Semantic Security and Key-Privacy With Random Split of St-Gen Codes

Danilo Gligoroski[1] and Simona Samardjiska[2]

[1]Department of Telematics, The Norwegian University of Science and Technology (NTNU), Trondheim, Norway,
[2]Faculty of Computer Science and Engineering, UKIM, Skopje, Macedonia
danilog@item.ntnu.no, simona.samardjiska@finki.ukim.mk

Abstract. Recently we have defined Staircase-Generator codes (St-Gen codes) and their variant with a random split of the generator matrix of the codes. One unique property of these codes is that they work with arbitrary error sets. In this paper we give a brief overview of St-Gen codes and the list decoding algorithm for their decoding. We also analyze the semantic security against chosen plaintext attack (IND-CPA) and key-privacy i.e. indistinguishability of public keys under chosen plaintext attack (IK-CPA) of the encryption scheme with random split of St-Gen codes. In a similar manner as it was done by Nojima et al., and later by Yamakawa et al., we show that padding the plaintext with a random bit-string provides IND-CPA and IK-CPA in the standard model. The difference with McEliece scheme is that with our scheme the length of the padded random string is significantly shorter.

Keywords: Public Key Cryptography, Code Based Cryptosystems, Semantic Security, Key-Privacy

1 Introduction

The idea about semantic security against chosen-plaintext attack (i.e. *indistinguishability against chosen-plaintext attack (IND-CPA)*) for a public-key cryptosystem (PKC) was initially presented by Goldwasser and Micali in [9]. By replacing the deterministic encryption with probabilistic one, they showed the existence of public key schemes where the ciphertext does not leak any useful information about the plaintext (except its length). Later, in the work of Belare et al., [3] the semantic security against chosen-plaintext attack was systematized in a broader security perspective in relation with other security notions in public-key encryption schemes. Then, in 2001 we got a definition for a yet another security notion: *key-privacy* or *anonymity* in public-key schemes. It was introduced by Bellare et al., in [2]. In a nutshell it asks that an adversary receiving a ciphertext is not able to determine which specific public-key, out of a set of known public keys was used to produce that ciphertext. Under the assumption that the Decision Diffie-Hellman problem is hard, they have showed that El Gamal scheme provides anonymity i.e. key-privacy under chosen-plaintext attack (CPA) and that the Cramer-Shoup scheme provides stronger security i.e. it provides anonymity under chosen-ciphertext attack (CCA). They have also showed that neither the classical RSA scheme nor the RSA-OAEP does not provide key-privacy. All these schemes do not belong to the so-called family of ”post-quantum” crypto schemes since they are vulnerable to attacks with quantum computers.

The McEliece public key scheme [14] was published in 1978 and is based on the theory of error-correcting codes and the NP-hardness of the problem of decoding random linear codes. It is considered as a post-quantum scheme. However, the original scheme does not provide neither CPA nor CCA security, even less it does not provide a key-privacy. A conversion of McEliece scheme that offers CCA security was proposed by Kobara and Imai in [10] in the random oracles model. The weaker security property of CPA, but in the so-called standard model where there is no reference to the random oracles, for a modified McEliece scheme was proposed by Nojima et al. in [16] and later based on that work Yamakawa et al., in [23] showed that Nojima's modification provides also a key-privacy.

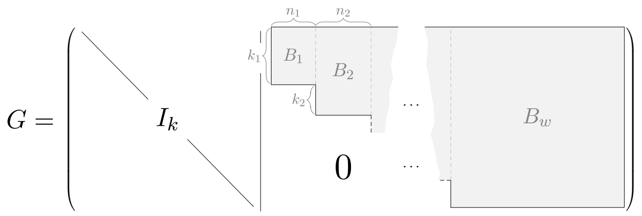
Recently we proposed an encryption and signature variant of the McEliece scheme based on Staircase-Generator matrix, a list decoding algorithm, and generalized error sets in [7,8]. Soon after its initial eprint publication, a distinguisher that distinguishes its public key from random matrices was proposed [20], and recently a very similar distinguising strategy and an Information Set Decoding (ISD) attack was presented as a full and practical key recovery attack by Moody and Perlner [15]. While the public key schemes based on Staircase-Generator matrices and a list decoding strategy have succumbed to the distinguishing attacks, there are some useful applications for the technique of matrix-embedding in steganography where the Staircase-Generator matrices are not public, but private [18]. In such cases, the matrix-embedding technique with Staircase-Generator matrices achieves almost the information theoretical bound with codelengths that are the smallest known in the literature.

In order to thwart distinguising and ISD attacks of [20,15] to the encryption scheme defined in [7,8] we proposed to split and replace the public generator matrix into s randomly generated matrices [19]. By this we made distinguishing attacks improbable to mount i.e. the probability of the attacker obtaining conditions under which a distinguisher or an ISD attack can be mounted, close to zero.

2 Definition of Staircase-Generator Codes and Random Split of Their Generator Matrix

Throughout the paper, we will denote by $\mathcal{C} \subseteq \mathbb{F}_2^n$ a binary (n, k) code of length n and dimension k. We will denote the generator matrix of the code by G, and $wt(\mathbf{x})$ will denote the Hamming weight of the word $\mathbf{x}$. We recall some of the basic definitions and properties for St-Gen codes from [7,18] and the types of errors used.

Definition 1. *Let $k_i, n_i \in \mathbb{N}$, and let $k = k_1 + k_2 + \cdots + k_w$ and $n = k + n_1 + n_2 + \cdots + n_w$. Further, let B_i be a random binary matrix of dimension $\sum_{j=1}^{i} k_j \times n_i$. A linear binary (n, k) code $\mathcal{C}$ with the following generator matrix in standard form:*



(1)

is called Staircase-Generator code (St-Gen code).

Definition 2. *Let ℓ be a positive integer and let $p_d \in \mathbb{F}_2[x_1, x_2, \ldots, x_\ell]$ be a multivariate polynomial of degree $\geqslant 2$. We say that E_ℓ is an* error set *if it is the solution set of p_d, i.e. $E_\ell = \{\mathbf{e} \in \mathbb{F}_2^\ell \mid p_d(\mathbf{e}) = 0\}$. We will refer to p_d as the defining polynomial.*

We define the density *of the error set E_ℓ to be $D(E_\ell) = |E_\ell|^{1/\ell}$. We will refer to the integer $\ell > 0$ as the* granulation *of E_ℓ. In [7] it was proven that if two error sets $E_{\ell_1} \subseteq \mathbb{F}_2^{\ell_1}$, $E_{\ell_2} \subseteq \mathbb{F}_2^{\ell_2}$, have the same density ρ, then $D(E_{\ell_1} \times E_{\ell_2}) = \rho$.*

Example 1. 1. Let $E_2 = \{\mathbf{x} \in \mathbb{F}_2^2 \mid wt(\mathbf{x}) < 2\} = \{(0,0), (0,1), (1,0)\}$. Then the error set can be described using the defining polynomial $p_d = x_1x_2$, and for the density of the error set we have $D(E_2) = |E_2|^{1/2} = 3^{1/2}$.

2. Let $E_4 = \{\mathbf{x} \in \mathbb{F}_2^4 \mid 2 \leq wt(\mathbf{x}) \leq 3\}$. Then, the defining polynomial for E_4 is $p_d = 1 + x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4$ and the density is $D(E_4) = D(E_4^m) = (\sum_{i=2}^{3} \binom{4}{i})^{1/4} = 10^{1/4}$ for any positive integer m.

The decoding of St-Gen codes relies on the technique of *list decoding*, a notion that dates back to the work of Elias [5] and Wozencraft [22] in the 1950's. In list decoding, the decoder is allowed to output a list of possible messages one of which is correct. List decoding can handle a greater number of errors than that allowed by unique decoding. In order for the decoding to be efficient, the size of the resulting list has to be polynomial in the code length. The following Proposition from [7] determines the parameters of a St-Gen code that provide an efficient decoding.

Proposition 1 ([7]). *Let $\mathcal{C}$ be any binary (n,k) code and $E \subset \mathbb{F}_2^n$ be an error set of density ρ. Let $\mathbf{w}$ be any word of length n, $W_E = \{\mathbf{w} + \mathbf{e} \mid \mathbf{e} \in E\}$ and let $\mathcal{C}_{W_E}$ denote the set of codewords in W_E. Suppose there exists a codeword $\mathbf{c} \in W_E$. Then the expected number of codewords in $W_E \setminus \{\mathbf{c}\}$ is approximately $\rho^n 2^{k-n}$ for large enough n and k.*

Let E_ℓ be an error set with density ρ where ℓ divides n and $m = n/\ell$. We recall Alg. 1 from [7], that is an efficient algorithm for decoding a code $\mathcal{C}$, that corrects errors from the set E_ℓ^m.

Algorithm 1 Decoding

Input: Vector $\mathbf{y} \in \mathbb{F}_2^n$, and generator matrix G of the form (1).

Output: A list $L_w \subset \mathbb{F}_2^k$ of valid decodings of $\mathbf{y}$.

Procedure:
Let $K_i = k_1 + \cdots + k_i$. Represent $\mathbf{x} \in \mathbb{F}_2^k$ as $\mathbf{x} = \mathbf{x}_1 \parallel \mathbf{x}_2 \parallel \cdots \parallel \mathbf{x}_w$ where each $\mathbf{x}_i$ has length k_i. Similarly, represent $\mathbf{y} \in \mathbb{F}_2^n$, as $\mathbf{y} = \mathbf{y}_0 \parallel \mathbf{y}_1 \parallel \mathbf{y}_2 \parallel \cdots \parallel \mathbf{y}_w$, where each $\mathbf{y}_i$ has length n_i and $|\mathbf{y}_0| = k$. We further identify $\mathbf{y}_0$ with $\mathbf{y}_0 = \mathbf{y}_0[1] \parallel \mathbf{y}_0[2] \parallel \cdots \parallel \mathbf{y}_0[w]$, where each $\mathbf{y}_0[i]$ is of length k_i.

During decoding, we will maintain lists $L_1, L_2, \ldots, L_w$ of possible decoding candidates of length K_i.

Step 0: Set a temporary list $T_0 = L_0$ to contain all possible decodings of the first k_1 coordinates of $\mathbf{y}$:

$$T_0 \leftarrow \{\mathbf{x}' = \mathbf{y}_0[1] + \mathbf{e} \mid \mathbf{e} \in E^{k_1/\ell}\}.$$

Step $1 \leq i \leq w$: Perform list-decoding to recover a list of valid decodings:
For each candidate $\mathbf{x}' \in T_{i-1} \subset \mathbb{F}_2^{K_i}$, add to L_i all the candidates for which $\mathbf{x}'B_i + \mathbf{y}_i \in E^{n_i/\ell}$:

$$L_i \leftarrow \{\mathbf{x}' \in T_{i-1} \mid \mathbf{x}'B_i + \mathbf{y}_i \in E^{n_i/\ell}\}. \tag{2}$$

If $i < w$ then create the temporary list T_i of candidates of length K_{i+1} from L_i:

$$T_i \leftarrow \{\mathbf{x}' \parallel (\mathbf{y}_0[i+1] + \mathbf{e}) \mid \mathbf{x}' \in L_i,\ \mathbf{e} \in E^{k_{i+1}/\ell}\}. \tag{3}$$

Return: L_w.

2.1 Random Split of the Staircase-Generator Matrix

A key parameter of the public-key encryption scheme where we split the staircase-generator matrix is the number of splits s, that determines the number of summands the generator matrix of the code is split in.

This parameter further determines the nature of the error used during encryption. We have the following:

Definition 3. *Let $E_\ell \subset \mathbb{F}_2^\ell$ be an error set of granulation ℓ and let s denote the number of splits. The s-tuple $ErrorSplit = (e_1, \dots, e_s)$, where $e_i \in \mathbb{F}_2^\ell, i \in \{1, \dots, s\}$ is called* A Valid Error Split *for E_ℓ if the sum of its elements permuted with any permutation $\sigma_i \in \mathcal{S}_\ell$ is an element of E_ℓ i.e. it holds that $e = \sum_{i=1}^{s} \sigma_i(e_i) \in E_\ell$. The set of all valid error splits is denoted as $ValidErrorSplits$ and its size with A i.e. $V = |ValidErrorSplits|$.*

Example 2. Let $\ell = 4$, $E_\ell = \{(0,0,0,0), (0,0,1,1), (0,1,0,1), (0,1,1,0), (1,0,1,1), (1,1,0,0), (1,1,0,1), (1,1,1,0), (1,1,1,1)\}$ and $s = 4$. The 4-tuple $ErrorSplit = ((1,0,0,0), (1,1,1,1), (1,1,1,1), (1,1,0,1))$ is *a valid error split for E_ℓ* because the sum of all its elements permuted by any of all possible $4! = 24$ permutations always gives an element in E_ℓ.

A formal description of the scheme is given through the next four algorithms for key generation, error set generation, encryption and decryption.

Algorithm 2 Key Generation

Parameters: Let $\ell | n$, $m = n/\ell$ and $E \subset \mathbb{F}_2^\ell$ be an error set of granulation ℓ and density ρ. Let s be the number of splits.

Key generation:
The following matrices make up the private key: - A generator matrix G of a binary (n, k) code of the form (1).
- An invertible matrix $S \in \mathbb{F}_2^{k \times k}$.
- An array of permutation matrices $P_1, P_2, \dots, P_s$ created as follows:

1. Select a permutation π on $\{1, 2, \dots, m\}$, and let $P \in \mathbb{F}_2^{n \times n}$ be the permutation matrix induced by π, so that for any $\mathbf{y} = \mathbf{y}_1 \parallel \mathbf{y}_2 \parallel \dots \parallel \mathbf{y}_m \in (\mathbb{F}_2^\ell)^m$:

$$\mathbf{y}P = \mathbf{y}_{\pi(1)} \parallel \mathbf{y}_{\pi(2)} \parallel \dots \parallel \mathbf{y}_{\pi(m)}, \tag{4}$$

i.e., P only permutes the m substrings of $\mathbf{y}$ of length ℓ.

2. For $i := 1$ to s:

Select randomly m permutations $\sigma_j^i \in \mathcal{S}_\ell$, $j \in \{1, \dots, m\}$.
Let P_i be defined by

$$\mathbf{y}P_i = \sigma_1^i(\mathbf{y}_{\pi(1)}) \parallel \sigma_2^i(\mathbf{y}_{\pi(2)}) \parallel \dots \parallel \sigma_m^i(\mathbf{y}_{\pi(m)}),$$

where $\sigma_j^i(\mathbf{x}) = \sigma_j^i(x_1, x_2, \dots, x_\ell)$.

The public key is formed as follows:

Generate uniformly at random $s-1$ matrices $G_1, \dots, G_{s-1}$ of size $k \times n$ over $\mathbb{F}_2$.
Set $G_s = G + G_1 + \dots + G_{s-1}$.
For all $i \in \{1, 2, \dots, s\}$, set $G_{\text{pub}}^i = SG_iP_i$.

Public key: $G_{\text{pub}}^1, \dots, G_{\text{pub}}^s$.
Private key: S, G and $P_1, P_2, \dots, P_s$.

Algorithm 3 Valid Error Splits (ℓ, E_ℓ, s)

Input: Granulation ℓ, error set E_ℓ, number of splits s.
Output: A set $ValidErrorSplits$ of all possible valid error splits.

```
Set $ValidErrorSplits \leftarrow \emptyset$
for all $(e_1, \ldots, e_s) \in (\mathbb{F}_2^\ell)^s$ do
    if $\sum_{i=1}^{s} \sigma_i(e_i) \in E_\ell, \forall(\sigma_1, \ldots, \sigma_s) \in (\mathcal{S}_\ell)^s$ then
        Add $(e_1, \ldots, e_s)$ to $ValidErrorSplits$.
    end if
end for
Return $ValidErrorSplits$.
```

Note that Algorithm 3 is run only once at the time of the initialization of the system with parameters ℓ, E_ℓ, s. Even more, in practice, this set can be pre-calculated and publicly available.

It is clear that the computational complexity of the encryption procedure with Random Split of St-Gen Codes compared to original St-Gen Codes is slower by a linear factor s, while the decryption complexity is almost the same (with a small overhead for Step 1 and Step 2 in the decryption algorithm).

Algorithm 4 Encryption $(\mathbf{m}, G^1_{\text{pub}}, \ldots, G^s_{\text{pub}}, ValidErrorSplits)$

Input: Message to be encrypted $\mathbf{m}$ the public key $G^1_{\text{pub}}, \ldots, G^s_{\text{pub}}$ and a set $ValidErrorSplits$ of all possible valid error splits.
Output: A ciphertext $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_s)$.

```
Set $\mathbf{c}_i = \mathbf{m}G^i_{\text{pub}} + \mathbf{e}_i$, $i = 1, \ldots, s$, where $\mathbf{e}_i = (e_{1,i}, \ldots, e_{\frac{n}{l},i})$ and where $(e_{j,1}, \ldots, e_{j,s})$, $j = 1, \ldots, \frac{n}{l}$ are
    randomly drawn from $ValidErrorSplits$.
Return $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_s)$
```

Algorithm 5 Decryption $(\mathbf{c}, S, G, P_1, P_2, \ldots, P_s)$

Input: Ciphertext $\mathbf{c}$, matrix S, the generator matrix G and the permutation matrices $P_1, P_2, \ldots, P_s$.
Output: A decrypted message $\mathbf{m}$.

```
1: Set $\mathbf{c}'_i = \mathbf{c}_i P_i^{-1}$
2: Set $\mathbf{c}' = \sum_{i=1}^{s} \mathbf{c}'_i$
3: Set $\mathbf{m}'$ as the output of Algorithm 1 (List decoding of $\mathbf{c}'$ with generator matrix $G$).
4: Set $\mathbf{m} = \mathbf{m}' S^{-1}$
5: Return $\mathbf{m}$
```

The initial proposal for encryption scheme that uses St-Gen codes [7], is vulnerable to an Information Set Decoding (ISD) attack [20,15]. The ISD technique was introduced by Prange [17], and later improved several times in the works of Lee and Brickell [11], Leon [12], Stern [21], and many others [6,4,13,1].

In a very recent analysis by Moody and Perlner [15] a modification of Stern's algorithm was provided, dedicated to cryptanalysis of the scheme in [7]. We refer the reader to [15] for details, and here we mention that complexity of the attack is in general given by $ISD_{St} = Pr_{St}^{-1} \cdot Cost_{St}$ where Pr_{St}^{-1} is the probability of success, and $Cost_{St}$ the cost of finding the low weight codeword.

In [19] we gave a detailed security analysis how and why the random split of the generator matrix prevents from ISD attacks.

2.2 Concrete parameter sets and their security

In [19] we gave the following parameter sets for practical use:

Parameter Set 1. $l = 3$, $s = 2$, $E_3 = \{(0,0,1), (0,1,0), (1,0,0), (0,1,1), (1,0,1), (1,1,0)\}$, $ValidErrorSplits = \Big\{ \big((0,0,0),(0,0,1)\big), \big((0,0,0),(0,1,0)\big), \big((0,0,0),(0,1,1)\big), \big((0,0,0),(1,0,0)\big),$ $\big((0,0,0),(1,0,1)\big), \big((0,0,0),(1,1,0)\big), \big((0,0,1),(0,0,0)\big), \big((0,0,1),(1,1,1)\big), \big((0,1,0),(0,0,0)\big),$ $\big((0,1,0),(1,1,1)\big), \big((0,1,1),(0,0,0)\big), \big((0,1,1),(1,1,1)\big), \big((1,0,0),(0,0,0)\big), \big((1,0,0),(1,1,1)\big),$ $\big((1,0,1),(0,0,0)\big), \big((1,0,1),(1,1,1)\big), \big((1,1,0),(0,0,0)\big), \big((1,1,0),(1,1,1)\big), \big((1,1,1),(0,0,1)\big),$ $\big((1,1,1),(0,1,0)\big), \big((1,1,1),(0,1,1)\big), \big((1,1,1),(1,0,0)\big), \big((1,1,1),(1,0,1)\big), \big((1,1,1),(1,1,0)\big)\Big\}.$
Note that $V = |ValidErrorSplits| = 24$. The defining polynomial for E_3 is $p_d = 1 + x_1 + x_2 + x_3 + x_1x_2 + x_1x_3 + x_2x_3$, and the density $\rho_3 = |E_3|^{1/\ell} = 6^{1/3}$. According to Proposition 1, and making a similar analysis as in [7], in order for the decryption process to be efficient, we need to keep the ratio $n_i/k_i \approx 6$, which implies $n \approx 7k$, and the size of the list in the end of the list decoding process will be $\rho^n 2^{k-n} \approx 1$.

Parameter Set 2. $l = 4$, $s = 2$, and $E_4 = \{(0,0,0,1), (0,0,1,0), (0,1,0,0), (1,0,0,0), (0,0,1,1), (0,1,1,0), (0,1,0,1), (1,0,0,1), (1,0,1,0), (1,1,0,0)\}$, $ValidErrorSplits =$
$\Big\{ \big((0,0,0,0),(0,0,0,1)\big), \big((0,0,0,0),(0,0,1,0)\big), \big((0,0,0,0),(0,0,1,1)\big), \big((0,0,0,0),(0,1,0,0)\big),$ $\big((0,0,0,0),(0,1,0,1)\big), \big((0,0,0,0),(0,1,1,0)\big), \big((0,0,0,0),(1,0,0,0)\big), \big((0,0,0,0),(1,0,0,1)\big),$ $\big((0,0,0,0),(1,0,1,0)\big), \big((0,0,0,0),(1,1,0,0)\big), \big((0,0,0,1),(0,0,0,0)\big), \big((0,0,1,0),(0,0,0,0)\big),$ $\big((0,0,1,1),(0,0,0,0)\big), \big((0,0,1,1),(1,1,1,1)\big), \big((0,1,0,0),(0,0,0,0)\big), \big((0,1,0,1),(0,0,0,0)\big),$ $\big((0,1,0,1),(1,1,1,1)\big), \big((0,1,1,0),(0,0,0,0)\big), \big((0,1,1,0),(1,1,1,1)\big), \big((0,1,1,1),(1,1,1,1)\big),$ $\big((1,0,0,0),(0,0,0,0)\big), \big((1,0,0,1),(0,0,0,0)\big), \big((1,0,0,1),(1,1,1,1)\big), \big((1,0,1,0),(0,0,0,0)\big),$ $\big((1,0,1,0),(1,1,1,1)\big), \big((1,0,1,1),(1,1,1,1)\big), \big((1,1,0,0),(0,0,0,0)\big), \big((1,1,0,0),(1,1,1,1)\big),$ $\big((1,1,0,1),(1,1,1,1)\big), \big((1,1,1,0),(1,1,1,1)\big), \big((1,1,1,1),(0,0,1,1)\big), \big((1,1,1,1),(0,1,0,1)\big),$ $\big((1,1,1,1),(0,1,1,0)\big), \big((1,1,1,1),(0,1,1,1)\big), \big((1,1,1,1),(1,0,0,1)\big), \big((1,1,1,1),(1,0,1,0)\big),$ $\big((1,1,1,1),(1,0,1,1)\big), \big((1,1,1,1),(1,1,0,0)\big), \big((1,1,1,1),(1,1,0,1)\big), \big((1,1,1,1),(1,1,1,0)\big)\Big\}.$
Note that $V = |ValidErrorSplits| = 40$. The defining polynomial for E_4 is $p_d = 1 + x_1 + x_2 + x_3 + x_4 + x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4$. Here, the density is $\rho_4 = |E_4|^{1/\ell} = 10^{1/4}$, and efficiency requires $n_i/k_i \approx 5$, i.e. $n \approx 6k$. For the size of the list in the end of the list decoding process, we again obtain that it is ≈ 1.

Regarding the ISD attacks, for the two parameter sets, we have that:

- for Par. Set 1, the highest probability is obtained for $t = 2$, and is $Pr(k, 2) = 2^{-0.5k}$,
- for Par. Set 2, the highest probability is obtained for $t = 3$, and is: $Pr(k, 3) = 2^{-0.86k}$.

Hence for any of the two parameter sets, $k > 256$ is enough for security of 128 bits against ISD attacks.

We proposed four concrete codes from the two parameter sets, two of each set, providing security of 80 and 128 bits, respectively.

We denote by $K = (k_1, \dots, k_w)$ and $N = (n_1, \dots, n_w)$ the vectors of values used in the definition of concrete generator matrices as defined in equation (1).

The following are concrete codes from Parameter Set 1:

- Code $(5514, 762)$,
  $w = 250$, $K = (15, 3, 3, \dots, 3)$, $N = (21, 21, 18, 18, 21, 18, 18 \dots, 21, 18, 18)$.
- Code $(7956, 1095)$,
  $w = 361$, $K = (15, 3, 3, \dots, 3)$, $N = (21, 21, 18, 18, 21, 18, 18 \dots, 21, 18, 18)$.

The following are concrete codes from Parameter Set 2:

- Code $(4600, 776)$.
  $w = 191$, $K = (16, 4, 4, \dots, 4)$, $N = (24, 20, \dots, 20, 20)$.
- Code $(7000, 1776)$.
  $w = 291$, $K = (16, 4, 4, \dots, 4)$, $N = (24, 20, \dots, 20, 20)$.

Based on all this we can formulate the following plausible Conjecture:

Conjecture 1. The public key $G^1_{\text{pub}}, \dots, G^s_{\text{pub}}$ produced by Algorithm 1 is indistinguishable from a set of s random $[n, k]$ codes and inverting the encryption with $G^1_{\text{pub}}, \dots, G^s_{\text{pub}}$ without the knowledge of the private key S, G and $P_1, P_2, \dots, P_s$ is infeasible in polynomial time.

3 Achieving IND-CPA and IK-CPA by padding the plaintext with a random bit-string

In this section we apply the ideas described by Nojima et al. in [16] and by Yamakawa et al., in [23] for the McEliece scheme, to show that our encryption scheme with random split of the generator matrix can achieve semantic security against chosen plaintext attack (IND-CPA) and key-privacy i.e. indistinguishability of public keys under chosen plaintext attack (IK-CPA).

Firs we give a definition of indistinguishability of encrypted data against the chosen plaintext attack (CPA) as it is given in [3].

Definition 4 (IND-CPA [3]). *Let a PKE scheme be the following tuple of polynomial-time algorithms: PKE = (Gen, Enc, Dec).*

1. *On input of security parameter κ, key generation algorithm $Gen(1^\kappa)$ outputs the set of private-key and public-key, (pk, sk).*
2. *Given (pk, sk), a polynomial-time adversary $\mathcal{A}$ chooses two equal-length plaintexts m_0, m_1, $(m_0 \neq m_1)$, and sends them to the encryption oracle.*
3. *Encryption oracle (algorithm) randomly flips coin $b \in \{0, 1\}$, to encrypt $Enc(pk, m_b) = c$.*
4. *Given target ciphertext c, adversary $\mathcal{A}$ outputs $b' \in \{0, 1\}$, where the advantage of success probability over random guess is defined as follows:*

$$\mathbf{Adv}^{ind-cpa}_{\mathcal{A}}(\kappa) = Pr[b' = 0 | b = 0] + Pr[b' = 1 | b = 1]. \tag{5}$$

*If $\mathbf{Adv}^{ind-cpa}_{\mathcal{A}}(\kappa)$ is negligible, then, we say underlying PKE is IND-CPA secure. Here "negligible" means that for any constant const, there exists $k_0 \in \mathbb{N}$, s.t. for any $\kappa > k_0$, **Adv** is less than $\left(\frac{1}{\kappa}\right)^{const}$.*

As we said, in order to achieve IND-CPA in a standard model, Nojima et al., [16] proposed a random prepadding for messages in McEliece scheme i.e. instead of encrypting messages m to encrypt messages $[r|m]$ where r is a random prepadding. However, in order to achieve more than 2^{80} security or more than 2^{128} security, r should have significant length. More concretely for the McEliece code $(2048, 1289, 69)$ to achieve security of 2^{85}, out of 1289 bits the random prepadding r has to have 1161 bits and only 128 bits are left for m. For the other code $(4096, 2560, 128)$, to achieve security of 2^{131}, out of 2560 bits the random prepadding r has to have 2048 bits and only 512 bits are left for m.

In what follows we investigate the IND-CPA security of the approach of encrypting messages in a form $\mathbf{m} = [r|m]$ for our scheme with a random split of St-Gen matrix (we abbreviate the name of that system as RRS-St-Gen - Randomized Random Split of St-Gen - in the mathematical formulas that refer to that system). By having this form for $\mathbf{m}$ the encryption gets the following form: The ciphertext is the s-tuple $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_s)$ where

$$\mathbf{c}_i = \mathbf{m}G^i_{\text{pub}} + \mathbf{e}_i = (rG^i_{\text{pub}_1} + \mathbf{e}_i) + mG^i_{\text{pub}_2}, \tag{6}$$

where $G^i_{\text{pub}} = \left[\frac{G^i_{\text{pub}_1}}{G^i_{\text{pub}_2}}\right]$, $\mathbf{e}_i = (e_{1,i}, \ldots, e_{\frac{n}{l},i})$ and where $(e_{j,1}, \ldots, e_{j,s})$, $j = 1, \ldots, \frac{n}{l}$ are randomly drawn from $ValidErrorSplits$.

Theorem 1. *Let RRS-St-Gen is a randomized public key encryption scheme as defined in Alg. 1 - Alg. 5 with the following parameters: n, k, l, s, E_l, $ValidErrorSplits$ and $V = |ValidErrorSplits|$, encrypting messages $\mathbf{m} = [r|m]$ where r is a random prepadding. If the s-tuple of matrices $(G^1_{pub}, \ldots, G^s_{pub})$ is indistinguishable from random, and the inversion of the encryption with $G^1_{pub}, \ldots, G^s_{pub}$ without the knowledge of the private key S, G and $P_1, P_2, \ldots, P_s$ is infeasible in polynomial time, then the advantage of an adversary given by relation (5) is:*

$$\mathbf{Adv}^{ind-cpa}_{\mathcal{A}-RRS-St-Gen}(\kappa) = 2\left(\left(1 - \frac{1}{2^{|r|}}\right)\left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}\left(1 - \left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}\right) + \frac{1}{2^{|r|}}\left(1 - \left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}\right)\right) \tag{7}$$

Proof. First note that the assumption about the infeasibility of the inversion of the encryption is a crucial one. If the adversary is capable to invert the encryption, it will simply obtain the whole random prepadding and will guess the value of b' with probability 1. The necessity of the assumption for the indistinguishability from random matrices is due to the attacks that reveal the secret key of the scheme and then is also connected with the inversion of the encryption. We discuss additionally these two assumptions at the end of this proof.

We will compute the probability $Pr[b' = 0|b = 0]$ and due to the symmetry, the probability for $Pr[b' = 1|b = 1]$ has the same value. First let us note that every $\mathbf{c}_i$ have $\frac{n}{l}$ chunks of length l bits. For a concrete value of $\mathbf{m} = [r|m]$ the encryption procedure randomly picks $\frac{n}{l}$ elements from set of s-tuples $ValidErrorSplits$. Note also that the set $ValidErrorSplits$ is a subset of all possible l-bit s-tuples which number is $2^{s\cdot l}$. In case when m_0 was encrypted there are two possible and disjunctive events that can lead the adversary to make the right guess $b' = 0$. Those two events are the following:

- $Event_1 : Event_{1,1} \cap Event_{1,2} \cap Event_{1,3}$
  1. $Event_{1,1}$: The adversary made a wrong guess about the prepadded random value r. $Pr(Event_{1,1}) = 1 - \frac{1}{2^{|r|}}$;

2. $Event_{1,2}$: The adversary computes $\mathbf{c}_i^{(m_0)} = rG^i_{\text{pub}_1} + m_0 G^i_{\text{pub}_2}$ and for all $\frac{n}{l}$ chunks in all $\mathbf{c}_i$, there exist a valid error split in $ValidErrorSplits$ as a connection between $\mathbf{c}_i^{(m_0)}$ and $\mathbf{c}_i$. $Pr(Event_{1,2}) = \left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}$;
3. $Event_{1,3}$: The adversary computes $\mathbf{c}_i^{(m_1)} = rG^i_{\text{pub}_1} + m_1 G^i_{\text{pub}_2}$, and there is at least one chunk for which there is no valid error split in $ValidErrorSplits$.
   $Pr(Event_{1,3}) = \left(1 - \left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}\right)$.

- $Event_2 : Event_{2,1} \cap Event_{2,2}$
  1. $Event_{2,1}$: The adversary made a correct guess about the prepadded random value r. $Pr(Event_{2,1}) = \frac{1}{2^{|r|}}$;
  2. $Event_{2,2}$: The adversary computes $\mathbf{c}_i^{(m_1)} = rG^i_{\text{pub}_1} + m_1 G^i_{\text{pub}_2}$, and there is at least one chunk for which there is no valid error split in $ValidErrorSplits$.
     $Pr(Event_{2,2}) = \left(1 - \left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}\right)$.

Composing all this gives us:

$$\begin{aligned} Pr[b' = 0 | b = 0] &= Pr(Event_1) + Pr(Event_2) \\ &= Pr(Event_{1,1})Pr(Event_{1,1})Pr(Event_{1,1}) + Pr(Event_{2,1})Pr(Event_{2,2}) \\ &= \left(1 - \frac{1}{2^{|r|}}\right)\left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}\left(1 - \left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}\right) + \frac{1}{2^{|r|}}\left(1 - \left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}\right) \end{aligned}$$

Since the case $Pr[b' = 1 | b = 1]$ is symmetrical, the total $\mathbf{Adv}^{ind-cpa}_{\mathcal{A}-RRS-St-Gen}(\kappa)$ is:

$$\mathbf{Adv}^{ind-cpa}_{\mathcal{A}-RRS-St-Gen}(\kappa) = 2\left(\left(1 - \frac{1}{2^{|r|}}\right)\left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}\left(1 - \left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}\right) + \frac{1}{2^{|r|}}\left(1 - \left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}\right)\right)$$

A direct conclusion from the security analysis in [19] is that if the length of the message that is encrypted with the scheme is bigger than 256 bits, than the claims in the Conjecture 1 about the indistinguishability of the public key from random matrices and the infeasibility of the inversion of the encryption are plausible. So, for any concrete instantiation of the scheme, the security levels that are achieved with the adversary advantage defined in the relation (7) have to give lengths of the random prepadding to be more than 256 bits. □

By checking the validity of the final part of the proof of Theorem 1 we obtain that our scheme achieves the IND-CPA security level of 2^{80} for $|r| = 264$ and the security level of 2^{128} for $|r| = 424$. So these values being higher than 256 are in accordance with the security analysis in [19] and the Conjecture 1. Moreover, they are still significantly lower than the lengths of the prepadded random part in the modified McEliece scheme.

For the key-privacy issue we use the same approach as Yamakawa et al., have in [23].
Definition 5 (IK-CPA [2]). *Let a PKE scheme be the following tuple of polynomial-time algorithms: PKE = (Gen, Enc, Dec). The security of key-privacy is defined as follows.*

1. *On input of security parameter κ, key generation algorithm $Gen(1^\kappa)$ outputs two independent sets of key pairs (pk_0, sk_0), (pk_1, sk_1).*

2. *Given (pk_0), (pk_1), a polynomial-time adversary $\mathcal{A}$ chooses a plaintext m and sends them to the encryption oracle.*
3. *Encryption oracle randomly flips coin $b \in \{0,1\}$, to output $Enc_{pk_b}(m) = c$.*
4. *Given target ciphertext c, adversary $\mathcal{A}$ outputs $b' \in \{0,1\}$, where the advantage of success probability over random guess is defined as follows:*

$$\mathbf{Adv}_{\mathcal{A}}^{ik-cpa}(\kappa) = Pr[b' = 0|b = 0] + Pr[b' = 1|b = 1]. \tag{8}$$

If $\mathbf{Adv}_{\mathcal{A}}^{ik-cpa}(\kappa)$ is negligible, then, we say underlying PKE is IK-CPA secure.

While the modeling of IK-CPA is not the same as IND-CPA, the value about $\mathbf{Adv}_{\mathcal{A}}^{ik-cpa}(\kappa)$ is the same as for IND-CPA case. Thus we have the following theorem that we give without a proof:

Theorem 2. *Let RRS-St-Gen is a randomized public key encryption scheme as defined in Alg. 1 - Alg. 5 with the following parameters: n, k, l, s, E_l, $ValidErrorSplits$ and $V = |ValidErrorSplits|$, encrypting messages $\mathbf{m} = [r|m]$ where r is a random prepadding. If the s-tuple of matrices $(G_{pub}^1, \ldots, G_{pub}^s)$ is indistinguishable from random, and the inversion of the encryption with $G_{pub}^1, \ldots, G_{pub}^s$ without the knowledge of the private key S, G and $P_1, P_2, \ldots, P_s$ is infeasible in polynomial time, then the advantage of an adversary given by relation (8) is:*

$$\mathbf{Adv}_{\mathcal{A}-RRS-St-Gen}^{ik-cpa}(\kappa) = 2\left(\left(1 - \frac{1}{2^{|r|}}\right)\left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}\left(1 - \left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}\right) + \frac{1}{2^{|r|}}\left(1 - \left(\frac{V}{2^{s\cdot l}}\right)^{\frac{n}{l}}\right)\right) \tag{9}$$

4 Conclusions

We have presented a public key encryption scheme based on St-Gen codes and its variant where we split and replace the public generator matrix into s randomly generated matrices. The split strategy is used to thwarts the ISD attacks on the encryption scheme. Then, we showed that randomized version of the encryption scheme offers semantic security against chosen plaintext attack (IND-CPA) and offers key-privacy i.e. offers indistinguishability of public keys under chosen plaintext attack (IK-CPA) in the standard model. The difference with McEliece scheme is that with our scheme the length of the prepadded random string is significantly shorter. It remains as a next goal to investigate the modification of the scheme for achieving CCA and CCA2 security both with and without the random oracle model.

References

1. Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in 2n/20: how 1 &#43; 1 = 0 improves information set decoding. In *Proceedings of the 31st Annual international conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'12, pages 520–536, Berlin, Heidelberg, 2012. Springer-Verlag. (Cited on page 5.)
2. Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *Advances in Cryptology?ASIACRYPT 2001*, pages 566–582. Springer, 2001. (Cited on pages 1 and 9.)
3. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology?CRYPTO'98*, pages 26–45. Springer, 1998. (Cited on pages 1 and 7.)

4. Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: ball-collision decoding. In *Proceedings of the 31st annual conference on Advances in cryptology*, CRYPTO'11, pages 743–760, Berlin, Heidelberg, 2011. Springer-Verlag. (Cited on page 5.)
5. P. Elias. List decoding for noisy channels, technical report 335. Technical report, Research Laboratory of Electronics, MIT, 1957. (Cited on page 3.)
6. Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In *Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '09, pages 88–105, Berlin, Heidelberg, 2009. Springer-Verlag. (Cited on page 5.)
7. Danilo Gligoroski, Simona Samardjiska, Håkon Jacobsen, and Sergey Bezzateev. McEliece in the world of Escher. Cryptology ePrint Archive, Report 2014/360, 2014. `http://eprint.iacr.org/`. (Cited on pages 2, 3, 5, and 6.)
8. Danilo Gligoroski, Simona Samardjiska, Håkon Jacobsen, and Sergey Bezzateev. A new code based public key encryption and signature scheme based on list decoding. Presented at ?Workshop on Cybersecurity in a Post-Quantum World,? NIST, Gaithersburg MD, USA, 2015. `http://www.nist.gov/itl/csd/ct/post-quantum-crypto-workshop-2015.cfm`, [Retrieved: October 2015]. (Cited on page 2.)
9. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984. (Cited on page 1.)
10. Kazukuni Kobara and Hideki Imai. Semantically secure mceliece public-key cryptosystems-conversions for mceliece pkc. In *Public Key Cryptography*, pages 19–35. Springer, 2001. (Cited on page 2.)
11. P. J. Lee and E. F. Brickell. An observation on the security of mceliece's public-key cryptosystem. In *Lecture Notes in Computer Science on Advances in Cryptology-EUROCRYPT'88*, pages 275–280, New York, NY, USA, 1988. Springer-Verlag New York, Inc. (Cited on page 5.)
12. J. S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Trans. Inf. Theor.*, 34(5):1354–1359, September 2006. (Cited on page 5.)
13. Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in &#213;(20.054n). In *Proceedings of the 17th international conference on The Theory and Application of Cryptology and Information Security*, ASIACRYPT'11, pages 107–124, Berlin, Heidelberg, 2011. Springer-Verlag. (Cited on page 5.)
14. R. J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44. (Cited on page 1.)
15. Dustin Moody and Ray Perlner. Vulnerabilities of "McEliece in the World of Escher". Cryptology ePrint Archive, Report 2015/966, 2015. `http://eprint.iacr.org/`. (Cited on pages 2 and 5.)
16. Ryo Nojima, Hideki Imai, Kazukuni Kobara, and Kirill Morozov. Semantic security for the mceliece cryptosystem without random oracles. *Designs, Codes and Cryptography*, 49(1-3):289–305, 2008. (Cited on pages 2, 7, and 8.)
17. E. Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8:5–9, 1962. (Cited on page 5.)
18. Simona Samardjiska and Danilo Gligoroski. Approaching maximum embedding efficiency on small covers using staircase-generator codes. In *Information Theory (ISIT), 2015 IEEE International Symposium on*, pages 2752–2756, June 2015. (Cited on page 2.)
19. Simona Samardjiska and Danilo Gligoroski. An Encryption Scheme based on Random Split of St-Gen Codes. Cryptology ePrint Archive, Report 2016/202, 2016. `https://eprint.iacr.org/2016/202`. (Cited on pages 2, 5, 6, and 9.)
20. Nicolas Sendrier and Jean-Pierre Tillich. Private communication, October 2014. (Cited on pages 2 and 5.)
21. Jacques Stern. A method for finding codewords of small weight. In *Proceedings of the 3rd International Colloquium on Coding Theory and Applications*, pages 106–113, London, UK, UK, 1989. Springer-Verlag. (Cited on page 5.)
22. J. M. Wozencraft. List decoding. quarterly progress report. Technical report, Research Laboratory of Electronics, MIT, 1958. (Cited on page 3.)
23. Shigenori Yamakawa, Yang Cui, Kazukuni Kobara, Manabu Hagiwara, and Hideki Imai. On the key-privacy issue of mceliece public-key encryption. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pages 168–177. Springer, 2007. (Cited on pages 2, 7, and 9.)