Ramp hyper-invertible matrices and their applications to MPC protocols

Hongqing Liu[1], Chaoping Xing[1], Yanjiang Yang[2], and Chen Yuan[1]

[1] Shanghai Jiao Tong University, Shanghai, China
[2] Huawei International, Singapore

Abstract. Beerliová-Trubíniová and Hirt introduced hyper-invertible matrix technique to construct the first perfectly secure MPC protocol in the presence of maximal malicious corruptions $\lfloor \frac{n-1}{3} \rfloor$ with linear communication complexity per multiplication gate [5]. This matrix allows MPC protocol to generate correct shares of uniformly random secrets in the presence of malicious adversary. Moreover, the amortized communication complexity of generating each sharing is linear. Due to this prominent feature, the hyper-invertible matrix plays an important role in the construction of MPC protocol and zero-knowledge proof protocol where the randomness needs to be jointly generated. However, the downside of this matrix is that the size of its base field is linear in the size of its matrix. This means if we construct an n-party MPC protocol over $\mathbb{F}_q$ via hyper-invertible matrix, q is at least $2n$.

In this paper, we propose the ramp hyper-invertible matrix which can be seen as the generalization of hyper-invertible matrix. Our ramp hyper-invertible matrix can be defined over constant-size field regardless of the size of this matrix. Similar to the arithmetic secret sharing scheme, to apply our ramp hyper-invertible matrix to perfectly secure MPC protocol, the maximum number of corruptions has to be compromised to $\frac{(1-\epsilon)n}{3}$. As a consequence, we present the first perfectly secure MPC protocol in the presence of $\frac{(1-\epsilon)n}{3}$ malicious corruptions with constant communication complexity. Besides presenting the variant of hyper-invertible matrix, we overcome several obstacles in the construction of this MPC protocol. Our arithmetic secret sharing scheme over constant-size field is compatible with the player elimination technique, i.e., it supports the dynamic changes of party number and corrupted party number. Moreover, we rewrite the public reconstruction protocol to support the sharings over constant-size field. Putting these together leads to the constant-size field variant of celebrated MPC protocol in [5].

We note that although it was widely acknowledged that there exists an MPC protocol with constant communication complexity by replacing Shamir secret sharing scheme with arithmetic secret sharing scheme, there is no reference seriously describing such protocol in detail. Our work fills the missing detail by providing MPC primitive for any applications relying on MPC protocol of constant communication complexity. As an application of our perfectly secure MPC protocol which implies perfect robustness in the MPC-in-the-Head framework, we present the constant-rate zero-knowledge proof with 3 communication rounds. The previous work achieves constant-rate with 5 communication rounds [32] due to the

statistical robustness of their MPC protocol. Another application of our ramp hyper-invertible matrix is the information-theoretic multi-verifier zero-knowledge for circuit satisfiability[43]. We manage to remove the dependence of the size of circuit and security parameter from the share size.

1 Introduction

Secure multiparty computation (MPC) is a technique that allows several parties to jointly compute a public function without disclosing their private inputs even if an adversary corrupts t out of n parties. The MPC protocols can be divided into several classes based on their security levels and threat models. A protocol is perfectly secure if an adversary's view of the protocol can be simulated given only his inputs and outputs, and the simulated view follows exactly the same distribution as the real view. An adversary is called malicious if the corrupted parties he controls can deviate the protocol in an arbitrary manner. It was shown in [6] that the maximal number of corrupted parties is $\lfloor \frac{n-1}{3} \rfloor$ for an n-party MPC protocol perfectly secure against malicious adversary.[3] Since then, there is a great effort to improve the communication complexity of MPC protocol in this adversary model. The first MPC protocol achieving linear communication complexity is due to [5]. They introduced a new technique called hyper-invertible matrices (HIM for short) that can generate a random sharing at the cost of linear communication complexity. They also borrow several ideas from previous works such as player elimination [31], public reconstruction [18]. We note that although they achieve the linear communication complexity, the actual amortized communication complexity of securely evaluating a multiplication gate is $O(n \log n)$ bits regardless of the size of the field. The work in [11] introduced a new technique called reverse multiplication friendly embedding which maps a vector in $\mathbb{F}_q^r$ into an element in extension field $\mathbb{F}_{q^m}$ while the component-wise product of two vectors is preserved by mapping it to a product of two elements(here m is linear in r). This technique enables their MPC protocol to securely evaluate $O(\log n)$ instances over binary field by invoking the protocol in [5] in a "black-box" way and thus they manage to achieve the linear communication complexity for any Boolean circuit. All above protocols use the Shamir secret sharing scheme (SSS) [39] as their building block. Thus, the share sizes of their protocols are least $\Omega(\log n)$.

The arithmetic SSS introduced in [12] generalizes the idea of Shamir SSS. The merit of the generalization is that one can obtain a variant of Shamir SSS over constant-size field while the downside of this variant is that there is an ϵn gap between privacy and reconstruction. Thus, such arithmetic SSS can not handle the maximal number of corruptions $\lfloor \frac{n-1}{3} \rfloor$ but the sub-optimal number of corruptions $\frac{(1-\epsilon)n}{3}$. Due to the Franklin-Yung paradigm [21] and arithmetic SSS, it was widely acknowledged that there exists an MPC protocol over constant-size

[3] The perfectly secure MPC protocol in this paper is assumed to have guaranteed output delivery since $t < n/3$.

field perfectly secure against $\frac{(1-\epsilon)n}{3}$ malicious corrupted parties with $O(1)$ amortized communication complexity. However, we are surprised to find that there is no literature seriously describing such a protocol in detail. In this paper, we present such a protocol by deriving constant-size field variant of the celebrated MPC protocol [5]. The first challenge we face is the constant-size field variant of hyper-invertible matrix which we name it ramp hyper-invertible matrix. The idea of ramp hyper-invertible matrix can be dated back to [11]. However, they do not seriously expand such idea by providing efficient constructions of this matrix. Instead, we present the explicit constructions of such matrix in this paper and apply it to MPC protocol. We believe that the applications of ramp hyper-invertible matrix are not limited to MPC protocol and might be of independent interests. Besides, the player elimination technique is not compatible with arithmetic SSS. The player elimination technique remove parties from the preprocessing phase which implies that the number of parties n and the number of corrupted parties t are dynamically changed during the preprocessing phase. Thus, we propose an arithmetic SSS that is compatible with the dynamic changes of n and t. Finally, we rewrite the public reconstruction protocol to make it applicable over constant-size field. Putting everything together, we are able to present the constant-size variant of MPC protocol [5]. As a consequence, we obtain a constant-rate zero-knowledge proof from MPC-in-the-head (MPCitH) framework [32]. We also provide two applications of our ramp hyper-invertible matrices in the zero-knowledge proof.

1.1 Our Contributions

The hyper-invertible matrix was proposed in [5] to amortize the communication complexity of generating random sharings. However, the downside of this matrix is that the size of its base field grows with the size of the matrix. Therefore, any MPC protocol based on hyper-invertible matrix must be defined over a field of size $\Omega(n)$. The motivation of our ramp hyper-invertible matrix is to construct a perfectly secure MPC protocol over constant-size field for n-parties in the presence of almost maximal malicious corruptions $\frac{(1-\epsilon)n}{3}$ such that the amortized communication complexity of evaluating single multiplication gate is $O(1)$. Such an MPC protocol implies a constant-rate zero-knowledge proof [32]. Although such an MPC protocol was assumed to exist by replacing the Shamir secret sharing scheme with the asymptotically good arithmetic secret sharing scheme [12] in the MPC protocol, we note that there are still several technical difficulties to be overcome which were not explored so far. In this work, we consider the constant-size field variant of the celebrated MPC protocol [5]. The first obstacle is the variant of hyper-invertible matrix defined over constant-size field which we believe to be of independent interest. The second obstacle is to construct arithmetic secret sharing scheme compatible with the dynamic change of the number of parties and the number of corrupted parties. This is due to the application of player elimination protocol which removes a pair of parties at a time. The third obstacle is to carry out error correction over constant-size field

as our shares are defined over constant-size field. In the rest of this subsection, we will introduce these obstacles in detail and how we overcome them.

Hyper-invertible matrix We introduce ramp hyper-invertible matrices which can be seen as a generalization of hyper-invertible matrices. Basically speaking, M is an $n \times n$ hyper-invertible matrix if for any subsets $I, J \subseteq [n]$ with $|I| = |J|$, the submatrix of M indexed by the rows in I and the columns in J is invertible. As a consequence, if $(y_1, \ldots, y_n)^T = M(x_1, \ldots, x_n)^T$, for any subsets $I, J \subseteq [n]$ with $|I| + |J| = n$, there is a linear bijective function $f : \mathbb{F}_q^n \to \mathbb{F}_q^n$ mapping $(x_i)_{i\in I}, (y_j)_{j\in J}$ onto $(x_i)_{i\in[n]/I}, (y_j)_{j\in[n]/J}$. This leads to the following two properties. If I is the set of corrupted parties, $(x_i)_{i\in[n]/I}$ and $(y_j)_{j\in[n]/J}$ uniquely determine $(y_j)_{j\in J}$. Moreover, $(y_j)_{j\in J}$ are distributed uniformly at random by knowing $(x_i)_{i\in I}$ and $(y_j)_{j\in[n]/J}$. The argument in [5] leverages these two properties to generate random double sharings with linear communication complexity. Our ramp hyper-invertible matrices still keep these two properties with slight relaxation. In particular, we require that $|I| + |J| \leq (1 - \epsilon)n$ for the first property to hold and $|I| + |J| \geq (1 + \epsilon)n$ for the second property to hold. We show that this ramp hyper-invertible matrix is closely related to linear code with large distance and dual distance. Such connection allows us to exploit the knowledge from the well-studied coding theory to produce ramp hyper-invertible matrix over any constant-size field.

Asymptotically good arithmetic secret sharing scheme and player elimination. The player elimination technique was introduced in [31] to divide the preprocessing phase into $\Omega(n)$ segments and in each segment if a party deviates from the protocol, a pair of parties containing this party will be identified and removed from the following computation. This technique can efficiently reduce the communication cost of identifying corrupted parties and thus was adopted in [5] and some follow-ups. To adapt such technique to our MPC protocol, our asymptotically good arithmetic SSS must be compatible with dynamic change of the number of parties and the number of corrupted parties. We note that in contrast to the Shamir SSS, the known construction of asymptotically good arithmetic SSS does not satisfy this dynamic property, i.e., based on algebraic geometry code, one can construct a family of t_i-strongly multiplicative SSS[4] on n_i parties such that $\frac{t_i}{n_i} = \Omega(1)$ and n_i tends to infinity with $\frac{n_i}{n_{i-1}} > 1$ is a constant. In Theorem 4, we show how to construct t'-strongly multiplicative SSS on n' parties for any $n' = n - 2(t - t')$ and $t' \leq t$ from a t-strongly multiplicative SSS on n parties.

Error-correcting codes and public reconstruction. The public reconstruction in [5] can efficiently and robustly open the secret at the cost of linear communication complexity. The first step is to treat k secrets waiting for opening as a message and re-encode such message to a codeword $(c_1, \ldots, c_n)$ via a

[4] We refer the reader to Section 4.1 for formal definition.

Reed-Solomon code. In this process, all parties locally compute the share of c_i according to the encoding algorithm. Then, all parties send their shares of c_i to the i-th party and let i-th party reconstruct c'_i. By applying the decoding algorithm to $(c'_1, \ldots, c'_n)$, all parties can robustly reconstruct the codeword and thus obtain k secrets. To adapt this protocol, we propose an error-correcting code over constant-size field with large distance. Moreover, the encoding and decoding algorithm of our code can be efficiently implemented.

Beaver triples. The Beaver triples are used to securely evaluate the multiplication gate in the online phase. The Beaver triple consists of two sharings of random elements $[a]_t, [b]_t$ and the share of their product $[ab]_t$ where $[\cdot]_t$ represents sharing of t-threshold Shamir SSS. To produce this triple in [5], the preprocessing phase first prepares two sharings of random element $[r]_t, [r]_{2t}$. Both of them can be efficiently produced via hyper-invertible matrix technique. $[r]_{2t}$ is used to mask the product $[a]_t[b]_t$ and $[r]_t$ is used to re-share the secret ab by computing $ab + r - [r]_t$. One can think of $[\cdot]_t$ as degree-t polynomial. To adapt this technique, we let $[\cdot]_t$ be the sharings of an SSS Σ_t with t-privacy. Since our SSS is t-strongly multiplicative, the product of two sharings belongs to a new SSS Σ_{2t} with $2t$-privacy. The reconstruction of Σ_{2t} is $2r$ if the reconstruction of Σ_t is r.[5]

Perfectly secure MPC protocol with constant amortized communication complexity. With all building blocks above at hand, we are able to present the perfectly secure MPC protocol with constant amortized communication complexity in the presence of $\frac{(1-\epsilon)n}{3}$ corrupted parties. The idea is to replace the building blocks in [5] defined over large field with our new building blocks which can be defined over constant-size field. To do this, we first replace the Shamir SSS with our arithmetic SSS to reduce the share size. Moreover, our new public reconstruction protocol is applicable to secret over constant-size field as we resort to error-correcting code over constant-size field. By replacing hyper-invertible matrix with ramp hyper-invertible matrix, we can generate double-sharings as efficient as in [5]. As a consequence, our new protocol can achieve the linear complexity as the celebrated MPC protocol in [5]. Since the number of corrupted parties $\frac{(1-\epsilon)n}{3}$ is suboptimal, our protocol use the packed arithmetic secret sharing to further reduce the communication complexity. If we simultaneously evaluate $\Omega(n)$ instances of the same circuit, we can reduce linear communication complexity to constant. In this sense, our protocol achieves the constant amortized communication complexity.

Constant-rate zero-knowledge proof. The communication complexity of constant-rate zero-knowledge proof is linear in circuit size $|C|$. The first construction was presented in [32] as a byproduct of the MPC-in-the-Head framework. This requires an MPC protocol over constant-size field with perfect or statistical

[5] In fact, we are only concerned about the reconstruction of Σ_{2t}.

t-robustness (in a malicious model) and t-privacy (in a semi-honest model). In [32], they show that a variant of the MPC protocol given in [16] using arithmetic SSS as the building block can serve this purpose. However, the MPC protocol in [16] only achieves statistical t-robustness. This forces the MPCitH protocol relying on this MPC protocol to be separated into two phases which causes more communication rounds. It is desirable to achieve perfect t-robustness so as to optimize the communication rounds. The building block of our MPCitH protocol is a perfectly secure MPC protocol over constant-size field and thus our MPCitH needs 3 communication rounds.

Joint sampling of multiple verifiers. In information-theoretic multi-verifier zero-knowledge(MVZK), n verifiers jointly generate one challenge for the prover. We consider MVZK for circuit satisfiability in the setting of honest majority verifiers. By applying HIM technique in [5] to generate random secret sharings, the coin-tossing protocol in MVZK [43] achieves communication overhead $O(\lambda + \log|C|)$ where λ is security parameter and $|C|$ is the number of multiplication gates in the circuit. If we relax the number of corrupt parties from $\frac{n-1}{2}$ to $(\frac{1-\epsilon}{2})n$, the ramp HIM can replace HIM to do the same job and reduce the communication overhead of coin-tossing protocol to $O(1)$. Moreover, we propose a new technique to remove the dependence of security parameter from the share size when checking the circuit satisfiability.

1.2 Related Work

The first perfectly secure MPC protocol was proposed in [6] for $t < n/3$. Since then, there are numerous efforts to reduce the communication complexity. The introduction of the hyper-invertible matrix in [5] leads to the first perfectly secure MPC protocol with linear communication complexity which also reaches the theoretical limit. The same linear communication complexity can be achieved for perfectly secure MPC protocol over *any* finite field [11]. The depth related communication complexity in the expression was further removed in [27]. For honest majority setting $t < (n-1)/2$, there are many constructions achieving linear complexity in security-with-abort model [18, 7, 24, 14, 36, 29, 26, 20]. In [13], they consider the honest majority MPC protocol tolerating $t < \frac{n(1-\epsilon)}{2}$ corruptions. Compared to the optimal corruption $\frac{n-1}{2}$, their scheme is defined over constant-size field and thus can save a $O(\log n)$ multiplicative factor. We note that once $t > n/3$, MPC protocols can not be zero-error but succeed with high probability with the help of broadcast channel. Thus, it is not comparable with our MPC protocol for $t < n/3$ where the broadcast can be simulated with perfect secure by communicating $O(n^2)$ bits.

The MPC-in-the-head paradigm establishes a close connection between zero-knowledge proof systems and MPC protocols. Its theoretical framework was proposed in [32] and the first practical instantiation was given in [25]. From practical point of view, the MPC protocols based on additive secret sharing play

a crucial role in the MPCitH protocol. The reason is that additive secret sharing can be efficiently generated by the pseudo-random generator and thus the MPCitH protocol can commit to the seed instead of sharings. The preprocessing phase was introduced to the MPCitH protocols in [34]. Since then, there are two ways of verification in the preprocessing phase: cut-and-choose (KKW [34]) and sacrificing (BN [4], Limbo [19], Helium [33]). They are practical MPCitH protocols requiring either more communication rounds or larger field size. In this work, we are concerned about the theoretical performance of zero-knowledge proof and thus propose the MPCitH protocol based on ramp HIM. Combined with our perfectly secure MPC protocol against malicious adversary, our MPCitH protocol is a 3-round constant-rate zero-knowledge proof.

MVZK was first proposed in [10] and non-interactive MVZK was instantiated in [1]. Some earlier works [1, 30] rely on public-key operations and thus achieve only computational security. In [2], they focus on minimal assumption for MVZK and achieves computational security and everlasting security. There are a few works investigating MVZK in the presence of honest majority verifiers [2, 3, 43]. The protocol in [3] aims to realize stronger security-with-identifiable-abort at a cost of tolerating a smaller number of corruptions($t < n/3$ or $t < n/4$). In this paper, we aim to reduce the communication overhead by replacing the HIM in [43] with the ramp HIM.

2 Preliminaries

For an integer $n > 1$, denote by $[n]$ the set $\{1, 2, \ldots, n\}$. For two integers a, b with $0 \leq a < b$, denote by $[a, b]$ the set $\{a, a+1, \cdots, b\}$. A finite field of size q is denoted by $\mathbb{F}_q$. Throughout this paper, we use bold face $\mathbf{v}$ to represent a vector. Given a vector $\mathbf{u} = (u_i)_{i\in[n]} \in \mathbb{F}_q^n$ and a subset $J \subset [n]$, we denote by $\mathbf{u}_J = (u_i)_{i\in J}$ the projection of $\mathbf{u}$ at J. The component-wise product of two vectors $\mathbf{c}_1$ and $\mathbf{c}_2$ is denoted by $\mathbf{c}_1 \star \mathbf{c}_2$. $\mathbb{F}_q^r$ is the collection of r-dimensional vectors over $\mathbb{F}_q$ and $\mathbb{F}_q^{r\times n}$ the collection of $r \times n$ matrices over $\mathbb{F}_q$. We assume the circuits evaluated by MPC protocol consist of c_I input gates, c_R random gates and c_M multiplicative gates. The depth of multiplication gates is denoted as D_M. We denote by λ the security parameter in zero-knowledge proof, which implies that soundness error is at most $2^{-\lambda}$.

2.1 Hyper-invertible matrices

The hyper-invertible matrix was introduced in [5] to amortize the communication complexity of generating random sharings. A prominent feature of hyper-invertible matrix is that every square submatrix of this matrix is invertible.

Definition 1. *A matrix $M \in \mathbb{F}_q^{r\times n}$ is called a hyper-invertible matrix if for any row index set $I \subseteq [r]$ and column index set $J \subseteq [n]$ with $|I| = |J|$, the square submatrix of M formed by rows indexed by I and columns indexed by J is invertible.*

The mapping of a hyper-invertible matrix implies a symmetry property.

Lemma 1 ([5]). *Let M be an $n \times n$ hyper-invertible matrix over $\mathbb{F}_q$. Let $(y_1, \ldots, y_n)^T = M(x_1, \ldots, x_n)^T$. Then for any subset $I, B \subseteq [n]$ with $|I| + |B| = n$, there is a linear bijective function $f : \mathbb{F}_q^n \to \mathbb{F}_q^n$ mapping $((x_i)_{i \in I}, (y_j)_{j \in B})$ onto $((x_i)_{i \in [n]/I}, (y_j)_{j \in [n]/B})$.*

2.2 Linear codes

A linear code C over $\mathbb{F}_q$ is a linear subspace in $\mathbb{F}_q^n$. The dimension of C is defined to be the $\mathbb{F}_q$-dimension of this subspace and the length of C is defined to be n. One can define Hamming distance for any pair of vectors $\mathbf{v} = (v_i), \mathbf{u} = (u_i)$ in $\mathbb{F}_q^n$, i.e., $d(\mathbf{v}, \mathbf{u}) = |\{i \in [n] : \ v_i \neq u_i\}|$. The Hamming weight of $\mathbf{u}$ is defined as $wt(\mathbf{u}) = d(\mathbf{u}, \mathbf{0})$, where $\mathbf{0}$ stands for the zero vector. For a linear code C, the minimum distance (distance for short) of C is defined to be the smallest Hamming weight of nonzero codewords. A linear code of length n, dimension k and distance d is denoted by $[n, k, d]$. A generator matrix G of a linear code C is a $k \times n$ matrix whose row vectors form an $\mathbb{F}_q$-basis. The dual code $C^\perp$ of C consists of the solutions to $G\mathbf{x}^T = \mathbf{0}^T$, i.e., $C^\perp = \{\mathbf{x} \in \mathbb{F}_q^n : G\mathbf{x} = \mathbf{0}\}$. We call the minimum distance of $C^\perp$ the dual distance of C. A generator matrix of $C^\perp$ is called a parity-check matrix of C.

2.3 Secret sharing scheme

Let us briefly introduce the background of secret sharing scheme.

Definition 2 (Secret sharing scheme). *A secret sharing scheme over $\mathbb{F}_q$ is a vector of random variables $\mathbf{X} = (X_0, X_1, \ldots, X_n)$ with each $X_i \in \mathbb{F}_q$ such that the following holds:*

- *The random variable X_0 is uniform over $\mathbb{F}_q$.*
- *t-privacy: Given any subset $B \subseteq [n]$ with $|B| \leq t$, any $x_0 \in \mathbb{F}_q$ and any $\mathbf{x}_B \in \mathbb{F}_q^{|B|}$ with $\Pr[(X_i)_{i \in B} = \mathbf{x}_B | X_0 = x_0] > 0$, $\Pr[X_0 = x_0 | (X_i)_{i \in B} = \mathbf{x}_B] = 1/q$. That is, the shares in the set B provide no information on the secret.*
- *r-reconstruction: Given any subset $B \subseteq [n]$ with $|B| \geq t + 1$ and any $\mathbf{x}_B \in \mathbb{F}_q^{|B|}$ with $\Pr[(X_i)_{i \in B} = \mathbf{x}_B | X_0 = x_0] > 0$, there is a unique $x_0 \in \mathcal{X}_0$ such that $\Pr[X_0 = x_0 | (X_i)_{i \in B} = \mathbf{x}_B] = 1$. That is, the shares in the set B uniquely determine the secret.*

In this paper, we use packed secret sharing scheme to reduce the communication complexity. A packed secret sharing scheme is a secret sharing scheme with its secret defined over a vector space instead of a field.

Definition 3 (Packed secret sharing scheme). *A packed secret sharing scheme over $\mathbb{F}_q$ with secret space $\mathbb{F}_q^s$ is a vector of random variables $\mathbf{X} = (X_0, X_1, \ldots, X_n)$ with each $X_i \in \mathbb{F}_q$ for $i \in [n]$ and $X_0 \in \mathbb{F}_q^s$.*

In most of the cases, a packed secret sharing scheme is obtained by first constructing a secret sharing scheme over $\mathbb{F}_q$ with $n+s-1$ shares $(X_1, \ldots, X_{n+s-1})$ and move $s-1$ shares $(X_{n+1}, \ldots, X_{n+s-1})$ to the secret space. Then, such secret sharing scheme has n shares $(X_1, \ldots, X_n)$ and the secret $(X_0, X_{n+1}, \ldots, X_{n+s-1}) \in \mathbb{F}_q^s$. It is easy to show that if the original secret sharing scheme has t-privacy and r-reconstruction, the resulting packed secret sharing scheme has $t-s$-privacy and r-reconstruction.

2.4 Algebraic curves

Let us briefly introduce some background on algebraic curves and function fields over finite fields. The reader may refer to [42, 41] for detail. An algebraic curve $\mathcal{X}$ defined over $\mathbb{F}_q$ is denoted by $\mathcal{X}/\mathbb{F}_q$. We denote by $\mathcal{X}(\mathbb{F}_q)$ the set of all $\mathbb{F}_q$-rational points on $\mathcal{X}$ (informally those points with coordinates belonging to $\mathbb{F}_q$). We denote by $\mathbb{F}_q(\mathcal{X})$ the function field of $\mathcal{X}/\mathbb{F}_q$. An element of $\mathbb{F}_q(\mathcal{X})$ is called a function. For a point P on $\mathcal{X}$, we denote by ν_P the normalized discrete valuation corresponding to the point P.

For a nonzero function x of $\mathbb{F}_q(\mathcal{X})$ and a point P, we denote by $\nu_P(x)$ the valuation of x at P. For $m \in \mathbb{Z}$, we form the vector space

$$\mathcal{L}(mP) = \{x \in \mathbb{F}_q(\mathcal{X}) \backslash \{0\} : \ \nu_P(x) \geq -m; \ \nu_Q(x) \geq 0 \text{ for all } Q \neq P\} \cup \{0\}. \quad (1)$$

This is a finite-dimensional vector space over $\mathbb{F}_q$. We have the following Riemann-Roch Theorem [41, Chapter 1].

Proposition 1 (Riemann-Roch Theorem). *Let $\mathcal{X}/\mathbb{F}_q$ be an algebraic curve of genus g. Then for any $m \in \mathbb{Z}$ and a point P, one has*

$$\dim_{\mathbb{F}_q} \mathcal{L}(mP) \geq m - g + 1 \quad (2)$$

and equality holds if $m \geq 2g-1$.

For algebraic geometry codes based on algebraic curves, we usually require curves have many rational points compared with genus. In other words, given an algebraic curve $\mathcal{X}/\mathbb{F}_q$ of genus g, we want the cardinality $|\mathcal{X}(\mathbb{F}_q)|$, denoted by $N(\mathcal{X})$, to be as large as possible. By the Hasse-Weil bound [42, 41, 35], we know that

$$N(\mathcal{X}) \leq q + 1 + 2g\sqrt{q}. \quad (3)$$

The above Hasse-Weil bound is tight for relatively small genus, i.e., for genus $g \leq q(q-1)/2$. For large genus, we have the following asymptotic Vlăduţ-Drinfeld bound [42, 41, 35]: for any family $\{\mathcal{X}/\mathbb{F}_q\}$ of algebraic curves with genus $g(\mathcal{X})$ of $\mathcal{X}$ tending to ∞, we have

$$\limsup_{g(\mathcal{X}) \to \infty} \frac{N(\mathcal{X})}{g(\mathcal{X})} \leq \sqrt{q} - 1. \quad (4)$$

If q is an even power of a prime, then based on modular curves, Garcia-Stichtenoth [22, 23] provided an explicit construction of a family $\{\mathcal{X}/\mathbb{F}_q\}$ of algebraic curves satisfying that $g(\mathcal{X}) \to \infty$ and

$$N(\mathcal{X}) \geq 1 + (\sqrt{q} - 1)g(\mathcal{X})$$

for every curve $\mathcal{X}$ in this family.

2.5 MPC-in-the-head

The MPC-in-the-Head paradigm was introduced by [32]. It applies an MPC protocol and a commitment scheme to construct a zero-knowledge proof for the witness w of any NP relation R. MPCitH tackles any NP relation $R(x, w)$ as an multiparty computation functionality $f(x, w)$ for input client I, n parties $P_1, \cdots, P_n$ and output client O. Input cient I receives witness w from the prover and shares to n parties, who can execute a protocol to verify the witness with public input x and send result to output client O.

The prover emulates the execution of an MPC protocol with n imaginary parties in his head and commits to the views of all parties. The view of a party consists of its private input, its random tapes, and all its received messages from other parties. The verifier selects a subset containing t parties. Finally, the prover reveals the views of chosen parties and the verifier checks the consistency of views. We say a pair of views V_i, V_j of party P_i, P_j are consistent if all ongoing messages of V_i are identical to the incoming messages in V_j and vice versa.

If we want to instantiate MPCitH paradigm with a concrete MPC protocol, it should satisfy following properties:

Definition 4 (Three properties of an MPC protocol). *Let Π_f be an MPC protocol realizing the function f representing a NP relation R for input client I, n parties $P_1, \cdots, P_n$ and output client O. Let $1 \leq t < n$ and the adversary could corrupt at most input client and t parties. We denote $I \subseteq [n]$ with $|I| \leq t$ as corrupted parties.*

- ***Correctness**: We say Π_f realizes perfect (statistical, respectively) correctness if for any $(x, w_1, \cdots, w_n)$, the probability that the outputs of some parties deviate from $f(x, w_1, \cdots, w_n)$ is 0 ($negl(\lambda)$, respectively).*
- *t-**Privacy**: We say Π_f realizes statistical (perfect, respectively) t-privacy in the presence of semi-honest adversary if for any input $(x, w_1, \cdots, w_n)$, there exists a PPT algorithm $\mathcal{S}$ such that the distribution of $\mathcal{S}(x, \{w_i\}_{i \in I}$, $f(x, w_1, \cdots, w_n))$ is statistically(perfectly, respectively) indistinguishable with the distribution of joint views $View_I(x, w_1, \cdots, w_n)$*
- *t-**Robustness**: We say Π_f realizes statistical (perfect, respectively) t-robustness in the presence of malicious adversary if for any input $(x, w_1, \cdots, w_n)$ satisfying $f(x, w_1, \cdots, w_n) = 0$, the probability that all parties outputs 1 and the views of honest parties are consistent is $negl(\lambda)$ (0, respectively).*

3 Ramp hyper-invertible matrix

In this section, we will introduce the notion of ramp hyper-invertible matrices and their constructions. We also provide an explicit construction via algebraic geometry codes as well as an existence result based on the Gilbert-Varshamov bound.

3.1 Ramp hyper-invertible matrices and functions

The ramp hyper-invertible matrix (ramp HIM for short) is a generalization of the hyper-invertible matrix. The formal definition is given as follows.

Definition 5. *A matrix $M \in \mathbb{F}_q^{m\times n}$ with $m \leq n$ is called an $(n, m; r, p)_q$-ramp hyper-invertible matrix if*

(i) For any integers s, t satisfying $0 \leq s \leq m$, $0 \leq t \leq n$ and $s + t \geq r$, every $s \times (n - t)$ submatrix of M has full column rank;
(ii) For any integers s, t satisfying $0 \leq s \leq m$, $0 \leq t \leq n$ and $s + t \leq p$, every $s \times (n - t)$ submatrix of M has full row rank.

Definition 5 implies that an $(n, m; n, n)_q$-ramp HIM is actually an HIM defined in [5]. However, it is not easy to see how to construct a ramp HIM meeting Definition 5. Thus, we propose an equivalence definition, i.e, ramp hyper-invertible function (HIF for short). The ramp HIF is a generalization of hyper-invertible function defined in [5].

Definition 6. *An $\mathbb{F}_q$-linear map from $\mathbb{F}_q^n$ to $\mathbb{F}_q^m$ is called an $(n, m; r, p)_q$-ramp hyper-invertible function if*

(i) Given every pair $\mathbf{x} \in \mathbb{F}_q^n$ and $\mathbf{y} \in \mathbb{F}_q^m$ with $\mathbf{y} = f(\mathbf{x})$; and any subsets $I \subseteq [n]$ and $J \subseteq [m]$ with $|I| + |J| \geq r$, the vectors $\mathbf{x}_I$ and $\mathbf{y}_J$ uniquely determine $\mathbf{x}_{\bar{I}}$.
(ii) Given any subsets $I \subseteq [n]$ and $J \subseteq [m]$ with $|I| + |J| \leq p$, and any vector $\mathbf{u}_I \in \mathbb{F}_q^{|I|}$, the composition map $\pi_J \circ f(\mathbf{u}_I, \mathbf{x}_{\bar{I}})$ is a surjective map from $\mathbb{F}_q^{|\bar{I}|}$ to $\mathbb{F}_q^{|J|}$:

$$\mathbb{F}_q^{|\bar{I}|} \xrightarrow{f(\mathbf{u}_I, \mathbf{x}_{\bar{I}})} \mathbb{F}_q^m \xrightarrow{\pi_J} \mathbb{F}_q^{|J|},$$

where π_J is the projection map at the index set J.

The following result proves the equivalence between ramp HIMs and ramp HIFs.

Theorem 1. *There exists an $(n, m; r, p)_q$-ramp HIM if and only if there exists an $(n, m; r, p)_q$-ramp hyper-invertible function.*

Proof. We first prove the if direction. Assume that there is an $(n, m; r, p)_q$-ramp hyper-invertible function ϕ. By fixing a basis $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ of $\mathbb{F}_q^n$ and a basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ of $\mathbb{F}_q^m$, we have $\phi(\mathbf{v}_i) = \sum_{j=1}^m a_{ij}\mathbf{u}_j$. Since ϕ is an $\mathbb{F}_q$-linear map, we conclude that $\phi(\mathbf{x}) = A\mathbf{x}^T$, where $A = (a_{ij})$. Next, we show that A is indeed an $(n, m; r, p)$-ramp HIM. Let $I \subseteq [n]$, $J \subseteq [m]$ be any subsets of size t and s respectively.

1. Assume $s + t \geq r$. Recall that the matrix A_{JI} is a submatrix of A whose rows are indexed by J and columns indexed by I. As $\phi(\mathbf{x}) = A\mathbf{x}^T = \mathbf{y}^T$, we have $\mathbf{y}_J^T = A_{JI}\mathbf{x}_I^T + A_{J\bar{I}}\mathbf{x}_{\bar{I}}^T$. By the first condition of a ramp HIF, $\mathbf{y}_J$ and $\mathbf{x}_I$ uniquely determine $\mathbf{x}_{\bar{I}}$. Suppose that $A_{J\bar{I}}$ would not have full column rank, then there exists a nonzero vector $\mathbf{a}$ such that $A_{J\bar{I}}\mathbf{a}^T = \mathbf{0}$. This implies that $A_{J\bar{I}}\mathbf{x}_{\bar{I}}^T = A_{J\bar{I}}(\mathbf{a} + \mathbf{x}_{\bar{I}})^T = \mathbf{y}_J^T - A_{JI}\mathbf{x}_I^T$. This contradicts the first condition of a ramp HIF. Thus, we conclude that any $s \times (n-t)$ submatrix of A has full column rank if $s + t \geq r$.
2. Assume $s + t \leq p$. Similarly, we have $\mathbf{y}_J^T = A_{JI}\mathbf{x}_I^T + A_{J\bar{I}}\mathbf{x}_{\bar{I}}^T$. By fixing $\mathbf{x}_I$ and the second condition of a ramp HIF, the map $\phi_J : \mathbb{F}_q^{|\bar{I}|} \to \mathbb{F}_q^{|J|}$ given by

$$\phi_J(\mathbf{x}_{\bar{I}}) := A_{J\bar{I}}\mathbf{x}_{\bar{I}}^T + A_{JI}\mathbf{x}_I^T$$

   is surjective. This implies that $A_{J\bar{I}}$ has full row rank. Thus, we conclude that any $s \times (n-t)$ submatrix of A has full row rank if $s + t \leq p$.

We proceed to prove the only if direction. Given an $[n, m; r, p]_q$-ramp HIM A, we define the linear map $\phi : \mathbb{F}_q^n \to \mathbb{F}_q^m$ given by $\phi(\mathbf{x}) = A\mathbf{x}^T$. Let $I \subseteq [n]$, $J \subseteq [m]$ be any subsets of size t and s respectively.

1. Assume $s + t \geq r$. Given every pair $\mathbf{x} \in \mathbb{F}_q^n$ and $\mathbf{y} \in \mathbb{F}_q^m$ with $\mathbf{y}^T = \phi(\mathbf{x})$, we have $\mathbf{y}_J^T = A_{JI}\mathbf{x}_I^T + A_{J\bar{I}}\mathbf{x}_{\bar{I}}^T$. Since any $s \times (n-t)$ submatrix of A has full column rank, a similar proof in the "if" part shows that $\mathbf{x}_{\bar{I}}$ is uniquely determined by $\mathbf{y}_J$ and $\mathbf{x}_I$.
2. Assume $s + t \leq p$. Observe that $\mathbf{y}_J^T = A_{JI}\mathbf{x}_I^T + A_{J\bar{I}}\mathbf{x}_{\bar{I}}^T$. Fixing any $\mathbf{x}_I \in \mathbb{F}_q^{|I|}$, the map $\pi_J \circ \phi(\mathbf{x}) = \mathbf{y}_J$ is surjective as $A_{J\bar{I}}$ is an $s \times (n-t)$ submatrix of A with full row rank.

The proof is completed.

From Theorem 1, it suffices to construct a ramp HIF so as to construct a ramp HIM. In the following subsection, we show how to construct the ramp HIF from the linear code. This provides a machinery for the constructions of ramp HIMs.

3.2 Connections with linear codes

In this subsection, we establish the connection between ramp HIFs and linear codes. We show that a linear code with large distance and dual distance can be used to construct a ramp HIF. Since linear codes are well studied, this provides a very good source for explicitly constructing HIMs. In the following theorem, we prove that a ramp HIF exists if and only if a linear code with certain property exists.

Theorem 2. *There exists an $(n, m; r, p)_q$-ramp HIF if and only if there exists an $[n + m, n, n + m - r + 1]$-linear code C with dual distance $p + 1$ over $\mathbb{F}_q$.*

Proof. We first prove the if direction. Since the dimension of C is n, without loss of generality, we may assume the first n indices of C form an information set, i.e., the first n columns of every generator matrix are linearly independent. We proceed to show how to construct a ramp HIF from C. Define a linear map $\phi : \mathbb{F}_q^n \to \mathbb{F}_q^m$ given by $\phi(c_1, \ldots, c_n) = (c_{n+1}, \ldots, c_{n+m})$ with $(c_1, \ldots, c_{n+m}) \in C$. We first prove that this map is well defined. Note that $[n]$ is an information set of C. This implies that the projection map $\pi_{[n]} : C \to \mathbb{F}_q^n$ is a bijection. For any vector $(c_1, \ldots, c_n) \in \mathbb{F}_q^n$, there exists unique codeword $\mathbf{c} \in \mathbb{F}_q^{n+m}$ such that $\mathbf{c}_{[n]} = (c_1, \ldots, c_n)$. Thus, this map is well defined. It is clear that ϕ is an $\mathbb{F}_q$-linear map. We proceed to show that ϕ is an $(n, m; r, p)_q$-ramp HIF. Let $I \subseteq [n]$, $J \subseteq [m]$ be any subsets of size t and s, respectively.

1. Assume $s + t \geq r$. Given every pair $\mathbf{x} \in \mathbb{F}_q^n$ and $\mathbf{y} \in \mathbb{F}_q^m$ with $\mathbf{y} = \phi(\mathbf{x})$, we have $(\mathbf{x}, \mathbf{y}) \in C$. Since C has minimum distance $n + m - r + 1$, knowing $\mathbf{x}_I$ and $\mathbf{y}_J$, we can uniquely identify a codeword $(\mathbf{x}, \mathbf{y}) \in C$. Otherwise, if there exists another codeword $(\mathbf{x}', \mathbf{y}') \in C$ such that $\mathbf{x}'_I = \mathbf{x}_I$ and $\mathbf{y}'_J = \mathbf{y}$. Due to the linearity of C, $(\mathbf{x} - \mathbf{x}', \mathbf{y} - \mathbf{y}') \in C$ is a nonzero codeword of weight at most $n + m - r$. A contradiction occurs. Since we can uniquely identify a codeword $(\mathbf{x}, \mathbf{y}) \in C$, $\mathbf{x}_{\bar{I}}$ is unique.
2. Assume $s + t \leq p$. Given any vector $\mathbf{u}_I \in \mathbb{F}_q^{|I|}$, we want to prove that the map $\pi_J \circ \phi(\mathbf{u}_I, \mathbf{u}_{\bar{I}})$ is a surjection from $\mathbb{F}_q^{|\bar{I}|} \to \mathbb{F}_q^{|J|}$. To see this, we recall that the dual distance of C is $p+1$. This implies that for any $\mathbf{u}_I \in \mathbb{F}_q^{|I|}$ and $\mathbf{v}_J \in \mathbb{F}_q^{|J|}$, there exists a codeword $(\mathbf{x}, \mathbf{y}) \in C$ such that $\mathbf{x}_I = \mathbf{u}_I$ and $\mathbf{y}_J = \mathbf{v}_J$ as $|I| + |J| \leq p$. By the definition of the map ϕ, the identity $\pi_J \circ \phi(\mathbf{x}_I, \mathbf{x}_{\bar{I}}) = \mathbf{y}_J$ holds for any $\mathbf{y}_J \in \mathbb{F}_q^{|J|}$.

We proceed to prove the only if direction. Let ϕ be an $(n, m; r, p)_q$-ramp HIF. Thus, by Theorem 1, we have $\phi(\mathbf{x}) = A\mathbf{x}^T$ for some $m \times n$ matrix A over $\mathbb{F}_q$. To define a linear code C, it suffices to define a generator matrix G of C. Let $G = (I_n, A^T)$ be an $n \times (n + m)$ matrix over $\mathbb{F}_q$, where I_n is the identity matrix of size n. We want to show that the linear code with generator matrix G has dimension n, minimum distance at least $n + m - r + 1$ and dual distance at least $p + 1$. The dimension of this code is clear as $rank(G) = n$.

(i) We now show that the minimum distance of C is at least $n+m-r+1$. Suppose that the minimum distance were less than $n + m - r + 1$. Let $(\mathbf{x}, \mathbf{x}A^T) \in C$ be a codeword of weight at most $n + m - r$. This means there exists two index subsets $I \subseteq [n]$ and $J \subseteq [m]$ with $|I| + |J| \geq r$ such that $\mathbf{x}_I = \mathbf{0}$ and $\pi_J(A\mathbf{x}^T) = A_{JI}\mathbf{x}_I^T + A_{J\bar{I}}\mathbf{x}_{\bar{I}}^T = \mathbf{0}$. This gives $A_{J\bar{I}}\mathbf{x}_{\bar{I}} = \mathbf{0}$. Put $s = |I|$ and $t = |J|$. Since ϕ is an $(n, m; r, p)_q$-ramp HIF, by Theorem 1, any $s \times (n - t)$ submatrix of A has full column rank. This implies $A_{J\bar{I}}$ has full column rank and $\mathbf{x}_{\bar{I}}$ has to be $\mathbf{0}$. Therefore, the distance of C is at least $n + m - r + 1$.

(ii) Finally we show that the dual distance of C is at least $p + 1$. Since the generator matrix G of C is systematic, a generator matrix of dual code $C^\perp$ of C has the form $(-A, I_m)$. We turn to bound the minimum distance of this dual code. Let $\mathbf{c} = (-\mathbf{x}A, \mathbf{x}) \in C^\perp$ be a codeword of weight at most p.

Let $I \subseteq [n]$ and $J \subseteq [m]$ with $s = |I|$ and $t = |J|$ be the support sets of $\mathbf{x}A$ and $\mathbf{x}$, respectively. This implies $\pi_{\bar{I}}(-\mathbf{x}A) = -\mathbf{x}_J A_{J\bar{I}} = \mathbf{0}$. Since ϕ is an $(n, m; r, p)_q$-ramp HIF, by Theorem 1, any $s \times (n-t)$ submatrix of A has full row rank as long as $s + t \leq p$. This forces $\mathbf{x}_J = \mathbf{0}$ and thus $\mathbf{c} = \mathbf{0}$. Therefore, the dual distance of C is at least $p + 1$.

The proof is completed.

By combining Theorem 2 and Theorem 1, we obtain the following corollary.

Corollary 1. *The following are equivalent.*

(i) *There exists an $[n, m; r, p]_q$-ramp HIM.*
(ii) *There exists an $[n, m; r, p]_q$-ramp HIF.*
(iii) *There exists an $[n + m, n, n + m - r + 1]$ linear code C over $\mathbb{F}_q$ with dual distance $p + 1$.*

Moreover, if $G = (I_n, A^T)$ is the generator matrix of C, then A is an $[n, m; r, p]_q$-ramp HIM.

3.3 Construction of q-ary ramp HIM

By Subsection 3.2, we know that in order to construct a ramp HIM with smaller reconstruction r and larger privacy p, we need a linear code with both large distance and dual distance. A good candidate for such a code is the algebraic geometry code. Before instantiating our construction of ramp HIMs through linear codes, let us briefly introduce algebraic geometry codes in this subsection. The reader may refer to [42, 41] for details. In this subsection, we instantiate the construction of q-ary ramp HIM. The binary ramp HIM is deferred to the Appendix.

Let $\mathcal{X}/\mathbb{F}_q$ be an algebraic curve of genus g with $\ell+1$ pairwise distinct rational points $P_\infty, P_1, \ldots, P_\ell$. Denote by $\mathcal{P}$ the set $\{P_1, \ldots, P_\ell\}$. For an integer κ with $g \leq \kappa < \ell$, define an algebraic geometry code

$$C(\mathcal{P}, \kappa P_\infty) := \{(f(P_1), \ldots, f(P_\ell)) : \ f \in \mathcal{L}(\kappa P_\infty)\}. \tag{5}$$

Then the code $C(\mathcal{P}, \kappa P_\infty)$ is a linear code over $\mathbb{F}_q$. Furthermore, $C(\mathcal{P}, \kappa P_\infty)$ and its dual $C^\perp(\mathcal{P}, \kappa P_\infty)$ have the following parameters

Proposition 2 (see [42, 41]). *Assume $2g - 1 < k < \ell$, then $C(\mathcal{P}, \kappa P_\infty)$ is a q-ary $[\ell, k, d]$-linear code and $C^\perp(\mathcal{P}, \kappa P_\infty)$ is a q-ary $[\ell, k^\perp, d^\perp]$-linear code with the parameters $k, k^\perp, d, d^\perp$ satisfying*

$$k = \kappa - g + 1, \quad k^\perp = \ell - \kappa + g - 1, \quad d \geq \ell - \kappa \quad d^\perp \geq \kappa - 2g + 2.$$

Corollary 2 (via Garcia-Stichtenoth tower). *Assume that q is an even power of a prime. There exists a family of $[n, k, d]$-linear codes over $\mathbb{F}_q$ with efficient encoding and decoding algorithms and $k + d \geq n(1 - \frac{1}{\sqrt{q}-1}) + 1$. Here $g = \frac{n}{\sqrt{q}-1}$.*

If the curve is a projective line, then the genus g is equal to 0 and the algebraic geometry code defined above is a Reed-Solomon code. Now we instantiate the above algebraic geometry code to Corollary 1 to obtain a ramp HIM.

Proposition 3. *Let $\mathcal{X}/\mathbb{F}_q$ be an algebraic curve of genus g with at least $m+n+1$ pairwise distinct rational points. If $g-1 < m \leq n$, then for any κ there exists an $(n, m; r, p)_q$-ramp HIM with $r \leq n+g$ and $p \geq n-g$.*

Proof. Put $\ell = m+n$ and $\kappa = n-g+1$. By Proposition 2, the code $C(\mathcal{P}, \kappa P_\infty)$ is a q-ary $[m+n, n]$-linear code with minimum distance $d \geq m+n-\kappa = m-g+1$ and dual distance $d^\perp \geq \kappa - 2g + 2 = n-g+1$. By Corollary 1, there is an $(n, m; r, p)$-ramp HIM with $d = n+m-r+1$ and $p+1 = d^\perp$. This gives $r = n+m+1-d \leq n+g$ and $p = d^\perp - 1 \geq n-g$. This completes the proof.

Note that if the base curve is a projective line, then the genus $g = 0$. Thus, we obtain an $(n, m; n, n)$-ramp HIM, i.e., an $(n, m)_q$-HIM.

Theorem 3. *If $q \geq 4$ is an even power of a prime, then there exists a family of $(n, m; r, p)$-ram HIM with $n \to \infty$, $m \geq \frac{n}{\sqrt{q}-1}$ and*

$$\limsup_{n\to\infty} \frac{r}{n} \leq 1 + \frac{2}{\sqrt{q}-1}, \quad \liminf_{n\to\infty} \frac{p}{n} \geq 1 - \frac{2}{\sqrt{q}-1}.$$

Furthermore, this family can be constructed in time $O(n^3)$.

Proof. Let $\{\mathcal{X}/\mathbb{F}_q\}$ be a family of algebraic curves given in [22]. Then we have $N(\mathcal{X}) \geq 1 + g(\mathcal{X})(\sqrt{q}-1)$. Put $g = g(\mathcal{X})$ and let $g-1 < m \leq n$ satisfy $N(\mathcal{X}) = n+m$. By Proposition 3, there exists a family of $(n, m; r, p)_q$-ramp HIMs with $r \leq n+g$ and $p \geq n-g$. As $m \leq n$ and $\frac{g}{m+n} \to \frac{1}{\sqrt{q}-1}$, we have $\frac{g}{n} \leq \frac{2}{\sqrt{q}-1}$. The desired result follows. As the Riemann-Roch space $\mathcal{L}(\kappa P_\infty)$ can be constructed in time $O(\kappa^3)$ [40], A generator matrix of $C(\mathcal{P}, \kappa P_\infty)$ can be constructed in time $O(n^3)$. By Corollary 1, the corresponding ramp HIM can be constructed in time $O(n^3)$ as well.

Corollary 3. *If $q = O(1/\epsilon^2)$ for a real $\epsilon \in (0, 1)$, then there exists a family of $(n, n; (1+\epsilon)n, (1-\epsilon)n)$-ramp HIM with $n \to \infty$. Furthermore, this family can be constructed in time $O(n^3)$.*

4 Perfectly secure MPC for $t < \frac{n(1-\epsilon)}{3}$ over constant-size fields

In this section, we will present a perfectly secure MPC protocol over constant-size fields by modifying the one in [5]. The challenge is to replace each gadget over large field in [5] with the one over constant-size fields. We emphasize that our security proof follows the line of [5]. The missing proof can be found in [5] such as the player elimination and so on. Since our MPC protocol is perfectly secure, most of the efforts are taken to detect the corruptions and remove the corrupted parties.

4.1 Arithmetic secret sharing schemes

Let us briefly explain the downside of the Shamir secret sharing scheme. Since the Shamir secret sharing scheme is derived from polynomial evaluation, the number of parties is at most the size of underlying field. If an n-parties MPC protocol securely evaluates an arithmetic circuit over the constant-size field $\mathbb{F}_q$, then Shamir secret sharing scheme will cause a $\Omega(\log n)$ overhead by embedding this constant field $\mathbb{F}_q$ into $\mathbb{F}_{q^r}$ with $q^r \geq n$. Thus, it is desirable to design secret sharing scheme over constant-size field. Our protocol utilizes the gap ϵn to simultaneously compute $\Omega(n)$ instances. We emphasize that although we present our perfectly secure MPC protocol over field of size $\Omega(\frac{1}{\epsilon^2})$, it is possible to construct perfectly secure MPC protocol over binary field by replacing each gadget with the one over the binary field.

An arithmetic secret sharing scheme [15] is a generalization of the Shamir secret sharing scheme which can be instantiated over any constant-size field. The formal definition of arithmetic secret sharing scheme is tedious and too general for our application. We briefly explain the motivation of this scheme and only provide the necessary definitions for our purpose. We note that the Shamir secret sharing scheme supports multiplication, i.e., the component-wise product of two sharings from t-threshold Shamir secret sharing scheme is a sharing from $2t$-threshold Shamir secret secret sharing scheme. It can be generalized to component-wise product of d sharings. In arithmetic secret sharing scheme, we first have a base scheme called C and let the component-wise product of d sharings consist of a scheme C^{*d}. We require that the sharing in C^{*d} can be used to recover the product of d secrets corresponding to these d sharings. Moreover, the base scheme must have t-privacy and r-reconstruction. These are the properties necessary for MPC protocols. In this sense, the arithmetic secret sharing scheme captures the essence of the Shamir secret sharing scheme for MPC application. The merit of such generalization is that we can find a large number of codes except Reed-Solomon codes meet the definition of arithmetic secret sharing and are applicable to MPC protocol and other cryptography primitives.

Definition 7. *Let $C \subseteq \mathbb{F}_q^s \times \mathbb{F}_q^n$ (packed secret sharing scheme for $s > 1$) be a linear secret sharing scheme whose secret space is $\mathbb{F}_q^s$ and share space is $\mathbb{F}_q^n$.*[6] *We say C is t-strongly multiplicative secret sharing scheme*[7] *if*

1. *C has t-privacy: for any subset A of $[n]$ of size at most t, and any pair of secret $\mathbf{s}, \mathbf{s}' \in \mathbb{F}_q^s$, one has that $|\{\mathbf{c} \in C : \mathbf{c}_A = \mathbf{s}\}| = |\{\mathbf{c} \in C : \mathbf{c}_A = \mathbf{s}'\}|$.*
2. *C has $(n-2t)$-reconstruction: i.e., for any subset A of $[n]$ of size at least $n-2t$ and $\mathbf{c}, \mathbf{c}' \in C^{*2}$, one has that $\mathbf{c}_A \neq \mathbf{c}'_A$.*
3. *The secret sharing scheme $C^{\star 2} = span_{\mathbb{F}_q}\{\mathbf{c}_1 \star \mathbf{c}_2 : \mathbf{c}_1, \mathbf{c}_2 \in C\}$ has $(n-t)$-reconstruction, i.e., for any subset A of $[n]$ of size at least $n-t$ and $\mathbf{c}, \mathbf{c}' \in C^{*2}$, one has that $\mathbf{c}_A \neq \mathbf{c}'_A$.*

[6] We use 0 to represent the index of the secret and $[n]$ to represent n indices of the shares.

[7] In [15], a t-strongly multiplicative LSSS on n players for $\mathbb{F}_q^k$ over $\mathbb{F}_q$ is also called an $(n, t, 2, t)$-arithmetic secret sharing scheme with secret space $\mathbb{F}_q^k$ and share space $\mathbb{F}_q$

It is desirable to fix the field size q and let the number of parties n approach infinity. If the ratio $\frac{t}{n}$ is a constant, such t-strongly SSS is asymptotically good. The instantiation of asymptotically good t-strongly multiplicative SSS is based on algebraic geometry code. By applying Garcia-Stichtenoth tower [22], we obtain the following construction.

Proposition 4 (via Garcia-Stichtenoth tower). *Assume q is an even power of a prime. Let $\gamma \in \left(0, \frac{1}{3} - \frac{2}{\sqrt{q}-1}\right)$. Then there exists a sequence $\{C_i\}$ of q-ary LSSS on n_i players with the secret space $\mathbb{F}_{q^{k_i}}$, the share space $\mathbb{F}_q$ such that*

(i) $\lim_{i\to\infty} \frac{k_i}{n_i} = \gamma$.
(ii) *C_i has r_i-reconstruction and t_i-privacy satisfying $\frac{t_i}{n_i} = \frac{r_i}{n_i} - \frac{2}{\sqrt{q}-1} - \gamma$.*
(iii) *$C_i^{*2} = span_{\mathbb{F}_q}\{\mathbf{c}_1 \star \mathbf{c}_2 : \mathbf{c}_1, \mathbf{c}_2 \in C_i\} \subseteq \mathbb{F}_q^{k_i} \times \mathbb{F}_q^{n_i}$ is a SSS with $2r_i$-reconstruction and $2t_i$-privacy.*
(iv) *The sharing and reconstruction algorithms of C_i and C_i^{*2} can be efficiently implemented.*
(v) *The decoding algorithm of C_i can efficiently correct up to $\frac{n_i - r_i - 1}{2}$ corrupted shares for any sharing in C_i*

If $2r_i \leq n_i - t_i$, then C_i is a t_i-strongly multiplicative LSSS. Let $r_i = \frac{n_i}{3}$ and we obtain the following SSS.

Corollary 4. *Let $q \approx \frac{144}{\epsilon^2} = O(\frac{1}{\epsilon^2})$ and $\gamma = \frac{\epsilon}{6}$. There exists a family of $(\frac{1-\epsilon}{3})n_i$-strongly multiplicative secret sharing scheme $C_i \subseteq \mathbb{F}_q^{\frac{\epsilon n_i}{6}} \times \mathbb{F}_q^{n_i}$ when $n_i \to \infty$. This multiplicative SSS has $\frac{(1-\epsilon)n_i}{3}$-privacy and $\frac{n_i}{3}$-reconstruction.*

The player elimination introduced in [31] is used to transform a non-robust protocol into a robust protocol with no additional costs. Each time the inconsistent sharings are detected, this player elimination protocol is initiated to localize and remove a pair of parties containing at least one corrupted party from the preprocessing phase. To apply this protocol, our arithmetic secret sharing scheme should be compatible with the reduced number of the parties and corrupted parties. In the following theorem, we show how to obtain a t_i'-strongly multiplicative LSSS from a t_i-strongly multiplicative LSSS with $t_i' < t_i$. Then, we can apply Corollary 4 for any privacy t_i and number n_i of parties.

Theorem 4. *Assume that $C_i \subseteq \mathbb{F}_q^s \times \mathbb{F}_q^{n_i}$ is a t_i-strongly multiplicative LSSS on n_i players in Proposition 4. Then, there exists a t_i'-strongly multiplicative LSSS $C' \subseteq \mathbb{F}_q^s \times \mathbb{F}_q^{n_i'}$ with $n_i' = n_i - 2(t_i - t_i')$. Moreover, $r_i' \leq \frac{n_i'}{3}$ and $t_i' \leq \frac{n_i'(1-\epsilon)}{3}$ if $r_i = \frac{n_i}{3}$ and $t_i = \frac{n_i(1-\epsilon)}{3}$.*

Proof. Since C_i is a t_i-strongly multiplicative LSSS, we have $2r_i \leq n_i - t_i$. We first fix the number of parties n_i and the dimension of secret space s_i, and let the privacy be t_i' and reconstruction be $r_i' = r_i - (t_i - t_i')$ in Proposition 4. Such SSS $\hat{C}_i$ exists as

$$\frac{t_i'}{n_i} = \frac{t_i}{n_i} - \frac{(t_i - t_i')}{n_i} = \frac{r_i - (t_i - t_i')}{n_i} - \frac{2}{\sqrt{q}-1} - \gamma = \frac{r_i'}{n_i} - \frac{2}{\sqrt{q}-1} - \gamma$$

We obtain C_i' by puncturing the last $2(t_i - t_i')$ shares of $\hat{C}_i$. The privacy and reconstruction of C_i' are exactly the same as that of $\hat{C}_i$ which are t_i' and $r_i' = r_i - (t_i - t_i')$ respectively. Similarly, the reconstruction of $C_i'^{*2}$ is $2r_i'$. The proof is completed as $2r_i' = 2r_i - 2(t_i - t_i') \leq n_i' - t_i \leq n_i' - t_i'$. It is clear that $r_i' \leq \frac{n_i'}{3}$ and $t_i' \leq \frac{n_i'(1-\epsilon)}{3}$ if $r_i = \frac{n_i}{3}$ and $t_i = \frac{n_i(1-\epsilon)}{3}$.

Remark 1 *Although we only present t-strongly multiplicative SSS over field of size $\Omega(1/\epsilon^2)$, it is possible to construct t-strongly multiplicative SSS over binary field [38]. The same trick in Theorem 4 can be applied to this SSS. The resulting SSS becomes the building block of perfectly secure MPC over the binary field.*

Remark 2 *To check the consistency of this linear secret sharing scheme, we note that it suffices to run the reconstruction algorithm of this linear secret sharing scheme and then compare the shares recovered by this reconstruction algorithm with the shares at hand. Since our secret sharing scheme is obtained from algebraic geometry code with an efficient decoding algorithm, this reconstruction algorithm is in fact the decoding algorithm for this algebraic geometry code.*

4.2 Randomization based on ramp hyper-invertible matrices

Protocol RandEl(d)

Setup: a set of n' parties $\mathcal{I} = \{P_1, \ldots, P_{n'}\}$ and at most $t' \leq \frac{n'(1-\epsilon)}{3}$ of them are corrupted. Let M be an $(n', n', n'(1+\epsilon), n'(1-\epsilon))$-ramp hyper-invertible matrix over $\mathbb{F}_q$ in Definition 6.

- For $P_i \in \mathcal{I}$, P_i generates a random secret $\mathbf{s}_i \in \mathbb{F}_q^s$ and shares this secret among parties in $\mathcal{I}$ by invoking Σ_d.
- All parties locally compute $([\mathbf{r}_1]_d, \ldots, [\mathbf{r}_{n'}]_d)^T = M([\mathbf{s}_1]_d, \ldots, [\mathbf{s}_{n'}]_d)^T$.
- For $i = T+1, \ldots, n'$ for some $T \leq t'$, all parties open $\mathbf{r}_i$ to P_i. P_i checks the consistency of $[\mathbf{r}_i]_d$ and becomes unhappy if the sharing is not correct.
- Output the remaining unopened sharings $[\mathbf{r}_1]_d, \ldots, [\mathbf{r}_T]_d$.

The secret sharing scheme in Corollary 4 is our building block for our MPC protocol. Our MPC protocol starts with n parties and at most $t = \frac{(1-\epsilon)n}{3}$ of them are corrupted where ϵ can be an arbitrarily small value. In what follows, we fix $t = \frac{(1-\epsilon)n}{3}$. During the preprocessing phase, the player elimination technique introduced in [31] divides the computation into $O(n)$ segments and in each segment the protocol tries to identify the corrupted parties when inconsistent shares are detected. This protocol can locate a pair of parties such that at least

one of them is corrupted. Then, this pair of parties are removed from the protocol. The number of parties and the number of corrupted parties are updated to $n-2$ and $t-1$ respectively. All remaining parties repeat current segment. Thus, in the preprocessing phase, our protocol assumes that the number of parties and corrupted parties are $n' \leq n$ and $t' \leq t$ respectively.

In what follows, we assume $q = O(\frac{1}{\epsilon^2})$ and our MPC protocol is defined over $\mathbb{F}_q$.

Let $\Sigma_d \in \mathbb{F}_q^s \times \mathbb{F}_q^{n'}$ be an SSS in Theorem 4 with privacy $d \in [t', \frac{n'(1-\epsilon)}{3}]$, reconstruction $r = \frac{n'}{3}$ and $s = \frac{\epsilon n}{6}$. Due to the player elimination protocol, the value of d may decrease throughout the prepocessing protocol. The initial value of d is t. Denote by $[\mathbf{s}]_d = (x_1, \ldots, x_n) \in \Sigma_d$ the packed secret-sharing of $\mathbf{s} \in \mathbb{F}_q^s$.[8] Since our SSS supports multiplication, to open the secret of $[\mathbf{s}_1]_d \star [\mathbf{s}_2]_t$ safely, we need to mask it with $[\mathbf{r}]_{2d}$, a sharing of a random vector $\mathbf{r}$ generated by Σ_{2d}. This is because Σ_{2d} is actually the "squared" Σ_d, i.e., Σ_d corresponds to C_i and Σ_{2d} corresponds to C_i^{*2} by Proposition 4. In this sense, Σ_{2d} has $2d$-privacy and $2r$-reconstruction.

Proposition 5. *Assume that Σ_d in the Protocol **RandEl** has d-privacy with $d \geq t'$. If $T \leq t'$ and all honest parties are happy, then $[\mathbf{r}_1]_d, \ldots, [\mathbf{r}_T]_d$ are correct sharings of uniformly random secrets $\mathbf{r}_1, \ldots, \mathbf{r}_T$ and the adversary learns no information about them. The total communication complexity is $O(n^2)$ for generating $\Omega(n)$ correct sharings.*

Proof. The proof follows the same step in [5] except that we replace hyper-invertible matrix with ramp hyper-invertible matrix in Definition 5. For convenience, we use $[\mathbf{r}_i]$ to represent the sharing $[\mathbf{r}_i]_d$ in the proof. We first consider the robustness, i.e., the unopened sharings $[\mathbf{r}_1], \ldots, [\mathbf{r}_T]$ are correct. Let $S \subseteq \{T+1, \ldots, n'\}$ be the index set of honest parties and $\bar{S} = \{T+1, \ldots, n'\}/S$. Since there are at most t' corrupted parties in $\mathcal{I}$, $|S| \geq n' - T - t' \geq n' - 2t'$. $[\mathbf{r}_i]_{i \in S}$ are correct sharings checked by the honest parties in S as all parties do not complain. Moreover, there are at least $n' - t'$ honest parties in $\mathcal{I}$. Let H be the collection of honest parties in $\mathcal{I}$ and we have $|H| \geq n' - t'$. Observe that $|H| + |S| \geq 2n' - 3t' \geq n + \epsilon n$ as $t' \leq \frac{n'(1-\epsilon)}{3}$. This implies that $n' - t'$ sharings $[\mathbf{s}_i]_{i \in H}$ generated by the honest parties together with $[\mathbf{r}_i]_{i \in S}$ uniquely determine T sharings $[\mathbf{r}_i]_{i \in [T]}$ as M is an $(n', n', n'(1+\epsilon), n'(1-\epsilon))$-ramp HIM. Since $[\mathbf{s}_i]_{i \in H}$ and $[\mathbf{r}_i]_{i \in S}$ are correct sharings, the unopened sharings $[\mathbf{r}_i]_{i \in [T]}$ are correct as well.

We proceed to the privacy argument. The secret sharing scheme Σ has $d \geq t'$-privacy. This implies that the adversary can not obtain $\mathbf{r}_i$ from any t' shares of $[\mathbf{r}_i]$. Moreover, the adversary knows the random vectors $\mathbf{r}_i$ for $i \in \bar{S}$ opening to the corrupted parties and $\mathbf{s}_i$ for $i \in [n'] \setminus H$ generated by the corrupted parties. They are at most $2t'$ vectors in total. If we fix these vectors, as M is an $(n', n', n'(1+\epsilon), n'(1-\epsilon))$-ramp HIM and $|\bar{S} \cup [T]| + |[n] \setminus H| \leq 3t' \leq n'(1-\epsilon)$, there is a surjection from $\mathbf{s}_i, i \in H$ to $\mathbf{r}_i, i \in [T]$. Since $\mathbf{s}_i, i \in H$ are distributed

[8] In the Shamir SSS, one can identify this privacy d as the degree of polynomials.

uniformly at random, $\mathbf{r}_i, i \in [T]$ are distributed uniformly at random as well. This means the distribution $\mathbf{r}_i, i \in [T]$ is independent of $\mathbf{r}_i$ for $i \in \bar{S}$ and $\mathbf{s}_i$ for $i \in [n'] \setminus H$.

As for the communication complexity, we note that each party sends and receives $n' - 1$ shares. Thus, the total communication complexity is $O(n'^2) = O(n^2)$. If no one complains, invoking **RandEl** can generate $\Omega(n)$ correct sharings and thus each sharing cost $O(n)$ communication complexity. We also note that each sharing belongs to a packed secret sharing scheme and thus the amortized communication complexity is further reduced to constant.

The **RandEl** was invoked in [5] to produce random double sharings. Assume there are t' corrupted parties and n' parties remaining in the preprocessing phase. Instead of using HIM, we use ramp HIM over constant-size field to generate the random double sharings. All parties want to obtain the double sharings $([\mathbf{a}]_d, [\mathbf{a}]_{d'})$ of random vector $\mathbf{a}$ for some $t' \leq d, d' \leq n' - t'$. In order to do that, one can modify **RandEl** protocol as follows. P_i generates random double sharings $[\mathbf{s}_i]_d$, $[\mathbf{s}_i]_{d'}$ and applies the ramp hyper-invertible matrix to obtain $[\mathbf{r}_i]_d$ and $[\mathbf{r}_i]_{d'}$. For $i = T+1, \ldots, n$, P_i not only checks the consistency of $[\mathbf{r}_i]_d$ and $[\mathbf{r}_i]_{d'}$ but also makes sure that the opened secrets $\mathbf{r}_i$ are the same. This will force the remaining unopened sharings $[\mathbf{r}_i]_d$ and $[\mathbf{r}_i]_{d'}$ to be correct and associated with the same secret. We call this modified protocol **DoubleSharings**. The same argument implies the following result.

Corollary 5. *Assume that $t' \leq d, d' \leq n' - t'$. If $T \leq t'$ and all honest parties are happy, then **DoubleSharings** will output correct sharings $([\mathbf{r}_1]_d, [\mathbf{r}_1]_{d'}), \ldots, ([\mathbf{r}_T]_d, [\mathbf{r}_T]_{d'})$ with uniformly random secrets $\mathbf{r}_1, \ldots, \mathbf{r}_T$ and the adversary learns no information about them. The total communication complexity is $O(n^2)$ for generating $\Omega(n)$ correct sharings.*

4.3 Public Reconstruction

The public reconstruction protocol is used to efficiently and robustly open the secret [18, 5]. The idea is that instead of reconstructing one secret, this protocol allows all parties to simultaneously reconstruct $\Omega(n)$ secrets. Meanwhile, the communication complexity keeps the same $O(n^2)$ and thus the amortized communication complexity is reduced to $O(n)$. To achieve this goal, this protocol treats $k = \Omega(n)$ secrets as a message of length k and re-encodes this message to a codeword of length n' such that this linear code has minimum distance at least $2t' + 1$. Each party reconstructs one secret corresponding to one component of this codeword. Since there are at most t' corrupted parties, this message can be robustly recovered subject to at most t' errors. Our Protocol **ReconPub** is a generalization of the counterpart in [18, 5]. Because our protocol is defined over constant-size field, we resort to algebraic geometry codes for error correction.

Theorem 5. *For $d \leq t$, **ReconPub** robustly reconstructs the secrets $\mathbf{a}_1, \ldots, \mathbf{a}_T$ towards all parties in $\mathcal{I}$. For $d \leq 2t'$, **ReconPub** detectably reconstructs the*

Protocol ReconPub$(d, [\mathbf{a}_1]_d, \ldots, [\mathbf{a}_T]_d)$

Setup: a set of n' parties $\mathcal{I} = \{P_1, \ldots, P_{n'}\}$ and at most $t' \leq \frac{n'(1-\epsilon)}{3}$ of them are corrupted. Let $T = n'(1-\epsilon) - 2t' - 1 = \Omega(n)$ and $M = (m_{ij})_{n' \times T}$ be a generator matrix of a $[n', T, 2t'+1]$ linear code C over $\mathbb{F}_q$ in Corollary 2 as $2t'+1+T = n'(1-\epsilon)$.

Input: T sharings $[\mathbf{a}_1]_d, \ldots, [\mathbf{a}_T]_d \in \Sigma_d$ in Corollary 4.

- For $i = 1, \ldots, n'$, the parties in $\mathcal{I}$ locally compute $[\mathbf{r}_i]_d = \sum_{j=1}^{T} m_{ij}[\mathbf{a}_j]_d$.
- The parties in $\mathcal{I}$ send their shares of $[\mathbf{r}_i]_d$ to P_i.
- P_i checks the consistency of $[\mathbf{r}_i]_d$. If $d \leq t'$, P_i robustly reconstructs the secret $\mathbf{r}_i$ by invoking decoding algorithm of Σ_d and sends them to other parties in $\mathcal{I}$. Otherwise if $d \leq 2t'$, P_i either reconstructs the secret $\tilde{\mathbf{r}}_i$ and sends it to other parties in $\mathcal{I}$ or becomes unhappy if there are inconsistent shares.
- If no one becomes unhappy, the parties in $\mathcal{I}$ robustly reconstruct $\mathbf{a}_1, \ldots, \mathbf{a}_T$ from $\tilde{\mathbf{r}}_1, \ldots, \tilde{\mathbf{r}}_{n'}$. More precisely, write $\tilde{\mathbf{r}}_i = (r_{i1}, \ldots, r_{is}) \in \mathbb{F}_q^s$ and decode the codeword $(r_{1j}, \ldots, r_{n'j}) \in C$ for $j = 1, \ldots, s$ to obtain the message $(a_{1j}, \ldots, a_{Tj})$ for $j = 1, \ldots, s$.

Output: $\mathbf{a}_i = (a_{i1}, \ldots, a_{is}), i = 1, \ldots, T$.

secrets $\mathbf{a}_1, \ldots, \mathbf{a}_T$ *towards all parties in* $\mathcal{I}$*. The total communication complexity is* $O(n'^2)$

Proof. We prove the first claim. Without loss of generality, we assume $d = t$. For each party $P_i \in \mathcal{I}$, $[\mathbf{r}_i]_d$ consists of at most t' incorrect shares. Since $[\mathbf{r}_i]_d \in \Sigma_d$, by Proposition 4, the reconstruction of Σ_d is $r = t + \epsilon n'/3$ and it can correct up to $\frac{(n'-r)}{2} \geq t'$ errors. Thus, the honest party P_i can robustly reconstruct $\mathbf{r}_i$. After this reconstruction, there are at least $n' - t'$ correct secrets $\mathbf{r}_i$. Thus, the decoding algorithm of C can correct errors and output correct messages.

We proceed to the second claim $d = 2t'$. The argument is divided into two cases.

1. Some honest party P_j receives corrupted shares of $[\mathbf{r}_i]_d$ from the adversary. By Proposition 4 and Corollary 4, the minimum distance of $\Sigma_{2t'}$ is at least $n' - 2r' \geq n'/3$. Since there are at most $t' < n'/3$ corrupted shares, P_j can detect them and become unhappy.
2. All honest parties receive consistent shares. This implies that there are at least $n' - t'$ correct secrets $\mathbf{r}_i$. Thus, the decoding algorithm of C can correct errors and output correct messages.

As for the communication complexity, party P_i sends his share of $[\mathbf{r}_j]_d$ to P_j and the secret $\mathbf{r}_i$ to all other parties. Thus, the total communication complexity is $O(n^2)$ for opening $T = \Omega(n)$ secrets. The proof is completed.

Now we proceed to generate Beaver triples $([\mathbf{a}]_t, [\mathbf{b}]_t, [\mathbf{c}]_t)$ to prepare for multiplication gates. Random sharings $[\mathbf{a}]_t$ and $[\mathbf{b}]_t$ could be generated by invoking

RandEl. Since arithmetic secret sharing has strong multiplicativity, all parties could locally compute $[\mathbf{c}]_{2t}$. The key of transforming $[\mathbf{c}]_{2t}$ to $[\mathbf{c}]_t$ is degree reduction, which can be done with double sharings generated in **RandEl**.

Protocol Triples

Setup: The set of parties $\mathcal{I} = \{P_1, \ldots, P_{n'}\}$, the number of parties n' and the number of corrupted parties t'.

- The parties in $\mathcal{I}$ invoke **DoubleSharings** three times to generate $([\mathbf{a}_1]_t, [\mathbf{a}_1]_{t'}), \ldots, ([\mathbf{a}_T]_t, [\mathbf{a}_T]_{t'})$, $([\mathbf{b}_1]_t, [\mathbf{b}_1]_{t'}), \ldots, ([\mathbf{b}_T]_t, [\mathbf{b}_T]_{t'})$ and $([\mathbf{r}_1]_t, [\mathbf{r}_1]_{2t'}), \ldots, ([\mathbf{r}_T]_t, [\mathbf{r}_T]_{2t'})$.
- The parties in $\mathcal{I}$ locally compute $[\mathbf{d}_k]_{2t'} = [\mathbf{a}_i]_{t'} \star [\mathbf{b}_i]_{t'} + [\mathbf{r}_i]_{2t'}$ for $i = 1, \ldots, T$.
- The parties in $\mathcal{I}$ invoke **ReconPub**$(2t', [\mathbf{d}_1]_{2t'}, \ldots, [\mathbf{d}_T]_{2t'})$ to publicly reconstruct $\mathbf{d}_1, \ldots, \mathbf{d}_T$,
- The parties in $\mathcal{I}$ locally compute $[\mathbf{c}_i]_t = \mathbf{d}_i - [\mathbf{r}_i]_t$.

Output: T triples $([\mathbf{a}_1]_t, [\mathbf{b}_1]_t, [\mathbf{c}_1]_t), \ldots, ([\mathbf{a}_T]_t, [\mathbf{b}_T]_t, [\mathbf{c}_T]_t)$.

Theorem 6. *If all honest parties are happy, Protocol **Triples** successfully outputs Beaver triples* $([\mathbf{a}_i]_t, [\mathbf{b}_i]_t, [\mathbf{c}_i]_t)_{i\in[T]}$[9] *such that* $\mathbf{a}_i$ *and* $\mathbf{b}_i$ *are uniformly random vectors and* $\mathbf{c}_i = \mathbf{a}_i \star \mathbf{b}_i$*. Moreover, the total communication complexity of **Triples** is* $O(n^2)$*.*

Proof. Corollary 5 shows that if all honest parties are happy, then **DoubleSharings** generates correct sharings $([\mathbf{a}_i]_{t'}, [\mathbf{b}_i]_{t'}, [\mathbf{r}_i]_{2t'})_{i\in[T]}$ such that $\mathbf{a}_i, \mathbf{b}_i, \mathbf{r}_i$ are uniformly random vectors by Corollary 5. Theorem 5 shows that **ReconPub** can reconstruct correct secrets towards all parties if all honest parties are happy. It is clear that $\mathbf{c}_i = \mathbf{a}_i \star \mathbf{b}_i$ as our SSS are multiplicative by Proposition 4. The privacy argument is straightforward. **DoubleSharings** does not reveal any information to the adversary by Corollary 5. Moreover, **ReconPub** only opens the random elements which contain no information about $\mathbf{c}_i$. The total communication complexity is the cost of invoking **DoubleSharings** and **ReconPub**, which is $O(n^2)$ due to Corollary 5 and Theorem 5.

4.4 Put Together

In this subsection, we briefly explain how to replace the protocols in [5] with our new protocols so as to obtain perfectly secure MPC for $t < \frac{n(1-\epsilon)}{3}$ over constant-size fields. The prominent feature of our MPC protocol is constant share size. We use the same player elimination protocol in [5] containing fault detection, fault localization and player elimination.

Put everything together, we obtain the following theorem.

[9] Invoking **Triples** once can generate $T = n'(1-\epsilon) - 2t' - 1 = \Omega(n)$ triples. Each triple contains $\frac{\epsilon n}{6}$ secrets.

Theorem 7. *The protocol* ***PreprocessingPhase*** *generates* $c_M + c_R + c_I$ *independent random Beaver triples* $[\mathbf{a}_i]_t, [\mathbf{b}_i]_t, [\mathbf{a}_i \star \mathbf{b}_i]_t$ *with independently random vectors* $\mathbf{a}_i, \mathbf{b}_i \in \mathbb{F}_q^{\frac{\epsilon n}{6}}$. *The total communication complexity is* $O((c_I + c_M + c_R)n + n^3)$. *The amortized communication complexity of generating one triple is* $O(n)$. [10]

Proof. The proof is quite straightforward since we have already proven Theorem 6. If every party is happy, then **Triples** guarantees that all the Beaver triples generated in this segment is correct. Otherwise, all parties invoke the player elimination protocol to localize a pair of parties containing at least one corrupt party. The privacy argument can be derived directly from Theorem 6. It remains to compute the communication complexity. We note that we divide the preprocessing phase into t segments. In each segment, we either remove two parties or complete this segment and obtain ℓ Beaver triples. Since there are n parties, we invoke at most $t + \frac{n}{2} = O(t)$ segments. For each segment, we invoke **Triples** $\frac{\ell}{tT}$ times which incurs $O(\frac{\ell n^2}{tT}) = O(\ell)$ communication complexity. The player elimination protocol in Appendix incurs the same amount of communication complexity in this segment plus the cost of three broadcasts $O(n^2)$. Thus, the total communication complexity for preprocessing phase is $O(\ell t + n^3) = O((c_I + c_M + c_R)n + n^3)$.

Protocol PreprocessingPhase

Setup: the set of actual parties is $\mathcal{I} = \{P_1, \ldots, P_{n'}\}$, the number of parties is $n' = n$ and the number of corrupted parties is $t' = t$. The preprocessing phase generates $\ell = c_I + c_M + c_R$ Beaver triples.

- For $1st, \ldots, t$th segment,
  - Each party in $\mathcal{I}$ sets his happy-bit to happy.
  - The party in $\mathcal{I}$ invokes **Triples** $\lfloor \frac{\ell}{tT} \rfloor$ times to generate $\frac{\ell}{t}$ Beaver triples $([\mathbf{a}]_{t'}, [\mathbf{b}]_{t'}, [\mathbf{a} \star \mathbf{b}]_{t'})$.
  - If there is at least one party unhappy, invoke player elimination protocol to localize a pair of parties $P' = \{P_i, P_j\}$.
  - Set $\mathcal{I} = \mathcal{I} \setminus P'$ and $n' = n' - 2, t' = t' - 1$. Repeat this segment.

Theorem 8. *The protocol* ***OnlinePhase*** *perfectly securely evaluates a single instruction multiple data (SIMD) circuit with* $\frac{\epsilon n c_I}{6}$ *input,* $\frac{\epsilon n c_M}{6}$ *multiplication,* $\frac{\epsilon n c_R}{6}$ *random gates and* D_M *depth in the presence of* $t = \frac{(1-\epsilon)n}{3}$ *actively corrupted parties, given* $c_I + c_M + c_R$ *pre-shared multiplication triples. The total*

[10] Since such triple consists of packed secret sharing scheme, we can further reduce the amortized communication complexity to constant if we evaluates $\Omega(n)$ instances of the same circuit in the online phase.

Protocol OnlinePhase

Input Gate: (P_i input $\mathbf{s}$)

- The parties in $\mathcal{I}$ send their shares of $[\mathbf{r}]_t$ to P_i. P_i robustly reconstructs $\mathbf{r}$ by running decoding algorithm in Proposition 4.
- P_i broadcasts $\mathbf{s}-\mathbf{r}$ and the parties in $\mathcal{I}$ locally compute $[\mathbf{s}]_t = \mathbf{s}-\mathbf{r}+[\mathbf{r}]_t$.

Addition Gate: The parties in $\mathcal{I}$ locally compute $[\mathbf{x}+\mathbf{y}]_t = [\mathbf{x}]_t + [\mathbf{y}]_t$.
Scalar Gate: The parties in $\mathcal{I}$ locally compute $[\lambda\mathbf{x}]_t = \lambda[\mathbf{x}]_t$
Random Gate: Pick a random sharing $[\mathbf{r}]_t$ associated with this gate.
Multiplication Gate: Up to $\frac{T}{2}$ multiplication gates are processed simultaneously. The input of each multiplication gate is $[\mathbf{x}_i]_t, [\mathbf{y}_i]_t$ for $i = 1, \ldots, T/2$. The Beaver triples $([\mathbf{a}_i]_t, [\mathbf{b}_i]_t, [\mathbf{c}_i]_t), i = 1, \ldots, T/2$ are given.

- For $i = 1, \ldots, T/2$, the parties in $\mathcal{I}$ locally compute $[\mathbf{d}_i]_t = [\mathbf{x}_i]_t - [\mathbf{a}_i]_t$ and $[\mathbf{e}_i]_t = [\mathbf{y}_i]_t - [\mathbf{b}_i]_t$.
- Invoke **ReconPub** to robustly reconstruct the secrets $\mathbf{d}_i, \mathbf{e}_i$ for $i = 1, \ldots, T/2$.
- The parties in $\mathcal{I}$ locally compute $[\mathbf{x}_i \star \mathbf{y}_i]_t = \mathbf{d}_i \star \mathbf{e}_i + \mathbf{e}_i \star [\mathbf{x}_i]_t + \mathbf{d}_i \star [\mathbf{y}_i]_t + [\mathbf{c}_i]_t$ for $i = 1, \ldots, T/2$.

Output Gate: (Output $[\mathbf{s}]_t$ to all parties) The parties in $\mathcal{I}$ send their shares of $[\mathbf{s}]_t$ to other parties. All parties robustly reconstruct $\mathbf{s}$ by running decoding algorithm in Proposition 4.

communication complexity is $O((c_I + c_M + c_R)n + D_M n^2 + n^3)$ and thus the amortized communication complexity of computing each gate is $O(\frac{D_M n + n^2}{c_I + c_M + c_R})$. If $c_I + c_M + c_R$ is bigger than $n^2 + D_M n$, the amortized communication complexity for each gate is a constant.

Proof. The online phase follows the line of *Computationphase* protocol in [5]. Since our circuit is a SIMD circuit, we use the packed secret sharing obtained in preprocessing phase to compute the *Computationphase* protocol. Each triple in the preprocessing phase can compute $\frac{\epsilon n}{6}$ instances simultaneously. Thus, it suffices to generate $c_I + c_M + c_R$ Beaver triples in the preprocessing phase. The total communication complexity in the preprocessing phase is $O((c_I + c_M + c_R)n + n^3)$ by Theorem 7.

We proceed to the online phase. At the input gate, we use a pre-shared random vector $\mathbf{r}$ to mask the input $\mathbf{s}$ and then broadcast the difference $\mathbf{s}-\mathbf{r}$. Thus, we broadcast n times and each broadcast can be simulated by communicating $O(n^2)$ bits. All parties obtain their shares of the secret $\mathbf{s}$ by locally computing $[\mathbf{r}]_t + \mathbf{s} - \mathbf{r}$. The addition and scalar gate can be done locally. Thus, the total communication complexity for computing input gates is $O(n^3 + c_I n)$. At the multiplication gate, the Beaver triple $([\mathbf{a}]_t, [\mathbf{b}]_t, [\mathbf{a} \star \mathbf{b}]_t)$ is used to securely compute a sharing of a product at the cost of two public reconstructions. The **ReconPub** amortizes the communication complexity of public reconstruction by reconstructing $T = \Omega(n)$ secrets simultaneously. Thus, we can evaluate $\Omega(n)$ multiplication gates by invoking **ReconPub** once. Since the secret space of our SSS has dimension $\frac{\epsilon n}{6}$,

this packed secret sharing scheme can evaluate $\frac{\epsilon n}{6}$ instances simultaneously. Each random gate picks a random sharing. Thus, the total communication complexity of random gates is $O(c_R n)$. This means the total communication complexity of computing multiplication gates is $O(c_M n + D_M n^2)$. Therefore, the total communication complexity of **OnlinePhase** is $O((c_I + c_M + c_R)n + D_M n^2 + n^3)$.

We proceed to the robustness argument. At the input gate, all parties use a random sharing generated in the preprocessing phase to generate the sharing of one input. No corruption happens in this stage. At the addition gate and scalar gate, all parties do local computation and no corruption happens. At the multiplication gate, all parties open secrets by invoking **ReconPub**. We note that the the sharings to be opened belong to Σ_t which can be error corrected by Proposition 4. Thus, the corruptions caused by the adversary in this stage will be corrected. At the output gate, we obtain the same conclusion as the sharings to be opened belong to Σ_t as well.

We proceed to the privacy argument. We note that the secret sharing scheme we use has t-privacy and there are $t' \leq t$ corrupted parties in the online phase. At the input gate, the input is masked by a random element and thus the element broadcasted is a random element revealing no information about the input. At the multiplication gate, the opened secrets are random elements which reveal no information. Thus, the adversary learns nothing except the output in the online phase.

Remark 3 *Since our MPC protocol uses the packed secret sharing scheme, to achieve constant amortized communication complexity, our MPC protocol must run over single instruction multiple data (SIMD) circuit which carries out the exact same computation to several inputs simultaneously. However, it is also possible to adapt it to other circuits although the protocol will be more complicated. We briefly explain the modification required for this goal. In [17], they propose a way to reroute the network. We can replace our double sharings with the sharings of random vectors and the permutation of their coordinates in the ramp HIM protocol. When we open the pair of secrets, we compare if the secret and the permutation of the secret are consistent. Thus, we can apply the technique in [17] to modify the circuit to achieve small communication complexity. The technique in [17] is to embed the computation in a special form of a universal circuit based on the so-called Bene's network [9] which requires the sharings with the permutation of their coordinates.*

5 MPC-in-the-head

5.1 Check consistency of shares via ramp HIM

The application of HIM in zero-knowledge proof was due to [8] where HIM was used to check the consistency of sharings. To check consistency of $n - 2t$ sharings, HIM requires $n + t$ additional sharings. Thus, the overhead of checking one sharing is roughly $\frac{t+n}{n-2t} = O(1)$ field element. The downside is that HIM

requires that $|\mathbb{F}| \geq 2n$. As a generalization of HIM, the ramp HIM is defined over constant-size field which can save the communication complexity.

Proposition 6. *Assume at most $t = \frac{1-\epsilon}{3}n$ parties of $P_1, \cdots, P_n$ are corrupted and Σ_d has d-privacy with $d \geq t$. Protocol **CheckConsistency**(d) verify the d-consistency of 2t secret sharings with zero error probability. It is t-private in the presence of semi-honest adversary and perfectly t-robust in the presence of malicious adversary.*

Proof. We use $[\boldsymbol{r}_i]$ to represent $[\boldsymbol{r}_i]_d$. We begin by proving that protocol **Check-Consistency**(d) is t-robust in the presence of malicious adversary. Let $H \subseteq [n]$ be the index set of honest parties and $\bar{H} = [n]/H$ be the index set of corrupted parties. Since there are at most t corrupted parties, $|H| \geq n - t$. If no party complains, $[\boldsymbol{s}_i]$ for $i \in H$ are correct sharings. Moreover, $[\boldsymbol{r}_i]$ for $i \in [2t+1, n]$ provided by the input client can not be corrupted by the adversary. The fact that $|H| + |[2t+1, n]| \geq 2n - 3t \geq (1+\epsilon)n$ and M is an $(n, n, (1+\epsilon)n, (1-\epsilon)n)$-ramp HIM implies that $[\boldsymbol{s}_i]$ for $i \in H$ and $[\boldsymbol{r}_i]$ for $i \in [2t+1, n]$ uniquely determine the sharings $[\boldsymbol{r}_i]$ for $i \in [2t]$. Since $[\boldsymbol{r}]$ for $i \in [2t]$ and $[\boldsymbol{s}_i]$ for $i \in H$ are correct sharings, $[\boldsymbol{r}_i]$ for $i \in [2t]$ are also correct sharings.

We proceed to the argument of t-privacy in the presence of semi-honest adversary. Since Σ_d has $d \geq t$-privacy, the adversary learns nothing from any t shares of $[\boldsymbol{s}_1], \ldots, [\boldsymbol{s}_n]$. If we fix $[\boldsymbol{s}_i]$ for $i \in H$ and $[\boldsymbol{r}_i]$ for $i \in [2t]$, the fact that M is an $(n, n, (1+\epsilon)n, (1-\epsilon)n)$-ramp HIM and $|\bar{H}| + 2t \leq 3t \leq (1-\epsilon)n$ implies that there is a surjection from $\boldsymbol{r}_i, i \in [2t+1, n]$ to $\boldsymbol{s}_i, i \in \bar{H}$. Since $\boldsymbol{r}_i, i \in [2t+1, n]$ provided by the input client are distributed uniformly at random, $\boldsymbol{s}_i, i \in \bar{H}$ are also distributed uniformly.

Protocol CheckConsistency(d)

Setup: n parties $P_1, \cdots, P_n$ and an input client I
Public input: an $(n, n, (1+\epsilon)n, (1-\epsilon)n)$-ramp HIM M
Private input: P_i obtains corresponding shares of $[\boldsymbol{r}_1]_d, \cdots, [\boldsymbol{r}_{2t}]_d$

- Input client randomly generates $[\boldsymbol{r}_{2t+1}]_d, \cdots, [\boldsymbol{r}_n]_d$ and distributes corresponding shares to $P_1, \cdots, P_n$
- Parties locally compute $([\boldsymbol{s}_1]_d, \cdots, [\boldsymbol{s}_n]_d)^T = M([\boldsymbol{r}_1]_d, \cdots, [\boldsymbol{r}_n]_d)^T$
- Party P_i receives all shares of $[\boldsymbol{s}_i]$ from other parties and checks the consistency. If the sharing is incorrect, P_i complains and the protocol aborts.
- If no party complains, the parties conclude that $[\boldsymbol{r}_1]_d, \cdots, [\boldsymbol{r}_{2t}]_d$ are consistent.

Remark 4 *Protocol **CheckConsistency**(d) is similar to Protocol **RandEl**(d) as ramp HIM is used to guarantee d-consistency. The major difference is input and output. In MPC protocol, preprocessing data come from secret sharings generated by each party including both honest parties and corrupted parties while in MPCitH protocol, preprocessing data are directly provided by the prover.*

5.2 Constant-rate zero-knowledge proof

The MPC protocol in [5] is perfectly secure against malicious adversary. This MPC protocol has perfect robustness and the MPCitH protocol relying on it thus saves two rounds of communication [32].

The communication cost of the prover consists of commitment and decommitment. According to [32], we need a statistically-binding commitment scheme, whose output length grows linearly in message length. The communication cost of commitment is $O(n|C|\log q)$ bits. The decommitment requires the prover to reveal the views of t parties selected by the verifier which includes witness, preprocessing data and broadcast value, which takes $O(t|C|\log q)$ bits communication. In summary, the communication complexity of MPCitH protocol is $O((n+t)|C|\log q)$ bits, which is equivalent to $O(n|C|\log n)$ as $t = \Omega(n)$ and HIM forces $q \geq 2n$ [8].

We briefly describe how to reduce communication complexity to $O(|C|)$ with the help of ramp HIM. Firstly, we apply a packed secret sharing which batches $\Omega(n)$ evaluations together to remove the multiplicative factor n in the communication complexity. The next step is to apply an MPC protocol over constant-size field in Section 4. By replacing **RandEl**(d) with **CheckConsistency**(d), we obtain an MPC protocol Π_f that has t-privacy and perfect t-robustness due to Proposition 6. Plugging this MPC protocol in [8], finally we obtain a 3-round constant-rate zero-knowledge proof. In contrast, the constant-rate zero-knowledge proof proposed in [32] has 5 communication rounds since it relies on a MPC protocol[16] with statistical robustness and coin tossing between prover and verifier causes more interaction.

Theorem 9. *Given a statistically binding commitment scheme, for any NP relation $R(x,w)$ which can be verified with a circuit with $O(|C|)$ gates, there exists a two-party* **3-round**[11] *constant-rate zero-knowledge proof in the random oracle model. The protocol has communication complexity $O(|C|)$ and soundness error $2^{-\Omega(n)}$.*

6 Information-theoretic multi-verifier zero-knowledge proof

In MVZK, the prover $\mathcal{P}$ wants to convince n verifiers $\mathcal{V}_1, \cdots, \mathcal{V}_n$ that regarding to a NP relation R, it holds a witness w for a statement such that $R(x,w) = 1$. In this paper, we focus on a special NP relation:circuit satisfiability, which aims to find a witness $w \in \mathbb{F}_q$ for a circuit C such that $C(w) = 1$. We assume that at most t verifiers are corrupted by the adversary and can collude with the prover. There are two types of communications, the communications between the verifiers and prover and the communications between different verifiers. In [43], they present an efficient MVZK in the presence of honest-majority verifiers.

[11] If we consider random oracle model, then statistically binding commitment scheme needs only one round [37]. We emphasize that regardless the model, our new MPCitH protocol saves two rounds of communication compared to [32].

In the information-theoretic MVZK, the verifiers invoke a coin-tossing functionality $\mathcal{F}_{coin}$ to jointly sample an random element in the challenge set. In this process, HIM plays a central role in producing random sharings. However, due to circuit size and the security parameter, the share size of MVZK has to be large enough. There is another challenge related to the share size which is the verification technique [43]. We briefly introduce this technique and show how to adapt it to our constant-size field later.

1. For a circuit with $|C|$ multiplication gates, the prover distributes corresponding share of $([x_i], [y_i], [z_i])_{i\in[|C|]}$ to n verifiers, which needs communication of $O(n|C|)$ field elements in $\mathbb{F}_q$.
2. All verifiers jointly sample a uniform challenge $\chi \in \mathbb{F}_{q^r}$ and compute the inner-product tuple:
$$[\mathbf{x}] = ([x_1], \chi \cdot [x_2], \ldots, \chi^{|C|-1} \cdot [x_{|C|}])$$
$$[\mathbf{y}] = ([y_1], [y_2], \ldots, [y_{|C|}])$$
$$[z] = \sum_{i=1}^{|C|} \chi^{i-1} \cdot [z_i]$$
Note that inner product tuples are Shamir SSS defined over $\mathbb{F}_{q^r}$. In this step, joint sampling communicates $O(n^2)$ field elements in $\mathbb{F}_{q^r}$.
3. All verifiers apply the inner-product checking method in [28, 29]. As [28] has analyzed, the verification procedure incurs communication of $O((n\tau + n^2)\log_\tau |C|)$ field elements in $\mathbb{F}_{q^r}$(Here τ is compress parameter).

We notice that the soundness error is dependent on $\mathbb{F}_{q^r}$. If there exists one incorrect multiplication triple, then the inner-product tuple passes the verification of inner product with probability at most $\frac{|C|-1}{q^r}$. To achieve a soundness error of $2^{-\lambda}$, we have to set $q^r = 2^\lambda(|C| - 1)$, which incurs a communication overhead $O(\lambda + \log|C|)$ if our computation is carried out over $\mathbb{F}_q$ instead of $\mathbb{F}_{q^r}$.

It is clear that moving from $\mathbb{F}_q$ to $\mathbb{F}_{q^r}$ increases the communication complexity in Step 2 and Step 3. We can save the communication complexity in Step 2 by introducing ramp secret sharing and ramp HIM. The communication complexity in step 3 can be saved by "batched checking".

Functionality $\mathcal{F}_{coin}$

This functionality runs for n verifiers and an adversary $\mathcal{A}$ as follows:

- Upon receiving $(coin, \mathcal{C})$ from all verifiers where $\mathcal{C}$ is the challenge set, sample $r \leftarrow \mathcal{C}$ and sends $(random, r)$ to $\mathcal{A}$.
- If $\mathcal{A}$ returns the message $(deliver)$, then sends $(random, r)$ to all verifiers. Otherwise $\mathcal{A}$ returns the message $(abort)$, then outputs abort for all verifiers.

To begin with, we instantiate $\mathcal{F}_{coin}$ over constant-size field. The building blocks of our protocol are ramp secret sharing scheme and ramp HIM, i.e., we use

the ramp secret sharing scheme over constant-size field with $\frac{1-\epsilon}{2}n$-privacy and $\frac{1}{2}n$-reconstruction. This ramp SSS requires that the number of corrupted verifiers is sub-optimal $t = \frac{(1-\epsilon)n}{2}$ and its secret space is $\mathbb{F}_q^{\Omega(\epsilon n)}$ with $\epsilon = O(\frac{\lambda \log |C|}{n})$.

Protocol Rand

Public input: an $(n, t, a, (1-\epsilon)n)$-ramp HIM M(a can be any number since **Rand** only uses $(1-\epsilon)n$-privacy instead of $(1+\epsilon)n$-reconstruction of ramp HIM).

- Each verifier $\mathcal{V}_i$ samples a random sharing $[\boldsymbol{s}_i]_t$ and distributes corresponding share to other verifiers.
- All verifiers locally compute $([\boldsymbol{r}_1]_t, \ldots, [\boldsymbol{r}_t]_t)^T = M([\boldsymbol{s}_1]_t, \cdots, [\boldsymbol{s}_n]_t)^T$.

Protocol Coin

Setup: The protocol **Rand** generates t random sharings at one time. The protocol **Coin** picks a random sharing $[\mathbf{r}]_t$ from these t sharings.

- Each verifier sends his share of $[\mathbf{r}]_t$ to all other verifiers. After receiving shares from other $n-1$ verifiers, each verifier checks whether $[\boldsymbol{r}]_t$ is a valid secret sharing.
- If each verifier $\mathcal{V}_i$ concludes that all shares are correct, the secret $\boldsymbol{r}$ are reconstructed and outputted. Otherwise, $\mathcal{V}_i$ broadcasts the message (*abort*) and the protocol aborts.

Similar to [43], we obtain the following result.

Theorem 10. *The protocol **Coin** realizes $\mathcal{F}_{coin}$ in security-with-abort model in the presence of a malicious adversary corrupting $t = \frac{1-\epsilon}{2}n$ verifiers.*

Proof. The only difference from [43] is that our **Rand** protocol uses ramp HIM instead of HIM. It suffices to prove that the randomness produced by **Rand** protocol is independent of the adversary. Since M is an $(n, t, a, (1-\epsilon)n)$-ramp HIM and $2t = (1-\epsilon)n$, there is a surjection from $(\mathbf{s}_i)_{i\in H}$ to $(\mathbf{r}_i)_{i\in[t]}$ where H is the set of honest parties. This means $(\mathbf{r}_i)_{i\in[t]}$ distributes uniformly at random conditioning on the sharings of the corrupted parties $(\mathbf{s}_i)_{i\in[n]/H}$.

We proceed to the batch checking. Assume that $[x]$ is a sharing over $\mathbb{F}_q$ and $\chi \in \mathbb{F}_{q^r}$. We redefine the inner product in this setting. We note that one can represent $\mathbb{F}_{q^r}$ as a linear subspace over $\mathbb{F}_q$. Let $v_1, \ldots, v_r$ be the basis of $\mathbb{F}_{q^r}$ over $\mathbb{F}_q$. Then, for any element $\lambda \in \mathbb{F}_{q^r}$, we have $\lambda v_i = \sum_{j=1}^{r} m_{ij} v_j$. Thus, λ can be thought of as a linear map $M_\lambda = (m_{ij})_{r\times r}$ from $\mathbb{F}_q^r$ to its self. In this sense, we redefine $\cdot$ as

$$\lambda \cdot (x_1, \ldots, x_r) = M_\lambda (x_1, \ldots, x_r)$$

where $x_1, \ldots, x_r \in \mathbb{F}_q$. We now show how to check circuit satisfiability.

1. For a circuit with $|C|$ multiplication gates, the prover distributes corresponding share of $([x_i], [y_i], [z_i])_{i\in[|C|]}$ over $\mathbb{F}_q$ to n verifiers. Pad $([0], [0], [0])$'s to these $|C|$ sharings so that the number of sharings is divisible by r. We now assume that $|C|$ is divisible by r.
2. All verifiers jointly sample a uniform challenge $\chi \in \mathbb{F}_{q^r}$ and compute the inner-product tuple:

$$[\mathbf{x}] = (([x_1], \ldots, [x_r]), \chi \cdot ([x_{r+1}], \ldots, [x_{2r}]), \ldots, \chi^{|C|/r-1} \cdot [x_{|C|-r+1}], \ldots, [x_{|C|}])$$
$$[\mathbf{y}] = (([y_1], \ldots, [y_r]), ([y_{r+1}], \ldots, [y_{2r}]), \ldots, ([y_{|C|-r+1}], \ldots, [y_{|C|}]))$$
$$[z] = \sum_{i=1}^{|C|/r} \chi^{i-1} \cdot ([z_{(i-1)r+1}], \ldots, [z_{ir}])$$

3. All verifiers apply the inner-product checking method in [28, 29].

Therefore, the amortized share size is now independent of the security parameter and circuit size.

Acknowledgments

The work of Chaoping Xing was supported in part by the National Key Research and Development Project under Grant 2022YFA1004900, in part by the National Natural Science Foundation of China under Grant 12031011. The work of Chen Yuan was supported in part by the National Natural Science Foundation of China under Grant 12101403.

References

1. Masayuki Abe, Ronald Cramer, and Serge Fehr. Non-interactive distributed-verifier proofs and proving relations among commitments. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 206–224. Springer, 2002.
2. Benny Applebaum, Eliran Kachlon, and Arpita Patra. Verifiable relation sharing and multi-verifier zero-knowledge in two rounds: trading nizks with honest majority. In *Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part IV*, pages 33–56. Springer, 2022.
3. Carsten Baum, Robin Jadoul, Emmanuela Orsini, Peter Scholl, and Nigel P Smart. Feta: Efficient threshold designated-verifier zero-knowledge proofs. *Cryptology ePrint Archive*, 2022.
4. Carsten Baum and Ariel Nof. Concretely-efficient zero-knowledge arguments for arithmetic circuits and their application to lattice-based cryptography. In *Public-Key Cryptography–PKC 2020: 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4–7, 2020, Proceedings, Part I*, pages 495–526. Springer, 2020.

5. Zuzana Beerliová-Trubíniová and Martin Hirt. Perfectly-secure MPC with linear communication complexity. In Ran Canetti, editor, *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 213–230. Springer, 2008.
6. Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 1–10. ACM, 1988.
7. Eli Ben-Sasson, Serge Fehr, and Rafail Ostrovsky. Near-linear unconditionally-secure multiparty computation with a dishonest minority. *IACR Cryptol. ePrint Arch.*, page 629, 2011.
8. Rikke Bendlin and Ivan Damgård. Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In *Theory of Cryptography Conference*, pages 201–218. Springer, 2010.
9. V. E. Beneš. Optimal rearrangeable multistage connecting networks. *The Bell System Technical Journal*, 43(4):1641–1656, 1964.
10. Mike Burmester and Yvo Desmedt. Broadcast interactive proofs. In *Advances in Cryptology—EUROCRYPT'91: Workshop on the Theory and Application of Cryptographic Techniques Brighton, UK, April 8–11, 1991 Proceedings*, pages 81–95. Springer, 2001.
11. Ignacio Cascudo, Ronald Cramer, Chaoping Xing, and Chen Yuan. Amortized complexity of information-theoretically secure mpc revisited. In *Annual International Cryptology Conference*, pages 395–426. Springer, 2018.
12. Hao Chen and Ronald Cramer. Algebraic geometric secret sharing schemes and secure multi-party computations over small fields. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 521–536. Springer, 2006.
13. Hao Chen, Ronald Cramer, Shafi Goldwasser, Robbert de Haan, and Vinod Vaikuntanathan. Secure computation from random error correcting codes. In Moni Naor, editor, *Advances in Cryptology - EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007, Proceedings*, volume 4515 of *Lecture Notes in Computer Science*, pages 291–310. Springer, 2007.
14. Koji Chida, Daniel Genkin, Koki Hamada, Dai Ikarashi, Ryo Kikuchi, Yehuda Lindell, and Ariel Nof. Fast large-scale honest-majority MPC for malicious adversaries. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 34–64. Springer, 2018.
15. Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.
16. Ivan Damgård and Yuval Ishai. Scalable secure multiparty computation. In *Advances in Cryptology-CRYPTO 2006: 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006. Proceedings 26*, pages 501–520. Springer, 2006.
17. Ivan Damgård, Yuval Ishai, and Mikkel Krøigaard. Perfectly secure multiparty computation and the computational overhead of cryptography. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International*

Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings, volume 6110 of *Lecture Notes in Computer Science*, pages 445–465. Springer, 2010.
18. Ivan Damgård and Jesper Buus Nielsen. Scalable and unconditionally secure multiparty computation. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, volume 4622 of *Lecture Notes in Computer Science*, pages 572–590. Springer, 2007.
19. Cyprien Delpech de Saint Guilhem, Emmanuela Orsini, and Titouan Tanguy. Limbo: efficient zero-knowledge mpcith-based arguments. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3022–3036, 2021.
20. Daniel Escudero, Vipul Goyal, Antigoni Polychroniadou, and Yifan Song. Turbopack: Honest majority MPC with constant online communication. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 951–964. ACM, 2022.
21. Matthew K. Franklin and Moti Yung. Communication complexity of secure computation (extended abstract). In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 699–710. ACM, 1992.
22. Arnaldo Garcia and Henning Stichtenoth. A tower of Artin - Schreier extensions of function fields attaining the Drinfeld - Vlâdut bound. *Inventiones Mathematicae*, 121:211–222, 01 1995.
23. Arnaldo Garcia and Henning Stichtenoth. On the asymptotic behaviour of some towers of function fields over finite fields. *Journal of Number Theory*, 61:248–273, 12 1996.
24. Daniel Genkin, Yuval Ishai, and Antigoni Polychroniadou. Efficient multi-party computation: From passive to active security via secure SIMD circuits. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 721–741. Springer, 2015.
25. Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. Zkboo: Faster zero-knowledge for boolean circuits. In *USENIX Security Symposium*, volume 16, 2016.
26. Vipul Goyal, Hanjun Li, Rafail Ostrovsky, Antigoni Polychroniadou, and Yifan Song. ATLAS: efficient and scalable MPC in the honest majority setting. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part II*, volume 12826 of *Lecture Notes in Computer Science*, pages 244–274. Springer, 2021.
27. Vipul Goyal, Yanyi Liu, and Yifan Song. Communication-efficient unconditional MPC with guaranteed output delivery. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 85–114. Springer, 2019.
28. Vipul Goyal and Yifan Song. Malicious security comes free in honest-majority mpc. *Cryptology ePrint Archive*, 2020.

29. Vipul Goyal, Yifan Song, and Chenzhi Zhu. Guaranteed output delivery comes free in honest majority mpc. In *Advances in Cryptology–CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part II*, pages 618–646. Springer, 2020.
30. Jens Groth and Rafail Ostrovsky. Cryptography in the multi-string model. *Journal of cryptology*, 27(3):506–543, 2014.
31. Martin Hirt, Ueli M. Maurer, and Bartosz Przydatek. Efficient secure multi-party computation. In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, pages 143–161. Springer, 2000.
32. Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 21–30, 2007.
33. Daniel Kales and Greg Zaverucha. Efficient lifting for shorter zero-knowledge proofs and post-quantum signatures. *Cryptology ePrint Archive*, 2022.
34. Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 525–537, 2018.
35. Harald Niederreiter and Chaoping Xing. *Rational Points on Curves over Finite Fields–Theory and Applications*. Cambridge University Press, 2001.
36. Peter Sebastian Nordholt and Meilof Veeningen. Minimising communication in honest-majority MPC by batchwise multiplication verification. In Bart Preneel and Frederik Vercauteren, editors, *Applied Cryptography and Network Security - 16th International Conference, ACNS 2018, Leuven, Belgium, July 2-4, 2018, Proceedings*, volume 10892 of *Lecture Notes in Computer Science*, pages 321–339. Springer, 2018.
37. Rafael Pass. On deniability in the common reference string and random oracle model. In *Advances in Cryptology-CRYPTO 2003: 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003. Proceedings 23*, pages 316–337. Springer, 2003.
38. Ignacio Cascudo Pueyo, Hao Chen, Ronald Cramer, and Chaoping Xing. Asymptotically good ideal linear secret sharing with strong multiplication over *Any* fixed finite field. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 466–486. Springer, 2009.
39. Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.
40. Kenneth Shum, Ilia Aleshnikov, P. Vijay Kumar, Henning Stichtenoth, and Vinay Deolalikar. A low-complexity algorithm for the construction of algebraic-geometric codes better than the Gilbert-Varshamov bound. *IEEE Transactions on Information Theory*, 47(6):2225–2241, 2001.
41. H. Stichtenoth. *Function Fields and Codes*. Springer, 2003.
42. M. A. Tsfasman and S. G. Vlăduţ. *Algebraic-Geometric Codes*. Springer, 1991.
43. Kang Yang and Xiao Wang. Non-interactive zero-knowledge proofs to multiple verifiers. *Cryptology ePrint Archive*, 2022.

A Player Elimination

Player elimination was first proposed in [31] to transform a non-robust (but detectable) protocol into a robust protocol at essentially no additional costs. This protocol cuts the preprocessing phase into many segments. At the beginning of each segment, all parties are happy. If some party detects the inconsistency, he becomes unhappy in this segment. At the end of this segment, if there is some party unhappy, the protocol enters into fault localization and removes a pair of parties from the rest of the computation. Then, the player elimination protocol repeats this segment. For completeness, we present the player elimination protocol in [5].

Player Eliminiation

Setup: a set of n' parties $\mathcal{I} = \{P_1, \ldots, P_{n'}\}$ and at most $t' \leq \frac{n'(1-\epsilon)}{3}$ of them are corrupted. Divide the computation into several segment and do the following in each segment.

Initialization: All parties set their happy-bit happy.

Fault Detection: Reach agreement whether or not at least one party is unhappy.

Fault Localization: Find a pair of parties E in $\mathcal{I}$ that contain at least one corrupted party.

- Denote the player $P_r \in \mathcal{I}$ with the smallest index r as the referee.
- Every $P_i \in \mathcal{I}$ sends everything he received and all random values he chose during the computation of the actual segment (including fault detection) to P_r.
- Given the value received above, P_r can reproduce all message that should be sent and compare it with the value from the recipient that claims to have. Then, P_r broadcasts (ℓ, i, j, x, x') where ℓ is the index of the message, x is the message sent by P_i and x' is the message received by P_j with $x \neq x'$.
- The accused parties P_i, P_j broadcast whether they agree with P_r. If P_i disagrees, set $E = \{P_r, P_i\}$. If P_j disagrees, set $E = \{P_r, P_j\}$. Otherwise set $E = \{P_i, P_j\}$.

Player Elimination: Set $\mathcal{I} = I \setminus E, n' = n' - 2, t' = t' - 1$ and repeat this segment.

B Construction of binary ramp HIM

Binary ramp HIMs have particular interest due to applications. In order to construct good binary ramp HIMs, we require binary codes with both large distance and dual distance. However, the construction based on algebraic geometry codes does not work well for binary ramp HIMs. This is because there are fewer points compared with genus for algebraic curves over the binary fields. In this subsection, we briefly report the result of explicit binary ramp HIMs based on the trivial concatenation of algebraic geometry codes. We also present an existence

result of binary ramp HIMs from the Gilbert-Varshomov bound. It is surprisingly noted that there is no big difference between these two results.

By employing the algebraic geometry codes from the Garcia-Stichtenoth tower and trivial binary codes, we obtain the following explicit construction on binary HIMs.

Theorem 11. *Let t be an even positive integer. Let $\delta \in \left(\frac{1}{2^{t/2}-2}, 1\right]$ be a real. Then there is a family of binary $(n, m; r, p)_2$-ramp HIM with $n \to \infty$, $\frac{m}{n} \to \delta$ and*

$$\limsup_{n\to\infty} \frac{r}{n} \leq 1 + \frac{t-1}{t} \times \delta + \frac{1+\delta}{t(2^{t/2}-1)}, \quad \liminf_{n\to\infty} \frac{p}{n} \geq \frac{1}{t} + \frac{1-\delta}{t(2^{t/2}-1)}.$$

In particular, there is a family of $(n, n; r, p)_2$-ramp HIM with $n \to \infty$ and

$$\limsup_{n\to\infty} \frac{r}{n} \leq 1 + \frac{t-1}{t} + \frac{2}{t(2^{t/2}-1)}, \quad \liminf_{n\to\infty} \frac{p}{n} \geq \frac{1}{t} - \frac{2}{t(2^{t/2}-1)}.$$

Furthermore, this family can be constructed in time $O(n^3)$.

The random linear code can approach Gilbert-Varshamov bound and so does its dual code.

Lemma 2. *With probability at least $1 - 2^{-\epsilon\ell}$, a random binary linear code and its dual code both achieve the Gilbert-Varshamov bound. Precisely speaking, given a random binary $[\ell, k, d]$-linear code C and its dual $C^\perp = [\ell, \ell - k, d^\perp]$, with high probability one has*

$$k = \ell\left(1 - H_2\left(\frac{d}{\ell}\right) - \epsilon\right), \quad \ell - k = \ell\left(1 - H_2\left(\frac{d^\perp}{\ell}\right) - \epsilon\right). \tag{6}$$

By Lemma 2, we have the following existence result.

Theorem 12. *Let $\delta \in (0, 1]$ be a real. Let $\tau, \tau^\perp$ be reals in $[0, 1/2]$ satisfying the equation*

$$(\delta + 1)(1 - H_2(\tau)) = 1, \quad (\delta^{-1} + 1)(1 - H_2(\tau^\perp)) = 1.$$

Then there is a family of binary $(n, m; r, p)_2$-ramp HIM with $n \to \infty$, $\frac{m}{n} \to \delta$ and

$$\lim_{n\to\infty} \frac{r}{n} \leq (1+\delta)(1-\tau), \quad \lim_{n\to\infty} \frac{p}{n} \geq (1+\delta)\tau^\perp.$$

In particular, there is a family of binary $(n, n; r, p)_2$-ramp HIM with $n \to \infty$ and

$$\lim_{n\to\infty} \frac{r}{n} \leq 2(1-\tau), \quad \lim_{n\to\infty} \frac{p}{n} \geq 2\tau^\perp.$$