A New Approach for Algebraically Homomorphic Encryption

Frederik Armknecht and Ahmad-Reza Sadeghi

Ruhr-Universität Bochum, Germany
{Frederik.Armknecht,Ahmad.Sadeghi}@trust.rub.de

Abstract. The existence of an efficient and provably secure algebraically homomorphic scheme (AHS), i.e., one that supports both addition and multiplication operations, is a long stated open problem. All proposals so far are either insecure or not provable secure, inefficient, or allow only for one multiplication (and arbitrary additions). As only very limited progress has been made on the existing approaches in the recent years, the question arises whether new methods can lead to more satisfactory solutions.
In this paper we show how to construct a provably secure AHS based on a coding theory problem. It allows for arbitrary many additions and for a fixed, but arbitrary number of multiplications and works over arbitrary finite fields. Besides, it possesses some useful properties: i) the plaintext space can be extended adaptively without the need for re-encryption, ii) it operates over arbitrary infinite fields as well, e.g., rational numbers, but the hardness of the underlying decoding problem in such cases is less studied, and iii) depending on the parameter choice, the scheme has inherent error-correcting up to a certain number of transmission errors in the ciphertext.
However, since our scheme is symmetric and its ciphertext size grows exponentially with the expected total number of encryptions, its deployment is limited to specific client/server applications with few number of multiplications. Nevertheless, we believe room for improvement due to the huge number of alternative coding schemes that can serve as the underlying hardness problem. For these reasons and because of the interesting properties of our scheme, we believe that using coding theory to design AHS is a promising approach and hope to encourage further investigations.

Keywords: Algebraically Homomorphic Encryption, Coding Theory, Provable Security

1 Introduction

Homomorphic encryption schemes preserve the underlying algebraic structure which allows for performing operations in encrypted domain without the need for re-encryption. More precisely, a (group) homomorphic encryption scheme over a group $(G, *)$ has the following properties: given the encryptions $\mathsf{E}_K(m)$ and $\mathsf{E}_K(m')$ where $m, m' \in G$ and K is the encryption key, one can efficiently and securely compute $\mathsf{E}_K(m * m')$ without revealing m and m'. Homomorphic encryption schemes have many applications, such as electronic voting [9, 2, 12, 13], private information retrieval [25, 26], or multiparty computation [11]. Up to now, several secure and efficient group homomorphic encryption schemes are known, e.g., RSA [34], ElGamal [20], Paillier [30], and Damgaard and Jurik [14].

Algebraically homomorphic encryption schemes (AHS) that support both operations, i.e., addition and multiplication, will benefit all these problems. The problem of constructing efficient and secure AHS is a long standing open question already mentioned by Rivest et al. [33]. Indeed, Boneh and Lipton [6] gave a partially negative answer to the problem by proving that any *deterministic* AHS can be broken in sub-exponential time. So far only a few algebraic encryption schemes have been proposed. Fellows and Koblitz [18] proposed an asymmetric scheme named 'Polly Cracker' which is based on the difficulty of solving systems of non-linear equations. According to the current state of knowledge, all its instantiations (and variations like PollyTwo [27]) are either insecure,

inefficient, or loose their homomorphic property (e.g., see [19, 15]). Domingo-Ferrer proposed two symmetric schemes based on polynomial interpolation [16, 17] but these have been broken afterwards [36, 1, 8]. Rappe [31] showed that AHS can be constructed from (single-)homomorphic schemes over certain semigroups but for the latter no efficient solutions are known. Sander, Young and Yung [35] described a scheme that is algebraically homomorphic over a semigroup. However, the homomorphism comes with the cost of a constant factor expansion per semigroup operation. Recently, Melchor, Gaborit, and Herranz [28] introduced the concept of t-chained pseudo-homomorphic schemes to (theoretically) construct AHS which support arbitrary many additions and up to t multiplications. However, no formal proof of security is given and the considered constructions have a large ciphertext size but operate over a small plaintext space only. To the best of our knowledge, the only provably secure AHS so far was given by Boneh, Goh, and Nissim [5]. It allows for arbitrary many additions but only one multiplication. A further problem is that the plaintext space needs to be small; the authors consider the binary field GF(2). In summary, it is fair to say that the problem of finding an efficient and provable secure AHS is not solved yet. As only very limited progress has been made on the existing approaches, the question arises if new methods may lead to satisfactory solutions.

Our contribution. In this paper, we show a novel way for constructing AHS. The proposed scheme is a modification of a non-homomorphic scheme by Kiayias and Yung [22]. It works over arbitrary finite fields and allows for an unlimited number of additions and a fixed, but arbitrary number of multiplications. It is provable secure under a known decoding problem, namely to decode a special class of *interleaved Reed-Solomon codes* [32]. Furthermore, the problem seems to remain difficult in the quantum computational model (see Goldreich, Rubinfeld, Sudan [21] and Bennett, Bernstein, Brassard [3]).

The basic idea can be sketched as follows: A plaintext is encoded into a codeword of an error-correcting code where some artificial errors are induced at fixed (but secret) locations (called bad locations). Decoding is efficient when the bad locations are known. Otherwise, breaking the ciphertext is equivalent to decoding highly noisy codewords. The homomorphic property follow from the fact that the sum and the componentwise product of two codewords yield a codeword again.

Besides being algebraically homomorphic, our scheme has some additional remarkable properties:

- *Adaptive plaintext space extension:* The plaintext space can be extended subsequently *after* having already computed and stored a number of encryptions. This could be for example the case if it turns out that the encoding of the data needs a lager range than initially expected. Usually, this requires decryption and re-encryption for all data. In our scheme, the plaintext space can be easily extended to any extension fields without the need to decrypt and re-encrypt.
- *Support for infinite fields:* The proposed scheme works correctly over infinite fields as well, e.g., over rational numbers. However, the decoding problem over infinite fields is less explored and hence the hardness assumption requires further investigation.
- *Inherent error-correction:* The scheme tolerates a certain number of transmission errors, depending on the parameter choices.

Discussion. The scheme has some limitations. Firstly, it is a symmetric key scheme as opposed to most known homomorphic encryption schemes. However, for many client-server-applications this

may not be relevant since the encrypted result of the computation is returned to the client who knows the decryption key. Secondly, to guarantee security, the ciphertext length has to be chosen in dependence of the expected total number of encryptions (where combinations of existing ciphertexts do not count as new encryptions) and the blow-up factor is exponential. Therefore, the applicability of the scheme is limited to specific client-server-applications with few multiplications. However, this blow-up is an immediate result from the existence of dedicated decoding algorithms for interleaved Reed-Solomon codes. It might well be (and we are not aware of any counterarguments) that more efficient schemes are realizable by switching to other coding schemes.

We do not mean to minimize the above concerns, only to suggest how they might be overcome. For these reasons and because of the interesting properties of our scheme, we believe that using coding theory to design AHS is a promising approach and hope to encourage further investigations.
Outline. The paper is organized as follows. In Section 2, we provide the necessary preliminaries. In Section 3, we describe the encryption scheme and prove some of its properties. In Section 4, we prove that the scheme is semantically secure under a given hardness assumption taken from coding theory. In Section 5, we discuss the parameter choices of our scheme and in Section 6 some possible extensions. Section 7 concludes the paper.

2 Preliminaries

2.1 Notation

For an integer $n \geq 1$, we denote by $[n]$ the set of integers $1, \ldots, n$. In the following, s denotes a security parameter. $\mathbb{F}$ will denote an arbitrary field that can be finite or non-finite, e.g. the field of rationals (in the latter case, we interpret the expression $1/|\mathbb{F}|$ as zero). Let $\mathbb{F}[x]$ denote the ring of univariate polynomials in the indeterminate x with coefficients from $\mathbb{F}$. $\mathrm{HW}(\overrightarrow{v})$ denotes the Hamming weight of of vector $\overrightarrow{v}$, that is the number of non-zero entries. For two vectors $\overrightarrow{v}$ and $\overrightarrow{w}$, the expression $\overrightarrow{v} \bullet \overrightarrow{w}$ stands for the component wise product (not to be mixed up with the vector product). For example, for $\overrightarrow{v} = (v_1, \ldots, v_n)$ and $\overrightarrow{w} = (w_1, \ldots, w_n)$, it is $\overrightarrow{v} \bullet \overrightarrow{w} = (v_1 \cdot w_1, \ldots, v_n \cdot w_n)$. For a polynomial $p(x) \in \mathbb{F}[x]$ and a vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}^n$, we define $p(\overrightarrow{x}) := (p(x_1), \ldots, p(x_n)) \in \mathbb{F}^n$. A function $f : \mathbb{N} \to \mathbb{R}$ is called negligible if for any $n' \in \mathbb{N}$ exists a polynomial $p(x)$ over the real numbers such that $|f(n)| < |1/p(n)|$ for all $n \geq n'$. We sometimes write $f = \mathrm{negl}(n)$.

2.2 The Synchronized Polynomial Reconstruction Problem

In this section, we describe the Synchronized Polynomial Reconstruction Problem (SRP) on which our scheme is based on. The SRP is a special case of the Polynomial Reconstruction Problem which has been used several times to design cryptographic algorithms, e.g. Naor and Pinkas [29] and Kiayias and Yung [24]. For an overview, we recommend [23].

The PRP is derived from Reed-Solomon codes [32]. The key idea behind a Reed-Solomon code $\mathrm{RS}[n, k]$ with integers $n > k$ is that the data is represented by a polynomial of degree $< k$. The code relies on a theorem from linear algebra stating that any k distinct points uniquely determine a polynomial of degree $< k$. The polynomial is then "encoded" by its evaluation at n various points, and these values are what is actually sent. During transmission, some of these values may become corrupted. Therefore, more than k points are actually sent. As long as sufficient values are received correctly, the receiver can deduce what the original polynomial was, and hence decode the original

data. The decoding problem of Reed-Solomon codes where at most $n - t$ errors have occurred can be equivalently described as the following polynomial reconstruction problem (see [24]):

Definition 1. ***[Polynomial Reconstruction Problem (PRP)]*** *Let $\mathbb{F}$ be an arbitrary field. Given parameters $n, k, t \in \mathbb{N}_{\geq 0}$, and two vectors $\overrightarrow{x} = (x_1, \ldots, x_n), \overrightarrow{y} = (y_1, \ldots, y_n) \in \mathbb{F}^n$ with $x_i \neq x_j$ for $i \neq j$, output a tuple $(p_{\overrightarrow{y}}(x); I(\overrightarrow{y}))$ such that*

- *$p_{\overrightarrow{y}}(x) \in \mathbb{F}[x]$ is a polynomial over $\mathbb{F}$ with $\deg(p_{\overrightarrow{y}}(x)) < k$ (the solution polynomial),*
- *$I(\overrightarrow{y}) \subseteq [n]$ is a subset of distinct indices i with $|I(\overrightarrow{y})| \geq t$ (the indices of the error-free entries), and*
- *$p_{\overrightarrow{y}}(x_i) = y_i$ for all $i \in I(\overrightarrow{y})$.*

$(p_{\overrightarrow{y}}(x); I(\overrightarrow{y}))$ is called a solution of $\overrightarrow{y}$. A PRP instance is a vector $\overrightarrow{y} = (y_1, \ldots, y_n) \in \mathbb{F}^n$ and $\mathcal{PR}_{\overrightarrow{x},k,t} \subset \mathbb{F}^n$ denotes the set of PRP instances.

In [24] it was shown that if $\log(|\mathbb{F}| - 1) \geq \frac{\log(\binom{n}{t}) + s}{t-k}$, then a PRP instance has a unique solution with probability $\geq 1 - 2^{-s}$. For the remainder of the paper, we assume that this is the case and talk about *the* solution of a PRP instance. By decoding a PRP instance $\overrightarrow{y}$, we mean to find its solution $(p_{\overrightarrow{y}}; I(\overrightarrow{y}))$ which corresponds to the notion of decoding Reed-Solomon codes as explained above. Furthermore, we call the positions $i \in I$ as *good* locations and the others *bad* locations. One important property of PRP instances is that they can be efficiently sampled (see [24]):

Definition 2. ***[Sampling PRP-instances]*** *Consider the following sampler $\mathcal{S}$ that samples instances $\overrightarrow{y}$ of $\mathcal{PR}_{\overrightarrow{x},k,t}$: $\mathcal{S}$ on input $(\overrightarrow{x}, k, t)$ samples a random subset $I \subset [n]$ of size t, a polynomial $p_{\overrightarrow{y}} \in \mathbb{F}[x]$ with $\deg(p_{\overrightarrow{y}}) < k$; it then sets $y_i := p(x_i)$ for all $i \in I$, whereas for all $i \notin I$ it samples y_i at random from the set $\mathbb{F} \setminus \{p(x_i)\}$. $\mathcal{S}$ terminates by returning the vector $\overrightarrow{y} = (y_1, \ldots, y_n)$. We denote the induced distribution on $\mathbb{F}^n$ by $\mathcal{D}_{\overrightarrow{x},k,t}$.*

The hardness assumption from [24] is defined as follows:

Definition 3. ***[Decisional-PRP*** $\mathrm{DPRP}[\overrightarrow{x}, k, t]$***]*** *Given parameters $\overrightarrow{x}, k, t$, the sampler $\mathcal{S}^{\mathrm{bad}}$ first selects an instance $\overrightarrow{y}$ following the sampler $\mathcal{S}$ of definition 2, then it selects i at random from the set $[n] \setminus I(\overrightarrow{y})$ and then outputs $(i, \overrightarrow{y})$. $\mathcal{S}^{\mathrm{good}}$ is defined similarly but i is selected at random from the set $I(\overrightarrow{y})$ instead. For any probabilistic polynomial-time algorithm $\mathcal{A}$ we define:*

$$Adv^{DPRP,\mathcal{A}}_{\overrightarrow{x},k,t}(s) = \left| Pr[\mathcal{A}(\mathcal{S}^{\mathrm{good}}(\overrightarrow{x}, k, t) = 1] - Pr[\mathcal{A}(\mathcal{S}^{\mathrm{bad}}(\overrightarrow{x}, k, t) = 1] \right| \tag{1}$$

where the probability is taken over the random coins from $\mathcal{A}$ and the samplers. The $DPRP[\overrightarrow{x}, k, t]$ assumption holds if $Adv^{dpr}_{\overrightarrow{x},k,t}(s) := \max_{\mathcal{A}} Adv^{dpr,\mathcal{A}}_{\overrightarrow{x},k,t}(s) = \mathrm{negl}(s)$, that is any algorithm $\mathcal{A}$ has a negligible advantage.

Informally speaking, the assumption says that it is hard to decide for a given PRP instance $\overrightarrow{y}$ if an index i belongs to the good or to the bad locations. Observe that decoding an instance is equivalent to finding out $I(\overrightarrow{y})$. The hardness assumption is motivated by the fact that for certain ranges of parameters, no efficient decoding algorithms are known. Based on this hardness assumption, Kiayias and Yung [24] constructed several cryptographic primitives, amongst them a stateful symmetric encryption scheme which encrypts messages into a PRP instances. The secret key is the position of the good locations. Given this knowledge, reconstructing the message means to interpolate a

polynomial over the good locations, but without the knowledge, this task is equivalent to decode a codeword.

In this paper, we adopt the Kiayias-Yung-scheme [24] to design an algebraically homomorphic encryption scheme. Like in [24], we consider an encryption scheme where the ciphertexts are PRP instances $\overrightarrow{y}$ and where the secret key is the location of the error-free codeword entries. But in contrast to [24], where the error locations alter from encryption to encryption, the positions of the error free entries remain the same for all encryptions (and in fact depicts the secret key). The reason for this design choice is motivated by the following observation:

Proposition 1. *Let $\overrightarrow{y}, \overrightarrow{y}' \in \mathcal{PR}_{\overrightarrow{x},k,t}$ be two PRP instances with $I := I(\overrightarrow{y}) = I(\overrightarrow{y}')$. Let $\overrightarrow{y}^+ := \overrightarrow{y} + \overrightarrow{y}'$ where "+" here denotes the usual vector addition. Then, it holds that $\overrightarrow{y}^+ \in \mathcal{PR}_{\overrightarrow{x},k,t}$ with $I(\overrightarrow{y}^+) = I$ and $p_{\overrightarrow{y}^+} = p_{\overrightarrow{y}} + p_{\overrightarrow{y}'}$.*

Similarly, let $\overrightarrow{y}^\bullet := \overrightarrow{y} \bullet \overrightarrow{y}'$ denote the componentwise product of $\overrightarrow{y}$ and $\overrightarrow{y}'$. If $\deg(p_{\overrightarrow{y}}) + \deg(p_{\overrightarrow{y}'}) < k$, then $\overrightarrow{y}^\bullet$ is an instance of $\mathcal{PR}_{\overrightarrow{x},k,t}$ as well with $I(\overrightarrow{y}^\bullet) = I$ and $p_{\overrightarrow{y}^\bullet} = p_{\overrightarrow{y}} \cdot p_{\overrightarrow{y}'}$.

Proof. Let y_i denote the entries of $\overrightarrow{y}$ and y'_i the entries of $\overrightarrow{y}'$. We set $p^+(x) := p_{\overrightarrow{y}}(x) + p_{\overrightarrow{y}'}(x)$. Observe that $\deg(p^+) < k$ as $\deg(p_{\overrightarrow{y}}) < k$ and $\deg(p_{\overrightarrow{y}'}) < k$. By assumption it holds that $y_i = p_{\overrightarrow{y}}(x_i)$ and $y'_i = p_{\overrightarrow{y}'}(x_i)$ and for all $i \in I$ and y_i, y'_i being some random values from $\mathbb{F}$ for $i \notin I$. This implies that $y_i^+ = p_{\overrightarrow{y}}(x_i) + p_{\overrightarrow{y}'}(x_i) = p^+(x_i)$ for all $i \in I$ and $y_i^+ = y_i + y'_i$ being some random value in $\mathbb{F}$ for all $i \notin I$. Hence $\overrightarrow{y}^+ \in \mathcal{PR}_{\overrightarrow{x},k,t}$ with solution $(I; p^+(x))$.

The proof for the second claim is similar. We define the polynomial $p^\bullet(x) := p_{\overrightarrow{y}}(x) \cdot p_{\overrightarrow{y}'}(x)$. It holds for all $i \in I$ that the entries $y_i^\bullet$ of $\overrightarrow{y}^\bullet$ fulfill the following equation: $y_i^\bullet = y_i \cdot y'_i = p_{\overrightarrow{y}}(x_i) \cdot p_{\overrightarrow{y}'}(x_i) = p^\bullet(x_i)$. The other entries are the product of random values, hence being random as well. The fact that $\deg(p^\bullet(x)) = \deg(p_{\overrightarrow{y}}) + \deg(p_{\overrightarrow{y}'}) < k$ holds by assumption concludes the proof. □

The property that $\overrightarrow{y}^+ \in \mathcal{PR}_{\overrightarrow{x},k,t}$ with $I(\overrightarrow{y}^+) = I$ guarantees that the addition of two ciphertexts is again a valid ciphertext. The fact that $p_{\overrightarrow{y}^+} = p_{\overrightarrow{y}} + p_{\overrightarrow{y}'}$ ensures the additive homomorphic property of the scheme. Likewise does the second claim implies the multiplicative homomorphic property of our scheme.

As in the Kiayias-Yung-scheme, recovering the plaintext from *one* ciphertext without knowing the secret key is equivalent to decoding a Reed-Solomon code. The difference is that recovering the plaintexts from *several* ciphertexts without knowing the secret key is equivalent to decoding *several* Reed-Solomon codes where the errors are always *at the same locations*. This is a special case of Reed-Solomon codes which belongs to the class of interleaved Reed-Solomon codes. As one might expect, decoding this type of codewords is easier than for the normal case. In fact, there exist several algorithms [4, 10, 7] which are explicitly dedicated to this class. Their efficiency increases with the number of given codewords. Hence, we integrate the number of instances into the problem description and into the hardness assumption. Adopting the terminology from [10], we term our problem *Synchronized Polynomial Reconstruction Problem*:

Definition 4. ***[Synchronized Polynomial Reconstruction Problem (SPRP)]*** *Let $\mathbb{F}$ be an arbitrary field and $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}^n$ be a vector of length n with pairwise distinct entries. Given three positive integer values k, t, and r and a sequence of vectors $\overrightarrow{Y} = (\overrightarrow{y_1}, \ldots, \overrightarrow{y_r})$ with $\overrightarrow{y_\ell} = (y_{\ell,1}, \ldots, y_{\ell,n}) \in \mathbb{F}^n$ for each $\ell \in [r]$, output a sequence $(p_{\overrightarrow{y_1}}, \ldots, p_{\overrightarrow{y_r}}; I)$ such that*

- *for all $\ell \in [m]$ it holds that $p_{\overrightarrow{y_\ell}} \in \mathbb{F}[x]$ is a polynomial over $\mathbb{F}$ with $\deg(p_{\overrightarrow{y_\ell}}) < k$ (the solution polynomials),*

– $I \subseteq [n]$ *is the subset of distinct indices* i *with* $|I| = t$ *(the solution index, being the indices of the error-free entries), and*
– $p_{\overrightarrow{y_\ell}}(x_i) = y_{\ell,i}$ *for all* $i \in I$*.*

The PRP sampler from Definition 2 can be easily adapted to sample SPRP instances:

Definition 5. ***[Sampling SPRP-instances]*** *Let* $\overrightarrow{x}, k, t, r$ *be parameters as specified in Definition 4. We define* $\mathcal{SPR}_{\overrightarrow{x},k,t,r} \subset (\mathcal{PR}_{\overrightarrow{x},k,t})^r$ *to be the set of* r*-tuples of PRP instances* $\overrightarrow{Y} = (\overrightarrow{y_1}, \ldots, \overrightarrow{y_r})$ *such that for each* $\ell \in [r]$*,* $\overrightarrow{y_\ell} \in \mathcal{PR}_{\overrightarrow{x},k,t}$ *and* $I(\overrightarrow{y_\ell}) = I$ *for some* $I \subset [n]$ *of size* t*.*

The following sampler $\tilde{\mathcal{S}}$ *is an adaption from the sampler* $\mathcal{S}$ *from Definition 2 and samples instances* $\overrightarrow{Y}$ *of* $\mathcal{SPR}_{\overrightarrow{x},k,t,r}$ *:* $\tilde{\mathcal{S}}$ *on input* $(\overrightarrow{x}, k, t, r)$ *samples* $I \subset [n]$ *and* r *polynomials* $p_{\overrightarrow{y_1}}, \ldots, p_{\overrightarrow{y_r}} \in \mathbb{F}[x]$ *with* $\deg(p_{\overrightarrow{y_\ell}}) < k$ *for all* ℓ*. It then sets* $y_{\ell,i} := p_{\overrightarrow{y_\ell}}(x_i)$ *for all* $i \in I$*, whereas for all* $i \notin I$ *it samples* $y_{\ell,i}$ *at random from the set* $\mathbb{F} \setminus \{p_{\overrightarrow{y_\ell}}(x_i)\}$*.* $\tilde{\mathcal{S}}$ *terminates by returning* $\overrightarrow{Y} = (\overrightarrow{y_1}, \ldots, \overrightarrow{y_\ell})$ *where* $\overrightarrow{y_\ell} = (y_{\ell,1}, \ldots, y_{\ell,n})$*. We denote the induced distribution on* $(\mathbb{F}^n)^r$ *by* $\tilde{\mathcal{D}}_{\overrightarrow{x},k,t,r}$*.*

Analogously, we define the DSPRP assumption from the DPRP assumption:

Definition 6. ***[Decisional-SPRP*** DSPRP$[\overrightarrow{x}, k, t, r]$*]* *Let the samplers* $\tilde{\mathcal{S}}^{\text{good}}$ *and* $\tilde{\mathcal{S}}^{\text{bad}}$ *be defined analogously from* $\tilde{\mathcal{S}}$ *like* $\mathcal{S}^{\text{good}}$ *and* $\mathcal{S}^{\text{bad}}$ *from* $\mathcal{S}$ *in Definition 3. For any probabilistic polynomial-time algorithm* $\mathcal{A}$*, we define:*

$$Adv^{DSPRP,\mathcal{A}}_{\overrightarrow{x},k,t,r}(s) = \left| Pr[\mathcal{A}(\tilde{\mathcal{S}}^{\text{good}}(\overrightarrow{x}, k, t, r)) = 1] - Pr[\mathcal{A}(\tilde{\mathcal{S}}^{\text{bad}}(\overrightarrow{x}, k, t, r)) = 1] \right|. \tag{2}$$

The DSPRP assumption is that $Adv^{DSPRP}_{\overrightarrow{x},k,t,r}(s) = \max_{\mathcal{A}} Adv^{DSPRP,\mathcal{A}}_{\overrightarrow{x},k,t,r}(s) = \text{negl}(s)$*.*

Observe that although dedicated algorithms exist (e.g., [10]) which solve the SPRP problem, there are (similar to the PRP problem) parameter ranges for which no efficient algorithms are known. Hence, the current state of knowledge is that the DSPRP assumptions holds for certain parameter choices.[1] More on parameter selection will be given in Section 5.

3 The Encryption Scheme

In this section, we formally describe the encryption scheme. In a nutshell, it encodes plaintexts, which are vectors over $\mathbb{F}$, into SPRP-instances where the index set I is the secret key. The scheme is composed of three algorithms: *Setup*, *Encrypt*, and *Decrypt*.

***Setup*:** The input are three positive integer values s, r, and μ where the first denotes the security parameter, the second the expected total number of encryptions[2], and the third the number of supported multiplications.

The *Setup* algorithm chooses integer values n, k, t such that $\mu \cdot k < t < n$ and such that the conditions in Section 5 are met. Next, it selects two vectors $\overrightarrow{x} = (x_1, \ldots, x_n) \in \mathbb{F}^n$ and

[1] Observe that this is similar to, for example, the factorization problem where the parameters are chosen according to best currently known algorithms.

[2] This means an upper bound on the value on how many messages are going to be encrypted. It does not include the number of possible combinations of existing ciphertexts.

$\overrightarrow{z} = (z_1, \ldots, z_{\lfloor k/2 \rfloor}) \in \mathbb{F}^{\lfloor k/2 \rfloor}$ where all entries are pairwise distinct[3] and an index set $I \subset [n]$ of size t.[4] *Setup* outputs $\overrightarrow{x}, k, t$ as public parameters and I as secret key.

***Encrypt*:** *Encrypt* transforms a plaintext $\overrightarrow{m} \in \mathbb{F}^{\lfloor k/2 \rfloor}$ into a PR instance $\overrightarrow{c} \in \mathcal{PR}_{\overrightarrow{x},\mu\cdot k,t}$ with $I(\overrightarrow{c}) = I$. Given $\overrightarrow{m} \in \mathbb{F}^{\lfloor k/2 \rfloor}$ and the secret key I, the algorithm first selects a random polynomial $p(x) \in \mathbb{F}[x]$ of degree $\leq k$ such that $p(\overrightarrow{z}) = \overrightarrow{m}$. The random choice of $p(x)$ will yields a randomized encryption. Next, it generates a vector $\overrightarrow{c} = (c_1, \ldots, c_n) \in \mathbb{F}^n$ as follows. For each $i \in I$, it sets $c_i := p(x_i)$, and for each $j \notin I$, it selects c_j uniformly random from $\mathbb{F} \setminus \{p(x_j)\}$. Obviously, this yields a PR instance $\overrightarrow{c} \in \mathcal{PR}_{\overrightarrow{x},\mu\cdot k,t}$ with solution $(p(x); I)$ (see also the definition of the analog PR sampler in Definition 2). The ciphertext is the pair $(\overrightarrow{c}, 1)$ where the first entry is, in principle, an erroneous codeword that encodes the plaintext $\overrightarrow{m}$ while the second entry, the integer, is a counter to keep track of the number of multiplications.

***Decrypt*:** *Decrypt* gets as input the secret key I and a pair $(\overrightarrow{c}, ctr)$ with $\overrightarrow{c} = (c_1, \ldots, c_n) \in \mathbb{F}^n$ and $m \leq \mu$. In a nutshell, it simply decodes the codeword $\overrightarrow{c}$ using the knowledge of the error-free locations, being the set I. More precisely, it interpolates $p_{\overrightarrow{c}}$ based on the knowledge that $p_{\overrightarrow{c}}(x_i) = c_i$, $i \in I$, and outputs $p_{\overrightarrow{c}}(\overrightarrow{z})$.

As the scheme is algebraically homomorphic, there exist two additional algorithms *Add* and *Mult* to compute the addition and multiplication of encryptions, respectively:

***Add*:** This procedure takes two ciphertexts $(\overrightarrow{c}, ctr)$ and $(\overrightarrow{c}', ctr')$ and produces an encryption of the sum of the plaintexts from the two input ciphertexts via

$$(\overrightarrow{c}^+, ctr^+) := (\overrightarrow{c} + \overrightarrow{c}', \max(ctr, ctr')) \tag{3}$$

where "+" denotes the usual vector addition.

***Mult*:** This procedure get as input two ciphertexts $(\overrightarrow{c}, ctr)$ and $(\overrightarrow{c}', ctr')$ with $ctr + ctr' \leq \mu$ and generates an encryption of the product of the plaintexts from the two input ciphertexts by

$$(\overrightarrow{c}^\bullet, ctr^\bullet) := (\overrightarrow{c} \bullet \overrightarrow{c}', ctr + ctr'). \tag{4}$$

Here, "$\bullet$" is the componentwise vector product as explained in Section 2.

Theorem 1. *The described scheme is correct and is algebraically homomorphic.*

Proof. To show the correctness, we have to prove that the decryption of an encrypted plaintext yields the same plaintext again. Let a ciphertext $(\overrightarrow{c}, ctr)$ be given where $\overrightarrow{c} \in \mathcal{PR}_{\overrightarrow{x},\mu\cdot k,t}$ with solution $(p_{\overrightarrow{c}}; I)$ and let $\overrightarrow{m}$ be the underlying plaintext. By definition, it holds that $p_{\overrightarrow{c}}(\overrightarrow{z}) = \overrightarrow{m}$ and that $c_i = p_{\overrightarrow{c}}(x_i)$ for all $i \in I$. We make now use of the following claim which will be proven at the end.

Claim. It holds for any ciphertext $(\overrightarrow{c}, ctr)$ that $\deg(p_{\overrightarrow{c}}) \leq ctr \cdot k \leq \mu \cdot k$.

The claim implies that $|I| = t > \mu \cdot k \geq ctr \cdot k \geq \deg(p_{\overrightarrow{c}})$. Hence, $p_{\overrightarrow{c}}$ is uniquely determined by the pairs $\{(x_i, c_i)\}_{i \in I}$. Therefore, the decryption algorithm recovers $p_{\overrightarrow{c}}$ and in particular outputs $p_{\overrightarrow{c}}(\overrightarrow{z}) = \overrightarrow{m}$.

[3] In a nutshell, the value $\lfloor k/2 \rfloor$ is chosen to ensure one degree of freedom per plaintext entry for randomization. Hence, the plaintext length should be at most the half of the degree k.

[4] The current state of knowledge is that the hardness of the DSPRP does not depend on the choices of $\overrightarrow{x}, \overrightarrow{z}, I$ if I is unknown and uniformly chosen. In the case of new insights, this part of *Setup* has to be changed accordingly.

The homomorphic properties are an immediate consequence of Proposition 1. We show only the homomorphism regarding the multiplication; the additive homomorphic property can be proved analogously. Consider two encryptions $(\overrightarrow{c}, ctr)$ and $(\overrightarrow{c}', ctr')$ of plaintexts $\overrightarrow{m}$ and $\overrightarrow{m}'$, respectively. By definition of the encryption scheme, the solution of the instance $\overrightarrow{c}$ is $(p_{\overrightarrow{c}}, I)$ with $p_{\overrightarrow{c}}(\overrightarrow{z}) = \overrightarrow{m}$ and the solution of the PR instance $\overrightarrow{c}'$ is $(p_{\overrightarrow{c}'}, I)$ with $p_{\overrightarrow{c}'}(\overrightarrow{z}) = \overrightarrow{m}'$. Let $(\overrightarrow{c}^{\bullet}, ctr^{\bullet})$ be the output of $Mult((\overrightarrow{c}, ctr), (\overrightarrow{c}', ctr'))$. Observe that $\overrightarrow{c}^{\bullet}$ is computed exactly as $\overrightarrow{y}^{\bullet}$ in Proposition 1. It holds by assumption and the claim that $\mu \cdot k \geq (m + m') \cdot k \geq \deg(p_{\overrightarrow{c}}) + \deg(p_{\overrightarrow{c}'})$. Hence, the prerequisites of Proposition 1 are fulfilled and it follows that $\overrightarrow{c}^{\bullet}$ is an instance in $\mathcal{PR}_{\overrightarrow{x},\mu\cdot k,t}$ with solution $(p_{\overrightarrow{c}^{\bullet}} = p_{\overrightarrow{c}} \cdot p_{\overrightarrow{c}'}; I)$. In particular, recovering $p_{\overrightarrow{c}^{\bullet}}$ from $\overrightarrow{c}^{\bullet}$ and I and evaluating it at $\overrightarrow{z}$ yields $p_{\overrightarrow{c}^{\bullet}}(\overrightarrow{z}) = (p_{\overrightarrow{c}} \cdot p_{\overrightarrow{c}'})(\overrightarrow{z}) = p_{\overrightarrow{c}}(\overrightarrow{z}) \bullet p_{\overrightarrow{c}'}(\overrightarrow{z}) = \overrightarrow{m} \bullet \overrightarrow{m}'$.

Observe that, unlike to case of direct encryption, it might happen by coincidence that $c_i^{\bullet} = p_{\overrightarrow{c}^{\bullet}}(x_i)$ for some $i \notin I$, or, in terms of the coding theory, that the noise cancels out at some locations. Obviously, this has no impact on the correctness of the encryption as the good locations are not affected. However, it might make the decryption of this particular ciphertext easier. But as we propose in the parameter selection (see Section 5) to choose the field $\mathbb{F}$ such that $1/|\mathbb{F}| = \mathrm{negl}(s)$ whereas $n - t$ is polynomial in s, we expect this case to occur only with negligible probability.

It remains to prove the claim. We prove it by induction. For direct encryptions, that is outputs of the algorithm *Encrypt*, the claim holds trivially by definition. Now let $(\overrightarrow{c}, ctr)$ and $(\overrightarrow{c}', ctr')$ be two ciphertexts for which the claim holds, that is $\deg(p_{\overrightarrow{c}}) \leq ctr \cdot k \leq \mu \cdot k$ and $\deg(p_{\overrightarrow{c}'}) \leq ctr' \cdot k \leq \mu \cdot k$. For the addition procedure *Add*, one sees easily that

$$\mu \cdot k \geq \max(ctr, ctr') \cdot k \geq \deg(p_{\overrightarrow{c}}) + \deg(p_{\overrightarrow{c}'})) = \deg(p_{\overrightarrow{c}^{+}}). \tag{5}$$

Similarly, under the condition of $ctr + ctr' \leq \mu$, it holds that

$$\mu \cdot k \geq \underbrace{(ctr + ctr')}_{=ctr^{\bullet}} \cdot k \geq \deg(p_{\overrightarrow{c}}) \cdot \deg(p_{\overrightarrow{c}'})) = \deg(p_{\overrightarrow{c}^{\bullet}}). \tag{6}$$

□

Observe that encryption is in principle evaluating a polynomial and replacing some outputs while decryption is simple polynomial interpolation. Both operations can be done by computing a matrix-vector product (with the matrix being the Vandermonde matrix or its inverse). Furthermore, all statements and computations remain valid if one replaces $\mathbb{F}$ by an extension field of $\mathbb{F}$. The only thing one has to do is to embed the entries into the extension field which can be done without knowing the key.

4 Security

In this section, we prove that the encryption system is semantically secure for some parameters $\overrightarrow{x}, k, I, r, \mu$ under the DSPRP assumption. This is done by the usual reduction approach. We prove that any probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$ which breaks the semantic security of our scheme for some parameters $\overrightarrow{x}, k, I, r$ with non-negligible advantage can be transformed into a PPT algorithm $\mathcal{A}'$ that decides the DSPRP$[\overrightarrow{x}, \lfloor k/2 \rfloor, I, r]$ with non-negligible advantage. Hence, if the DSPRP assumption is true, the existence of such an attacker would lead to a contradiction. In consequence, no such attacker can exist which shows the semantic security.

4.1 Semantic Security

Semantic security requires that it should be infeasible for an attacker to gain for a given ciphertext any partial information about the underlying plaintext, even if the set of possible plaintexts is reduced to two different messages which have been chosen by the attacker before. The formal definition of semantic security is covered by the following game-based approach. In this game, two players are involved: an attacker $\mathcal{A}$ and an encryption oracle $\mathcal{O}^{\text{encr.}}$. The game is divided into two query phases, a challenge phase inbetween, and a decision phase at the end (see also [24]):

First query phase: The attacker $\mathcal{A}$ queries a number of times (where the number is polynomial in the security parameter s) the encryption oracle $\mathcal{O}^{\text{encr.}}$ with adaptively chosen plaintexts which are encrypted by $\mathcal{O}^{\text{encr.}}$ and returned to $\mathcal{A}$.

Challenge phase: $\mathcal{A}$ chooses two different plaintexts $\overrightarrow{m}_0 \neq \overrightarrow{m}_1$ and gives them to the oracle. $\mathcal{O}^{\text{encr.}}$ selects uniformly random $b \in \{0,1\}$, creates an encryption $\overrightarrow{c}$ of $\overrightarrow{m}_b$, and returns the result to $\mathcal{A}$.

Second query phase: The second query phase is like the first query phase. That is, the attacker $\mathcal{A}$ adaptively asks from $\mathcal{O}^{\text{encr.}}$ a number of encryptions.

Decision phase: $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for b, that is she assumes that $\overrightarrow{c}$ is the encryption of $\overrightarrow{m}_{b'}$. $\mathcal{A}$ wins if $b = b'$, that is if she guessed correctly.

A trivial strategy of $\mathcal{A}$ would be to randomly choose $b' \in \{0,1\}$, independent of the previously exchanged messages. Obviously, such an attacker would succeed with probability $1/2$. Therefore, an attacker $\mathcal{A}$ is called *successful* if the difference between the success probability, that is the probability of $b = b'$, and $1/2$ is non-negligible. We call this value the *advantage* $Adv^{\mathcal{A}}$ of $\mathcal{A}$. More formally, let $\mathcal{O}_b^{\text{encr.}}$ be the encryption oracle that always returns the encryption of $\overrightarrow{m}_b$ in the challenge phase. A scheme is semantically secure if it holds for any breaking adversary is a PPT $\mathcal{A}$ that

$$|Prob_{b\in\{0,1\}}[\mathcal{A}^{\mathcal{O}_b^{\text{encr.}}}(1^s) = b] - \frac{1}{2}| = \text{negl}(s), \tag{7}$$

where the probability is taken over all internal coin-tosses of $\mathcal{O}_b^{\text{encr.}}$ and $\mathcal{A}$. Informally speaking, no PPT adversary $\mathcal{A}$ is has a significant better success probability than the trivial attacker described above.

4.2 Proof of security

In this section, we prove that our encryption scheme is semantically secure for parameters $\overrightarrow{x}, k, t, r$ under the DSPRP$[\overrightarrow{x}, \lfloor k/2 \rfloor, t, r]$ assumption. For the proof of security, we make use of the following theorem on the pseudorandomness of sampled instances:

Theorem 2. *For any distinguisher $\mathcal{A}$ between the distributions $\tilde{\mathcal{D}} := \tilde{\mathcal{D}}_{\overrightarrow{x},\lfloor k/2 \rfloor,t,r}$ (induced by the sampler $\tilde{\mathcal{S}}$ from Definition 5) and the uniform distribution $\mathcal{U}$ on $(\mathbb{F}^n)^r$, it holds that*

$$|Pr[\mathcal{A}(\overrightarrow{Y}) = 1 | \overrightarrow{Y} \leftarrow \tilde{\mathcal{D}}] - Pr[\mathcal{A}(\overrightarrow{Y}) = 1 | \overrightarrow{Y} \leftarrow \mathcal{U}]| \leq \frac{t \cdot r \cdot (n - t + 3)}{|\mathbb{F}|} + 9t \cdot Adv^{DSPRP}_{\overrightarrow{x}',\lfloor k/2 \rfloor,t,r} \tag{8}$$

where $\overrightarrow{x}' \in \mathbb{F}^{n-1}$ is derived from $\overrightarrow{x}$ by removing one coordinate.

The Theorem is an adaption of Theorem 3.4 given in [24] and the proof is very similar. However, there are some subtle differences due to the fact that we are dealing with a set of synchronized PRP instances here. For this reason and for the sake of completeness, we give the proof in Appendix A.

Theorem 3. *The encryption scheme from Section 3 is semantically secure for parameters* $\overrightarrow{x}, k, t, r$ *if the* DSPRP$[\overrightarrow{x}', \lfloor k/2 \rfloor, t, r]$ *assumption holds.*

Proof. Let $\mathcal{A}$ be a PPT algorithm that breaks the semantic security for parameters $\overrightarrow{x}, k, t, r$ with at most r queries (including the challenge). Let $\overrightarrow{Y} = (\overrightarrow{y_1}, \ldots, \overrightarrow{y_r}) \in (\mathbb{F}^n)^r$ be given which is either distributed according to $\tilde{\mathcal{D}}_{\overrightarrow{x}, \lfloor k/2 \rfloor, t, r}$ or according to $\mathcal{U}$. We show how to transform $\mathcal{A}$ directly into a distinguisher $\mathcal{A}'$ which distinguishes between these two distributions. If the DSPRP$[\overrightarrow{x}', \lfloor k/2 \rfloor, t, r]$ assumption holds, then it follows from equation 8 from Theorem 2 that the advantage of $\mathcal{A}'$ is negligible. Consequently, this must be true for $\mathcal{A}$ as well which proves the semantic security.

$\mathcal{A}'$ uses $\mathcal{A}$ to solve the distinguishing problem. For this purpose, it has to simulate the encryption oracle $\mathcal{O}^{\text{encr.}}$ for $\mathcal{A}$. This is done as follows. For each encryption query from $\mathcal{A}$, $\mathcal{A}'$ picks one of the vectors $\overrightarrow{y_\ell} = (y_{\ell,1}, \ldots, y_{\ell,n})$ which has not used before. To keep the description simple, we assume that the PR instances $\overrightarrow{y_1}, \ldots, \overrightarrow{y_r}$ are used in the same order as their indices. That is, the response $\overrightarrow{c_1}$ to the first query will be computed from $\overrightarrow{y_1}$, and so on.

On input $\overrightarrow{m}_\ell$, $\mathcal{A}'$ chooses a polynomial $p_\ell \in \mathbb{F}[x]$ of degree $< k$ such that $p_\ell(\overrightarrow{z}) = \overrightarrow{m}_\ell$ and computes

$$c_{\ell,i} := p_\ell(x_i) + y_{\ell,i} \cdot \prod_{j=1}^{\lfloor k/2 \rfloor} (x_i - z_j). \tag{9}$$

It returns $\overrightarrow{c_\ell} = (c_{\ell,1}, \ldots, c_{\ell,n})$ to $\mathcal{A}$. For the challenge request, $\mathcal{A}'$ picks uniform at random one of the two challenge plaintexts and encrypts it in the same manner as described above. We denote by τ the transformation $(\overrightarrow{y_\ell}, p_\ell) \mapsto \overrightarrow{c}_\ell$ defined by equation 9.

Assume now that $\overrightarrow{Y}$ is distributed according to $\mathcal{U}$. This means that all values $y_{\ell,i}$ are chosen uniformly random from $\mathbb{F}$ which implies that the values $c_{\ell,i}$ from equation 9 are uniformly random as well. In particular, the responses from $\mathcal{A}'$ to $\mathcal{A}$ are independent of $\mathcal{A}'$'s choice of b and thus $\mathcal{A}$ gains no information on the value of b which shows that its advantage is negligible in this case.

Now assume that $\overrightarrow{Y}$ is distributed according to $\tilde{\mathcal{D}}_{\overrightarrow{x}, \lfloor k/2 \rfloor, t, r}$. That is the vectors $\overrightarrow{y_\ell}$ are PR instances with a common solution index set I. Let $\ell \in [r]$ be arbitrary, $p_{\overrightarrow{y_\ell}}$ be the solution polynomial of $\overrightarrow{y_\ell}$, and $\overrightarrow{c_\ell} = (c_{\ell,1}, \ldots, c_{\ell,n})$ denote the created response to the ℓ-th query. By assumption, it holds that

$$y_{\ell,i} = \begin{cases} \in_R \mathbb{F} \setminus \{p_{\overrightarrow{y_\ell}}(x_i)\} & , i \notin I \\ p_{\overrightarrow{y_\ell}}(x_i) & , i \in I \end{cases} \tag{10}$$

We define $p_{\overrightarrow{c_\ell}}(x) := p_\ell(x) + p_{\overrightarrow{y_\ell}}(x) \cdot \prod_{j=1}^{\lfloor k/2 \rfloor}(x - z_j)$. Putting equations 9 and 10 together yields

$$c_{\ell,i} = \begin{cases} \in_R \mathbb{F} \setminus \{p_{\overrightarrow{c_\ell}}(x_i)\} & , i \notin I \\ p_{\overrightarrow{c_\ell}}(x_i) & , i \in I \end{cases} \tag{11}$$

Observe that the second part of $p_{\overrightarrow{c_\ell}}(x)$ vanishes on $\overrightarrow{z}$. Hence $p_{\overrightarrow{c_\ell}}(\overrightarrow{z}) = p_\ell(\overrightarrow{z}) = \overrightarrow{m}$ and $\overrightarrow{c_\ell}$ is a valid encryption of $\overrightarrow{m_\ell}$.

Claim: For any given plaintext $\overrightarrow{m}$ and any index set I, the transformation τ is a surjection and each image has the same number of preimages.

Hence, this procedure can yield any possible encryption of a given plaintext. Therefore, $\mathcal{A}$'s view is that it received valid encryptions and any encryption for a chosen plaintext is possible. Hence, it observes no difference to communicating with an encryption oracle $\mathcal{O}^{\text{encr.}}$. In particular, $\mathcal{A}$ has by assumption a non-negligible advantage to guess b correctly.

The remainder of the proof follows the usual arguments. $\mathcal{A}'$ runs $\mathcal{A}$ sufficiently often to estimate $\mathcal{A}$'s advantage with sufficient precision. If the advantage is negligible, $\mathcal{A}'$ assumes that $\overrightarrow{Y}$ was uniformly sampled from $(\mathbb{F}^n)^r$. Otherwise, it assumes that $\overrightarrow{Y}$ was sampled by $\tilde{\mathcal{S}}$. □

Proof. (Proof of the claim in the proof of Theorem 3) We have to show for any $\overrightarrow{m}$ and any index set I that the mapping τ is surjective and that the sets of preimages have all the same size. The correctness of the mapping, e.g., that the images are indeed encryptions of $\overrightarrow{m}$, has been shown already. Let $\overrightarrow{c} = (c_1, \ldots, c_n)$ be an arbitrary encryption of $\overrightarrow{m}$. That is, $\overrightarrow{c} \in \mathcal{PR}_{\overrightarrow{x},\mu\cdot k,t}$ with solution $(p_{\overrightarrow{c}}; I)$ such that $p_{\overrightarrow{c}}(\overrightarrow{z}) = \overrightarrow{m}$. By assumption, it holds that

$$c_i = \begin{cases} \in_R \mathbb{F} \setminus \{p_{\overrightarrow{c}}(x_i)\} & , i \notin I \\ p_{\overrightarrow{c}}(x_i) & , i \in I \end{cases} . \tag{12}$$

Now, let $p(x)$ be any polynomial from $\mathbb{F}[x]$ of degree $< k$ such that $p(\overrightarrow{z}) = \overrightarrow{m}$. Then, $q(x) := p_{\overrightarrow{c}}(x) - p(x)$ is a polynomial of degree $< k$ which maps each value in $\overrightarrow{z}$ to zero. Hence, $q(x)$ can be rewritten as $q(x) = q'(x) \cdot \prod_{j=1}^{\lfloor k/2 \rfloor}(x - z_j)$ where $q'(x)$ is of degree $< \lfloor k/2 \rfloor$.

Next, we define for each $i \in [n]$ the value $y_i := (c_i - p(x_i))/\prod_{j=1}^{\lfloor k/2 \rfloor}(x_i - z_j)$. Observe that the values x_i and z_j are all pairwise distinct by assumption, so there is no risk to divide by zero. Together with equation 12, this implies for each $i \in [n]$:

$$y_i = \begin{cases} \in_R \mathbb{F} \setminus \{q'(x_i)\} & , i \notin I \\ q'(x_i) & , i \in I \end{cases} . \tag{13}$$

This shows that $\overrightarrow{y} := (y_1, \ldots, y_n) \in \mathcal{PR}_{\overrightarrow{x},k,t}$ with solution $(q'(x); I)$. Furthermore, $\tau(p(x), \overrightarrow{y}) = \overrightarrow{c}$. As $\overrightarrow{c}$ is an arbitrary encryption of $\overrightarrow{m}$, this shows the surjectivity of τ.

Regarding the number of preimages, observe that $p(x)$ was arbitrary and that $\overrightarrow{y}$ was uniquely determined by $\overrightarrow{c}$ and $p(x)$. Hence, there exists for any ciphertext $\overrightarrow{c}$ and for every polynomial $p(x)$ with the above explained properties exactly one PR instance $\overrightarrow{y}$ such that $\tau(p(x), \overrightarrow{y}) = \overrightarrow{c}$. This shows that the number of preimages is the same for each ciphertext $\overrightarrow{c}$. □

5 Parameter selection

Following the approach of [24], we propose to select parameters which prevent the application of straight-forward attacks or dedicated decoding algorithms. We will consider the values $n' := |\overrightarrow{x}'| = n' - 1$, $\lfloor k/2 \rfloor$, and $t := |I|$ as functions in the security parameter s for given values r, the number of encryptions, and μ, the number of multiplications. As we are interested in the size of the ciphertext only, we abstract from the choice of $\overrightarrow{x}$ and I and consider only the integer values n', k, t. Observe that $t \geq \mu \cdot k$ is necessary to enable unique decrypting and that the decoding problem gets easier, the higher t (for fixed n'). Hence, we set $t := \mu \cdot k$.

The straightforward brute-force algorithm for solving DSPRP$[n', \lfloor k/2 \rfloor, t]$ is either by trying all possibilities subset of $\lfloor k/2 \rfloor$ to interpolate a polynomial or by guessing the $n' - t$ erroneous locations. These approaches have a complexity proportional to $\min\{\binom{n'}{\lfloor k/2 \rfloor}, \binom{n'}{t}\}$. Moreover, the SPRP instances should withstand the dedicated decoding algorithms for interleaved Reed-Solomon codes. To the best of our knowledge, the most efficient decoding algorithms for this problem are the ones by Coppersmith and Sudan [10] and by Brown, Minder, and Shokrollahi [7]. For both algorithms, parameter ranges are specified within the algorithms work for sure. This poses two

necessary conditions on the parameter choices. These conditions can be transformed into lower bounds for the ration n'/k which marks lower bounds for the ciphertext size n'. The lower bound from [10] is $n'/k \geq \frac{(2\mu-1)^{r+1}}{2}$ and from [7] is $n'/k \geq (r+1) \cdot \mu - \frac{r}{2}$. Observe that the first condition implies an exponential blow-up in the number r of encryptions.

6 Possible extensions

Observe that all arguments given in the scheme description in Section 3 and in the security proof in Section 4.2 hold for any fields. Hence, the scheme securely operates over any field, including non-finite fields like the field of rational numbers, if the DSPRP assumption holds. However, it is an open issue whether the DSPRP assumption is plausible over non-finite fields.

Regarding the huge ciphertext size, notice that it results as a precaution against dedicated decoding algorithms for Reed-Solomon codes. We see no reasons why this should equally hold for other coding schemes as well. In other words, building the scheme upon other coding schemes, e.g., algebraic codes, might lead to more efficient results. Besides, varying the underlying problem is another approach. For example, one could keep the support vectors $\overrightarrow{x}$ and $\overrightarrow{z}$ hidden and treat them as part of the secret key. Without doubt, this makes an attack more difficult which might help to reduce the ciphertext size. Of course, this requires more research.

Our scheme shares with the Kiayias-Yung-scheme [24] the property of intrinsic error tolerance. Assume that the ciphertexts are transmitted over a noisy channel such that some entries change to random error values. Any error that happens at bad locations has actually no effect as only the good locations are taken into account for decryption. In the case that error occur at the good locations, one might use the fact that the sequence of values y_i for $i \in I$ is actually a Reed-Solomon codeword itself of size t. Hence, depending on the ration between k and t, a certain amount of errors can be corrected at the good locations. In that sense allows the proposed scheme to directly combine decryption and error-correcting without the need of additional error-correction codes.

7 Conclusions and Future Work

The existence of efficient and secure algebraically homomorphic encryption schemes is a long standing open question since [33]. Although some proposals exist, none of them are fully satisfactory. As only very little progress in answering this question has been made in the recent years, there is a need for completely novel, yet unexplored approaches. In this paper, we introduce the idea of using coding theory into this subject. Although we do not solve the problem completely, we show that provable secure algebraically homomorphic schemes can be constructed which are suitable for specific classes of applications.

It remains for further research to explore this approach more deeply. Although we picked Reed-Solomon codes for our concrete instantiations, the general approach should be transferable to other coding schemes as well, e.g., algebraic codes. From our point of view, the interesting properties of our scheme (in particular the support for non-finite fields) makes this approach promising for other applications as well. Thus, we see our result as a first step for possibly establishing a new research direction.

References

1. F. Bao. Cryptanalysis of a provable secure additive and multiplicative privacy homomorphism. In *International Workshop on Coding and Cryptography (WCC)*, 2003.
2. J. Benaloh. *Verifiable secret-ballot elections.* PhD thesis, Yale University, New Haven, CT, USA, 1987.
3. C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997.
4. D. Bleichenbacher, A. Kiayias, and M. Yung. Decoding of interleaved reed solomon codes over noisy data. In Jos C. M. Baeten, J. Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *ICALP*, volume 2719 of *Lecture Notes in Computer Science*, pages 97–108. Springer, 2003.
5. D. Boneh, E. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In Joe Killian, editor, *Proceedings of Theory of Cryptography Conference 2005*, volume 3378 of *LNCS*, pages 325–342. Springer, 2005.
6. D. Boneh and R. Lipton. Algorithms for black-box fields and their application to cryptography (extended abstract). In *CRYPTO '96: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, pages 283–297, London, UK, 1996. Springer-Verlag.
7. A. Brown, L. Minder, and A. Shokrollahi. Improved decoding of interleaved ag codes. In Nigel P. Smart, editor, *IMA Int. Conf.*, volume 3796 of *Lecture Notes in Computer Science*, pages 37–46. Springer, 2005.
8. J. Cheon, W. Kim, and H. Nam. Known-plaintext cryptanalysis of the Domingo-Ferrer algebraic privacy homomorphism scheme. *Inf. Process. Lett.*, 97(3):118–123, 2006.
9. J. Cohen and M. Fischer. A robust and verifiable cryptographically secure election scheme (extended abstract). In *FOCS*, pages 372–382. IEEE, 1985.
10. D. Coppersmith and M. Sudan. Reconstructing curves in three (and higher) dimensional space from noisy data. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 136–142, New York, NY, USA, 2003. ACM.
11. R. Cramer, I. Damgaard, and J. Nielsen. Multiparty computation from threshold homomorphic encryption. In *EUROCRYPT '01: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pages 280–299, London, UK, 2001. Springer-Verlag.
12. R. Cramer, M. Franklin, L. Schoenmakers, and M. Yung. Multi-authority secret-ballot elections with linear work. Technical report, Amsterdam, The Netherlands, The Netherlands, 1995.
13. R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. *European Transactions on Telecommunications*, 8(5):481–490, September 1997.
14. I. Damgaard and M. Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *PKC '01: Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography*, pages 119–136, London, UK, 2001. Springer-Verlag.
15. F. Levy dit Vehel, M. Marinari, L. Perret, and C. Traverso. *Gröbner Bases, Coding Theory, and Cryptography*, chapter A Survey on Polly Cracker Systems. RISC Book Series. Springer, Heidelberg, to appear.
16. J. Domingo-Ferrer. A new privacy homomorphism and applications. *Inf. Process. Lett.*, 60(5):277–282, 1996.
17. J. Domingo-Ferrer. A provably secure additive and multiplicative privacy homomorphism. In *ISC '02: Proceedings of the 5th International Conference on Information Security*, pages 471–483, London, UK, 2002. Springer-Verlag.
18. M. Fellows and N. Koblitz. Combinatorial cryptosystems galore! *Contemporary Mathematics*, 168:51–61, 1993.
19. C. Fontaine and F. Galand. A survey of homomorphic encryption for nonspecialists. *EURASIP J. Inf. Secur.*, 2007(1):1–15, 2007.
20. T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.
21. O. Goldreich, R. Rubinfeld, and M. Sudan. Learning polynomials with queries: The highly noisy case. In *FOCS '95: Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS'95)*, page 294, Washington, DC, USA, 1995. IEEE Computer Society.
22. A. Kiayias and M. Yung. Secure games with polynomial expressions. In *ICALP '01: Proceedings of the 28th International Colloquium on Automata, Languages and Programming,*, pages 939–950, London, UK, 2001. Springer-Verlag.
23. A. Kiayias and M. Yung. Directions in polynomial reconstruction based cryptography. *IEICE transactions on fundamentals of electR.ics, communications and computer sciences*, 87(5):978–985, 20040501.
24. A. Kiayias and M. Yung. Cryptographic hardness based on the decoding of reed-solomon codes. Cryptology ePrint Archive, Report 2007/153, 2007. `http://eprint.iacr.org/`.
25. E. Kushilevitz and R. Ostrovsky. Replication is not needed: single database, computationally-private information retrieval. In *FOCS '97: Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS '97)*, page 364, Washington, DC, USA, 1997. IEEE Computer Society.

26. H. Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In C. Laih, editor, *ASIACRYPT*, volume 2894 of *Lecture Notes in Computer Science*, pages 398–415. Springer, 2003.
27. L. Van Ly. Polly two : A new algebraic polynomial-based public-key scheme. *Appl. Algebra Eng. Commun. Comput.*, 17(3-4):267–283, 2006.
28. C. Melchor, P. Gaborit, and J. Herranz. Additive homomorphic encryption with t-operand multiplications. Cryptology ePrint Archive, Report 2008/378, 2008. `http://eprint.iacr.org/`.
29. M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation. In *STOC '99: Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 245–254, New York, NY, USA, 1999. ACM.
30. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, pages 223–238, 1999.
31. D. Rappe. *Homomorphic cryptosystems and their applications.* PhD thesis, University of Dortmund, Germany, 2004.
32. I. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8:300–304, June 1960.
33. R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation*, pages 169–179, 1978.
34. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 26(1):96–99, 1983.
35. T. Sander, A. Young, and M. Yung. Non-interactive cryptocomputing for nc^1. In *FOCS*, pages 554–567, 1999.
36. D. Wagner. Cryptanalysis of an algebraic privacy homomorphism. In C. Boyd and W. Mao, editors, *ISC*, volume 2851 of *Lecture Notes in Computer Science*, pages 234–239. Springer, 2003.

A Proof

In this section, we prove Theorem 2 from Section 4.2. First, we state a result from [24]:

Lemma 1. *Let v_i^b, v_i^g be independent samplable binary random variables for $i \in [n]$ with means μ_i^b and μ_i^g respectively for which it holds:*

- *There exists an $i \in [n]$ such that $|Pr[v_i^g = 1] - Pr[v_i^b = 1]| \geq \alpha$ where α is a non-negligible function in n.*

Then, for all $\epsilon > 0$, there exists a PPT B that returns an i that satisfies $|Pr[v_i^g = 1] - Pr[v_i^b = 1]| \geq \alpha/4$ with probability $1 - \epsilon$. B requires $\mathcal{O}(\alpha^{-2}(log(\epsilon^{-1}) + \log n))$ samples of each of the given random variables.

We are now ready to prove Theorem 2. As already stated, the proof is an adaption of a proof given in [24]. However, it differs in several points and some steps are explained into more detail.

Proof. (Proof of Theorem 2) Let $\mathcal{A}$ be the distinguisher between the distributions $\tilde{\mathcal{D}}_{\overrightarrow{x},\lfloor k/2 \rfloor,t,r}$ and $\mathcal{U}$ with distinguishing probability α, that is

$$\alpha := |Pr[\mathcal{A}(\overrightarrow{Y}) = 1 | \overrightarrow{Y} \leftarrow \tilde{\mathcal{D}}_{\overrightarrow{x},\lfloor k/2 \rfloor,t,r}] - Pr[\mathcal{A}(\overrightarrow{Y}) = 1 | \overrightarrow{Y} \leftarrow \mathcal{U}]|. \tag{14}$$

We assume now that α is not negligible, that is α^{-1} is polynomial in s, and show that this leads to a contradiction.

We define the sampler $\tilde{\mathcal{S}}_i$ to first sample $\overrightarrow{Y} = (\overrightarrow{y_1}, \ldots \overrightarrow{y_r})$ according to $\tilde{\mathcal{S}}$ from Definition 5 and then eventually to give out $\left(i; \overrightarrow{Y}\right)$. Consider the following procedure $\mathcal{A}_1$ that operates on inputs of the form $\left(i, \overrightarrow{Y}\right)$ as follows: it first selects a random permutation π and then overwrites for each PRP instance $\overrightarrow{y_\ell}$ the values $y_{\ell,\pi(1)}, ..., y_{\ell,\pi(i)}$ (for $i \in [n]$ and $\ell \in [r]$) by substituting them with

i random values over $\mathbb{F}$. In this way A_1 produces a "partially randomized" SPRP instance $\overrightarrow{Y}'$. Then $\mathcal{A}_1$ simulates $\mathcal{A}$ on $\overrightarrow{Y}'$. We will denote the operation of $\mathcal{A}_1$ as $\mathcal{A}(R^\pi\left(i;\overrightarrow{Y}\right))$ where R^π is the probabilistic operator that given $\left(i;\overrightarrow{Y}\right)$ randomizes the first (according to π) i locations of the contained PRP instances $\overrightarrow{y_1}\ldots,\overrightarrow{y_r}$. It is immediate that

$$Pr[\mathcal{A}_1(\tilde{\mathcal{S}}_0(\overrightarrow{x},\lfloor k/2\rfloor,t,r))=1]=Pr[\mathcal{A}(\tilde{\mathcal{D}}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}))=1]$$

as well as that

$$Pr[\mathcal{A}_1(\tilde{\mathcal{S}}_n(\overrightarrow{x},\lfloor k/2\rfloor,t,r))=1]=Pr[\mathcal{A}(U)=1].$$

As a result $|Pr[\mathcal{A}_1(\tilde{\mathcal{S}}_0(\overrightarrow{x},\lfloor k/2\rfloor,t,r))=1]-Pr[\mathcal{A}_1(\tilde{\mathcal{S}}_n(\overrightarrow{x},\lfloor k/2\rfloor,t,r))=1]|\geq\alpha$ since

$$|Pr[\mathcal{A}(\tilde{\mathcal{D}}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}))=1]-Pr[\mathcal{A}(U)=1]|\geq\alpha$$

from the statement of the theorem. By employing the triangular inequality we obtain that there exists $i\in[n]$ such that

$$|Pr[\mathcal{A}_1(\tilde{\mathcal{S}}_i(\overrightarrow{x},\lfloor k/2\rfloor,t,r))=1]-Pr[\mathcal{A}_1(\tilde{\mathcal{S}}_{i-1}(\overrightarrow{x},\lfloor k/2\rfloor,t,r))=1]|\geq\alpha/n.$$

Below we will denote by $E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}$ the event $\mathcal{A}(R^\pi(\tilde{\mathcal{S}}_i(\overrightarrow{x},\lfloor k/2\rfloor,t,r)))=1$. Using this notation and the above results we obtain that:

$$\forall\pi\exists i\in[n]s.t.|Pr[E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}]-Pr[E^{i-1,\pi}_{n,\lfloor k/2\rfloor,t,r}]|\geq\alpha' \tag{15}$$

where $\alpha'=\alpha/n$. Next, consider the event Bad_i^π to correspond to the coin tosses of the sampler $\tilde{\mathcal{S}}(\overrightarrow{x},\lfloor k/2\rfloor,t,r)$ that the location $\pi(i)$ is among the bad locations, that is $\pi(i)\notin I$ where I is the index set chosen by $\tilde{\mathcal{S}}$. One sees easily that $Pr[Bad_i^\pi]=\frac{n-t}{n}=1-\frac{t}{n}$. Analogously, We denote by $Good_i^\pi$ the negation of this event, that is the event that $\pi(i)$ is one of the good locations. In the remainder of the proof we will use three claims which will be proven later.

Claim 1. $|Pr[E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}|Bad_i^\pi]-Pr[E^{i-1,\pi}_{n,\lfloor k/2\rfloor,t}|Bad_i^\pi]|\leq r/|\mathbb{F}|$.

Claim 2. $Pr[E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}|Good_i^\pi]=|Pr[E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t-1,r}|Bad_i^\pi]$.

Claim 3. $|Pr[E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t-1,r}|Bad_i^\pi]-|Pr[E^{i-1,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}|Bad_i^\pi]|\leq Adv^{DSPRP}_{\overrightarrow{x}',\lfloor k/2\rfloor,t,r}+3r/|\mathbb{F}|$.

Next we use the fact: if $|Pr[E_1]-Pr[E_2]|\geq p_1$ and $|Pr[E_1|B]-Pr[E_2|B]|\leq p_2$ then it holds that $|Pr[E_1|\neg B]-Pr[E_2|\neg B]|\geq(p_1-p_2\cdot Pr[B])(Pr[\neg B])^{-1}$. In our case, it is that $p_1=\alpha/n$ and $p_2=r/|\mathbb{F}|$ by Claim 1. Putting this together we obtain the following:

$$|Pr[E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}|Good_i^\pi]-Pr[E^{i-1,\pi}_{n,\lfloor k/2\rfloor,t}|Good_i^\pi]|\geq\alpha'' \tag{16}$$

where $\alpha''=\frac{n}{t}(\alpha'-(1-\frac{t}{n})|\cdot\frac{r}{|\mathbb{F}|})=\frac{\alpha}{t}-\frac{r\cdot(n-t)}{|\mathbb{F}|}$ with $t:=|I|$.

For the following computations, we abbreviate $E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}$ to E^i_t, $E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t-1,r}$ to E^i_{t-1}, Bad^π_i to B, and $Good^\pi_i$ to G. By applying the results of claims 2 and 3 to the inequality 16 we obtain the following:

$$\begin{aligned}
|Pr[E^{i-1}_t|B] - Pr[E^{i-1}_t|G]| &= |Pr[E^{i-1}_t|B] - Pr[E^{i-1}_t|G] + \underbrace{PR[E^i_t|G] - PR[E^i_t|G]}_{=0}| \\
&\overset{Claim\ 2}{=} |Pr[E^{i-1}_t|B] - Pr[E^{i-1}_t|G] + PR[E^i_t|G] - PR[E^i_{t-1}|B]| \\
&= |Pr[E^i_t|G] - Pr[E^{i-1}_t|G] - (Pr[E^i_{t-1}|B] - Pr[E^{i-1}_t|B])| \\
&\geq \underbrace{|Pr[E^i_t|G] - Pr[E^{i-1}_t|G]|}_{\geq \frac{\alpha}{t} - \frac{r\cdot(n-t)}{|\mathbb{F}|}} - \underbrace{|(Pr[E^i_{t-1}|B] - Pr[E^{i-1}_t|B])|}_{\leq Adv^{DSPRP}_{\overrightarrow{x}',\lfloor k/2\rfloor,t-1,r} + 3r/|\mathbb{F}|} \\
&\geq \frac{\alpha}{t} - \frac{r\cdot(n-t+3)}{|\mathbb{F}|} - Adv^{DSPRP}_{\overrightarrow{x}',\lfloor k/2\rfloor,t-1,r} =: \alpha'''.
\end{aligned}$$

Using the definition of $E^{i-1,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}$ we rewrite the inequality above as follows:

$$|Pr[A(R^\pi_{i-1}(\tilde{\mathcal{S}}(\overrightarrow{x},\lfloor k/2\rfloor,t,r))) = 1|Bad^\pi_i] - Pr[A(R^\pi_{i-1}(\tilde{\mathcal{S}}(\overrightarrow{x},\lfloor k/2\rfloor,t,r))) = 1|Good^\pi_i]| \geq \alpha''' \quad (17)$$

where i is some index in $[n]$ that while it is unknown, its existence is guaranteed from equation 15. Next we observe that we can simulate the behavior of the sampler $\tilde{\mathcal{S}}$ in the conditional probability spaces Bad^π_i and $Good^\pi_i$. In particular this can be done easily by the samplers $\tilde{\mathcal{S}}^{Bad^\pi_i}$ and $\tilde{\mathcal{S}}^{Good^\pi_i}$ that operate exactly as $\tilde{\mathcal{S}}$ with the exception the selection of the set of indices I that is done as follows: for the case of $\tilde{\mathcal{S}}^{Good^\pi_i}$ a random subset $I \subseteq [n] \setminus \{\pi(i)\}$ is selected that has cardinality $t-1$ and then the element $\pi(i)$ is added to it; on the other hand, for the case of $S^{Bad^\pi_i}$, a random subset $I \subseteq [n] \setminus \{\pi(i)\}$ is selected with cardinality t. Based on this it follows that we can rewrite equation 17 in this way:

$$|Pr[A(R^\pi_{i-1}(\tilde{\mathcal{S}}^{Bad^\pi_i}(\overrightarrow{x},\lfloor k/2\rfloor,t,r))) = 1] - Pr[A(R^\pi_{i-1}(\tilde{\mathcal{S}}^{Good^\pi_i}(\overrightarrow{x},\lfloor k/2\rfloor,t,r))) = 1]| \geq \alpha''' \quad (18)$$

From here on, one can proceed exactly as in the proof of Theorem 3.4 in [24] to show that

$$\alpha \leq t\cdot r\cdot(n-t+3)/|\mathbb{F}| + t\cdot Adv^{DSPRP}_{\overrightarrow{x}',\lfloor k/2\rfloor,t,r} + 8t\cdot Adv^{DSPRP}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}. \quad (19)$$

Observe now that any instance of DSPRP$[\overrightarrow{x}',\lfloor k/2\rfloor,t,r]$ can easily be augmented to an instance of DSPRP$[\overrightarrow{x},\lfloor k/2\rfloor,t,r]$ by inserting a new supporting coordinate in $\overrightarrow{x}$ and random values at the particular position in the PR instances $\overrightarrow{y}_i$. Hence, it holds $Adv^{DSPRP}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r} \leq Adv^{DSPRP}_{\overrightarrow{x}',\lfloor k/2\rfloor,t,r}$ which finishes the proof. □

It remains to show the claims made during the proof. This is done next.

Proof. (Proof of Claim 1) The claim is that

$$|Pr[E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}|Bad^\pi_i] - Pr[E^{i-1,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}|Bad^\pi_i]| \leq r/|\mathbb{F}|.$$

Indeed, observe that in the conditional space Bad^π_i for the sampler $\tilde{\mathcal{S}}$ the $\pi(i)$-th location of the vector $\overrightarrow{y_\ell}$ for each $\ell \in [r]$ is distributed uniformly over the set $\mathbb{F} \setminus p_{\overrightarrow{y_\ell}}(x_{\pi(i)})$ where $p_{\overrightarrow{y_\ell}}$ is the solution polynomial that is selected by the sampler for the PRP instance $\overrightarrow{y_\ell}$.

The probabilistic operator R^π_i will substitute the $\pi(i)$-th location with a random element over $\mathbb{F}$. It follows by a standard argument that the statistical distance between the two distributions is at most $r/|\mathbb{F}|$ from which Claim 1 follows. □

Proof. (Proof of Claim 2) We have to show that

$$Pr[E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}|Good^{\pi}_i] = |Pr[E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t-1,r}|Bad^{\pi}_i].$$

The validity of the second claim can be established by directly corresponding the random coins of event $E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}$ in the conditional space $Good^{\pi}_i$ to the random coins of event $E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t-1,r}$ in the conditional space Bad^{π}_i. The event $E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}$ in the conditional space $Good^{\pi}_i$ can be thought of containing tuples of the form $\left(I^{Good}, (p^{Good}_{\ell}, \overrightarrow{e^{Good}_{\ell}}, \overrightarrow{r^{Good}_{\ell}})_{\ell=1,\ldots,r}\right)$ so that

- I^{Good} is a subset of $[n]$ of size t that necessarily includes $\pi(i)$,
- $p^{Good}_1, \ldots, p^{Good}_r \in \mathbb{F}[x]$ are polynomials of degree $< \lfloor k/2\rfloor$,
- $\overrightarrow{e^{Good}_1}, \ldots, \overrightarrow{e^{Good}_r}$ are vectors in $\mathbb{F}^n$ that are zero in (and only in) I^{Good}, and finally
- $\overrightarrow{r^{Good}_1}, \ldots, \overrightarrow{r^{Good}_r}$ are random vector of $\mathbb{F}^i$ that specify the coins of the probabilistic operator R^{π}_i.

On the other hand, the event $E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t-1,r}$ in the conditional space Bad^{π}_i can be thought of containing tuples of the form $\left(I^{Bad}, (p^{Bad}_{\ell}, \overrightarrow{e^{Bad}_{\ell}}, \overrightarrow{r^{Bad}_{\ell}})_{\ell=1,\ldots,r}\right)$ where

- I^{Bad} is a subset of $[n]$ with cardinality $t-1$ that excludes $\pi(i)$,
- $p^{Bad}_1, \ldots, p^{Bad}_r \in \mathbb{F}[x]$ are polynomials of degree $< \lfloor k/2\rfloor$,
- $\overrightarrow{e^{Bad}_1}, \ldots, \overrightarrow{e^{Bad}_r}$ are vectors in $\mathbb{F}^n$ that are zero in (and only in) I^{Bad}, and
- $\overrightarrow{r^{Bad}_1}, \ldots, \overrightarrow{r^{Bad}_r}$ are random vector of $\mathbb{F}^i$ that specify the coins of the probabilistic operator R^{π}_i.

Consider the following correspondence: given a tuple $\left(I^{Good}, (p^{Good}_{\ell}, \overrightarrow{e^{Good}_{\ell}}, \overrightarrow{r^{Good}_{\ell}})_{\ell=1,\ldots,r}\right)$ we define a tuple $\left(I^{Bad}, (p^{Bad}_{\ell}, \overrightarrow{e^{Bad}_{\ell}}, \overrightarrow{r^{Bad}_{\ell}})_{\ell=1,\ldots,r}\right)$ as follows: $I^{Bad} := I^{Good}\backslash\{\pi(i)\}$, $p^{Bad}_{\ell} := p^{Good}_{\ell}$, $\overrightarrow{r^{Bad}_{\ell}} := \overrightarrow{r^{God}_{\ell}}$ and also we set $(\overrightarrow{e^{Bad}_{\ell}})_j := (\overrightarrow{e^{Good}_{\ell}})_j$ for all $j \neq \pi(i)$ (note that $(\overrightarrow{e^{Good}_{\ell}})_{\pi(i)} = 0$ since $\pi(i)$ is not an error location, that is $\pi(i) \notin I^{Good}$ by assumption). Finally we select $(\overrightarrow{e^{Bad}_{\ell}})_{\pi(i)}$ at random from $\mathbb{F} \setminus \{p^{Bad}_{\ell}(x_{\pi(i)})\}$. We remark that the choice of $(\overrightarrow{e^{Bad}_{\ell}})_{\pi(i)}$ does not affect the outcome of the experiment since it substituted with the same random value in both cases. It follows that for every tuple of $E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}$ in the conditional space $Good^{\pi}_i$ we have a correspondence of the same number of tuples of $E^{i-1,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t-1,r}$ in the conditional space Bad^{π}_i. Based on this the statement of the claim follows. □

Proof. (Proof of Claim 3) The claim is that

$$|Pr[E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t-1,r}|Bad^{\pi}_i] - |Pr[E^{i-1,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}|Bad^{\pi}_i]| \leq Adv^{DSPRP}_{\overrightarrow{x}',\lfloor k/2\rfloor,t,r} + 3r/|\mathbb{F}|. \tag{20}$$

Recall that the event $E^{i,\pi}_{\overrightarrow{x},\lfloor k/2\rfloor,t,r}$ is defined as $A(R^{\pi}(\tilde{\mathcal{S}}_i(\overrightarrow{x}, \lfloor k/2\rfloor, t, r))) = 1$. We will argue that the two probability ensembles $R^{\pi}(\tilde{\mathcal{S}}_i(\overrightarrow{x}, \lfloor k/2\rfloor, t-1, r))$ and $R^{\pi}(\tilde{\mathcal{S}}_{i-1}(\overrightarrow{x}, \lfloor k/2\rfloor, t, r))$ are computationally indistinguishable when considered over the conditional probability spaces based on the event Bad^{π}_i, that is the event that $\pi(i) \notin I$. Suppose that D is any PPT distinguisher between the two ensembles. We define next a PPT distinguisher D' for DSPRP$[\overrightarrow{x}', \lfloor k/2\rfloor, t, r]$ over the support set $\overrightarrow{x}' = (\overrightarrow{x}_1, \ldots, \overrightarrow{x}_{i-1}, \overrightarrow{x}_{i+1}, \ldots, \overrightarrow{x}_n)$. That is it distinguishes between the ensembles

$\tilde{\mathcal{S}}^{\text{good}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)$ and $\tilde{\mathcal{S}}^{\text{bad}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)$. Let $\left(j, \overrightarrow{Y}\right)$ denote the challenge given to D' with $\overrightarrow{Y} = (\overrightarrow{y_1}, \ldots, \overrightarrow{y_r})$.

D' first randomizes the values $y_{1,j}, \ldots, y_{r,j}$. Then, in the next step, it parses each vector $\overrightarrow{y_\ell}$ as $(y_{\ell,1}, \ldots, y_{\ell,\pi(i)-1}, y_{\ell,\pi(i)+1}, \ldots y_{\ell,n})$, that is the value $y_{\ell,\pi(i)}$ is not defined yet. Next, it inserts a random value of $\mathbb{F}$ at location $\pi(i)$ and finally it selects $y'_{\ell,\pi(1)}, ..., y'_{\ell,\pi(i-1)}$ from $\mathbb{F}$ and overwrites the corresponding $i-1$ locations of $\overrightarrow{y_\ell}$. The resulting vector $\overrightarrow{y_\ell}^{new}$ is of length n. Let $\overrightarrow{Y}^{new}$ denote the collection of these new vectors. D' terminates by simulating D on $\overrightarrow{Y}^{new}$ and returning the output that D returns. This implies that

$$|Pr[D(\tilde{\mathcal{S}}^{\text{good}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)) = 1] - Pr[D(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)) = 1]| \leq Adv^{DSPRP}_{\overrightarrow{x}', \lfloor k/2 \rfloor, t, r}. \quad (21)$$

Suppose that the DSPRP$[\overrightarrow{x}', \lfloor k/2 \rfloor, t, r]$ challenge $\left(j, \overrightarrow{Y}\right)$ was drawn according to the sampler $\tilde{\mathcal{S}}^{bad}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)$. As $j \notin I$ by assumption, this means for each vector $\overrightarrow{y_\ell}$ that the j-th entry contains an element of $\mathbb{F} \setminus \{p_{\overrightarrow{y_\ell}}(x_j)\}$. Hence the SPRP instance $\overrightarrow{Y}$ with the j-th location of each vector being randomized is at a statistical distance $r/|\mathbb{F}|$ from $\tilde{\mathcal{S}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)$. Next, recall that we consider the conditional probability space based on Bad_i^π which means that $\pi(i) \notin I$. With a similar argument as the one just discussed, after the injection of random values at the $\pi(i)$-th locations yields a statistical distance $2r/|\mathbb{F}|$ from $R^\pi_{i-1}(\tilde{\mathcal{S}}(\overrightarrow{x}, \lfloor k/2 \rfloor, t, r))$. This implies that

$$|Pr[\mathcal{A}(R^\pi_{i-1}(\tilde{\mathcal{S}}(\overrightarrow{x}, \lfloor k/2 \rfloor, t, r))) = 1 | Bad_i^\pi] - Pr[D(\tilde{\mathcal{S}}^{\text{bad}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)) = 1]| \leq 2r/|\mathbb{F}|. \quad (22)$$

On the other hand, consider the case that the DSPRP$[\overrightarrow{x}', \lfloor k/2 \rfloor, t, r]$ challenge $\left(j, \overrightarrow{Y}\right)$ was drawn according to the $\tilde{\mathcal{S}}^{good}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)$ sampler, that is $j \in I$. We have the following: the vector $\overrightarrow{Y}$ with the j-th location randomized of each vector is at a statistical distance $r/|\mathbb{F}|$ from $\tilde{\mathcal{S}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t-1, r)$ where the index set t is reduced to $t-1$. It follows that, after injecting the random $\pi(i)$-th location elements and randomizing in each vector the $i-1$ locations according to π, the resulting vector $\overrightarrow{Y}^{new}$ is at a distance $r/|\mathbb{F}|$ from the ensemble $R^\pi_i(\tilde{\mathcal{S}}(\overrightarrow{x}, \lfloor k/2 \rfloor, t-1, r))$. This implies that

$$|Pr[\mathcal{A}(R^\pi_i(\tilde{\mathcal{S}}(\overrightarrow{x}, \lfloor k/2 \rfloor, t-1, r))) = 1 | Bad_i^\pi] - Pr[D(\tilde{\mathcal{S}}^{\text{good}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)) = 1]| \leq r/|\mathbb{F}|. \quad (23)$$

Putting equations 21, 22, and 23 together yields the statement of Claim 3 as follows:

$$\begin{aligned}
&|Pr[E^{i,\pi}_{\overrightarrow{x}, \lfloor k/2 \rfloor, t-1, r} | Bad_i^\pi] - |Pr[E^{i-1,\pi}_{\overrightarrow{x}, \lfloor k/2 \rfloor, t, r} | Bad_i^\pi]| \\
&= |Pr[\mathcal{A}(R^\pi_i(\tilde{\mathcal{S}}(\overrightarrow{x}, \lfloor k/2 \rfloor, t-1, r))) = 1 | Bad_i^\pi] - |Pr[\mathcal{A}(R^\pi_{i-1}(\tilde{\mathcal{S}}(\overrightarrow{x}, \lfloor k/2 \rfloor, t-1, r))) = 1 | Bad_i^\pi] \\
&\quad + (Pr[D(\tilde{\mathcal{S}}^{\text{good}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)) = 1] - Pr[D(\tilde{\mathcal{S}}^{\text{bad}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)) = 1]) \\
&\quad - (Pr[D(\tilde{\mathcal{S}}^{\text{good}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)) = 1] - Pr[D(\tilde{\mathcal{S}}^{\text{bad}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)) = 1]) \\
&\leq |Pr[\mathcal{A}(R^\pi_i(\tilde{\mathcal{S}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t-1, r))) = 1 | Bad_i^\pi] - Pr[D(\tilde{\mathcal{S}}^{\text{good}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)) = 1]| \\
&\quad + |Pr[\mathcal{A}(R^\pi_{i-1}(\tilde{\mathcal{S}}(\overrightarrow{x}, \lfloor k/2 \rfloor, t, r))) = 1 | Bad_i^\pi] - Pr[D(\tilde{\mathcal{S}}^{\text{bad}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)) = 1]| \\
&\quad + |Pr[D(\tilde{\mathcal{S}}^{\text{good}}(\overrightarrow{x}, \lfloor k/2 \rfloor, t, r)) = 1] - Pr[D(\tilde{\mathcal{S}}^{\text{bad}}(\overrightarrow{x}', \lfloor k/2 \rfloor, t, r)) = 1]| \\
&\leq r/|\mathbb{F}| + 2r/|\mathbb{F}| + Adv^{DSPRP}_{\overrightarrow{x}', \lfloor k/2 \rfloor, t, r}.
\end{aligned}$$