IET Information Security

Research Article

IET Journals
The Institution of Engineering and Technology

Security evaluation and design elements for a class of randomised encryptions

ISSN 1751-8709
Received on 28th May 2017
Revised 21st March 2018
Accepted on 21st July 2018
E-First on 31st August 2018
doi: 10.1049/iet-ifs.2017.0271
www.ietdl.org

Miodrag J. Mihaljević[1] ✉, *Frédérique Oggier*[2]

[1]*Mathematical Institute, Serbian Academy of Sciences and Arts, Belgrade, Serbia*
[2]*Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore*
✉ *E-mail: miodragm@turing.mi.sanu.ac.rs*

Abstract: This study considers a class of randomised encryption techniques, where the encrypted data suffers from noise through transmission over a communication channel. It focuses on the encoding–encryption framework, where the data is first encoded using error correction coding for reliability, then encrypted with a stream cipher. A dedicated homophonic encoder is added to enhance the protection of the stream cipher key, on which relies the security of all the system transmissions. This study presents a security evaluation of such systems in a chosen plaintext attack scenario, which shows that the computational complexity security is lower bounded by the related LPN (learning from parity with noise) complexity in both the average and worst cases. This gives guidelines to construct a dedicated homophonic encoder which maximises the complexity of the underlying LPN problem for a given encoding overhead. A generic homophonic coding strategy that fulfils the proposed design criteria is then given, which thus both enhances security while minimising the induced overhead. Finally, a comparison of encryption schemes based on the LPN problem with and without homophonic coding is considered.

1 Introduction

Randomised encryption techniques form a class of encryption procedures, where a message is encrypted by randomly choosing a ciphertext inside a set of ciphertexts, that corresponds to the message under a given current encryption key [1]. Examples of randomised encryption schemes include, for example [2–7]. Another class of cryptographic primitives which uses pure randomness to provide security is that of wiretap coding [8], where randomness and a dedicated coding at a transmitter are combined to enhance the security of data transmission over noisy channels.

This paper considers randomised encryptions where the encrypted data will then be transmitted over a noisy channel, and the security is enhanced by the addition of a dedicated homophonic encoding, similarly to the techniques used in the context of wiretap coding. We focus on the *encoding–encryption* framework, where the communication system first encodes the data, and then encrypts it. A famous illustrative example of the encoding–encryption paradigm is GSM (global system for mobile communications), the standard for mobile telephony (see [9, 10], for the coding, respectively, security details). Encryption–decryption is done through a stream cipher, since the receiver needs to first decrypt the data despite the noise, before performing the decoding, which cannot be done through block ciphers. Consequently, the security of the system relies crucially on the private key used in the randomised stream cipher, and thus when we refer to the security of systems based on the encoding–encryption paradigm, we implicitly mean the security of the keystream generator and the users' secret key.

Motivation for the work: Homophonic coding (see [11–13]) is a natural technique to increase the security of systems using the encoding–encryption paradigm, since it injects extra randomness in the system, which increases the confusion of a possible adversary by amplifying the channel noise that he experiences. This idea has been exploited in [6], for the same class of randomised encryption schemes as that considered in this work. It was shown from an information-theoretic view point that with a dedicated homophonic encoder, the amount of uncertainty about the secret key, given the information that an adversary could gather during different passive or active attacks he can mount, is a decreasing function of the samples available for cryptanalysis. This means that there is a threshold before which the homophonic encoding does provide a level of unconditional security, but after a large sample is collected, the uncertainty tends to zero. We then enter a regime in which a computational security analysis is needed for estimating the resistance against the secret key recovery. This paper addresses this computational complexity security evaluation, and highlights how the computational security is related to the homophonic coding design.

Summary of the results: The paper contains a security evaluation of the considered randomised encryption schemes under the chosen plaintext attack (CPA). The algebraic representation of their linearised model is shown to be equivalent to a certain LPN (learning from parity with noise) problem, and the hardness of this problem is proven to be increased by using a suitable homophonic encoder. The proposed analysis gives guidelines for the design of this homophonic encoder, that comprises five conditions, two related to the information-theoretical security, two regarding the computational security, and one concerning implementation costs. A homophonic coding strategy which fulfils all the given criteria is exhibited. Also, a comparison of encryption schemes based on the LPN problem with and without homophonic coding shows the benefits obtained with the aid of an appropriate homophonic encoder.

Organisation of the paper: Section 2 formalises a generic model for the class of randomised encryptions addressed in this paper and briefly summarises its known information-theoretic security evaluation. Section 3 contains the security evaluation from a computational complexity view point. Implications of this security evaluation on the design of a dedicated homophonic coding are discussed in Section 4, where the code design criteria are established, while the code constructions are given in Section 5. An illustrative comparison of encryption schemes based on the LPN problem with and without homophonic coding is discussed in Section 6. Concluding remarks are given in Section 7. We would like to mention that a preliminary computational complexity security evaluation of homophonic coding has been reported informally in [14]. (In order to prevent submissions to SKEW 2011 from conflicting with submissions to forthcoming conferences with proceedings, SKEW 2011 will have no formal proceedings, http://skew2011.mat.dtu.dk/submission.html.)

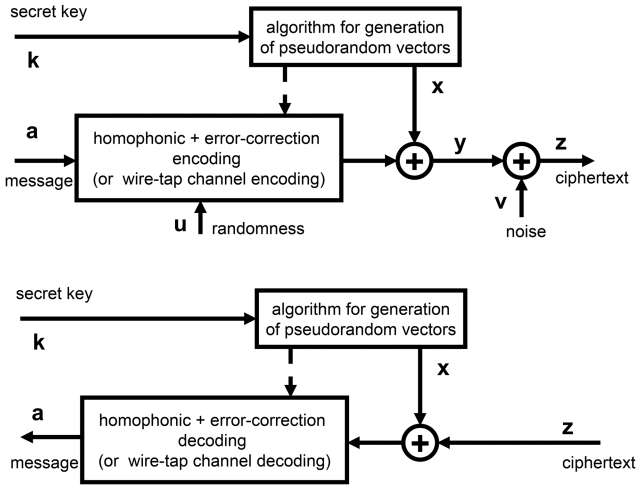

Fig. 1 *Model of a security enhanced randomised encryption within the encoding–encryption paradigm: the upper part shows the transmitter, and the lower part the receiver*

2 Background and model

2.1 Summary on homophonic and wiretap channel coding

Homophonic coding has been introduced in [11] as a source coding technique which transforms a stream of message symbols with an arbitrary frequency distribution into a uniquely decodable stream of symbols which all have the same frequency. A number of homophonic coding schemes have been reported, and as illustrative ones, we point out to [12, 13, 15, 16]. This paper employs the universal homophonic coding approach [13] which is based on an invertible transformation of the source information vector with embedded random bits, and this approach does not require knowledge of the source statistics. The source information vector can be recovered from the homophonic coder output without knowledge of the random bits by passing the codeword to the decoder (inverter) and then discarding the random bits.

The goal of an error correction coding scheme is to provide reliable communications over a noisy channel. Wiretap channel coding has been proposed in [8] for providing at the same time reliable and secret communications between legitimate parties without employing a secret key. Wiretap channel coding is as a coding technique in the setting where the legitimate parties communicate over a channel with known noise or errors-free, and the wiretapper can monitor input into this channel only through another noisy channel with the noise higher than the noise in the channel connecting the legitimate parties. The coding scheme should be designed so that the wiretapper cannot learn anything from the codewords received, because of the confusion implied by the higher noise in his channel. A number of different wiretap coding schemes have been reported for different settings on the channel between legitimate parties and the one towards the wiretapper (see [17] for an illustration of recent developments).

Regarding employment of homophonic and wiretap channel paradigms in this paper, note the following. Homophonic coding is a class of source coding techniques, but in combination with appropriate error correction schemes, it could act as a wiretap channel coding scheme. Universal homophonic encoding is based on an invertible mapping of an information vector with embedded or simply padded random bits into the corresponding codeword. Wiretap channel encoding is based on a random assignment of a codeword from the set of ones corresponding to the considered information vector. In the case of homophonic encoding, all the codewords obtained by embedding/padding of the information vector with random bits could be generated in advance, so that encoding as in wiretap coding schemes could be based on a random selection of one of the codewords assigned to the considered information vector. Particularly, the following should be noted: (i) both homophonic coding and wiretap channel coding are based on assigning multiple codewords to an information vector and delivering as a codeword a randomly selected one from all assigned codewords; (ii) accordingly, in both coding schemes, the codeword contains effects of a number of random bits. Also both homomorphic coding and wiretap coding generate codewords in form of randomised vectors which could be employed for masking other vectors.

2.2 System model

A generic model for randomised encryption schemes which are integrated in communication systems where error correction is performed before encryption is described below, which encompasses the randomised encryption schemes proposed and discussed, e.g. in [3, 4, 6, 18].

This paper employs homophonic or wiretap channel for a purpose different from the ones these coding techniques have been designed for. The main purpose is not just randomisation of the source message vectors (the goal of homophonic coding) nor secrecy without secret key (the goal of wiretap channel coding) but enhancing cryptographic security of certain encryption schemes employing underlying features of the homophonic or wiretap channel coding. The goal is security enhancement of a cryptographic keystream generator for encryption employing a dedicated coding scheme where the codewords provide additional 'masking' of the keystream vectors employed for encryption. The encryption scheme in Fig. 1 performs modulo 2 addition of the outputs of the encoding block and the keystream generator which could not be considered only as 'masking' the message vector with a vector generated by secret key, but also as masking the keystream vector by a randomised mapping of the information vector.

We assume that the encryption from Fig. 1 employs concatenation of the following coding algorithms: (i) universal homophonic coding [13] which performs the following mapping $\{0,1\}^{\ell} \to \{0,1\}^{m}$, $\ell < m$, (ii) linear block error correction code which performs $\{0,1\}^{m} \to \{0,1\}^{n}$, $m < n$, and which provides reliable communication over a binary symmetric channel with the known probability of the bits complementation. Note that any suitable binary linear block code designed to work over a binary symmetric channel with crossover probability p could be employed. There is a lot of these coding schemes reported in the literature and one which best fits into a particular implementation scenario (hardware- or software-oriented) could be selected. Also, Section 5 as well as the Appendix provide a dedicated construction of a coding scheme.

Finally, note that the local generation of the vector $\boldsymbol{x}$ at the transmitter and receiver could be considered as error-free transmission of $\boldsymbol{x}$ through the channel between the legitimate parties, and on the other side, the wiretapper can observe just a noisy version of $\boldsymbol{x}$, degraded by modulo 2 addition with the vector $\boldsymbol{u} \oplus \boldsymbol{v}$. Accordingly, we face a wiretap system where the main channel for transmission of $\boldsymbol{x}$ is error-free and the wiretap channel is a highly noisy one.

Consider a communication system (see Fig. 1) where some message $\boldsymbol{a} = [a_i]_{i=1}^{l} \in \{0,1\}^{l}$ is sent to a transmitter over a noisy channel.

At the transmitter: To ensure reliable communication, a linear error correcting encoder $C_{\mathrm{ECC}}(\cdot)$ is used, that maps an m-bit message to a codeword of $n > m$ bits, using an $m \times n$ binary code generator matrix $\boldsymbol{G}_{\mathrm{ECC}}$. A homophonic (wiretap) encoder $C_{\mathrm{H}}(\cdot)$ is added prior to $C_{\mathrm{ECC}}(\cdot)$, which requires the use of a vector $\boldsymbol{u} = [u_i]_{i=1}^{m-l} \in \{0,1\}^{m-l}$ of pure randomness, i.e. each u_i is the realisation of a random variable U_i with distribution $\Pr(U_i = 1) = \Pr(U_i = 0) = 1/2$. The wiretap encoding $C_{\mathrm{H}}(\boldsymbol{a} \parallel \boldsymbol{u})$ consists of coset encoding [8], which may be described by an $m \times m$ binary matrix $\boldsymbol{G}_{\mathrm{H}}$ such that

$$C_{\mathrm{H}}(\boldsymbol{a} \parallel \boldsymbol{u}) = [\boldsymbol{a} \parallel \boldsymbol{u}]\boldsymbol{G}_{\mathrm{H}}, \quad \boldsymbol{G}_{\mathrm{H}} = \begin{bmatrix} \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_l \\ \boldsymbol{G}^{C} \end{bmatrix} \tag{1}$$

where $\boldsymbol{G}^C$ is an $(m-l)\times m$ generator matrix for an $(m, m-l)$ linear error correction code C, and $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_l$ are l linearly independent row vectors from $\{0,1\}^m \setminus C$.

We get a joint encoding

$$\boldsymbol{a} \in \{0,1\}^l \mapsto C_{\mathrm{ECC}}(C_{\mathrm{H}}(\boldsymbol{a} \parallel \boldsymbol{u})) \in \{0,1\}^n,$$

which may alternatively be written as

$$\begin{aligned} &C_{\mathrm{ECC}}(C_H(\boldsymbol{a} \parallel \boldsymbol{u})) \\ &\quad = C_{\mathrm{ECC}}([\boldsymbol{a} \parallel \boldsymbol{u}]\boldsymbol{G}_{\mathrm{H}}) = [\boldsymbol{a} \parallel \boldsymbol{u}]\boldsymbol{G}_{\mathrm{H}}\boldsymbol{G}_{\mathrm{ECC}} = [\boldsymbol{a} \parallel \boldsymbol{u}]\boldsymbol{G} \end{aligned} \tag{2}$$

where $\boldsymbol{G} = \boldsymbol{G}_{\mathrm{H}}\boldsymbol{G}_{\mathrm{ECC}}$ is an $m \times n$ binary matrix containing the two successive encoders at the transmitter.

Both homophonic coding [13] and generic wiretap coding [8] share the same idea of randomness-based encoding. Indeed, even though the correspondence

$$\boldsymbol{a} \mapsto a_1\boldsymbol{h}_1 \oplus a_2\boldsymbol{h}_2 \oplus \cdots \oplus a_l\boldsymbol{h}_l \oplus C$$

between an l-bit message $\boldsymbol{a} = [a_1, \ldots, a_l]$ and a coset is deterministic, a random codeword:

$$\begin{aligned} \boldsymbol{c} = a_1\boldsymbol{h}_1 \oplus a_2\boldsymbol{h}_2 \oplus \cdots \oplus a_l\boldsymbol{h}_l \oplus u_1\boldsymbol{g}_1^C \oplus u_2\boldsymbol{g}_2^C \oplus \cdots \\ \oplus u_{m-l}\boldsymbol{g}_{m-l}^C \end{aligned}$$

is chosen inside the coset, where $\boldsymbol{u} = [u_1, u_2, \ldots, u_{m-l}]$ is a uniformly distributed random $(m-l)$-bit vector as defined above and $\boldsymbol{g}_1^C, \boldsymbol{g}_2^C, \ldots, \boldsymbol{g}_{m-l}^C$ are the rows of $\boldsymbol{G}^C$.

The codeword sent is finally an encrypted version $\boldsymbol{y}$ of $C_{\mathrm{ECC}}(C_{\mathrm{H}}(\boldsymbol{a} \parallel \boldsymbol{u}))$ given by

$$\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{k}) = C_{\mathrm{ECC}}(C_{\mathrm{H}}(\boldsymbol{a} \parallel \boldsymbol{u})) \oplus \boldsymbol{x} \tag{3}$$

where $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{k}) = [x_i]_{i=1}^n \in \{0,1\}^n$ is a pseudorandom vector needed for encryption, which is generated by either a keystream generator or by a block cipher working in the cipher feedback mode (CFB) as in [4, 5, 18]. Notice the important dependency of $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{k})$ in the secret key $\boldsymbol{k}$. Also note that, for simplicity of the exposition, the data employed for generation of the pseudorandom vectors $\boldsymbol{x}$, which are publicly known (like a public seed and a synchronization parameter), are not explicitly shown. Finally, the model includes the assumption that the concatenation of binary vectors $\boldsymbol{x}$ appears as pseudorandom binary sequences and from a statistical point of view is indistinguishable from a random binary sequence.

At the receiver: The noisy communication channel is modelled by the addition of a noise vector $\boldsymbol{v} = [v_i]_{i=1}^n \in \{0,1\}^n$, where each v_i is the realisation of a random variable V_i with $\Pr(V_i = 1) = p$ and $\Pr(V_i = 0) = 1 - p$. The receiver obtains

$$z = z(\boldsymbol{k}) = \boldsymbol{y} \oplus \boldsymbol{v} = C_{\mathrm{ECC}}(C_{\mathrm{H}}(\boldsymbol{a} \parallel \boldsymbol{u})) \oplus \boldsymbol{x} \oplus \boldsymbol{v} \tag{4}$$

and starts by decrypting

$$\begin{aligned} \boldsymbol{y} = (C_{\mathrm{ECC}}(C_{\mathrm{H}}(\boldsymbol{a} \parallel \boldsymbol{u})) \oplus \boldsymbol{x} \oplus \boldsymbol{v}) \oplus \boldsymbol{x} = C_{\mathrm{ECC}}(C_{\mathrm{H}}(\boldsymbol{a} \parallel \boldsymbol{u})) \\ \oplus \boldsymbol{v}. \end{aligned}$$

He then first decodes $C_{\mathrm{H}}(\boldsymbol{a} \parallel \boldsymbol{u})$. In the case of a successful decoding, he computes $\boldsymbol{a}$ using C_{H}^{-1} and informs the transmitter he could decode. Otherwise, he asks the transmitter for a retransmission. This assumes a noiseless feedback between the receiver and the transmitter. The notations employed are summarised in Table 1.

Note that a variant of the system from Fig. 1 could be considered where all the vectors $\boldsymbol{v}$ have equal pre-specified weight. In such setting, results from [19] could be employed for the security evaluation, but this direction is out of the scope of this paper.

Table 1 Summary of the main notations

$\boldsymbol{a} \in \{0,1\}^l$	confidential message to be transmitted
$\boldsymbol{u} \in \{0,1\}^{m-l}$	vector of pure randomness
C_{ECC}	encoder of the error correction code, which maps a m bit message to an n bit codeword
$\boldsymbol{G}_{\mathrm{ECC}}$	$m \times n$ generator matrix corresponding to the encoder C_{ECC}
C_{H}	wiretap coding (homophonic code) encoder
$\boldsymbol{G}_{\mathrm{H}}$	$m \times m$ generator matrix corresponding to C_{H}
$\boldsymbol{G} = \boldsymbol{G}_{\mathrm{H}}\boldsymbol{G}_{\mathrm{ECC}}$	combined generator matrix
$\boldsymbol{k}$	secret key, of length n
$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{k}) \in \{0,1\}^n$	pseudorandom vector used for encryption
$\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{k})$	encrypted version of the codeword to be sent
$\boldsymbol{v} \in \{0,1\}^n$	random (Bernoulli distributed) noise at the receiver
$z \in \{0,1\}^n$	received noisy codeword
$\boldsymbol{x}^{(t)} = \boldsymbol{k}\boldsymbol{S}^{(t)}$	linearised version of $\boldsymbol{x}$ as a function of time

2.3 Information-theoretic security evaluation

In [6], the above model of randomised encryption schemes was studied from an information-theoretic point of view. The goal was to analyse the security enhancement provided by the wiretap encoding, in terms of the secret key $\boldsymbol{k}$ equivocation, that is, the uncertainty that an adversary must face about the secret key, given all the information he could collect during passive or active attacks.

This analysis demonstrated a gain of unconditional security, and thus confirmed the security benefit of the additional wiretap encoder, through tight lower bounds (Lemmas 1 and 2 in [6]) and asymptotic values (Theorems 1 and 2 in [6]) of the secret key equivocation. The cost of this enhanced security is only a slight/moderate increase in the implementation complexity and the communications overhead.

However, it also revealed that if the same secret key is used for too long, the adversary may gather large enough samples for cryptanalysis. The uncertainty then tends to zero. Then starts a regime in which a computational security analysis is needed to estimate the resistance against the secret key recovery, which motivated the current paper.

Recall that in a CPA scenario, the claim that a scheme is secure in an information-theoretic sense means that even an attacker with unlimited resources for recovering the secret key, in the considered evaluation scenario, faces an uncertainty about the secret key employed for encryption, i.e. after all, a set of equally probable candidates for the true secret key will exist. On the other hand, claim that an encryption scheme is secure in a computational complexity sense means the following: although the secret key could be recovered in a CPA scenario, and so it is not possible to claim information-theoretic security, the computational complexity of this recovery is as hard as solving a problem which belongs to a class of proven hard problems, as the LPN problem is.

3 Computational complexity security evaluation

In this section, we analyse the security of the encryption scheme (3) from a computational complexity view point. We perform the security evaluation over a simplified linearised version of the scheme assuming that it determines a lower bound on the computational security of the original scheme. We show that the problem of secret key recovery under a CPA in the linearised scheme is equivalent to a certain LPN problem, implying that the lower bound on the encryption security is determined by the hardness of the related LPN problem.

Accordingly, we consider the following CPA security evaluation scenario:

- a simplified version of the scheme from Fig. 1 is considered where instead of the keystream generator, a simple linear feedback shift register is employed for generation of keystream

bits assuming that this setting provides lower bound on the cryptographic security;
- arbitrary long sequences of message bits (concatenation of the vectors $\boldsymbol{a}$) selected by the attacker are known;
- arbitrary long sequences of ciphertext bits are available for cryptanalysis (concatenation of the vectors $\boldsymbol{z}$);
- the goal of the cryptanalysis is the recovery of the secret key employed for the generation of the keystream (i.e. a sequence of consecutive vectors $\boldsymbol{z}$).

3.1 Preliminaries

We follow an instantiation of the following security evaluation approach (see [20], for example): a given construction is secure as long as some underlying problem is hard, given a reduction which shows how any efficient adversary that succeeds in 'breaking' the construction with non-negligible probability also can play the role of an efficient algorithm that succeeds in solving the problem that was assumed to be hard. Our security evaluation follows the above approach, and our initial assumption is that the considered encryption is at least as secure as its linearised simplification. Consequently, our security evaluation objectives are focused towards showing that: (i) an attack on a linearised algebraic model of the considered encryption is equivalent to solving a certain LPN problem, and (ii) homophonic coding increases the hardness of the source LPN problem.

We show that an algebraic representation of a linearised version of the considered model of encryption is equivalent to an algebraic representation of a certain LPN problem. Thus, if an algorithm for breaking the encryption exists, it can be employed for solving the corresponding LPN problem as well. Also, we show that the considered encryption provides an increase in a parameter (the effective corrupting noise) of the underlying LPN problem which determines its hardness. The impact of this parameter (the noise level) on the LPN problem complexity has been considered in a several papers, e.g. [21–26]. Finally, consideration of the linearised model does not assume any restriction on the power of an attacker, i.e. we assume that an attacker with the same power could attack the actual (non-linearised) model as well as the linearised one.

3.2 Linearised model

The analysis starts with the assumption that the security level of the considered encryption is at least that of the scheme where the keystream generator is replaced with a linear finite state machine with the same key. So, we assume that hardness of breaking the actual system is at least as hard as breaking the linearised one.

Consequently, we will show in our complexity analysis that the hardness of breaking the linearised scheme relies on the LPN problem hardness (see [21–25], for example), and that the identified LPN problem is also equivalent to the problem of secret key recovery under CPA. The analysis will pinpoint which features the homophonic encoding must have to incur an increased complexity of the underlying LPN problem in the average case.

We perform the security evaluation under the assumptions that the length of $\boldsymbol{k}$ is n, and that:

i. $\boldsymbol{x}^{(t)} = f^{(t)}(\boldsymbol{k}) = \boldsymbol{k}\boldsymbol{S}^{(t)}$, $t = 1, 2, \ldots, \tau$, where $f^{(t)}(\cdot)$ is a linear function which maps the secret $\boldsymbol{k}$ into the keystream segment at the time instance t, employing a randomly selected and publicly known balanced $n \times n$ binary matrix $\boldsymbol{S}^{(t)} = [s_{i,j}^{(t)}]_{i=1}^{n}{}_{j=1}^{n}$ and

$$\boldsymbol{S}^{(t)} = [\boldsymbol{S}_1^{(t)}, \boldsymbol{S}_2^{(t)}, \ldots, \boldsymbol{S}_n^{(t)}] \tag{5}$$

where each $\boldsymbol{S}_i^{(t)}$, $i = 1, 2, \ldots, n$, denotes a column of the matrix $\boldsymbol{S}^{(t)}$. The function $f^{(t)}(\cdot)$ is usually heavily non-linear, and considering it as linear actually implies a lower bound on the security. Similarly, instead of $\boldsymbol{x}^{(t)} = \boldsymbol{k}\boldsymbol{S}^{(t)}$, we consider the setting $\boldsymbol{x}^{(t)} = f^{(t)}(\boldsymbol{k}) = \boldsymbol{k}\boldsymbol{S}^{t}$ which maps the session secret $\boldsymbol{k}$ (which depends on the secret seed and the public initial value) into the keystream segment at the time instance t, $\boldsymbol{S} = [s_{i,j}]_{i=1}^{n}{}_{j=1}^{n}$ is a known binary matrix, and $\boldsymbol{S}^{t} = [\boldsymbol{S}_1^{(t)}, \boldsymbol{S}_2^{(t)}, \ldots, \boldsymbol{S}_n^{(t)}]$ where each $\boldsymbol{S}_i^{(t)}$, $i = 1, 2, \ldots, n$, denotes a column of the tth power of the matrix $\boldsymbol{S}$, $t = 1, 2, \ldots, \tau$.

ii. We consider a CPA where the data is the zero vector $\boldsymbol{a}^{(t)} = \boldsymbol{0}$, for each t.

Under the above assumptions, and recalling from (2) that C_{ECC} and C_{H} are both linear encoders, with $\boldsymbol{G} = \boldsymbol{G}_{\mathrm{H}}\boldsymbol{G}_{\mathrm{ECC}}$, we write

$$\boldsymbol{z}^{(t)} \oplus \boldsymbol{v}^{(t)} = \boldsymbol{k}\boldsymbol{S}^{(t)} \oplus [\boldsymbol{0} \parallel \boldsymbol{u}^{(t)}]\boldsymbol{G},$$

from which we get an algebraic representation of the security evaluation problem in terms of a noisy system of linear equations, as seen by the adversary (for $t = 1, 2, \ldots, \tau$):

$$\begin{bmatrix} \boldsymbol{k}\boldsymbol{S}_1^{(t)} \\ \boldsymbol{k}\boldsymbol{S}_2^{(t)} \\ \vdots \\ \boldsymbol{k}\boldsymbol{S}_n^{(t)} \end{bmatrix} \oplus \begin{bmatrix} [\boldsymbol{0} \parallel \boldsymbol{u}^{(t)}]\boldsymbol{G}_1 \\ [\boldsymbol{0} \parallel \boldsymbol{u}^{(t)}]\boldsymbol{G}_2 \\ \vdots \\ [\boldsymbol{0} \parallel \boldsymbol{u}^{(t)}]\boldsymbol{G}_n \end{bmatrix} = \begin{bmatrix} z_1^{(t)} \\ z_2^{(t)} \\ \vdots \\ z_n^{(t)} \end{bmatrix} \oplus \begin{bmatrix} v_1^{(t)} \\ v_2^{(t)} \\ \vdots \\ v_n^{(t)} \end{bmatrix}, \tag{6}$$

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

From (6), we have the following system of τn overdefined consistent but probabilistic equations over $\{0, 1\}$:

$$\begin{aligned} \boldsymbol{k}\boldsymbol{S}_1^{(t)} &\oplus [\boldsymbol{0} \parallel \boldsymbol{u}^{(t)}]\boldsymbol{G}_1 = z_1^{(t)} \oplus v_1^{(t)} \\ \boldsymbol{k}\boldsymbol{S}_2^{(t)} &\oplus [\boldsymbol{0} \parallel \boldsymbol{u}^{(t)}]\boldsymbol{G}_2 = z_2^{(t)} \oplus v_2^{(t)}, \quad t = 1, 2, \ldots, \tau, \\ &\vdots \\ \boldsymbol{k}\boldsymbol{S}_n^{(t)} &\oplus [\boldsymbol{0} \parallel \boldsymbol{u}^{(t)}]\boldsymbol{G}_n = z_n^{(t)} \oplus v_n^{(t)} \end{aligned} \tag{7}$$

where each equation is correct with probability equal to $1 - p$, $\boldsymbol{0}$ is an ℓ-dimensional zero vector, and $\boldsymbol{u}^{(t)} = [u_i^{(t)}]_{i=1}^{m-\ell}$.

The above system of equations fits the hypotheses of Lemma 2, since we have $N = \tau n$ equations, for $L = n$ unknown we want to find, where $\oplus_{j=1}^{L} \alpha_j^{(k)} x_j$, $k = 1, \ldots, N$, correspond to $\boldsymbol{k}\boldsymbol{S}_i^{(t)}$, $i = 1, \ldots, n$, $t = 1, \ldots, \tau$, and $\oplus_{j=1}^{M} \beta_j^{(k)} y_j$ for $k = 1, \ldots, N$ correspond to $[\boldsymbol{0} \parallel \boldsymbol{u}^{(t)}]\boldsymbol{G}_i$, $i = 1, \ldots, M$, $t = 1, \ldots, \tau$, implying that we can separate the $\{[\boldsymbol{0} \parallel \boldsymbol{u}^{(t)}]\boldsymbol{G}_i\}_{i=1}^{n}$ for every t into one set of linearly independent vectors, and another set which contains linear combinations of the first set.

Statement 1: The above system of τn equations contains only $n + \tau(m - \ell)$ unknowns, and the aim is to recover $\boldsymbol{k}$ only, i.e. we do not have any interest in recovering $\{u_i^{(t)}\}_{i=1}^{m-\ell}$, $t = 1, 2, \ldots, \tau$. Using Gaussian elimination, we can eliminate the $\tau(m - l)$ unknown $\{u_i^{(t)}\}_{i=1}^{m-\ell}$, $t = 1, 2, \ldots, \tau$, and obtain $\tau(n - m + \ell)$ equations where only $\boldsymbol{k}$ is unknown. The initial system of τn equations is transformed into the following one with $\tau(n - m - \ell)$ equations (in total) and n unknowns $\boldsymbol{k} = [k_i]_{i=1}^{n}$:

$$\begin{aligned} \mathscr{L}_1^{(k)}(\boldsymbol{k}) &= \mathscr{L}_1^{(z)}\left([z_i^{(t)}]_{i=1}^{n}\right) \oplus \mathscr{L}_1^{(v)}\left([v_i^{(t)}]_{i=1}^{n}\right) \\ \mathscr{L}_2^{(k)}(\boldsymbol{k}) &= \mathscr{L}_2^{(z)}\left([z_i^{(t)}]_{i=1}^{n}\right) \oplus \mathscr{L}_2^{(v)}\left([v_i^{(t)}]_{i=1}^{n}\right) \\ &\vdots \\ \mathscr{L}_{n-m+\ell}^{(k)}(\boldsymbol{k}) &= \mathscr{L}_{n-m+\ell}^{(z)}\left([z_i^{(t)}]_{i=1}^{n}\right) \oplus \mathscr{L}_{n-m+\ell}^{(v)}\left([v_i^{(t)}]_{i=1}^{n}\right) \end{aligned} \tag{8}$$

$t = 1, 2, \ldots, \tau$, where $\mathscr{L}_j^{(k)}(\cdot)$, $\mathscr{L}_j^{(z)}(\cdot)$, and $\mathscr{L}_j^{(v)}(\cdot)$, $j = 1, 2, \ldots, n - m + \ell$, are linear functions, described by the matrix $\boldsymbol{G}$ and the Gaussian elimination used to remove the random bits $\boldsymbol{u}^{(t)}$, while $\mathscr{L}_j^{(k)}(\cdot)$ further depends on the known matrix $\boldsymbol{S}^{(t)}$. Recall that the Gaussian elimination of the variables $\{u_i^{(t)}\}_{i=1}^{m-\ell}$ can be performed independently for each t, thus, the complexity (for

$t = 1, 2, \ldots, \tau$) is upperbounded by $\tau O(n^{2.7})$, assuming that the most efficient algorithm for Gaussian elimination is used.

Finally, we introduce the following assumptions.

Assumption 1: Consider the following N equations over the binary field $GF(2)$ to be solved for $x_1, \ldots, x_L$, $L = N - M$:

$$\left(\bigoplus_{j=1}^{L} \alpha_j^{(i)} x_j \right) \oplus \left(\bigoplus_{j=1}^{M} \beta_j^{(i)} y_j \right) = z_i \oplus e_i, \quad i = 1, 2, \ldots, N, \tag{9}$$

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 realisation of a random variable E_i, such that $\Pr(E_i = 1) = p < 1/2$, $i = 1, 2, \ldots, N$. We assume that:

i. the Hamming weight of each vector $[\beta_1^{(i)}, \ldots, \beta_M^{(i)}]$ is greater or equal to some parameter w, for $i = 1, 2, \ldots, N$,
ii. any $\Omega^{(i)} \subset \{1, 2, \ldots, N\} \setminus i$, such that $\oplus_{k \in \Omega^{(i)}} \left(\oplus_{j=1}^{M} \beta_j^{(k)} y_j \right) = \oplus_{j=1}^{M} \beta_j^{(i)} y_j$, $i = 1, 2, \ldots, N$, has cardinality at least equal to w, i.e. there are at least w linearly independent sums $\oplus_{j=1}^{M} \beta_j^{(i)} y_j$ among those $i \in \{1, 2, \ldots, N\}$.

Assumption 2: The homophonic encoder matrix $\boldsymbol{G}_{\mathrm{H}} = [g_{ij}^{(H)}]_{i=1\ j=1}^{m\ m}$ satisfies:

$$\sum_{i=1}^{m-\ell} g_{\ell+i, m-\ell+j}^{(H)} \geq w, \quad j = 1, m - \ell + 2, \ldots, l,$$

for some parameter w, and the submatrix of the encoder matrix $\boldsymbol{G} = \boldsymbol{G}_{\mathrm{H}} \boldsymbol{G}_{\mathrm{ECC}}$ consisting of its $m - \ell$ last rows is such that any of the columns is a linear combination of at least w other columns.

3.3 LPN problem hardness and security

3.3.1 Search and decisional LPN problems: Solving a system of noisy linear equations is related to the LPN problem, defined formally as follows (see e.g. [27]).

Definition 1: (LPN search problem): Let $\boldsymbol{s}$ be a random binary string of length n. Consider the Bernoulli distribution $\mathscr{B}_\theta$ with parameter $\theta \in (0, 1/2)$ (if $e \leftarrow \mathscr{B}_\theta$ then $\Pr(e = 0) = \theta$ and $\Pr(e = 1) = 1 - \theta$). Let $\mathscr{Q}_{\boldsymbol{s},\theta}$ be the following distribution:

$$\{(\boldsymbol{a}, \boldsymbol{a}\boldsymbol{s}^{\mathrm{T}} \oplus e) | \boldsymbol{a} \leftarrow \{0,1\}^n, e \leftarrow \mathscr{B}_\theta\}.$$

Given the security parameter n, for an adversary $\mathscr{A}$ trying to discover the random string $\boldsymbol{s}$, we define its advantage as

$$\mathrm{Adv}_{\mathrm{LPN}_\theta}(n) = \Pr[\mathscr{A}^{\mathscr{Q}_{\boldsymbol{s},\theta}} = \boldsymbol{s} | \boldsymbol{s} \leftarrow \{0,1\}^n].$$

The LPN_θ problem with parameter θ is hard if the advantage of all PPT (probabilistic polynomial time) adversaries $\mathscr{A}$ is negligible.

Given a security parameter n, a secret vector $\boldsymbol{s}$, and $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N$ randomly chosen binary vectors, the LPN search problem captures the possibility, knowing $y_i = \langle \boldsymbol{s} | \boldsymbol{a}_i \rangle$ and $\{\boldsymbol{a}_i\}_{i=1}^{N}$, to solve for $\boldsymbol{s}$ using standard linear algebra techniques, assuming there is no noise. When each y_i is, however, flipped (independently) with probability p, the problem of computing $\boldsymbol{s}$ becomes hard, and is referred to as the LPN problem.

The LPN search problem is equivalent to the problem of linear block code decoding which is known to be NP-complete [28].

In [27], a distinguishing variant of the problem has been introduced. Roughly speaking, the decisional LPN problem asks to distinguish a number of noisy samples $\boldsymbol{y}$ of a linear function (specified by a secret vector $\boldsymbol{x}$ which stand for the secret $\boldsymbol{s}$) from a uniform distribution. The problem is, given $\boldsymbol{A}$ and $\boldsymbol{y}$, to decide whether $\boldsymbol{y}$ is distributed according to $\boldsymbol{A} \cdot \boldsymbol{x} \oplus \boldsymbol{e}$ or chosen uniformly at random.

The LPN decisional problem has been used to analyse the security of encryption techniques for stream ciphers in e.g. [3]. The following formal definition can be found in [27].

Definition 2: (LPN decisional (distinguishing) problem): Let $\boldsymbol{s}$, $\boldsymbol{a}$ be binary strings of length n. Let further $\mathscr{Q}_{\boldsymbol{s},\theta}$ be as in Definition 1. Let $\mathscr{A}$ be a PPT adversary. The distinguishing advantage of $\mathscr{A}$ between $\mathscr{Q}_{\boldsymbol{s},\theta}$ and the uniform distribution $\mathscr{U}_{n+1}$ is defined as

$$\mathrm{Adv}_{\mathrm{LPND}_\theta}(n) = \left| \Pr[\mathscr{A}^{\mathscr{Q}_{\boldsymbol{s},\theta}} = 1 | \boldsymbol{s} \leftarrow \{0,1\}^n] - \Pr[\mathscr{A}^{\mathscr{U}_{n+1}} = 1] \right|.$$

The LPND_θ problem with parameter θ is hard if the advantage of all PPT adversaries $\mathscr{A}$ is negligible.

It has been shown in [27] that the distinguishing problem is as hard as the search problem with similar parameters.

It should be noted that the average case hardness of the above two problems cannot be reduced to the worst-case hardness of an NP-hard problem. The confidence on the average case hardness of solving these problems comes from the lack of efficient solutions, despite the efforts over the years.

For the computational complexity evaluation of this paper, we next define the matrix LPN (MLPN) distinguishing problem.

3.3.2 MLPN distinguishing problem: The MLPN distinguishing problem is defined analogously to the LPND problem, with the difference that the secret key is now a matrix, not a vector.

Definition 3: (MLPN distinguishing problem): Let $\boldsymbol{S} \in \{0,1\}^{m \times n}$ be a secret key, and $\theta \in (0, 1/2)$. Let $\mathscr{Q}_{\boldsymbol{S},\theta}$ be the distribution with samples

$$\{(\boldsymbol{a}, \boldsymbol{a}\boldsymbol{S} \oplus \boldsymbol{e}) | \boldsymbol{a} \leftarrow \{0,1\}^m, \boldsymbol{e} \leftarrow \mathscr{B}_\theta^n\}.$$

Let $\mathscr{A}$ be a PPT adversary. The matrix variant of the LPND_θ problem is to distinguish the access to the oracle $\mathscr{Q}_{\boldsymbol{S},\theta}$ from the access to the oracle $\mathscr{U}_{m+n}$ (we use the same notation $\mathscr{U}^{m+n}$ and $\mathscr{Q}_{\boldsymbol{S},\theta}$ to denote, respectively, both the distribution and the corresponding oracle), which takes samples from the uniform distribution over $\{0,1\}^{m+n}$. The distinguishing advantage of $\mathscr{A}$ is defined as:

$$\mathrm{Adv}_{\mathrm{MLPND}_\theta}(m, n) = \left| \Pr[\mathscr{A}^{\mathscr{Q}_{\boldsymbol{S},\theta}} = 1 | \boldsymbol{S} \leftarrow \{0,1\}^{m \times n}] - \Pr[\mathscr{A}^{\mathscr{U}_{m+n}} = 1] \right|.$$

The MLPND_θ problem is ϵ-hard if for all PPT adversaries $\mathscr{A}$, the advantage $\mathrm{Adv}_{\mathrm{MLPND}_\theta}(m, n)$ is less than ϵ. In asymptotic terms, MLPND_θ is hard if the advantage of all PPT adversaries $\mathscr{A}$ is negligible. The following lemma shows that hardness of the LPND problem implies hardness of the MPLND problem. It is a slight adaptation of [29, Proposition 2].

Lemma 1: : If LPND_θ is ϵ-hard, then MLPN_θ is ϵn-hard.

Proof: The proof technique is standard [29, Proposition 2]: we shall assume that there exists an adversary $\mathscr{A}^{\mathscr{Q}_{X,\theta}}$ with advantage at least $n\epsilon$ for MLPN_θ and use it as a subroutine to construct an adversary $\mathscr{A}^{\mathscr{Q}_{x,\theta}}$ whose advantage is at least ϵ, which contradicts the hardness assumption of LPND_θ which implies that the advantage should be less than ϵ.

Let $\boldsymbol{X}$ be a binary $m \times n$ matrix with columns $\boldsymbol{x}_1^{\mathrm{T}}, \ldots, \boldsymbol{x}_n^{\mathrm{T}}$. We define the probability distribution $\mathscr{Q}_{\boldsymbol{X},\theta}^{i}$ over $\{0,1\}^{m \times n}$ whose samples are pairs

$$\{(\boldsymbol{a}, z), \quad z = (\boldsymbol{a}\boldsymbol{x}_1^{\mathrm{T}} \oplus e_1, \ldots, \boldsymbol{a}\boldsymbol{x}_i^{\mathrm{T}} \oplus e_i, b_{i+1}, \ldots, b_n),$$
$$\boldsymbol{a} \leftarrow \{0,1\}^m, e_1, \ldots, e_i \leftarrow \mathscr{B}_\theta, b_{i+1}, \ldots, b_n \leftarrow \{0,1\}\}$$

for $i = 0, \ldots, n$. Let

$$p_i = \Pr[\mathscr{A}^{\mathscr{Q}^i_{\boldsymbol{X},\theta}} = 1 | \boldsymbol{X} \leftarrow \{0,1\}^{m \times n}]$$

be the probability that an adversary $\mathscr{A}^{\mathscr{Q}_{\boldsymbol{X},\theta}}$ outputs 1, when its input is a sample $\mathscr{Q}^i_{\boldsymbol{X},\theta}$, with particular cases

$$p_0 = \Pr[\mathscr{A}^{\mathscr{U}_{m+n}} = 1], \quad p_n = \Pr[\mathscr{A}^{\mathscr{Q}_{\boldsymbol{X},\theta}} = 1 | \boldsymbol{X} \leftarrow \{0,1\}^{m \times n}].$$

We next construct an adversary $\mathscr{A}^{\mathscr{Q}_{\boldsymbol{x},\theta}}$ for LPND_θ:

i. A sample $(\boldsymbol{a}, z) \in \{0,1\}^{m+1}$ is given from an oracle, either $\mathscr{U}^{m+1}$ (then $z \leftarrow \{0,1\}$) or $\mathscr{A}^{\mathscr{Q}_{\boldsymbol{x},\theta}}$ (in which case $z = \boldsymbol{a}\boldsymbol{x}^{\mathrm{T}} \oplus e$, $e \leftarrow \mathscr{B}_\theta$).
ii. Now the adversary for LPND_θ chooses

$$i \leftarrow \{0, \ldots, n\}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1} \leftarrow \{0,1\}^m,$$
$$e_1, \ldots, e_{i-1} \leftarrow \mathscr{B}_\theta, b_{i+1}, \ldots, b_n \leftarrow \{0,1\}$$

and forwards the sample

$$(\boldsymbol{a}, \tilde{z}), \quad \tilde{z} = (\boldsymbol{a}\boldsymbol{x}_1^{\mathrm{T}} \oplus e_1, \ldots, \boldsymbol{a}\boldsymbol{x}_{i-1}^{\mathrm{T}} \oplus e_{i-1}, z, b_{i+1}, \ldots, b_n)$$

as input to MLPND_θ.
iii. The adversary outputs the same output value IND (0 or 1) returned by MLPND_θ.

Note that if $(\boldsymbol{a}, z)$ is the output of $\mathscr{A}^{\mathscr{Q}_{\boldsymbol{x},\theta}}$, then the sample $(\boldsymbol{a}, \tilde{z})$ is a sample from $\mathscr{Q}^i_{\boldsymbol{X},\theta}$. If instead $(\boldsymbol{a}, z)$ is the output of $\mathscr{U}^{m+1}$, then $(\boldsymbol{a}, \tilde{z})$ is a sample of $\mathscr{Q}^{i-1}_{\boldsymbol{X},\theta}$.

The advantage of $\mathscr{A}^{\mathscr{Q}_{\boldsymbol{x},\theta}}$ is computed as follows:

$$\begin{aligned}
&\mathrm{Adv}_{\mathrm{LPND_q}}(m) \\
&= \left|\Pr[\mathscr{A}^{\mathscr{Q}_{\boldsymbol{x},\theta}} = 1 | \boldsymbol{x} \leftarrow \{0,1\}^m] - \Pr[\mathscr{A}^{\mathscr{U}^{m+1}} = 1]\right| \\
&= |\sum_{j=1}^{n} \Pr[\mathscr{A}^{\mathscr{Q}_{\boldsymbol{x},\theta}} = 1 | \boldsymbol{x} \leftarrow \{0,1\}^m, i = j] \Pr[i = j] \\
&\quad - \Pr[\mathscr{A}^{\mathscr{U}^{m+1}} = 1 | i = j] \Pr[i = j]| \\
&= \frac{1}{n} |\sum_{j=1}^{n} \Pr[\mathscr{A}^{\mathscr{Q}^j_{\boldsymbol{X},\theta}} = 1 | \boldsymbol{X} \leftarrow \{0,1\}^{m \times n}] \\
&\quad - \Pr[\mathscr{A}^{\mathscr{Q}^{j-1}_{\boldsymbol{X},\theta}} = 1 | \boldsymbol{X} \leftarrow \{0,1\}^{m \times n}]| \\
&= \frac{1}{n}\left|\sum_{j=1}^{n} (p_j - p_{j-1})\right| = \frac{1}{n}|p_n - p_0| \geq \epsilon\,.
\end{aligned}$$

Thus, the adversary $\mathscr{A}^{\mathscr{Q}_{\boldsymbol{x},\theta}}$ achieves an advantage greater or equal to ϵ, which contradicts that LPND_θ is ϵ-hard. □

3.4 Security of encryption and hardness of the LPN problem

With the background on the LPN problem established, we are now ready to go back to solve (9) for $x_1, \ldots, x_L$, given by:

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

The following lemma shows how Gaussian elimination applied on this randomised system of equations eliminates the unknowns we are not interested in, while increasing the randomisation noise. We thus get a lower bound on this randomisation noise.

Lemma 2: Under Assumption 1, the problem of recovering the unknown $x_1, x_2, \ldots, x_L$ is reduced to the problem of solving the following system of equations, $i = 1, 2, \ldots, N - M$:

$$\left(\bigoplus_{k \in \Omega^{(i)}} \left(\bigoplus_{j=1}^{L} \alpha_j^{(k)} x_j\right)\right) \oplus \left(\bigoplus_{j=1}^{L} \alpha_j^{(i)} x_j\right) = z_i \oplus \left(\bigoplus_{k \in \Omega^{(i)}} z_k\right) \oplus e_i^*, \quad (10)$$

where e_j^*, the corruption noise of the system of equations with reduced number of unknown variables, is a realisation of a random variable E_j^* such that $\Pr(E_j^* = 1) > p_w = \left((1 - (1-2p)^{w+1})/2\right)$.

Proof: The unknowns y_i, $i = 1, 2, \ldots, M$, can be eliminated from the considered system of N equations by Gaussian elimination, and we obtain a transformed system of $N - M$ equations where only x_i, $i = 1, 2, \ldots, L$, are the unknowns. In each of the new equations, the term $\bigoplus_{j=1}^{M} \beta_j^{(i)} y_j$ from the initial equation, is cancelled by the corresponding linear combination $\bigoplus_{k \in \Omega^{(i)}} \bigoplus_{j=1}^{M} \beta_j^{(k)} y_j = \bigoplus_{j=1}^{M} \beta_j^{(i)} y_j$, where $\Omega^{(i)} \subset \{1, 2, \ldots, N\} \setminus i$ and, according to the lemma assumption, its cardinality is at least w, $i = 1, 2, \ldots, N - M$. Consequently, the new system of equations has the form, $i = 1, 2, \ldots, N - M$:

$$\begin{aligned}
&\left(\bigoplus_{k \in \Omega^{(i)}} \left(\bigoplus_{j=1}^{L} \alpha_j^{(k)} x_j\right)\right) \oplus \left(\bigoplus_{j=1}^{L} \alpha_j^{(i)} x_j\right) \\
&\quad = z_i \oplus \left(\bigoplus_{k \in \Omega^{(i)}} z_k\right) \oplus e_i \oplus \left(\bigoplus_{k \in \Omega^{(i)}} e_k\right).
\end{aligned} \quad (11)$$

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

$$E_i^* = E_i \oplus \left(\bigoplus_{k \in \Omega^{(i)}} E_k\right), \quad i = 1, \ldots, N - M\,.$$

Note that E_i and E_k, $k \in \Omega^{(i)}$, are mutually independent, and

$$\begin{aligned}
&\Pr(E_i^* = 1) = 1 - \Pr(E_i^* = 0) \\
&= 1 - \Pr(E_i \oplus (\bigoplus_{k \in \Omega^{(i)}} E_k) = 0)\,.
\end{aligned}$$

The probability that an even number of digits are 1 in a sequence of $w + 1$ independent binary digits is [30, Lemma 1]

$$\frac{1 + (1-2p)^{w+1}}{2}$$

if p is the probability that a digit is 1. Since

$$\frac{1 + (1-2p)^{w}}{2} > \frac{1 + (1-2p)^{w+1}}{2}, \quad p < 1/2,$$

we have that

$$1 - \frac{1 + (1-2p)^{w}}{2} < 1 - \frac{1 + (1-2p)^{w+1}}{2}\,.$$

Accordingly

$$\begin{aligned}
p_w = \Pr(E_i^* = 1) &= 1 - \Pr\left(E_i \oplus \left(\bigoplus_{k \in \Omega^{(i)}} E_k\right) = 0\right) \\
&> 1 - \frac{1 + (1-2p)^{w+1}}{2} = \frac{1 - (1-2p)^{w+1}}{2}
\end{aligned}$$

since by Assumption 1, 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 main evaluation result:

Theorem 1: Consider the encryption (4), where the homophonic encoder matrix $\boldsymbol{G}_{\mathrm{H}} = [g_{ij}^{(H)}]_{i=1}^{m}{}_{j=1}^{m}$ satisfies Assumption 2. Then when $\boldsymbol{x}^{(t)} = \boldsymbol{k}\boldsymbol{S}^{(t)}$, $t = 1, 2, \ldots, \tau$, the encryption (4) is CPA secure assuming that the underlying LPN problem is hard. Furthermore, an adversary is facing the complexity of solving the $\mathrm{LPN}_{\varepsilon}$ problem where $\epsilon = \left((1-(1-2p)^{w+1})/2\right)$, assuming that this complexity is upperbounded by $\min_{t, t=1,\ldots,\tau}\left\{\binom{n}{\mathrm{Hw}(\boldsymbol{e}_t)}\right\} n^{2.7}$, where $\mathrm{Hw}(\boldsymbol{e}_t)$ denotes the Hamming weight of the noise vector $\boldsymbol{e}_t$.

Proof: The proof is done in two steps. First, we prove that the considered scheme is CPA secure, assuming that the underlying LPN problem is hard, and then we prove that the employed homophonic coding enhances hardness of the underlying LPN problem.

According to the assumptions, the ciphertext (4) can be written as

$$\begin{aligned} z_t &= C_{\mathrm{ECC}}(C_{\mathrm{H}}(\boldsymbol{a}_t \parallel \boldsymbol{u}_t)) \oplus \boldsymbol{x}_t \oplus \boldsymbol{v}_t \\ &= C^*(\boldsymbol{a}_t \parallel \boldsymbol{u}_t) \oplus \boldsymbol{k}\boldsymbol{S}^{(t)} \oplus \boldsymbol{v}_t \end{aligned} \tag{12}$$

$t = 1, 2, \ldots, \tau$, where $C^*(\cdot)$ is an encoding operator. On the other hand, the encryption proposed in [2] has the following algebraic representation:

$$z_t = C(\boldsymbol{a}_t) \oplus \boldsymbol{r}_t\boldsymbol{K} \oplus \boldsymbol{v}_t \tag{13}$$

$t = 1, 2, \ldots, \tau$, where $C(\cdot)$ is the encoding operator, $\boldsymbol{r}_t$ a randomly selected vector, and $\boldsymbol{K}$ the secret key matrix.

The representations (12) and (13) directly imply that the considered encryptions fit into the same encryption paradigm with the following main difference: the encryption (12) is based on the LPN problem and the encryption (13) is based on the MLPN problem. Note that the different encodings $C^*(\cdot)$ and $C(\cdot)$ have no security impact since they are public.

It has been proved in [2] that the scheme is CPA secure, assuming that the underling MLPN problem is hard. On the other hand, assuming that the encryption (12) is not CPA secure when the underlying LPN problem is hard, as a consequence of Lemma 1, would imply that it can be used for compromising the security of the encryption (13), for which we have proof that it is CPA secure, and accordingly, we face a contradiction which implies that the encryption (12) is CPA secure when the underlying LPN problem is hard.

In continuation, it is proved that homophonic encryption enhances hardness of the underlying LPN problem. For simplicity of further consideration, we assume $\boldsymbol{a}_t = \boldsymbol{0}$, $t = 1, 2, \ldots, \tau$. Accordingly, the algebraic model of encryption corresponds to the system of equations given by Statement 1. The assumption that the complexity of the considered LPN problem is upperbounded by $\min_t \left\{\binom{n}{\mathrm{Hw}(\boldsymbol{e}_t)}\right\} n^{2.7}$ implies that a generic approach for solving a stochastic problem by reducing it to a deterministic one based on guessing any of the noise vectors $\boldsymbol{e}_t$, $t = 1, 2, \ldots, \tau$, does not provide any gain.

Lemma 2 and its underlying assumptions ensure that each equation in (8) is correct with some probability lower than $1 - p_w$, where

$$p_w = \frac{1-(1-2p)^{w+1}}{2},$$

since the noise $(\boldsymbol{v}_1^*)^{(t)} = \mathscr{L}_1^{(v)}([v_i^{(t)}]_{i=1}^{n}), \ldots, (\boldsymbol{v}_{n-m+\ell}^*)^{(t)} = \mathscr{L}_{n-m+\ell}^{(v)}([v_i^{(t)}]_{i=1}^{n})$ has coefficients that are the realisation of a random variable which takes value 1 with probability greater than $p_w = \left((1-(1-2p)^{w+1})/2\right)$.

Using the definition of the LPN problem and the above representation, the considered encryption is as secure as a particular LPN_{ϵ} problem with $\epsilon = \left((1-(1-2p)^{w+1})/2\right)$ is hard, which concludes the proof of the theorem. □

Remark 1: According to the proof of Theorem 1, note that the considered linearised model of encryption is as secure as the problem of its secret key recovery under CPA is hard because from the security point of view, the model appears as an incarnation of the LPN problem, and hardness of the LPN search problem also implies hardness of the LPN decisional problem (see [27], for example).

Remark 2: In a special case when $\boldsymbol{x}_t = \boldsymbol{k}\boldsymbol{S}^t$, $t = 1, 2, \ldots, \tau$, the LPN problem considered in Theorem 1 is a specific incarnation of the LPN problem with a certain structure, which implies that instead of using a generic approach for solving the LPN problem, a technique known as the fast correlation attack (FCA) (see, for example [31]). On the other hand, even if the FCA-based approach is employed, again its complexity heavily depends on the parameter ϵ. Finally, note that the technique for solving LPN problems reported in [22] is an employment of the FCA paradigm. Accordingly, the considered nature of the underlying LPN problem has no impact on Theorem 1.

4 Homophonic encoder design criteria

From the above computational security evaluation, it is clear that the design of the homophonic encoder influences the computational complexity of cryptanalysis. In this section, we explicit code design criteria for homophonic coding, taking into account both the computational and information-theoretical security and the implementation complexity. Requirements can be expressed either as a function of the homophonic encoder $\boldsymbol{G}_{\mathrm{H}}$ given the error correction encoder $\boldsymbol{G}_{\mathrm{ECC}}$ or as a joint function of $\boldsymbol{G}_{\mathrm{H}}$ and $\boldsymbol{G}_{\mathrm{ECC}}$.

Indeed, the latter holds for a design of the encoding–encryption system from scratch, which includes a coding box for performing both the homophonic and error correction coding in a manner which fits the rate of the concatenated code to the given constraints. The former applies when upgrading existing systems within the encoding–encryption paradigm, in which case, the existing binary linear (m, n) error correction code which encodes m bits into a codeword from $GF(2^n)$ is replaced with a binary linear (m', n) block code with the same error correction capability but with $m' > m$. Then $m' - m$ random bits are concatenated with m information bits and mapped into the new m'-bits via a homophonic encoder, whose output is the input for the error correcting one.

4.1 Computational complexity design criteria

Recall from (2) that

$$C_{\mathrm{ECC}}(C_{\mathrm{H}}(\boldsymbol{a} \parallel \boldsymbol{u})) = [\boldsymbol{a} \parallel \boldsymbol{u}]G_{\mathrm{H}}G_{\mathrm{ECC}} = [\boldsymbol{a} \parallel \boldsymbol{u}]\boldsymbol{G}$$

where $\boldsymbol{G} = [g_{i,j}]_{i=1}^{m}{}_{j=1}^{n}$ is an $m \times n$ matrix containing both the homophonic and the error correction encoding.

We recall the basic requirements on the matrix $\boldsymbol{G}_{\mathrm{H}}$, as far as information-theoretical security is concerned [6, 32]:

Invertibility: The matrix $\boldsymbol{G}_{\mathrm{H}}$ should be an invertible matrix, so that the receiver can decode the homophonic encoding.
Security: The matrix $\boldsymbol{G}_{\mathrm{H}}$ should map $[\boldsymbol{a} \parallel \boldsymbol{u}]$ so that in the resulting vector, each bit of data from $\boldsymbol{a}$ (that is, each bit of the ciphertext) is affected by at least one random bit from $\boldsymbol{u}$.

This section contains additional guidelines to design a dedicated homophonic encoding which provides maximum complexity of the

underlying LPN problem for given implementation and communications overhead.

It is well known that the hardness of the LPN_ϵ problem in the average case heavily depends on the parameter ϵ (see [22, 23], for example). On the other hand, Theorem 1 implies that the parameter ϵ depends on the minimal value of the basic equations which should be linearly combined in order to eliminate the random variables from each equation of the system. Theorem 1 thus implies the following design criteria for the construction of the matrix $\boldsymbol{G}_{\mathrm{H}}$:

Weight: For a given error correcting code generator matrix $\boldsymbol{G}_{\mathrm{ECC}}$, and parameter w, specify the homophonic code matrix $\boldsymbol{G}_{\mathrm{H}}$, so that the resulting matrix $\boldsymbol{G}_{\mathrm{H}} = [g_{ij}^{(H)}]_{i=1}^{m}{}_{j=1}^{m}$ satisfies:

$$\sum_{i=1}^{m-\ell} g_{\ell+i,m-\ell+j}^{(H)} \geq w, \quad j = 1, m-\ell+2, \ldots, l.$$

Dependability: According to (7) and (8), the submatrix of the matrix $\boldsymbol{G}$ consisting of its $m-\ell$ last rows should be such that any of the columns is a linear combination of at least w other columns. Consider thus the submatrix $\boldsymbol{G}^*$ determined by the rows $m-\ell+1, m-\ell+2, \ldots, m$ and columns $1, 2, \ldots, n$ of the matrix $\boldsymbol{G}$. We require that no column of the matrix $\boldsymbol{G}^*$ is equal to a linear combination of w or less other columns of $\boldsymbol{G}^*$, that is

$$\mathrm{rank}(\boldsymbol{G}^*) \geq w+1.$$

4.2 Implementation design criteria

At the sender, both the homophonic and error correcting encodings are done via a single multiplication by $\boldsymbol{G} = \boldsymbol{G}_{\mathrm{H}}\boldsymbol{G}_{\mathrm{CEE}}$.

At the receiver, the error correction decoding and the homophonic decoding should be performed independently. First, the errors should be corrected by the error correction decoding, because the homophonic decoding requires error-free decoding input. This implies that in order to minimise the implementation complexity, a desirable property is sparseness of the related matrices $\boldsymbol{G}_{\mathrm{H}}$, $\boldsymbol{G}$, and $\boldsymbol{G}_{\mathrm{H}}^{-1}$:

Sparsity: For a given error correcting code generator matrix $\boldsymbol{G}_{\mathrm{ECC}}$, and a given security parameter w, specify the homophonic encoding matrix $\boldsymbol{G}_{\mathrm{H}}$, so that either it is sparse or the resulting matrix $\boldsymbol{G}$ is sparse to provide minimisation of the implementation complexity on the sender side, and at the same time, the matrix $\boldsymbol{G}_{\mathrm{H}}^{-1}$ is sparse in order to avoid too high computation overhead for the receiver.

5 Homophonic code constructions

Let us first write the $m \times m$ wiretap matrix $\boldsymbol{G}_{\mathrm{H}}$ and the $m \times n$ error correcting matrix $\boldsymbol{G}_{\mathrm{ECC}}$ as

$$\boldsymbol{G}_{\mathrm{H}} = \begin{bmatrix} \boldsymbol{G}_{\mathrm{H}}^{(1)} & \boldsymbol{G}_{\mathrm{H}}^{(2)} \\ \boldsymbol{I}_{m-l} & \boldsymbol{G}_{\mathrm{H}}^{(4)} \end{bmatrix}, \quad \boldsymbol{G}_{\mathrm{ECC}} = \begin{bmatrix} \boldsymbol{G}_{\mathrm{ECC}}^{(1)} \\ \boldsymbol{G}_{\mathrm{ECC}}^{(2)} \end{bmatrix} \tag{14}$$

where $\boldsymbol{G}_{\mathrm{H}}^{(1)}$ is an $l \times (m-l)$ matrix, $\boldsymbol{G}_{\mathrm{H}}^{(2)}$ an $l \times l$ matrix, $\boldsymbol{I}_{m-l}$ denotes the $(m-l) \times (m-l)$ identity matrix, $\boldsymbol{G}_{\mathrm{H}}^{(4)}$ is an $(m-l) \times l$ matrix written as

$$\boldsymbol{G}_{\mathrm{H}}^{(4)} = \begin{bmatrix} g_{\ell+1,m-\ell+1}^{(H)} & g_{\ell+1,m-\ell+2}^{(H)} & \cdots & g_{\ell+1,m}^{(H)} \\ g_{\ell+2,m-\ell+1}^{(H)} & g_{\ell+2,m-\ell+2}^{(H)} & \cdots & g_{\ell+2,m}^{(H)} \\ \vdots & \vdots & \cdots & \vdots \\ g_{m,m-\ell+1}^{(H)} & g_{m,m-\ell+2}^{(H)} & \cdots & g_{m,m}^{(H)} \end{bmatrix},$$

$\boldsymbol{G}_{\mathrm{ECC}}^{(1)}$ is an $(m-l) \times n$ matrix and $\boldsymbol{G}_{\mathrm{ECC}}^{(2)}$ is an $l \times n$ matrix, so

$$\begin{aligned} \boldsymbol{G} = \boldsymbol{G}_{\mathrm{H}}\boldsymbol{G}_{\mathrm{ECC}} &= \begin{bmatrix} \boldsymbol{G}_{\mathrm{H}}^{(1)} & \boldsymbol{G}_{\mathrm{H}}^{(2)} \\ \boldsymbol{I}_{m-l} & \boldsymbol{G}_{\mathrm{H}}^{(4)} \end{bmatrix} \begin{bmatrix} \boldsymbol{G}_{\mathrm{ECC}}^{(1)} \\ \boldsymbol{G}_{\mathrm{ECC}}^{(2)} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{G}_{\mathrm{H}}^{(1)}\boldsymbol{G}_{\mathrm{ECC}}^{(1)} + \boldsymbol{G}_{\mathrm{H}}^{(2)}\boldsymbol{G}_{\mathrm{ECC}}^{(2)} \\ \boldsymbol{G}_{\mathrm{ECC}}^{(1)} + \boldsymbol{G}_{\mathrm{H}}^{(4)}\boldsymbol{G}_{\mathrm{ECC}}^{(2)} \end{bmatrix}. \end{aligned}$$

5.1 Generic construction

We now give a general construction method for the matrix $\boldsymbol{G}_{\mathrm{H}}$. Choose first

$$\boldsymbol{G}_{\mathrm{H}}^{(1)} = \boldsymbol{0}_{l \times (m-l)}, \quad \boldsymbol{G}_{\mathrm{H}}^{(2)} = \boldsymbol{I}_l,$$

which satisfy the information-theoretical requirements.

Invertibility: Since $\boldsymbol{G}_{\mathrm{H}}$ is a square matrix, we can rephrase its invertibility using its determinant by asking

$$\det(\boldsymbol{G}_{\mathrm{H}}) \neq 0.$$

Using Schur complement, this is equivalent to

$$\det(\boldsymbol{G}_{\mathrm{H}}^{(2)} - \boldsymbol{G}_{\mathrm{H}}^{(1)}\boldsymbol{G}_{\mathrm{H}}^{(4)}) \neq 0.$$

The above choice of $\boldsymbol{G}_{\mathrm{H}}^{(1)}$ and $\boldsymbol{G}_{\mathrm{H}}^{(2)}$ gives $\det(\boldsymbol{I}_l) \neq 0$ which always holds, and the invertibility condition is taken care of.

Security: The matrix $\boldsymbol{G}_{\mathrm{H}}$ should map $[\boldsymbol{a} \parallel \boldsymbol{u}]$ so that in the resulting vector, each data bit from $\boldsymbol{a}$ is affected by at least one random bit from $\boldsymbol{u}$. Since

$$[\boldsymbol{a} \parallel \boldsymbol{u}] \begin{bmatrix} \boldsymbol{0}_{l \times (m-l)} & \boldsymbol{I}_l \\ \boldsymbol{I}_{m-l} & \boldsymbol{G}_{\mathrm{H}}^{(4)} \end{bmatrix} = [\boldsymbol{u}, \boldsymbol{a} + \boldsymbol{u}\boldsymbol{G}_{\mathrm{H}}^{(4)}],$$

it is enough that $\boldsymbol{G}_{\mathrm{H}}^{(4)}$ has no zero column to get that each data bit from $\boldsymbol{a}$ is affected by at least one random bit from $\boldsymbol{u}$.

We next look at the conditions coming from computational security. The weight condition can be rephrased as requiring that each column of $\boldsymbol{G}_{\mathrm{H}}^{(4)}$ has Hamming weight at least w, which automatically makes sure that $\boldsymbol{G}_{\mathrm{H}}^{(4)}$ has no zero column.

The dependability condition relates to the submatrix $\boldsymbol{G}^*$ determined by the rows $m-\ell+1, m-\ell+2, \ldots, m$ and columns $1, 2, \ldots, n$ of the matrix $\boldsymbol{G}$. Since $m-l$ counts the number of random bits, it is reasonable to assume that

$$m - l \leq l \Leftarrow m \leq 2l,$$

that is we use at most as many random bits as data bits. Since

$$\begin{aligned} \boldsymbol{G} = \boldsymbol{G}_{\mathrm{H}}\boldsymbol{G}_{\mathrm{ECC}} &= \begin{bmatrix} \boldsymbol{0}_{l \times (m-l)} & \boldsymbol{I}_l \\ \boldsymbol{I}_{m-l} & \boldsymbol{G}_{\mathrm{H}}^{(4)} \end{bmatrix} \begin{bmatrix} \boldsymbol{G}_{\mathrm{ECC}}^{(1)} \\ \boldsymbol{G}_{\mathrm{ECC}}^{(2)} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{G}_{\mathrm{ECC}}^{(2)} \\ \boldsymbol{G}_{\mathrm{ECC}}^{(1)} + \boldsymbol{G}_{\mathrm{H}}^{(4)}\boldsymbol{G}_{\mathrm{ECC}}^{(2)} \end{bmatrix} \end{aligned}$$

with, assuming w.l.o.g that $\boldsymbol{G}_{\mathrm{ECC}}$ is in systematic form

$$\begin{aligned} \boldsymbol{G}_{\mathrm{ECC}}^{(1)} &= \begin{bmatrix} g_{1,1}^{(\mathrm{ECC})} & g_{1,2}^{(\mathrm{ECC})} & \cdots & g_{1,n}^{(\mathrm{ECC})} \\ \vdots & \vdots & \cdots & \vdots \\ g_{m-\ell,1}^{(\mathrm{ECC})} & g_{m-\ell,2}^{(\mathrm{ECC})} & \cdots & g_{m-\ell,n}^{(\mathrm{ECC})} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{I}_{m-l} & \boldsymbol{0}_{(m-l)\times l} & \boldsymbol{P}_{(m-l)\times(n-m)} \end{bmatrix} \end{aligned}$$

and

$$\boldsymbol{G}^{(2)}_{\mathrm{ECC}} = \begin{bmatrix} g^{(\mathrm{ECC})}_{m-\ell+1,1} & g^{(\mathrm{ECC})}_{m-\ell+1,2} & \cdots & g^{(\mathrm{ECC})}_{m-\ell+1,n} \\ \vdots & \vdots & \cdots & \vdots \\ g^{(\mathrm{ECC})}_{m,1} & g^{(\mathrm{ECC})}_{m,2} & \cdots & g^{(\mathrm{ECC})}_{m,n} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{l\times(m-l)} & \boldsymbol{I}_l & \boldsymbol{Q}_{l\times(n-m)} \end{bmatrix},$$

we can write the $l \times n$ matrix $\boldsymbol{G}^*$ as

$$\begin{bmatrix} g^{(\mathrm{ECC})}_{2m-2l+1,1} & g^{(\mathrm{ECC})}_{2m-2l+1,2} & \cdots & g^{(\mathrm{ECC})}_{2m-2l+1,n} \\ \vdots & \vdots & \cdots & \vdots \\ g^{(\mathrm{ECC})}_{m,1} & g^{(\mathrm{ECC})}_{m,2} & \cdots & g^{(\mathrm{ECC})}_{m,n} \\ \boldsymbol{G}^{(1)}_{\mathrm{ECC}} & +\boldsymbol{G}^{(4)}_{\mathrm{H}}\boldsymbol{G}^{(2)}_{\mathrm{ECC}} & & \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{(2l-m)\times(2m-3l)} & \boldsymbol{I}_{2l-m}\boldsymbol{0}_{(2l-m)\times l} & \boldsymbol{P}' \\ & \boldsymbol{G}^{(1)}_{\mathrm{ECC}} + \boldsymbol{G}^{(4)}_{\mathrm{H}}\boldsymbol{G}^{(2)}_{\mathrm{ECC}} & \end{bmatrix}$$

with $\boldsymbol{P}' = \boldsymbol{P}'_{(2l-m)\times(n-m)}$. Now

$$\boldsymbol{G}^{(4)}_{\mathrm{H}}\boldsymbol{G}^{(2)}_{\mathrm{ECC}} = \boldsymbol{G}^{(4)}_{\mathrm{H}}[\boldsymbol{0}_{l\times(m-l)} \quad \boldsymbol{I}_l \quad \boldsymbol{Q}_{l\times(n-m)}] = [\boldsymbol{0}_{m-l} \quad \boldsymbol{G}^{(4)}_{\mathrm{H}} \quad \boldsymbol{G}^{(4)}_{\mathrm{H}}\boldsymbol{Q}_{l\times(n-m)}]$$

so that finally $\boldsymbol{G}^*$ is given by

$$\begin{bmatrix} \boldsymbol{0}_{(2l-m)\times(2m-3l)}\boldsymbol{I}_{2l-m} & \boldsymbol{0}_{(2l-m)\times l} & \boldsymbol{P}'_{(2l-m)\times(n-m)} \\ \boldsymbol{I}_{m-l} & \boldsymbol{G}^{(4)}_{\mathrm{H}} & \boldsymbol{G}^{(4)}_{\mathrm{H}}\boldsymbol{Q}_{l\times(n-m)} + \boldsymbol{P} \end{bmatrix}$$

with $\boldsymbol{P} = \boldsymbol{P}_{(m-l)\times(n-m)}$. The requirement is that

$$\mathrm{rank}(\boldsymbol{G}^*) \geq w + 1.$$

Since $l \leq m < n$, the rank of $\boldsymbol{G}^*$ is at most l, and it is enough to look at the rank of the $l \times m$ submatrix

$$\begin{bmatrix} \boldsymbol{0}_{(2l-m)\times(2m-3l)}\boldsymbol{I}_{2l-m} & \boldsymbol{0}_{(2l-m)\times l} \\ \boldsymbol{I}_{m-l} & \boldsymbol{G}^{(4)}_{\mathrm{H}} \end{bmatrix} \tag{15}$$

which varies from $m-l$ to l since the first $m-l$ columns are linearly independent. Thus, if $w+1 \leq m-l$, the dependency condition is satisfied naturally. Otherwise, we need to build $\boldsymbol{G}^{(4)}_{\mathrm{H}}$ such that k of its columns, $k = 1, \ldots, 2l-m$, are linearly independent from the $m-l$ first columns of the above matrix. To do so, it is enough to consider the $2l-m$ first columns of $\boldsymbol{G}^{(4)}_{\mathrm{H}}$, and we consider the truncated matrix (15)

$$\begin{bmatrix} \boldsymbol{0}_{(2l-m)\times(2m-3l)}\boldsymbol{I}_{2l-m} & \boldsymbol{0}_{(2l-m)\times 2l-m} \\ \boldsymbol{I}_{m-l} & \boldsymbol{A} \end{bmatrix}$$

where $\boldsymbol{A}$ contains the $2l-m$ first columns of $\boldsymbol{G}^{(4)}_{\mathrm{H}}$. Write

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix},$$

where $\boldsymbol{A}_1$ is a $(2m-3l)\times(2l-m)$ matrix, and $\boldsymbol{A}_2$ a square $2l-m$ matrix. To control the rank of $\boldsymbol{G}^*$, we set $\boldsymbol{A}_1 = \boldsymbol{0}$ and

$$\mathrm{rank}(\boldsymbol{G}^*) = m - l + k$$

where k is the number of columns of $\boldsymbol{A}_2$ which are linearly independent from the matrix

$$\begin{bmatrix} \boldsymbol{0}_{(2l-m)\times(2m-3l)}\boldsymbol{I}_{2l-m} \\ \boldsymbol{I}_{m-l} \end{bmatrix}. \tag{16}$$

Setting $\boldsymbol{A}_2 = \boldsymbol{0}$ makes the computation easier. Indeed, to get k such columns, it is enough to pick k columns from the $2l-m$ identity matrix. This might give some columns with very few ones, which looks like contradicting the weight condition. However, columns with higher Hamming weight can be easily obtained by taking linear combinations of the columns without changing the rank.

Dependability and weight: Since

$$\mathrm{rank}(\boldsymbol{G}^*) = m - l + k$$

where k is the number of columns of the submatrix of $\boldsymbol{G}^{(4)}_{\mathrm{H}}$ formed by taking its first $2m-l$ columns and last $2m-l$ rows which are linearly independent from (16), it is enough to ask

$$k \geq w + 1 + l - m\,.$$

To ensure that each column of $\boldsymbol{G}^{(4)}_{\mathrm{H}}$ has Hamming weight w, it is enough to take linear combinations of the columns.

Sparsity: The choice of $\boldsymbol{G}^{(1)}_{\mathrm{H}} = \boldsymbol{0}_{l\times(m-l)}$ and $\boldsymbol{G}^{(2)}_{\mathrm{H}} = \boldsymbol{I}_l$ makes the l first rows of $\boldsymbol{G}_{\mathrm{H}}$ as sparse as possible, since removing any 0 would make the matrix non-invertible. Also, the way the dependability condition is constructed is optimal in the sense that it starts with the least number of 1 to get the wanted rank, and then obtains the desired Hamming weight of each column by linear combinations.

Examples of constructions are given in the Appendix.

6 LPN-based encryption with/without homophonic coding

We start by reporting two known randomised encryption schemes, after which we propose a new one based on our analysis above. Following [14], the security of the three schemes against key recovery as a function of the underlying LPN problem is discussed next in Section 6.2.

6.1 Three LPN-based encryption schemes

We summarise the symmetric encryption schemes [2, 33].

Encryption scheme 1 [2]: Let $C\colon\{0,1\}^{\ell} \to \{0,1\}^n$ be an (n, ℓ, d) error correcting code, given by $C(\boldsymbol{a}) = \boldsymbol{a}\cdot\boldsymbol{G}$, where $\boldsymbol{G}$ is a binary $\ell \times n$ matrix, with minimal distance d and correction capacity $t = \lfloor (d-1)/2 \rfloor$, which is publicly known. Let S be the $k \times n$ cryptosystem secret key. To encrypt an ℓ-bit vector $\boldsymbol{a}$, draw a k-bit random vector $\boldsymbol{u}$ and compute

$$z = C(\boldsymbol{a}) \oplus \boldsymbol{u}\cdot\boldsymbol{S} \oplus \boldsymbol{v}, \tag{17}$$

where $\boldsymbol{v} \leftarrow \mathrm{Ber}_{n,p}$ is an n-bit noise vector whose bits are (independently) 1 with probability p and 0 with probability $1-p$. The ciphertext $(\boldsymbol{u}, z)$ is decrypted by computing $z \oplus \boldsymbol{u}\boldsymbol{S} = C(\boldsymbol{a}) \oplus \boldsymbol{v}$, and decoding, if possible, the resulting value (otherwise a 'decryption error' is returned). The message, if needed, is padded, so its length is the smallest multiple of ℓ, and encoded block-wise.

Encryption scheme 2 [33]: This is a variation of [2], which was proved secure without key-dependent messages (KDM), and did not achieve linear time efficiency. In [33], ciphertexts are a constant factor larger than the plaintexts, and both encryption and decryption are computed by Boolean circuits of (approximately) linear size (in the message length), which is close to optimal even for standard CPA security.

Let $\ell = \ell(k)$ be the message length which is an arbitrary polynomial in the security parameter k (shorter messages are zero-padded), and let $\epsilon = 2^{-m}$ and $0 < \delta < 1$ be constants. The scheme [33] uses good $(\ell(k), n = n(k))$ binary linear codes with $n \times \ell$ binary generator matrix $\boldsymbol{G} = \boldsymbol{G}_{\ell}$, whose efficient decoding algorithm D corrects up to $(\epsilon + \delta)\cdot n$ errors.

Let $N = N(k)$ be an arbitrary polynomial controlling the trade-off between the key-length and the time complexity of the scheme. The private key $\boldsymbol{S}$ is chosen uniformly at random from $\{0,1\}^{k\times N}$.

Encryption: For a message $\boldsymbol{A} \in \{0,1\}^{\ell \times N}$, choose a balanced random matrix $\boldsymbol{U} \leftarrow \{0,1\}^{n \times k}$ and a random noise matrix $\boldsymbol{V} \leftarrow \mathrm{Ber}_{\epsilon}^{n \times N}$. The ciphertext is $(\boldsymbol{U}, \boldsymbol{Z})$, where

$$\boldsymbol{Z} = \boldsymbol{G} \cdot \boldsymbol{A} \oplus \boldsymbol{U} \cdot \boldsymbol{S} \oplus \boldsymbol{V}. \tag{18}$$

Decryption: Given the ciphertext $(\boldsymbol{U}, \boldsymbol{Z})$, apply the decoding algorithm D to each column of the matrix $\boldsymbol{Z} \oplus \boldsymbol{U} \cdot \boldsymbol{S}$. The decryption algorithm will fail only when there exists a column in $\boldsymbol{V}$ whose Hamming weight is larger than $(\epsilon + \delta)n$.

(New) Encryption scheme 3: Recall the model of Section 2.2, and let $C_{\mathrm{H}}(\cdot)$ and $C_{\mathrm{H}}^{-1}(\cdot)$ denote the homophonic encoder and decoder, respectively, which perform a mapping $\{0,1\}^m \to \{0,1\}^m$, and let $\boldsymbol{r} = [r_i]_{i=1}^{m-\ell}$ be an $(m-\ell)$-dimensional binary random vector where each r_i is a realisation of the binary random variable R_i such that $\Pr(R_i = 1) = \Pr(R_i = 0) = 1/2$, $i = 1, 2, \ldots, m-\ell$. We next propose a symmetric key encryption scheme based on our analysis.

Encryption: compute $C_{\mathrm{ECC}}(C_{\mathrm{H}}(\boldsymbol{a} \parallel \boldsymbol{r}))$ where $\parallel$ denotes the concatenation, and generate the ciphertext:

$$z = C_{\mathrm{ECC}}(C_{\mathrm{H}}(\boldsymbol{a} \parallel \boldsymbol{r})) \oplus \boldsymbol{u} \cdot \boldsymbol{S} \oplus \boldsymbol{v}. \tag{19}$$

Decryption: Assuming the pair $(\boldsymbol{u}, z)$ is available, decrypt the ciphertext as follows:

$$\boldsymbol{a} = \mathrm{tcat}_{\ell}(C_{\mathrm{H}}^{-1}(C_{\mathrm{ECC}}^{-1}(z \oplus \boldsymbol{u} \cdot \boldsymbol{S}))), \tag{20}$$

where $\mathrm{tcat}_{\ell}(\cdot)$ truncates the argument vector to the first ℓ bits and we assume that the code $C_{\mathrm{ECC}}(\cdot)$ with corresponding inverse $C_{\mathrm{ECC}}^{-1}(\cdot)$ correct the errors introduced by a binary symmetric channel with crossover probability p.

Encryption scheme 3 follows the paradigm elements of encryption schemes 1 [2] and 2 [33] and consequently, all three schemes share the following properties: (i) they are symmetric encryption based on the LPN problem hardness; (ii) they use controlled noise (randomness) and error correction coding; (iii) they rely on simple binary additions/multiplications and vector/matrix operations. Encryption scheme 3 further employs homophonic coding to provide an enhanced security.

6.2 Security with/without homophonic coding

The security of encryption schemes 1 [2], 2 [33], and of the newly proposed encryption scheme 3 relies on the hardness of solving the LPN problem. Since the employment of homophonic encoding can be considered as a randomised mapping of the plaintext, encryption scheme 3 has at least the same security features as encryption schemes 1 and 2 assuming that the same noise is employed, and consequently, we directly have that it is as secure as the underlying LPN problem is hard. We thus only need to consider the hardness of the underlying LPN problem in these schemes.

Table 2 Comparison of the main features of the proposed homophonic coding-based LPN encryption with those of encryption schemes 1 [2] and 2 [33]. 'Balanced random bit' refers to a bit which takes values 0 and 1 with probability 1/2

	parameters of the underlying LPN problem	expected # of unknown balanced random bits involved in a ciphertext bit
LPN encryptions [2, 33]	k, n, ϵ	0
proposed homophonic coding-based LPN encryption	$\epsilon^* = \frac{1-(1-2p)^{(m-\ell)/2}}{2}$ typically: $k^* \ll k$ $n^* \simeq n$, $p \ll \epsilon^*$	$(m-\ell)/2$

Underlying LPN problems: Let $\mathrm{LPN}_{\epsilon}(k, n)$ denote the underlying LPN problem invoked in the security of encryption schemes 1 and 2. Since the proposed encryption scheme 3 uses a dedicated homophonic encoding which involves pure randomness, its underlying LPN problem is different, and will be denoted by $\mathrm{LPN}_{\epsilon^*}(k^*, n^*)$.

We assume that encryption scheme 3 is such that Theorem 1 implies $\epsilon^* = \left(\left(1-(1-2p)^{(m-\ell)/2}\right)/2\right)$. Referring to the best known algorithms for solving the LPN problem [22–25], its hardness heavily depends on the parameter ϵ and when it increases, the LPN complexity increases. For the same parameters k, n, encryption scheme 3 provides substantially higher security compared with the previously reported ones with $\epsilon = p$.

Pure randomness: The considered encryption has balanced random bits, unknown to the receiver, in each ciphertext bit, which are easily learnt only if the secret is known. The homophonic encryption scheme inserts pure randomness into each bit of the ciphertext which can easily removed when the secret key is known, but removing these balanced random bits from the ciphertext without knowledge of the secret key is as hard as solving a certain LPN problem. Furthermore: (i) the algebraic representation of encryption scheme 3 shows that each ciphertext bit is affected with the expected number of $(m-\ell)/2$ random bits which are chosen uniformly at random with probability 1/2; (ii) no purely random bit is involved into a ciphertext bit in the schemes from [2, 33].

Indistinguishable CPA (IND-CPA) and indistinguishable chosen ciphertext attack (IND-CCA): All three encryption schemes are IND-CPA secure, assuming that the underlying LPN problem is hard: for encryption schemes 1 and 2, IND-CPA security has been proven in [2, 33], respectively, and IND-CPA security of the proposed encryption scheme 3 is a direct consequence of the fact that it is equivalent to encryption scheme 1 when the messages are subject of error correction and homophonic encoding instead error correction encoding only. The chosen ciphertext attack (CCA) security of encryption scheme 3 can be achieved as in [2]. The most obvious way to get an encryption scheme secure against chosen ciphertext attacks from one secure against chosen plaintext attacks is to add message authenticity, e.g. by using a message authentication code.

Besides the CPA-related indistinguishability (under the hardness of the underlying LPN problem), this paper also considers the hardness of recovering the secret key by processing the algebraic equations corresponding to the encryption process which employs homophonic coding. In particular, the security goals of this paper are related to the complexity of a generic algebraic attack against the algebraic representation of the scheme under CPA.

A comparison summary and final remarks: In Table 2, we compare the main security features of the proposed scheme with the ones from [2, 33]. The illustrative numerical values given in Table 2 are based on the results reported in [2, 23]. Finally, note that one of the roles of the encryption schemes from [33] is to provide security against certain KDM attacks, which is beyond the scope of this paper.

7 Conclusion

The paper addresses the security evaluation and the design of a homophonic code for a class of randomised encryption schemes, used in the context of the encoding–encryption paradigm, reported and analysed from information-theoretic point of view in [6]. The design is based on the guidelines implied by security evaluation from the computational complexity point of view. Particularly, this paper provided: (i) a security evaluation of the randomised encryption schemes from the computational complexity point of view; (ii) guidelines for designing a dedicated homophonic coding implied by the performed security evaluation; (iii) constructive dedicated homophonic codes which provide the desired level of security, with low implementation overhead.

Security evaluation has shown that a lower bound on the security of the considered encryption corresponds to the hardness of recovering the secret key based on the algebraic representation

of the linearised encryption in CPA scenario. It was shown that the secret key recovery is at least as hard as the LPN_{ϵ} problem when, assuming an appropriate design, the corrupting noise is $\epsilon = \left((1-(1-2p)^{w+1})/2\right)$ and $p < 0.5$ and w are the system parameters. Note that in the average complexity consideration, the LPN problem corresponding to the parameter ϵ is much harder than the one with the parameter p. Assuming an appropriate selection of the scheme parameters, the complexity of the secret key recovery based on its algebraic representation appears almost as hard as an exhaustive search over all possible secret keys.

The results of security evaluation are considered as guidelines for design of a dedicated homophonic encoder which provides a desired security level and minimises the implementation complexity. Assuming that the homophonic code should be linear, besides the basic requirement on the invertibility and the mixing properties of the generator matrix, the following three additional criteria are pointed out and specified in Sections 4.1 and 4.2: (i) weight on columns of the generator matrix; (ii) rank of the generator matrix; (iii) sparsity of the generator matrix. The criteria (i) and (ii) appear as an implication of the security requirements, and the criterion (iii) is related to minimisation of the implementation overhead. The previous design criteria are employed for design of a dedicated homophonic code. A generic design of the homophonic coding dedicated to the considered security enhanced communication system is proposed and it is shown that the design fulfils all the given criteria.

As an illustration, a comparison of the encryption schemes based on the LPN problem with and without homophonic coding is considered showing the benefits which can be obtained when appropriate homophonic coding is employed.

8 Acknowledgments

The authors thank the anonymous reviewers for their careful checking of the initial version of this paper and valuable comments and suggestions resulting in an improved presentation of the results. The research of F.O. was supported by the Singapore National Research Foundation under Research Grant NRF-RF2009-07. This work was done partly while M.J.M. was visiting the division of mathematical sciences, Nanyang Technological University, Singapore. M.J.M. is partly supported via the Project of the Ministry for Education, Science and Technology, Republic Serbia.

9 References

[1] Rivest, R., Sherman, T.: 'Randomized encryption techniques'. Proc. Advances in Cryptology – CRYPTO '82, Boston, MA, USA, 1983, pp. 145–163

[2] Gilbert, H., Robshaw, M.J.B., Seurin, Y.: 'How to encrypt with the LPN problem'. Proc. 35th Int. Colloquium on Automata, Languages and Programming – ICALP 2008, Part II, Reykjavik, Iceland, July 2008 (LNCS, **5126**), pp. 679–690

[3] Mihaljević, M.J., Imai, H.: 'An approach for stream ciphers design based on joint computing over random and secret data', *Computing*, 2009, **85**, (1–2), pp. 153–168

[4] Khiabani, Y.S., Wei, S., Yuan, J., *et al.*: 'Enhancement of secrecy of block ciphered systems by deliberate noise', *IEEE Trans. Inf. Forensics Sec.*, 2012, **7**, (5), pp. 1604–1613

[5] Wei, S., Wang, J., Yin, R., *et al.*: 'Trade-off between security and performance in block ciphered systems with erroneous ciphertexts', *IEEE Trans. Inf. Forensics Sec.*, 2013, **8**, (4), pp. 636–645

[6] Oggier, F., Mihaljević, M.J.: 'An information-theoretic security evaluation of a class of randomized encryption schemes', *IEEE Trans. Inf. Forensics Sec.*, 2014, **9**, (2), pp. 158–168

[7] Khiabani, Y.S., Wei, S.: 'A joint Shannon cipher and privacy amplification approach to attaining exponentially decaying information leakage', *Inf. Sci.*, 2016, **357**, (1), pp. 6–22

[8] Wyner, A.D.: 'The wire-tap channel', *Bell Syst. Tech. J.*, 1975, **54**, pp. 1355–1387

[9] GSM Technical Specifications: 'Digital cellular telecommunications system (Phase 2+); physical layer on the radio path; general description', European Telecommunications Standards Institute (ETSI), TS 100 573 (GSM 05.01). Available at http://www.etsi.org

[10] GSM Technical Specifications: 'Digital cellular telecommunications system (Phase 2+); channel coding', European Telecommunications Standards Institute (ETSI), TS 100 909, GSM 05.03. Available at http://www.etsi.org

[11] Gunther, C.G.: 'A universal algorithm for homophonic coding'. Proc. Advances in Cryptology – EUROCRYPT'88, Davos, Switzerland, May 1988 (LNCS, **330**), pp. 405–414

[12] Jendal, H.N., Kuhn, Y.J.B., Massey, J.L.: 'An information-theoretic treatment of homophonic substitution'. Proc. Advances in Cryptology – EUROCRYPT'89, Houthalen, Belgium, April 1989 (LNCS, **434**), pp. 382–394

[13] Massey, J.L.: 'Some applications of source coding in cryptography', *Eur. Trans. Telecommun.*, 1994, **5**, pp. 421–429

[14] Mihaljević, M.J., Imai, H.: 'Employment of homophonic coding for improvement of certain encryption approaches based on the LPN problem'. Symmetric Key Encryption Workshop – SKEW 2011, Lyngby, Denmark, February 2011. Available at http://skew2011.mat.dtu.dk/program.html

[15] Hoshi, M., Han, T.S.: 'Interval algorithm for homophonic coding', *IEEE Trans. Inf. Theory*, 2001, **47**, (3), pp. 1021–1031

[16] Simoesa, D.R., Portugheis, J., da Rocha, V.C.Jr.: 'Universal homophonic coding scheme using differential encoding and interleaving', *Inf. Process. Lett.*, 2013, **113**, pp. 628–633

[17] Gulcu, T.C., Barg, A.: 'Achieving secrecy capacity of the wiretap channel and broadcast channel with a confidential component', *IEEE Trans. Inf. Theory*, 2017, **63**, (2), pp. 1311–1324

[18] Yin, R., Wei, S., Yuan, J., *et al.*: 'Tradeoff between reliability and security in block ciphering systems with physical channel errors'. Proc. 2010 IEEE Military Communications Conf. – MILCOM 2010, San Jose, USA, October 2010, pp. 2156–2161

[19] Alekhnovich, M.: 'More on average case vs approximation complexity'. Proc. 44th Annual IEEE Symp. on Foundations of Computer Science – FOCS'03, Cambridge, USA, October 2003, doi: 10.1109/SFCS.2003.1238204

[20] Katz, J., Lindell, Y.: '*Introduction to modern cryptography*' (CRC Press, Boca Ratton, 2007)

[21] Blum, A., Kalai, A., Wasserman, H.: 'Noise-tolerant learning, the parity problem, and the statistical query model', *J. ACM*, 2003, **50**, (4), pp. 506–519

[22] Fossorier, M., Mihaljević, M.J., Imai, H., *et al.*: 'An algorithm for solving the LPN problem and its application to security evaluation of the HB protocols for RFID authentication'. Proc. Progress in Cryptology – INDOCRYPT 2006, Kolkata, India, December 2006 (LNCS, **4329**), pp. 48–62

[23] Levieil, E., Fouque, P.-A.: 'An improved LPN algorithm'. Proc. 5th Int. Conf. on Security and Cryptography for Networks – SCN 2006, Maiori, Italy, September 2006 (LNCS, **4116**), pp. 348–359

[24] Guo, Q., Johansson, T., Löndah, C.: 'Solving LPN using covering codes'. Proc. Advances in Cryptology – ASIACRYPT2014, Part I, Kaoshiung, Taiwan, R.O.C., December 2014 (LNCS, **8873**), pp. 1–20

[25] Bogos, S., Tramer, F., Vaudenay, S.: 'On solving LPN using BKW and variants: implementation and analysis', *Cryptogr. Commun.*, 2016, **8**, pp. 331–369

[26] Zhang, B., Jiao, L., Wang, M.: 'Faster algorithms for solving LPN'. Proc. Advances in Cryptology – EUROCRYPT 2016, Part I, Vienna, Austria, May 2016 (LNCS, **9665**), pp. 168–195

[27] Katz, J., Shin, J.S.: 'Parallel and concurrent security of the HB and HB^{+} protocols'. Proc. Advances in Cryptology – EUROCRYPT 2006, St. Petersburg, Russia, May 2006 (LNCS, **4004**), pp. 73–87

[28] Berlekamp, E.R., McEliece, R.J., van Tilborg, H.C.A.: 'On the inherent intractability of certain coding problems', *IEEE Trans. Inf. Theory*, 1978, **24**, pp. 384–386

[29] Kosei, E., Kunihiro, N.: 'On the security proof of an authentication protocol from Eurocrypt 2011'. Proc. 9th Int. Workshop on Security – IWSEC 2014, Hirosaki, Japan, August 2014 (LNCS, **8639**), pp. 187–203

[30] Gallager, R.G.: 'Low-density parity-check codes', *IRE Trans. Inf. Theory*, 1968, **IT-8**, (1), pp. 21–28

[31] Agren, M., Londahl, C., Hell, M., *et al.*: 'A survey on fast correlation attacks', *Cryptogr. Commun.*, 2012, **4**, (3–4), pp. 173–202

[32] Mihaljević, M.J., Oggier, F.: 'A wire-tap approach to enhance security in communication systems using the encoding-encryption paradigm'. Proc. IEEE 17th Int. Conf. on Telecommunications 2010 – ICT 2010, Doha, Qatar, April 2010, pp. 484–489

[33] Applebaum, B., Cash, D., Peikert, C., *et al.*: 'Fast cryptographic primitives and circular-secure encryption based on hard learning problems'. Proc. Advances in Cryptology – CRYPTO 2009, Santa Barbara, USA, August 2009 (LNCS, **5677**), pp. 595–618

10 Appendix

10.1 Examples of constructions

Take $m = 2l$ so that $m - l = l$, and

$$\boldsymbol{G}_{\mathrm{H}} = \begin{bmatrix} \boldsymbol{G}_{\mathrm{H}}^{(1)} & \boldsymbol{G}_{\mathrm{H}}^{(2)} \\ \boldsymbol{I}_l & \boldsymbol{G}_{\mathrm{H}}^{(4)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_l & \boldsymbol{I}_l \\ \boldsymbol{I}_l & \boldsymbol{I}_l \end{bmatrix}.$$

The matrix $\boldsymbol{G}_{\mathrm{H}}$ is clearly invertible. The Hamming weight of each column of $\boldsymbol{G}_{\mathrm{H}}^{(4)}$ is 1, thus, w must be taken to be 0 or 1. The matrix $\boldsymbol{G}_{\mathrm{H}}^{(2)}$ is chosen to be zero for increasing the sparsity of $\boldsymbol{G}_{\mathrm{H}}$.

As a toy example, consider the (7, 4) Hamming code with

$$\boldsymbol{G}_{\mathrm{ECC}} = \begin{bmatrix} & 1 & 1 & 0 \\ \boldsymbol{I}_4 & 1 & 0 & 1 \\ & 0 & 1 & 1 \\ & 1 & 1 & 1 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{G}_{\mathrm{H}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

We have that

$$\boldsymbol{G}_{\mathrm{H}}\boldsymbol{G}_{\mathrm{ECC}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

The matrix $\boldsymbol{G}^*$ is thus

$$\boldsymbol{G}^* = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

Since the requirement is that the rank of $\boldsymbol{G}^*$ is at least w, this is clearly satisfied here since $w = 1$. Note that

$$\boldsymbol{G}_{\mathrm{H}}^{-1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

and the cost of encoding and decoding the homophonic code is the same. In order to increase w to 2, we could take

$$\boldsymbol{G}_{\mathrm{H}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}.$$

Then, continuing with the (7, 4) Hamming code

$$\boldsymbol{G}_{\mathrm{H}}\boldsymbol{G}_{\mathrm{ECC}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix},$$

where

$$\boldsymbol{G}^* = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

has rank $w = 2$ as required. This time

$$\boldsymbol{G}_{\mathrm{H}}^{-1} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

and as expected, increasing w correspondingly decreases the sparsity of $\boldsymbol{G}_{\mathrm{H}}$ and its inverse.