr(\mathbf{Z}^{(n)}=\mathbf{z}^{(n)}|\mathbf{K}=\mathbf{k},\mathbf{U}^{(m-l)}=\mathbf{u}^{(m-l)})\,\Pr(\mathbf{U}^{(m-l)}=\mathbf{u}^{(m-l)})\,.
\end{aligned}$$

Further on:

$$\begin{aligned}
&\Pr(\mathbf{Z}^{(n)}=\mathbf{z}^{(n)}|\mathbf{K}=\mathbf{k},\,\mathbf{U}^{(m-l)}=\mathbf{u}^{(m-l)}) \\
&\qquad\qquad = \Pr(\mathbf{V}^{(n)}=\mathbf{v}^{(n)}=\mathbf{z}^{(n)}\oplus f^{(n)}(\mathbf{k})\oplus[\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G})
\end{aligned}$$

and accordingly

$$\begin{aligned}
&\Pr(\mathbf{Z}^{(n)}=\mathbf{z}^{(n)}|\mathbf{K}=\mathbf{k}) \qquad (4.4)\\
&= \sum_{\mathbf{u}^{(m-l)}}\Pr(\mathbf{U}^{(m-l)}=\mathbf{u}^{(m-l)})\,\Pr(\mathbf{V}^{(n)}=\mathbf{v}^{(n)}=\mathbf{z}^{(n)}\oplus f^{(n)}(\mathbf{k})\oplus[\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G}) \\
&= \frac{1}{2^{m-l}}\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}, \quad w = Hwt(\mathbf{z}^{(n)}\oplus f^{(n)}(\mathbf{k})\oplus[\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G}),
\end{aligned}$$

and $\alpha_{\mathbf{k},\mathbf{z}}(w)$ is the number of different vectors $\mathbf{u}^{(m-l)}$ which, for given $\mathbf{k}$ and $\mathbf{z}$, yield the same w. The above imply the lemma statement. □

Corollary 4.1. *According to (4.4), when* $\mathbf{a}^{(l)}$ *is known, we have the following:*

- *when* $p = 0$,

$$\Pr(\mathbf{Z}^{(n)}=\mathbf{z}^{(n)}|\mathbf{K}=\mathbf{k})$$

$$= \begin{cases} \frac{1}{2^{m-l}}\alpha_{\mathbf{k},\mathbf{z}}(0) = 2^{-(m-l)} & \text{if } \mathbf{z} = [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G} \oplus f^{(n)}(\mathbf{k}) \\ 0 & \text{otherwise} \end{cases}$$

- *when* $p = 1/2$,

$$\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}|\mathbf{K} = \mathbf{k}) = \frac{1}{2^{m-l}}\frac{1}{2^n}\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w) = \frac{1}{2^{m-l}}\frac{1}{2^n}2^{m-l} = 2^{-n}.$$

Lemma 4.3. *The coefficients* $\{\alpha(w)\}_{w=0}^{n}$ *corresponds to the weight distribution of a modified linear block code with the generator matrix* $\mathbf{G}$.

Sketch of the Proof. Let a binary code C be specified by the generator matrix $\mathbf{G}$ of dimension $m \times n$. Each vector $[\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G}$ is a codeword of C. When $\mathbf{a}^{(l)}$ is a vector constant, all the codewords are specified by the set of all possible 2^{m-l} vectors $\mathbf{u}^{(m-l)}$. Let a modified code C' be obtained from C via $mod2$ addition of the codewords of C with a given n-dimensional constant vector $c^{(n)}$. Accordingly, $\{\alpha(w)\}_{w=1}^{n}$ specifies the weight distribution of C' if each $\alpha(w)$ is equal to number of the codeword with Hamming weight equal to w, namely $Hwt(\mathbf{c}^{(n)} \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G}) = w$. □

Theorem 4.2. *The equivocation of secret key in the known plaintext attack scenario (when* $\mathbf{a}^{(l)}$ *is known) is given by the following:*

$$H(\mathbf{K}|\mathbf{Z}^{(n)}) \tag{4.5}$$

$$= 2^{-(|\mathbf{k}|+m-l)}\sum_{\mathbf{z}^{(n)}}\left(\sum_{\mathbf{k}}\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}\right)\cdot\log_2\sum_{\mathbf{k}}\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}$$

$$-2^{-(|\mathbf{k}|+m-l)}\sum_{\mathbf{z}^{(n)}}\sum_{\mathbf{k}}\left(\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}\right)\cdot\log_2(\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}),$$

where $|\mathbf{k}|$ *is length of the secret key* $\mathbf{k}$, $\alpha_{\mathbf{k},\mathbf{z}}(w) \geqslant 0$ *is the number of different vectors* $\mathbf{u}^{(m-l)}$ *which, for given* $\mathbf{k}$ *and* $\mathbf{z}^{(n)}$ *imply the same* $w = Hwt(\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G})$, *and* $\sum_{\mathbf{k}}(\cdot)$ *denotes summation over all possible keys.*

Proof. We have

$$H(\mathbf{K}|\mathbf{Z}^{(n)}) = E_{\mathbf{Z}^{(n)}}\{H(\mathbf{K}|\mathbf{z}^{(n)})\}$$

$$= \sum_{\mathbf{z}^{(n)}}\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)})\sum_{\mathbf{k}}\Pr(\mathbf{K} = \mathbf{k}|\mathbf{Z}^{(n)} = \mathbf{z}^{(n)})\log_2\frac{1}{\Pr(\mathbf{K} = \mathbf{k}|\mathbf{Z}^{(n)} = \mathbf{z}^{(n)})}$$

and employment of Lemma 4.2 yields

$$H(\mathbf{K}|\mathbf{Z}^{(n)}) = \sum_{\mathbf{z}^{(n)}}\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)})$$

$$\cdot\sum_{\mathbf{k}}\frac{\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}}{\sum_{\mathbf{k}}\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}}\log_2\frac{\sum_{\mathbf{k}}\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}}{\sum_{w=0}^{n}\alpha_{\mathbf{k},\mathbf{z}}(w)p^w(1-p)^{n-w}} \quad .$$

Via manipulations over the above expression we obtain the following:

$$H(\mathbf{K}|\mathbf{Z}^{(n)}) =$$

$$\sum_{\mathbf{z}^{(n)}} \Pr(\mathbf{Z}^{(n)}=\mathbf{z}^{(n)}) \frac{\log_2 \sum_{\mathbf{k}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w}}{\sum_{\mathbf{k}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w}} \sum_{\mathbf{k}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w}$$

$$+ \sum_{\mathbf{z}^{(n)}} \frac{\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)})}{\sum_{\mathbf{k}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w}}$$

$$\cdot \sum_{\mathbf{k}} \left(\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w} \right) \log_2 \frac{1}{\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w}} .$$

On the other hand, when all the keys are equally-probable

$$\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) = \sum_{\mathbf{k}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)} | \mathbf{K} = \mathbf{k}) \cdot \Pr(\mathbf{K} = \mathbf{k}),$$

and employing (4.4) we obtain

$$\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) = 2^{-(|\mathbf{k}|+m-l)} \sum_{\mathbf{k}} (\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w})$$

where $|\mathbf{k}|$ denotes the secret key length. Combining the above directly implies the theorem statement. □

Corollary 4.2. *When $p = 0$,*

$$H(\mathbf{K}|\mathbf{Z}^{(n)}) = \begin{cases} |\mathbf{k}| + m - l - n, & \text{if } n < |\mathbf{k}| + m - l \\ 0, & \text{otherwise} \end{cases}$$

noting that $n > m - l$.

When $p = 1/2$, $H(\mathbf{K}|\mathbf{Z}^{(n)}) = |\mathbf{k}|$.

Proof. When $p = 0$, note that

$$\sum_{\mathbf{k}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w} = \sum_{\mathbf{k}} \alpha_{\mathbf{k},\mathbf{z}}(0)$$

because only $p^0 = 1$ yields non-zero terms. On the other hand each $\alpha_{\mathbf{k},\mathbf{z}}(0) \in \{0, 1\}$, and

$$\sum_{\mathbf{k}} \alpha_{\mathbf{k},\mathbf{z}}(0) = \begin{cases} 2^{|\mathbf{k}|+m-l-n} & \text{if } n < |\mathbf{k}| + m - l \\ 1 & \text{if } n \geqslant |\mathbf{k}| + m - l \end{cases}$$

because a system of n equations with $|\mathbf{k}| + m - l > n$ unknowns has $2^{|\mathbf{k}|+m-l-n}$ equally likely solutions, noting as well that $m - l < n$. The above and Theorem 4.2 yields the corollary statement for $p = 0$.

When $p = 1/2$, note that $\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) = 2^{m-l}$ and so $\sum_{\mathbf{k}} \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) = 2^{|\mathbf{k}|+m-l}$. Accordingly, statement of Theorem 4.2 implies the corollary for $p = 1/2$. □

Additionally, we point out to the following. Let $\pi = \pi(\mathbf{K}|\mathbf{Z}^{(n)})$ denotes the minimal probability of error which corresponds to employment of the maximum a-posteriori probability (MAP) decision rule for recovering the secret key. In the considered setting, $\pi(\mathbf{K}|\mathbf{Z}^{(n)})$ is specified by the next statement.

Corollary 4.3. *When* $n > |\mathbf{k}| + m - l$,

$$\pi = 1 - 2^{-(|\mathbf{k}|+m-l)} \sum_{\mathbf{z}^{(n)}} \max_{\mathbf{k}} \Big\{ \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w} \Big\},$$

and

$$\pi = \begin{cases} 0 & \text{if } p = 0 \\ 1 - 2^{|\mathbf{k}|} & \text{if } p = 1/2 \end{cases}$$

where $\alpha_{\mathbf{k},\mathbf{z}}(w)$ *denotes the number of vectors* $\mathbf{u}^{(m-l)}$ *which for given* $\mathbf{a}^{(l)}$, $\mathbf{z}^{(n)}$ *and* $\mathbf{k}$ *yield* $\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G} = w$.

Proof. Taking into account the definition of π and the implication of the exhaustive search based minimum distance decoding paradigm, we have the following:

$$\pi = \sum_{\mathbf{z}^{(n)}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) \big[1 - \max_{\mathbf{k}} \Pr(\mathbf{K} = \mathbf{k} | \mathbf{Z}^{(n)} = \mathbf{z}^{(n)})\big]$$

$$= \sum_{\mathbf{z}^{(n)}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) \Bigg[1 - \max_{\mathbf{k}} \frac{\sum_{\mathbf{u}^{(m-l)}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)} | \mathbf{K} = \mathbf{k}, \mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)}) \cdot \Pr(\mathbf{K} = \mathbf{k}, \mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)})}{\Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)})} \Bigg],$$

$$= \sum_{\mathbf{z}^{(n)}} \Pr(\mathbf{Z}^{(n)} = \mathbf{z}^{(n)}) - \sum_{\mathbf{z}^{(n)}} \max_{\mathbf{k}} \Bigg\{ \sum_{\mathbf{u}^{(m-l)}} \Pr(\mathbf{Z}^{(n)} \oplus f^{(n)}(\mathbf{K}) \oplus [\mathbf{A}^{(l)}||\mathbf{U}^{(m-l)}]\mathbf{G} = \mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G} \mid \mathbf{K} = \mathbf{k}, \mathbf{U}^{(m-l)} = \mathbf{u}^{(m-l)}) \cdot \Pr(\mathbf{K} = \mathbf{k}) \cdot \Pr(\mathbf{U}^{(u-l)} = \mathbf{u}^{(m-l)}) \Bigg\},$$

where $\mathbf{a}^{(l)}$ is given.

The above implies

$$\pi = 1 - \sum_{\mathbf{z}^{(n)}} 2^{-(|\mathbf{k}|+m-l)} \max_{\mathbf{k}} \Big\{ \sum_{\mathbf{u}^{(m-l)}} p^w (1-p)^{n-w} \Big\},$$

where w is equal to the Hamming weight of the vector $\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G}$, and accordingly

$$\pi = 1 - 2^{-(|\mathbf{k}|+m-l)} \sum_{\mathbf{z}^{(n)}} \max_{\mathbf{k}} \Big\{ \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w} \Big\},$$

where $\alpha_{\mathbf{k},\mathbf{z}}(w)$ denotes the number of vectors $\mathbf{u}^{(m-l)}$ which for given $\mathbf{a}^{(l)}$, $\mathbf{z}^{(n)}$ and $\mathbf{k}$ yield $\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)}||\mathbf{u}^{(m-l)}]\mathbf{G} = w$, noting that $\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) = 2^{m-l}$.

When $p = 0$, only the sum $\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w}$ reduces to $\alpha_{\mathbf{k},\mathbf{z}}(0)$ taking into account that $p^0 = 1$ and $p^w = 0$, $w = 1, 2, \ldots, n$. On the other hand, for $n > |\mathbf{k}| + m - l$, we have $\alpha_{\mathbf{k},\mathbf{z}}(0) = 1$ because just one out of 2^{m-l} vectors provides $w = 0$. Accordingly, for $p = 0$, $\pi = 1 - \sum_{\mathbf{z}} 2^{|\mathbf{k}|+m-l} = 0$ because when $p = 0$ $\mathbf{z}^{(n)}$ runs over exactly $2^{|\mathbf{k}|+m-l}$ different patterns. When $p = 1/2$, for any $w = 0, 1, \ldots n$, $p^w (1-p)^{n-w} = 2^{-n}$ implying $\sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w} = 2^{m-l-n}$, and accordingly $\max_{\mathbf{k}}\{\cdot\} = 2^{m-l-n}$. Taking into account that when $p = 1/2$ the vector $\mathbf{z}^{(n)}$ can take 2^n different values we obtain $\pi = 1 - \sum_{\mathbf{z}^{(n)}} 2^{-(|\mathbf{k}|+m-l)} 2^{m-l-n} = 1 - 2^{-|\mathbf{k}|}$.

Note that the distribution of $\alpha_{\mathbf{k},\mathbf{z}}(w)$ is implied by the following: Assuming $n > |\mathbf{k}| + m - l$, note the following:

- When $\mathbf{z}^{(n)}$ is generated employing the key $\mathbf{k}$ and the vector $\mathbf{u}^{(m-l)}$ we have:

$$\mathbf{z}^{(n)} \oplus f^{(n)}(\mathbf{k}) \oplus [\mathbf{a}^{(l)} || \mathbf{u}^{(m-l)}]\mathbf{G} = \mathbf{v}^{(n)},$$

- When $\mathbf{z}^{(n)}$ is not generated employing the key $\mathbf{k}$ and the vector $\mathbf{u}^{(m-l)}$, the binary vector variable $\mathbf{Z}^{(n)} \oplus f^{(n)}(\mathbf{K}) \oplus [\mathbf{A}^{(l)} || \mathbf{U}^{(m-l)}]\mathbf{G}$ appears as a random one where each component of the vector takes values one and zero with the probability equal to 1/2. □

As a finalization of the above discussion we point out to the following statement.

Proposition 4.2. *Let i be an integer, $2 \leqslant i < 2^{|k|}$, such that $\frac{i-1}{i} \leqslant \pi \leqslant \frac{i}{i+1}$. Then the tight bounds on the equivocation are as follows:*

$$\log_2 i + i(i+1)\Big(\log_2 \frac{i+1}{i}\Big)\Big(\pi - \frac{i-1}{i}\Big) \leqslant H(\mathbf{K}|\mathbf{Z}^{(n)})$$
$$\leqslant \pi \log_2 \frac{1}{\pi} + (1-\pi)\log_2 \frac{1}{1-\pi} + \pi \log_2(2^{|k|} - 1)$$

where

$$\pi = 1 - 2^{-(|\mathbf{k}|+m-l)} \sum_{\mathbf{z}^{(n)}} \max_{\mathbf{k}} \Big\{ \sum_{w=0}^{n} \alpha_{\mathbf{k},\mathbf{z}}(w) p^w (1-p)^{n-w} \Big\}.$$

Proof. The Fano inequality [78] yields the following upper bound:

$$H(\mathbf{K}|\mathbf{Z}^{(n)}) \leqslant h(\pi) + \pi \log_2(2^{|\mathbf{k}|} - 1),$$
$$h(\pi) = \pi \log_2 \pi - (1-\pi)\log_2(1-\pi).$$

The following lower bound is implication of the general lower bound reported in [91]:

$$H(\mathbf{K}|\mathbf{Z}^{(n)}) \geqslant \log_2 i + i(i+1)(\log_2 \frac{i+1}{i})(\pi - \frac{i-1}{i})$$

where $2 \leqslant i < 2^{|k|}$, and $\frac{i-1}{i} \leqslant \pi \leqslant \frac{i}{i+1}$. Taking into account Corollary 4.3 which specifies π, we have the theorem statement. □

4.3.4. Computational Complexity Security Evaluation. This section analyzes the security of the proposed scheme from a computational complexity point of view in the chosen plaintext attacking (CPA) scenario. In this case, the security evaluation consists of establishing how hard it is to find the secret key based on the algebraic representation of the encryption. We will show in our complexity analysis that the hardness of recovering the key relies on the hardness of the LPN problem (see [71],[9], [92], for example). The analysis will pinpoint the features that the homophonic encoder should have so as to create an increased complexity of the underlying LPN problem in the average case.

Preliminaries. We consider the scenario where enough large samples $\{\mathbf{z}^{(t)}\}_{t=1}^{\tau}$ have been recorded by an attacker, who can now try to find the employed secret key $\mathbf{k}$ contained in $\mathbf{x}^t = \mathbf{x}^{(t)}(\mathbf{k})$ using

$$\mathbf{z}^{(t)} = C_{ECC}(C_H(\mathbf{a}^{(t)}||\mathbf{u}^{(t)})) \oplus \mathbf{x}^{(t)} \oplus \mathbf{v}^{(t)}, \quad t = 1, 2, \ldots, \tau,$$

since he has a probability of error in recovering the key which now tends to zero.

For the simplicity of exposition, we assume from now on that $|\mathbf{K}| = n$. We further perform the security evaluation under the following two assumptions:

- $\mathbf{x}^{(t)} = f^{(t)}(\mathbf{k}) = \mathbf{k}\mathbf{S}^t$, $t = 1, 2, \ldots, \tau$, where $\mathbf{S} = [s_{i,j}]_{i=1\ \ j=1}^{n\quad n}$, is a binary matrix, and
$$\mathbf{S}^t = [\mathbf{S}_1^{(t)}, \mathbf{S}_2^{(t)}, \ldots, \mathbf{S}_n^{(t)}]$$
where each $\mathbf{S}_i^{(t)}$, $i = 1, 2, \ldots, n$, denotes a column of the tth power of the matrix $\mathbf{S}$; note that usually $f^{(t)}(\cdot)$ is a heavily nonlinear function, and its consideration as a linear one actually implies a scenario for evaluation of a lower bound of the complexity for the secret key recovery;
- we consider the chosen plaintext attack where the data is the whole zero vector, i.e. $\mathbf{a}^{(t)} = \mathbf{0}$, for each t.

Under the above assumptions, and recalling that both C_{ECC} and C_H are linear encoders, we can write $\mathbf{k}\mathbf{S}^t \oplus [\mathbf{0}||\mathbf{u}^{(t)}]\mathbf{G} = \mathbf{z}^{(t)} \oplus \mathbf{v}^{(t)}$, from which we obtain an algebraic representation of the recovery problem in terms of a noisy system of linear equations, as seen by the adversary:

$$\begin{bmatrix} \mathbf{k}\mathbf{S}_1^{(t)} \\ \mathbf{k}\mathbf{S}_2^{(t)} \\ \vdots \\ \mathbf{k}\mathbf{S}_n^{(t)} \end{bmatrix} \oplus \begin{bmatrix} [\mathbf{0}||\mathbf{u}^{(\mathbf{t})}]\mathbf{G}_1 \\ \left[\mathbf{0}||\mathbf{u}^{(\mathbf{t})}\right]\mathbf{G}_2 \\ \vdots \\ \left[\mathbf{0}||\mathbf{u}^{(\mathbf{t})}\right]\mathbf{G}_n \end{bmatrix} = \begin{bmatrix} z_1^{(t)} \\ z_2^{(t)} \\ \vdots \\ z_n^{(t)} \end{bmatrix} \oplus \begin{bmatrix} v_1^{(t)} \\ v_2^{(t)} \\ \vdots \\ v_n^{(t)} \end{bmatrix}, \quad t = 1, 2, \ldots, \tau, \tag{4.6}$$

where $\mathbf{u}^{(t)} = [u_i^{(t)}]_{i=1}^{m-l}$ and $\mathbf{G}_i$ denotes the ith column of $\mathbf{G}$.

Remark 4.1. Note that in the set $\{\left[\mathbf{0}||\mathbf{u}^{(\mathbf{t})}\right]\mathbf{G}_i\}_{i=1}^{n}$ all the elements could be split into two non-overlapping subsets such that a subset contains k linearly independent elements, k at most $m-l$, and the other subset contains $n-k$ elements each of which is a linear combination of the elements from the first set, since $\left[\mathbf{0}||\mathbf{u}^{(\mathbf{t})}\right]\mathbf{G}$ only involves the lower part of $\mathbf{G}$, which is an $(m-l)\times n$ matrix, which has thus at

most $m - l$ linearly independent columns, and the other columns can be obtained as linear combinations.

On the LPN Problem. The problem of solving a system of linear equations in the presence of noise is directly related to LNP problem. What the LPN problem captures is that, given a security parameter k, a secret vector $\mathbf{x}$, and $\mathbf{g}_1, \ldots, \mathbf{g}_n$ randomly chosen binary vectors of length $n = O(k)$, it is possible knowing $y_i = \langle \mathbf{x} | \mathbf{g}_i \rangle$ and $\{\mathbf{g}_i\}_{i=1}^n$ to solve for $\mathbf{x}$ using standard linear-algebraic techniques as long as there is no noise. However, when each y_i is flipped (independently) with probability p, finding $\mathbf{x}$ becomes much more difficult. The problem of learning $\mathbf{x}$ in this latter case is refereed to as the learning parity in noise (LPN) problem.

Finally note that the LPN problem is equivalent to the problem of decoding of a general linear block code and it is known that this problem is NP-complete [68], and that relating security of an encryption technique to the LPN problem has been employed for security evaluation of certain stream ciphers (see [4]), for example).

Complexity Evaluation. A systematic way to solve a system of linear equations, with or without noise, is to perform a Gaussian elimination. If the system furthermore contains unknowns that we are not interested in finding, it is natural to start by removing them, so as to obtain a system with a smaller number of equations, where only the unknowns we would like to find are left. We will now show how such a strategy changes the noise present in the system of equations.

Lemma 4.4. *Consider the following system of N equations over the binary field $GF(2)$ to be solved for $x_1, \ldots, x_L$, $L \leqslant N$:*

$$\left(\bigoplus_{j=1}^{L} \alpha_j^{(i)} x_j \right) \oplus y_i = z_i \oplus e_i, \quad i = 1, 2, \ldots, M,$$

$$\left(\bigoplus_{j=1}^{L} \alpha_j^{(i)} x_j \right) \oplus \left(\bigoplus_{j=1}^{M} \beta_j^{(i)} y_j \right) = z_i \oplus e_i, \quad i = M+1, M+2, \ldots, N,$$

where $\{z_i\}_{i=1}^N$, $\{\alpha_j^{(i)}\}_{j=1\ i=1}^{L\ N}$ and $\{\beta_j^{(i)}\}_{j=1\ i=1}^{M\ N}$ are known, $\{x_j\}_{j=1}^L$, $\{y_j\}_{j=1}^M$ and $\{e_i\}_{i=1}^N$ are unknown, and each e_i is a realization of a random variable E_i such that $\Pr(E_i = 1) = p < 1/2$, $i = 1, 2, \ldots, N$. If

(1) *the Hamming weight of each vector $[\beta_1^{(i)}, \ldots, \beta_M^{(i)}]$ is greater or equal to some parameter w, for $i = M+1, M+2, \ldots, N$,*
(2) *and no $\bigoplus_{j=1}^M \beta_j^{(k)} y_j$, $k \in \{M+1, M+2, \ldots, N\}$, is a linear combination of any other w or less $\bigoplus_{j=1}^M \beta_j^{(i)} y_j$, $i \in \{M+1, M+2, \ldots, N\}$, i.e., there are at least w linearly independent sums $\bigoplus_{j=1}^M \beta_j^{(i)} y_j$ among those $i \in \{M+1, \ldots, N\}$,*

then, the problem of recovering the unknown $x_1, x_2, \ldots, x_L$ is the problem of solving the following system of equations:

$$\left(\bigoplus_{k=1}^{M} \beta_k^{(i)} \left(\bigoplus_{j=1}^{L} \alpha_j^{(k)} x_j \right) \right) \oplus \left(\bigoplus_{j=1}^{L} \alpha_j^{(i)} x_j \right) = z_i \oplus \left(\bigoplus_{k=1}^{M} \beta_k^{(i)} z_k \right) \oplus e_i^*,$$

for $i = M+1, M+2, \ldots, N$, where e_j^* is a realization of a random variable E_j^* such that $\Pr(E_j^* = 1) > p_w = \frac{1}{2}\big(1-(1-2p)^{w+1}\big)$.

Proof. For every $i \in \{M+1, M+2, \ldots, N\}$, adding the following linear combination of the first M equations

$$\Big(\bigoplus_{k=1}^{M} \beta_k^{(i)} \Big(\bigoplus_{j=1}^{L} \alpha_j^{(k)} x_j\Big)\Big) \oplus \Big(\bigoplus_{k=1}^{M} \beta_k^{(i)} y_k\Big) = \bigoplus_{k=1}^{M} \beta_k^{(i)} (z_k \oplus e_k),$$

to the ith equations of the system yields:

$$\Big(\bigoplus_{k=1}^{M} \beta_k^{(i)} \Big(\bigoplus_{j=1}^{L} \alpha_j^{(k)} x_j)\Big) \oplus \Big(\bigoplus_{j=1}^{L} \alpha_j^{(i)} x_j\Big) = z_i \oplus \Big(\bigoplus_{k=1}^{M} \beta_k^{(i)} z_k\Big) \oplus e_i \oplus \Big(\bigoplus_{k=1}^{M} \beta_k^{(i)} e_k\Big).$$

We are left to compute the probability $\Pr(E_i^* = 1)$, where

$$E_i^* = E_i \oplus \Big(\bigoplus_{k=1}^{M} \beta_k^{(i)} E_k\Big), \quad i = M+1, \ldots, N.$$

Since $i \geqslant M+1$, E_i is independent of $\beta_k^{(i)} E_k$ for every $1 \leqslant k \leqslant M$. We are thus summing the components of the vector $[E_i, E_1\beta_1^{(i)}, \ldots, E_M\beta_M^{(i)}]$ and

$$\Pr(E_i^* = 1) = 1 - \Pr(E_i^* = 0) = 1 - \Pr(E_i \oplus \Big(\bigoplus_{k=1}^{M} \beta_k^{(i)} E_k = 0)\Big).$$

Now the probability that an even number of digits are 1 in a sequence of $M+1$ independent binary digits is $\frac{1}{2}\big(1+(1-2p)^{M+1}\big)$ [79, Lemma1] if p is the probability that every digit is 1. Since $\frac{1}{2}\big(1+(1-2p)^{M}\big) > \frac{1}{2}\big(1+(1-2p)^{M+1}\big)$, $p < 1/2$, we have that $1 - \frac{1}{2}\big(1+(1-2p)^{M}\big) < 1 - \frac{1}{2}\big(1+(1-2p)^{M+1}\big)$, and

$$\begin{aligned}\Pr(E_i^* = 1) &= 1 - \Pr\Big(E_i \oplus \Big(\bigoplus_{k=1}^{M} \beta_k^{(i)} E_k = 0\Big)\Big) \\ &> 1 - \frac{1+(1-2p)^{w+1}}{2} = \frac{1-(1-2p)^{w+1}}{2}\end{aligned}$$

since by the assumption 1, the weight of each vector of $[\beta_1^{(i)}, \ldots \beta_M^{(i)}]$ is at least w, and according to the assumption 2., there is no linear combination of the equations which can reduce the corruption noise value lower bounded by p_w (i.e., it cannot be reduced via any further linear processing of the system of equations). □

This leads to the following evaluation result.

Theorem 4.3. *The complexity of recovering the secret key* **k** *based on the algebraic representation of the scheme is lower bounded by the complexity of solving the $LPN_{n,\epsilon}$ problem where, $\epsilon = \frac{1}{2}\big(1-(1-2p)^{w+1}\big)$ and n, w and p are the parameters of the scheme, representing resp. the length of the key, a parameter of the homophonic encoder and the probability of the BSC.*

Proof. From (4.6), we have the following system of τn overdefined consistent but probabilistic equations over the binary field $GF(2)$:

$$\begin{aligned}
\mathbf{kS}_1^{(t)} \oplus [\mathbf{0}||\mathbf{u}^{(t)}]\mathbf{G}_1 &= z_1^{(t)} \oplus v_1^{(t)} \\
\mathbf{kS}_2^{(t)} \oplus [\mathbf{0}||\mathbf{u}^{(t)}]\mathbf{G}_2 &= z_2^{(t)} \oplus v_2^{(t)} \\
&\vdots \qquad\qquad , \qquad t = 1, 2, \ldots, \tau, \\
\mathbf{kS}_n^{(t)} \oplus [\mathbf{0}||\mathbf{u}^{(t)}]\mathbf{G}_n &= z_n^{(t)} \oplus v_n^{(t)}
\end{aligned}$$

where each equation is correct with probability equal to p, $\mathbf{0}$ is a l-dimensional vector of all zeroes, and $\mathbf{u}^{(t)} = [u_i^{(t)}]_{i=1}^{m-l}$.

The above system of equations fits the setting of Lemma 4.3, since we have $N = \tau n$ equations, for $L = n$ unknown, where $\bigoplus_{j=1}^{L} \alpha_j^{(k)} x_j$, $k = 1, \ldots, N$ correspond to $\mathbf{kS}_i^{(t)}$, $i = 1, \ldots, n$, $t = 1, \ldots, \tau$, and y_j, $j = 1, \ldots, M$ together with $\bigoplus_{j=1}^{M} \beta_j^{(k)} y_j$ for $k = M+1, \ldots, N$ correspond to $[\mathbf{0}||\mathbf{u}^{(t)}]\mathbf{G}_i$, $i = 1, \ldots, M$, $t = 1, \ldots, \tau$, since according to Remark4.1, we can indeed separate the $\{[\mathbf{0}||\mathbf{u}^{(t)}]\mathbf{G}_i\}_{i=1}^{n}$ for every t into one set of linear independent vectors, and another set which is obtained as linear combinations of the first set (M is then τk, where k is at most $m - l$).

Note that the above system of τn equations contains only $n + \tau(m-l)$ unknown variables, and that our goal is to recover $\mathbf{k}$ only, i.e., we do not have any interest in recovering $\{u_i^{(t)}\}_{i=1}^{m-l}$, $t = 1, 2, \ldots, \tau$. Thus, via Gaussian elimination, we can remove the $\tau(m-l)$ unknown $\{u_i^{(t)}\}_{i=1}^{m-l}$, $t = 1, 2, \ldots, \tau$, and obtain $\tau(n-m+l)$ equations where only $\mathbf{k}$ is unknown. This transforms the initial system of τn equations into the following one with $\tau(n-m-l)$ equations (in total) and n unknowns $\mathbf{k}$:

$$\begin{aligned}
\mathcal{L}_1^{(k)}(\mathbf{k}) &= \mathcal{L}_1^{(z)}\big([z_i^{(t)}]_{i=1}^{n}\big) \oplus \mathcal{L}_1^{(v)}\big([v_i^{(t)}]_{i=1}^{n}\big) \\
\mathcal{L}_2^{(k)}(\mathbf{k}) &= \mathcal{L}_2^{(z)}\big([z_i^{(t)}]_{i=1}^{n}\big) \oplus \mathcal{L}_2^{(v)}\big([v_i^{(t)}]_{i=1}^{n}\big) \\
&\vdots \qquad\qquad\qquad\qquad\qquad\qquad , \quad t = 1, 2, \ldots, \tau, \\
\mathcal{L}_{n-m+l}^{(k)}(\mathbf{k}) &= \mathcal{L}_{n-m+l}^{(z)}\big([z_i^{(t)}]_{i=1}^{n}\big) \oplus \mathcal{L}_{n-m+l}^{(v)}\big([v_i^{(t)}]_{i=1}^{n}\big)
\end{aligned} \tag{4.7}$$

where $\mathcal{L}_j^{(k)}(\cdot)$, $\mathcal{L}_j^{(z)}(\cdot)$ and $\mathcal{L}_j^{(v)}(\cdot)$, $j = 1, 2, \ldots, n-m+l$, are linear functions, all of them specified by the matrix $\mathbf{G}$ and the Gaussian elimination used to remove the random bits $\mathbf{u}^{(t)}$, while $\mathcal{L}_j^{(k)}(\cdot)$ further depends on the matrix $\mathbf{S}^t$. Note that the Gaussian elimination of the variables $\{u_i^{(t)}\}_{i=1}^{m-l}$, can be performed independently for each t, implying that the entire complexity (for $t = 1, 2, \ldots, \tau$) is upper-bonded by $\tau O(n^{2.7})$ assuming employment of the most efficient algorithm for the Gaussian processing.

Lemma 4.3 and its underlying assumptions provide that each equation in (4.7) is correct with some probability lower than $1 - p_w$, where $p_w = \frac{1}{2}\big(1 - (1-2p)^{w+1}\big)$, since the noise $(\mathbf{v}_1^*)^{(t)} = \mathcal{L}_1^{(v)}([v_i^{(t)}]_{i=1}^{n}), \ldots, (\mathbf{v}_{n-m+l}^*)^{(t)} = \mathcal{L}_{n-m+l}^{(v)}([v_i^{(t)}]_{i=1}^{n})$ has coefficients that are the realization of a random variable which takes value 1 with

probability greater than $p_w = \frac{1}{2}\big(1-(1-2p)^{w+1}\big)$. The above system of $\tau(n-m+l)$ equations can consequently be rewritten as:

$$\begin{aligned}
\mathcal{L}_1^*([k_i]_{i=1}^n) &= \mathcal{L}_1^{(z)}([z_i^{(1)}]_{i=1}^n)\\
\mathcal{L}_2^*([k_i]_{i=1}^n) &= \mathcal{L}_2^{(z)}([z_i^{(1)}]_{i=1}^n)\\
&\vdots\\
\mathcal{L}_{n-m+l}^*([k_i]_{i=1}^n) &= \mathcal{L}_{n-m+l}^{(z)}([z_i^{(1)}]_{i=1}^n)\\
\mathcal{L}_{n-m+l+1}^*([k_i]_{i=1}^n) &= \mathcal{L}_1^{(z)}([z_i^{(2)}]_{i=1}^n)\\
\mathcal{L}_{n-m+l+2}^*([k_i]_{i=1}^n) &= \mathcal{L}_2^{(z)}([z_i^{(2)}]_{i=1}^n)\\
&\vdots\\
\mathcal{L}_{\tau(n-m+l)}^*([k_i]_{i=1}^n) &= \mathcal{L}_{n-m+l}^{(z)}([z_i^{(\tau)}]_{i=1}^n)
\end{aligned}$$

where $\mathcal{L}_j^*$, $j = 1, 2, \ldots, \tau(n-m+l)$, are linear functions, and where each equation is incorrect with probability greater than $p_w = \frac{1}{2}\big(1-(1-2p)^{w+1}\big)$.

We finally get:

$$\begin{aligned}
\langle \mathbf{k}|\mathbf{c}_1\rangle &= d_1\\
\langle \mathbf{k}|\mathbf{c}_2\rangle &= d_2\\
&\vdots\\
\langle \mathbf{k}|\mathbf{c}_{n-m+l}\rangle &= d_{m-n+l}\\
\langle \mathbf{k}|\mathbf{c}_{n-m+l+1}\rangle &= d_{m-n+l+1}\\
\langle \mathbf{k}|\mathbf{c}_{n-m+l+2}\rangle &= d_{m-n+l+2}\\
&\vdots\\
\langle \mathbf{k}|\mathbf{c}_{\tau(n-m+l)}\rangle &= d_{\tau(m-n+l)}
\end{aligned}$$

where each equation is correct with a probability upper-bounded by $1-p_w = 1-\frac{1}{2}\big(1-(1-2p)^{w+1}\big)$, and where the n-dimensional binary vectors $\{\mathbf{c}_j\}_{j=1}^{\tau(n-m+l)}$ and $\{d_j\}_{j=1}^{\tau(n-m+l)}$ are known.

According to the definition of the LPN problem and the above representation, the problem of recovering the secret key is at least as hard as the $\mathrm{LPN}_{n,\epsilon}$ problem with $\epsilon = \frac{1}{2}\big(1-(1-2p)^{w+1}\big)$, which concludes the proof of the theorem. □

4.4. A Generalization of the LPN Problem and Its Hardness. The LPN problem has a number of equivalent formulations and under consideration of this section is a formulation which corresponds to the decoding problem. It has been shown in [68] that the decoding incarnation of the LPN problem is NP-complete which implies suitability of this problem as an underlying problem for certain cryptographic applications. The basic LPN problem can be considered as a problem of solving an

overdefined and consistent system of linear but noisy equations corresponding to the following vector equation:

$$\mathbf{z} = \mathbf{kS} \oplus \mathbf{v}, \tag{4.8}$$

where $\mathbf{k}$ is $|\mathbf{k}|$-dimensional binary vector of the variables which are target of recovering, $\mathbf{z}$ is given n-dimensional, $n \gg |\mathbf{k}|$, binary vector, $\mathbf{S}$ is known $|\mathbf{k}| \times n$ binary matrix, and $\mathbf{v}$ is unknown n-dimensional binary vector of independent identically distributed elements which take value 1 with the probability $\epsilon = p < 1/2$ and value 0 with the probability $1 - p$.

The goal is recovering of $\mathbf{k}$ with minimization of the probability of error and this goal corresponds to decoding of a linear block code. Accordingly, the goal can be achieved employing the minimum distance decoding paradigm based on an exhaustive search according to the following:

- For each possible candidate $\hat{\mathbf{k}}$ for $\mathbf{k}$ evaluate the Hamming weight, $Hwt(\cdot)$, of the vector $\mathbf{z} \oplus \hat{\mathbf{k}}\mathbf{S}$;
- Select as the true candidate $\hat{\mathbf{k}}$ the one which provides minimum $Hwt(\mathbf{z} \oplus \hat{\mathbf{k}}\mathbf{S})$.

A generalized LPN problem can be formulated as a problem of solving an overdefined and consistent system of linear but noisy equations corresponding to the following vector equation:

$$\mathbf{z} = [\mathbf{a}||\mathbf{u}]\mathbf{G} \oplus \mathbf{kS} \oplus \mathbf{v}, \tag{4.9}$$

where $\mathbf{a}$ is known l-dimensional binary vector, $\mathbf{u}$ is an unknown random $(m - l)$-dimensional binary vector of independent identically distributed elements which take values 0 and 1 with the same probability equal to 1/2, $\mathbf{G}$ is known $m \times n$ binary matrix, $l < m < n$, and $\mathbf{z}$, $\mathbf{k}$, $\mathbf{S}$ and $\mathbf{v}$ are defined by the above basic LPN problem specification.

Regarding the generalized LPN problem specified by (4.9), a similar approach can be employed as regarding the basic one (4.8) but taking into account that not only $\mathbf{k}$ is unknown but $\mathbf{u}$ as well. Accordingly, we can employ the following approach for recovering unknown $\mathbf{k}$:

- For each possible candidate $\hat{\mathbf{k}}$ for $\mathbf{k}$ and all possible vectors $\mathbf{u}$ evaluate the Hamming weight, $Hwt(\cdot)$, of the corresponding vectors $\mathbf{z} \oplus [\mathbf{a}||\hat{\mathbf{u}}]\mathbf{G} \oplus \hat{\mathbf{k}}\mathbf{S}$;
- – If a unique minimum $Hwt(\cdot)$ exists, select as the true candidate $\hat{\mathbf{k}}$ the one which yields this minimum value;
  – If the unique minimum $Hwt(\cdot)$ does not exist, make a list $\mathcal{L}$ of the final candidates $\hat{\mathbf{k}}$ for the true $\hat{\mathbf{k}}$ such that each final candidate yields:

$$Hwt(\mathbf{z} \oplus [\mathbf{a}||\hat{\mathbf{u}}]\mathbf{G} \oplus \hat{\mathbf{k}}\mathbf{S}) \leqslant w_{thr},$$

where w_{thr} is certain threshold value.

Note that the outcome of the considered approach depends on the parameters $|\mathbf{k}|$, l, m and n. If $|\mathbf{k}| + m - l - n > 0$ after the exhaustive search we obtain $2^{|\mathbf{k}|+m-l-n}$ pairs $(\hat{\mathbf{k}}, \hat{\mathbf{u}})$ which yield that $Hwt(\mathbf{z} \oplus [\mathbf{a}||\hat{\mathbf{u}}]\mathbf{G} \oplus \hat{\mathbf{k}}\mathbf{S}) = 0$ yielding that the approach outcome is $2^{|\mathbf{k}|+m-l}(1-p)^n$ equally likely candidates for the true $\mathbf{k}$. On the other hand, the all zeros noise pattern is not the most likely one, and in

order not to miss inclusion into the list of candidates, the true one, instead of only minimum $Hwt(\cdot)$, all error patterns up to certain weight should be included as the eligible candidates yielding the list of candidates $\mathcal{L}$.

Note the following: For given $\mathbf{z}$ and $\mathbf{a}$ and assumed $\mathbf{k}$, 2^{m-l} different vectors $\mathbf{u}$ will yield 2^{m-l} different vectors $\mathbf{z} \oplus \mathbf{kS} \oplus [\mathbf{a}||\mathbf{u}]\mathbf{G}$ assuming appropriate matrix $\mathbf{G}$. Employing the random arguments, among these vectors $2^{m-l}\binom{n}{w}p^w(1-p)^{n-w}$ will have the Hamming weight w. Accordingly, the expected dimension of the list $\mathcal{L}$ can be estimated as follows:

$$|\bar{\mathcal{L}}| = \min\left\{2^{|\mathbf{k}|}, 2^{|\mathbf{k}|+m-l}\sum_{w=0}^{w_{thr}}\binom{n}{w}p^w(1-p)^{n-w}\right\}.$$

and lower-bounded as

$$|\bar{\mathcal{L}}| \geqslant \min\left\{2^{|\mathbf{k}|}, 2^{|\mathbf{k}|+m-l-n}\sum_{w=0}^{w_{thr}}\binom{n}{w}\right\}.$$

When $w_{thr} = pn$,

$$\sum_{w=0}^{w_{thr}}\binom{n}{w} \leqslant 2^{h(p)n}$$

where $h(p) = -p \cdot \log_2 p - (1-p) \cdot \log_2(1-p)$.

The above consideration has the following implication: If the parameters of the generalized LPN problem are such that $m - l > n(1 - h(p))$, after the considered optimal search for the true hypothesis on $\mathbf{k}$, the expected number of equally-likely candidates can be close to the total number of candidates.

5. A Security Evaluation of Broadcast Encryption Key Management Schemes

5.1. Introduction. A conventional approach for access control to broadcasted (multicasted) data employs the following paradigm: the data are encrypted and only legitimate users are provided with the information on how to decrypt them. We consider schemes where the data encryption is performed based on a symmetric cipher and the updatable secret session encrypting key (SEK). To make SEK updating possible, another set of keys called the key-encrypting keys (KEKs) are involved. There are two basic approaches for establishing the required security based on the above paradigm. One approach uses static KEKs (see [99], [64] and [94], for example), and the other one employs updatable KEKs (see [101], [102], and [110], for example). BE schemes with static KEKs (stateless receivers) have the desirable feature that members do not need to be always connected online, which is especially preferable for applications over lossy channels. Since rekey messages in stateless schemes are independent of each other, members once being offline or inactive can always decrypt the latest group key instantly.

In this section the security evaluation of certain BE schemes with static KEKs is considered. In order to enhance the security of these schemes, before the encryption,

the SEK is XOR-ed with the identifier (ID) of the KEK employed for its encryption, as proposed in [94] and [99].

This section points out to a weakness of certain broadcast encryption schemes in which the protected delivery of a session key (SEK) is based on XOR-ing this SEK with the IDs of the keys employed for its encryption is addressed. The weakness can be effectively explored assuming passive attacking which in the cases corresponding to a malicious legitimate user being the attacker, is a ciphertext only attack. A dedicated algorithm for cryptanalysis is discussed based on a generalized time-memory-data trade-off approach and its main characteristics are derived. The algorithm points out a security weakness of employing a block cipher with block length shorter than the key length in the considered BE schemes.

5.2. Models of Certain Broadcast Encryption and Problem Statement. Let KEK_i denote a KEK employed in the system, and let ID_i denote its name or ID, assuming that ID_i does not disclose any information on KEK_i itself. BE is based on the following approach. The system center generates all the employed KEKs. A user of the BE system is in advance provided with a subset of all KEKs employed in the system. Note that different users can have overlapping subsets of KEKs, but no pair of users have an identical subset.

In a basic BE setting, the procedures at the center and for each of the users are based on the following. When the current SEK should be updated, the center finds a subset $I = I(SEK)$ of KEKs $\{KEK_i\}_{i\in I}$ such that each of the legitimate users possesses at least one of these keys and none of the un-legitimate users possesses any of these keys. The center encrypts the data with SEK, generates all encrypted forms of SEK employing each KEK_i, $i \in I$, and broadcasts $\langle [header]; G_{SEK}(data)\rangle = \langle [\{(ID_i, E_{KEK_i}(SEK))\}_{i\in I}]; G_{SEK}(data)\rangle$, where $E(\cdot)$ and $G(\cdot)$ are certain encryption algorithms.

In order to address certain weaknesses of this basic BE model, in [94], an enhanced security approach for BE is proposed, which corresponds to the following BE header model:

$$[header] = [\{(ID_i, E_{KEK_i}(SEK \oplus ID_i))\}_{i\in I}], \tag{5.1}$$

where $\oplus$ denotes bit-by-bit XOR-ing of the vectors SEK and ID_i. This enhanced approach is employed in[64] as well.

The problem addressed in this section is the security evaluation of the BE schemes which follow the header model specified by (5.1). Recently, vulnerabilities of certain BE schemes have been reported in [10] and [8] and they provide the origins for the approach given in this letter. Particularly note that a security weakness of the approach proposed in [94] is reported in [10] employing an active attack scenario, while the scope of this consideration is restricted to passive attacking.

5.3. Scenario for the Security Evaluation. The considered settings for cryptanalysis originate from the following issues: (i) It is a realistic scenario that different schemes of the same class are deployed and are subject to malicious monitoring; (ii) In a typical BE scheme with stateless receivers KEKs are in a tamper-proof

(resistant) hardware and accordingly the system should be considered as broken even if an attacker can recover only one of the KEKs employed in the system.

We assume a system where N BE schemes of the same structure but with different (non-overlapping) KEKs are employed and that the attacker can monitor J SEK updates in each of these schemes. In the considered system, in order to provide the legitimate users with the decryption key $SEK_j^{(n)}$, the set $S_j^{(n)}$ of the following pairs is publicly available:

$$S_j^{(n)} = \{(ID_i^{(n)}, C_{i,j}^{(n)})\}_{i \in I^{(n)}(SEK_j^{(n)})}, \tag{5.2}$$

where $C_{i,j}^{(n)} = E_{KEK_i^{(n)}}(SEK_j^{(n)} \oplus ID_i^{(n)})$, $j = 1, 2, \ldots, J$, $n = 1, 2, \ldots, N$, and where $E(\cdot)$ is a block cipher which employs length-K secret key and operates over L-dimensional binary blocks. We also assume that the following is valid:

- i_{max} KEKs are employed in each of the considered N BE schemes and for each KEK: (i) $KEK_i^{(n)}$ is a randomly generated binary vector of length-K and $2^K \gg Ni_{\max}$; (ii) $ID_i^{(n)}$ is a length-L binary vector, $2^L > i_{\max}$, which only indicates that the encrypted form of $SEK_j^{(n)}$ is obtained employing the key $KEK_i^{(n)}$ and does not provide any information on the binary vector $KEK_i^{(n)}$;
- For each SEK: (i) SEK is a binary vector of dimension L, $K/2 \leqslant L < K$; (ii) each $I(SEK)$ is a subset of $\{1, 2, \ldots, i_{max}\}$, and for simplicity, we assume that $|I(SEK)| = I$.
- The employed encryption algorithm $E(\cdot)$ is secure, so that the following holds: any $C_i = E_{KEK_i}(SEK \oplus ID_i)$ does not yield any information on KEK_i and SEK.

The attacker's prior knowledge is limited to the following: (i) The attacker knows the entire structure of the BE scheme under cryptanalysis including the employed encryption algorithm $E(\cdot)$; (ii) The attacker does not know any of the keys $KEK_i^{(n)}$, $i = 1, 2, \ldots, i_{max}$, $n = 1, 2, \ldots, N$, employed in the considered N BE schemes.

The goal of the attacker is to recover at least *one of the secret keys* $KEK_i^{(n)}$, $i = 1, 2, \ldots, i_{\max}$, $n = 1, 2, \ldots, N$, employed in the considered system of BE schemes.

The scenario for cryptanalysis assumes that the attacker has a suitable (large) collection of the following data: (i) the headers $S_j^{(n)}$ specified by (5.2), and (ii) the employed $SEK_j^{(n)}$.

5.4. A Method for Cryptanalysis of Certain Broadcast Encryption Schemes.

The developed technique for security evaluation of the considered class of BE schemes includes the following: (i) collecting a suitable sample via monitoring SEKs update in a number of different BE schemes of the considered class; (ii) employment of a sophisticated and dedicated "dictionary" with implicitly memorized words which provides a time-memory-data trade-off.

Assuming that $SEK_j^{(n)}$ is selected randomly and independently of $ID_i^{(n)}$, $i = 1, 2, \ldots, I$, the probability that the given $SEK_j^{(n)}$ is equal to one of $ID_i^{(n)}$, $i =$

$1, 2, \ldots, I$, is equal to $I2^{-L}$ as all $ID_i^{(n)}$'s are distinct. Accordingly, the probability $P^*(k)$ that, for a given n, there are exactly k common elements in the sets $\{SEK_j^{(n)}\}_{j=1}^{J}$ and $\{ID_i^{(n)}\}_{i=1}^{I}$ is:

$$P^*(k) = \binom{J}{k} 2^{-k(L-\log_2 I)} (1 - 2^{-L+\log_2 I})^{J-k}. \tag{5.3}$$

Note that (5.3) implicitly assumes that all $SEK_j^{(n)}$'s are distinct, which is readily achieved for $2^L \gg J$. This implies that in the sets $\{SEK_j^{(n)} \oplus ID_i^{(n)}\}_{i=1}^{I}{}_{j=1}^{J}$, $n = 1, 2, \ldots, N$, the expected number $\hat{D}$ of the the elements which sum to a certain pattern is given by:

$$\hat{D} = NJI2^{-L}. \tag{5.4}$$

In the following, for simplicity of the exposition, we consider the all zeros pattern $\mathbf{0}$, and in the same manner any other pattern can be employed.

A time-memory-data trade-off (TMD-TO) approach for cryptanalysis has been reported in [69] as a generalization of the time-memory trade-off based cryptanalysis [84]. In this section, a further generalized dedicated time-memory-data trade-off approach for cryptanalysis of the considered BE system is proposed, assuming that the parameters are such that $IJN > 2^L$. The main steps are:

• Perform a suitable pre-processing for a dedicated TMD-TO based cryptanalysis assuming that the encryption algorithm encrypts only all zeros L-dimensional binary vectors. The pre-processing output is a set of tables. This pre-processing should be done only once and is independent of the sample for cryptanalysis and the KEKs employed in the system.

• Collect the sample for processing consisting of the ciphertext corresponding to the vectors $SEK_j^{(n)} \oplus ID_i^{(n)} = \mathbf{0}$ (on average D values based on (5.4)).

• Using the tables prepared during pre-processing and the collected sample perform the cryptanalysis employing a dedicated TMD-TO based cryptanalysis and generate a list of candidates.

• From the list of candidates recover one or more KEKs via an additional check of each candidate.

5.4.1. Algorithm for Cryptanalysis.

Pre-Processing

- *Input Data*: The algorithm parameters K, L, M, T, and D such that $M^2D^2T = 2^{2L}$.
- *Pre-Processing Steps*
  For $u = 1, 2, \ldots, 2^{K-L}$ and $i = 1, 2, \ldots, M$, do the following:
  (1) Randomly select an L-dimensional binary vector X_0' and define $X_i(0) = X_0'||U$ where U is the length-$(K-L)$ binary representation of $u-1$, and $||$ denotes the concatenation of two vectors.
  (2) For $t = 1, 2, \ldots, T$, perform the following recursive calculation: $X = E_{X_i(t-1)}(\mathbf{0})$, $X_i(t) = X||U$.

(3) Memorize $X_i(0)$ in the first column and the first L elements of $X_i(T)$ in the second column of the i-th row of the $M \times 2$ matrix M_u.

- *Output*: Tables M_u, $u = 1, 2, \ldots, 2^{K-L}$, of the pairs memorized in step 3.

Processing

- *Input Data*: Set S_D of D different values $C_{i,j}^{(n)} = E_{KEK_i^{(n)}}(SEK_j^{(n)} \oplus ID_i^{(n)} = \mathbf{0})$, $i \in I(j)$, $j \in \{1, 2, \ldots, J\}$, $n \in \{1, 2, \ldots, N\}$, $I = |I(j)|$.
- *Processing Steps*

  I. *Generation of the List of Candidates*

  For each triple (n, j, i) such that $C_{i,j}^{(n)} \in S_D$, and all tables M_u, $u = 1, 2, ..., 2^{K-L}$, do the following:

  (1) Set: $t = 0$, $X_t' = C_{i,j}^{(n)}$, and $X_t = X_t' || U$ where U is the length-$(K - L)$ binary representation of $u - 1$.

  (2) Check the identity of the considered X_t' to any of the second column elements of the matrix M_u; if an identity appears in the i-th row, go to step 4; otherwise go to step 3.

  (3) Set $t \to t + 1$. If $t \leqslant T$, calculate $X_t' = E_{X_{t-1}}(\mathbf{0})$, $X_t = X_t' || U$, and go to step 2; if $t > T$, go to step 5.

  (4) (a) Select the corresponding $X_i(0)$ and set $X_0 = X_i(0)$;
  (b) Perform the following iterative calculation: $X_{v+1}' = E_{X_v}(\mathbf{0})$, $X_{v+1} = X_{v+1}' || U$ until $X_{v+1}' = C_{i,j}^n$; Memorize X_v into the list of candidates *List*, and go to step 5.

  (5) Select a previously not considered $C_{i,j}^{(n)}$ and go to step 1; If all elements of S_D have been considered go to phase II.

  II. *Recovering KEKs from the List of Candidates*

  For each candidate Y from *List* do the following:

  (1) For $O(1)$ different randomly selected indices j, $j \in \{1, 2, \ldots, J\}$, check the equality of $E_{KEK_i = Y}(SEK_j^{(n)} \oplus ID_i^{(n)}) = C_{i,j}^{(n)}$;

  (2) If all the checks in the previous step are fulfilled, memorize the considered Y as the recovered $KEK_i^{(n)}$.
- *Output*: Set of the recovered KEKs obtained via the memorized pairs in step II.2.

5.4.2. Complexity of Cryptanalysis. Based on the structure of the considered algorithm, and the results on TMD-TO reported in [69] the following statements are readily proved.

Proposition 5.1. *The proposed algorithm has space complexity $2^{K-L}M$, preprocessing time complexity proportional to $2^{K+L}M^{-1}D^{-2}$ and expected processing time complexity $O(2^{K+L}M^{-2}D^{-2}) + O(2^{K-L})$, assuming $D > 1$ and the goal is recovering one KEK. It provides different possible trade-offs between the parameters T,M, D and L, assuming that the following trade-off condition holds $TM^2D^2 = 2^{2L}$.*

5.5. Security Evaluation of Certain BE Schemes. A numerical illustration of Proposition 5.1 is given in Table 4 assuming employment of a block cipher which

operates over 64-bit blocks and uses a length-K secret key with $K > 64$. The column regarding 80-bit KEKs can correspond to a block cipher with a variable-length key (see [77], for example). The column corresponding to 128-bit KEKs holds even if a highly secure block-cipher MISTY1 (accepted as a standard block-cipher in [85]) is employed, and the column corresponding to 112-bit KEKs holds when the Triple DEA (see also [85]) with two 56-bit keys, which is a standard encryption primitive in many commercial products, is employed.

TABLE 4. Complexity of recovering one KEK of K bits in a system with N BE schemes and $J = 2^{30}$ SEK updates in each one, when SEKs and IDs consists of $L = 64$ bits, $K > L$, and a secure block cipher which operates over $L = 64$-bit blocks is employed.

KEK dimension K	80 bits	112 bits	128 bits
number N of monitored BE schemes	2^{20}	2^{30}	2^{35}
number I of KEKs in a BE scheme	$\sim 2^{30}$	$\sim 2^{35}$	$\sim 2^{40}$
the algorithm parameter M	2^{40}	2^{21}	2^{21}
space complexity of the algorithm	$\sim 2^{56}$	$\sim 2^{69}$	$\sim 2^{85}$
time complexity of pre-processing	$\sim 2^{72}$	$\sim 2^{94}$	$\sim 2^{99}$
expected time complexity of recovering one KEK	$\sim 2^{32}$	$\sim 2^{70}$	$\sim 2^{78}$

5.6. Concluding Remarks. It has been shown that indirect encryption of SEKs (modified by XOR-ing with IDs of KEKs) employing KEKs longer than SEKs does not provide the desired protection of KEKs in a number of scenarios. The developed generalized TMD-TO algorithm for cryptanalysis shows that the employment of block ciphers which operate over blocks shorter than the key in certain BE schemes implies a security weakness of these schemes regardless of the security level of the considered block cipher. Particularly, note that the employment of even highly secure block cipher has no impact against the proposed technique for cryptanalysis because the performance of the proposed algorithm for cryptanalysis does not depend on the security of the employed cryptographic primitives but on the considered BE system parameters. In the process, we also generalized the algorithms [84, 69] to the case where the secret key length is larger than the length of the encrypted blocks.

6. Design of Certain Broadcast Encryption Schemes

This section addresses the following issues:

- An approach for the cryptographic keys management in the broadcasting scenario with a conditional access control is considered. It employs the reconfiguration concept, and it is based on a collection of the underlying structures-at each instant of time a structure from the collection is employed for updating the session key so that the communication overhead of updating is minimized.

A receiver has a fixed set of cryptographic keys and in a general case, each of these keys plays a different role determined by the employed underlying structure.

- The problem of minimizing the amount of secret information (secret bits) required for certain key management schemes related to data access control techniques is addressed. The importance of secret storage minimization originates from the fact that this storage should be both read-proof and tamper-proof. The proposed approach intends to minimize the protected (secret) storage by introducing public storage in conjunction with an efficient one-way mapping of the secret bits in the exchange of information from private to public storage. This method achieves reduction of the secret storage overhead at the user's side and provides an appropriate trade-off between the reduced private storage, and the required public storage and associated processing complexity. The method is applied to the minimization of the secret storage required by two recently proposed key management schemes, and the overheads of the modified schemes are compared with the related previously reported ones, pointing out the advantages of the novel approach.

6.1. Reconfigurable Broadcast Encryption. This section addresses a problem of conditional access control to the broadcasted data where at each instant of time only the legitimate receivers have the access assuming that the set of these receivers is a highly dynamical one. The same scenario as in [83] is considered. The broadcasting is towards receivers with the pre-configured and not updatable states and has the following main requirements: (i) Each user is initially given a collection of symmetric encryption keys; (ii) The keys do not change when other users join or leave the system.

This section considers an approach for the key management which employs the reconfigurability concept. A generic framework for the reconfigurable key management is shown and an illustrative particular scheme is discussed. In the considered particular case, the developed approach is compared with the best previously reported schemes, and the advantages of the novel one are pointed out.

6.1.1. Background: Conditional Access Control and Key Management. As pointed out in the previous section, when cryptography is employed for securing broadcasting communications, a session-encrypting key (SEK) is used to encrypt the data. Ensuring that only the valid members of the group have SEK at any given time instance is the key management problem in secure broadcasting/multicast communications. To make this updating possible, another set of keys called the key-encrypting keys (KEKs) should be involved so that it can be used to encrypt and transmit the updated SEK to the valid members of the group. Hence, the key management problem reduces to the problem of distributing KEKs to the members such that at any given time instant all the valid members can be securely updated with the new SEK.

In [99], a generic framework, is given by encapsulating several previously proposed revocation methods called Subset-Cover algorithms. These algorithms are based on the principle of covering all non-revoked users by disjoint subsets from

a predefined collection, together with a method for assigning KEKs to subsets in the collection. For further discussion we assume that there are N users and R revocations.

Two types of revocation schemes in the Subset-Cover Framework, are proposed [99] with a different performance trade-off. Both schemes are tree-based, namely the subsets are derived from a virtual tree structure imposed on all receivers in the system. The first proposed scheme, Complete Sub-Tree (CST) scheme, requires a message length of $R\log_2(N/R)$ and storage of $1+\log_2 N$ keys at the receiver and constitutes a moderate improvement over previously proposed schemes. The second scheme, called the Subset Difference (SD) algorithm exhibits a more improvement: it requires a message length of $2R$. The improved performance of SD is primarily due to its more sophisticated choice of covering sets. Let i be any vertex in the tree and let j be any descendent of i. Then $S_{i,j}$ is the subset of leaves which are descendants of i but are not descendants of j. Note that $S_{i,j}$ is empty if $i=j$. Otherwise, $S_{i,j}$ looks like a tree with a smaller subtree cut out. The SD scheme covers any privileged set P defined as the complement of R revoked users by the union of $O(R)$ of these $S_{i,j}$ sets.

What is shown in [83] is that this collection of sets can be reduced: The basic idea of the Layered Subset Difference (LSD) scheme is to use only a small subcollection of $S_{i,j}$ sets employed by SD scheme which suffices to represent any P as the union of $O(R)$ of the remaining sets, with a slightly larger constant. Since there are fewer possible sets, it is possible to reduce the number of initial keys given to each user. In [83], it is shown that if we allow the number of sets in the cover to grow by a factor of two, we can reduce the keys storage from $O((\log_2 N)^2)$ to $O((\log_2 N)^{3/2})$.

6.1.2. Reconfigurable Key Management. Underlying Ideas. An approach for the key management is proposed in [19] and [18] which has the following basic characteristics: (i) It employs the reconfiguration concept, and it is based on a collection of the underlying structures-at each instant of time a structure from the collection is employed for updating the session key SEK so that the communication overhead of updating is minimized; (ii) A receiver has the single set of cryptographic keys which (and each of these keys) plays a role determined by the employed underlying structure.

A main component of the reconfigurable key management is a collection of the underlying structures, and regarding these structures note the following.

- The underlying structures could be very different but all of them should fulfil the following condition: They should be able to work with the same set of keys (KEKs) assuming that a key can be employed in different modes.
- Selection of the underlying structures for the collection depends on the functional requirements of the key management, and particularly on the identified most likely classes of the revocation patterns.

Generic Framework

- Center forms a collection of different underlying structures suitable for different revocation scenarios; usually, each underlying structure is a graph with the keys

and users assigned to the graph nodes, and the employed graph is a tree where the users are assigned to the tree leaves, and the keys are assigned to all the tree nodes.

- Taking into account all the underlying structures, center selects a set of the keys (KEKs) to be assigned to each of the receivers, as an appropriate subset of all the keys related to the employed underlying structures. (Note that in a general case, the subsets corresponding to the different receivers are the overlapping ones.)
- For each of the session key, SEK, updating, center selects one of the underlying structures in such a way to minimize the communication overhead under given revocation requests.
- Center performs SEK updating by broadcasting the following: (i) encrypted forms of the new SEK obtained by employing different KEKs; (ii) labels of the employed KEKs, and (iii) the mod of KEKs use (which depend on currently used underlying structure).
- If not revoked, during the key management communication, a receiver will find the following information in the updating message: (i) which of its KEKs should be employed for the new SEK recovering, and (ii) in which mode the KEK should be used. Accordingly, a not revoked receiver is able to recover the new SEK.

Note that the proposed framework employs a reconfigurable logical key hierarchy (RLKH), and accordingly we call it RLKH. Also note that the proposed framework for design of particular reconfigurable, i.e. time varying, key management schemes could provide minimization of the cumulative main system overheads (storage and processing at receiver, and updating communications) over highly dynamical set of privileged users.

Implementation Issues. At the center side RLKH implementation includes establishing of the RLKH system, and during this phase the center selects the component key management schemes so that each of them is suitable for certain class of the revocation patterns. Accordingly, during the establishing phase the center forms a list of the following pairs: (*revocation pattern class; key management scheme*). Storage requirements for this list of pairs and related information on the component schemes is usually negligible in comparison with the number of keys which should be stored at the center. One-to-one correspondence between the revocation pattern class and the component scheme implies that RLKH employment does not require any additional processing for selecting a particular key management at any time instance.

At a receiver, in a general case, according to the extracted information from the broadcast, a mapping of a KEK should be performed. Note that this mapping is not a secret operation and usually it should be the cryptographic one-way hashing (see [98], for example).

6.1.3. Illustrative Example of Reconfigurable Key Management. This section yields an illustrative toy example of the reconfigurable key management when only two underlying tree structures are employed. It is assumed that there

are $N = 2^n$ receivers and that each receiver holds $1 + log_2 N$ keys. The following two underlying structures, Tree-A and Tree-B, are employed:

• Tree-A: a binary balanced tree graph of height $log_2 N$;

• Tree-B: a tree graph with M branches from the root, and a binary balanced sub-tree of height $log_2(N/M)$ rooted at each of M branches, where M is a parameter which value will be discussed bellow.

An usual assignment of the center, users and keys to the considered trees is assumed: (i) the center is assigned to the root; (ii) each receiver is a leaf of the tree; (iii) all the employed keys in the scheme are assigned to the tree nodes.

Illustrative examples of the considered trees are given in Fig. 3 and 4.

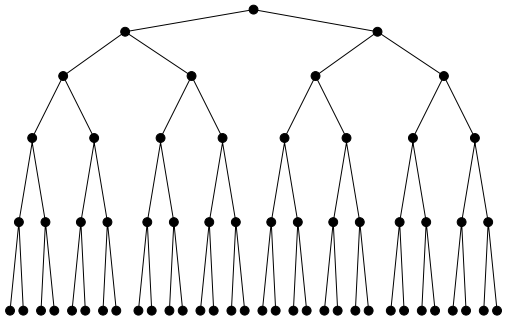

FIGURE 3. An illustration for the underlying structure Tree-A when $N = 32$.

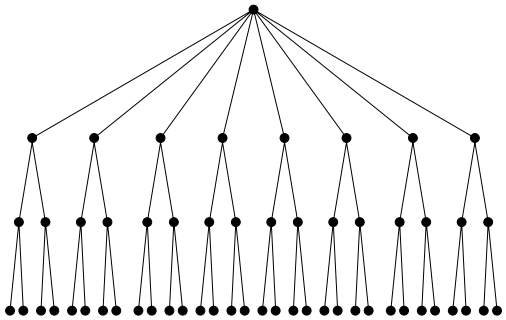

FIGURE 4. An illustration for the underlying structure Tree-B when $N = 32$ and $M = 8$.

The above trees are related to certain key management schemes according to the following:

• CST[99] is employed over Tree-A;

• basic LSD[83] is employed over each of M binary balanced sub-trees in Tree-B.

It is assumed that in the system developing phase the center has established one-to-one correspondence between the revocation pattern class and appropriate component scheme so that the appearance of the revocation pattern directly selects the scheme to be employed.

Having in mind that according to [83] employment of LSD based key management over a binary balanced tree of height $\log_2(N/M)$ requires that each receiver should store $(\log_2(N/M))^{1.5}+1$ keys (assuming appropriate values of N and M), and taking into account limit on total number of keys at a receiver, it appears that parameter M should be the smallest integer such that the following inequality holds,

$$\log_2 N \geqslant (\log_2(N/M))^{1.5}+1. \tag{6.1}$$

Assuming, for simplicity, the equality in (6.1), the following assigning and processing of the secret bits is employed.

- Each receiver stores the secret bits in form of the keys required for Tree-B with LSD.
- Required mapping of the keys for employment over Tree-A with CST is performed based on dedicated one-way (collisionful) hash functions.

According to the previous, and taking into account the results reported in [99] and [83], it can be shown that the considered reconfigurable key management has the following main characteristics.

Proposition 6.1. *The considered reconfigurable key management requires the following overhead for R revocations in total assuming that $R=\sum_{m=1}^{M} r_m$, where r_m is the number of revocations corresponding to the m-th subtree in Tree-B structure:*

- *dimension of the storage needed by a receiver: $O(\log_2 N)$;*
- *dimension of the communication overhead:* $\min\{O(R\log_2(N/R)); O(R)+\sum_{m=1}^{M}\delta_{0,r_m}\}$;
- *dimension of the processing at receiver overhead:*

$O(\log_2 log_2 N)$ *if* $\min\{O(R\log_2(N/R)); O(R)+\sum_{m=1}^{M}\delta_{0,r_m}\}=O(R\log_2(N/R))$, *or* $O(\log_2(N/M))$ *if* $\min\{O(R\log_2(N/R)); O(R)+\sum_{m=1}^{M}\delta_{0,r_m}\}=O(R)+\sum_{m=1}^{M}\delta_{0,r_m}$; *where $\delta_{0,x}$ takes value 1 for $x=0$ and 0 otherwise.*

Sketch of the Proof. According to [99], when Tree-A with CST is employed, the key management requires the following overhead:

(i) dimension of the communication overhead-$O(R\log_2(N/R))$;

(ii) dimension of the processing at receiver overhead-$O(\log_2\log_2 N)$;

(iii) dimension of the storage needed by a receiver-$O(\log_2 N)$.

When Tree-B with LSD is employed we have the following. Revocation of r_m receivers corresponding to the m-th subtree requires the communication overhead of $O(r_m)$. Accordingly, the communication overhead for revocation of all $R=\sum_{m=1}^{M} r_m$ receivers is $O(R)$. Also, the new session key should be sent, as well, to all clusters of users where no one revocation has been made, and this rekeying requires additional $\sum_{m=1}^{M}\delta_{0,r_m}$ messages. On the other hand, when LSD based key

management is employed over a subtree of height $\log_2(N/M)$, according to[83], the processing at receiver overhead is no more than $O(\log_2(N/M))$.

Finally, note that at each instant of time, according to the current revocation request, the center selects the underlying structure which minimizes the communication overhead. The above, and taking into account (6.1), yields the proposition statement.

Accordingly, based on the characteristics of CST, SD and LSD reported in [99] and [83], a comparison of these schemes and the considered RLKH is summarized in the following Table 5 assuming N receivers and $R = \sum_{m=1}^{M} r_m$ revocations, $N > R$, $r_m \geqslant 0$, $m = 1, 2, \ldots, M$, that the parameters N, M are related by (6.1), and that $\delta_{0,x}$ takes value 1 for $x = 0$ and 0 otherwise.

TABLE 5. Comparison of the considered BE schemes.

	storage at receiver	processing at receiver	communication
CST [99]	$O(\log_2 N)$	$O(\log_2 \log_2 N)$	$O(R \log_2 \frac{N}{R})$
SD [99]	$O((\log_2 N)^2)$	$O(\log_2 N)$	$O(R)$
basic LSD [83]	$O((\log_2 N)^{1.5})$	$O(\log_2 N)$	$O(R)$
proposed RLKH	$O(\log_2 N)$	$\leqslant$ $O(\log_2(N/M))$	$\min\{O(R \log_2(N/R)); O(R) + \sum_{m=1}^{M} \delta_{0,r_m}\}$

6.1.4. Discussion. The main characteristics of the up to now reported key management schemes include employment of a static underlying structure for the key management, and addressing the subset covering problem over the entire underlying structure. Oppositely, the main underlying ideas for developing of the improved key management schemes based on RLKH include the following: (i) employment of a reconfigurable underlying structure; and (ii) in a general case employment of a divide-and-conquer approach over the underlying structure. RKLH appears as a very suitable approach for highly dynamic revocation scenarios.

Employment of RLKH for a SEK updating requires just a slight (almost negligible) increase of required processing at the both sides, at the center and at the receiver. On the other hand, RLKH requires a moderate processing at the center side in order to establish the system, but this operation should be done just once.

6.2. A Broadcast Encryption Approach Based on Coding.

6.2.1. Introduction. As already pointed out in Section 5.1, the KEKs are used to encrypt and deliver the updated SEK to the valid members of the group. In order to obtain the desired security, the KEKs must be kept in a protected storage called the secret storage. This storage should be as small as possible to allow an efficient implementation.

Generally, the employment of a key management scheme in a communications system introduces certain system overheads. The main ones are: (i) the required (protected) secret storage at the receiver; (ii) the required public storage at the receiver; (iii) the processing overhead at the receiver; (iv) the communications overhead. Different trade-offs between the overheads are possible. Some of these overheads are discussed in [72] and [102], for example, but the objective of this letter is towards different trade-offs.

A framework for minimization of the secret storage based on the secret-public storage trade-off has been reported in [17]. This section, according to [15] considers an alternative approach for minimizing the secret storage in certain key management schemes employing an appropriate trade-off between the required secret storage, the public storage and the processing overheads. The proposal employs a technique for one-way mapping of the secret bits whose security originates from the uncertainty associated with decoding a binary block code after transmission over a binary erasure channel.

A number of recently proposed key management schemes for broadcast encryption (SD [99], LSD [83], and reconfigurable key management schemes [19]-[18]) require a significant amount of secret data to be stored at a receiver. This constraint appears to be inappropriate in certain scenarios implying the need for small secret storage overhead. Consequently assigning multiple roles to the same secret bits via one-way mapping is required. A motivation for this work is to propose certain key management schemes with minimized secret storage employing a low complexity mapping technique. As a result, an implementation based only on very simple arithmetic and logical operations like mod2 additions and look-up table operations is possible. An additional motivation for this work is to yield appropriate techniques required for reconfigurable key management (RKM) [19]-[18], and to support the generic framework of assigning different roles to the secret key bits.

6.2.2. A Framework for Minimization of the Secret Storage. Following [17], this section provides a general framework for minimization of the required secret storage: This framework is based on the employment of a secret seed and its mapping into the required KEKs. Particularly note that this approach is very different from the one which employs encryption of KEKs by a master secret key and storing the encrypted forms of the KEKs in a public storage. The considered approach is not based on encryption of KEKs. Accordingly it does not require exposition of any transformation of KEKs in a public storage and its security does not depend on the security of any encryption technique, implying a number of related advantages.

Suppose that a key management scheme with non-updatable keys requires that the following I KEKs are stored in a secret (protected) storage at a receiver: $KEK_1, KEK_2, \ldots, KEK_I$.

Let: $f(\cdot)$ and $g(\cdot)$ be functions which fulfill certain requirements; S be arbitrary data; (R_i, Q_i), $i = 1, 2, \ldots, I$, be certain data such that the following holds:

$$KEK_i = g(f(S, R_i), Q_i), \quad i = 1, 2, \ldots, I. \tag{6.2}$$

Assuming that the composition of $f(\cdot)$ and $g(\cdot)$ yields the appropriate one-wayness, instead of keeping all $KEKs$ in a protected storage, the following strategy can be employed:
–keep S in a protected storage which provides the data secrecy;
–keep (R_i, Q_i), $i = 1, 2, \ldots, I$, in a public storage;
–when required, calculate any KEK_i, employing (6.2).

The main architectural components for the implementation of the above method are the following: (i) temper-proof block T for the seed S and the mappings $f(\cdot)$ and $g(\cdot)$; and (ii) public storage for the non-secret data R_i, Q_i, $i = 1, 2, \ldots, I$. The block T has two inputs and one output. The role of T is to yield KEK_i as its output for the given input pair (R_i, Q_i) preserving the secrecy of the secret seed S in a computationally secure manner.

6.2.3. Dedicated One-Way Mapping for Secret-to-Public Storage Exchange. Let S be a binary k-dimensional vector, and let K_i, $i = 1, 2, \ldots, I$, be I different binary n-dimensional vectors, $k \geqslant n$. A goal is to propose certain methods for mapping the vector S into any of the vectors K_i, $i = 1, 2, \ldots, I$, under the following conditions:

- it is computationally infeasible to recover S knowing all K_i, $i = 1, 2, \ldots, I$, and all the related public information;
- the mapping of S into any K_i should *not* include public key cryptography;
- the mapping of S into any K_i should be a low complexity one and include only mod2 additions and simple logical operations.

Preliminaries For any input vector, a communication channel with erasures yields a vector of the same dimension in which a certain fraction of the symbols is no longer known due to erasures. Accordingly, the output of a binary erasure channel (BEC) yields exact information on the the input bits in a certain subset and no information on the erased e bits outside this subset.

The list decoding problem for a binary error-correcting code consists of outputting the list of all codewords that lie within a certain Hamming distance of the received word. The decoding is considered successful if and only if the correct codeword is included in the list.

Mapping Specification Let C be an (m, k) block code which maps k information bits into a codeword of length m.

For $i = 1, 2, \ldots, I$, let the public information associated to each K_i be in the form of a pair of binary vectors (R_i, E_i) of dimensions n and m, respectively, where:

- R_i is selected randomly, and
- for a given S, the vector E_i is selected so that the non erased bits of the codeword generated by $\phi(S, R_i)$ yield K_i, where $\phi(\cdot)$ is a suitable nonlinear function.

Mapping E (where E corresponds to the erasure channel) is defined as follows:

(1) employing C, perform encoding of the vector $\phi(S, R_i)$ into the codeword C_{S,R_i} which is a binary m-dimensional vector;
(2) employing the vector E_i erase $e = m - n$ bits in the codeword C_{S,R_i};
(3) define K_i as the consecutive sequence of nonerased bits.

Proposition 6.2. *Assuming sufficiently large values for the parameters m and $e = m - n > m/2$, Mapping E can map any S into K_i, with probability close to 1.*

Proof. Given an m-dimensional binary vector, let $P(m, n)$ be the probability that one random n-tuple can be embedded into the given vector. Then we have the following (see [82], as well):

$$P(m,n) = \sum_{l=0}^{m-n} 2^{-(n+l)} \binom{n+l-1}{l} = 1 - 2^{-m} \sum_{l=0}^{n-1} \binom{m}{l}$$
$$\geqslant 1 - 2^{-m(1-h((n-1)/m))}$$

where $h(\cdot)$ is the binary entropy function. □

Proposition 6.3. *When $k > 2n + \delta$, $\delta > 0$, the complexity of recovering any K_i is proportional to 2^n and the complexity of recovering S is proportional to $2^{n+\delta}$, given all other vectors K_j, $j \neq i$, $j = 1, 2, \ldots, I$, and all public information.*

Proof. Any unknown K_i can be recovered via simple guessing which has complexity proportional to 2^n, or via recovering of S and employment of Mapping E for obtaining K_i. The complexity of recovering S is determined by the following consideration. The capacity $C(\epsilon)$ of a BEC with erasure probability $\epsilon = e/m$ is given by $C(\epsilon) = 1 - e/m = n/m$. The code rate k/m is always greater than the capacity for $k > n$, implying that unique decoding is not feasible. Then, the best possible is list decoding which is equivalent to the following algorithm:

(i) fix certain $k^* \leqslant k$ bits so that the code rate of the modified code is below the capacity, i.e. $(k - k^*)/m \leqslant C(\epsilon)$;

(ii) generate a list for 2^{k^*} decodings.

When $k > 2n + \delta$, the list dimension becomes greater than $2^{n+\delta}$ implying that it is greater than the number of hypotheses required by the direct exhaustive search for any K_i. Consequently, the simple guessing approach provides the lower bound on the recovering complexity for any K_i. □

Proposition 6.4. *The implementation complexity is proportional to nk.*

Proof. In Mapping E, the implementation complexity mainly depends on step 1, i.e. encoding of S into the n codeword coordinates of C_{S,R_i} corresponding to the bits which are not erased. This requires a number of mod2 additions upperbounded by nk. □

6.2.4. SD and LSD Key Managements with Minimized Secret Storage. This section proposes and discusses modified SD [99] and LSD [83] key management schemes based on the proposed Mapping E. For further considerations we assume that the employment of the original SD and LSD schemes requires that each receiver stores in secret storage a sequence of n-dimensional binary vectors K_i, $i = 1, 2, \ldots, I$. Assuming that $I^{(SD)}$ and $I^{(LSD)}$ denote the values of I related to SD and basic LSD, respectively, we have (see [99] and [83]):

$$I^{(SD)} = \frac{1}{2}[(\log_2 N)^2 + \log_2 N] + 1,$$

$$I^{(LSD)} = (\log_2 N)^{3/2} + 1,$$

where N denotes the number of users in the system.

Modified SD and LSD: We propose modified versions of SD and LSD as follows:

• In the modified SD/LSD, each receiver keeps in the secret storage the seed S in the form of a $3n$-dimensional binary vector, and employing Mapping E evaluates any of the required $I^{(SD)}/I^{(LSD)}$ vectors; all other issues are identical to that of the original SD/LSD.

Recall that the employment of Mapping E requires that certain information is stored in public storage and that certain processing is employed. Also note that the proposed modification does not affect the communication overhead (i.e., it is same as in the original schemes).

Security of Modified SD and LSD: The original SD and LSD schemes are only computationally secure ones, and the complexity of step by step straightforward recovering of all employed KEKs is upperbounded by $I^{(SD)}2^n$ and $I^{(LSD)}2^n$, respectively. On the other hand, the only security difference between the original and the modified schemes is that an attempt to recover the employed KEKs in the modified schemes could be done either directly employing the same complexity as in the original schemes, or via recovering S. Note that when $k = 3n$, Proposition 2 implies that the complexity of recovering S is proportional to 2^{2n}. Hence it is always greater than the upperbound of step-by-step recovering of all the KEKs because $2^n \gg I^{(SD)} > I^{(LSD)}$. Accordingly, Proposition 2, the nature of the modification, and the selected dimension of S directly imply that Mapping E preserves the security of the original schemes.

Comparison of Modified and Original SD and LSD: According to the results reported in [99] and [83], the nature of the modifications and the characteristics of Mapping E, Tables 6 and 7 provide a summary comparison of the main overheads regarding the modified and original SD and LSD, respectively, assuming a group of N users and that certain R users should be revoked.

TABLE 6. Comparison of original and modified SD schemes.

	proposed SD (Mapping E)	original SD [99]
secret storage overhead at receiver	$O(1)$	$O((\log_2 N)^2)$
public storage overhead at receiver	$O((\log_2 N)^2)$	–
processing overhead at receiver	$nk + O(\log_2 N)$	$O(\log_2 N)$
communications overhead	$O(R)$	$O(R)$

TABLE 7. Comparison of original and modified LSD schemes.

	proposed LSD (Mapping E)	original LSD [83]
secret storage overhead at receiver	$O(1)$	$O((\log_2 N)^{3/2})$
public storage overhead at receiver	$O((\log_2 N)^{3/2})$	–
processing overhead at receiver	$nk \ + \ O(\log_2 N)$	$O(\log_2 N)$
communications overhead	$O(R)$	$O(R)$

6.2.5. Concluding Remarks. A generic framework and a particular mapping technique have been pointed out for minimization of the required secret storage in certain key management schemes for broadcast encryption. The proposed modified SD and LSD based key management schemes require secret storage overhead independent of the parameter N of only $O(1)$, public storage overheads $O((\log_2 N)^2)$ and $O((\log_2 N)^{3/2})$, respectively, and a low additional processing overhead. For example, when the number of users is about one million, i.e. $N = 2^{20}$, and assuming that each KEK consists of 100 bits, the modified SD/LSD schemes require only 300 secret bits, while the original SD and LSD schemes require approximately 20000 and 9000 secret bits, respectively.

Finally note that in [19], with an additional refinement in [18], as well as discussed in Section 6.1, RKM has been proposed as an advanced technique for broadcast encryption (appropriate for high dynamic scenarios). RKM assumes that the secret bits can play different roles, and that the employed volume of secret information is as small as possible. Accordingly, the proposed method for minimizing the secret storage is also of direct interest for RKM.

7. Concluding Discussion

The considerations in this chapter could also be employed as particular guidelines for future research directions because they point out and provide background for further work regarding the following active and important topics of cryptology:

- security evaluation of stream ciphers and authentication protocols employing decoding techniques and algorithms for the LPN problem;
- design of advanced encryption techniques based on employment of coding theory and randomness moving towards provable secure encryption in information-theoretic sense;
- security evaluation and design of certain key management schemes based on the broadcast encryption paradigm.

Accordingly, note the following issues which support the above claims.

Significance of solving consistent and overdefined systems of algebraic equations which are true with certain probability (algebraic equations corrupted by noise) is

a well recognized mathematical topic of the LPN and LPN-related problems, and of direct interest for decoding of linear block codes and for security evaluation of certain cryptographic techniques.

Coding theory has accumulated a huge amount of results of potential interest for design of advanced encryption schemes based on employment of pseudorandomness, randomness and dedicated coding. This approach also provides a framework for employment of certain results of information theory for achieving security which could be provable in information-theoretic sense.

Particular open research problems related to the topics discussed in this chapter include the following ones:

- Solving probabilistic systems of algebraic equations over GF(2);
- developing joint homophonic and error-correction coding schemes or the wire-tap channel coding schemes dedicated to particular encryption and authentication paradigms;
- developing graphs dedicated to certain subset-covering problems.

The results of the above addressed research problems could be employed for developing advanced cryptographic techniques regarding the following:

- Security evaluation of certain cryptographic primitives for encryption and authentication;
- developing of advanced encryption and authentication techniques which provide low implementation complexity and high level of provable security;
- developing of advanced key management schemes which provide low overhead to the system which employs these cryptographic keys.

Through an in details consideration of selected techniques for security evaluation and design of certain cryptographic primitives this chapter provides a background for further research activities as well as textbooks-like introduction of important topics of cryptology.

Particularly, this chapter points out to a number of mathematical techniques for addressing different problems of developing basic components for cyber-security issues.

Acknowledgement. This work is partially supported by the Serbian Ministry of Education and Science, the projects ON174008, "Advanced Techniques of Cryptology, Image Processing and Computational Topology for Information Security", and III44006, "Development of new information and communication technologies, based on advanced mathematical methods, with applications in medicine, telecommunications, power systems, protection of national heritage and education".

References

[1] M. J. Mihaljević, S. Gangopadhyay, G. Paul and H. Imai, *State Recovery of Grain-v1 Employing Normality Order of the Filter Function*, IET Information Security 6, no. 2, pp. 55–64, June 2012.

[2] M. J. Mihaljević, S. Gangopadhyay, G. Paul and H. Imai, *Internal state recovery of keystream generator LILI-128 based on a novel weakness of the employed Boolean function*, Information Processing Letters 112, pp. 805–810, Nov. 2012.

[3] M. J. Mihaljević, S. Gangopadhyay, G. Paul and H. Imai, *Generic Cryptographic Weakness of k-normal Boolean Functions in Certain Stream Ciphers and Cryptanalysis of Grain-128*, Periodica Mathematica Hungarica (Springer), vol. 65, pp. 205–227, Dec. 2012.

[4] M. J. Mihaljević and H. Imai, *An approach for stream ciphers design based on joint computing over random and secret data*, Computing 85, no. 1–2, pp. 153–168, June 2009.

[5] M. J. Mihaljević, M. Fossorier and H. Imai, *Security Evaluation of Certain Broadcast Encryption Schemes Employing a Generalized Time-Memory-Data Trade-Off*, IEEE Communications Letters 11, no. 12, pp. 988–990, Dec. 2007.

[6] M. P. C. Fossorier, M. J. Mihaljević and H. Imai, *Modeling Block Encoding Approaches for Fast Correlation Attack*, IEEE Transactions on Information Theory 53, no. 12, pp. 4728–4737, Dec. 2007.

[7] M. J. Mihaljević, *Generic framework for secure Yuen 2000 quantum-encryption employing the wire-tap channel approach*, Physical Review A 75, no. 5, pp. 052334–1-5, May 2007.

[8] M. J. Mihaljević, M. P. C. Fossorier and H.Imai, *Birthday Paradox Based Security Analysis of Certain Broadcast Encryption Schemes*, IEICE Trans. Fundamentals E90-A, pp. 1248–1251, June 2007.

[9] M. P. C. Fossorier, M. J. Mihaljević, H. Imai, Y. Cui and K. Matsuura, *An Algorithm for Solving the LPN Problem and its Application to Security Evaluation of the HB Protocols for RFID Authentication*, Lect. Notes Comput. Sci. 4329, pp. 48–62, Dec. 2006.

[10] M. J. Mihaljević, M. P. C. Fossorier and H. Imai, *Security Weaknesses of Certain Broadcast Encryption Schemes*, DRMtics2005, Lect. Notes Comput. Sci. 3919, pp. 228–245, 2006.

[11] M. J. Mihaljević, M. Fossorier and H. Imai, *A Novel Broadcast Encryption Based on Time-Bound Cryptographic Keys*, DRMtics2005, Lect. Notes Comput. Sci. 3919, pp. 258–276, July 2006.

[12] J. Wang, M. J. Mihaljević, L. Harn, and H. Imai, *A Hierarchical Key Management Approach for Secure Multicast*, ARCS2006, Lect. Notes Comput. Sci. 3894, pp. 422–434, March 2006.

[13] M. J. Mihaljević, M. P. C. Fossorier and H. Imai, *A General Formulation of Algebraic and Fast Correlation Attacks Based on Dedicated Sample Decimation*, AAECC2006, Lect. Notes Comput. Sci. 3857, pp. 203–214, 2006.

[14] M. J. Mihaljević, M. P. C. Fossorier and H. Imai, *Cryptanalysis of keystream generator by decimated sample based algebraic and fast correlation attacks*, INDOCRYPT2005, Lect. Notes Comput. Sci. 3797, pp. 155–168, Dec. 2005.

[15] M. J. Mihaljević, M. Fossorier and H. Imai, *Key management with minimized secret storage employing an erasure channel approach*, IEEE Communications Letters 9, no. 8, pp. 741–743, Aug. 2005.

[16] M. J. Mihaljević and H. Imai, *The decimated sample based improved algebraic attacks on nonlinear filters*, SCN 2004, Lect. Notes Comput. Sci. 3352, pp. 310–323, Jan. 2005.

[17] M. J. Mihaljević, M. Fossorier and H. Imai, *Secret-public storage trade-off for broadcast encryption key management*, ICICS 2004, Lect. Notes Comput. Sci. 3269, pp. 375–387, October 2004.

[18] M. J. Mihaljević, *Reconfigurable key management for broadcast encryption*, IEEE Communications Letters 8, pp. 440–442, July 2004.

[19] M. J. Mihaljević, *Key management schemes for stateless receivers based on time varying heterogeneous logical key hierarchy*, ASIACRYPT 2003, Lect. Notes Comput. Sci. 2894, pp. 137–154, Dec. 2003.

[20] M. J. Mihaljević, *On vulnerabilities and improvements of Fast Encryption Algorithm for Multimedia FEA-M*, IEEE Trans. Cons. Electr. 49, no. 4, pp. 1199–1207, Nov. 2003.

[21] M. Mihaljević, *Broadcast encryption schemes based on the sectioned key tree*, ICICS2003, Lect. Notes Comput. Sci. 2836, pp. 158–169, Oct. 2003.

[22] P. Camion, M. J. Mihaljević and H. Imai, *Two alerts for design of certain stream ciphers: Trapped LFSR and weak resilient function over GF(q)*, SAC2002, Lect. Notes Comput. Sci. 2595, pp. 196–213, Feb. 2003.

[23] L. Michael, M. J. Mihaljević, S. Haruyama and R. Kohno, *Security issues for software defined radio: Design of a secure download system*, IEICE Transactions on Communications E85-B, pp. 2588–2600, Dec. 2002.

[24] M. J. Mihaljević and R. Kohno, *Cryptanalysis of fast encryption algorithm for multimedia FEA-M*, IEEE Communications Letters 6, pp. 382–384, Sept. 2002.

[25] L. Michael, M. J. Mihaljević, S. Haruyama and R. Kohno, *A framework for secure download for software defined radio*, IEEE Communications Magazine 40, no. 7, pp. 88–96, July 2002.

[26] M. J. Mihaljević, M. P. C. Fossorier and H. Imai, *Fast Correlation Attack Algorithm with List Decoding and an Application*, FSE2001, Lect. Notes Comput. Sci. 2355, pp. 196–210, 2002.

[27] M. J. Mihaljević and H. Imai, *Cryptanalysis of TOYOCRYPT-HS1 stream cipher*, IEICE Transactions on Fundamentals E85-A, pp. 66–73, Jan. 2002.

[28] M. J. Mihaljević, M. P. C. Fossorier and H. Imai, *On decoding techniques for cryptanalysis of certain encryption algorithms*, IEICE Transactions on Fundamentals E84-A, pp. 919–930, Apr. 2001.

[29] M. J. Mihaljević, M. P. C. Fossorier and H. Imai, *A low-complexity and high-performance algorithm for the fast correlation attack*, FSE2000, Lect. Notes Comput. Sci. 1978, pp. 196–212, 2001.

[30] M. J. Mihaljević, M. P. C. Fossorier and H. Imai, *An algorithm for cryptanalysis of certain keystream generators suitable for high-speed software and hardware implementations*, IEICE Transactions on Fundamentals E84-A, pp. 311–318, Jan. 2001.

[31] M. J. Mihaljević and J. Golić, *A method for convergence analysis of iterative probabilistic decoding*, IEEE Transactions on Information Theory 46, pp. 2206–2211, Sept. 2000.

[32] M. P. C. Fossorier, M. J. Mihaljević and H. Imai, *Reduced complexity iterative decoding of Low Density Parity Check codes based on Belief Propagation*, IEEE Transactions on Communications 47, pp. 673–680, May 1999.

[33] M. P. C. Fossorier, M. J. Mihaljević and H. Imai, *Critical noise for convergence of iterative probabilistic decoding with belief propagation in cryptographic applications*, Applied Algebra, Algebraic Algorithms and Error Correcting Codes-AAECC 13, Lect. Notes Comput. Sci. 1719, pp. 282–293, 1999.

[34] M. J. Mihaljević and H. Imai, *A family of fast keystream generators based on programmable linear cellular automata over GF(q) and time variant table*, IEICE Transactions on Fundamentals E82-A, pp. 32–39, Jan. 1999.

[35] M. Mihaljević, Y. Zheng and H. Imai, *A family of fast dedicated one-way hash functions based on linear cellular automata over GF(q)*, IEICE Transactions on Fundamentals E82-A, pp. 40–47, Jan. 1999.

[36] M. Mihaljević, Y. Zheng and H. Imai, *A cellular automaton based fast one-way hash function suitable for hardware implementation*, PKC'98, Lect. Notes Comput. Sci. 1431, pp. 217–233, 1998.

[37] M. Mihaljević, *An improved key stream generator based on the programmable cellular automata*, ICICS'97, Lect. Notes Comput. Sci. 1334, pp. 181–191, 1997.

[38] M. J. Mihaljević, *Security examination of a cellular automata based pseudorandom bit generator using an algebraic replica approach*, AAECC12, Lect. Notes Comput. Sci. 1255, pp. 250–262, 1997.

[39] M. J. Mihaljević, *An iterative probabilistic decoding algorithm for binary linear block codes beyond the half minimum distance*, AAECC12, Lect. Notes Comput. Sci. 1255, pp. 237–249, 1997.

[40] M. J. Mihaljević, *A faster cryptanalysis of the self-shrinking generator*, ACISP'96, Lect. Notes Comput. Sci. 1172, pp. 182–188, 1996.

[41] M. J. Mihaljević, *A sequence comparison approach for decoding of general binary block codes after the binary symmetric channel with synchronization errors*, Zeitschrift fur Angewandte Mathematik und Mechanik-ZAMM 76, pp. 479–481, 1996.

[42] M. J. Mihaljević, *A correlation attack on the binary sequence generators with time-varying output function*, ASIACRYPT'94, Lect. Notes Comput. Sci. 917, pp. 67–79, 1995.
[43] M. J. Mihaljević, *On message protection in crypto systems modeled as the generalized wire-tap channel II*, Lect. Notes Comput. Sci. 829, pp. 13–24, 1994.
[44] M. J. Mihaljević, *An approach to the initial state reconstruction of a clock-controlled shift register based on a novel distance measure*, AUSCRYPT'92, Lect. Notes Comput. Sci. 718, pp. 349–356, 1993.
[45] M. J. Mihaljević and J. Golić, *Convergence of a Bayesian iterative error-correction procedure on a noisy shift register sequence*, EUROCRYPT'92, Lect. Notes Comput. Sci. 658, pp. 124–137, 1993. (reprinted in Lect. Notes Comput. Sci. 1440, 1999)
[46] M. J. Mihaljević and J.Dj. Golić, *A comparison of cryptanalytic principles based on iterative error-correction*, Advances in Cryptology-EUROCRYPT '91, Lect. Notes Comput. Sci. 547, pp. 527–531, 1991.
[47] J. Golić and M. Mihaljević, *A generalized correlation attack on a class of stream ciphers based on the Levenshtein distance*, J. of Cryptology 3, pp. 201–212, 1991.
[48] J. Golić and M. J. Mihaljević, *A noisy clock-controlled shift register cryptanalysis concept based on sequence comparison approach*, EUROCRYPT'90, Lect. Notes Comput. Sci. 473, pp. 487–491, 1991. (reprinted in Lect. Notes Comput. Sci. 1440, 1999)
[49] M. J. Mihaljević and J.Dj. Golić, *A fast iterative algorithm for a shift register initial state reconstruction given the noisy output sequence*, Advances in Cryptology-AUSCRYPT '90, Lect. Notes Comput. Sci. 453, pp. 165–175, 1990.
[50] J. Golić and M. J. Mihaljević, *Minimal linear equivalent analysis of a variable memory binary sequence generator*, IEEE Transactions on Information Theory 36, pp. 190–192, Jan. 1990.
[51] M. J. Mihaljević, *A Framework for Stream Ciphers Based on Pseudorandomness, Randomness and Error-Correcting Coding*, in Enhancing Cryptographic Primitives with Techniques from Error Correcting Codes, Editors B. Preneel, S. Dodunekov, V. Rijmen and S. Nikova, Vol. 23 in the NATO Science for Peace and Security Series-D: Information and Communication Security, pp. 117–139, IOS Press, Amsterdam, The Netherlands, June 2009.
[52] M. J. Mihaljević, *Decimation Based Correlation and Algebraic Attacks and Design of Boolean Functions*, in Boolean Functions in Cryptology and Information Security, Editors B. Preneel and O. A. Logachev, Vol. 18 in the Series Information and Communication Security, pp. 183–199, IOS Press, Amsterdam, The Netherlands, July 2008.
[53] M. Matsui and M. J. Mihaljević, *Security Evaluation Techniques for Symmetric Cryptography*, Chapter in Information Security Handbook, Editors H. Imai and E. Okamoto, IEICE, Ohmusha Ltd., Tokyo, Japan, pp. 145–160, 2004. (ISBN 4–274–07980–5)
[54] *Japan Patent JP 4863283*: M. J. Mihaljević and H. Watanabe, *Authentication System Using Light-Weight Authentication Protocol*, November 18, 2011.
[55] *United States Patent US 8023649*: M. J. Mihaljević and J. Abe, *Method and apparatus for cellular automata based generation of pseudorandom sequences with controllable period*, 21 pages, September 2011.
[56] *China Patent CN 1698306*: M. J. Mihaljevicć and J. Abe, *Data processing method*, 51 pages, October 2010.
[57] *Japan Patent JP 4432350*: M. J. Mihaljevic and J. Abe, *Data Processing Method, Program Thereof, Data Processor, and Receiver*, 35 pages, March 2010.
[58] *United States Patent US 7502941*: L. Michael and M. J. Mihaljević, *Wireless data communication method and apparatus for software download system*, 36 pages. March 2009.
[59] *Japan Patent JP 3918578*: R. Morelos-Zaragoza and M. J. Mihaljević, *Method and apparatus for loss correction and limited reception in streaming data, and data communication apparatus*, May 2007.
[60] M. J. Mihaljevic, *An Approach for Light-Weight Encryption Employing Dedicated Coding*, to appear in IEEE GLOBECOM 2012 Proceedings, 7 pages (Anaheim CA, USA, 03–07 Dec. 2012).

[61] M. J. Mihaljević and H. Imai, *A Security Evaluation of Certain Stream Ciphers which Involve Randomness and Coding*, *2010 IEEE Int. Symp. on Inform. Theory and its Appl.-ISITA 2010*, Taichung, Taiwan, Oct. 17–20, 2010, Proceedings, pp. 789–794, IEEE, 2010.

[62] M. P. C. Fossorier, M. J. Mihaljević and H. Imai, *Decimation Based Fast Correlation Attack*, *2007 IEEE Int. Symp. Inform. Theory-ISIT'2007*, Nice, France, June 24–29, 2007, Proceedings, pp. 456–460 (ISBN 1–4244–1429–6).

[63] M. P. C. Fossorier, M. J. Mihaljević and H. Imai, *A Unified Analysis for the Fast Correlation Attack*, *2005 IEEE Int. Symp. Inform. Theory-ISIT'2005*, Adelaide, Australia, Sept. 2005, Proceedings, pp. 2012–2015 (ISBN 0–7803–9151–9).

[64] Advanced Access Content System (AACS): Introduction and Common Cryptographic Elements, Feb. 2006. Available at http://www.aacsla.com.

[65] B. Applebaum, D. Cash, C. Peikert and A. Sahai, *Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems*, CRYPTO 2009, Lect. Notes Comput. Sci. 5677, pp. 595–618, Aug. 2009.

[66] J.-P. Aumasson, M. Finiasz, W. Meier, S. Vaudenay, *STCHo: A Hardware-Oriented Trapdoor Cipher*, ACISP 2007, Lect. Notes Comput. Sci. 4586, pp. 184–199, 2007.

[67] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, *Optimal decoding of linear codes for minimizing symbol error rate*, IEEE Trans. Inform. Theory IT-20, pp. 284–287, March 1974.

[68] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, *On the Inherent Intractability of Certain Coding Problems*, IEEE Trans. Info. Theory 24, pp. 384–386, 1978.

[69] A. Biryukov and A. Shamir, *Cryptanalytic time/memory/data tradeoffs for stream ciphers*, ASIACRYPT 2000, Lect. Notes Comput. Sci. 1976, pp. 1–13, 2000.

[70] A. Blum, M. Furst, M. Kearns and R. Lipton, *Cryptographic Primitives Based on Hard Learning Problems*, CRYPTO 1993, Lect. Notes Comput. Sci. 773, pp. 278291, 1994.

[71] A. Blum, A. Kalai and H. Wasserman, *Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model*, J. of the ACM 50, no. 4, pp. 506–519, July 2003.

[72] R. Canetti, T. Malkin and K. Nissim, *Efficient communication-storage tradeoffs for multicast encryption*, EUROCRYPT'99, Lect. Notes Comput. Sci. 1592, pp. 459–474, 1999.

[73] P. Chose, A. Joux and M. Mitton, *Fast Correlation Attacks: An Algorithmic Point of View*, EUROCRYPT2002, Lect. Notes Comput. Sci. 2332, pp. 209–221, 2002.

[74] A. Clark, J. Dj. Golić, and E. Dawson, *A comparison of fast correlation attacks*, Fast Software Encryption-FSE'96, Lect. Notes Comput. Sci. 1039, pp. 145–157, 1996.

[75] N. T. Courtois and W. Meier, *Algebraic attacks on stream ciphers with linear feedback*, EUROCRYPT'2003, Lect. Notes Comput. Sci. 2656, pp. 345–359, 2003.

[76] N. T. Courtois, *Fast algebraic attacks on stream ciphers with linear feedback*, CRYPTO'2003, Lect. Notes Comput. Sci. 2729, pp. 176–194, 2003.

[77] Ecrypt: Network of Excellence in Cryptology, EU FP6 Project, 2004–2008. http://www.ecrypt.eu.org.

[78] R. M. Fano, *Transmission of Information: A Statistical Theory of Communication*. New York: MIT, 1961.

[79] R. G. Gallager, *Low-density parity-check codes*, IRE Trans. Inform. Theory IT-8, pp.21–28, Jan. 1962.

[80] H. Gilbert, M. J. B. Robshaw and Y. Seurin, *HB$^{\#}$: Increasing the Security and Efficiency of HB^{+}*, EUROCRYPT2008, Lect. Notes Comput. Sci. 4965, pp. 361–378, 2008.

[81] H. Gilbert, M. J. B. Robshaw, and Y. Seurin, *How to Encrypt with the LPN Problem*, ICALP 2008, Part II, Lect. Notes Comput. Sci. 5126, pp. 679–690, 2008.

[82] J.Dj. Golić and L. O'Connor, *Embedding and probabilistic correlation attacks on clock-controilled shift registers*, EUROCRYPT'94, Lect. Notes Comput. Sci. 950, pp. 230–243, 1995.

[83] D. Halevy and A. Shamir, *The LCD broadcast encryption scheme*, CRYPTO 2002, Lect. Notes Comput. Sci. 2442, pp. 47–60, 2002.

[84] M. E. Hellman, *A cryptanalytic time-memory trade-off*, IEEE Transactions on Information Theory 26, pp. 401–406, July 1980.

[85] ISO/IEC Standard 18033–3:2005.
[86] H. N. Jendal, Y. J. B. Kuhn, and J. L. Massey, *An information-theoretic treatment of homophonic substitution*, EUROCRYPT'89, Lect. Notes Comput. Sci. 434, pp. 382–394, 1990.
[87] T. Johansson and F. Jonsson, *Improved fast correlation attacks on stream ciphers via convolutional codes*, Advances in Cryptology-EUROCRYPT'99, Lect. Notes Comput. Sci. 1592, pp. 347–362, 1999.
[88] T. Johansson and F. Jonsson, *Fast correlation attacks based on turbo code techniques*, Advances in Cryptology-CRYPTO'99, Lect. Notes Comput. Sci. 1666, pp. 181–197, 1999.
[89] O. Kara and I. Erguler, I, *A New Approach to Keystream Based Cryptosystems*, SASC 2008, Workshop Record, pp. 205–221, 2008.
[90] J. Katz, *Efficient Cryptographic Protocols Based on the Hardness of Learning Parity with Noise*, Cryptography and Coding 2007, Lect. Notes Comput. Sci. 4887, pp. 1–15, 2007.
[91] V. A. Kovalevskij, *The problem of character recognition from the point of view of mathematical statistics*, in Character Readers and Pattern Recognition, pp. 3–30, New York: Spartan, 1968. Russian edition 1965.
[92] E. Levieil and P.-A. Fouque, *An Improved LPN Algorithm*, SCN 2006, Lect. Notes Comput. Sci. 4116, pp. 348–359, 2006.
[93] J. Lotspiech, S. Nusser and F. Prestoni, *Broadcast encryption's bright future*, IEEE Computer 35, pp. 57–63, Aug. 2002.
[94] J. Lotspiech, V. Mirles, D. Naor and I. Nin, *Coincidence-free media key block for content protection for recordable media*, United States Patent 6,883,097, filed May 2000.
[95] J. Massey, *Some Applications of Source Coding in Cryptography*, European Transactions on Telecommunications 5, pp. 421–429, July-August 1994.
[96] R. J. McEliece, *A public key cryptosystem based on algebraic coding theory*, DSN progress report 42–44, pp. 114–116, 1978.
[97] W. Meier and O. Staffelbach, *Fast Correlation Attacks on Certain Stream Ciphers*, J. of Cryptology 1, pp. 159–176, 1989.
[98] A. Menezes, P. C. van Oorschot and S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Roton, 1997.
[99] D. Naor, M. Naor and J. Lotspiech, *Revocation and tracing schemes for stateless receivers*, CRYPTO 2001, Lect. Notes Comput. Sci. 2139, pp. 41–62, Aug. 2001.
[100] D. Naor and M. Naor, *Protecting cryptographic keys: The trace-and-revoke approach*, IEEE Computer 36, pp. 47–53, July 2003.
[101] B. Pinkas, *Efficient state updates for key management*, Proc. IEEE 92, pp. 910–917, June 2004.
[102] R. Poovendran and C. Bernstein, *Design of secure multicast key management schemes with communication budget constraint*, IEEE Commun. Lett. 6, pp. 108–110, March 2002.
[103] R. Rivest and T. Sherman, *Randomized Encryption Techniques*, Advances in Cryptology: Proceedings of CRYPTO '82, Plemum, New Yourk, pp. 145–163, 1983.
[104] C. E. Shannon, *Communication theory of secrecy systems*, Bell Systems Technical J. 28, pp. 656–715, Oct. 1949.
[105] N. J. A. Sloane, *Error-correcting codes and cryptography–part I*, Cryptologia 6, pp. 128–153, 1982.
[106] T. Siegenthaler, *Decrypting a Class of Stream Ciphers Using Ciphertext Only*, IEEE Transactions on Compututers C-34, pp. 81–85, 1985.
[107] C. K. Wong, M. Gouda, and S. S. Lam, *Secure group communications using key graphs*, IEEE/ACM Trans. Networking 8, pp. 16–31, Feb. 2000.
[108] A. D. Wyner, *The wire-tap channel*, Bell Systems Technical J. 54, pp. 1355–1387, 1975.
[109] K. Zeng and M. Huang, *On the linear syndrome method in cryptanalysis*, Advances in Cryptology-CRYPTO '88, Lect. Notes Comput. Sci. 403, pp. 469–478, 1990.
[110] W. T. Zhu, *Optimizing the tree structure in secure multicast key management*, IEEE Commun. Lett. 9, (5), pp. 477–479, May 2005.