Invertible Polynomial Representation for Private Set Operations

Jung Hee Cheon(✉), Hyunsook Hong, and Hyung Tae Lee

CHRI and Department of Mathematical Sciences, Seoul National University,
1 Gwanak-ro, Gwanak-gu, Seoul 151-747, Korea
{jhcheon,hongsuk07,htsm1138}@snu.ac.kr

Abstract. In many private set operations, a set is represented by a polynomial over a ring $\mathbb{Z}_\sigma$ for a composite integer σ, where $\mathbb{Z}_\sigma$ is the message space of some additive homomorphic encryption. While it is useful for implementing set operations with polynomial additions and multiplications, it has a limitation that it is hard to recover a set from a polynomial due to the hardness of polynomial factorization over $\mathbb{Z}_\sigma$.

We propose a new representation of a set by a polynomial over $\mathbb{Z}_\sigma$, in which σ is a composite integer with known factorization but a corresponding set can be efficiently recovered from a polynomial except negligible probability. Since $\mathbb{Z}_\sigma[x]$ is not a unique factorization domain, a polynomial may be written as a product of linear factors in several ways. To exclude irrelevant linear factors, we introduce a special encoding function which supports early abort strategy. Our representation can be efficiently inverted by computing all the linear factors of a polynomial in $\mathbb{Z}_\sigma[x]$ whose roots locate in the image of the encoding function.

As an application of our representation, we obtain a constant-round private set union protocol. Our construction improves the complexity than the previous without honest majority.

Keywords: Polynomial representation · Polynomial factorization · Root finding · Privacy-preserving set union

1 Introduction

Privacy-preserving set operations (PPSO) are to compute set operations of participants' dataset without revealing any information other than the result. There have been many proposals to construct PPSO protocols with various techniques such as general MPC [1,9], polynomial representations [7,8,10,12,18], pseudorandom functions [11], and blind RSA signatures [4,5]. While the last two techniques are hard to be generalized into multi-party protocols, polynomial representations combining with additive homomorphic encryption (AHE) schemes enable us to have multi-party PPSO protocols for various operations including set intersection [8,12,18], (over-)threshold set union [12], element reduction [12]

This work includes some part of the third author's PhD thesis [14].

H.-S. Lee and D.-G. Han (Eds.): ICISC 2013, LNCS 8565, pp. 277–292, 2014.
DOI: 10.1007/978-3-319-12160-4_17

and so on. Among these constructions, set intersection protocols run in constant rounds, but others run in linear of the number of participants.

Let us focus on privacy-preserving set union protocols. There are two obstacles to construct constant round privacy-preserving multi-party set union protocols based on the polynomial representation with AHE schemes. First, in the polynomial representation set union corresponds to polynomial multiplication, which is not supported by an AHE scheme in constant rounds. Second, to recover the union set from the resulting polynomial, we need a root finding algorithm of a polynomial over $\mathbb{Z}_\sigma$, where $\mathbb{Z}_\sigma$ is the message space of the AHE scheme.

Recently, Seo et al. [19] proposed a constant round set union protocol based on a novel approach in which a set is represented as a rational function using the reversed Laurent series. In their protocol, each participant takes part in the protocol with a rational function whose poles consist of the elements of his set and at the end of the protocol he obtains a rational function whose poles correspond to the set union. Then each participant recovers the denominator of the rational function using the extended Euclidean algorithm and finds the roots of the denominator. Since each rational function is summed up to the resulting function after encrypted under an AHE scheme, the first obstacle is easily overcome.

However, a root finding is still problematic on the message space $\mathbb{Z}_\sigma$ of the AHE schemes. Since the message space has unknown order [16] or is not a unique factorization domain (UFD) [2,15,17] in the current *efficient* AHE schemes, there is no proper polynomial factorization or root finding algorithm working on the message space. To avoid this obstacle, the authors in [19] utilized a secret sharing scheme. However, it requires computational and communicational costs heavier than the previous and requires an honest majority for security since their protocol exploits a secret sharing scheme to support privacy-preserving multiplications in constant rounds.

Our Contribution. Let $\sigma = \prod_{j=1}^{\bar{\ell}} q_j$ for distinct primes q_j, which is larger than the size of the universe of set elements. We propose a new representation of a set by a polynomial over $\mathbb{Z}_\sigma$ in which a corresponding set can be efficiently recovered from a polynomial except negligible probability when the factorization of σ is given.

For a given polynomial $f(x) = \prod_{i=1}^{d}(x - s_i) \in \mathbb{Z}_\sigma[x]$, if the factorization of σ is given, one can obtain all roots of f in $\mathbb{Z}_{q_j}$ for each j by exploiting a polynomial factorization algorithm over a finite field $\mathbb{Z}_{q_j}$ [22]. By reassembling the roots of f in $\mathbb{Z}_\sigma$ using the Chinese Remainder Theorem (CRT), we can obtain all the candidates. However, the number of candidates amounts to $d^{\bar{\ell}}$, which is exponential in the size of the universe.

We introduce a special encoding function ι to exclude irrelevant candidates efficiently. For a polynomial $f = \prod_{i=1}^{d}(x - \iota(s_i)) \in \mathbb{Z}_\sigma[x]$, our encoding function aborts most irrelevant candidates without $d^{\bar{\ell}}$ CRT computations, by giving a certain relation among roots of f in $\mathbb{Z}_{q_j}[x]$ and roots of f in $\mathbb{Z}_{q_{j+1}}[x]$. As a

Table 1. Comparison with previous set-union protocols

HBC	Rounds	Communication cost	Computational cost	# of honest party
[12]	$O(n)$	$O(n^3 k \tau_N)$	$O(n^4 k^2 \tau_N \rho_N)$	≥ 1
[8]	$O(n)$	$O(n^2 k \tau_N)$	$O(n^2 k^2 \tau_N \rho_N)$	≥ 1
[19]	$O(1)$	$O(n^4 k^2 \tau_{p'})$	$O(n^5 k^2 \rho_{p'})$	$\geq n/2$
Ours	$O(1)$	$O(n^3 k \tau_N)$	$O(n^3 k^2 \tau_N \rho_N)$	≥ 1
Malicious	Rounds	Communication cost	Computational cost	# of honest party
[8]	$O(n)$	$O((n^2 k^2 + n^3 k)\tau_N)$	$O(n^2 k^2 \tau_N \rho_N)$	≥ 1
[19]	$O(1)$	$O(n^4 k^2 \tau_p)$	$O(n^5 k^2 \tau_p \rho_p)$	$\geq n/2$
Ours	$O(1)$	$O(n^3 k^2 \tau_N)$	$O(n^3 k^2 \tau_N \rho_N)$	≥ 1

n: the number of participants, k: the maximum size of sets

$\tau_N, \tau_{p'}, \tau_p$: the size of modulus N for Paillier encryption scheme or NS encryption scheme, the size p' of representing domain, the order p of a cyclic group for Pedersen commitment scheme, respectively

$\rho_N, \rho_{p'}, \rho_p$: modular multiplication cost of modulus N for Paillier encryption scheme or NS encryption scheme, p' for the size of representing domain, p for the order of a cyclic group for Pedersen commitment scheme, respectively

result, our encoding function enables us to efficiently recover all the roots of f with negligible failure probability if they are in the image of ι.

As an application of our representation, combining with Naccache-Stern (NS) AHE scheme which is the factorization of σ is public, we obtain an efficient constant round privacy-preserving set union protocol without an honest majority. In Table 1, we compare our set union protocols with the previous main results [8,12,19].

Organization. In Sect. 2 we look into some components of our privacy-preserving set union protocol, including polynomial representation and AHE schemes. We provide our new polynomial representation that enables us to uniquely factorize a polynomial satisfying some criteria in Sect. 3. Our constant round privacy-preserving set union protocols are presented in Sect. 4. Some supplying materials including analysis of our representation are given in Appendix.

2 Preliminaries

In this section, we look into the polynomial representation of a set for PPSO protocols and introduce efficient AHE schemes utilized in PPSO protocols to support polynomial operations between encrypted polynomials.

2.1 Basic Definitions and Notations

Throughout the paper, let $\mathcal{U}$ be the universe, n the number of participants in the protocol, and k the maximum size of participants' datasets S_i's. Also, d denotes the size of (multi-)set union among participants' datasets in the protocol.

Let $R[x]$ be a set of polynomials defined over a ring R and $R(x)$ be a set of rational functions defined over R, *i.e.*, $R[x] = \{f(x) | f(x) = \sum_{i=0}^{\deg f} f[i]x^i$ and $f[i] \in R$ for all $i\}$ and $R(x) = \{\frac{f(x)}{g(x)} | f(x), g(x) \in R[x], g(x) \neq 0\}$. For a polynomial $f \in R[x]$, we denote the coefficient of x^i in a polynomial f by $f[i]$, *i.e.*, $f(x) = \sum_{i=0}^{\deg f} f[i]x^i \in R[x]$. For a polynomial $f(x) = \sum_{i=0}^{\deg f} f[i]x^i \in \mathbb{Z}_\sigma[x]$ and a factor q of σ, $f \bmod q$ denotes a polynomial $\sum_{i=0}^{\deg f}(f[i] \bmod q)x^i \in \mathbb{Z}_q[x]$.

We also define a negligible function as follows: a function $g : \mathbb{N} \to \mathbb{R}$ is *negligible* if for every positive polynomial $\mu(\lambda)$, there exists an integer N such that $g(\lambda) < 1/\mu(\lambda)$ for all $\lambda > N$.

2.2 Polynomial Representation of a Set

Let R be a commutative ring with unity and S be a subset of R. We may represent a set S by a polynomial or a rational function over R.

Polynomial Representation. In some previous works [7,8,10,12,19], a set S can be represented by a polynomial $f_S(x) \in R[x]$ whose roots are the elements of S. That is, $f_S(x) := \prod_{s_i \in S}(x - s_i)$. This representation gives the following relation: $f_S(x) + f_{S'}(x) = \gcd(f_S(x), f_{S'}(x)) \cdot u(x)$ for some polynomial $u(x) \in R[x]$ and hence the roots of a polynomial $f_S(x) + f_{S'}(x)$ are the elements of $S \cap S'$ with overwhelming probability. Also, the roots of $f_S(x) \cdot f_{S'}(x)$ are the elements of $S \cup S'$ as multi-sets.

Rational function Representation. Recently, Seo et al. [19] introduced a novel representation of a set $S \subset R$ by a rational function F_S over R whose poles consist of the elements of S. That is, $F_S(x) := \dfrac{1}{\prod_{s_i \in S}(x - s_i)} = \dfrac{1}{f_S(x)}$. This representation provides the following relation:

$$F_S(x) + F_{S'}(x) = \frac{f_S(x) + f_{S'}(x)}{f_S(x) \cdot f_{S'}(x)} = \frac{\gcd(f_S(x), f_{S'}(x)) \cdot u(x)}{f_S(x) \cdot f_{S'}(x)}$$
$$= \frac{u(x)}{\operatorname{lcm}(f_S(x), f_{S'}(x))}$$

for some polynomial $u(x) \in R[x]$ which is relatively prime to $\operatorname{lcm}(f_S(x), f_{S'}(x))$ with overwhelming probability. Hence the poles of $F_S(x) + F_{S'}(x)$ are exactly the roots of $\operatorname{lcm}(f_S(x), f_{S'}(x))$, which are the elements of $S \cup S'$ as sets, not multisets, if $u(x)$ and $\operatorname{lcm}(f_S(x), f_{S'}(x))$ have no common roots. This rational function is represented again by an infinite formal power series, so called a *Reversed Laurent Series* (RLS), in [19].

2.3 Additive Homomorphic Encryption

Let us consider a commutative ring R with unity and a R-module G where $r \cdot g := g^r$ for $r \in R$ and $g \in G$. Let $\mathsf{Enc}_{\mathsf{pk}} : R \to G$ be a public key encryption under the public key pk. We can define a public key encryption for a polynomial $f = \sum_{i=0}^{\deg f} f[i]x^i \in R[x]$ as follows: $\mathcal{E}_{\mathsf{pk}}(f) := \sum_{i=0}^{\deg f} \mathsf{Enc}_{\mathsf{pk}}(f[i])x^i$.

Assume $\mathsf{Enc}_{\mathsf{pk}}$ has an additive homomorphic property. Then one can easily induce polynomial addition between encrypted polynomials and polynomial multiplication between an unencrypted polynomial and an encrypted polynomial.

There have been several efficient AHE schemes [15–17]: Under the assumption that factoring $N = p^2q$ is hard, Okamoto and Uchiyama [16] proposed a scheme with $R = \mathbb{Z}_p$ and $G = \mathbb{Z}_N$, in which the order p of the message space R is hidden. With the decisional composite residuosity assumption, Paillier scheme [17] has $R = \mathbb{Z}_N$ and $G = \mathbb{Z}_{N^2}$ for $N = pq$, in which the size of message spaces is a hard-to-factor composite integer N. Naccache and Stern [15] proposed a scheme with $R = \mathbb{Z}_\sigma$ and $G = \mathbb{Z}_N$ under the higher residuosity assumption, where $N = pq$ is a hard-to-factor integer and σ is a product of small primes dividing $\phi(N)$ for Euler's totient function ϕ.

In the above schemes, it is hard to find the roots of a polynomial in $R[x]$ without knowing a secret key. For the second case, in fact, Shamir [20] showed that to find a root of a polynomial $f(x) = \prod_{i=1}^{d}(x - s_i) \in \mathbb{Z}_N[x]$ is equivalent to factor N. While, in the NS scheme, it may be possible to compute some roots of a polynomial in $\mathbb{Z}_\sigma[x]$ since the factorization of σ is public. But $\mathbb{Z}_\sigma[x]$ is not a UFD and hence the number of roots of a polynomial $f \in \mathbb{Z}_\sigma[x]$ can be larger than $\deg f$. In fact, if $f(x) = \prod_{i=1}^{d}(x-s_i) \in R[x]$, then the number of candidates of roots of the polynomial f is $d^{\bar{\ell}}$ where $\bar{\ell}$ is the number of prime factors of σ. We will use the NS scheme by presenting a method to efficiently recover all the roots of a polynomial $f \in \mathbb{Z}_\sigma[x]$ satisfying some criteria.

3 Invertible Polynomial Representation

In this section, we provide our new polynomial representation that enables us to efficiently recover the exact corresponding set from the polynomial represented by our suggestion.

Focus on the fact that the factorization of σ is public in the NS encryption scheme. Using this fact, given a polynomial $f = \prod_{i=1}^{d}(x - s_i) \in \mathbb{Z}_\sigma[x]$ for a set $S = \{s_1, \ldots, s_d\}$, one can obtain all roots of $f \mod q_j$ for each j by applying a polynomial factorization algorithm over a finite field $\mathbb{Z}_{q_j}$ such as Umans' [22]. To recover S, one can perform CRT computation for obtaining less than $d^{\bar{\ell}}$ candidates of roots of f over $\mathbb{Z}_\sigma$. In general, however, the number of roots of f over $\mathbb{Z}_\sigma$ is larger than $\deg f$ and there is no criteria to determine the exact set S. To remove irrelevant roots which are not in S, we give some relations among all roots of polynomials $f \mod q_j$'s by providing an encoding function.

3.1 Our Polynomial Representation

We present our polynomial representation for supporting to recover a set from a polynomial over $\mathbb{Z}_\sigma$ represented by our suggestion.

Parameter Setting. Let us explain parameters for our polynomial representation and PPSO protocols. First, set the bit size of the modulus N of the NS encryption scheme by considering a security parameter λ. For the given universe $\mathcal{U}$ and the maximum size d of the resulting set union (here, $d = nk$ for the number n of participants and the maximum size k of participants' datasets), let $d_0 = \max\{d, \lceil \log N \rceil\}$ and set $\tau = \frac{1}{3}(\log d + 2\log d_0)$. This setting comes from the computational complexity analysis of our set union protocol and the value τ will influence the bit size of prime factors of σ and the size of the message space of the NS encryption scheme. See Sect. 4 for details.

Set the parameter ℓ and α so that ℓ is the smallest positive integer such that $\mathcal{U} \subseteq \{0,1\}^{3\tau\alpha\ell}$ for some rational number $0 < \alpha < 1$ satisfying $3\alpha\tau$ and $3(1-\alpha)\tau$ are integers. Note that the proper size of α is $\frac{1}{3}$,*i.e.*, $\mathcal{U} \subseteq \{0,1\}^{\tau\ell}$ for optimization. If $\alpha \neq \frac{1}{3}$, the expected computation is in polynomial time only when the size of the universe is restricted. Details about the proper size of α is given in the full version of this paper [3].

Then, set the proper size $\bar{\ell}$ larger than ℓ and let $\ell' = \bar{\ell} - \ell$. The analysis of the proper size of $\bar{\ell}$ will be discussed at the end of Sect. 3.1. Choose $\bar{\ell}$ $(3\tau + 1)$-bit distinct primes q_j's and set $\sigma = \prod_{j=1}^{\bar{\ell}} q_j$. Note that the size of the message space of the NS encryption scheme is less than $\frac{N}{4}$ for its security [15]. Hence, the parameters have to be satisfied the condition $\sigma < \frac{N}{4}$ and so $\bar{\ell} < \frac{\lfloor \log N \rfloor - 2}{3\tau}$. Also, we assume that $\bar{\ell}$ is smaller than d for optimal complexity of our proposed protocol. In summary, the parameter $\bar{\ell}$ is smaller than $\min\{d, \frac{\lfloor \log N \rfloor - 2}{3\tau}\}$ (Fig. 1).

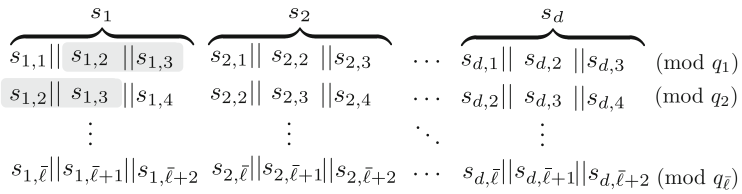
s_1
s_2
s_d
$s_{1,1}||\ s_{1,2}\ ||s_{1,3}$
$s_{2,1}||\ s_{2,2}\ ||s_{2,3}$
$\cdots$
$s_{d,1}||\ s_{d,2}\ ||s_{d,3}$
(mod q_1)
$s_{1,2}||\ s_{1,3}\ ||s_{1,4}$
$s_{2,2}||\ s_{2,3}\ ||s_{2,4}$
$\cdots$
$s_{d,2}||\ s_{d,3}\ ||s_{d,4}$
(mod q_2)
$\vdots$
$\vdots$
$\ddots$
$\vdots$
$s_{1,\bar{\ell}}||s_{1,\bar{\ell}+1}||s_{1,\bar{\ell}+2}$
$s_{2,\bar{\ell}}||s_{2,\bar{\ell}+1}||s_{2,\bar{\ell}+2}$
$\cdots$
$s_{d,\bar{\ell}}||s_{d,\bar{\ell}+1}||s_{d,\bar{\ell}+2}$
(mod $q_{\bar{\ell}}$)

Fig. 1. Our encoding function ι

Encoding by Repetition. Let $h : \{0,1\}^* \to \{0,1\}^{2\tau}$ and $h_j : \{0,1\}^* \to \{0,1\}^{\tau}$ be uniform hash functions for $1 \leq j \leq \ell'$. Parse a message $s_i \in \mathcal{U} \subseteq \{0,1\}^{\tau\ell}$ into ℓ blocks $s_{i,1}, \ldots, s_{i,\ell}$ of τ-bit so that $s_i = s_{i,1}||\cdots||s_{i,\ell}$. Let $s_{i,\ell+j} = h_j(s_i)$ for $1 \leq j \leq \ell'$ and parse $h(s_i)$ into two blocks $s_{i,\bar{\ell}+1}$ and $s_{i,\bar{\ell}+2}$ of τ-bit. We define our encoding function $\iota : \mathcal{U} \subseteq \{0,1\}^{\tau\ell} \to \mathbb{Z}_{\sigma}$, in which $\iota(s_i)$ is the unique element in $\mathbb{Z}_{\sigma}$ satisfying $\iota(s_i) \equiv s_{i,j}||s_{i,j+1}||s_{i,j+2} \mod q_j$ for $1 \leq j \leq \bar{\ell}$. Then a set S is represented as a polynomial $f_S(x) = \prod_{s_i \in S}(x - \iota(s_i)) \in \mathbb{Z}_{\sigma}[x]$.

Decoding Phase. Denote by $s_j^{(i)} := \iota(s_i) \mod q_j$ for each message $s_i = s_{i,1}||\cdots||s_{i,\ell}$. For $1 \leq j \leq \bar{\ell} - 1$, we define $(s_j^{(i)}, s_{j+1}^{(i')}) \in \mathbb{Z}_{q_j} \times \mathbb{Z}_{q_{j+1}}$ to be a *linkable pair* if the last (2τ)-bit of $s_j^{(i)}$ is equal to the first (2τ)-bit of $s_{j+1}^{(i')}$, *i.e.*, $s_{i,j+1}||s_{i,j+2} = s_{i',j+1}||s_{i',j+2}$. Inductively, we also define $(s_1^{(i_1)}, \cdots, s_{j+1}^{(i_{j+1})}) \in \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_{j+1}}$ to be a *linkable pair* if $(s_1^{(i_1)}, \cdots, s_j^{(i_j)})$ and $(s_j^{(i_j)}, s_{j+1}^{(i_{j+1})})$ are linkable pairs (Fig. 2).

$$\begin{aligned} s_1^{(i_1)} &= s_{i_1,1}||s_{i_1,2}||s_{i_1,3}\\ s_2^{(i_2)} &= \qquad\;\; s_{i_2,2}||s_{i_2,3}||s_{i_2,4}\\ s_3^{(i_3)} &= \qquad\qquad\quad s_{i_3,3}||s_{i_3,4}||s_{i_3,5}\\ \Rightarrow &\left(s_1^{(i_1)}, s_2^{(i_2)}, s_3^{(i_3)}\right) \text{ is a linkable pair.}\end{aligned}$$

Fig. 2. Linkable pair

Let $\iota(s_i)$ and $\iota(s_{i'})$ be images of elements s_i and $s_{i'}$ of the function ι with $s_i \neq s_{i'}$. We can easily check the following properties:

- $\left(s_1^{(i)}, \cdots, s_{j+1}^{(i)}\right)$ is always a linkable pair.
- When s_i and $s_{i'}$ are uniformly chosen strings from $\{0,1\}^{\tau\ell}$,

$$\begin{aligned}\Pr[(s_j^{(i)}, s_j^{(i')}) \text{ is a linkable pair}] &= \Pr\left[s_{i,j+1}||s_{i,j+2} = s_{i',j+1}||s_{i',j+2}\right]\\ &= \frac{1}{2^{2\tau}} \qquad (1)\end{aligned}$$

for a fixed $1 \leq j \leq \bar{\ell}$.

At decoding phase, when a polynomial $f(x) = \prod_{i=1}^{d}(x - \iota(s_i)) \in \mathbb{Z}_\sigma[x]$ is given, we perform two phases to find the correct d roots of the polynomial $f(x)$. In the first stage, one computes all the roots $\{s_j^{(1)}, \cdots, s_j^{(d)}\}$ over $\mathbb{Z}_{q_j}[x]$ for each j. For each j sequentially from 1 to $\bar{\ell} - 1$, we find all the linkable pairs among $\{s_j^{(1)}, \cdots, s_j^{(d)}\}$ and $\{s_{j+1}^{(1)}, \cdots, s_{j+1}^{(d)}\}$ by checking whether the last (2τ)-bit of $s_j^{(i)}$ and the first (2τ)-bit of $s_{j+1}^{(i')}$ are the same. It can be done by d^2 comparisons or $O(d \log d)$ computations using sorting and determining algorithms.

After $\bar{\ell} - 1$ steps, we obtain d' linkable pairs of $\bar{\ell}$-tuple, which are candidates of roots of the polynomial f and elements of the set. It includes the d elements corresponding to $\iota(s_1), \ldots, \iota(s_d)$. If d' is much larger than d, it can be a burden. However, we can show that the expected value of d' is at most $3d$ in Theorem 1. See the end of this section.

After obtaining d' linkable pairs of $\bar{\ell}$-tuple, in the second phase, we check whether each pair belongs to the image of ι with the following equalities:

$$s_{i,\ell+j} = h_j(s_i) \qquad \text{for all } 1 \leq j \leq \ell', \qquad (2)$$

$$s_{i,\bar{\ell}+1}||\, s_{i,\bar{\ell}+2} = h(s_i). \qquad (3)$$

The linkable pairs of $\bar{\ell}$-tuple, corresponding to $\iota(s_i)$ for some i clearly satisfies the above equations. However, for a random $\bar{\ell}$-tuple in $\mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_{\bar{\ell}}}$, the probability that it satisfies the relation (2) is about $\frac{1}{2^{\tau\ell'}}$ and the probability that it satisfies the relation (3) is about $\frac{1}{2^{2\tau}}$ under the assumption that h and h_j's are uniform hash functions. Hence, the expected number of wrong $\bar{\ell}$-tuples passing both phases is less than $d \times \frac{1}{2^{\tau(2+\ell')}}$. It is less than $2^{-\lambda}$ for a security parameter λ if we take the parameter ℓ' to satisfy

$$\ell' > \frac{3(\lambda + \log d)}{\log d + 2\log d_0} - 2. \tag{4}$$

For example, when $\lambda = 80$ and $d \approx d_0 \approx 2^{10}$, then ℓ' is about 8. Therefore, one can recover a set from the given polynomial represented by our suggestion without negligible failure probability in the security parameter.

3.2 Analysis of Our Polynomial Representation

Computational Complexity. Let us count the computational cost of our representation. The encoding phase consists of two steps: (1) the CRT computation per each element to obtain a value of the encoding function ι and (2) the polynomial expansion. The first step requires $O(d \log^2 \sigma)$ bit operations for d elements and the second step requires $O(d^2)$ multiplications. Hence, the complexity for the encoding phase is $O(d^2)$ multiplications.

The decoding phase may be divided into three steps: (1) finding roots of a polynomial f in $\mathbb{Z}_{q_j}$ for each j, (2) finding all linkable pairs of length $\bar{\ell}$, and (3) checking the Eqs. (2) and (3). These steps require $O(\bar{\ell}d^{1.5})$ multiplications, $O(\bar{\ell}d \log d)$ bit operations, and $O(\ell' d)$ hash computations, respectively. Hence, the complexity for the decoding phase is dominated by $O(\bar{\ell}d^{1.5})$ multiplications.

The Expected Number of Linkable Pairs. We analyze the expected number of linkable pairs of $\bar{\ell}$-tuple when we recover a set from a polynomial of degree d, represented by our suggestion. Theorem 1 gives a rigorous analysis of the upper bound of the expected number of linkable pairs of $\bar{\ell}$-tuple. The proof is given in Appendix A.

Theorem 1. *Assume that $S = \{s_1, \ldots, s_d\}$ is a uniformly and randomly chosen set in the set of subsets of cardinality d of the set $\{0,1\}^{\tau\ell}$. Define an encoding function $\iota : \{0,1\}^{\tau\ell} \to \mathbb{Z}_\sigma$ so that $\iota(s_i)$ is the unique element in $\mathbb{Z}_\sigma$ satisfying $\iota(s_i) \equiv s_{i,j}||s_{i,j+1}||s_{i,j+2} \mod q_j$ for all $1 \le j \le \bar{\ell}$ when $s_i = s_{i,1}||\ldots||s_{i,\ell}$ and $s_{i,j}$'s are τ-bit. Assume h and h_j's utilized in the encoding function ι are uniform hash functions. Then the expected number of linkable pairs of $\bar{\ell}$-tuple is at most $3d$ for all polynomials $f_S = \prod_{s_i \in S}(x - \iota(s_i))$.*

4 Applications: Set Union for Honest-but-Curious Case

In this section, we present our set union protocol based on our polynomial representation described in Sect. 3. Our construction exploits the NS AHE scheme to encrypt a rational function whose denominator corresponds to a participant's set. For this we generalize a reversed Laurent series presented in [19] to work on $\mathbb{Z}_\sigma$ with a composite σ, which the domain of the NS scheme. As a result, we obtain set union protocols which improve the complexity than the previous.

4.1 Set Union for Honest-but-Curious Case

Rational Function Representation. We adopt the rational function representation presented in [19]. To represent a set as a rational function, the authors in [19] exploited a reversed Laurent series (RLS): For a positive integer q, a RLS over $\mathbb{Z}_q$ is a singly infinite, formal sum of the form $f(x) = \sum_{i=-\infty}^{m} f[i]x^i$ ($f[m] \neq 0$) with an integer m and $f[i] \in \mathbb{Z}_q$ for all i. For a RLS $f(x)$, we denote $f(x)_{[d_1,d_2]} = \sum_{i=d_1}^{d_2} f[i]x^i$. For polynomials $f, g \in \mathbb{Z}_q[x]$ with $g \neq 0$, we define *the RLS representation of a rational function* f/g by a RLS of f/g. In Fig. 3, we provide an algorithm which takes polynomials $f, g \in \mathbb{Z}_q[x]$ with $\deg f < \deg g$ and an integer k larger than $\deg g$ as inputs and outputs k higher-order terms of the RLS representation of f/g. We also note that if one knows $2\deg g$ higher-order terms of the RLS representation of a rational function f/g, one can recover f' and g' such that $\frac{f'}{g'} = \frac{f}{g}$ [19].

```
Input $f(x), g(x) \in \mathbb{Z}_q[x]$ with $\deg f < \deg g$ and an integer $k > \deg g$
Output $k$ higher-order terms of the RLS representation of a rational function $f/g$

1. $F(x) \leftarrow f(x) \cdot x^k$
2. Compute $Q(x), R(x)$ such that $F(x) = g(x)Q(x) + R(x)$ and $\deg R < \deg g$ using
   a polynomial division algorithm
3. Return $Q(x) \cdot x^{-k}$
```

Fig. 3. RationalToRLS(f, g, k)

While Seo et al.'s constructions work on $\mathbb{Z}_q[x]$ for a prime q, our constructions are based on $\mathbb{Z}_\sigma[x]$ for a composite σ. Hence, one may doubt a RLS representation works well on $\mathbb{Z}_\sigma[x]$. In our protocol, we will represent each participant's set S_i as our polynomial representation $f_{S_i} := \prod_{s_{i,j} \in S_i}(x - \iota(s_{i,j})) \in \mathbb{Z}_\sigma[x]$ with our encoding function ι. Then we convert a rational function of $1/f_{S_i}$ to its RLS over $\mathbb{Z}_\sigma$. Since $\mathbb{Z}_\sigma$ is not a Euclidean domain, one may doubt whether the RationalToRLS algorithm works on $\mathbb{Z}_\sigma[x]$. However, in our protocol, since the conversion requires polynomial divisions only by monic polynomials, it works well on $\mathbb{Z}_\sigma[x]$.

After the end of interactions among participants in our protocol, each participant obtains the $2nk$ higher-order terms of the RLS representation of a rational function $\dfrac{u(x)}{U(x)} = \dfrac{r_1}{f_{S_1}} + \dfrac{r_2}{f_{S_2}} + \cdots + \dfrac{r_n}{f_{S_n}}$ where $U(x) = \operatorname{lcm}(f_{S_1}(x), \ldots, f_{S_n}(x))$ and r_i's are hidden polynomials. There is no algorithm to recover $u'(x)$ and $U'(x)$ in $\mathbb{Z}_\sigma[x]$ such that $\dfrac{u(x)}{U(x)} = \dfrac{u'(x)}{U'(x)}$. However, from our polynomial representation, it only requires $U'(x) \mod q_j$ for each j and we can obtain $U'(x) \mod q_j$ from the RLS representation modulo q_j by running polynomial recovering algorithm on $\mathbb{Z}_{q_j}[x]$'s.

The correctness and the security of our set union protocol are induced from properties of a RLS representation. We omit the details due to the space limitation. See the full version [3] of this paper for these.

Threshold Naccache-Stern Encryption. For a group decryption, it requires a semantically secure, threshold NS AHE scheme in our protocol. One can easily construct a threshold version of the NS encryption scheme using the technique of Fouque et al. [6], which transforms the original Paillier homomorphic encryption scheme into a threshold version working from Shoup's technique [21].

Parameter Setting. Let $\mathcal{U}$ be the universe, n be the number of participants, and k be the maximum size of participants' datasets. Let d be the possible maximum size of the set union, *i.e.*, $d = nk$. Take the bit size of N by considering the security of the threshold NS AHE scheme, which is the modulus of the threshold NS AHE scheme. Put $d_0 = \max\{d, \lceil \log N \rceil\}$ and $\tau = \frac{1}{3}(\log d + 2 \log d_0)$. Set ℓ so that $\mathcal{U} \subseteq \{0,1\}^{\tau\ell}$, a proper size of ℓ' so that ℓ' satisfies the relation (4) and let $\bar{\ell} = \ell + \ell'$. Note that $\bar{\ell}$ is to be smaller than $\min\left\{d, \frac{\lfloor \log N \rfloor - 2}{3 \log \log N}\right\}$ since $\tau \geq \log \log N$. Generate the parameters of the threshold NS encryption scheme, including the size of message space σ, which is a product of $\bar{\ell}$ $(3\tau+1)$-bit distinct primes q_j's.

Our Set Union Protocol for Honest-but-Curious Case. Our set union protocol against honest-but-curious (HBC) adversaries is described in Fig. 4. In our set union protocol, each participant computes the $2nk$ higher-order terms of the RLS representation of $F_{S_i} = \frac{1}{f_{S_i}} \in \mathbb{Z}_\sigma[x]$ where $f_{S_i} = \prod_{s_{i,j} \in S_i}(x - \iota(s_{i,j}))$ for our encoding function ι and sends its encryption to all others. With the received encryptions of F_{S_j} for $1 \leq j \leq n$, each participant $\mathcal{P}_i$ multiplies a polynomial $r_{i,j}$ using additive homomorphic property, which is a randomly chosen polynomial by the participant $\mathcal{P}_i$ and adds all the resulting polynomials to obtain the encryption of $\phi_i(x) = \sum_{j=1}^{n} F_{S_j} \cdot r_{i,j}$. Then, he sends the encryption of $\phi_i(x)$ to all others. After interactions among participants, each participant can obtain the $2nk$ high-order term of the RLS representation of $F(x) = \sum_{i=1}^{n}\left(\sum_{j=1}^{n} \frac{1}{f_{S_j}} \cdot r_{i,j}\right) \in \mathbb{Z}_\sigma[x]$. Then each participant obtains the $2nk$ high-order terms of the RLS representation of F in $\mathbb{Z}_\sigma[x]$ with group decryption and recovers polynomials $u_j(x)$ and $U_j(x)$ such that $\left(\frac{u_j(x)}{U_j(x)}\right)_{[-2nk,-1]} = (F(x) \mod q_j)_{[k-1,(2n+1)k-2]} \cdot x^{-(2n+1)k+1}$ and $\gcd(u_j(x), U_j(x)) = 1$ in $\mathbb{Z}_{q_j}[x]$ from these values. Thereafter, each participant extracts all roots of $U_j(x)$ over $\mathbb{Z}_{q_j}$ for each j and recovers all elements based on the criteria of our representation.

4.2 Analysis

Security Analysis. Now, we consider the correctness and privacy of our proposed protocol described in Fig. 4. The following theorems guarantee the correctness

Input: There are $n \geq 2$ HBC participants $\mathcal{P}_i$ with a private input set $S_i \subseteq \mathcal{U}$ of cardinality k. Set $d = nk$. The participants share the secret key sk, to which pk is the corresponding public key to the threshold NS AHE scheme. Let $\iota : \{0,1\}^* \to \mathbb{Z}_\sigma$ be the encoding function provided in Section 3.

Each participant $\mathcal{P}_i$, $i = 1, \ldots, n$:

1. (a) constructs the polynomial $f_{S_i}(x) = \prod_{s_{i,j} \in S_i}(x - \iota(s_{i,j})) \in \mathbb{Z}_\sigma[x]$, runs $\mathsf{RationalToRLS}(1, f_{S_i}, (2n+1)k-1)$ to obtain $\left(\frac{1}{f_{S_i}(x)}\right)_{[-(2n+1)k+1,-k]}$, and computes $F_{S_i}(x) = \left(\frac{1}{f_{S_i}(x)}\right)_{[-(2n+1)k+1,-k]} \cdot x^{(2n+1)k-1}$.
   (b) computes $\tilde{F}_{S_i}$, the encrypted polynomial of F_{S_i}, and sends $\tilde{F}_{S_i}$ to all other participants.
2. (a) chooses random polynomials $r_{i,j}(x) \in \mathbb{Z}_\sigma[x]$ of degree at most k for all $1 \leq j \leq n$.
   (b) computes the encryption, $\tilde{\phi}_i$, of the polynomial $\phi_i(x) = \sum_{j=1}^n F_{S_j} \cdot r_{i,j}$ and sends it to all participants.
3. (a) calculates the encryption of the polynomial $F(x) = \sum_{i=1}^n \phi_i(x)$.
   (b) performs a group decryption with all other players to obtain the $2nk$ higher-order terms of $F(x)$.
4. (a) recovers a polynomial pair of $u_j(x)$ and $U_j(x)$ in $\mathbb{Z}_{q_j}[x]$ for all $1 \leq j \leq \bar{\ell}$ such that $\left(\frac{u_j(x)}{U_j(x)}\right)_{[-2nk,-1]} = (F(x) \bmod q_j)_{[k-1,(2n+1)k-2]} \cdot x^{-(2n+1)k+1}$ and $\gcd(u_j(x), U_j(x)) = 1$ in $\mathbb{Z}_{q_j}[x]$, using the $2nk$ higher-order terms of $F(x)$ obtained in Step 3 (b).
   (b) extracts all roots of $U_j(x)$ in $\mathbb{Z}_{q_j}[x]$ for all j using a factorization algorithm.
   (c) determines the set union using the encoding rule of ι.

Fig. 4. PPSU-HBC protocol in the HBC case

and privacy of our construction in Fig. 4. We provide proofs of the following theorems in the full version of this paper [3].

Theorem 2. *In the protocol described in Fig. 4, every participant learns the set union of private inputs participating players, with high probability.*

Theorem 3. *Assume that the utilized additive homomorphic encryption scheme is semantically secure. Then, in our set union protocol for the HBC case described in Fig. 4, any adversary $\mathcal{A}$ of colluding fewer than n HBC participants learns no more information than would be gained by using the same private inputs in the ideal model with a trusted third party.*

Performance Analysis. It is clear that our protocol runs in $O(1)$ rounds. Let us count the computational and communicational costs for each participant.

Step 1 (a) requires $\tilde{O}(k)$ multiplications in $\mathbb{Z}_\sigma$ for a polynomial expansion of degree k and $O(kd)$ multiplications to run the $\mathsf{RationalToRLS}$ algorithm and compute F_{S_i}.

Step 1 (b) requires $O(d)$ exponentiations for $2d$ encryptions and $O(nd)$ communication costs.

Step 2 (b) requires $O(d^2)$ exponentiations for computing the encryption $\tilde{\phi}_i := \sum_{j=1}^{n} \tilde{F}_{S_j} \cdot r_{i,j}$ using additive homomorphic property and $O(nd)$ communication costs.

Step 3 (a) requires $O(nd)$ multiplications for computing $\sum_{i=1}^{n} \tilde{\phi}_i$.

Step 3 (b) requires $O(d)$ exponentiations for decryption share computation for $2d$ ciphertexts and $O(\bar{\ell}\sqrt{dq_j})$ multiplications for solving d DLPs for $\bar{\ell}$ groups of order q_j's.[1] The communication cost is $O(nd)$.

Step 4 (a) requires $O(d^2)$ multiplications in $\mathbb{Z}_{q_j}$ to recover $U_j(x)$ using extended Euclidean algorithm for each j.

Step 4 (b) requires $O(d^{1.5+o(1)})$ multiplications in $\mathbb{Z}_{q_j}$ for each j to factor a polynomial of degree d.

Step 4 (c) requires $O(\bar{\ell}d \log d \log q_j)$ bit operations for sorting and $O(d)$ hash computations.

Then the computational complexity is dominated by one of terms $O(d^2)$ exponentiations in Step 2 (b) and $O(\bar{\ell}\sqrt{dq_j})$ multiplications in Step 3 (b). Since one modular exponentiation for a modulus N requires $O(\log N)$ multiplications and $\bar{\ell} < \min\left\{d, \frac{\lfloor \log N \rfloor - 2}{3 \log\log N}\right\}$, the computational complexity for each participant is dominated by $O(d^2) = O(n^2k^2)$ exponentiations in $\mathbb{Z}_N$ and the total complexity is $O(n^3k^2)$ exponentiations in $\mathbb{Z}_N$. The total communication cost for our protocol is $O(n^2d) = O(n^3k)$ $(\log N)$-bit elements.

For the malicious case, we can also obtain the set union protocol using the techniques in [12,19]. We omit the details about our set union protocol for malicious case due to the space limitation. See the full version [3] of this paper.

5 Conclusion

In this paper, we provided a new representation of a set by a polynomial over $\mathbb{Z}_\sigma$, which can be efficiently inverted by finding all the linear factors of a polynomial whose root locates in the image of our encoding function, when the factorization of σ is public. Then we presented an efficient constant-round set union protocols, transforming our representation into a rational function and then combining it with threshold NS AHE scheme.

We showed that our encoding function is quite efficient on average-case, but it still requires exponential time in the degree of a polynomial to recover a set from the polynomial represented by our encoding function at worst-case although the probability of the worst-case is sufficiently small. Hence it would be interesting to construct an encoding function that enables us to recover a set in polynomial time even at worst-case.

[1] Note that one has to solve $\bar{\ell}$ DLPs over a group of order q_j for one decryption in the NS encryption scheme. In Step 3 (b), one has to solve $2d = 2nk$ DLPs over a group of order q_j for each q_j. It requires $O(\sqrt{dq_j})$ multiplications to solve d DLPs over a group of order q_j [13] and hence total complexity of this step is $O(\bar{\ell}\sqrt{dq_j})$ multiplications.

Acknowledgements. We thank Jae Hong Seo for helpful comments on our preliminary works and anonymous reviewers for their valuable comments. This work was supported by the IT R&D program of MSIP/KEIT. [No. 10047212, Development of homomorphic encryption supporting arithmetics on ciphertexts of size less than 1kB and its applications].

A Proof of Theorem 1

Let E_j be the expected number of linkable pairs of j-tuple in $\mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_j}$ for $j \geq 2$. For $1 \leq j \leq j' \leq \bar{\ell}$, let $\mathsf{S}_{j'-j+1}(i_j, \ldots, i_{j'})$ be the event that $(s_j^{(i_j)}, \ldots, s_{j'}^{(i_{j'})})$ is a linkable pair. Then,

$$\begin{aligned}\mathsf{E}_2 &= \sum_{i_1, i_2 \in \{1,\ldots,d\}} 1 \cdot \Pr[\mathsf{S}_2(i_1, i_2)] \\ &= \sum_{i_1, i_2 \in \{1,\ldots,d\}} \Pr[\mathsf{S}_2(i_1, i_2) \wedge (i_1 = i_2)] + \sum_{i_1, i_2 \in \{1,\ldots,d\}} \Pr[\mathsf{S}_2(i_1, i_2) \wedge (i_1 \neq i_2)] \\ &= d + d(d-1)\frac{1}{2^{2\tau}} = d\left(1 + \frac{d-1}{2^{2\tau}}\right)\end{aligned}$$

since $\Pr[\mathsf{S}_2(i_1, i_1)] = 1$ for $i_1 \in \{1, \ldots, d\}$ and $\Pr[\mathsf{S}_2(i_1, i_2)] = \frac{1}{2^{2\tau}}$ for distinct $i_1, i_2 \in \{1, \ldots, d\}$ from the Eq. (1).

Now, we consider the relation between E_j and E_{j+1}. When $(s_1^{(i_1)}, \ldots, s_j^{(i_j)})$ is a linkable pair, consider the case that $(s_1^{(i_1)}, \ldots, s_j^{(i_j)}, s_{j+1}^{(i_{j+1})})$ is a linkable pair. One can classify this case into the following three cases:

1. $i_{j+1} = i_j$,
2. $(i_{j+1} \neq i_j) \ \wedge \ (i_{j+1} = i_{j-1})$,
3. $(i_{j+1} \neq i_j) \ \wedge \ (i_{j+1} \neq i_{j-1})$.

At the first case, if $i_{j+1} = i_j$ and $(s_1^{(i_1)}, \ldots, s_j^{(i_j)})$ is a linkable pair, then $(s_1^{(i_1)}, \ldots, s_j^{(i_j)}, s_{j+1}^{(i_{j+1})})$ is always a linkable pair. Hence,

$$\begin{aligned}\mathsf{E}_{j+1}^{(1)} &:= \sum_{i_1, \ldots, i_{j+1}} \Pr\left[\mathsf{S}_{j+1}(i_1, \ldots, i_j, i_{j+1}) \wedge (i_{j+1} = i_j)\right] \\ &= \sum_{i_1, \ldots, i_j} \Pr\left[\mathsf{S}_j(i_1, \ldots, i_j)\right] = \mathsf{E}_j.\end{aligned}$$

At the second case, if $i_{j+1} = i_{j-1} \neq i_j$ and $(s_1^{(i_1)}, \ldots, s_j^{(i_j)})$ is a linkable pair, then the relation $s_{i_{j-1}, j+1} = s_{i_j, j+1} = s_{i_{j+1}, j+1}$ is satisfied from the encoding rule of ι. Hence,[2]

[2] Due to the space limitation, the detailed computation of Eqs. (5) and (6) are given in the full version of this paper [3].

$$\begin{aligned}\mathsf{E}^{(2)}_{j+1} &:= \sum_{i_1,\ldots,i_{j+1}\in\{1,\ldots,d\}} \Pr[\mathsf{S}_{j+1}(i_1,\ldots,i_j,i_{j+1}) \wedge (i_{j+1} = i_{j-1} \neq i_j)] \\ &\leq \frac{1}{2^\tau} \sum_{i_1,\ldots,i_j\in\{1,\ldots,d\}} \Pr\left[\mathsf{S}_j(i_1,\ldots,i_j)\right] = \frac{1}{2^\tau}\mathsf{E}_j. \end{aligned} \tag{5}$$

At the last case, we can obtain the following result:

$$\begin{aligned}\mathsf{E}^{(3)}_{j+1} &:= \sum_{i_1,\ldots,i_{j+1}\in\{1,\ldots,d\}} \Pr[\mathsf{S}_{j+1}(i_1,\ldots,i_j,i_{j+1}) \wedge ((i_{j+1} \neq i_j) \ \wedge \ (i_{j+1} \neq i_{j-1}))] \\ &\leq \frac{d-1}{2^{2\tau}} \sum_{i_1,\ldots,i_j\in\{1,\ldots,d\}} \Pr\left[\mathsf{S}_j(i_1,\ldots,i_j)\right] = \frac{d-1}{2^{2\tau}}\mathsf{E}_j. \end{aligned} \tag{6}$$

From the above results, we obtain the recurrence formula of E_j as follows:

$$\mathsf{E}_{j+1} = \mathsf{E}^{(1)}_{j+1} + \mathsf{E}^{(2)}_{j+1} + \mathsf{E}^{(3)}_{j+1} \leq \left(1 + \frac{1}{2^\tau} + \frac{d-1}{2^{2\tau}}\right)\mathsf{E}_j$$

for $j \geq 2$ and hence $\mathsf{E}_{\bar{\ell}} \leq d\left(1 + \frac{1}{2^\tau} + \frac{d-1}{2^{2\tau}}\right)^{\bar{\ell}-1}$ since $\mathsf{E}_2 = d\left(1 + \frac{d-1}{2^{2\tau}}\right) \leq d\left(1 + \frac{1}{2^\tau} + \frac{d-1}{2^{2\tau}}\right)$.

Now, we show that $\bar{\ell} \leq \frac{2^{2\tau}}{2^\tau+d}$. From the parameter setting, it is satisfied that $\bar{\ell} \leq \min\{d, \frac{\lfloor\log N\rfloor-2}{3\tau}\}$. When $d_0 \geq 8d$, it holds

$$\min\left\{d, \frac{\lfloor\log N\rfloor-2}{3\tau}\right\} \leq d \leq \frac{d_0^{1/3}d^{2/3}}{2}.$$

Consider the case that $d_0 < 8d$. Then, it also holds

$$\min\left\{d, \frac{\lfloor\log N\rfloor-2}{3\tau}\right\} \leq \frac{\lfloor\log N\rfloor-2}{3\tau} \leq \frac{d_0}{3\tau} \leq \frac{d_0^{1/3}d^{2/3}}{2}$$

since $\tau \geq 3$. Hence

$$\bar{\ell} \leq \min\left\{d, \frac{\lfloor\log N\rfloor-2}{3\tau}\right\} \leq \frac{d_0^{1/3}d^{2/3}}{2} \leq \frac{\left(d_0^2 d\right)^{2/3}}{2d_0} \leq \frac{2^{2\tau}}{2^\tau+d}$$

since $2d_0 > 2^\tau + d$. Therefore we obtain the following result:

$$\mathsf{E}_{\bar{\ell}} \leq d\left(1 + \frac{1}{2^\tau} + \frac{d-1}{2^{2\tau}}\right)^{\bar{\ell}-1} < ed < 3d,$$

where $e \approx 2.718$ is the base of the natural logarithm. In other words, the upper bound of the expected number of linkable pairs of $\bar{\ell}$-tuple is $3d$. □

References

1. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: Simon, J. (ed.) ACM Symposium on Theory of Computing (STOC), pp. 1–10. ACM (1988)
2. Camenisch, J., Shoup, V.: Practical verifiable encryption and decryption of discrete logarithms. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 126–144. Springer, Heidelberg (2003)
3. Cheon, J.H., Hong, H., Lee, H.T.: Invertible polynomial representation for set operations. Cryptology ePrint Archive, Report 2012/526 (2012). http://eprint.iacr.org/2012/526
4. De Cristofaro, E., Kim, J., Tsudik, G.: Linear-complexity private set intersection protocols secure in malicious model. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 213–231. Springer, Heidelberg (2010)
5. De Cristofaro, E., Tsudik, G.: Practical private set intersection protocols with linear complexity. In: Sion, R. (ed.) FC 2010. LNCS, vol. 6052, pp. 143–159. Springer, Heidelberg (2010)
6. Fouque, P.-A., Poupard, G., Stern, J.: Sharing decryption in the context of voting or lotteries. In: Frankel, Y. (ed.) FC 2000. LNCS, vol. 1962, pp. 90–104. Springer, Heidelberg (2001)
7. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient private matching and set intersection. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 1–19. Springer, Heidelberg (2004)
8. Frikken, K.B.: Privacy-preserving set union. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 237–252. Springer, Heidelberg (2007)
9. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Aho, A.V. (ed.) ACM Symposium on Theory of Computing (STOC), pp. 218–229. ACM (1987)
10. Hong, J., Kim, J.W., Kim, J., Park, K., Cheon, J.H.: Constant-round privacy preserving multiset union. Bull. Korean Math. Soc. **50**(6), 1799–1816 (2013)
11. Jarecki, S., Liu, X.: Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 577–594. Springer, Heidelberg (2009)
12. Kissner, L., Song, D.: Privacy-preserving set operations. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 241–257. Springer, Heidelberg (2005)
13. Kuhn, F., Struik, R.: Random walks revisited: extensions of pollard's rho algorithm for computing multiple discrete logarithms. In: Vaudenay, S., Youssef, A.M. (eds.) SAC 2001. LNCS, vol. 2259, pp. 212–229. Springer, Heidelberg (2001)
14. Lee, H.T.: Polynomial Factorization and Its Applications. Ph.D. thesis, Seoul National University, February 2013
15. Naccache, D., Stern, J.: A new public key cryptosystem based on higher residues. In: Gong, L., Reiter, M.K. (eds.) ACM Conference on Computer and Communications Security (ACM CCS), pp. 59–66. ACM (1998)
16. Okamoto, T., Uchiyama, S.: A new public-key cryptosystem as secure as factoring. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 308–318. Springer, Heidelberg (1998)
17. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)

18. Sang, Y., Shen, H.: Efficient and secure protocols for privacy-preserving set operations. ACM Trans. Inf. Syst. Secur. **13**(1), 9:1–9:35 (2009)
19. Seo, J.H., Cheon, J.H., Katz, J.: Constant-round multi-party private set union using reversed Laurent series. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 398–412. Springer, Heidelberg (2012)
20. Shamir, A.: On the generation of multivariate polynomials which are hard to factor. In: Kosaraju, S.R., Johnson, D.S., Aggarwal, A. (eds.) ACM Symposium on Theory of Computing (STOC), pp. 796–804. ACM (1993)
21. Shoup, V.: Practical threshold signatures. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 207–220. Springer, Heidelberg (2000)
22. Umans, C.: Fast polynomial factorization and modular composition in small characteristic. In: Dwork, C. (ed.) ACM Symposium on Theory of Computing (STOC), pp. 481–490. ACM (2008)