Constant-Round Multi-Party Private Set Union Using Reversed Laurent Series

Jae Hong Seo[1], Jung Hee Cheon[2⋆], and Jonathan Katz[3⋆⋆]

[1] National Institute of Information and Communications Technology
Tokyo, Japan
jaehong@nict.go.jp
[2] ISaC & Dept. of Mathematical Sciences
Seoul National University
Seoul, Korea
jhcheon@snu.ac.kr
[3] Dept. of Computer Science
University of Maryland
Maryland, USA
jkatz@cs.umd.edu

Abstract. We introduce the idea of associating a set of elements with a *rational function* represented using a *reversed Laurent series*. Using this representation, we propose private set-union protocols in the multi-party setting, assuming an honest majority. Our protocols are the first efficient protocol for private set union with constant round complexity (in both the semi-honest and malicious settings), as well as the first with statistical security (in the semi-honest setting).

1 Introduction

We focus here on constructing protocols for *privacy-preserving set operations*. In this setting, we have a set of parties $\mathcal{P}_1, \ldots, \mathcal{P}_n$ with each party $\mathcal{P}_i$ holding a set $S_i \subseteq \mathcal{U}$ of elements in some known universe $\mathcal{U}$; the parties want to compute some function of their sets such as their intersection $\bigcap_i S_i$ or union $\bigcup_i S_i$. Of course, the problem can be solved using protocols for generic secure multi-party computation [13, 3], but we are interested in more efficient solutions. This problem, for various types of set operations, has received a lot of attention in both the two-party [10, 4, 14, 7, 6, 17, 9, 15, 8] and multi-party [19, 11, 23] settings.

In this paper, we propose a new framework for privacy-preserving set operations based on representing sets using *rational polynomial functions* and manipulating this representation using *reversed Laurent series*. (See the following section for an overview.) Although our framework can be extended to apply to a more general class of set operations, we focus here on computing *set union*

⋆ Supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012-0001243)
⋆⋆ Supported by NSF grant #1111599.

Semi-honest:

Ref.	Rounds	Communication	Computation	Assumptions	Threshold
[19]	$O(n)$	$O(n^2\, k\, \tau_N)$	$O(n^3 k^2 \tau_N \rho_N)$	DCR	$t < n$
[11]	$O(n)$	$O(n^2\, k\, \tau_N)$	$O(nk^2 \tau_N \rho_N)$	DCR	$t < n$
Here	$O(1)$	$O(n^3 k^2 \tau_p)$	$O((n^4 k^2 + n^2 k^2 \tau_p)\rho_p)$	none	$t < n/2$

Malicious:

Ref.	Rounds	Communication	Computation	Assumptions	Threshold
[11]	$O(n)$	$O((n^3\, k\, + n^2\, k^2)\, \tau_N)$	$O(nk^2 \tau_N \rho_N)$	DCR	$t < n$
Here	$O(1)$	$O(n^3 k^2 \tau'_p)$	$O(n^4 k^2 \tau'_p \rho'_p)$	DL	$t < n/2$

Table 1. Privacy-preserving set-union protocols. DCR denotes the decisional composite-residuosity assumption [21], and DL denotes the discrete-logarithm assumption. The number of parties is n, the maximum set size is k, the number of corrupted parties is t, and τ_N, ρ_N (resp., τ_p, ρ_p) are the size and multiplication cost for a modulus N (resp. prime p) used for Paillier encryption (resp., representing domain elements). τ'_p and ρ'_p are the size and multiplication cost for a cyclic group of order p used for Pedersen commitment scheme and Gennaro-Rabin-Rabin verifiable secret sharing scheme.

in the multi-party setting. Set union is not trivial to compute securely, and in particular the solution in which each party publicly reveals its set is not secure since it reveals which parties hold which elements, as well as the *multiplicity* of each element in the union.

Our framework yields efficient multi-party protocols for private set union that are secure against any dishonest minority, and in particular we obtain the first efficient multi-party protocols for set union (in both the semi-honest and malicious settings) that use a *constant* number of rounds. Moreover, our protocol achieves *statistical* security in the semi-honest (aka, honest-but-curious) setting. In contrast, previous protocols [19, 11] have round complexity linear in the number of parties, and achieve computational security even in the semi-honest setting. On the other hand, previous protocols tolerate any number of corrupted users. We compare our work to prior work in Table 1.

Beyond the result just stated, we believe our techniques are of independent interest as they provide what is, to the best of our knowledge, a novel approach to privacy-preserving computation on sets. We explain our approach in more detail in the following section.

1.1 Overview of Our Techniques

As in some prior work (e.g., [10, 19, 11, 6, 7]), we begin with the observation that a set S can be represented by a polynomial $f_S(x)$ over a field $\mathbb{F} \supseteq S$ such that the roots of $f_S(x)$ are exactly the elements of S; namely,

$$f_S(x) = \prod_{s \in S} (x - s).$$

In contrast to previous work, however, we then switch to viewing S as being represented by the *rational* polynomial $1/f_S(x)$. This representation is well suited

for computing set union, since

$$\frac{1}{f_S(x)} + \frac{1}{f_{S'}(x)} = \frac{f_S(x) + f_{S'}(x)}{f_S(x) \cdot f_{S'}(x)} = \frac{\gcd\left(f_S(x), f_{S'}(x)\right) \cdot p(x)}{f_S(x) \cdot f_{S'}(x)}$$
$$= \frac{p(x)}{\operatorname{lcm}\left(f_S(x), f_{S'}(x)\right)}, \tag{1}$$

for some polynomial $p(x)$. That is, the denominator of (the reduced representation of) the rational polynomial $\frac{1}{f_S(x)} + \frac{1}{f_{S'}(x)}$ is a polynomial $f_{S \cup S'}(x) \stackrel{\text{def}}{=} \operatorname{lcm}(f_S(x), f_{S'}(x))$ with no repeated roots, whose roots are exactly the elements of $S \cup S'$. Because of how it is defined (in particular, the fact that it has no repeated roots), the polynomial $f_{S \cup S'}(x)$ reveals nothing beyond $S \cup S'$ and therefore provides a starting point for secure computation of the union.

The above does not yet give a secure protocol for computing the union, as we must still address several challenges. First, we need an efficient way to manipulate rational polynomials. For this, we rely on the *reversed Laurent series* representation of rational functions [25, Section 16.8]; see Section 2 for details. Second, we need to deal with the fact that the numerator in (1) might reveal information beyond the union $S \cup S'$. We thus modify the above, having the parties choose random polynomials $r(x), r'(x)$ of degree at most $|S| - 1$ and $|S'| - 1$, respectively, and then compute

$$\frac{r(x)}{f_S(x)} + \frac{r'(x)}{f_{S'}(x)} = \frac{f_S(x)r'(x) + f_{S'}(x)r(x)}{f_S(x) \cdot f_{S'}(x)} = \frac{\gcd\left(f_S(x), f_{S'}(x)\right) \cdot u(x)}{f_S(x) \cdot f_{S'}(x)}$$
$$= \frac{u(x)}{\operatorname{lcm}\left(f_S(x), f_{S'}(x)\right)}.$$

We prove that $u(x)$, above, is a *uniformly distributed* polynomial of degree at most $\deg(\operatorname{lcm}(f_S(x), f_{S'}(x))) - 1$. Thus, assuming $|\mathbb{F}| \gg |S|$, it holds with overwhelming probability that $u(x)$ and $f_{S \cup S'}(x)$ have no roots in common and so recovering the denominator of the above still yields the correct result. Moreover, uniformity of $u(x)$ implies that computing the above leaks no information about either party's original set.

Although we describe the two-party case above for simplicity, we can easily extend the above argument to the case $n > 2$ in which we are mostly interested. See Section 3.1 for details.

1.2 Related Work

Private set-union protocols should hide both (1) which parties hold which elements, and (2) the multiplicity of each element in the union. There are only a few multi-party protocols satisfying these two requirements. Kissner and Song [19] proposed a protocol which can be utilized for multi-party set union in the semi-honest setting. Frikken [11] proposed a privacy-preserving set-union protocol in the malicious setting. Both protocols rely on a "mix-net" approach, where $t + 1$

parties mix encrypted elements (when security against t corruptions is required). This approach inherently requires round complexity $O(t)$.

Some protocols achieving relaxed privacy guarantees have been proposed. In particular, Kissner and Song [19], Sang and Shen [23], and Hong et al. [16] proposed multi-party set-union protocols that leak the multiplicity of each element in the union.

In the two-party case, other protocols are known. Brickell and Shmatikov [4] proposed two-party set-union protocols secure against *honest-but-curious* adversaries. Recently, Hazay and Nissim [15] proposed very efficient protocols for privacy-preserving set union secure against malicious adversaries; their protocol achieves (almost) linear complexity in the number of private inputs. Neither of these protocols appear to generalize easily to the multi-party case.

1.3 Outline of the Paper

In the next section, we recall the notion of the *reversed Laurent series* (RLS) representation of a rational function, and discuss efficient conversions between a rational function and its RLS representation. In Section 3, we show how to use the RLS representation of rational functions to perform set union. As applications of our technique, we give constant-round protocols for computing set union in both the semi-honest and malicious settings.

2 Reversed Laurent Series

We let $\mathbb{Z}_p$ denote the set of integers modulo p. In this paper, we always take p prime so that $\mathbb{Z}_p$ is also the finite field of size p. As usual, $\mathbb{Z}_p[x]$ denotes the set of polynomials over $\mathbb{Z}_p$. We use $[a, b]$ (with $a \leq b$ and both possibly negative) to denote the set of integers between a and b, inclusive.

2.1 Reversed Laurent Series and Rational Functions

A *reversed Laurent series* (RLS) over $\mathbb{Z}_p$ is a singly infinite, formal sum of the form

$$f(x) = \sum_{i=-\infty}^{m} a_i \, x^i \quad (a_m \neq 0),$$

for m an integer and $a_i \in \mathbb{Z}_p$. We refer to m as the *degree* of f, denoted $\deg(f)$. Given $d_1 \leq d_2 \leq m$, we define

$$f(x)_{[d_1,d_2]} = \sum_{i=d_1}^{d_2} a_i \, x^i.$$

The set of all reversed Laurent series, denoted $\mathbb{Z}_p((x^{-1}))$, forms a field with addition and multiplication defined in the natural way. Since $\mathbb{Z}_p[x]$ is a subring of $\mathbb{Z}_p((x^{-1}))$, any rational function f/g with $f, g \in \mathbb{Z}_p[x]$ and $g \neq 0$ can be

expressed as a reversed Laurent series and we refer to this as the *RLS representation* of the rational function f/g. Note that the RLS representation for a given rational function is unique. That is, if $f/g = f'/g'$ then the RLS representations of f/g and f'/g' are identical.

2.2 Conversion from a Rational Function to its RLS

Let $f, g \in \mathbb{Z}_p[x]$, and assume $\deg(f) < \deg(g) \leq \ell$. (The case $\deg(f) \geq \deg(g)$ can be reduced to this case by first performing polynomial division with remainder.) One can compute $k > \deg(g)$ high-order terms of the RLS representation of f/g using the following algorithm:

RationalToRLS(f, g, k):

(1) Compute $F(x) = f(x) \cdot x^k$.
(2) Use polynomial division to compute $Q(x)$ and $R(x)$ with $F(x) = g(x) \cdot Q(x) + R(x)$ and $\deg(R) < \deg(g)$.
(3) Output $Q(x) \cdot x^{-k}$.

Since $F/g = Q + R/g$ and $\deg(R) < \deg(g)$, we have $Q = (F/g)_{[0,k+\deg(f)-\deg(g)]}$. Since $F/g = x^k \cdot f/g$ and we assumed $\deg(f) < \deg(g)$, we see that the output consists of exactly the k high-order terms of the RLS of f/g; that is, $Q(x) \cdot x^{-k} = (f/g)_{[-k,\deg(f)-\deg(g)]} = (f/g)_{[-k,-1]}$.

The computational cost of the above algorithm is essentially just the complexity of polynomial division.

2.3 Conversion from an RLS Representation to a Rational Function

The RLS representation of a rational function f/g will, in general, have infinitely many terms. However, all "information" about f/g is contained in a finite number of high-order terms. Specifically, let $f, g \in \mathbb{Z}_p[x]$ with $\deg(f) < \deg(g) \leq \ell$ and $g \neq 0$. Then the rational function f/g is determined by the 2ℓ high-order terms of its RLS representation. Moreover, there is an efficient algorithm to recover f/g (in reduced terms) given these 2ℓ high-order terms and the bound ℓ on the degree of g. See [25, Section 17.5.1] for details.

3 Privacy-Preserving Set Union

We begin with an overview of our approach to computing set union, followed by formal descriptions of protocols in the semi-honest and malicious settings. We consider n parties, each of whom holds a set over some universe $\mathcal{U} \subset \mathbb{Z}_p$ where $p > n$ is known and $\mathcal{U}$ is a negligible fraction of $\mathbb{Z}_p$. (This can be easily obtained by padding every element in the original universe with sufficiently many 0s.) We further assume the size k_i of each party's set is known. (In fact, for simplicity

here we assume that $k_i = k$ for all i. A treatment of the general case will be found in the full version.) By having parties pad out their sets to some maximum size using random elements, this can be relaxed to requiring only that $\sum_i k_i$ is known; we omit the details.

3.1 Representing Sets and Computing their Union

Given a set $S \subseteq \mathcal{U}$ of size $|S| = d$, we define the polynomial

$$f_S(x) \stackrel{\text{def}}{=} \prod_{s \in S} (x - s).$$

Note that $\deg(f_S) = |S|$. We are actually going to work with the RLS representation of $1/f_S(x)$. The set S can be recovered from the $2|S|$ high-order terms of the RLS representation of $1/f_S(x)$: given the high-order terms, we first reconstruct $f_S(x)$ using the conversion algorithm; the entire set S can then be obtained by factoring $f_S(x)$ (which can be done in polynomial time over the finite field $\mathbb{Z}_p$).

Given sets $S_1, \ldots, S_n$ held by n parties $\mathcal{P}_1, \ldots, \mathcal{P}_n$, note that $f_{\cup_i S_i}(x) = \text{lcm}(f_{S_1}(x), \ldots, f_{S_n}(x))$. Rather than have the parties compute the least common multiple directly (which would be difficult to do securely), we have them compute it using the following high-level approach:

1. The parties collectively define random polynomials $r_1(x), \ldots, r_n(x)$ of degree (at most) $d - 1$ in such a way that no coalition of up to t parties knows anything about any of the $r_i(x)$. This is done via standard techniques using Shamir secret sharing (in the semi-honest setting) or a form of verifiable secret sharing (in the malicious setting).
2. The parties securely compute (sufficiently many terms of the RLS of) the sum $\sum_i \frac{r_i(x)}{f_{S_i}(x)}$. Note that

$$\sum_{i=1}^{n} \frac{r_i(x)}{f_{S_i}(x)} = \frac{u(x)}{\text{lcm}(f_{S_1}(x), \ldots, f_{S_n}(x))}$$

   for some polynomial $u(x)$ of degree at most $\deg(\text{lcm}(f_{S_1}(x), \ldots, f_{S_n}(x))) - 1$.
3. Each party locally computes $u'(x)$ and $L(x)$ such that $u'(x)/L(x) = \sum_i \frac{r_i(x)}{f_{S_i}(x)}$ and $\gcd(u'(x), L(x)) = 1$. Each party then factors $L(x)$ over $\mathbb{Z}_p$ and outputs the roots as $\bigcup_i S_i$.

We need to prove both correctness and privacy of the above. To do so we will rely on the following result:

Lemma 1. *Let $f_1(x), \ldots, f_n(x)$ be polynomials of degree $d_1, \ldots, d_n \geq 1$. Say $r_1(x), \ldots, r_n(x)$ are chosen uniformly and independently from the set of polynomials of degree at most $d_i - 1$, respectively, and let $u(x)$ be such that*

$$\frac{u(x)}{\text{lcm}(f_1(x), \ldots, f_n(x))} = \sum_{i=1}^{n} \frac{r_i(x)}{f_i(x)}.$$

Then $u(x)$ is uniformly distributed among polynomials having degree at most $\deg(\mathrm{lcm}(f_1(x), \cdots, f_n(x))) - 1$.

Proof We prove the lemma for $n = 2$; the general case follows by induction. Let $f_1(x)$ and $f_2(x)$ be polynomials of degree d_1, d_2, respectively. Say $r_1(x)$ and $r_2(x)$ are chosen uniformly and independently from the set of polynomials of degree at most $d_1 - 1$ and $d_2 - 1$, respectively, and let $u(x)$ be such that $\frac{r_1(x)}{f_1(x)} + \frac{r_2(x)}{f_2(x)} = \frac{u(x)}{\mathrm{lcm}(f_1(x), f_2(x))}$. We show that $u(x)$ is uniformly distributed among polynomials of degree at most $\deg(\mathrm{lcm}(f_1(x), f_2(x))) - 1$.

Define $f_1'(x) = f_1(x)/\gcd(f_1(x), f_2(x))$, with $f_2'(x)$ defined analogously. We have

$$\begin{aligned}\frac{r_1(x)}{f_1(x)} + \frac{r_2(x)}{f_2(x)} &= \frac{r_1(x)f_2(x) + r_2(x)f_1(x)}{f_1(x)f_2(x)} \\ &= \frac{\gcd\left(f_1(x), f_2(x)\right) \cdot u(x)}{f_1(x)f_2(x)} = \frac{u(x)}{\mathrm{lcm}\left(f_1(x), f_2(x)\right)}\end{aligned}$$

where $u(x) = r_1(x)f_2'(x) + r_2(x)f_1'(x)$ has degree at most

$$d' \stackrel{\text{def}}{=} \deg(\mathrm{lcm}(f_1(x), f_2(x))) - 1.$$

Identifying a polynomial of degree at most d with a vector over $\mathbb{Z}_p$ of length $d + 1$, consider the map $M : \mathbb{Z}_p^{d_1} \times \mathbb{Z}_p^{d_2} \to \mathbb{Z}_p^{d'+1}$ defined via $M(r_1(x), r_2(x)) = r_1(x)f_2'(x) + r_2(x)f_1'(x)$. Say $M(r_1(x), r_2(x)) = M(r_1'(x), r_2'(x))$. This implies

$$(r_1(x) - r_1'(x)) \cdot f_2'(x) = (r_2'(x) - r_2(x)) \cdot f_1'(x).$$

Since $\gcd(f'(x), g'(x)) = 1$, the above holds iff there exists some $h(x) \in \mathbb{Z}_p[x]$ such that

$$\begin{aligned}r_1(x) - r_1'(x) &= h(x) \cdot f_1'(x) \\ r_2'(x) - r_2(x) &= h(x) \cdot f_2'(x).\end{aligned}$$

Note that $\deg(h) \leq \gcd(f_1(x), f_2(x)) - 1$ because of the bound on the degrees of $r_1(x)$, $r_1'(x)$, $r_2(x)$, and $r_2'(x)$. The above means that each point $M(r_1(x), r_2(x))$ in the image of M has exactly $K \stackrel{\text{def}}{=} p^{\gcd(f_1(x), f_2(x))}$ pre-images. Furthermore, since

$$\begin{aligned}|\mathbb{Z}_p^{d_1} \times \mathbb{Z}_p^{d_2}|/K &= p^{d_1+d_2}/p^{\gcd(f_1(x), f_2(x))} \\ &= p^{\mathrm{lcm}(f_1(x), f_2(x))} = p^{d'+1} = |\mathbb{Z}_p^{d'+1}|,\end{aligned}$$

we see that M is also surjective. Since M is regular and surjective, choosing $r_1(x), r_2(x)$ uniformly and independently at random yields a uniform element $u(x) = M(r_1(x), r_2(x))$ in its range. ■

Correctness and privacy now follow easily from the lemma. Since $u(x)$ is random, and the universe $\mathcal{U}$ is a negligible fraction of $\mathbb{Z}_p$, the probability that

$u(x)$ and $\mathrm{lcm}(f_{S_1}(x), \ldots, f_{S_n}(x))$ have a factor in common is negligible. Thus, $u'(x) = u(x)$ and $L(x) = \mathrm{lcm}(f_{S_1}(x), \ldots, f_{S_n}(x))$ with overwhelming probability and so correctness holds. Moreover, the view of any coalition of up to t parties can be simulated given the result $\bigcup_i S_i$, implying privacy. This simulation is done as follows. Let $D = |\bigcup_i S_i|$. Compute $f_{\cup_i S_i}(x)$, choose a random polynomial $u(x)$ of degree at most $D-1$, and then compute (sufficiently many high-order terms of) the RLS representation of $u(x)/f_{\cup_i S_i}(x)$.

In the next sections, we fill in the missing details in the above description and give a protocol for computing set union in the semi-honest setting. We then show how to extend the protocol to the malicious setting as well.

3.2 (Verifiable) Secret Sharing of Polynomials

In our protocols, we use Shamir's secret-sharing scheme [24] in the semi-honest model, and the verifiable secret-sharing (VSS) protocol of Gennaro et al. [12], denoted GRR-VSS scheme, in the malicious model. (We assume the availability of private channels between all pairs of parties.) In either case, addition of shares can be performed locally (without interaction), and multiplication of shares can be done using a suitable multiplication sub-protocol (i.e., Simple-Mult in the semi-honest model, and Mult in the malicious model [12]).

A polynomial can be (verifiably) shared by (verifiably) sharing each of its coefficients. Addition and multiplication of polynomial shares follows from addition and multiplication of the underlying shares of the coefficients. In particular, addition of polynomial shares can be done locally. Multiplication of two shared polynomials of degrees d_1, d_2 requires $O(d_1 \cdot d_2)$ invocations of an underlying Mult protocol (plus local additions); nevertheless, because these can be parallelized, the entire process takes only a constant number of rounds.

3.3 A Protocol Secure against Honest-but-Curious Adversaries

We propose a privacy-preserving set-union protocol, denoted PPSU-HBC, for the honest-but-curious (HBC) adversary model. Every party contributes to obtaining $\cup_{i\in[1,n]} S_i$, where S_i is the private set of the i-th party; however, a semi-honest adversary corrupting less than $n/2$ parties should not obtain additional information about the set of any other party (except for its size). For simplicity here, we assume that for each set S_i has the same cardinality, denoted by k.

In Figure 1, we present the protocol. The basic idea follows the overview from Section 3.1. Each party $\mathcal{P}_i$ contributes random polynomials $r_{ij}(x)$ for $j \in \{1, \ldots, n\}$. Define $r_j(x) = \sum_{i=1}^{n} r_{ij}(x)$. The parties then (privately) compute the high-order $2nk$ terms of the RLS representation of

$$U(x) = \sum_{j\in[1,n]} \frac{r_j(x)}{f_j(x)},$$

where $f_j(x)$ is the polynomial associated with the set of party $\mathcal{P}_j$. To compute the $2nk$ higher-order terms of $U(x)$, we utilize the fact that the $2nk$ higher-order

Private Input for each party $\mathcal{P}_i$ ($i \in [1,n]$): A set $S_i \subset \mathcal{U}$ of size k.
Goal: Each party obtains $\bigcup_{i\in[1,n]} S_i$.

Each party $\mathcal{P}_i$:

1. Constructs $f_i(x) = \prod_{\alpha\in S_i}(x-\alpha)$, runs RationalToRLS$(1, f_i(x), (2n+1)k-1) \to (\frac{1}{f_i(x)})_{[-(2n+1)k+1,-k]}$, defines $\tilde{f}_i(x) := (\frac{1}{f_i(x)})_{[-(2n+1)k+1,-k]} \cdot x^{(2n+1)k-1}$, and chooses random polynomials $r_{ij}(x)$ of degree at most $k-1$ for $j \in [1,n]$.
2. Secret shares $\tilde{f}_i(x)$ and $r_{ij}(x)$ for $\forall j \in [1,n]$ in parallel.
3. Locally sums his shares of $r_{ij}(x)$ to obtain shares of $r_j(x) = \sum_{i\in[1,n]} r_{ij}(x)$ for $\forall j \in [1,n]$.
4. Runs (in parallel) a shared polynomial multiplication protocol to compute shares of $\tilde{f}_j(x) \cdot r_j(x)$ for $\forall j \in [1,n]$.
5. Locally sums his shares of $\tilde{f}_j(x) r_j(x)$ to obtain shares of the $2nk$ high-order terms of $U'(x) = \sum_{j\in[1,n]} \tilde{f}_j(x) R_j(x)$ (i.e., $U'(x)_{[k-1,(2n+1)k-2]}$).
6. All parties reconstruct the $2nk$ high-order terms of the RLS representation of $U(x)$, and then use these to recover two polynomials $u(x)$ and $L(x)$ such that $(\frac{u(x)}{L(x)})_{[-2nk,-1]} = U(x)_{[k-1,(2n+1)k-2]} \cdot x^{-(2n+1)k+1}$ and $\gcd(u(x), L(x)) = 1$. Then, each party extracts all roots of $L(x)$.

Fig. 1. The PPSU-HBC protocol.

terms of

$$\sum_{j\in[1,n]} r_j(x) \cdot (\frac{1}{f_j})_{[-(2n+1)k-1,-k]}$$

is equal to that of $U(x)$, where the degree of r_j is $k-1$. Then, each party can recover (the rational function) $U(x)$ from its RLS representation using the conversion algorithm; each party can then compute the union by factoring the denominator of $U(x)$.

Privacy follows from Lemma 1, along with the fact that the $r_j(x)$ are random polynomials for any coalition of fewer than $n/2$ corrupted parties. (This security threshold comes from the threshold needed by the Simple-Mult protocol.)

Theorem 1. *The PPSU-HBC protocol presented in Figure 1 is statistically t-secure against a semi-honest adversary, for any $t < n/2$.*

Proof Let C be a coalition of t corrupted parties controlled by the adversary $\mathcal{A}$, and let H be the set of honest parties. Given all private inputs of corrupted parties and the result $S = \cup_{i\in[1,n]} S_i$, we construct a simulator Sim as follows: It first divides $S \setminus (\cup_{i\in C} S_i)$ into sets $\hat{S}_i$ (for $i \in H$) such that the number of elements in each set is exactly k. (An element may appears in multiple sets, if necessary.) Now, $\cup_{i\in H} \hat{S}_i = S \setminus (\cup_{i\in C} S_i)$ and $|\hat{S}_i| \le k$. Then, Sim runs the PPSU-HBC protocol by treating each $\hat{S}_i$ as private input of an honest party.

We argue that the view of $\mathcal{A}$ in the simulation is identically distributed to the view of $\mathcal{A}$ in the real world. It is easy to see that this holds for steps (1)–

(5) of the protocol. In step (6), the only information revealed consists of the polynomials $u(x)$ and $L(x)$. But (with all but negligible probability) $L(x)$ exactly encodes the union (i.e., $L(x) = \prod_{\alpha \in S}(x - \alpha)$), and $u(x)$ is a random polynomial of appropriate degree (using here the fact that the r_i are uniform conditioned on the adversary's view, since they are generated by summing over random contributions from all parties). ■

Complexity Analysis: Secret sharing requires $O(n^2)$ multiplications in $\mathbb{Z}_p$, and an execution of also uses $O(n^2)$ multiplications.

The computation overheads of $\mathcal{P}_i$ in each step of PPSU-HBC protocol is as follows:

- To compute $f_i(x)$ and $\tilde{f}_i(x)$, $O(nk^2)$ multiplications are required.
- To secret-share the $((2n+1)k-1)$-degree polynomial $\tilde{f}_i(x)$ and $(k-1)$-degree polynomial $r_{ij}(x)$ for $j \in [1, n]$, $O(n^3k)$ multiplications are required.
- To compute $U'(x) = \sum_{j \in [1,n]} \tilde{f}_j(x)(\sum_{i \in [1,n]} r_{ij}(x))$ from $\tilde{f}_j(x)$ and $r_{ij}(x)$, we need $O(n^2k^2)$ multiplications and $O(n^2k^2)$ additions. Therefore, all parties should run Simple-Mult and the local addition $O(n^2k^2)$ times; hence, each party requires $O(n^4k^2)$ multiplications in all.
- To recover $U'(x)_{[k-1,(2n+1)k-2]}$, $O(n^3k)$ multiplications are required.
- To recover a rational function $\frac{G(x)}{F(x)}$ from $U'(x)_{[k-1,(2n+1)k-2]} \cdot x^{-(2n+1)k+1}$, $O(n^2k^2)$ multiplications are required. To factor a polynomial $F(x)$ with a degree of at most nk, $O((nk)^{1.5+o(1)} + (nk)^{1+o(1)} \log p)$ multiplications are required [20, 26].

Therefore, the total computation cost is $O(n^4k^2 + n^2k^2 \log p)$ multiplications in $\mathbb{Z}_p$.

The communication overheads of secret-sharing and Simple-Mult are $O(n)$ integers modulus p for each party; hence, the PPSU-HBC protocol's communication cost is $O(n^3k^2)$ elements in $\mathbb{Z}_p$. Further, the round complexity is constant since, in each step, all transmissions can be performed in parallel.

3.4 A Protocol Secure against Malicious Adversaries

We can extend the protocol presented in Section 3.3 to obtain security in the presence of malicious adversaries by using verifiable secret sharing and adding zero-knowledge proofs. Intuitively, in the PPSU-HBC protocol, if we utilize GRR-VSS and Mult instead of secret sharing and Simple-Mult, respectively, no coalition of fewer-than-half corrupted parties can behave maliciously without detection. In addition, however, we require each party to prove that they honestly follow Step (1). Namely, they must prove that $\tilde{f}_j(x)$ is well-formed; that is, that it is the RLS representation of $1/f(x)$ for some f of degree k. We let $\mathsf{ZKPK}[\mathsf{Com}(f(x)), \mathsf{Com}(g(x))]$ denote a zero-knowledge proof that (committed) polynomials f, g of known degree satisfy $g(x) = (\frac{1}{f(x)})_{[-(2n+1)k+1,-k]} \cdot x^{(2n+1)k-1}$. We give the details of such a proof now.

Pedersen commitment scheme. To commit an element in $a \in \mathbb{Z}_p$, we use the Pedersen commitment scheme [22]. Here, a commitment of a is $\mathsf{Com}(a; r) = g^a h^r$ for random $r \in \mathbb{Z}_p$, where g, h are group elements of a cyclic group $\mathbb{G}$ of order p. (When there will be no confusion, we write $\mathsf{Com}(\cdot)$ instead of $\mathsf{Com}(\cdot;\cdot)$.) The Pedersen commitment scheme is additively homomorphic. That is,

$$\mathsf{Com}(a; r) \cdot \mathsf{Com}(b; s) = g^a h^r g^b h^s = g^{a+b} h^{r+s} = \mathsf{Com}(a + b; r + s).$$

In addition, the Pedersen commitment scheme is perfectly hiding and computationally binding under the *discrete logarithm* assumption in $\mathbb{G}$.

Define a commitment of a polynomial $f(x) = \sum_{i\in[0,k]} a_i x^i$ to be a tuple of commitments to its coefficients. Given $\mathsf{Com}(f(x))$ and $\mathsf{Com}(g(x))$ where f, g are monic polynomials of degree k and $\deg(g(x)) = 2nk-1$, respectively, we provide a zero-knowledge proof that $g(x) = (\frac{1}{f(x)})_{[-(2n+1)k+1-,-k]} \cdot x^{(2n+1)k-1}$. (Note that the degrees of f, g can be verified if they are known to be monic by simply decommitting to their high-order coefficient.) The main observation is that the desired relation holds iff $\deg(f(x)g(x) - x^{\deg(f(x))+\deg(g(x))}) < \deg(f(x))$, using the following lemma.

Lemma 2. *If* $f(x), g(x)$ *satisfy* $\deg(f(x)g(x)-x^{\deg(f(x))+\deg(g(x))}) < \deg(f(x))$, *then* $g(x) = (\frac{1}{f(x)})_{[-\deg(f(x))-\deg(g(x)),-\deg(f(x))]} \cdot x^{\deg(f(x))+\deg(g(x))}$.

Proof Let $d_f = \deg(f)$ and $d_g = \deg(g)$. The assumption of the lemma is that $\deg(f(x)g(x) - x^{d_f+d_g}) < d_f$. Then,

$$f(x)g(x) = x^{d_f+d_g} + \sum_{i\in[0,d_f-1]} a_i x^i,$$

and hence,

$$g(x) \cdot x^{-d_f-d_g} = \frac{1}{f(x)} + \frac{1}{f(x)} \cdot \left(\sum_{i\in[-d_f-d_g,-d_g-1]} a_i x^i \right)$$

for some $a_i \in \mathbb{Z}_p$.

Since $g(x) \cdot x^{-d_f-d_g}$ and $\frac{1}{f(x)} \cdot (\sum_{i\in[-d_f-d_g,-d_g-1]} a_i x^i)$ have no common monomials, we obtain

$$g(x) \cdot x^{-d_f-d_g} = (\frac{1}{f(x)})_{[-d_f-d_g,-d_f]}.$$

This concludes the proof. ■

Given the above, a zero-knowledge protocol for $\mathsf{ZKPK}[\mathsf{Com}(f(x)), \mathsf{Com}(g(x))]$ can be constructed using standard techniques, following [5]. We omit the details.

By applying all above changes to PPSU-HBC, we obtain a protocol PPSU-MAL for the malicious adversary model; see Figure 2. Note that GRR-VSS and our zero-knowledge proofs use the Pedersen commitment scheme, which is binding under the *discrete logarithm assumption* so that the security of PPSU-MAL requires such a computational assumption. The following theorem proves the security of PPSU-MAL.

Common Input: A domain of private data $P \subset \mathbb{Z}_p$. Description of GRR-VSS and Mult: common reference strings for public parameters in GRR-VSS.
Private Input for each party $\mathcal{P}_i$ ($i \in [1,n]$): A set S_i of k private elements in P.
Goal: Each party obtains $\cup_{i\in[1,n]} S_i$ without learning other information.

Each party $\mathcal{P}_i$

1. Constructs $f_i(x) = \prod_{\alpha\in S_i}(x-\alpha)$, runs RationalToRLS$(1, f_i(x), (2n+1)k-1) \to (\frac{1}{f_i(x)})_{[-(2n+1)k+1,-k]}$, defines $\tilde{f}_i(x) := (\frac{1}{f_i(x)})_{[-(2n+1)k+1,-k]} \cdot x^{(2n+1)k-1}$, and chooses random polynomials $r_{ij}(x)$ of degree at most $k-1$ for $j \in [1,n]$.
2. Commits to $f_i(x)$ and $\tilde{f}_i(x)$, runs ZKPK[Com(f_i), Com$(\tilde{f}_i)$] protocol.
3. Verifiably secret shares $\tilde{f}_i(x)$ and $r_{ij}(x)$ for $\forall j \in [1,n]$ in parallel.
4. Locally sums his shares of $r_{ij}(x)$ to obtain shares of $r_j(x) = \sum_{i\in[1,n]} r_{ij}(x)$ for each $j \in [1,n]$.
5. Runs (in parallel) a shared polynomial multiplication protocol to compute shares of $\tilde{f}_j(x) \cdot r_j(x)$ for $\forall j \in [1,n]$.
6. Locally sums his shares of $\tilde{f}_j(x) r_j(x)$ to obtain shares of $2nk$ high-order terms of $U'(x) = \sum_{j\in[1,n]} \tilde{f}_j(x) r_j(x)$ (i.e., $U'(x)_{[k-1,(2n+1)k-2]}$).
7. All parties reconstruct $U'(x)_{[k-1,(2n+1)k-2]}$, and then use it to recover two polynomials $u(x)$ and $L(x)$ such that $(\frac{u(x)}{L(x)})_{[-2nk,-1]} = U'(x)_{[k-1,(2n+1)k-2]} \cdot x^{-(2n+1)k+1}$ and $\gcd(u(x), L(x)) = 1$. Then, each party extracts all roots of $L(x)$.

Fig. 2. PPSU-MAL protocol in the malicious adversary model

Theorem 2. *Assuming that the number of corrupted parties is $t < n/2$, where n is the number of all parties of the PPSU-MAL protocol in Figure 2, and that the discrete logarithm assumption holds in the underlying cyclic group, then PPSU-MAL protocol is computationally t-secure in the malicious setting.*

Proof We prove this theorem by showing that for any arbitrarily malicious adversary $\mathcal{A}$ controlling all corrupted parties ($t < n/2$), there exists an efficient simulator $\mathcal{S}$ such that for any inputs to all parties, the view of the corrupted parties and the outputs of the honest parties in the PPSU-MAL protocol are computationally indistinguishable from the outputs of $\mathcal{S}$ and the honest parties in the ideal world interacting with a trusted third party $\mathcal{F}$ computing set union.

Now, we describe $\mathcal{S}$. Let C be a coalition of corrupted parties controlled by $\mathcal{A}$, and H be a set of honest parties.

1. $\mathcal{S}$ generates public parameters for GRR-VSS and Mult and publishes them with securely keeping the discrete logarithms between parameters as a trapdoor. Then, Sim interacts with C on behalf of H. First, it chooses random polynomials $f_i(x)$ of degree at most k, the corresponding $\tilde{f}_i(x)$, and random polynomials $r_{ij}(x)$ of degree at most $k-1$ for $i \in H$ and $j \in [1,n]$. Then, it runs Steps 2 of the PPSU-MAL protocol.

2. From the $\mathsf{ZKPK}[\mathsf{Com}(f_i(x)), \mathsf{Com}(\tilde{f}_i(x))]$ protocol for $i \in C$, Sim extracts witnesses $f_i(x)$ for $i \in C$ using the strong soundness property of the zero-knowledge proof protocol. Then, it computes all roots of $f_i(x)$, which are the inputs of the corrupted parties, using a polynomial factoring algorithm.
3. Let C' be a set of corrupted parties who correctly pass the zero-knowledge proofs protocol in Step 2 and secret-share their inputs in Step 3. Sim participates with the inputs of C' in the ideal world. It receives the result of the ideal PPSU functionality, which is the union of the inputs of C' and H. Then, it computes $U(x) = \frac{u(x)}{L(x)}$, where $L(x)$ is the polynomial associated with the result set of the ideal PPSU functionality and $u(x)$ is a random polynomial of degree at most $\deg(L(x)) - 1$.
4. Sim rewinds $\mathcal{A}$ with the same auxiliary inputs and runs protocol through Step 6 with the same public parameters and polynomials $f_i(x)$ of H as the previous execution.
5. In Step 7, Sim contributes to recover $U(x)_{[-2nk,-1]} \cdot x^{(2n+1)k-1}$. Since Sim has a trapdoor of $\mathsf{GRR\text{-}VSS}$, Sim can equivocate on the recovered secret-shared values. Further, Sim already knows shared secrets of corrupted parties so that Sim can contribute $U(x)_{[-2nk,-1]} \cdot x^{(2n+1)k-1}$ to be recovered in Step 7.
6. Sim outputs a transcript of all interactions with $\mathcal{A}$ in the last execution.

At the end of the simulation, $\mathcal{A}$ obtains the union of all inputs of C' and H. $\mathsf{GRR\text{-}VSS}$, Mult and ZKPK are secure against any probabilistic polynomial-time adversary $\mathcal{A}$ under the discrete logarithm assumption. That is, any probabilistic polynomial-time adversary $\mathcal{A}$ cannot anomalously behave without detection during the protocols $\mathsf{GRR\text{-}VSS}$, Mult and ZKPK when the discrete logarithm assumption holds in the underlying cyclic group. Furthermore, if $\mathcal{A}$ follows the predetermined description of $\mathsf{PPSU\text{-}MAL}$ protocol, then Sim's output $(\text{OUT}^{\mathcal{S}}_{\mathcal{F},\mathcal{S}(\mathsf{aux})}(1^\lambda, \boldsymbol{x}))$ and outputs of H $(\text{OUT}^{\mathsf{hon}}_{\mathcal{F},\mathcal{S}(\mathsf{aux})}(1^\lambda, \boldsymbol{x}))$ in the ideal world is identical to the view of $\mathcal{A}$ $(\text{VIEW}_{\Pi,\mathcal{A}(\mathsf{aux})}(1^\lambda, \boldsymbol{x}))$ and outputs of H $(\text{OUT}^{\mathsf{hon}}_{\Pi,\mathcal{A}(\mathsf{aux})}(1^\lambda, \boldsymbol{x}))$ in the real world, respectively, since $\mathsf{GRR\text{-}VSS}$, Mult and ZKPK are perfectly simulatable when the honest parties are majority. Therefore, there exists only negligible chance in the security parameter that two distributions will be different so that

$$\left\{\text{REAL}_{\Pi,\mathcal{A}(\mathsf{aux})}(1^\lambda, \boldsymbol{x})\right\}_{\lambda\in\mathbb{N},\boldsymbol{x}\in\{0,1\}^*} \quad \text{and} \quad \left\{\text{IDEAL}_{\mathcal{F},\mathcal{S}(\mathsf{aux})}(1^\lambda, \boldsymbol{x})\right\}_{\lambda\in\mathbb{N},\boldsymbol{x}\in\{0,1\}^*}$$

are computationally indistinguishable. ■

Complexity Analysis: In $\mathsf{GRR\text{-}VSS}$, the dealer requires $O(n)$ exponentiations and $O(n^2)$ multiplications in $\mathbb{G}$. The verifier requires $O(1)$ exponentiations. In the reconstruction phase of $\mathsf{GRR\text{-}VSS}$, each party requires $O(n)$ exponentiations and $O(n^2)$ multiplications. In Mult, each party requires $O(n^2)$ exponentiations and $O(n^2)$ multiplications. In local addition, $O(n)$ multiplications are required for each party. The zero-knowledge proofs protocol do not significantly affect the computational and communication complexity. The overall computational overheads and communication overheads of $\mathsf{PPSU\text{-}MAL}$ are $O(n^4k^2)$ exponentiations in $\mathbb{G}$, and $O(n^3k^2)$ group elements $\mathbb{G}$, respectively. The round complexity of the malicious protocol is still $O(1)$.

4 Conclusion

We introduced the Reversed Laurent Series (RLS) representation of a rational function, and showed a surprising relationship between rational function arithmetics (in particular, addition and multiplication) and set union computations. On the basis of these approach, we developed constant-round private set union protocol in both the semi-honest setting and the malicious setting. Our protocol is the first constant-round multi-party private set union protocol without aids of third party.

To the best of our knowledge, this paper shows the first instantiation of using the Reversed Laurent Series for cryptographic purpose. We leave finding other cryptographic applications, either inside or outside secure computing of set operations, as an interesting open problem.

References

1. G. Ateniese, E. De Cristofaro, and G. Tsudik. (If) Size Matters: Size-Hiding Private Set Intersection. In *14th Intl. Conference on Practice and Theory in Public Key Cryptography — PKC 2011*, volume 6571 of *LNCS*, pages 156-173. Springer, 2011.
2. D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In *TCC 2005*, volume 3378 of *LNCS*, pages 325-341, Springer, 2005.
3. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *20th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1–10. ACM Press, 1988.
4. J. Brickell and V. Shmatikov. Privacy-preserving graph algorithms in the semi-honest model. In *Advances in Cryptology — Asiacrypt 2005*, volume 3788 of *LNCS*, pages 236–252. Springer, 2005.
5. J. Camenisch. Proof systems for general statements about discrete logarithms. Technical Report 260, Dept. of Computer Science, ETH Zurich, March 1997.
6. J. Camenisch and G. M. Zaverucha. Private intersection of certified sets. In *Financial Cryptography and Data Security*, volume 5628 of *LNCS*, pages 108–127. Springer, 2009.
7. D. Dachman-Soled, T. Malkin, M. Raykova, and M. Yung. Efficient robust private set intersection. In *7th Intl. Conference on Applied Cryptography and Network Security (ACNS)*, volume 5536 of *LNCS*, pages 125–142. Springer, 2009.
8. E. De Cristofaro, J. Kim, and G. Tsudik. Linear-complexity private set intersection protocols secure in malicious model. In *Advances in Cryptology — Asiacrypt 2010*, volume 6477 of *LNCS*, pages 213–231. Springer, 2010.
9. E. De Cristofaro and G. Tsudik. Practical private set intersection protocols with linear complexity. In *Financial Cryptography and Data Security 2010*, volume 6052 of *LNCS*, pages 143–159. Springer, 2010.
10. M. J. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Advances in Cryptology — Eurocrypt 2004*, volume 3027 of *LNCS*, pages 1–19. Springer, 2004.
11. K. B. Frikken. Privacy-preserving set union. In *5th Intl. Conference on Applied Cryptography and Network Security (ACNS)*, volume 4521 of *LNCS*, pages 237–252. Springer, 2007.

12. R. Gennaro, M. O. Rabin, and T. Rabin. Simplified VSS and fast-track multiparty computations with applications to threshold cryptography. In *17th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 101–111. ACM Press, 1998.
13. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game, or a completeness theorem for protocols with honest majority. In *19th Annual ACM Symposium on Theory of Computing (STOC)*, pages 218–229. ACM Press, 1987.
14. C. Hazay and Y. Lindell. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. In *5th Theory of Cryptography Conference — TCC 2008*, volume 4948 of *LNCS*, pages 155–175. Springer, 2008.
15. C. Hazay and K. Nissim. Efficient set operations in the presence of malicious adversaries. In *13th Intl. Conference on Theory and Practice of Public Key Cryptography — PKC 2010*, volume 6056 of *LNCS*, pages 312–331. Springer, 2010.
16. J. Hong, J. Kim, J. Kim, K. Park and J. Cheon. Constant-Round Privacy Preserving Multiset Union. Available at `http://eprint.iacr.org/2011/138`.
17. S. Jarecki and X. Liu. Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection. In *6th Theory of Cryptography Conference — TCC 2009*, volume 5444 of *LNCS*, pages 577–594. Springer, 2009.
18. E. Kaltofen and V. Shoup. Subquadratic-time factoring of polynomials over finite fields. In Mathematics of Computation, volume 67, number 223, July 1998, pages 1179-1197
19. L. Kissner and D. X. Song. Privacy-preserving set operations. In *Advances in Cryptology — Crypto 2005*, volume 3621 of *LNCS*, pages 241–257. Springer, 2005. See also Technical Report CMU-CS-05-133, Carnegie Mellon University.
20. K. S. Kedlaya and C. Umans. Fast modular composition in any characteristic. in *49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 146-155, IEEE computer Society, 2008.
21. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology — Eurocrypt '99*, volume 1592 of *LNCS*, pages 223–238. Springer, 1999.
22. T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology — Crypto '91*, volume 576 of *LNCS*, pages 129–140. Springer, 1992.
23. Y. Sang and H. Shen. Efficient and secure protocols for privacy-preserving set operations. *ACM Trans. Information and System Security*, 13(1), 2009.
24. A. Shamir. How to share a secret. In *Communications of the ACM*, volume 22, pages 612–613. ACM, 1979.
25. V. Shoup. *A Computational Introduction to Number Theory and Algebra*, second edition. Cambridge University Press, 2009.
26. C. Umans. Fast polynomial factorization and modular composition in small characteristic. In *40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 481-490, ACM, 2008.