Hindawi
Security and Communication Networks
Volume 2017, Article ID 4321296, 10 pages
https://doi.org/10.1155/2017/4321296

Research Article

Semantically Secure Symmetric Encryption with Error Correction for Distributed Storage

Juha Partala

Physiological Signal Analysis Team, Center for Machine Vision and Signal Analysis, Oulu, Finland

Correspondence should be addressed to Juha Partala; juha.partala@oulu.fi

Received 8 February 2017; Revised 21 April 2017; Accepted 7 May 2017; Published 22 June 2017

Academic Editor: Huaizhi Li

A distributed storage system (DSS) is a fundamental building block in many distributed applications. It applies linear network coding to achieve an optimal tradeoff between storage and repair bandwidth when node failures occur. Additively homomorphic encryption is compatible with linear network coding. The homomorphic property ensures that a linear combination of ciphertext messages decrypts to the same linear combination of the corresponding plaintext messages. In this paper, we construct a linearly homomorphic symmetric encryption scheme that is designed for a DSS. Our proposal provides simultaneous encryption and error correction by applying linear error correcting codes. We show its IND-CPA security for a limited number of messages based on binary Goppa codes and the following assumption: when dividing a scrambled generator matrix $\widehat{\mathbf{G}}$ into two parts $\widehat{\mathbf{G}_1}$ and $\widehat{\mathbf{G}_2}$, it is infeasible to distinguish $\widehat{\mathbf{G}_2}$ from random and to find a statistical connection between $\widehat{\mathbf{G}_1}$ and $\widehat{\mathbf{G}_2}$. Our infeasibility assumptions are closely related to those underlying the McEliece public key cryptosystem but are considerably weaker. We believe that the proposed problem has independent cryptographic interest.

1. Introduction

The world's ability to generate, process, and store information is growing at an exponential rate [1]. The Internet of Things (IoT) has enabled objects to collect and share a vast amount of data enabling new applications and improving efficiency. In distributed IoT, intelligence is pushed to the very edge of the networks. Such decentralized approach has created challenges regarding the security and privacy of the collected data [2]. A distributed storage system (DSS) is a widely used technology for storing data in a reliable way. It is one of the essential building blocks for distributed applications. Such a system consists of a collection of n storage nodes that may be individually unreliable but apply redundancy to make the system reliable as a whole. Coding schemes are applied to ensure its reliability and to reduce the bandwidth required for repair. In particular, linear network coding has turned out to offer good performance both in theory and in practice.

Complications arise if we cannot be certain that the storage nodes are well-behaved. Encryption needs to be applied to ensure the confidentiality of the data. However, traditional cryptographic primitives are ill-suited for network coding which requires that data packets from different nodes can be combined according to the coding scheme. Secure network coding [3–6] has been applied to ensure confidentiality in the information-theoretic security model. However, secure network coding incurs a cost on the storage capacity of the system. It decreases exponentially with the number of compromised nodes [7]. Furthermore, in many cases the storage nodes are provided by a third party storage service provider leading to systems with zero secrecy capacity [8].

In this paper, we consider the confidentiality of network coding and, in particular, distributed storage systems in a setting where the adversary has complete control of the nodes but is computationally bounded. We devise a linear error correcting code based symmetric additively homomorphic encryption scheme that is compatible with linear network coding. There are several advantages of our scheme compared to ordinary encryption:

(1) Linear network coding can be applied as if working directly with the plaintext messages. Linear operations on the ciphertext space transfer to the plaintext space upon decryption.

(2) The encrypted parts of the file do not disclose which part is which. The part information can be kept in the plaintext domain. It makes it impossible for the storage nodes or the adversary to eavesdrop on which subsets of the data the user requests.

(3) The plaintext data can be first authenticated and then encrypted. For storage systems, this ordering is often desirable to ensure plaintext integrity. Our scheme can support this functionality with an additively homomorphic message authentication code such as [9] meaning that all linear combinations of the plaintext messages are authenticated.

(4) Our scheme provides simultaneous encryption and error correction.

There are encryption schemes possessing additively homomorphic properties such as the Goldwasser-Micali scheme [9] and the Paillier cryptosystem [10]. However, to apply coding schemes for distributed storage we need flexibility in choosing the ciphertext space field which, for efficiency reasons, is often an extension field of the binary field $\mathbb{F}_2$ when working with big data [11]. The required flexibility is not provided by existing proposals.

We construct a symmetric encryption scheme AddHomSE that is homomorphic from $(\mathbb{F}_q^n, +)$ to $(\mathbb{F}_q^k, +)$, where $k < n$ and $\mathbb{F}_q$ is a finite field. In particular, our security proofs are shown in the case where $\mathbb{F}_q = \mathbb{F}_2$ the binary field resulting in a scheme that is homomorphic from the additive group $(\mathbb{F}_2^n, \oplus)$ to $(\mathbb{F}_2^k, \oplus)$. We also show that our construction is semantically (IND-CPA) secure in the standard model (on $\mathbb{F}_2$) for a fixed number of messages showing that it *provides indistinguishability for each individual part of the file*. We apply problems that are closely related to the McEliece cryptosystem [12]. In particular, we formulate an assumption that is related to the pseudorandomness of the McEliece generator matrix. However, our assumption is much weaker. We believe that the corresponding problem has cryptographic interest in its own right.

The paper is organized as follows. In Section 2 we present work that is related to ours. Section 3 describes the preliminaries for the rest of the paper. We formulate AddHomSE in Section 4. We show that the scheme is IND-CPA secure for a limited number of messages in Sections 5 and 6. In Section 7 we consider the infeasibility of the applied problems and discuss how the scheme can be applied in practice with compact keys. Finally, Section 8 provides the conclusion.

2. Related Work

The theory of confidentiality of distributed storage is related to that of network coding. Cai and Yeung were the first to consider secure network coding [3, 4]. In their security model, a passive wiretapper is able to eavesdrop on a subset of the links between nodes. The adversary is computationally unbounded and privacy is considered information-theoretically. A similar model was considered in [13–16]. The security model of eavesdropping nodes, which is more natural for distributed storage, was suggested by Pawar et al. [8]. In their model, a computationally unbounded eavesdropper can access data on her selection of the nodes. The maximum file size that can be stored with information-theoretic security in the DSS using an optimal bandwidth MDS code (with exact repair) is called the *secrecy capacity* of the DSS. Regenerating codes achieving the secrecy capacity were suggested by Shah et al. [5]. Regenerating codes and locally repairable secure codes that achieve minimum storage requirements for a DSS were suggested by Rawat et al. [6]. Multiple simultaneous node failures, cooperative regenerating codes, and their secrecy capacity were considered in [17]. Kosut et al. considered networks where a node behaves traitorously [18]. Multiple nodes containing adversarial errors were considered by Dikaliotis et al. [19]. Pawar et al. considered an active omniscient adversary that has complete knowledge of the data on all nodes and can corrupt b nodes, where $2b < k$ [20].

The concept of homomorphic encryption was introduced by Rivest et al. [21]. While fully homomorphic encryption enables arbitrary computations on ciphertexts, many proposed schemes have homomorphic properties over specific operations. For example, RSA [22] is homomorphic over multiplication. Additively homomorphic schemes enable the computation of linear combinations of the ciphertexts. For the Goldwasser-Micali scheme [9] and the Paillier cryptosystem [10] multiplication in the ciphertext space corresponds to addition in the plaintext space. The scheme proposed by Lyubashevsky et al. is additively homomorphic with a polynomial ring as the ciphertext space [23]. Other asymmetric schemes with additively homomorphic properties can be found, for example, from [24–29]. The functionality of public key encryption incurs a computational burden that is not needed in certain situations. For many applications, symmetric encryption suffices. Few symmetric schemes with the additive homomorphic property have been proposed. Some constructions, mostly concentrating on realizing fully homomorphic encryption, can be found from [30–33]. In addition, the ciphertext and plaintext spaces in these schemes cannot be easily applied with linear network coding where we want to work with extension fields of the binary field $\mathbb{F}_2$ for efficiency reasons.

3. Preliminaries

3.1. Notation. Standard notation will be used for probabilistic algorithms [34]. We denote by $y \leftarrow \mathsf{A}(x; r)$ the result of running a probabilistic algorithm A on input x with randomness r and setting y to be equal to the output. We denote the uniform probability distribution on a set X by $U(X)$. If A is a random variable and $\mathscr{F}$ is a distribution, we denote $A \sim \mathscr{F}$ when A is distributed according to $\mathscr{F}$. A probability ensemble $X = \{X_k\}_{k\in\mathbb{N}}$ is a collection of random variables indexed by the integers. The problem of computationally distinguishing between two probability ensembles A and B is denoted by $D(A, B)$.

Whenever we refer to indistinguishability of probability ensembles, we mean computational indistinguishability unless stated otherwise. Security proofs are considered in the standard model. That is, all algorithms are considered to be probabilistic polynomial time (PPT) and time complexity is

considered in the average case. The success probability (called the *advantage*) of an adversary A on a problem P is considered asymptotically as a function of a security parameter s and is denoted by $\mathbf{Adv}_{\mathsf{A}}^{\mathsf{P}}(s)$. A function $\epsilon : \mathbb{N} \to \mathbb{R}$ is negligible if for every $n \in \mathbb{N}$ there is $k' \in \mathbb{N}$ such that $\epsilon(k) \leq 1/k^n$ for every $k \geq k'$. A problem P is considered infeasible if for all PPT algorithms A the advantage $\mathbf{Adv}_{\mathsf{A}}^{\mathsf{P}}(s)$ is negligible.

3.2. Dynamic Distributed Storage. Let F be a file consisting of M elements from a finite field $\mathbb{F}_q$. A dynamic distributed storage system (DSS) consists of n live nodes each storing α symbols over $\mathbb{F}_q$. These nodes can be individually unreliable but the system is designed to apply redundancy in a clever way to achieve robust and efficient data recovery against failures. The file $F = (F_1, F_2, \ldots, F_M) \in \mathbb{F}_q^M$ is encoded into a codeword $\mathbf{x}$ consisting of n blocks $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) \in (\mathbb{F}_q^\alpha)^n$. Given such a codeword $\mathbf{x}$, the part $\mathbf{x}_i$ is stored into node i. During operation, some of the nodes of the DSS may fail. If node i fails, a new node is added to the network. It contacts d live nodes and downloads β symbols from each. The total amount of downloaded data, $\gamma = \beta d$, is called the repair bandwidth. The new node processes these symbols to reconstruct $\mathbf{x}_i$. The repair process is conducted so that data stored at $k < n$ nodes allows F to be completely constructed (the "k out of n property"). A DSS satisfying such a property is often referred to as a (n, k)-DSS.

There is a tradeoff between the repair bandwidth γ and the amount of data that can be stored in each node [35]. Dimakis et al. suggested network coding [36, 37] for distributed data storage in order to reduce the bandwidth of node repair [35]. They introduced regenerating codes that achieve the optimal tradeoff between storage and repair bandwidth. This tradeoff can be achieved with linear network coding [20]. See Figure 1 for an example of a DSS and the repair process after node failure.

3.3. Mutual Information. Mutual information of two random variables X and Y is

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log_2 \left(\frac{p(x,y)}{p(x) \cdot p(y)} \right), \quad (1)$$

where $p(x, y)$ is the joint probability distribution function of X and Y, $p(x)$ is the marginal probability distribution function of X, and $p(y)$ is the marginal probability distribution function of Y. We say that X and Y are *dependent* if

$$I(X;Y) > 0. \quad (2)$$

Generalizing this to probability ensembles $X = \{X_s : s \in \mathbb{N}\}$ and $Y = \{Y_s : s \in \mathbb{N}\}$ we say that X and Y are *dependent* if

$$I(X_s; Y_s) > 0 \quad (3)$$

for every $s \in \mathbb{N}$.

3.4. McEliece Cryptosystem and Related Problems. The McEliece scheme $\mathsf{McEliece} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ applies binary Goppa codes [38] to enable asymmetric encryption. The key generation algorithm Gen outputs a private/public key pair

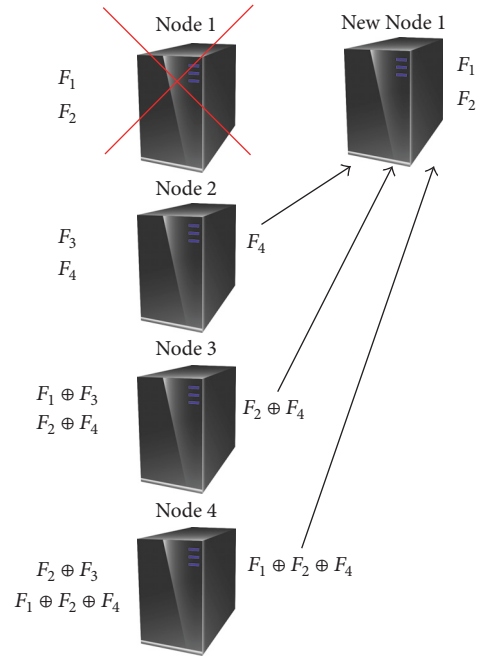


FIGURE 1: An example of a distributed storage system with linear coding. A file $F = (F_1, F_2, F_3, F_4)$ is distributed to $n = 4$ nodes each storing a vector of two parts of the file ($\alpha = 2$). The file is safe if one node fails. If Node 1 fails, it can be replaced by communicating only three blocks $(F_4, F_2 \oplus F_4, F_1 \oplus F_2 \oplus F_4)$ instead of all four.

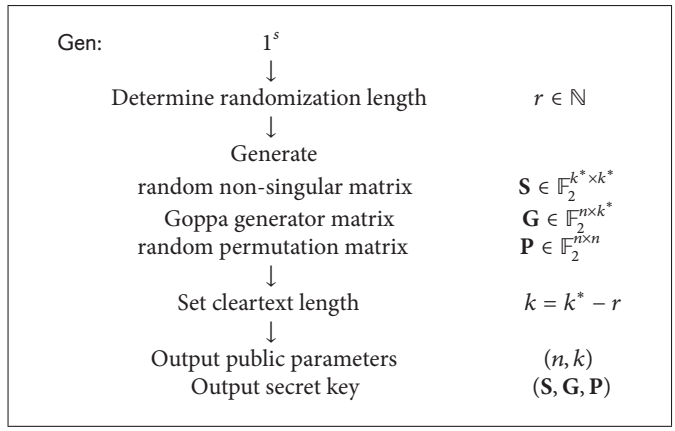
such that the private key consists of three matrices $(\mathbf{S}, \mathbf{G}, \mathbf{P})$ with entries in $\mathbb{F}_2$, where $\mathbf{P}$ is an $n \times n$ permutation matrix, $\mathbf{S}$ is a nonsingular $k \times k$ matrix, and $\mathbf{G}$ is the generator matrix for a binary Goppa code that is able to correct up to t errors. The public key is the $k \times n$ composition matrix $\mathbf{SGP}$. A message $\mathbf{m} \in \mathbb{F}_2^k$ is encrypted by Enc by computing $\mathbf{c} = \mathbf{mSGP} + \mathbf{e}$, where $\mathbf{e}$ is a randomly chosen error vector of Hamming weight t. For the decryption, Dec first computes $\mathbf{cP}^{-1} = \mathbf{mSG} + \mathbf{eP}^{-1}$ and then decodes the corresponding Goppa codeword to obtain $\mathbf{mS}$. Since $\mathbf{S}$ is nonsingular, the message $\mathbf{m}$ is computed by multiplying with $\mathbf{S}^{-1}$ from the right. A semantically secure version of the scheme can be found in [39]. Here, semantic security refers to indistinguishability of ciphertexts under chosen plaintext attack. For details on semantic security, see, for example, [40].

The security of $\mathsf{McEliece}$ is based on a certain assumption on the generator matrix $\mathbf{SGP}$. Let Mc_s denote the random variable determined by the probability distribution of sampling a generator matrix $\mathbf{SGP}$ according to $\mathsf{Gen}(1^s)$, where s is a security parameter. Let the probability ensemble $Mc = \{Mc_s : s \in \mathbb{N}\}$. Let McU denote the probability ensemble of random matrices with the same size as Mc. The following hardness assumption was first formulated in [41].

Assumption 1 (pseudorandomness of McEliece generator matrix). There exists a negligible function ϵ_M such that

$$\mathbf{Adv}^{D(Mc,McU)}(s) \leq \epsilon_M(s) \quad (4)$$

for every $s \geq 1$.



Algorithm 1: AddHomSE key generation.

In addition to this pseudorandomness assumption, McEliece relies on the hardness of the *learning parity with noise* problem. However, we do not need to apply it in our scheme.

4. Additively Homomorphic Symmetric Encryption Scheme

In this section, we give a construction of a symmetric encryption scheme that is homomorphic from the additive group $(\mathbb{F}_2^n, \oplus)$ to $(\mathbb{F}_2^k, \oplus)$, where $k, n \in \mathbb{N}$ and $k < n$. Due to linearity, it will be compatible with linear network coding. Our construction is inspired by the symmetric scheme suggested in [42], the homomorphic scheme suggested in [43] and the McEliece public key encryption scheme [12], and, especially, its IND-CPA variant [39]. Similarly to the McEliece scheme, our scheme is based on binary Goppa error correcting codes [38]. However, contrary to the McEliece scheme, we do not disclose the scrambled generator matrix. We also do not add any errors while encrypting which means that the full error correction capacity of the code can be utilized in applications. It would also be easy to adapt our proposal to apply other codes on an arbitrary finite field $\mathbb{F}_q$. However, binary fields and their extensions are useful for many applications since they enable efficient data combination due to efficiency of addition modulo 2 [11].

In general, the scheme operates as follows. Suppose that our file is divided into r parts constituting r plaintext messages $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_r$. Each of these messages are padded with a random suffix $\mathbf{z}$ and encrypted by encoding with a scrambled generator matrix $\widehat{\mathbf{G}}$ of a linear error correcting code: $\mathbf{c}_i = (\mathbf{m}_i, \mathbf{z})\widehat{\mathbf{G}}$. Note that the resulting ciphertexts can be linearly combined and the corresponding combination translates back to the plaintext space upon decoding due to linearity of the code. Furthermore, since the generator matrix is scrambled, an adversary is not able to determine the applied code and thus not able to decrypt the ciphertexts. In the following, we rigorously formulate this construction and the related computational assumptions. Based on computational indistinguishability, we then proceed to show its semantic security.

Definition 2 (AddHomSE). The symmetric encryption scheme

$$\mathsf{AddHomSE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}) \tag{5}$$

consists of a three-tuple of algorithms given in the following:

(1) $\mathsf{Gen}(1^s)$: based on the security parameter 1^s, Gen chooses a randomization length r, a linear $[n, k^*, d]$-error correcting Goppa code over $\mathbb{F}_2$ with a generator matrix $\mathbf{G}$ such that $k^* > r$. It also samples a random nonsingular $k^* \times k^*$ matrix $\mathbf{S}$ and a random $n \times n$ permutation matrix $\mathbf{P}$. It then sets the cleartext length to be k such that $k^* = k + r$, where $k \leq r - 1$ and sets n, k as public parameters and outputs $(\mathbf{S}, \mathbf{G}, \mathbf{P})$ as the secret key.

(2) $\mathsf{Enc}((\mathbf{S}, \mathbf{G}, \mathbf{P}), \mathbf{m})$: the input consists of a key $(\mathbf{S}, \mathbf{G}, \mathbf{P})$, a plaintext $\mathbf{m} \in \mathbb{F}_2^k$. It then samples a random

$$\mathbf{z} \longleftarrow U\left(\mathbb{F}_2^r\right) \tag{6}$$

and encodes the concatenation $(\mathbf{m}, \mathbf{z}) \in \mathbb{F}^{k^*}$ using $\mathbf{SGP}$ to obtain a ciphertext message

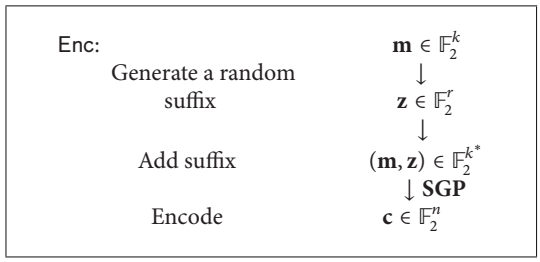
$$\mathbf{c} = (\mathbf{m}, \mathbf{z})\,\mathbf{SGP} \in \mathbb{F}^n. \tag{7}$$

(3) $\mathsf{Dec}((\mathbf{S}, \mathbf{G}, \mathbf{P}), \mathbf{c})$: the input consists of a key $(\mathbf{S}, \mathbf{G}, \mathbf{P})$ and a ciphertext $\mathbf{c} \in \mathbb{F}_2^n$. The plaintext message $\mathbf{m} \in \mathbb{F}_2^k$ is obtained by decoding $\mathbf{cP}^{-1}$ using the Goppa code, mapping the decoded message by $\mathbf{S}^{-1}$ and discarding the last r bits.

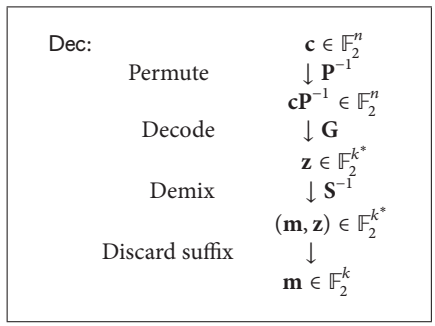
The key generation, encryption, and decryption processes are depicted in Algorithms 1, 2, and 3, respectively.

Note that contrary to the McEliece cryptosystem, the matrix $\mathbf{SGP}$ is not public. Instead, it is kept as a secret key. In addition, no error vectors are added in the encryption process.

We shall now proceed to show the IND-CPA security of our construction. Our plan is the following. We first show that AddHomSE can be divided into two parts, Enc^1 and Enc^2, such that the output of Enc is the sum of the outputs of these two algorithms. We then proceed to show that Enc_2 produces



Algorithm 2: AddHomSE encryption.



Algorithm 3: AddHomSE decryption.

a probability ensemble that is indistinguishable from random under a certain (reasonable) assumption. We then consider the sum of the outputs of these two algorithms and proceed to show that (under another reasonable assumption) the complete encryption algorithm produces ciphertexts that are indistinguishable from random.

We start by showing that Enc can be expressed as a sum of two algorithms. Let the scrambled generator matrix $\widehat{\mathbf{G}} = \mathbf{SGP}$ be partitioned into $k^* \times n$ and $r \times n$ submatrices $\widehat{\mathbf{G}_1}$ and $\widehat{\mathbf{G}_2}$ such that $(\mathbf{SGP})^T = (\widehat{\mathbf{G}_1}^T, \widehat{\mathbf{G}_2}^T)$, where T denotes transpose. Then we have

$$\mathsf{Enc}\left((\mathbf{S}, \mathbf{G}, \mathbf{P}), \mathbf{m}; r'\right) = \underbrace{\mathbf{m}\widehat{\mathbf{G}_1}}_{\mathsf{Enc}^1((\mathbf{S},\mathbf{G},\mathbf{P}),\mathbf{m})} \oplus \underbrace{\mathbf{z}\widehat{\mathbf{G}_2}}_{\mathsf{Enc}^2((\mathbf{S},\mathbf{G},\mathbf{P});r')}, \quad (8)$$

where Enc^1 is deterministic PT, Enc^2 is PPT, and r' is the internal randomness used by Enc.

Now, Enc^2 adds a different element $\mathbf{z}\widehat{\mathbf{G}_2} \in \mathbb{F}_2^n$ to the output of Enc^1 determined by the randomness r'. Suppose that we are encrypting q messages and that the output of Enc^2 is a truly random $U^{(h)} \sim U(\mathbb{F}_2^n)$ for every $h \leq q$. Then for every $h \leq q$ and every plaintext message $\mathbf{m}_h$ the output of Enc would be characterized by

$$\mathsf{Enc}^1\left((\mathbf{S}, \mathbf{G}, \mathbf{P}), \mathbf{m}_h\right) \oplus U^{(h)} \sim U\left(\mathbb{F}_2^n\right) \quad (9)$$

and AddHomSE would satisfy perfect secrecy for q encryptions. In reality, the output of Enc^2 is not truly random. However, in the following we show that it is indistinguishable from random under a certain assumption. Then we consider the connection between Enc^1 and Enc^2 and, finally, the indistinguishability of encryptions from random. For easier reference, variables used in the description of the scheme, as well as in the following proofs, have been collected into Table 1. Similarly, the used random variables have been collected into Table 2.

Table 1: The variables used in AddHomSE and in its proof of security and their descriptions.

Variable	Description
1^s	The security parameter
r, r_s	A randomization length; determines the maximum number of file parts
$\mathbf{m}, \mathbf{m}_h$	A plaintext message
$\mathbf{c}, \mathbf{c}_h$	A ciphertext message
$\mathbf{z}, \mathbf{z}_h$	A random binary suffix of length r
n, n_s	The length of the used Goppa code
k^*, k_s^*	The dimension of the used Goppa code
d	The distance of the used Goppa code
$\mathbf{G}$	The generator matrix of the Goppa code
$\mathbf{S}$	A random non-singular binary $k^* \times k^*$ matrix
$\mathbf{P}$	A random $n \times n$ permutation matrix
k, k_s	The cleartext length such that $k^* = k + r$ and $k \leq r - 1$
$\widehat{\mathbf{G}}$	The scrambled generator matrix $\widehat{\mathbf{G}} = \mathbf{SGP}$
$\widehat{\mathbf{G}_1}$	$k^* \times n$ submatrix of $\widehat{\mathbf{G}}$; $(\widehat{\mathbf{G}})^T = (\widehat{\mathbf{G}_1}^T, \widehat{\mathbf{G}_2}^T)$
$\widehat{\mathbf{G}_2}$	$r \times n$ submatrix of $\widehat{\mathbf{G}}$; $(\widehat{\mathbf{G}})^T = (\widehat{\mathbf{G}_1}^T, \widehat{\mathbf{G}_2}^T)$
r'	The internal randomness used by Enc
q	The total number of encrypted messages
$\mathbf{Z}$	A uniformly random $q \times r_s$ matrix
$\mathbf{G}'$	A uniformly random $r_s \times n_s$ matrix
$\mathbf{M}$	A message matrix $\mathbf{M} = (\mathbf{m}_0^T, \mathbf{m}_1^T, \ldots, \mathbf{m}_{q-1}^T)^T$

Table 2: The used random variables and their descriptions.

Random variable	Description
$U^{(h)}$	Distributed uniformly on $\mathbb{F}_2^n$ for every $h \leq q$
$E_s^{2,(h)}$	Induced by $\mathsf{Enc}^2(\mathbf{S}, \mathbf{G}, \mathbf{P}) = \mathbf{z}_h\widehat{\mathbf{G}_2}$ for $h \leq q$
$E_s^{2,q}$	A q-tuple of random variables $(E_s^{2,(0)}, E_s^{2,(1)}, \ldots, E_s^{2,(q-1)})$
Mc_s	Corresponds to the choice of $\widehat{\mathbf{G}}$
Mc_s^1	Corresponds to the choice of $\widehat{\mathbf{G}_1}$
Mc_s^2	Corresponds to the choice of $\widehat{\mathbf{G}_2}$
McU_s^2	Distributed uniformly on $\mathbb{F}_2^{r_s \times n_s}$
V_s^q	Determined by $\mathbf{ZG}'$
U_s^q	Distributed uniformly on $\mathbb{F}_2^{r_s \times n_s}$
K_s	Determined by key generation; $K_s = \mathsf{Gen}(1^s)$
$E_s^{(h)}$	Determined by encryption of $\mathbf{m}_h$; $E_s^{(h)} = \mathsf{Enc}(K_s, \mathbf{m}_h)$
E_s^q	Determined by encryption of q messages; $E_s^q = (E_s^{(0)}, E_s^{(1)}, \ldots, E_s^{(q-1)})$

5. The Probability Ensemble Induced by Enc^2

In the following, we consider the probability ensemble $E^{2,q} = \{E_s^{2,q}\}_{s\in\mathbb{N}}$ induced by Enc^2 for q encryptions. That is, we have a q-tuple

$$E_s^{2,q} = \left(E_s^{2,(0)}, E_s^{2,(1)}, \ldots, E_s^{2,(q-1)}\right) \tag{10}$$

such that $E_s^{2,(h)} = \mathsf{Enc}^2(\mathbf{S}, \mathbf{G}, \mathbf{P}) = \mathbf{z}_h\widehat{\mathbf{G}_2}$ for every $h \in \{0, 1, \ldots, q-1\}$, where $(\mathbf{S}, \mathbf{G}, \mathbf{P}) \leftarrow \mathsf{Gen}(1^s)$ and $\mathbf{z}_h \leftarrow U(\mathbb{F}_2^r)$. Note that n, k^*, k, and r depend on the security parameter s. In the following, we have made the dependence explicit. We can consider $E^{2,q}$ as a random variable over $\mathbb{F}_2^{q\times n_s}$ by setting $E^{2,q} = \mathbf{Z}\widehat{\mathbf{G}_2}$, where $\mathbf{Z}$ is a $q \times r_s$ matrix chosen uniformly at random. For convenience, we assume that $E^{2,q}$ is written in such a matrix form.

5.1. Indistinguishability of $\widehat{\mathbf{G}_2}$ from Random. Our plan is to show the indistinguishability of $E^{2,q}$ from random for all $q \leq r_s$. In order to do that we want $\widehat{\mathbf{G}_2}$ to be also indistinguishable from random. We could apply the McEliece assumption (Assumption 1) that states that the complete generator matrix **SGP** satisfies this property. However, such an assumption is too strong in our case. We derive a weaker assumption that relates only to $\widehat{\mathbf{G}_2}$.

Definition 3. Let $Mc = \{Mc_s\}_{s\in\mathbb{N}}$ denote a probability ensemble of McEliece generator matrices (chosen according to some schema) such that Mc_s is distributed over matrices of size $k_s^* \times n_s^*$ for every $s \in \mathbb{N}$. Let $Mc^1 = \{Mc_s^1\}_{s\in\mathbb{N}}$ and $Mc^2 = \{Mc_s^2\}_{s\in\mathbb{N}}$ denote the probability ensembles such that $Mc_s^T = ((Mc_s^1)^T, (Mc_s^2)^T)$ for every $s \in \mathbb{N}$, where Mc^1 is distributed over matrices of size $k_s \times n_s$ and Mc^2 is distributed over matrices of size $r_s \times n_s$, where $k_s + r_s = k_s^*$ and k_s and r_s are chosen according to $\mathsf{Gen}(1^s)$.

Assumption 4 ($\widehat{\mathbf{G}_2}$ indistinguishable from random). Let $McU^2 = \{McU_s^2\}_{s\in\mathbb{N}}$ denote the uniform probability ensemble such that $McU_s^2 \sim U(\mathbb{F}_2^{r_s\times n_s})$ for every $s \in \mathbb{N}$. For every PPT algorithm A, there is a negligible function ϵ such that

$$\mathbf{Adv}_{\mathsf{A}}^{D(Mc^2, McU^2)}(s) \leq \epsilon(s) \tag{11}$$

for every $s \geq 1$.

If the generator matrix satisfies the formulated assumption, then $\widehat{\mathbf{G}_2}$ cannot be distinguished from random. Suppose that $\widehat{\mathbf{G}_2}$ is exchanged with truly random matrix. Let $V^q = \{V_s^q\}_{s\in\mathbb{N}}$ be a probability ensemble such that $V_s^q = \mathbf{Z}\mathbf{G}'$, where $\mathbf{Z} \leftarrow U(\mathbb{F}_2^{q\times r_s})$ and $\mathbf{G}' \leftarrow U(\mathbb{F}_2^{r_s\times n_s})$. Let $U^q = \{U_s^q\}_{s\in\mathbb{N}}$ denote the uniform probability ensemble such that $U_s^q \sim U(\mathbb{F}_2^{q\times n_s})$, where n_s is determined by $\mathsf{Gen}(1^s)$. Clearly, the statistical distance

$$\Delta\left(V_s^q, U_s^q\right) = 0 \tag{12}$$

for every $s \in \mathbb{N}$ and $q \leq r_s$ since all of the elements of $\mathbf{Z}\mathbf{G}'$ are uniformly random.

(1) **procedure** $\mathsf{B}(1^s, \mathbf{X})$ ▷ $\mathbf{X}$ is a $r_s \times n_s$ matrix
(2) $\quad\mathbf{Z} \leftarrow U(\mathbb{F}_2^{q\times r_s})$
(3) $\quad b \leftarrow \mathsf{A}(1^s, \mathbf{ZX})$
(4) $\quad$**return** b
(5) **end procedure**

Algorithm 4

We shall now provide a connection between Assumption 4 and the indistinguishability of $E^{2,q}$ from V^q for $q \leq r_s$.

Proposition 5. *For every PPT algorithm* A *there is a PPT algorithm* B *such that*

$$\mathbf{Adv}_{\mathsf{B}}^{D(Mc^2, McU^2)}(s) \geq \mathbf{Adv}_{\mathsf{A}}^{D(E^{2,q}, V^q)}(s) \tag{13}$$

for every $q \leq r_s$ *and* $s \in \mathbb{N}$.

Proof. The reduction is straightforward. Let $s \in \mathbb{N}$ be given and let A be a PPT algorithm considered as a distinguisher for $E^{2,q}$ and V^q. Let us define the distinguisher B for Mc^2 and McU^2 that is shown in Algorithm 4.

If $\mathbf{X} \leftarrow Mc_s^2$, then B is invoked with r_s rows of a McEliece generator matrix. By the description of B, A is queried with a matrix sampled according to $E_s^{2,q}$. Let now $\mathbf{X} \leftarrow McU_s^2$. Then A is invoked with an element sampled according to V_s^q and since B outputs the same bit as A, we have

$$\mathbf{Adv}_{\mathsf{B}}^{D(Mc^2, McU^2)}(s) = \mathbf{Adv}_{\mathsf{A}}^{D(E^{2,q}, V^q)}(s). \tag{14}$$

□

A direct consequence of Proposition 5 is the result we aimed for: indistinguishability of $E^{2,q}$ from random under Assumption 4.

Proposition 6. *For every PPT algorithm* A *and* $q \leq r$,

$$\begin{aligned}\mathbf{Adv}_{\mathsf{A}}^{D(E^{2,q}, U^q)}(s) &= \mathbf{Adv}_{\mathsf{A}}^{D(E^{2,q}, V^q)}(s)\\ &\leq \mathbf{Adv}^{D(Mc^2, McU^2)}(s)\end{aligned} \tag{15}$$

for every $s \in \mathbb{N}$.

6. Semantic Security for r Messages

Let us now turn to the probability ensemble induced by the complete encryption algorithm Enc. We establish the semantic security of $\mathsf{AddHomSE}$ by proving that it satisfies ciphertext indistinguishability for up to r_s messages under two assumptions: Assumption 4 and a new one regarding independence of $\widehat{\mathbf{G}_1}$ and $\widehat{\mathbf{G}_2}$. Let $\mathbf{m}_0, \mathbf{m}_1, \ldots, \mathbf{m}_{q-1} \in \mathbb{F}_2^k$ be any plaintext messages. Let $E^q = \{E_s^q\}_{s\in\mathbb{N}}$ such that $E_s^q = (E_s^{(0)}, E_s^{(1)}, \ldots, E_s^{(q-1)})$, where $E_s^{(h)} = \mathsf{Enc}(K_s, \mathbf{m}_h)$ and $K_s = \mathsf{Gen}(1^s)$. As before, let us consider E_s^q in the matrix form. Set also $\mathbf{M} = (\mathbf{m}_0^T, \mathbf{m}_1^T, \ldots, \mathbf{m}_{q-1}^T)^T$. That is, the rows of $\mathbf{M}$ consist of the plaintext messages. We call $\mathbf{M}$ the *message matrix* of $\mathbf{m}_0, \mathbf{m}_1, \ldots, \mathbf{m}_{q-1} \in \mathbb{F}_2^k$.

```
(1) procedure DepExp(A, A, B, X)(s)        ▷ Dependability experiment
(2)   b ← U({0, 1})
(3)   if b = 1 then
(4)     b′ ← A(1^s, A_s, B_s)
(5)   else
(6)     b′ ← A(1^s, A_s, X_s)
(7)   end if
(8)   if b = b′ then
(9)     output 1
(10)  else
(11)    output 0
(12)  end if
(13) end procedure
```

Algorithm 5

6.1. Computational Independence. Assumption 4 concerns the last part $\widehat{\mathbf{G}_2}$ of the generator matrix $\mathbf{G}$. However, we need to also make an assumption regarding $\widehat{\mathbf{G}_1}$. For example, suppose that it was possible that $\widehat{\mathbf{G}_1} = \widehat{\mathbf{G}_2}$. Then E^q would be easily distinguishable with high probability by choosing $\mathbf{M} = \mathbf{I}$, the identity matrix. To foil such attempts, we want $\widehat{\mathbf{G}_1}$ and $\widehat{\mathbf{G}_2}$ to be sufficiently independent of each other. We shall formulate an assumption concerning the mutual information of Mc_s^1 and Mc_s^2.

Let us define the following experiment in which we attempt to determine whether two probability ensembles are dependent. Suppose that we have three probability ensembles A, B, and X. Suppose also that B is indistinguishable from X. Furthermore, suppose that $I(A_s; B_s) > 0$ while $I(A_s; X_s) = 0$ for every $s \in \mathbb{N}$. We define the experiment that is shown in Algorithm 5.

In the experiment, A is either given an element from B_s such that $I(A_s; B_s) > 0$ or an element from X_s that is indistinguishable from B_s such that $I(A_s; X_s) = 0$. Since B and X are indistinguishable, A succeeds in this experiment with nonnegligible probability only if it is able to find the dependability of B_s from A_s.

Definition 7. Let $A = \{A_s : s \in \mathbb{N}\}, B = \{B_s : s \in \mathbb{N}\}$ be probability ensembles. We say that A and B are *computationally independent* if for every PPT algorithm A and every probability ensemble $X = \{X_s : s \in \mathbb{N}\}$ such that X is computationally indistinguishable from B and $I(B_s; X_s) = 0$ for every $s \in \mathbb{N}$ there is a negligible function ϵ such that

$$\begin{aligned}&\mathbf{Adv}_{\mathsf{A}}^{\mathsf{Dep}(A,B,X)}(s)\\ &\quad = \left|2 \cdot \Pr\left[\mathsf{DepExp}(\mathsf{A}, A, B, X) = 1\right] - 1\right| \leq \epsilon(s)\end{aligned} \tag{16}$$

for every $s \in \mathbb{N}$. If this does not hold, then we say that A and B are *noticeably dependent*.

Note that it follows from the definition of $\mathsf{Dep}(A, B, X)$ that

$$\mathbf{Adv}^{\mathsf{Dep}(A,B,X)}(s) \geq \mathbf{Adv}^{D(B,X)}(s) \tag{17}$$

for every $s \in \mathbb{N}$. We formulate the following assumption concerning the relationship between $\widehat{\mathbf{G}_1}$ and $\widehat{\mathbf{G}_2}$.

Assumption 8 ($\widehat{\mathbf{G}_1}$ and $\widehat{\mathbf{G}_2}$ computationally independent). For every probability ensemble X indistinguishable and independent from Mc^2 and every PPT algorithm A there is a negligible function ϵ such that

$$\mathbf{Adv}_{\mathsf{A}}^{\mathsf{Dep}(Mc^1, Mc^2, X)}(s) \leq \epsilon(s) \tag{18}$$

for every $s \geq 1$.

The assumption states that it is not feasible to find any information that links $\widehat{\mathbf{G}_1}$ and $\widehat{\mathbf{G}_2}$. The assumption is still weaker than the McEliece assumption that states that the whole $\widehat{\mathbf{G}} = \mathbf{SGP}$ is indistinguishable from random. (If they are, then necessarily $\widehat{\mathbf{G}_1}$ and $\widehat{\mathbf{G}_2}$ are computationally independent.) However, Assumption 8 does not require $\widehat{\mathbf{G}_1}$ to be indistinguishable from random. In fact, our proofs do not depend at all on the structure of $\widehat{\mathbf{G}_1}$ as long as $\widehat{\mathbf{G}_1}$ and $\widehat{\mathbf{G}_2}$ are computationally independent. To make the scheme faster, we could, for instance, omit $\mathbf{S}$ and $\mathbf{P}$ from affecting the first k rows of the generator matrix $\mathbf{G}$.

We are now ready to show the semantic security of AddHomSE by showing the indistinguishability of E^q from random.

Proposition 9. AddHomSE *has indistinguishable encryptions for r_s messages under Assumptions 4 and 8.*

Proof. Suppose that Assumption 4 holds. We establish the claim by showing that for every set of $q \leq r_s$ plaintext messages $\mathbf{m}_0, \mathbf{m}_1, \ldots, \mathbf{m}_{q-1} \in \mathbb{F}_2^{k_s}$ and every PPT algorithm A there is a PPT algorithm B such that

$$\mathbf{Adv}_{\mathsf{A}}^{D(E^q, U^q)}(s) \leq \mathbf{Adv}_{\mathsf{B}}^{\mathsf{Dep}(Mc^1, Mc^2, McU^2)}(s) \tag{19}$$

for $s \in \mathbb{N}$, where E^q is induced by $\mathbf{m}_0, \mathbf{m}_1, \ldots, \mathbf{m}_{q-1}$. Then, under Assumption 8, the advantage of A is negligible.

Since McU^2 is truly random, we have $I(Mc^2; McU^2) = 0$. In addition, by Assumption 4, Mc^2 is computationally

(1) **procedure** $\mathsf{B}(1^s, \widehat{\mathbf{G}_1}, \mathbf{X})$ ▷ $\mathbf{X}$ is either $\widehat{\mathbf{G}_2}$ or a random matrix
(2) $\quad \mathbf{Z} \leftarrow U(\mathbb{F}_2^{q \times r_s})$
(3) $\quad \mathbf{Y} \leftarrow \mathbf{M}\widehat{\mathbf{G}_1} \oplus \mathbf{ZX}$
(4) $\quad b \leftarrow \mathsf{A}(1^s, \mathbf{Y})$
(5) $\quad$ **output** b
(6) **end procedure**

Algorithm 6

indistinguishable from McU^2 and therefore $\mathsf{Dep}(Mc^1, Mc^2, McU^2)$ is well defined. Let the security parameter s be fixed and let $\mathbf{m}_0, \mathbf{m}_1, \ldots, \mathbf{m}_{q-1} \in \mathbb{F}_2^k$ be any messages. Let $\mathbf{M}$ be the message matrix of $\mathbf{m}_0, \mathbf{m}_1, \ldots, \mathbf{m}_{q-1}$. Written in the matrix form, we have $E_s^q = \mathbf{M}Mc_s^1 \oplus E_s^{2,q}$ and the elements are of the form

$$\mathbf{M}\widehat{\mathbf{G}_1} \oplus \mathbf{Z}\widehat{\mathbf{G}_2}, \tag{20}$$

where $\mathbf{G} = (\widehat{\mathbf{G}_1}^T, \widehat{\mathbf{G}_2}^T)^T \leftarrow Mc_s$ and $\mathbf{Z} \leftarrow U(\mathbb{F}_2^{q \times r_s})$.

Let A be any PPT algorithm considered as a distinguisher for $D(E^q, U^q)$. Using A, we construct an algorithm B that determines the dependability of Mc^1 and Mc^2 (see Algorithm 6).

Suppose that the input $\mathbf{X}$ is random matrix. Then

$$\mathbf{Y} = \mathbf{M}\widehat{\mathbf{G}_1} \oplus \underbrace{\mathbf{ZX}}_{\leftarrow U(\mathbb{F}_2^{q \times n_s})} \tag{21}$$

is a truly random matrix. Therefore, A was invoked with a matrix sampled according to U_s^q. Suppose now that $\mathbf{X} = \widehat{\mathbf{G}_2}$. Then

$$\mathbf{Y} = \mathbf{M}\widehat{\mathbf{G}_1} \oplus \underbrace{\mathbf{Z}\widehat{\mathbf{G}_2}}_{\leftarrow E_s^{2,q}} \tag{22}$$

and $\mathbf{Y}$ was sampled according to E_s^q. Since B outputs the same bit as A, we have

$$\mathbf{Adv}_{\mathsf{B}}^{\mathsf{Dep}(Mc^1, Mc^2, McU^2)}(s) \geq \mathbf{Adv}_{\mathsf{A}}^{D(E^q, U^q)}(s). \tag{23}$$

□

AdHomSE is IND-CPA secure under Assumptions 4 and 8 whenever the adversary is restricted to at most r_s queries to the encryption oracle (the test query included). Considering a DSS, whenever the dataset is divided into at most r_s parts, each of those parts remains secret even under a chosen ciphertext attack where the adversary is able to choose each of those parts separately and adaptively.

7. Infeasibility, Key Size, and Error Correction Capacity

7.1. Infeasibility of the Problems. Let us briefly consider the infeasibility of the underlying problems related to AddHomSE. The IND-CPA security is based on assumptions that are weaker but closely related to the ones underlying the McEliece scheme. The selection of parameters for the McEliece scheme has been considered in [44] and the best performing attacks are based on information set decoding. In addition, due to algebraic attacks against Goppa codes [45, 46] the rate $R^* = k^*/n$ cannot be close to one and the degree t of the Goppa polynomial has to satisfy $t \geq z_{\min}$, where $z_{\min}$ is the smallest integer satisfying $\alpha z(\alpha z - (2\beta+1)z + 2^\beta)/2 \geq n - \alpha z$, where $\alpha = \lceil \log_2 n \rceil$ and $\beta = \lceil \log_2 z \rceil + 1$ [47]. Choosing $R^* \approx 0.8$ maximizes the complexity of information set decoding attacks [44].

For AddHomSE, the attacker is not given the generator matrix. Instead, the attacker gets at most r scrambled messages under an adaptive chosen plaintext attack. Therefore, n can be drastically lower for AddHomSE. We suggest $k = \lfloor k^*/2 \rfloor - 1$ and $r = n^* - k$ so that randomization length is slightly more than half of the input. The rate R^* should be kept close to a constant. We suggest choosing a rate R^* that is close to 0.8 due to information set decoding attacks [44].

7.2. Key Size. The key size of AddHomSE is big if truly random matrices are used. In a practical setting, we want to use pseudorandom matrices for $\mathbf{S}$ and $\mathbf{P}$. The key size is dramatically decreased by exchanging these matrices with a short seed c and generating $\mathbf{S}$ and $\mathbf{P}$ using a pseudorandom generator G. The generating matrix $\mathbf{G}$ of the Goppa code can be derived from the Goppa polynomial $g(x)$ and pseudorandom elements generated by G. Therefore, in practice, the key can be compactly presented by the seed c and the polynomial $g(x)$.

Typically, in a distributed storage systems we want to encrypt files or file systems that are huge. If a large file is divided into few parts, we do not want to consider each part as a single plaintext message since such an approach would require k^* and n to be at least as large as the length of the file part. In such a case, we can further divide the part into smaller blocks and encrypt those block independently. Such an approach enables us to select small and efficient values for k^* and n. Note that such a division does not affect the homomorphic property of the scheme provided that each of the file parts are processed similarly and encrypted with the same keys. It also does not have an effect on the key size since the keys of those individual blocks can be derived from the same seed c and the polynomial $g(x)$.

7.3. Error Correction. Due to requirements of semantic security and error correction, ciphertexts contain overhead compared to plaintext messages. For example, with $(n, k^*, r) = (256, 200, 100)$, where the rate $R^* \approx 0.78$, plaintexts of length 100 will be encrypted into ciphertexts of length 256. The scheme can correct up to t errors, where t is the degree of

the Goppa generator polynomial. With these parameters, we should choose $t \geq z_{\min} = 5$ [47]. Choosing the smallest t, which results in the most efficient implementation, enables us to correct up to 5 errors in each 256 bits meaning that the plaintext messages are correctly decrypted with high probability whenever the error rate is less than 2%. If more error correction capacity is needed, then a higher degree Goppa generator polynomial needs to be selected and/or the rate R^* should be lowered. As a final remark, we note that the binary Goppa code can be exchanged with another linear code on a finite field $\mathbb{F}_q$. However, we have only shown the security of AddHomSE based on the indistinguishability of a scrambled Goppa generator matrices. The applied linear code has to satisfy a similar infeasibility result.

8. Conclusion

We propose an additively homomorphic symmetric encryption scheme AddHomSE that is compatible with linear network coding: a linear combination of ciphertext messages decrypts to the same linear combination of corresponding plaintext messages. The scheme can be used for the encryption of data stored in a distributed storage system (DSS), for example, in the distributed Internet of Things. We show that the scheme is semantically secure (IND-CPA) and provides computational indistinguishability for each individual part of the file stored in the DSS. In combination with an additively homomorphic MAC our scheme supports the authenticate- then-encrypt paradigm that ensures plaintext integrity. Finally, based on Goppa codes, our scheme offers simultaneous error correction. Our proofs are shown for the binary field $\mathbb{F}_2$ which is commonly used for the implementation of a DSS due to computational efficiency reasons. We also discuss the selection of secure parameters for the scheme and explain how it can be applied with compact keys.

Disclosure

Work related to this manuscript has first appeared in the author's doctoral thesis [48].

Conflicts of Interest

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

References

[1] M. Hilbert and P. López, "The world's technological capacity to store, communicate, and compute information," *Science*, vol. 332, no. 6025, pp. 60–65, 2011.

[2] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, 2013.

[3] N. Cai and R. W. Yeung, "Secure network coding," in *Proceedings of the IEEE International Symposium on Information Theory*, 323 pages, 2002.

[4] N. Cai and R. W. Yeung, "Secure network coding on a wiretap network," *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, vol. 57, no. 1, pp. 424–435, 2011.

[5] N. B. Shah, K. V. Rashmi, and P. V. Kumar, "Information-theoretically secure regenerating codes for distributed storage," in *Proceedings of the 54th Annual IEEE Global Telecommunications Conference: "Energizing Global Communications", GLOBECOM 2011*, USA, 2011.

[6] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 60, no. 1, pp. 212–236, 2014.

[7] S. Goparaju, S. E. Rouayheb, R. Calderbank, and H. V. Poor, "Data secrecy in distributed storage systems under exact repair," in *Proceedings of the 2013 International Symposium on Network Coding, NetCod 2013*, can, June 2013.

[8] S. Pawar, S. El Rouayheb, and K. Ramchandran, "On secure distributed data storage under repair dynamics," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '10)*, pp. 2543–2547, IEEE, Austin, Tex, USA, June 2010.

[9] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270–299, 1984.

[10] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in cryptology—EUROCRYPT '99*, vol. 1592, pp. 223–238, Springer, Berlin, 1999.

[11] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos et al., "XORing elephants: Novel erasure codes for big data," in *Proceedings of the 39th international conference on Very Large Data Bases (PVLDB '13)*, pp. 325–336, VLDB Endowment, 2013.

[12] R. J. McEliece, "A Public-Key Cryptosystem Based On Algebraic Coding Theory. Deep Space Network Progress Report," in *R. J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. Deep Space Network Progress Report*, pp. 44–114, 44, 114–116, January 1978.

[13] T. Cui, T. Ho, and J. Kliewer, "On secure network coding with unequal link capacities and restricted wiretapping sets," in *Proceedings of the IEEE Information Theory Workshop (ITW '10)*, 2010.

[14] J. Feldman, T. Malkin, A. Rocco, and C. Stein, "On the capacity of secure network coding," in *In Proceedings of the 42nd Annual Allerton Conference on Communication, Control, and Computing*, Cambridge University Press, 2004.

[15] S. Y. El Rouayheb and E. Soljanin, "On Wiretap networks II," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '07)*, pp. 551–555, 2007.

[16] D. Silva and F. R. Kschischang, "Security for wiretap networks via rank-metric codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '08)*, pp. 176–180, 2008.

[17] O. O. Koyluoglu, A. S. Rawat, and S. Vishwanath, "Secure cooperative regenerating codes for distributed storage systems," *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, vol. 60, no. 9, pp. 5228–5244, 2014.

[18] O. Kosut, L. Tong, and D. Tse, "Nonlinear network coding is necessary to combat general byzantine attacks," in *Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton '09)*, pp. 593–599, 2009.

[19] T. K. Dikaliotis, A. G. Dimakis, and T. Ho, "Security in distributed storage systems by communicating a logarithmic number of bits," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 10)*, pp. 1948–1952, 2010.

[20] S. Pawar, S. El Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems against eavesdropping and adversarial attacks," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6734–6753, 2011.

[21] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.

[22] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the Association for Computing Machinery*, vol. 21, no. 2, pp. 120–126, 1978.

[23] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *Journal of the ACM*, vol. 60, no. 6, pp. 1–35, 2013.

[24] J. D. Cohen and M. J. Fischer, "A robust and verifiable cryptographically secure election scheme," in *Proceedings of the IEEE 26th Annual Symposium on Foundations of Computer Science*, pp. 372–382, 1985.

[25] D. Naccache and J. Stern, "A new public key cryptosystem based on higher residues," in *Proceedings of the 5th ACM Conference on Computer and Communications Security (CCS-5 98)*, pp. 59–66, ACM, New York, NY, USA, 1998.

[26] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," in *Advances in Cryptology—EUROCRYPT '98*, K. Nyberg, Ed., vol. 1403, pp. 308–318, Springer, Berlin, Germany, 1998.

[27] I. Damgård and M. Jurik, "A Generalisation, a Simpli.cation and Some Applications of Paillier's Probabilistic Public-Key System," in *Public Key Cryptography*, vol. 1992 of *Lecture Notes in Computer Science*, pp. 119–136, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[28] E. Bresson, D. Catalano, and D. Pointcheval, "A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications," in *Advances in Cryptology - ASIACRYPT 2003*, C.-S. Laih, Ed., vol. 2894 of *Lecture Notes in Comput. Sci.*, pp. 37–54, Springer, Berlin, Germany, 2003.

[29] M. Joye and B. t. Libert, "Efficient cryptosystems from 2^k-th power residue symbols," in *Advances in Cryptology—EUROCRYPT 2013*, J. Thomas and P. Q. Nguyen, Eds., vol. 7881, pp. 76–92, Springer, Berlin, Germany, 2013.

[30] J. Domingo-Ferrer, "A provably secure additive and multiplicative privacy homomorphism," in *Information Security: 5th International Conference, (ISC 2002)*, C. Agnes Hui and G. Virgil, Eds., vol. 2433, pp. 471–483, Springer, Berlin, Germany, 2002.

[31] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '05)*, pp. 109–117, IEEE Computer Society, July 2005.

[32] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *Advances in Cryptology—CRYPTO 2011*, R. Phillip, Ed., vol. 6841, pp. 505–524, Springer, Berlin, Germany, 2011.

[33] P. Burtyka and O. Makarevich, "Symmetric fully homomorphic encryption using decidable matrix equations," in *Proceedings of the 7th International Conference on Security of Information and Networks, (SIN '14)*, pp. 186–196, ACM, New York, NY, USA, 2014.

[34] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM Journal on Computing*, vol. 17, no. 2, pp. 281–308, 1988.

[35] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.

[36] R. W. Yeung and Z. Zhang, "Distributed source coding for satellite communications," *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, vol. 45, no. 4, pp. 1111–1120, 1999.

[37] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[38] V. D. Goppa, "A new class of linear correcting codes," *Problemy Peredachi Informatsii*, vol. 6, no. 3, pp. 24–30, 1970.

[39] R. Nojima, H. Imai, K. Kobara, and K. Morozov, "Semantic security for the McEliece cryptosystem without random oracles," *Designs, Codes and Cryptography. An International Journal*, vol. 49, no. 1-3, pp. 289–305, 2008.

[40] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, Chapman & Hall/CRC, 2007.

[41] N. T. Courtois, M. Finiasz, and N. Sendrier, "How to achieve a McEliece-based digital signature scheme," in *Advances in Cryptology—ASIACRYPT 2001*, C. Boyd, Ed., vol. 2248 of *Lecture Notes in Computer Science*, pp. 157–174, Springer, Berlin, Germany, 2001.

[42] A. Kiayias and M. Yung, "Cryptographic hardness based on the decoding of Reed-Solomon codes," in *Proceedings of the 29th International Colloquium on Automata, Languages and Programming, (ICALP '02)*, pp. 232–243, Springer, London, UK, 2002.

[43] F. Armknecht, D. Augot, L. Perret, and A.-R. Sadeghi, "On constructing homomorphic encryption schemes from coding theory," in *Cryptography and Coding*, C. Liqun, Ed., vol. 7089 of *Lecture Notes in Computer Science*, pp. 23–40, Springer, Berlin, Germany, 2011.

[44] R. Niebuhr, M. Meziani, S. Bulygin, and J. Buchmann, "Selecting parameters for secure McEliece-based cryptosystems," *International Journal of Information Security*, vol. 11, no. 3, pp. 137–147, 2012.

[45] J.-C. Faugère, A. Otmani, L. Perret, and J.-P. Tillich, "Algebraic cryptanalysis of McEliece variants with compact keys," in *Advances in Cryptology—EUROCRYPT 2010*, H. Gilbert, Ed., vol. 6110 of *Lecture Notes in Computer Science*, pp. 279–298, Springer, Berlin, Germany, 2010.

[46] J.-C. Faugère, A. Otmani, L. Perret, and J.-P. Tillich, "Algebraic cryptanalysis of compact McEliece's variants—toward a complexity analysis," in *2nd International Conference on Symbolic Computation and Cryptography*, pp. 45–55, Springer, Berlin, 2010.

[47] J.-C. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret, and J.-P. Tillich, "A distinguisher for high rate McEliece cryptosystems," in *EEE Information Theory Workshop (ITW '11)*, pp. 282–286, 2011.

[48] J. Partala, *Algebraic methods for cryptographic key exhange [Ph. D. thesis]*, University of Oulu, 2015.