entropy

MDPI

Article

On Share Conversions for Private Information Retrieval

Anat Paskin-Cherniavsky * and Leora Schmerler

Computer Science Department, Ariel University, Ariel 40700, Israel
* Correspondence: anatpc@ariel.ac.il

Received: 10 June 2019; Accepted: 6 August 2019; Published: 23 August 2019

Abstract: Beimel et al. in CCC 12' put forward a paradigm for constructing Private Information Retrieval (PIR) schemes, capturing several previous constructions for $k \geq 3$ servers. A key component in the paradigm, applicable to three-server PIR, is a share conversion scheme from corresponding linear three-party secret sharing schemes with respect to a certain type of "modified universal" relation. In a useful particular instantiation of the paradigm, they used a share conversion from $(2,3)$-CNF over $\mathbb{Z}_m$ to three-additive sharing over $\mathbb{Z}_p^\beta$ for primes p_1, p_2, p where $p_1 \neq p_2$ and $m = p_1 \cdot p_2$. The share conversion is with respect to the modified universal relation C_{S_m}. They reduced the question of whether a suitable share conversion exists for a triple (p_1, p_2, p) to the (in)solvability of a certain linear system over $\mathbb{Z}_p$. Assuming a solution exists, they also provided a efficient (in $m, \log p$) construction of such a sharing scheme. They proved a suitable conversion exists for several triples of small numbers using a computer program; in particular, $p = p_1 = 2, p_2 = 3$ yielded the three-server PIR with the best communication complexity at the time. This approach quickly becomes infeasible as the resulting matrix is of size $\Theta(m^4)$. In this work, we prove that the solvability condition holds for an infinite family of (p_1, p_2, p)'s, answering an open question of Beimel et al. Concretely, we prove that if $p_1, p_2 > 2$ and $p = p_1$, then a conversion of the required form exists. We leave the full characterization of such triples, with potential applications to PIR complexity, to future work. Although larger (particularly with $max(p_1, p_2) > 3$) triples do not yield improved three-server PIR communication complexity via BIKO's construction, a richer family of PIR protocols we obtain by plugging in our share conversions might have useful properties for other applications. Moreover, we hope that the analytic techniques for understanding the relevant matrices we developed would help to understand whether share conversion as above for C_{S_m}, where m is a product of more than two (say three) distinct primes, exists. The general BIKO paradigm generalizes to work for such $\mathbb{Z}_m$'s. Furthermore, the linear condition in Beimel et al. generalizes to m's, which are products of more than two primes, so our hope is somewhat justified. In case such a conversion does exist, plugging it into BIKO's construction would lead to major improvement to the state of the art of three-server PIR communication complexity (reducing Communication Complexity (CC) in correspondence with certain matching vector families).

Keywords: PIR; Share conversion; CNF secret sharing; communication complexity

1. Introduction

A Private Information Retrieval (PIR) protocol [1] is a protocol that allows a client to retrieve the ith bit in a database, which is held by two or more servers, each holding a copy of the database, without exposing information about i to any single server (assuming the servers do not collaborate). In the protocol specification, the servers do not communication amongst each other. The main complexity measure to optimize in this setting is the communication complexity between client ans servers. In the single-server setting, the Communication Complexity (CC) is provably very high - provably, the whole database needs to be communicated. In the computational setting [2],

communication-efficient single-server PIR is essentially solved with essentially optimal CC [3]. In this work, we focus on information theoretic (in fact perfect) PIR protocols. See [4] and the references within for the additional motivation for the study of information theoretic PIR.

All known PIR protocols only use one round of communication (although it is not part of the definition of PIR), so we will only consider this setting. In this setting, the client sends a query to each server and receives an answer in return. PIR is a special case of secure Multi-Party Computation (MPC), which is a very general and useful cryptographic primitive, allowing a number of parties to compute a function f over their inputs while keeping that input private from an adversary that may corrupt certain subsets of the parties (to the extent allowed by knowing the output of f) [5,6]. The PIR setting is useful on its own right, as a minimalistic useful client-server setting of MPC, where the goal is to minimize communication complexity, potentially minimizing the overhead on the client, which may be much weaker than the server.

In [4], the authors proposed the following approach to constructing PIR protocols, which captures some of the previous protocols for three servers or more, and put forward a new three-server PIR protocol, with the best known asymptotic communication complexity to date.

Let us describe their general framework, for a k-server PIR.

1.1. Biko's Framework for K-Server PIR

The framework uses two key building blocks. One is a pair of k-party linear secret sharing schemes Sh_1, Sh_2 over abelian rings G_1, G_2, respectively. The pair Sh_1, Sh_2 is also equipped with a share conversion scheme from Sh_1, Sh_2 with respect to some "useful" relation $R \subseteq G_1 \times G_2$. That is, for a value s shared according to Sh_1, the scheme allows locally (performing a computation on each share separately) computing a sharing of a value o according to Sh_2, such that $R(s, o)$ holds (note that this is generally non-trivial, as the conversion is performed locally on each share, without knowing anything about the other shares or the randomness used to generate the sharing).

In [4], G_1, G_2 used small (constant) finite rings. The notion of share conversion employed in [4] generalized [7]'s work on locally converting from the arbitrary linear secret-sharing schemes without changing the secret, by allowing an arbitrary relation between the secrets and by allowing Sh_1, Sh_2 to be linear over different rings. The second building block is an encoding of the inputs $x \in \{0,1\}^n$ as a longer element $u \in G_1^l$. The PIR protocol has the following structure:

1. The client "encodes" the input $x \in \{0,1\}^n$ via $u \in G_1^l$ for a quite large, but not very large $l(n)$ from a certain code $\mathcal{C} \subseteq G_1^l$.
2. Every bit u_j is shared via Sh_1. The client sends each share $s_{i,j}$ to server S_j.
3. The servers are able to evaluate locally any shallow circuit from a certain set $\mathcal{F}$ roughly as follows. At the bottom level, we use the linearity of Sh_1 to evaluate a sequence of gates taking linear combinations of the u_i's over G_1. Let y denote the resulting shared values vector. Apply the share conversion to the output of each of the y_i's to obtain a converted vector y' of values over G_2. Finally, locally evaluate some linear combination of the elements y', using the linearity of Sh_2. Each server sends its resulting share to the client. The particular circuit to evaluate is that of an "encoding" $f' : \mathcal{C} \to G_2$ of their database, in turn viewed as a function $f : \{0,1\}^n \to \{0,1\}$.
4. The client reconstructs the output value from the obtained shares, using Sh_2's reconstruction procedure. It then decodes $f(x)$ from $f'(u)$ (the decoding procedure takes only $f'(u)$ as input; it does not use x or the client's randomness).

Note that the "privacy" property of Sh_1 takes care of keeping the client's input private from any single server. It is not known how, and likely impossible, to devise a share conversion scheme for a relation that is sufficiently useful to compute locally all functions $f : \{0,1\}_1^l \to \{0,1\}$ following the framework suggested above, in a way that encodes every $f : \{0,1\}^l \to \{0,1\}$. Therefore, to encode all functions $f : \{0,1\}^l \to \{0,1\}$, BIKOencodes the inputs x and function $f : \{0,1\}^n \to \{0,1\}$ using a larger parameter $l > n$. To gain in communication complexity, the code $\mathcal{C}$ is carefully chosen so

that the resulting family $\mathcal{F} \subseteq \{f' : G_1^l \to \{0,1\}\}$ implemented by the above protocol is as large as possible (relative to l). More precisely, its VCdimension is at least 2^n, allowing one to implement any Boolean function on $\{0,1\}^n$. On the other hand, we must be able to match it with a relation R for which a share conversion exists (Recall that the VC dimension captures the ability of a set $\mathcal{F}$ of functions $f : D \to \{0,1\}$ to "encode" any function $f : \{0,1\}^n \to \{0,1\}$ by restriction to a subset $I \subseteq D$ of size 2^n of values. Here, D is typically larger than $\{0,1\}^n$. By Saur's lemma, the VC dimension is very closely related to the size of the family $\mathcal{F}$.) There is a tradeoff between the share conversions we are able to devise and the complexity of $\mathcal{F}$. The larger $\mathcal{F}$ in the above scheme is, the smaller l that can be used for the encoding, and the lower the communication complexity of the resulting PIR protocol. Note that the framework yields constructions where the communication complexity of $O(l)$ bits is dominated by the client's message length and only constant-length server replies. On the other hand, for more complex $\mathcal{F}$, one would need share conversion schemes for "trickier" relations, which are harder to come by.

Only a few families of instantiations of the framework are known, with [4]'s particular instantiation for three-server PIR as one example. Other instantiations are implicit in some previous PIR protocols; see Section 2, or [4] for additional examples and more intuition.

1.1.1. BIKO's New Family of Three-Server PIR

In their new three-server PIR, the work in [4] instantiated the components of the above framework using a certain (family of) rings G_1, G_2, for which suitable encodings and share conversion exist. Details follow. The rings are parameterized by three numbers p_1, p_2, p, where p_1, p_2 are distinct primes. Let us denote $m = p_1 \cdot p_1$. The various components are instantiated as follows:

- Sh_1 is the (2,3)-CNF secret sharing scheme for the ring $G_1 = \mathbb{Z}_m$, for certain $m = p_1 \cdot p_2$, where p_1, p_2 are distinct primes. Sh_2 is the three-additive secret sharing over the ring $G_2 = \mathbb{Z}_p^\beta$, for some $\beta \in \mathbb{N}$.
- Let $S \subseteq \mathbb{Z}_m \setminus \{0\}$. The input $x \in \{0,1\}^n$ is encoded as an element u of an S-Matching Vector (MV) $\mathcal{C} \subseteq \mathbb{Z}_m^l$ family of size at least 2^n over $\mathbb{Z}_m$ [8]. Briefly, for $S \subseteq \mathbb{Z}_m \setminus \{0\}$, such an S-MV family is a sequence of vectors $v_1, \ldots, v_L \in \mathbb{Z}_m^l$, where for all i, $< v_i, v_i >= 0$, and for all $i \neq j$, $< v_i, v_j > \in S$. In particular, [8] demonstrates that an S-MV family of any VC dimension exists for all $m = p_1 p_2$ where p_1, p_2 are distinct primes for S as small as S_m; the set of all elements in $\mathbb{Z}_m$ that are zero or one modulo each of the p_i's, expect for zero. In particular, the vector's length for an instance corresponding to VC dimension 2^n is $l = 2^{\tilde{O}(\sqrt{n})}$.
- A share conversion between Sh_1 and Sh_2 for a relation C_{S_m} is obtained for certain triples (p, p_1, p_2). Informally, this relation maps S_m to zero and zero to some non-zero value in G_2. This type of share conversion scheme is another novel technical contribution of [4] (and the focus of our work).

The complete BIKO construction is roughly as follows:

1. The client sends an encoding $u = u(x)$ of its input $x \in \{0,1\}^n$, which belongs to the the S_m MV-family $\mathcal{C} \subseteq \mathbb{Z}_m^l$ (here, $l = O(2^{\sqrt{nlog(n)}})$ can be achieved). The servers locally evaluate the following circuit:
$$C(u_1, \ldots, u_l) = OR_j(f_i \cdot < u_j, u >)$$
using the following "noisy" procedure:
2. Generate the vector of values $(y_i)_{f(i)=1}$ where $y_i =< u_i, u >$. This evaluation uses the linearity of CNF and produces a local $Sh_1 = (2,3)$-CNF over the $\mathbb{Z}_m$ share of each value y_i.
3. Apply share conversion from Sh_1 to Sh_2 as specified above on each y_i, obtaining a vector of y_i''s, shared according to Sh_2.
4. Locally evaluate $\sum_i y_i'$, and then, send (Sh_2) shares to the client. The client outputs zero if this value is zero, and one otherwise.

It is not hard to see that the above construction always produces the correct value. In particular, by the definition of C_{S_m}, only (the conversion of) $< u_i, u_i >$ contributes a non-zero value y_i' to $\sum_i y_i'$.

Theorem 1 ([4], informal). *Assume there exists a share conversion scheme from* $(2,3)$*-CNF over* $G_1 = \mathbb{Z}_m$ *and three-additive over* $G_2 = \mathbb{Z}_p^\beta$ *for some* β*, where* $m = p_1 \cdot p_2$ *is as above. Then, there exists a three-server PIR with communication complexity* $O(2^{2 \cdot p_2 \sqrt{nlog(n)}})$*, where* $p_2 > p_1$*. The best asymptotic communication complexity is obtained by setting* $m = 6, p = 2$*, using [8]'s* S_m*-MV family over* G_1^l *over* $\mathbb{Z}_6$ *with* $l = 2^{\tilde{O}(\sqrt{n})}$ *and a share conversion with respect to* C_{S_m} *for* $p_1 = p = \beta = 2, p_2 = 3$*. In particular, the servers' messages are of length* $O(1)$*.*

Intuitively, having S-MV families for a "small" S as above facilitates the construction of share conversion schemes as we needed for C_S, since a small S imposes fewer constraints on the pairs (s, s') in the relation.

To conclude, it follows from Theorem 1 that to obtain three-server PIR with sub-polynomial (in database size) CC, it suffices to design a share conversion for C_{S_m} from Sh_1 to Sh_2 over groups G_1, G_2 corresponding to any triple of parameters p_1, p_2, p as described above. Another useful contribution of [4] is reducing the question of whether such a conversion exists (for some β) to a system of equations and inequalities over $\mathbb{F}_p$; see the following sections for more details on the usefulness of this characterization (a solution to the system corresponds to the share conversion scheme).

1.1.2. On the Choice of Input and Output Schemes

In retrospect, trying to convert from CNF to additive is a natural choice. For once, the CNF scheme is known as the most redundant secret sharing scheme and additive is the least redundant one [7]. More specifically, CNF sharing is convertible to any other linear scheme over the same field for the same access structure, or one contained it with respect to the identity relation. This implies that for any relation, if share conversion between two linear schemes exists for that relation, it must exist for CNF to additive. Here the more standard linearity over fields is considered, and the same field us used for both schemes. Still, for the more general notion of linearity over rings, and allowing different rings for the two schemes, this still provides quite strong evidence that starting with CNF, and trying to convert to additive is a good starting point. As to the choice of the structure of the particular G_1, G_2 over which the conversion is defined, following are some of [4]'s considerations for their choice. One observation is that trying to convert from, say, $\mathbb{Z}_m$ to $\mathbb{Z}_p$ would require finding a solution to a system of inequalities over $\mathbb{Z}_p$ for a fixed p. This problem is generally NP-complete, which is likely to make the problem hard to analyze, even in this special case. Thus, the authors extend the search to allow conversion to three-additive over some extension field $\mathbb{F}_{p^\beta}$. As β is constant, it does not have much effect on the share complexity of the resulting PIR protocol.

Using composite sizes for both $G_1 = \mathbb{Z}_m$ and $G_2 = \mathbb{Z}_{m'}$ would lead to a greatly improved VC dimension of $\mathcal{F}$, with respect to the canonical relation C_S, where $S = \mathbb{Z}_m \setminus \{0\}$. In fact, the $\mathcal{F} = +_{\mathbb{Z}_{m'}}(mod_m)$ that can be locally evaluated over $\mathbb{Z}_m^l$ is universal, resulting in near-optimal $O(\log(n))$ communication complexity. However, [4] have proven this type of conversion does not exist, indicating that C_S for smaller S for such G_1, G_2 could also be hard to find.

1.2. Our Contribution

As described above, the contribution of [4] was two-fold. First, they put forward a useful framework for designing PIR protocols, capturing some of the best-known three-server (or more) PIR protocols. Second, they put forward an instantiation of the framework, which reduces the question of the existence of a three-server PIR protocol to the existence of a share conversion for certain parameters p_1, p_2, p and certain linear sharing schemes over abelian rings G_1, G_2 determined by the parameters. They further reduced the question of the existence of a share conversion as above for parameters p_1, p_2, p to the question of whether a corresponding system of linear equations is solvable over $\mathbb{Z}_p$. This system in turn is not solvable over some extension field $\mathbb{F}_{p^\beta}$ iff a certain system of both equalities and inequalities over $\mathbb{Z}_p$ is solvable over some field $\mathbb{F}_{p^\beta}$.

While the solvability of a system $Ax = b$ can be verified efficiently for a concrete instance, it does not provide a simple condition for characterizing triples (p, p_1, p_2) for which solutions exist. More concretely, even the question of whether an infinite set of such triples exists remains open. We resolve this question in the affirmative, proving the following.

Theorem 2. *Let $2 < p_1, p_2$ where $p_1 \neq p_2$ are primes. Then, a share conversion from $(2,3)$-CNF over $\mathbb{Z}_{p_1 p_2}$ to three-additive over $\mathbb{Z}_{p_1}^{\beta}$ exists for some $\beta > 0$.*

In a nutshell, our goal is to study the solvability of the particular system $Ax = b$ from [4] corresponding to a given p, p_1, p_2. Our main tool is a clever form of elimination operations on the (particular) matrix $(A; b)$, so that at each step, the intermediate system $A'x = b'$ is solvable iff $Ax = b$ is. The special type of elimination we develop is useful here as it allows keeping our particular system relatively simple at every step. This is as opposed to using standard Gaussian elimination, for instance, which would have made the resulting system quite messy. The operations are largely oblivious of the particular value of p_1, p_2, p until a very late point in the game. The proof consists of a series of a (small) number of our elimination steps. On a high level, we perform a few such steps. At this point, the matrix becomes quite complicated (at the end of Section 4.2). Then, we perform a change of basis (Section 4.5) to facilitate another convenient application of the step. At this point, we are ready to complete the proof by directly proving that the system is not solvable (and thus, a share conversion exists).

1.2.1. On the Potential Usefulness of Our Result for PIR

In this work, we identified a certain infinite set of parameters (p_1, p_2, p) for which a share conversion as required in Theorem 1 exists. This result itself does not appear to yield improved three-server PIR protocols by instantiating the share conversion with the newly-found parameters via Theorem 1. Indeed, increasing p_2 beyond the minimal possible $p_2 = 3$ (which was already known) does not seem to help. However, Theorem 1 generalizes to m's, which are products of a larger number of primes. If a share conversion for $G_1 = \mathbb{Z}_m, G_2 = \mathbb{Z}_p^{\beta}$ derived from such $m = p_1 \cdot p_2 \cdot p_3, p$ exists, it can be paired with an S_m-MV family (which is also known to exist from [8]) with significantly improved (over m a product of two primes) VC dimension, to obtain an improved three-server PIR seamlessly. Specifically for $m = p_1 \cdot \ldots \cdot p_r$, an S_m MV family with VC dimension 2^n in $\mathbb{Z}_m^l$ for $exp(\tilde{O}(log^{1/r}(n)))$ exists.

Therefore, it remains to design a corresponding share conversion scheme (or prove it impossible to rule out this direction). Furthermore, the linear algebraic characterization of the existence of such sharing schemes remains the same. Therefore, hopefully, the analytic techniques we developed here for the the case where m is a product of two primes that could help understand the case of a larger number of primes.

1.2.2. Road Map

In Section 2, we refer to some related work on PIR. In Section 3, we include some of the terminology and preliminary results from [4], which is our starting point. In Section 4, we present our main result, which is broken into subsections as follows. As explained above, we start with the reduction of the problem by [4] to verify whether a certain matrix-vector pair (A, b), $Rows(A)$ spans b. We perform a series of elimination steps on the matrix $(A|b)$ to bring it to a simpler form (A', b') so that $Rows(A')$ spans b' iff $Rows(A)$ spans b. All elimination steps are of the same general form. Section 4.1 formalizes this form as a certain lemma. Section 4.2 outlines an initial sequence of applications of that lemma. In Section 4.5, we perform a change of basis to help facilitate further use of the lemma. In Section 4.6, we indeed apply the lemma once again. Finally, in Section 4.7, we obtain a simple enough matrix $(A'|b')$ for which we are able to prove A' spans b' directly for the proper choice of parameters, proving our positive result.

2. Related Work

2.1. Schemes with Polynomial CC

As stated in the Introduction, our work is a followup on a particular approach to constructing PIR protocols [1], focusing on $k = 3$ servers (almost the hardest case). We will survey some of the most relevant prior works on PIR (omitting numerous others). Already in [1], the authors suggested a non-trivial solution to the hardest two-server case, with communication complexity $O(2^{n/3})$, and more generally, $O(2^{\frac{n}{k}})$ for $k \geq 3$ servers. Building on a series of prior improvements, the work in [9] put forward protocols with CC $2^{\frac{O(n \cdot loglogk}{klogk})}$ for large k. Their result also improved upon the state-of-the-art at the time for small values of k, in particular achieving CC of $O(2^{4n/21})$ for $k = 3$, improving upon the best previously-known CC of $O(2^{n/5})$.

Work falling in the framework on BIKO

In [9], they restated some of the previous results in a more arithmetic language, in terms of polynomials. Furthermore, they considered a certain encoding of the inputs and element-wise secret sharing the encoding, which is somewhat close to the BIKO framework.

Particularly interesting for our purposes is their presentation of a "toy" protocol achieving CC of $O(2^{n/2})$ from earlier literature reformulated in [9], which constituted one of the building blocks of the construction of [9]. That protocol was almost an instance of the BIKO framework as sketched in the Introduction, and it is instructive to consider the differences (the full-fledged result of [9] used additional ideas, and we will not go over it here for the sake of simplicity).

1. The client "encodes" its input $x \in \{0,1\}^n$ via $u \in \mathbb{F}_2^l$ where $l \approx 2^{n/2}$ by mapping the input to a vector with exactly two ones and zeroes elsewhere.
2. Every bit u_j is shared via Sh_1, which is $(2,3)$-CNF over $\mathbb{F}_2$.
3. The servers represent their database as a degree-two polynomial in $\mathbb{F}[x_1, \ldots, x_l]$ (where all monomials are of degree exactly two). To evaluate a polynomial:

$$f(x) = \sum_{i<j\leq l} a_{i,j} x_i x_j$$

   proceed as follows:

   (a) At the bottom level, perform a two-to-one share conversion for evaluating each monomial, where the output for gate $x_i x_j$ is a three-additive sharing of the result $x_i x_j$. In some more detail, let $x_{i,1} + x_{i,2} + x_{i,3}$ denote the additive shares produced in Step 1 of the CNF-sharing. The computation is made possible since $(x_{i,1} + x_{i,2} + x_{i,3}) \cdot (x_{j,1} + x_{j,2} + x_{j,3})$, which needs to be computed; for every share-monomial $x_{i,k} x_{k,d}$, at least one of the servers does not miss any shares in it. Each server outputs the sum of the monomials it knows as its new share.
   (b) Using the linearity of the three-additive scheme, the servers compute a three-additive sharing $f(x)$, based on the shares of the individual monomial evaluations y_i.

   Each server sends its resulting share to the client.
4. The client reconstructs the output value from the obtained shares, as in BIKO.

This framework differs from BIKO in a few ways. First, the structure of the circuit locally evaluated is different. In the above example, "many-to-one" rather than "one-to-one" secret sharing is used at the bottom level. Then, the linearity of Sh_2 is employed on the upper level. In BIKO, first, the linearity of Sh_1 is used to evaluate a vector of linear combinations, then a one-to-one share conversion from Sh_1 is used at the middle level, then again, the linearity of Sh_2 is used at the top gate. Besides the different structure of the circuit, using many-to-one share conversion is the main conceptual change. It employs the extra property of $(2,3)$-CNF, which allows it to evaluate degree-two polynomials,

rather than just linear functions. In the evaluation process, the output is already converted to the less redundant three-additive sharing. The relation the share conversion respects is just the evaluation of the monomial: $R(o, x_1, x_2)$ iff $o = x_1 x_2$. Conceptually, local evaluation of linear functions as used in BIKO is in fact already a many-to-one share conversion from a linear scheme to itself, evaluating a relation $R(x_1, \ldots, x_l, o)$, which is satisfied iff $o = \ell(x_1, \ldots, x_l)$ for a certain linear function ℓ.

2.2. Schemes with Sub-Polynomial CC

The first three-server PIR protocol with (unconditionally) sub-polynomial (in 2^n) CC was put forward by [10]. It falls precisely into BIKO's framework, using a share conversion from the so-called $(2,3)$-modified Shamir secret over $G_1 = \mathbb{Z}_m$ for $m = p_1 \cdot p_2$ where $p_1 \neq p_2$ are odd primes to three-additive over the field $G_2 = \mathbb{F}_{p^\beta}$ where p is some prime, such that m divides $|G_2^*| = p^\beta - 1$.

To get to three, rather than four servers, some additional properties of G_1, G_2 were in fact required. The share conversion is with respect to the relation C_{S_m} (actually, a slightly stricter relation, even). Concretely, the work in [10] found an example of such groups G_1, G_2 as above obtained from $m = 7 \cdot 73 = 511$, $p = 2, \beta = 9$, having an additional useful property, which allowed going down from four to three servers via computer-aided search. We note that a share conversion from $(2,3)$-CNF to three-additive over that pair of groups also exists since $(2,3)$-CNF can be converted to any linear scheme for an access structure containing a two-threshold (including modified $(2,3)$-Shamir) [7].

The encoding used is via a S_m-MV family over $\mathbb{Z}_m$, as in [4]. The evaluation points in the Shamir scheme used are tailored to S_m in the corresponding MV family. It achieves a CC of $O(2^{146\sqrt{n \log(n)}})$, later improved to $O(2^{6\sqrt{n \log(n)}})$ by BIKO's construction.

Qualitatively, the result of [10], preceded by [11] in terms of using a similar idea, and most later constructions greatly improved the CC relative to earlier constructions such as [9] by using MV-codes rather than low-degree polynomial codes. The former codes had a surprisingly large rate. Indeed, looking at MV-families from the perspective of [12], these are also polynomial codes, over a basis of monomials corresponding to an MV family, rather than low-degree monomials.

All the above examples generalize to larger k, with somewhat improved complexity; we focused on the (hardest) $k = 3$ here for simplicity.

Two-Server PIR

In a breakthrough result, the work in [13] matched the CC of two-server PIR with the CC of the best-known three-server PIR, improving from the best previously-known $2^{n/3}$ to $2^{O(\sqrt{nlog(n)})}$. Their work cleverly combined and extended several non-trivial ideas, both new and ones that appeared in previous work in some form. In a nutshell, one idea is to encode inputs via MV codes, where the operations on the vectors are done "in the exponent", resulting in a variant of MV codes as polynomials, as defined in [12] and implicitly used in [10]. Another idea is to output both evaluations of these "exponentiated" polynomials and the vector of their partial derivatives, to yield more information to facilitate reconstruction given only two servers.

As for casting their result into the BIKO framework(we believe it is a meaningful generalization thereof, as we describe below):

1. The client encodes its input $x \in \{0,1\}^n$ via a vector $u = u(x)$ in an S_6-MV family $\mathcal{C} \subseteq \mathbb{Z}_6^l$ (with l about $2^{\tilde{O}(\sqrt{n})}$ as explained above).
2. Every bit u_j is shared via Sh_1, which is the $(2,2)$-modified Shamir scheme over $\mathbb{Z}_m$. The evaluation points are zero and one. We note that the choice of m is not as constrained as in the construction of [10], and many other pairs would work as well. This holds as the purpose of the polynomial involved is somewhat different.

3. Let $R_{6,6}$ denote the ring $\mathbb{Z}_6[z]/(z^6-1)$. Let $\gamma = z$, which has multiplicative degree six in $R_{6,6}$. The servers are able to evaluate locally the encoded database:

$$F(x_1, \ldots, x_l) = \sum_{i \in [2^n]} a_i \mathbf{x}^{\mathbf{u_i}}$$

roughly as follows:

   (a) At the bottom level, use the linearity of Sh_1 to evaluate locally each $y_i =< \mathbf{u_i}, \mathbf{x} >$.
   (b) Convert each of the y_i's into a $(2,2)$-sharing scheme Sh_2 over a ring $G_2 = R_{6,6}^{\Omega(l)}$, obtaining a vector of y_i''s. The conversion is a many-to-one conversion with respect to a relation R over the entire input vector, the current y_i and y_i', so that $R(x_1, \ldots, x_{2^n}, y_i, y_i')$ that outputs $C_{S_6}(y_i, y_i')$ if $(x_1, \ldots, x_{2^n})$ is in $\mathcal{C}$, and otherwise undefined.
   (c) Use the "almost linearity" of Sh_2 over our particular subset of values to evaluate $\sum_i a_i y_i'$. Here. "almost linearity" means that to evaluate a sum of shared secrets, we add some of the coordinates and copy other coordinates.

4. The client reconstructs the output value from the obtained shares, according to Sh_2 (the reconstruction function is of degree-two.) In a nutshell, the coefficient a_x is the only one determining the free coefficient of a degree-six univariate polynomial obtained from restricting F to the line $z + ut$, where z is a random vector. The reconstruction procedure uses the shared evaluations as derivatives to compute this free coefficient, which is non-zero iff a_x is one. In the original protocol description of [13], the client used the input for reconstructing the output, which is not consistent with share conversion, where the input is not required for reconstruction. We restate the above protocol by letting the servers "play back" the original shares to the client to stay consistent with the BIKO share conversion framework.

To summarize, the construction falls into an extended BIKO framework, where the middle-layer share conversions are many-to-one rather than one-to-one. Furthermore, the scheme Sh_2 is non-linear and is defined over a non-constant domain. It is important to observe that the relations used for the conversions are independent of the database function F itself, but only depend on its size. The locally-evaluated relations are as small as $O(2^n)$, similarly to BIKO. One potential difficulty with using such an extended design framework is in verifying whether a many-to-one share conversion exists, or even one-to-one for a large share domain of Sh_2, as we have here.

One question that remains open in the two-server setting is to make the server's output size constant as in [4,10] for three servers.

3. Preliminaries

3.1. Secret Sharing Schemes

For a set $A \subseteq [n]$ and sequence of shares $\mathbf{s} = (s_1, \ldots, s_n)$, we denote $\mathbf{s}[A]$ by the sequence of shares $(s_i)_{i \in A}$. A secret sharing scheme for n parties implements an access structure specified by a monotone function $f : \{0,1\}^n \to \{0,1\}$ specifying the so-called *qualified* sets of parties that recover the secret (preimages of one), while other sets learn nothing about the secret (in this paper, we consider the standard perfect setting). We refer to such a scheme as an n-party secret sharing scheme. More formally, an n-party secret sharing scheme is a randomized mapping:

$$Sh : S \times R \to S_1 \times \ldots \times S_n$$

where S is a finite domain of secrets and R is a randomness set, while $S_1, \ldots, S_n$ are finite share domains such that the following holds:

- Correctness: For all $A \subseteq [n]$ with $f(A) = 1$, referred to as qualified sets, and all $s_1 \neq s_2 \in S$, we have:
$$support(Sh_{r \leftarrow R}(s_1, r)[A]) \cup support(Sh_{r \leftarrow R}(s_2, r)[A]) = \phi$$
We use sets and their characteristic vectors interchangeably.
- Privacy: For all $A \subseteq [n]$ with $f(A) = 0$ and all $s_1 \neq s_2 \in S$, the following distributions are equal.
$$Sh_{r \leftarrow R}(s_1, r)[A]) = (Sh_{r \leftarrow R}(s_2, r)[A])$$

Access structures where the set of qualified subsets is exactly those of size some t or more, are called threshold access structures.

We say that a secret sharing scheme is linear over some ring G, generalizing the standard notion of linearity over a field, if $S = G, R = G^l$ for some $l \in \mathbb{N}$ for some finite Abelian ring G. Each share consists of one or more linear functions for the form $l(s, r_1, \ldots, r_m) = \alpha_0 \cdot s + \sum_i \alpha_i \cdot r_i$ (Observe that unlike linear schemes over a field, such a scheme does not always have perfect correctness for all secrets $s \in R$, but this is not required in our context. We will require perfect privacy though.) A useful property of such schemes is that they allow evaluating locally linear functions of the shares. That is, for a pair of sharings $\mathbf{s^1} = (s_1^1, \ldots, s_n^1)$ $\mathbf{s^2} = (s_1^2, \ldots, s_n^2)$ of some s_1, s_2, respectively, $\mathbf{s^1} + \mathbf{s^2}$ (performing addition coordinate-wise on each group element in the share vector) is a sharing of $s_1 + s_2$. Similarly, for $a \in G$, $a \cdot \mathbf{s^1}$ is sharing of $a \cdot s_1$.

See [14] for a survey on secret sharing.

Let us recall the well-known schemes $(2,3)$-CNF and 3-additive for completeness. The schemes are more general, $(2,3)$-CNF is a special case of (t,n)-CNF and 3-additive is a special case of (t,n)-DNF with $t = n = 3$. Here we explicitly recall the definitions only of the special cases we need.

A 3-additive secret sharing scheme over a ring G is a randomized mapping $f_{ADD} : S \times R \to S_1 \times S_2 \times S_3$, where $S = S_i = G, R = G^2$, such that $f_{ADD}(s; a_1, a_2) = (a_1, a_2, s - a_1 - a_2)$. It is not hard to see that f_{ADD} indeed implements an access structure where only the set $\{1,2,3\}$ is qualified.

A $(2,3)$-CNF over a ring G is defined by a randomized mapping $f_{CNF} : S \times R \to G^3$ for $S = G$, $S_i = R = G \times G$. It is defined as follows $f_{CNF}(s; a_1, a_2)$:

1. Calculate a three-additive sharing $(a, b, c) = f_{ADD}(s; a_1, a_2)$.
2. Output shares (s_1, s_2, s_3) where s_i equals the two elements from the tuple in one, which are not at its index. For example, $s_3 = (a, b)$.

It is not hard to see that $(2,3)$-CNF is indeed a 2-threshold scheme. It is also not hard to see that the above schemes are linear over their respective rings.

A $(2,n)$-modified Shamir scheme over a ring G is defined by a randomized mapping $f_{Sh}(s; z)$: $G \times G \to G$ outputs shares $(s_1, \ldots, s_n)$ $s_i = z + sx_i$, where $x_i \in G$ is a distinct constant 'evaluation point'.

It is not hard to see that $(2,3)$-CNF is indeed a two-threshold scheme. It is also not hard to see that the above schemes are linear over their respective rings.

3.2. Share Conversion

We recall the definition of (generalized) share conversion schemes as considered in our paper. Our definition is exactly the definition in [4], in turn adopted from previous work.

Definition 1 ([4]). *Let L_1 and L_2 be two n-party secret-sharing schemes over the domains of secrets K_1 and K_2, respectively, and let $C \subseteq K_1 \times K_2$ be a relation such that, for every $a \in A_1$, there exists at least one $b \in K_2$ such that $(a, b) \in C$. A share conversion scheme* convert$(s_1, \ldots, s_n)$ *from L_1 to L_2 with respect to relation C is specified by (deterministic) local conversion functions $g_1, \ldots, g_n$ such that: If $(s_1, \ldots s_n)$ is a valid sharing for some secret s in L_1, then $g_1(s_1), \ldots g_n(s_n)$ is a valid sharing for some secret s' in L_2 such that $(s, s') \in C$.*

(In [4], they referred to such share conversion schemes as "local" share conversion. As this is the only type we consider here, we will refer to it as simply "share conversion".)

For a pair of Abelian groups G_1, G_2 (when G_1, G_2 are rings, we consider G_1, G_2 as groups with respect to the "$+$" operation of the rings), we define the relation C_S as in [4] (G_1, G_2 will be clear from the context, so we exclude the groups from the definition of C_S to simplify notation).

Definition 2 (The relation C_S [4]). *Let G_1 and G_2 be finite Abelian groups, and let $S \subseteq G_2 \setminus \{0\}$ (when G_1, G_2 are rings, we will by default refer to the additive groups of the rings in this context). The relation C_S converts $s = 0 \in G_1$ to any nonzero $s \in G_2$ and every $s \in S$ to $s = 0$. There is no requirement when $s \notin S \cup 0$. Formally,*

$$C_S = \{(s,0)|s \in S\} \cup \{(0,s') : s' \in G_2 \setminus \{0\}\} \cup \{(s,s')|s \notin S \cup \{0\}, s' \in G_2\}$$

Given $m = p_1 \cdot p_2$, where $p_1 \neq p_2$ are primes and p is a prime, we consider pairs of rings $G_1 = \mathbb{Z}_m, G_2 = \mathbb{Z}_p^\beta$. We denote $S_m = \{x \in G_1 | \forall i \in [2], x \mod p_i \in \{0,1\}\} \setminus \{0\}$.

3.3. Our Starting Point: The Modeling of [4]

In this work, we study the existence of share conversions for three parties with respect to the canonical relation C_{G_1,G_2} as above, from $(2,3)$-CNF to three-additive for various parameters p_1, p_2, p.

Our starting point is the characterization from [4] of triples (p_1, p_2, p) for which a share conversion with respect to (G_1, G_2) as above exists via a linear-algebraic constraint.

In some more detail, consider (p_1, p_2, p) as above. A share conversion from $(2,3)$-CNF to three-additive exists for S_m iff a certain condition is satisfied by the following matrix $M_{\equiv,\not\equiv}$ over $\mathbb{Z}_p$.

In the matrix $M_{\equiv,\not\equiv}$, the rows are indexed by triples $(a,b,c) \in \mathbb{Z}_m^3$, corresponding to $(2,3)$-CNF sharings of some $s \in S \cup \{0\}$. Namely, (a,b,c) are the (additive) shares generated by f_{CNF} in Step 1. The rows corresponding to $s \neq 0$ form $M_{\equiv}$. The rows corresponding to $s = 0$ form $M_{\not\equiv}$. The columns of $M_{\equiv,\not\equiv}$ are indexed by values in $[3] \times \mathbb{Z}_m \times \mathbb{Z}_m$. Intuitively, an index (i,x,y) of a column corresponds to share s_i of the $(2,3)$-CNF scheme being equal to (x,y). Row (a,b,c) has ones at three locations: $(3,a,b), (2,a,c), (1,b,c)$, and zeros elsewhere. That is, there are zeros at columns corresponding to the shares s_i output by f_{CNF} corresponding to (a,b,c) produced in Step 1 of f_{CNF}'s execution.

We are searching for a vector u that by multiplying it with each row of the "equality rows" of the matrix, it will be equal zero, and by multiplying it with one of the inequality rows, it will not be equal to zero.

The solution vector u (and thus, the columns of the matrix) is indexed by $[3] \times \mathbb{Z}_m \times \mathbb{Z}_m$. The index of an entry (i,x,y) corresponds to a CNF-share s_i that equals (x,y), and the value $u_{(i,x,y)}$ at this index is the value in $\mathbb{F}_{b^\beta}$ to which share $s_i = (x,y)$ is converted.

Indeed, it is not hard to see that a share conversion scheme exists iff a solution $x \in \mathbb{F}_{p^\beta}$ to the system:

$$\begin{cases} M_{\equiv} x = 0 \\ M_{\not\equiv} x \neq 0 \end{cases}$$

exists.

Some basic linear-algebraic observations imply that the above is equivalent to the fact that the rows of $M_{\equiv}$ do not span some row of $M_{\equiv}$ over $\mathbb{F}_p$ (this simplification matters, as it does not require us to know β in advance, if it exists).

Furthermore, the work in [4] provided a quantitative lower bound on β, depending on the degree difference between $M_{\equiv}$ and $M_{\not\equiv}$ (the latter is not significant towards our goal of just understanding feasibility). Their characterization is summarized in the following theorem.

Theorem 3 (Theorem 4.5 [4]). *Let $\beta = rank_{\mathbb{F}_p}(M_{\equiv,\not\equiv})) - rank_{\mathbb{F}_p}(M_{\equiv})) > 0$. Then, we have:*

- *If $\beta = 0$, then there is no conversion from $(2,3)$-CNF sharing over $\mathbb{Z}_m$ to additive sharing over $\mathbb{Z}_p^\kappa$ with respect to C_{S_m}, for every $\kappa > 0$.*
- *If $\beta > 0$, then there is a conversion from $(2,3)$-CNF sharing over $\mathbb{Z}_m$ to additive sharing over $\mathbb{Z}_p^\beta$ with respect to C_{S_m}.*
  Furthermore, in this case, every row v of $M_{\not\equiv}$ is not spanned by the rows of $M_{\equiv}$.

Moreover, it was proven in [4] that the above is in fact equivalent to having the rows of $M_{\equiv}$ not span any row of $M_{\equiv}$. The latter simplification is a result of certain symmetry existing for the particular relation and secret sharing schemes in question.

Corollary 1. *For every row v of $M_{\not\equiv}$, v is not spanned by the rows of $M_{\equiv}$ iff there exists some $\beta > 0$ for which a conversion from $(2,3)$-CNF sharing over $\mathbb{Z}_m$ to additive sharing over $\mathbb{Z}_p^\beta$ with respect to C_{S_m} exists.*

Theorem 3 provides a full characterization via a condition that given (p_1, p_2, p) can be verified in polynomial time in $(p_1, p_2, \log(p))$. More precisely, the size of our matrix $M_{\equiv,\not\equiv}$ is $4m^2 \times 3m^2$ (in fact slightly smaller, at $(3m^2+1) \times 3m^2$, if working with Corollary 1, but this is insignificant), so verifying the condition amounts to solving a set of linear equations, which naively takes about $O(m^6)$ time, or slightly better using improved algorithms for matrix multiplication, and the running time cannot be better than $\tilde{\Omega}(m^4)$ using generic matrix multiplication algorithms. Thus, the complexity of verification grows very fast with m, becoming essentially infeasible for p_1, p_2 circa 50.

In any case, direct verification of the condition on concrete inputs does not answer the following fundamental question.

Question 1 (Informal)**.** *Do there exist infinitely many triples (p_1, p_2, p) for which a share conversion scheme from $(2,3)$-CNF to three-additive secret sharing (with parameters as discussed above) exists?*

It was conjectured in [4] that all tuples where $p = p_1$ and p_1, p_2, p are all odd allow for such a sharing scheme. We answer this question in the affirmative. While it is not clear that our result may be directly useful towards constructing better PIR schemes, our work is a first step towards possibly improving PIR complexity using this direction, in terms of the tools developed. See the discussion in Section 5 for more details.

To resolve this question, we develop an analytic understanding of the condition for the particular type of matrices at hand. Our goal is to simplify the matrices into a more human-understandable form, so we are able to verify directly the linear-algebraic condition for infinitely many parameter triples.

3.4. Some Notation

In this paper, we will consider matrices over some field $\mathbb{F}$, typically over a finite field $\mathbb{F} = \mathbb{Z}_p$. For matrices A, B with the same number of columns, $(A;B)$ denotes the matrix comprised by concatenating B below A. For matrices A, B with the same number of rows, we denote by $(A|B)$ the matrix obtained by concatenating B to the right of A. For entry i, j of a matrix A, we use the standard notation of $A[i,j]$. More generally, for a matrix $A \in \mathbb{F}^{u \times v}$, for subsets $R \subseteq [v]$ of rows and $C \subseteq [u]$ of columns, $A[R,C]$ denotes the sub-matrix with rows restricted to R and columns restricted to C (ordered in the original order of rows and columns in A). As special cases, using a single index i instead of R (C) refers to a single row (column). A "$\cdot$" instead of R (C) stands for $[u]$ ($[v]$).

We often consider imposing a block structure upon a matrix A. The block structure is specified by a grid defined by a partition of the columns into non-empty sets of consecutive columns $C_1, \ldots, C_t$ and a partition of the rows into non-empty sets of consecutive rows $R_1, \ldots, R_h$. The matrix A viewed as a block matrix is not a $t \times h$ matrix where entry (i,j) is the sub-matrix $A[C_i, R_i]$. We denote the block matrix obtained from A by V (for instance, a matrix named $A^{(3)}$ is replaced by $V^{(3)}$). In a block matrix V, we typically index the matrix by subscripts: $V_{i,j}$ denotes (the matrix at) entry i, j of V. For instance, $V_{i,j}[i,k]$ denotes entry $[i,k]$ in (sub)matrix $V_{i,j}$.

Typically, the C_i's and the R_i's are of the same size, and C_0, R_0 start at zero. Sometimes, the sets will not be of the same size (typically, the first or last set will be of a different size than the rest). Furthermore, most generally, C_0, R_0 may start elsewhere. Additionally, the indices may be consecutive modulo u (v), so one of the sets C_i (R_i) may not consist of truly consecutive indices.

Most of the time, index arithmetic will be done modulo the matrices' number of rows and columns (we will however state this explicitly).

4. Our Result

4.1. Starting Point and Main Technical Tool

Starting off with Corollary 1, it suffices to prove the following theorem.

We prove the following result.

Theorem 4. *Assume $m = p_1 \cdot p_2$, $p = p_1$, and $p_1, p_2 > 2$. Then, there exists a row v in $M_{\not\equiv}$ such that $Rows(M_{\equiv})$ does not span v.*

As a corollary from Corollary 1 and Theorem 4, we immediately obtain:

Corollary 2. *Assume $m = p_1 \cdot p_2$, $p = p_1$, and $p_1, p_2 > 2$. Then, there exists some $\beta > 0$ for which a conversion from $(2,3)$-CNF sharing over $\mathbb{Z}_m$ to additive sharing over $\mathbb{Z}_p^\beta$ with respect to C_{S_m} exists.*

To prove Theorem 4, we choose any vector v in the $M_{\equiv,\not\equiv}$, outside of $M_{\equiv}$, and prove that $M_{\equiv}$ does not span v. The particular choice of v we will make is a somewhat convenient choice, but any v will do, so we will fix it later. To this end, we apply carefully-chosen row operations, but in a specific way, so we (as humans) can understand the matrices that result.

Our main technical tool is the following simple lemma.

Lemma 1. *Let A denote a matrix in $\mathbb{Z}_p^{v \times u}$, and let $b = A[v, [u]]$. Let $I_1 \subseteq [v-1]$, $I_2 \subseteq [u]$ denote non-empty sets of rows and columns, respectively. A' obtained from A by a sequence of row operations on A, so that $A'[I_1, I_2]$ is a basis of $A'[[v], I_2]$, and the rest of the rows in $A'[I_1, I_2]$ are zero. Then, $Rows(A'[[v] \setminus I_1, [u] \setminus I_2])$ span $b'[[u] \setminus I_2]$ iff $Rows(A[[v-1], [u]])$ span $A[v, [u]]$.*

The proof of the Lemma follows by the fact that any solution to $xA[[v-1], [u]] = A[v, [u]]$ must have zero at indices corresponding to I_1, to obtain zero at the coordinates corresponding to v_2 (and the fact that row operations on $[A; b]$ do not change the solvability of a system $Ax = b$).

We start with $M'_{\equiv} = [M_{\equiv}; v]$ and need to resolve the question of whether $Rows(M_{\equiv})$ span v. On a high level, we proceed by applying the following lemma several times, thereby reducing the problem to considering a certain submatrix of the original matrix.

4.2. A Few Initial Elimination Sequences

4.2.1. Elimination Step 0

We introduce some more notation we will use along the way. We think of the matrix $M'_{\equiv,\not\equiv}$ as divided into blocks of 4×3. We denote constants by capital letters (e.g., a secret value S_1, or index (A, B)) and running indices by small letters, typically a, b, c. Row 1 of the block matrix V, $V_{i,\cdot}$ is indexed inside by (a, b, c), going over all (a, b, c) that constitute an additive sharing of $S_1 = (0,1)_{\mathbb{Z}_m}$. Here, $(0,1)_{\mathbb{Z}_m}$ denotes a single element of $\mathbb{Z}_m$, corresponding to its residues modulo p_1 and p_2, respectively. We omit the $\mathbb{Z}_m$ subscript when it is clear from the context whether a pair (x, y) is in $\mathbb{Z}_m$ or in $\mathbb{Z}_m$. Similarly, rows $V_{2,\cdot}, V_{3,\cdot}, V_{4,\cdot}$ correspond to $S_2 = (1,0)_{\mathbb{Z}_m}, S_3 = (1,1)_{\mathbb{Z}_m}, S_4 = (0,0)_{\mathbb{Z}_m}$, respectively. The rows inside rows 1, 2, 3 in the block matrix are internally indexed by (a, b) (c is determined by a, b, S_i

as $S_i - a - b$), where the (a, b)'s are ordered lexicographically (we stress that the particular order is not important for analyzing the rank of the matrix, but it will play some role in creating a matrix that "looks simple" and is easier to understand). The last block is indexed by some $(A, B) \in \mathbb{Z}_m \times \mathbb{Z}_m$ to be chosen later; as follows from Corollary 1, any (A, B) can be chosen.

Column $V_{\cdot,1}$ in the block matrix is internally indexed by the CNF-share (a, b) received by P_3. Column $V_{\cdot,2}$ is indexed by (a, c) (share received by P_2), and $V_{\cdot,3}$ is indexed by (b, c) (share received by P_3). As in the case of rows, the values $(x, y) \in \mathbb{Z}_m$ internally indexing a column of V are ordered lexicographically.

Pictorially, the matrix $M'_{\equiv}$, has the following general form.

	$(0,0)$ $\cdots$ $(m-1,m-1)$	$(0,0)$ $\cdots$ $(m-1,m-1)$	$(0,0)$ $\cdots$ $(m-1,m-1)$
$(0,0)$ $\cdots$ $(m-1,m-1)$	I	$V_{1,2}$	$V_{1,3}$
$(0,0)$ $\cdots$ $(m-1,m-1)$	I	$V_{2,2}$	$V_{2,3}$
$(0,0)$ $\cdots$ $(m-1,m-1)$	I	$V_{3,2}$	$V_{3,3}$
(A,B)	$\mathbf{e}_{A,B}$	$\mathbf{e}_{A,S_4-A-B}$	$\mathbf{e}_{B,S_4-A-B}$

In the above, I denotes the identity matrix, and each $V_{i,1}$ indeed equals I. The other $V_{i,j}$'s for $j \in \{2, 3\}$ are $m^2 \times m^2$ permutation matrices, and $e_{x,y} \in \mathbb{Z}_p^{m \times n}$ denotes a row vector with one at index (x, y), and zeroes elsewhere.

For a fixed $i \in [3]$, row (a, b) equals $(\mathbf{e}_{a,b}, \mathbf{e}_{a,S_i-a-b}, \mathbf{e}_{b,S_i-a-b})$ for Block Columns 1, 2, 3 respectively.

Next, we apply Lemma 1 to $A = M'_{\equiv}$ with $I_1 = [m]$, and $I_2 = [m^2]$ (I_2 corresponds to the first column in the block). The sequence of operations to b-zero the columns in $A[[3m^2 + 1], [m^2]]$ is carried out by subtracting from each row $\mathbf{e}_{a,b}$ (indexed by (a, b)) in Blocks 2, 3, 4 the corresponding row $\mathbf{e}_{a,b}$ (indexed by (a, b)) in Block 1. The resulting matrix A' (of size $(2m^2 + 1) \times 2m^2$) is described in the following.

4.2.2. Elimination Step 1

We view the matrix A' as a block matrix with the same subdivision into blocks as in $M'_{\equiv}$ (with one less block in both rows and columns, with Row 1 and Column 1 removed). Let V' denote the corresponding 3×2 block matrix.

Consider the row indexed by (a, b) in $V'_{i,\cdot}$ for $i \in [3]$ (corresponds to Block Rows 2, 3, 4 respectively in $M'_{\equiv}$). We have:

Observation 1. *The row $V'_{i,\cdot}$ indexed by (a, b) - $V'_{i,\cdot}[(a, b), \cdot]$ equals:*

$$(e_{a,S_i-a-b} - e_{a,S_1-a-b}; e_{b,S_i-a-b} - e_{b,S_1-a-b}) \tag{1}$$

Here, $(a, b) = (A, B)$ for $i = 3$.

The resulting matrix A' is depicted next.

$$
\begin{array}{c|cc|cc}
 & \binom{0,}{0} & \cdots & \binom{m-1,}{m-1} & \binom{0,}{0} \quad \cdots \quad \binom{m-1,}{m-1} \\
\binom{0,}{0} & & & & \\
\vdots & & V_{2,2}-V_{1,2} & & V_{2,3}-V_{1,3} \\
\binom{m-1,}{m-1} & & & & \\
\hline
\binom{0,}{0} & & & & \\
\vdots & & V_{3,2}-V_{1,2} & & V_{3,3}-V_{1,3} \\
\binom{m-1,}{m-1} & & & & \\
\hline
\binom{A,}{B} & & \mathbf{e}_{A,S_4-A-B}- & & \mathbf{e}_{B,S_4-A-B}- \\
 & & \mathbf{e}_{A,S_1-A-B} & & \mathbf{e}_{B,S_1-A-B}
\end{array}
$$

Next, we find a basis for $Rows(A[I_1,[m^2]])$ and apply Lemma 1 again, "getting rid" of the first block column in A'. For $i \in \{1,2,3,4\}$, let $\Delta_{i,j} = S_i - S_j$.

We will need another simple observation.

Observation 2. *The rows of $V'_{i,1}$ for $i \in [1,2]$ are a permutation of the sequence of all m^2 vectors of the form $(x,y)-(x,y+\Delta_{1,i+1})$. Additionally, $\Delta_{1,2} = (-1,1)_{\mathbb{Z}_m}$, and therefore, $(-1,1)_{\mathbb{Z}_m}$ generates $\mathbb{Z}_m$.*

We are now ready to demonstrate that the set of rows of $V'_{1,1}$, according to the coordinate of one in their row: $\mathcal{B} = \{(a,b) | c = S_2 - a - b \neq m-1\}$ constitutes the basis of $V'_{\cdot,I_2}$ for $I_2 = [m^2]$, as we seek. Therefore, we will again be able to apply Lemma 1. The other rows are spanned by this set, in one of two ways. We classify them into two types according to these ways. We refer to them as Type 1 and Type 2.

Type 1 constitutes of rows in $V'_{1,1}$ indexed by (a,b) with $c = S_2 - a - b = m-1$, for which the location of the '1' in $V_{1,1}[(a,b),\cdot]$ is (a,c) for c as above. We observe that every such row is spanned by $V'_{1,1}[\mathcal{B},\cdot]$, as for every $a \in \mathbb{Z}_m$, we have:

$$\sum_{b \in Z_m} V'_{1,1}[(a,b),\cdot] = \tag{2}$$

$$\sum_{c \in Z_m} \mathbf{e}_{a,c} - \mathbf{e}_{a,c+\Delta_{1,2}} = \tag{3}$$

$$\sum_{c \in Z_m} \mathbf{e}_{a,c} - \sum_{c \in Z_m} e_{a,c+\Delta_{1,2}} = \tag{4}$$

$$\mathbf{1} - \mathbf{1} = \mathbf{0} \tag{5}$$

Here, the transition from Equation (2) to Equation (3) is by Observation 1 and the observation that for a fixed a, $(S_2 - a - b)_{b \in \mathbb{Z}_m}$ is a permutation over $\mathbb{Z}_m$, so the change of coordinates we perform is valid. We b-zero rows (a,b) of $V'_{1,1}$ of Type 1 by adding all other rows of the form (a,b') to each of them.

Type 2 includes all rows in $V'_{i,1}$ for $i \in \{2,3\}$ are of this type. Consider a row indexed by (a,b) in $V'_{i,1}$ (in particular, the row in block $V'_{3,\cdot}$ is of that type). It is spanned by rows in $\mathcal{B}$ as follows:

$$\sum_{j=0}^{\frac{\Delta_{1,i+1}}{\Delta_{1,2}}-1} V'_{1,1}[(a, b+\Delta_{2,i+1}-j\cdot\Delta_{1,2}),\cdot] = \tag{6}$$

$$\sum_{j=0}^{\frac{\Delta_{1,i+1}}{\Delta_{1,2}}-1} (\mathbf{e}_{a,S_{i+1}-a-b+j\cdot\Delta_{1,2}} - \mathbf{e}_{a,S_{i+1}-a-b+(j+1)\cdot\Delta_{1,2}}) = \tag{7}$$

$$\mathbf{e}_{a,S_{i+1}-a-b} - \mathbf{e}_{a,S_{i+1}-a-b+\Delta_{1,i+1}} = V'_{i,1}[(a,b),\cdot] \tag{8}$$

Here, the transition to Equation (8) is by observing that a telescopic sum is formed, where all but the first and last $\mathbf{e}_{x,y}$'s in the sum cancel out. Note that the number in the superscript of the sum indeed exists, as $\Delta_{1,2}$ generates $\mathbb{Z}_m$, and so, in particular, $\Delta_{1,i+1} = k\cdot\Delta_{1,2}$ for some integer $k > 0$.

Rearranging the equality above, we get:

$$V'_{i,1}[(a,b),\cdot] - \sum_{j=0}^{\frac{\Delta_{1,i+1}}{\Delta_{1,2}}-1} V'_{1,1}[(a, b+\Delta_{2,i+1}-j\cdot\Delta_{1,2}),\cdot] = 0 \tag{9}$$

That is, we have identified the linear combinations of rows in $\mathcal{B}$ that add up to $V'_{i,1}$.

4.3. Elimination Step 2

After applying Lemma 1 to each row to Type 1 and Type 2 and making it zero, we turn to considering the remaining matrix $A'' \in Z_p^{(m+m^2+1)\times m^2}$. It is a block matrix of size 3×1, with blocks corresponding to $V'_{1,2}, V'_{1,3}, V'_{1,4}$ in A', with m, m^2, and one row, respectively. Let us calculate the form of rows of both types, when restricted to V'' (following elimination step 1 from Section 4.2.2).

Type 1 in V'': From Equation (2), we conclude that for a fixed a, for the (single) b such that $(a,b) \notin \mathcal{B}$, we have:

$$V''_{1,1}[(a,b),\cdot] = \tag{10}$$

$$\sum_{b\in\mathbb{Z}_m} V'_{1,2}[(a,b),\cdot] = \tag{11}$$

$$\sum_{b\in\mathbb{Z}_m} (\mathbf{e}_{b,S_2-a-b} - \mathbf{e}_{b,S_1-a-b}) \tag{12}$$

Type 2 in V'': This type consists of row blocks $i \in \{2,3\}$ in V''. By construction, we have:

$$V''_{i,1}[(a,b),\cdot] = V'_{i+1,2}[(a,b),\cdot] - \sum_{j=0}^{\frac{\Delta_{1,i+1}}{\Delta_{1,2}}-1} V'_{1,2}[(a, b+\Delta_{2,i+1}-j\cdot\Delta_{1,2}),\cdot] \tag{13}$$

Fix some $(a,b) \in \mathbb{Z}_m \times \mathbb{Z}_m$. Now, from Observation 1 and Equation (9), it follows that for all $(a,b) \in \mathbb{Z}_m \times \mathbb{Z}_m$, we have:

$$V''_{i,1}[(a,b),\cdot] = \tag{14}$$

$$V'_{i,2}[(a,b),\cdot] - \sum_{j=0}^{\frac{\Delta_{1,i+1}}{\Delta_{1,2}}-1} V'_{1,2}[(a,b+\Delta_{2,i+1}-j\cdot\Delta_{1,2}),\cdot] = \tag{15}$$

$$\begin{aligned}&\mathbf{e}_{b,S_{i+1}-a-b} - \mathbf{e}_{b,S_1-a-b}\\ &- \sum_{j=0}^{\frac{\Delta_{1,i+1}}{\Delta_{1,2}}-1} \left(\begin{array}{l} \mathbf{e}_{b+\Delta_{2,i+1}-j\cdot\Delta_{1,2},S_{i+1}-a-b-\Delta_{2,i+1}+j\cdot\Delta_{1,2}} \\ -\mathbf{e}_{b+\Delta_{2,i+1}-j\cdot\Delta_{1,2},S_1-a-b-\Delta_{2,i+1}+(j+1)\cdot\Delta_{1,2}} \end{array} \right)\end{aligned} \tag{16}$$

4.4. Permuting the Columns

Next, we permute the columns of A'', to gain insight into its form. Note that this does not affect the question of whether $A''[m+m^2+1]$ (last line) is spanned by the rest of the rows in A'', $Rows(A''[[m+m^2],[m^2]])$. We refer to the new matrix as $A^{(3)}$ and the resulting block matrix as $V^{(3)}$.

The permutation is as follows: the content of a column indexed by $(b,c) \in \mathbb{Z}_m \times \mathbb{Z}_m$ moves to $(b, b+c)/\Delta_{1,2}$ in the new matrix. We therefore obtain the following matrix M for Types 1 and 2, respectively (from Equations (12) and (16)).

4.4.1. Type 1

$$V^{(3)}_{1,1}[(a,b),\cdot] = \sum_{b\in\mathbb{Z}_m} \mathbf{e}_{(b,S_2-a)/\Delta_{1,2}} - \mathbf{e}_{(b,S_1-a)/\Delta_{1,2}} = \tag{17}$$

$$\sum_{b,c\in\mathbb{Z}_m} \mathbf{e}_{(b,c)/\Delta_{1,2}} - \mathbf{e}_{(b,c+\Delta_{1,2})/\Delta_{1,2}} = \tag{18}$$

$$\sum_{b,c\in\mathbb{Z}_m} \mathbf{e}_{b,c} - \mathbf{e}_{b,c+1} \tag{19}$$

Here, the last transition is due to the fact that dividing by $\Delta_{1,2}$ is a permutation over $\mathbb{Z}_m$ (as $\Delta_{1,2}$ is invertible).

4.4.2. Type 2

For $i \in \{2,3\}$, we have:

$$V^{(3)}_{1,i}[(a,b),\cdot] = \tag{20}$$

$$\mathbf{e}_{(b,S_{i+1}-a)/\Delta_{1,2}} - \mathbf{e}_{(b,S_1-a)/\Delta_{1,2}} \tag{21}$$

$$- \sum_{j=0}^{\frac{\Delta_{1,i+1}}{\Delta_{1,2}}-1} (\mathbf{e}_{(b+\Delta_{2,i+1}-j\cdot\Delta_{1,2},S_{i+1}-a)/\Delta_{1,2}} - \mathbf{e}_{(b+\Delta_{2,i+1}-j\cdot\Delta_{1,2},S_1-a+\Delta_{1,2})/\Delta_{1,2}}) =$$

$$\mathbf{e}_{b',c'} - \mathbf{e}_{b',c'+\Delta_{1,i+1}/\Delta_{1,2}} - \sum_{j=0}^{\frac{\Delta_{1,i+1}}{\Delta_{1,2}}-1} (\mathbf{e}_{(b'-j+\Delta_{2,i+1}/\Delta_{1,2},c')} - \mathbf{e}_{(b'-j+\Delta_{2,i+1}/\Delta_{1,2},c'+1)} = \tag{22}$$

$$\mathbf{e}_{(b',c')} - \mathbf{e}_{(b',c'+\Delta_{1,i+1}/\Delta_{1,2})} - \sum_{j=0}^{\frac{\Delta_{1,i+1}}{\Delta_{1,2}}-1} \mathbf{e}_{b'+j,c'} - \mathbf{e}_{b'+j,c'+1} \tag{23}$$

In Equation (21), we simply rename $b' = b/\Delta_{1,2}, c' = (S_{i+1} - a)/\Delta_{1,2}$. In Equation (22), we observe that for $j = \frac{\Delta_{1,i+1}}{\Delta_{1,2}}$, we have:

$$\begin{aligned} &b' - j + \Delta_{2,i+1}/\Delta_{1,2} = \\ &b' + 1 + \frac{\Delta_{2,i+1} - \Delta_{1,i+1}}{\Delta_{1,2}} = \\ &b' + 1 + \frac{\Delta_{2,1}}{\Delta_{1,2}} = b' + 1 - 1 = b' \end{aligned}$$

Thus, making a change of coordinates and letting the index j in the $\sum$ in Equation (23) run from $j = \frac{\Delta_{1,i+1}}{\Delta_{1,2}} - 1$ down to zero yield the expression in Line 23.

4.4.3. Summary

We have obtained a matrix with rows of the following form.

- Type 1 (contributed by Row Block 1 in $V^{(3)}$) yields all vectors of the form:

$$\sum_{b \in \mathbb{Z}_m} \mathbf{e}_{b,c} - \mathbf{e}_{b,c+1}$$

  for $c \in \mathbb{Z}_m$.
- Type 2 (contributed by Row Block 2 in $V^{(3)}$) yields all vectors of the form:

$$\begin{aligned} &\mathbf{e}_{(b,c)} - \mathbf{e}_{(b,c+\Delta_{1,3}/\Delta_{1,2})} - \sum_{j=0}^{\frac{\Delta_{1,3}}{\Delta_{1,2}}-1} \mathbf{e}_{b+j,c} - \mathbf{e}_{b+j,c+1} = \\ &\mathbf{e}_{(b,c)} - \mathbf{e}_{(b,c+(1,0)_{\mathbb{Z}_m})} - \sum_{j=0}^{(0,-1)_{\mathbb{Z}_m}} \mathbf{e}_{b+j,c} - \mathbf{e}_{b+j,c+1} = \end{aligned} \tag{24}$$

  For all pairs $b, c \in \mathbb{Z}_m$.
- The last line of $V^{(3)}$ is subsequently referred to as Type 3. Similarly to Type 2, it follows from Equation (23) that the line in the last block ($i = 3$) has the form:

$$\mathbf{e}_{(B,C)} - \mathbf{e}_{(B,C+(0,1)_{\mathbb{Z}_m})} - \sum_{j=0}^{(-1,0)_{\mathbb{Z}_m}} \mathbf{e}_{B+j,C} - \mathbf{e}_{B+j,C+1} \tag{25}$$

  Here, $B, C \in \mathbb{Z}_m$ is some pair of constants (corresponding to A, B to be chosen above) to be fixed later.

In the above, $(-1,0)_{\mathbb{Z}_m} = (0,1)_{\mathbb{Z}_m} - 1$ in the sum limit corresponds to "lifting" the element of $\mathbb{Z}_m$ into $\mathbb{Z}$ and viewing as an integer in $\{0, \ldots, m-1\}$. Note that no "wrap around" occurs as $(0,-1) \neq 0$ in $\mathbb{Z}_m$, so we get the correct number in the sum limit.

4.5. A Change of Basis

Occasionally, we will refer to rows $A^{(3)}$ as matrices, with rows indexed by b and columns by c.

As we have witnessed, so far, we have performed applications of Lemma 1 on the original matrix A, which was a simple block matrix of size 4×3 blocks, and the upper three row blocks had three cells of size $\mathrm{m}^2 \times \mathrm{m}^2$ each and another "block" row with three cells of size $1 \times \mathrm{m}^2$ each cell. By performing two applications of the lemma, we obtained a much smaller matrix $A^{(3)}$. Namely, we obtained a block matrix with 3×1 blocks, with cells of size $\mathrm{m} \times \mathrm{m}^2$, $\mathrm{m}^2 \times \mathrm{m}^2$, and $1 \times \mathrm{m}^2$, respectively. The price

we have paid for the reduction in size is that the structure of the matrix has became more complex. In particular, it is no longer clear how to identify additional row sets I_1 to continue applying the lemma conveniently.

To create a matrix of more manageable form, we perform a change of basis. We suggest the following basis for the row space of $A^{(3)}$. $B = B_1 \cup B_2$. Here:

$$B_1 = \{\mathbf{e}_{b,i} - \mathbf{e}_{b,i+1} | b \in \mathbb{Z}_m, i \in \{0, \ldots, p_2 - 2\}, \}$$

and:

$$B_2 = \{-\mathbf{e}_{b,i+j\cdot(1,0)_{\mathbb{Z}_m}} + \mathbf{e}_{b,i+(j+1)\cdot(1,0)_{\mathbb{Z}_m}} | b \in \mathbb{Z}_m, i \in Z_{p_2}, j \in \mathbb{Z}_{p_1} \setminus \{p_1 - 1\}\}$$

In all indices here and elsewhere, the arithmetic is over $\mathbb{Z}_m$.

Indeed $B_1 \cup B_2$ is a basis of $Rows(A^{(3)})$. To prove this, we first note that it is an independent set. Roughly, this follows by separately considering the vectors in B with nonzero values in each row $(b, \cdot)$ separately and the fact that p_2 divides m.

Next, we show that the rows of $A^{(3)}$ are indeed spanned by the above set of vectors. Furthermore, the matrix re-written in this basis will have a nice block structure that we will be able to exploit for the purpose of using Lemma 1.

We denote by $T_{b,i} = \mathbf{e}_{b,i} - \mathbf{e}_{b,i+1}$ a vector in B_1 and by $R_{b,i+j(1,0)_{\mathbb{Z}_m}} = -\mathbf{e}_{b,i+j\cdot(1,0)_{\mathbb{Z}_m}} + \mathbf{e}_{b,i+(j+1)\cdot(1,0)_{\mathbb{Z}_m}}$ a vector in B_2.

To rewrite our matrix $A^{(3)}$, we will specify an ordering of the vectors in B, from left to right

1. The columns in B_1 come first, in increasing order of c, starting from zero.
2. The columns of the form $R_{b,i+j(1,0)}$ in B_2 are ordered on several levels:
   (a) On the highest level, order $R_{b,i+j\cdot(1,0)}$'s according to the index i above, starting from zero up to $p_2 - 1$. There are p_2 blocks of this form on the highest level.
   (b) For a fixed i, order $R_{b,i+j\cdot(1,0)}$'s according to increasing order of j starting from zero, up to $p_1 - 2$.
   (c) For fixed i, j, order in increasing order of $b's$, starting from zero up to $m - 1$.

We order the rows of the matrix as follows:

1. The rows of Type 2 appear first, then Type 1, then Type 3 (the distinguished row to span via the others).
2. Within Type 1, we denote $\mathbf{r}^1_c = \sum_{b\in\mathbb{Z}_m} \mathbf{e}_{b,c} - \mathbf{e}_{b,c+1}$.
3. Within Type 2, denote by $\mathbf{r}^2_{b,c}$ the vector $\mathbf{e}_{(b,c)} - \mathbf{e}_{(b,c+(1,0)_{\mathbb{Z}_m})} - \sum_{j=0}^{(0,-1)_{\mathbb{Z}_m}} \mathbf{e}_{b+j,c} - \mathbf{e}_{b+j,c+1}$. Consider one such vector, $\mathbf{r}^2_{b,c}$, We order these vectors according to i, then j, then b, where $c = i + j \cdot (1, 0)$. Similarly, for Type 3, denote $\mathbf{r}^3_{B,C} = \mathbf{e}_{(B,C)} - \mathbf{e}_{(B,C+(0,1)_{\mathbb{Z}_m})} - \sum_{j=0}^{(-1,0)_{\mathbb{Z}_m}} \mathbf{e}_{B+j,B} - \mathbf{e}_{B+j,C+1}$.

Let us now study the form of the resulting matrix, divided into blocks, as follows from the representation of the various vectors in $Rows(A^{(3)})$ in basis B.

Let us represent the matrix as a block matrix, then we further break down each block into lower level blocks as follows. Let us denote the new matrix by $A^{(4)}$.

The Type 2 set of rows in $A^{(4)}$ has structure as depicted in the following matrix. $A^{(4)} = (A^{(4),2}; A^{(4),1}, A^{(4),3})$. Let us describe each of the matrices $A^{(2),i}$ below.

4.5.1. The Matrix $A^{(4),2}$

$$A^{(4),2} = (A^{(4),L,2} | A^{(4),R,2})$$

Here, the "right side" $A^{R,2}$ is a $p_2 \times p_2$ block matrix. Its contents are as follows.

	0	1	2	$\cdots$	p_2-2	p_2-1
0	$R_1^1+R_1^2$	$-R_1^2$				
1		$R_1^1+R_1^2$	$-R_1^2$			
$\vdots$				$\ddots$		
p_2-2					$R_1^1+R_1^2$	$-R_1^2$
p_2-1	$R_1^3-R_1^2$					$R_1^1+R_1^2$

The left-side matrix $A^{(4),L,2}$ is a block matrix of size $p_2 \times 1$ (where indeed the number of rows in each of its p_2 blocks is consistent with that of $A^{(4),R,2}$). It has the following structure.

0	L_1^0
1	L_1^1
$\vdots$	$\vdots$
p_2-2	$L_1^{p_2-2}$
p_2-1	$\sum_{i=0}^{p_2-2} L_1^i$

We refer to this partition into $p_2 \times (p_2+1)$ blocks of $A^{(4),2}$ as the "Level-1" partition of $A^{(4),2}$.

We continue next with describing the "Level-0" detail of $R_2^1, R_2^2, R_2^3, L_2^i$ of $A^{(4),R,2}, A^{(4),L_2}$. The matrix L_1^i for $i \in \{0,\ldots,p_2-2\}$ is a $p_1 \times (p_2-1)$ block matrix of the following form:

	0	1	$\cdots$	i	$\cdots$	p_2-2
0				T_0		
1				T_0		
$\vdots$				$\vdots$		
p_1-2				T_0		
p_1-1				T_0		

for a matrix T_0 to be specified later. Note that by the structure of $A^{(4),L,2}$, the last matrix $L_1^{p_2-1}$ is a block matrix of size $p_1 \times (p_2-1)$, where each block equals $-T_0$.

The matrix R_1^1 is a $p_1 \times (p_1-1)$ matrix of the following form.

	0	1	$\cdots$	p_1-2
0	I			
1		I		
$\vdots$			$\ddots$	
p_1-2				I
p_1-1	$-I$	$-I$	$\cdots$	$-I$

Here, I is the $m \times m$ identity matrix. The matrix R_1^2 is a $p_1 \times (p_1-1)$ matrix of the following form.

	0	1	$\cdots$	p_1-3	p_1-2
0					
1	T_0				
2	T_0	T_0			
$\vdots$	T_0	T_0	$\ddots$		
p_1-2	T_0	T_0	$\cdots$	T_0	
p_1-1	T_0	T_0	$\cdots$	T_0	T_0

Here, T_0 is a $m \times m$ matrix, as above. Finally, the matrix R_2^3 is a $p_1 \times (p_1 - 1)$ block matrix of the following form.

	0	$\cdots$	$p_2 \mod p_1$	$\cdots$	$p_1 - 2$
0	$-T_0$	$\cdots$	$-T_0$		
$\cdots$	$\vdots$	$\ddots$	$\vdots$		
$p_1 - 1$	$-T_0$	$\cdots$	$-T_0$		

It remains to specify T_0. It is a $m \times m$ circulant matrix of the following form.

	0	$\cdots$	$(1,0)_{\mathbb{Z}_m} - 2$	$(1,0)_{\mathbb{Z}_m} - 1$	$(1,0)_{\mathbb{Z}_m}$	$\cdots$	$m-1$
0	1	$\cdots$	1	1			
		1	$\cdots$	1	1		
$\vdots$			$\ddots$	$\ddots$	$\ddots$	$\ddots$	
$m-1$	1	$\cdots$	1				1

4.5.2. The Matrix $A^{(4),3}$

We choose $(B, C) = (0, 0)$. Then, the line is of the form:

$$\sum_{b=1}^{(0,1)_{\mathbb{Z}_m}-1} T_{b,0} + \sum_{j=0}^{p_1-2} R_{0,1+j\cdot(1,0)_{\mathbb{Z}_m}}$$

4.5.3. The Matrix $A^{(4),1}$

The matrix $A^{(4),1}$ is of the form $A^{(4),1} = (A^{(4),L,1}; A^{(4),R,1})$. To describe the left and right parts, we apply a certain transformation to $A^{(4),L,2}$ and $A^{(4),R,2}$, respectively. First, view each as a block matrix comprised of blocks of size $m \times m$ ($A^{(4),L,2}$ has $m \times (p_2 - 1)$ blocks, and $A^{(4),R,2}$ has $m \times p_2(p_1 - 1)$). Now, $A^{(4),L,1}$ is obtained from $A^{(4),L,2}$ by applying a linear mapping satisfying $l(T_0) = \underbrace{(1, \ldots, 1)}_{m \text{ times}}$ to each block X, replacing X by $m(X)$ (it is not important how exactly we define it on other inputs). The matrix $A^{(4),L,1}$ is obtained from $A^{(4),L,1}$ by replacing each block by a linear transformation that maps I to the zero vector and T_0 to $\underbrace{(1, \ldots, 1)}_{m \text{ times}}$. That is, the resulting $A^{(4),L,1}$ equals:

0	$\tilde{L}_1^0$
1	$\tilde{L}_1^1$
$\vdots$	$\cdots$
$p_2 - 2$	$\tilde{L}_1^{p_2-2}$
$p_2 - 1$	$\sum_{i=0}^{p_2-2} \tilde{L}_1^i$

where $\tilde{L}_1^i$ equals:

	0	1	$\cdots$	i	$\cdots$	$p_2 - 2$
0				$(1, \cdots, 1)$		
1				$(1, \cdots, 1)$		
$\vdots$				$\vdots$		
$p_1 - 2$				$(1, \cdots, 1)$		
$p_1 - 1$				$(1, \cdots, 1)$		

The resulting matrix $A^{(4),R,1}$ equals:

	0	1	2	$\cdots$	p_2-2	p_2-1
0	$\tilde{R}_1^2$	$-\tilde{R}_1^2$				
1		$\tilde{R}_1^2$	$-\tilde{R}_1^2$			
$\vdots$				$\ddots$		
p_2-2					$\tilde{R}_1^2$	$-\tilde{R}_1^2$
p_2-1	$\tilde{R}_1^3-\tilde{R}_1^2$					$\tilde{R}_1^2$

where the resulting $\tilde{R}_1^2$ is of the form:

	0	1	$\cdots$	p_1-3	p_1-2
0					
1	$(1,\ldots,1)$				
2	$(1,\ldots,1)$	$(1,\ldots,1)$			
$\vdots$	$(1,\ldots,1)$	$(1,\ldots,1)$	$\ddots$		
p_1-2	$(1,\ldots,1)$	$(1,\ldots,1)$	$\cdots$	$(1,\ldots,1)$	
p_1-1	$(1,\ldots,1)$	$(1,\ldots,1)$	$\cdots$	$(1,\ldots,1)$	$(1,\ldots,1)$

4.6. Another Elimination Sequence

From now on, assume that $p = p_1$ and that $p_2 > 2$. We leave the full analysis of other cases for future work. We are now able to apply Lemma 1. We perform this step for I_2 corresponding to the L-part blocks of $A^{(4)}$ and proceed in several steps. We perform the row operations starting at a grosser resolution and then proceed to finer resolution.

4.6.1. Step 1: Working at the Resolution of Level-1 Blocks

View $A^{(4),2}$ as a block matrix of Level-1 as described above. Let $V^{(4),2}$ denote the corresponding block matrix. Replace the last row of $V^{(4),2}$, $V^{(4),2}[p_1-1,\cdot]$ by $\sum_{i=0}^{p_2-1} V^{(4),2}[i,\cdot]$. We thus obtain a new matrix $A^{(5),2}$ of the following form. $A^{(5),2} = (A^{(5),L,2}|A^{(5),R,2})$ has the same block structure as $A^{(4),2}$ on all levels, so we do not repeat that, but rather only review its content.

The resulting right side $A^{(5),R,2}$ is as follows.

	0	1	2	$\cdots$	p_2-2	p_2-1
0	$R_1^1+R_1^2$	$-R_1^2$				
1		$R_1^1+R_1^2$	$-R_1^2$			
$\vdots$				$\ddots$		
p_2-2					$R_1^1+R_1^2$	$-R_1^2$
p_2-1	$R_1^3+R_1^1$	R_1^1	R_1^1	$\cdots$	R_1^1	R_1^1

The resulting left-side matrix $A^{(5),L,2}$ is:

0	L_1^0
1	L_1^1
$\vdots$	$\vdots$
p_2-2	$L_1^{p_2-2}$
p_2-1	0

We perform a similar transformation on $A^{(4),1}$, resulting in:

$$A^{(5),1} = (A^{(5),R,1}|A^{(5),L,1})$$

where $A^{(5),R,1}$ equals:

	0	1	2	$\cdots$	p_2-2	p_2-1
0	$\tilde{R}_1^2$	$-\tilde{R}_1^2$				
1		$\tilde{R}_1^2$	$-\tilde{R}_1^2$			
$\vdots$				$\ddots$		
p_2-2					$\tilde{R}_1^2$	$-\tilde{R}_1^2$
p_2-1	$\tilde{R}_1^3$					

and $A^{(5),L,1}$ equals:

0	$\tilde{L}_1^0$
1	$\tilde{L}_1^1$
$\vdots$	$\vdots$
p_2-2	$\tilde{L}_1^{p_2-2}$
p_2-1	0

4.6.2. Step 2: Working at the Resolution of Level-0 Blocks

Here, we view the matrix $A^{(5)}$ as a block matrix over Level-0 blocks. That is, denote by (i,j) the row block corresponding to the j^{th} Level-0 block inside the i^{th} Level-1 block of $A^{(6)}$. We transform $A^{(5)}$ into a matrix $V^{(6)}$ as follows.

For each $i \in \{0,\ldots,p_2-2\}, j \in \{1,\ldots,p_1\}$, replace each row in $V^{(5),2}[(i,j),\cdot]$ with $V^{(5),2}-[(i,0),\cdot]$. The resulting matrix $A^{(6),2}$ is of the form $A^{(6),2}=A^{((6),L,2}|A^{(6),R,2})$.

The right side $A^{(6),R,2}$ is as follows.

	0	1	2	$\cdots$	p_2-2	p_2-1
0	$R_1^4+R_1^2$	$-R_1^2$				
1		$R_1^4+R_1^2$	$-R_1^2$			
$\vdots$				$\ddots$		
p_2-2					$R_1^4+R_1^2$	$-R_1^2$
p_2-1	$R_1^3+R_1^1$	R_1^1	$\cdots$		R_1^1	R_1^1

where R_1^4 equals:

	0	1	$\cdots$	p_1-2
0	I			
1	$-I$	I		
$\vdots$	$\vdots$		$\ddots$	
p_1-2	$-I$			I
p_1-1	$-2I$	$-I$	$\cdots$	$-I$

The resulting left-side matrix $A^{(6),L,2}$ is:

0	$L_1^{-,0}$
1	$L_1^{-,1}$
$\vdots$	$\vdots$
p_2-2	L_1^{-,p_2-2}
p_2-1	0

where $L_1^{-,i}$ is of the form:

	0	1	$\cdots$	i	$\cdots$	p_2-2
0				T_0		
1						
$\vdots$						
p_1-2						
p_1-1						

Finally, we apply a similar transformation to $A^{(5),1}$ resulting in $A^{(6),L,1}$ that equals:

0	$\tilde{L}_1^{-,0}$
1	$\tilde{L}_1^{-,1}$
$\vdots$	$\vdots$
p_2-2	$\tilde{L}_1^{-,p_2-2}$
p_2-1	0

where $\tilde{L}_1^{-,i}$ is of the form:

	0	1	$\cdots$	i	$\cdots$	p_2-2
0				$(1,\ldots,1)$		
1						
$\vdots$						
p_1-2						
p_1-1						

The resulting right-hand side is $A^{(6),R,1} = A^{(5),R,1}$ (there is no change, since the first block-row in $V^{(5),R,1}$ is zero).

4.6.3. Step 3: Working within Level-0 Blocks

Here, we move to working with individual rows and complete the task of leaving a basis of the original $A^{(4),L}$'s rows as the set of non-zero rows of the matrix $A^{(7),L}$ obtained by a series of row operations. To this end, our goal is to understand the set of remaining rows in $A^{(6),L}$. In the $A^{(6),L,2}$ part, each Level-0 block-column (with blocks of size $m \times m$) has only $G = Rows(T_0) \cup \{(1,\ldots,1)\}$ (here, one appears m times) as non-zero rows, and in each row, there are non-zero entries in only one of the blocks. Thus, it suffices to find a basis for the set G of vectors.

Lemma 2. *Assume $p = p_1$. Then, the index set $I = \{k | 0 \leq k \leq (p_1-1)p_2\}$ satisfies that $Rows(T_0[I,[m]])$ is a basis for G. In particular, we have $\sum_{j=0}^{p_1-1} T_0[j \cdot p_2,[m]] = x \cdot (1,\ldots,1)$. Here, x is computed as follows: first calculate y as p_2^{-1} modulo p_1 (that is, in $\mathbb{Z}_{p_1}$). Then, we "lift" y back into $\mathbb{Z}$ ($1 \leq y \leq p_1-1$) and then set x to be y modulo p – that is, x is an element of $\mathbb{Z}_p$ (note that all non-zero coefficients in the linear combination that results in $(1,\ldots,1)$ indeed belong to I).*

Another observation that will be useful to us identifies the dual of T_0.

Lemma 3. *Assume $p = p_1$. Then, the set of vectors:*

$$S = \{ \sum_{j=0}^{p_1-1} \mathbf{e}_{j \cdot p_2} - \mathbf{e}_{i+(j+1) \cdot p_2} | i \in \mathbb{Z}_{p_2} \setminus \{0\} \}$$

is a basis of $Ker(T_0)$, where $Ker(M) = \{v | v \cdot M\}$ denotes the left kernel of the matrix M.

The observations are rather simple to prove by basic techniques; see Appendix A. Note that the general theory of cyclotomic matrices is not useful here, as it holds over infinite or large (larger than matrix size) fields, so we proceed by ad-hoc analysis of the (particularly simple) matrices at hand.

We handle the $A^{(7),2}$ part first. We conclude from Lemma 2 that for every block specified by (i, j) where $i \neq p_2 - 1$, in $V^{(6),L,2}[(i, j), \cdot]$, the rows indexed by $b \in I$ (as in Lemma 2) span all rows in that block. Furthermore, for the purpose of Lemma 1, we b-zero the rest of the rows, by a sequence of row operations as specified by $Ker(T_0)$ in Lemma 3, starting from row $(p_1 - 1)p_2 + 1$ and moving forward up to $m - 1$. That is, for b -zeroing row $(p_1 - 1)p_2 + k$ (where $k > 0$) in $V^{(6),L,2}[(i, j), \cdot]$ as above, we store the combination:

$$\sum_{h=0}^{p_1-1} V^{(6),1}[(i, j, k + h \cdot p_2), \cdot] - V^{(6),1}[(i, j, k + (h + 1) \cdot p_2), \cdot]$$

in row (i, j, k) of $A^{(7),2}$.

Overall, the resulting $A^{(7),2}$ is as follows:

$A^{(7),R,2}$ is identical to $A^{(6),R,2}$, except for replacing R_1^4 with R_1^5.

That is, in the last block row R_1^5, all cells are R_{-1}^2, and there are p_1 such cells.

Here, R_{-1}^2 is of the form:

	0	1	2	$\cdots$	$p_2 - 2$	$p_2 - 1$
1	1	-1				
2	1		-1			
$\vdots$				$\ddots$		
$p_2 - 2$	1				-1	
$p_2 - 1$	1					-1

	0	$\cdots$	$p_2 - 1$	$\cdots$	$2p_2 - 1$	$\cdots$	$m - p_2$	$\cdots$	$m - 1$
0	I								
$\vdots$									
$m - p_2$									
$m - p_2 + 1$	R_{-1}^2			R_{-1}^2		$\cdots$	R_{-1}^2		
$\vdots$									
$m - 1$									

Next, we handle the $A^{(7),1}$ part. Here, we b-zero the remaining rows in $A^{(6),L,1}$ by adding the right combination of rows in $A^{(6),L,2}$. The combination is determined by the "in particular" part of Lemma 3.

The resulting matrix $A^{(7),L,2}$ is identical to $A^{(6),L,2}$, except for T_0 in each $L_1^{-,i}$ being replaced by T_0'. Here, T_0' has the form:

	0	1	$\cdots$	$(1,0)_{\mathbb{Z}_m} - p_2 - 1$	$\cdots$	$(1,0)_{\mathbb{Z}_m} - 1$	$(1,0)_{\mathbb{Z}_m}$	$\cdots$	$m - p_2$	$\cdots$	$m-1$
0	1	1		$\cdots$		1					
$\vdots$		1		$\cdots$			1				
					$\vdots$						
$m - p_2$	1	1	$\cdots$	1					1	$\cdots$	1
$m - p_2 + 1$											
$\vdots$											
$m-1$											

Here, $A^{(7),L,1}$ becomes zero, which was our goal. Note that as opposed to previous transformations, the transformation performed on $A^{(6),L,1}$ does not "mirror" the transformation performed on $A^{(6),L,2}$ and in fact involves rows from both $A^{(6),L,2}$ and $A^{(6),L,1}$. $A^{(7),R,1}$ is identical to $A^{(6),R,1}$, except that in each Level-1 block (i,i) for $i \in \{0,\ldots,p_2-2\}$, the first row of $\tilde{R}_1^2$ (the content of this block) is replaced by:

$$-\sum_{i=0}^{p_1-1} x\mathbf{e}_{i\cdot p_2}.$$

It remains to b-zero the L-part of $A^{(6),3}$. For simplicity, we focus on $V^{(6),L,3}[0,0]$ (which is the only non-zero block in $V^{(6),L,2}$) and then use the resulting linear dependence to produce the new row $V^{(6),3}[0,\cdot]$.

$$V^{(6),L,3}[0,0] = x\sum_{i=0}^{p_1-1} V^{(6),L,2}[(0,0,i\cdot p_2),0] - V^{((6),L,2)}[(0,0,(-1,0)_{\mathbb{Z}_m}+1),0]$$

This results in:

$$A^{(7),R,3} = -x\sum_{i=0}^{p_1-1} \mathbf{e}_{(0,0,i\cdot p_2)} + \mathbf{e}_{(0,0,(-1,0)_{\mathbb{Z}_m})} + \sum_{b=1}^{(0,1)_{\mathbb{Z}_m}-1} T_{b,0}[R] + \sum_{j=0}^{p_1-2} R_{0,1+j\cdot(1,0)_{\mathbb{Z}_m}}[R] \tag{26}$$

4.6.4. A Reduced Matrix We Will Analyze Directly

Taking I_1 to be the set of rows in $A^{(7)}$ that correspond to non-zero rows in $A^{(7),L,1}$ and I_2 corresponding to L, we obtain the following matrix $A^{(8)}$. On Level-1, it has a block structure similar to that of $A^{(7),R}$ (where the number of rows changes in some of the matrices). More concretely, $A^{(8),2}$ has the form:

	0	1	2	$\cdots$	p_2-2	p_2-1
0	$R_1^{5,-}+R_1^{2,-}$	$-R_1^{2,-}$				
1		$R_1^{5,-}+R_1^{2,-}$	$-R_1^{2,-}$			
$\vdots$				$\ddots$		
p_2-2					$R_1^{5,-}+R_1^{2,-}$	$-R_1^{2,-}$
p_2-1	$R_1^3+R_1^1$	R_1^1	$\cdots$		R_1^1	R_1^1

Here, $R_1^{5,-}$ is identical to R_1^5 except that the top $m - p_2 + 1$ rows in it are removed. That is, it is identical to R_1^4, except that the $(0,0)^{\text{th}}$ Level-0 block in R_1^4 replaces I by C, which are equal.

$$\begin{array}{r|} m-p_2+1 \\ \hline \vdots \\ \hline m-1 \end{array} \left(\begin{array}{|c|c|c|c|} \hline R^2_{-1} & R^2_{-1} & \cdots & R^2_{-1} \\ \hline \end{array} \right)$$

Similarly, $R_1^{2,-}$ is obtained from R_1^2 in the same manner. In this case, only zero rows are removed. $A^{(8),1}$ is precisely $A^{(7),R,1}$ (no rows were eliminated from there, as all corresponding rows on the left side became zero). Similarly, $A^{(8),3} = A^{(7),R,3}$.

4.7. Completing the Proof - Analysis of $A^{(8)}$

We are now ready to make our conclusion, assuming $p = p_1$ and $p_1, p_2 > 2$. We stress that further analysis of the matrix is needed for identifying all p's for which a share conversion exists. In fact, some of the detailed calculations of the resulting matrix structure are not needed for our conclusion, and we could instead identify only the properties that we need of various sub-matrices. However, some of the details may be useful for future analysis, so we made all the calculations.

Our last step is to reduce the matrix $A^{(8)}$ "modulo" the set G: for every row r in $A^{(8)}$ and every Level-0 block in this row, we reduce the contents of that row "modulo" $span(Rows(T_0))$. That is, we complement the basis of $Rows(T_0)$ specified in Lemma 2 into a basis of $\mathbb{Z}_p^m$, where $\mathbf{e}_0$ is one of the added vectors and define a linear mapping L taking elements of $Rows(T_0[I, \cdot])$ to zero and other elements of the basis onto themselves (it is inconsequential what the other base elements are). Indeed, observe that $\mathbf{e}_0$ is not in $span(Rows(T_0))$, as it is not in $Ker(Ker(span(Rows(T_0))))$, as implied by Lemma 2. To verify this, observe for instance that:

$$< \sum_{i=0}^{p_1-1} (\mathbf{e}_{i \cdot p_2} - \mathbf{e}_{1+i \cdot p_2}), \mathbf{e}_0 >= 1 \neq 0.$$

We apply a linear mapping L taking $x \in span(Rows(T_0))$ to $\mathbf{0}$ and other base elements to themselves. Recall that Level-0 blocks indeed have m columns each. We make the following observations. We let $A^{(9)}$ denote the resulting matrix.

Observation 3. *The rows of $A^{(9),1}$ are zero.*

Observation 4. *$A^{(9),3}$ maps to $\sum_{b=1}^{(0,1)_{\mathbb{Z}_m}-1} T_{b,0}[R] + \sum_{j=0}^{p_1-2} R_{0,1+j\cdot(1,0)_{\mathbb{Z}_m}}[R]$.*

Observation 3 follows easily from the form of the matrix $A^{(8)}$ and Lemmas 2 and 3, which implies that span$(Rows(T_0))$ is exactly the kernel of S from Lemma 2 (this is the reason we need Lemma 2: it is easier to verify that a given vector is not in $Ker(span(S))$, rather than verifying it is not in $span(T_0)$).

Observation 4 follows by the structure of $A^{(8)}$ and definition of L.

Now, if $A^{(8),3}$ is spanned by the rest of the rows in $A^{(8)}$, then it must be the case that the same dependence exists in $A^{(9)}$. Thus, it suffices to prove that the latter does not hold. Assume for the sake of contradiction that:

$$v(A^{(9),1}; A^{(9),2}) = A^{(9),3}$$

for some vector v. In the following, we use V^0 for viewing v as a block vector with Level-0 blocks. Note that unusually for this type of matrix, the blocks in the first row have $p_2 - 1$ rows, and in other block row, the cells have m rows, as usual. Similarly, we use V^1 to impose Level-1 structure onto v.

By the structure of $R_1^{2,-}$, we conclude that $A^{(9),2}$ is of the form:

	0	1	2	$\cdots$	p_2-2	p_2-1
0	$R_1^{5,-}$					
1		$R_1^{5,-}$				
$\vdots$				$\ddots$		
p_2-2					$R_1^{5,-}$	
p_2-1	R_1^1	R_1^1	$\cdots$		R_1^1	R_1^1

Observe that in $A^{(9),3}$, non-zero values exist only in Level-1 blocks $i = 0, 1$. As there are p_2 such blocks and $p_2 > 2$, by our assumption, we conclude that the last row contributes zero to $v(A^{(9),1}; A^{(9),2})$, as in the last block, the output needs to be zero, and it equals $V[p_2]A^{(9),2}$, which is the same contribution for all (Level-1) blocks.

To agree with $A^{(9),3}$ at block i, we must then have $v \cdot R_1^{5,-} = \mathbf{e}_0$. Viewing $R_1^{5,-}$ as a block-matrix of Level00, because of the zeroes at all blocks but Block 0, the contributions of all block-rows but the first one to $v \cdot R_1^{5,-}$ is:

$$-(2 + (p_1 - 2)) \cdot V^1[p_2] = -p_1 \cdot V^1[0] = 0.$$

In the above, the last equality is due to the fact that $p = p_1$. Thus, we must have $V^1[0] \cdot C = \mathbf{e}_0$. However, we observe that $Rows(C)$ is a subset of $Ker(\text{span}(S))$, where S is specified in Lemma 3 and thus cannot equal $\mathbf{e}_0$ (which is not in $Ker(\text{span}(S))$).

This concludes the proof of Theorem 4.

5. Future Directions

Our work leaves several interesting problems open.

- For what other parameters is a share conversion from $(2,3)$-CNF to three-additive possible? For instance, surprisingly, $p \neq p_1, p_2$ is possible for $(m = 7 \cdot 73, p = 2)$ as follows from [10]. Our analysis does not explain this phenomenon, as we did not complete the full analysis of the resulting matrix. We believe that given the work we have done, this is a very realistic goal.
- We need to understand share conversion for different sets S. One direction is by considering m's, which are a product of more than two primes. As discussed in the Introduction, already using three primes, a conversion from (2,3)-CNF over S_m would improve over the best known constructions for three-server PIR via the BIKO paradigm. One advantage of such schemes following the BIKO framework is the constant answer size achieved. Here, we initially worked with two primes, rather than with three, to develop the tools and intuition for a slightly technically simpler setting.
- As discussed in the Introduction, some of the previous results not falling in the BIKO framework can be viewed as instances of an extended BIKO framework using a "many-to-one" share conversion. Viewing PIR protocols as based on share conversions between secret sharing schemes is apparently "the right" way to look at it. As certain evidence, using the more redundant CNF instead of Shamir as in [10] was a useful insight from secret sharing allowing us to further improve CC. In particular, in the case that there are no share conversions for C_{S_m} for m's that are a product of three or more primes, perhaps a suitable many-to-one conversion may still exist.

  Further extending this view could lead to new insights on PIR design. In particular, in many of the existing schemes, a shallow circuit is evaluated essentially by performing share conversions. In particular, local evaluation of linear functions over the inputs is a special case of such a many-to-one conversion (from a linear scheme to itself). In all schemes we have surveyed here, PIR for a large family $\mathcal{F}$ of functions (of size 2^{2^n}) was implemented via share conversion for a small set of relations. For instance, BIKO's 3-PIR was based on 2^n linear functions (implementing

inner products with vectors in the MV family) and a share conversion for C_{S_m}, thus only $O(2^n)$ relations. Similarly, in the two-server PIR, the situation is similar.

One concrete direction may be proving lower bounds on PIR protocols based on "circuits" containing only certain share conversion "gates". Perhaps analogies to circuit complexity could be made, borrowing techniques from circuit lower bounds. Insights for such limited classes of schemes could hopefully advance our understanding of lower bounds for PIR CC; currently, the best-known lower bound even for two-server PIR is $5n$, only slightly above trivial [15].

Author Contributions: The authors contributed equally in this paper.

Funding: This research received no external funding.

Acknowledgments: We thank Roi Benita for greatly simplifying the transition from A' to A'' at an earlier step of working on this project.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Deferred Proofs

Appendix A.1. Proof of Lemma 2

We start with a few Claims to be used by the proof.

Lemma A1. *Let p be a prime and $0 < x < p$ an integer. Let $T_0^{(p)}$ denote a matrix in $\mathbb{Z}^{p\times p}$ where the ith row is:*

$$[T_0^{(p)}[i,\cdot] = \sum_{j=0}^{x-1} \mathbf{e}_{i+j}.$$

Then rank$(T_0^{(p)}) = p$.

Proof of Lemma A1. In the following, all index arithmetic in vectors and matrices is done over $\mathbb{Z}_p$. First, observe that transpose$(T_0^{(p)}) = T_0^{(p)}$. In particular:

$$\text{transpose}(T_0^{(p)})[\cdot,i] = T_0^{(p)}[p-x,\cdot]$$

Let us figure out the left kernel of $T_0^{(p)}$, K. In particular, for all $z \in K$, all coordinates of $z \cdot T_0^{(p)}$ are equal.

We note that since for all $i \in \mathbb{Z}_p$:

$$z \cdot T_0^{(p)}[\cdot,i] - z \cdot T_0^{(p)}[\cdot,i+1] \;=\; 0,$$

then by the structure of the columns, we have $z[i] = z[i+x]$. Since p is prime and $x < p$, the sequence $z[0], z[x], z[2x], \ldots, z[(p-1)x]$ goes over all entries in z. We thus conclude that $z = a(1,\ldots,1)$ for some $a \in \mathbb{Z}_p$. By a simple calculation, we have $z \cdot T_0^{(p)} = a \cdot x$. As $x \neq 0$ as an element of $\mathbb{Z}_p$, then to obtain $zT_0^{(p)} = 0$, we must have $a = 0$. That is, there is no non-zero linear combination of $Rows(T_0^{(p)})$ leading to zero, so $Rows(T_0^{(p)})$ are independent, as required. □

Observation 5. *There exist at least two α_j's that belong to distinct orbits that are non-zero.*

Proof of Observation 5. For any single orbit I_i ($0 \leq i < p_2$), one may view the matrix $T_0[I_i,\cdot]$ as consisting of $1 \times p_2$ blocks, with the first block in each row starting at entry i (and one of the blocks possibly wrapping around). The matrix $T_0[I_i,\cdot]$ (with rows permuted for convenience) is thus a $p_1 \times p_1$ block-matrix such that every block initially consists of entries of the form $a \cdot (1,\ldots,1)$ for $a \in \{0,1\}$

in each block. Consider a linear L mapping applied to the resulting block matrix V_0^i, replacing each block with a single element of $\mathbb{Z}_p$ that maps $(1,\ldots,1)$ to one. Then, clearly, $L_0^i = L(T_0[I_i,\cdot])$ is a $p_1 \times p_1$ matrix over $\mathbb{Z}_p$ satisfying the conditions of Lemma A1. This is the case since $p = p_1$, and each row has $p_2^{-1} \bmod p_1$ non-zero blocks. By Lemma A1, the rows of L_0^i are therefore linearly independent. We conclude that so are the rows of $T_0[I_i,\cdot]$ (or else the rows of L_0^i would also be dependent via the same linear combination). Therefore, the rows of the $T_0[I_i \cap I,\cdot]$ are also linearly independent. □

Proof of Lemma 2. We start with the "in particular" part. To prove this, recall that in each row in T_0, there are exactly $(1,0)_{\mathbb{Z}_m}$ consecutive one. That is, for $x = p_2^{-1} \bmod p_1$, we have $xp_2 = x(p_2,0) = (1,0)_{\mathbb{Z}_m}$. That is, p_2 fits into $(1,0)_{\mathbb{Z}_m}$ $p_2^{-1} \bmod p_1$ (lifted back to $\mathbb{Z}$) times. Indeed, $p_2^{-1} \bmod p_1$ exists, as $p_2 \neq p_1$ and is a prime, so it does not equal zero mod p_1.

We therefore get:

$$\sum_{i=0}^{p_1-1} T_0[i \cdot p_2,[m]] = x \cdot (1,\ldots,1)$$

over $\mathbb{Z}_p$. This holds, since at every point, we add exactly x vectors that contribute one to that point. Here, the integer x is viewed as an element of $\mathbb{Z}_p$. As $p = p_1$, x is non-zero in $\mathbb{Z}_p$.

It is not hard to see that every other vector in $Rows(T_0)$ is spanned by $Rows(T_0[I,[m]])$: Let use define an orbit as a sub-set of vectors $I_i = \{i + jp_2 | 0 \leq j \leq p_1 - 1\}$ for $i \in \mathbb{Z}_{p_2}$. It is easy to see that we have $\sum_{j \in S_i} T_0[j,\cdot] = \sum_{j \in I} T_0[j,\cdot] = x(1,\ldots,1)$ for any $i \in \mathbb{Z}_{p_2}$. Therefore, for any $T_0[i + (p_1-1)p_2,\cdot]$ for $i + (p_1-1)p_2 \notin I$, we have:

$$T_0[i + (p_1-1)p_2,\cdot] + \sum_{j=0}^{p_1-2} T_0[i + jp_2,\cdot] - x(1,\ldots,1) = \tag{A1}$$

$$T_0[i + (p_1-1)p_2,\cdot] + \sum_{j=0}^{p_1-2} T_0[i + jp_2,\cdot] - \sum_{i=0}^{p_1-1} T_0[i \cdot p_2,\cdot] = 0 \tag{A2}$$

Indeed, in the above equation, all summands but $T_0[i + (p_1-1)p_2,\cdot]$ are in $Rows(T_0[I,\cdot])$.

To complete the proof, it remains to prove that $Rows(T_0[I,\cdot])$ is an independent set. Assume for contradiction that a non-trivial linear combination of $Rows(T_0[I,\cdot])$ leads to zero. Let:

$$\sum_{i \in I} \alpha_i T_0[i,\cdot] = 0 \tag{A3}$$

where not all α_i's are zero. Splitting the sum into orbits, we get:

$$\sum_{i \in \mathbb{Z}_{p_2}} \sum_{j \in I \cap I_i} \alpha_j T_0[j,\cdot] = 0 \tag{A4}$$

Consider all orbits $i_1 < i_2 \ldots < i_t$ in which non-zero coefficients in the above combination exist. By Observation 5, $t \geq 2$. In particular, starting from $h = 2$, not all α_j's corresponding to $j \in I_h$ are non-zero (because $|I \cap I_i| = p_1$ only for $i = 0$ and equals $p_1 - 1$ otherwise).

Thus, we have $\sum_{j \in I_{i_1}} \alpha_j T_0[j,\cdot] = -\sum_{g=2}^{t} \sum_{j \in I_{i_g}} \alpha_j T_0[j,\cdot]$. Note that the vector on the left-hand side has the property that every block of p_2 consecutive elements starting at some index jp_2 is of the form $a(1,\ldots,1)$. On the right-hand side, as orbit I_{i_2} has non-zero α_j's, but not all of them non-zero, the sum:

$$\sum_{j \in I_{i_2}} \alpha_j T_0[j,\cdot] \tag{A5}$$

is non constant. The latter holds since all the rows indexed by this orbit are independent by Lemma A1, and $z = a(1,\ldots,1)$ lead to constant $zT_0[I_{i_2},\cdot]$. We thus conclude that only such z's lead to constant $zT_0[I_{i_2},\cdot]$. As at least one α_j in this orbit is zero, and at least one is not. z cannot be of this form,

and thus, the $zT_0[I_{i_2}, 0]$ is not constant. On the other hand, by the structure of $zT_0[I_{i_2}, \cdot]$, every block of size p_2 starting at $i_2 + jp_2$ is of the form $a(1, \ldots, 1)$. In particular, there must exist two consecutive blocks starting at $i_2 + jp_2, i_2 + (j+1)p_2$ respectively with different values $a_1(1, \ldots, 1)$ $a_2(1, \ldots, 1)$ respectively. In this pair of blocks, each intersects the block B_0 starting at $(j+1)p_2$ from orbit I_0. Thus, on the right-hand side, this block has the same value a_1 at indices $\{(j+1)p_2, \ldots, (j+1)p_2 + i_2 - 1\}$ (a non-empty set, as $i_2 \neq 0$) and a different value a_2 at indices $\{(j+1)p_2 + i_2, \ldots, (j+2)p_2 - 1\}$. Going over the following i_g's, a similar situation results. Spelling it out, i_3 (if it exists) will perhaps create one or more additional "imbalanced" blocks $(j'+1)p_2$ and maybe affect the original block we singled out, $(j+1)p_2$ affected by i_2. In the latter case, it will not fix the original imbalance. To see that, let us consider the sequence of orbits $0 < i'_1 < i'_2 < \ldots < i'_{t'}$ affecting the block B_0 in the way described above. As we add the contribution of the i_j's from left to right, we consider the sequence going from the end of the block to the left up to entry $(j+1)p_2 + i'_j$ included. It is not hard to show by induction on j that this sequence consists of the same value. When adding the contribution of $i'_{t'}$, this sequence (of length at least two at this point, as $t' \leq p_1 - 1$) is broken into two, where the first entry in the sequence now differs from the last. The delta equal between these entries is $a_2 - a_1$, where a_1, a_2 are the constants corresponding to $i'_{t'}$, as explained for i'_2 above. We conclude that the right-hand side is not a multiple of $(1, \ldots, 1)$ (as the block B_0 is not constant) and in particular may not equal zero. □

Appendix A.2. Proof of Lemma 3

Proof of Lemma 3. We prove the claim in two steps. First, we observe that S is the right kernel of T_0. Then, by recalling $\text{transpose}(T_0) = T_0$, the claim follows. To prove the first observation, we note that indeed, S is a subset of the right kernel of T_0, by calculating the products $T_0 \cdot v$ for each $v \in S$, essentially following from the observation that the sum over all rows indexed by any single orbit I_i (as defined in the previous section) is the same. Next, it is easy to see that the set S is independent; for instance, focus on the first block I_2 of p_2 columns, and observe that the rank of $T_0[\cdot, I_2]$ is by itself $p_2 - 1 = |S|$. Finally, $rank(T_0[I, \cdot])$ defined in Lemma 2 is $|I| = (p_1 - 1)p_2 + 1$. As $Rows(T_0[I, \cdot])$ span $T_0[I, \cdot]$, then by the Cayley–Hamilton theorem, the rank of the right kernel of T_0 is $m - (p_1 - 1)p_2 - 1 = p_2 - 1$. We conclude that S is a basis of the (right) kernel of T_0, which concludes the proof. □

References

1. Chor, B.; Kushilevitz, E.; Goldreich, O.; Sudan, M. Private Information Retrieval. *J. ACM* **1998**, *45*, 965–981. [CrossRef]
2. Kushilevitz, E.; Ostrovsky, R. Replication is NOT Needed: SINGLE Database, Computationally-Private Information Retrieval. In Proceedings of the 38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, FL, USA, 19–22 October 1997; pp. 364–373.
3. Gentry, C.; Ramzan, Z. Single-Database Private Information Retrieval with Constant Communication Rate. In Proceedings of the Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, 11–15 July 2005; pp. 803–815.
4. Beimel, A.; Ishai, Y.; Kushilevitz, E.; Orlov, I. Share Conversion and Private Information Retrieval. In Proceedings of the 27th Conference on Computational Complexity, CCC 2012, Porto, Portugal, 26–29 June 2012; pp. 258–268.
5. Ben-Or, M.; Goldwasser, S.; Wigderson, A. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation (Extended Abstract). In Proceedings of the 20th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, 2–4 May 1988; pp. 1–10.
6. Goldreich, O.; Micali, S.; Wigderson, A. How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. In Proceedings of the 19th Annual ACM Symposium on Theory of Computing, New York, NY, USA, 25–27 May 1987; pp. 218–229. [CrossRef]
7. Cramer, R.; Damgård, I.; Ishai, Y. Share Conversion, Pseudorandom Secret-Sharing and Applications to Secure Computation. In *Theory of Cryptography Conference*; Springer: Berlin/ Heidelberg, Germany, 2005; pp. 342–362.

8. Grolmusz, V. Constructing set systems with prescribed intersection sizes. *J. Algorithms* **2002**, *44*, 321–337. [CrossRef]
9. Beimel, A.; Ishai, Y.; Kushilevitz, E.; Raymond, J. Breaking the $O(n^{1/(2k-1)})$ Barrier for Information-Theoretic Private Information Retrieval. In Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS 2002), Vancouver, BC, Canada, 16–19 November 2002; pp. 261–270.
10. Efremenko, K. 3-Query Locally Decodable Codes of Subexponential Length. *SIAM J. Comput.* **2012**, *41*, 1694–1703. [CrossRef]
11. Yekhanin, S. Towards 3-query locally decodable codes of subexponential length. *J. ACM* **2008**, *55*. [CrossRef]
12. Dvir, Z.; Gopalan, P.; Yekhanin, S. Matching Vector Codes. *SIAM J. Comput.* **2011**, *40*, 1154–1178. [CrossRef]
13. Dvir, Z.; Gopi, S. 2-Server PIR with Subpolynomial Communication. *J. ACM* **2016**, *63*, 39. [CrossRef]
14. Beimel, A. Secret-Sharing Schemes: A Survey. In *International Conference on Coding and Cryptology*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 11–46.
15. Wehner, S.; de Wolf, R. Improved Lower Bounds for Locally Decodable Codes and Private Information Retrieval. In *International Colloquium on Automata, Languages, and Programming*; Springer: Berlin/Heidelberg, Germany, 2005.