# Cryptanalysis of An Oblivious Polynomial Evaluation Protocol Based On Polynomial Reconstruction Problem*

Huang Lin, Zhenfu Cao

Department of Computer Science and Engineering,

Shanghai Jiao Tong University, No. 800, Dongchuan Road, Shanghai, 200240, P. R. China

Email: faustlin@sjtu.edu.cn, zfcao@cs.sjtu.edu.cn

**Abstract**

In 1999, Naor and Pinkas [1] presented a useful protocol called oblivious polynomial evaluation(OPE). In this paper, the cryptanalysis of the OPE protocol is presented. It's shown that the receiver can successfully get the sender's secret polynomial $P$ after executing the OPE protocol only once, which means the privacy of the sender can be violated and the security of the OPE protocol will be broken. It's also proven that the complexity of the cryptanalysis is the same with the corresponding protocols cryptanalyzed.

**Keywords:** oblivious polynomial evaluation; polynomial reconstruction problem; cryptanalysis

## 1 Introduction

In 1999, Naor and Pinkas [1] presented a useful protocol called oblivious polynomial evaluation(OPE). It's a protocol with two parties. A sender whose input is a polynomial $P$ and a receiver whose input is a value $\alpha$. They are both in the field $F$. At the end of this protocol the receiver learns $P(\alpha)$ and the sender learns nothing. In 2006, Naor and Pinkas [2] published a full paper discussing this protocol.

In this paper, it will be pointed out that the receivers could learn the polynomial $P$ through learning $P(\alpha_1), P(\alpha_2), \cdots, P(\alpha_{d_P+1})$ ($d_P$ is the degree of the polynomial P) after executing the OPE protocol once. This means the privacy of the sender will be violated and the security of the OPE protocol could be broken. A detailed description of the cryptanalysis will also be given in this paper.

The rest of this paper will be divided into these sections: at first a brief review of the OPE protocol will be given, and then the weakness of the OPE protocol will be discussed. After that, we will talk about the primitive idea of our cryptanalysis. Two attacks against the generic OPE protocol and the OPE protocol secure against the malicious receivers are given respectively later. The properties of these two attacks including the correctness of these attacks will be proven in

*This work is supported in part by the National NaturalScience Foundation of China under Grant Nos. 60225007, 60572155 and 60673079, the National Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20060248008.

the later sections. Some applications based on the OPE protocol which could be attacked by the attacks presented in this paper will be pointed out and the suggestion which could prevent these application from these attacks will also be given in the remark section. At last the conclusion of this paper will be given.

## 2 A review of the OPE protocols

The functionality of OPE protocol [2] between a receiver and a sender over a field $F$ is as follow:

**Input**

- Receiver: an input $\alpha \in F$.
- Sender: A polynomial $P$ defined over $F$.

**Output**

- Receiver: $P(\alpha)$.
- Sender: nothing.

The requirements of a private OPE protocol can be divided into *correctness*, *receiverprivacy*, *serverprivacy*. The definition of these requirements are as follow:

**Definition 1** ***(Correctness, or Functionality)*** *At the end of the protocol the receiver obtains the output of the OPE functionality, namely $P(\alpha)$.*

**Definition 2** ***(Receiver's privacy - indistinguishability)*** *For any probabilistic polynomial time $B'$ executing the sender's part, for any x and x' in $F$, the views that $B'$ sees in case the receiver's input is x and in case the receiver's input is x' are computationally indistinguishable.*

**Definition 3** ***(Sender's privacy - comparison with the ideal model)*** *For every probabilistic polynomial-time $A'$ substituting the receiver, there exists a probabilistic polynomial-time machine $A''$ that plays the receiver's role in the ideal implementation, such that the view of $A'$ and the output of $A''$ are computationally indistinguishable.*

**Definition 4** ***(Private OPE protocol)*** *A two-party protocol satisfying the above three definitions.*

The whole OPE protocol is constructed based on Polynomial reconstruction problem, the polynomial reconstruction problem is defined as follow:

**Definition 5** ***(Polynomial reconstruction problem (PR problem))*** *INPUT: Integers $k$ and $t$, and $n$ points $\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i, y_i \in F$.*

*OUTPUT: Any univariate polynomial $P$ of degree at most $k$ such that $P(x_i) = y_i$ for at least $t$ values $i \in [1, n]$.*

Another polynomial list reconstruction problem really close to PR problem can be formulated as follow:

**Definition 6** ***(Polynomial list reconstruction problem)*** *INPUT: Integers $k$ and $t$, and $n$ points $\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i, y_i \in F$.*

*OUTPUT: All univariate polynomial $P$ of degree at most $k$ such that $P(x_i) = y_i$ for at least $t$ values $i \in [1, n]$.*

There are two intractability assumptions related to polynomial reconstruction problem[2], and we only introduce the first assumption here since it's the only one to be used.

The intractability assumption depends on the following parameters:

- $F$, the field over which the polynomial is defined.
- $k$, the degree of the hidden polynomial.
- $n$, the number of correct valued of the polynomial, which is also the number of queries made in the OPE protocol. (This parameter corresponds to "t" in the definition of the polynomial reconstruction problem.)
- $m$, the expansion ratio (namely, the ratio between the total number of points and $n$). (This parameter corresponds to $n/t$ in the definition of the polynomial reconstruction problem.)

The first intractability assumption assumes that given an input to the polynomial list reconstruction problem with all $x_i$ being distinct, the value of the polynomial at $x = 0$ is pseudo-random. [2] uses $A_{n,m}^{k,\alpha}$ to denote the probability distribution of sets generated in the following way:

- Pick a random polynomial $P$ over $F$, of degree at most $k$, for which it holds that $P(0) = \alpha$.
- Generate $nm$ random values $x_1, x_2, \cdots, x_{nm}$ in $F$ subject to the constraint that all $x_i$ values are distinct and different from 0.
- Choose a random subset $S$ of $n$ different indices in $[1, nm]$, and set $y_i = P(x_i)$ for all $i \in S$. For every $i \notin S$ set $y_i$ to be a random value in $F$.
- 4. Output the set $\{x_i, y_i\}_{i=1}^{nm}$.

The pseudo-randomness assumption is defined based on the notion of computationally indistinguishability as follow:

**Definition 7** ***(First pseudo-randomness assumption)*** *Let $l$ be a security parameter, and let $n(l)$, $m(l)$, $k(l)$, $F(l)$ be polynomially bounded functions that define the parameters $n$, $m$, $k$ and the size in bits of the representation of an element in the field $F$. Let $A_{n,m}^{k,\alpha}$ and $A_{n,m}^{k,\alpha'}$ be random variables that are chosen according to the distributions $A_{n,m}^{k,\alpha}$ and $A_{n,m}^{k,\alpha'}$ respectively. Then it holds that for every $\alpha, \alpha' \in F$ the probability ensembles $A_{n,m}^{k,\alpha}$ and $A_{n,m}^{k,\alpha'}$ are computationally indistinguishable for adversaries whose running time is polynomial in the security parameter $l$.*

There are mainly two OPE protocols given in [2]: A generic OPE protocol and an OPE protocol secure against malicious receivers.

The generic OPE protocol is as follow:

**A generic protocol for OPE**

- **Step 1. The sender hides $P$ of degree $d_P$ in a bivariate polynomial:** The sender generates a random masking polynomial $Z(x)$ of degree $d$, s.t., $Z_(0) = 0$. Namely $Z(x) = \sum_{i=1}^{d} a_i x^i$. The parameter $d$ equals the product of the degree of $P$ and the security parameter $k$. The sender then defines a bivariate polynomial $Q(x, y) = Z(x) + P(y) = \sum_{i=1}^{d} a_i x^i + \sum_{i=0}^{d_P} b_i y^i$ for which it holds that $Q(0, y) = P(y)$.

- **Step 2. The receiver hides a secret $\alpha$ in a univariate polynomials:** *The receiver chooses a random polynomial $S$ of degree $k$, such that $S(0) = \alpha$. The receiver's plan is to use the univariate polynomials $R(x) = Q[x, S(x)]$ to learn $P(\alpha)$: it holds that $R(0) = Q[0, S(0)] = P(S(0)) = P(\alpha)$ and, therefore, if the receiver is able to interpolate $R$ she can learn $R(0) = P(\alpha)$. The degree of $R$ is $d_R = k * d_P$.*

- **Step 3. The receiver learns points of $R$:** *The receiver learns $d_R + 1$ values of the form $< x_i, R(x_i) >$.*

- **Step 4. The receiver computes $P(\alpha)$:** *The receiver uses the values of $R$ that it learned to interpolate $R(0) = P(\alpha)$.*

The OPE protocol secure against malicious receivers is as follow:

**OPE protocol secure against malicious receivers** The sender's input is still $P$ and the receiver's input is still $\alpha$.

- *Step 1. The sender generates the $d_P$ linear polynomial $P_1, \cdots, P_{d_P}$ that are used for reducing the OPE of the polynomial $P$ to $d_P$ OPEs of linear polynomials, by the method of Lemma 3.4 in [2].*

- *Step 2. The parties execute $d_p$ instances of OPE in which the receiver evaluates the linear polynomials $P_1, P_2, \cdots, P_{d_P}$ at the point $\alpha$, under the following constraints:*

  - *The sender generates independent masking polynomials $Z_i(x)$, $1 \leq i \leq d_P$, and consequently the resulting bivariate polynomials $Q_i(x, y) = Z_i(x) + P_i(y)$, one for each of the $d_P$ OPEs. (step 1 of the above protocol).*
  - *The receiver generates a single polynomial $S$ for use in all the OPEs (step 2 of the above protocol). This step defines $d_P$ polynomials $R_1(x) = Q_1(x, S(x)), \cdots, R_{d_P}(x) = Q_{d_P}(x, S(x))$, such that for each $i$ it holds that $R_i(0) = P_i(\alpha)$.*
  - *The receiver learns $d_R + 1$ tuples of the form $(x_j, R_1(x_j), \cdots, R_{d_P}(x_j))$. These values enable it to interpolate $P_1(\alpha), \cdots, P_{d_P}(\alpha)$ (step3 and step4 of the above protocol). The implementation of this step is done by executing the same number of oblivious transfers as is required for a single OPE of a linear polynomial. In each OPE the sender sends to the receiver $d_P$ values of the polynomials, namely $(x_j, R_1(x_j), \cdots, R_{d_P}(x_j))$, instead of a single value $x_j, R(x_j)$.*

- *Step 3. The receiver uses $P_1(\alpha), \cdots, P_{d_P}(\alpha)$ to compute $P(\alpha)$ by the method of Lemma 3.4 in [2].*

## 3 The weakness of the OPE protocol

After the concept of oblivious polynomial evaluation has been presented, there were lots of work done by the cryptography community. These work mainly focused on the security of the OPE protocol. When Naor and Pinkas firstly presented the concept of OPE protocol [1], they use a new intractability assumption named "noisy polynomial interpolation problem" to build their protocol. However, Bleichenbacher and Nguyen [3] later showed that this problem can be transformed into a lattice shortest vector problem with high probability, which means this problem is much easier than expected. If the assumption used to build the OPE protocol isn't strong enough then we can't guarantee the privacy of the receiver since the privacy of the receiver depends on the underlying intractability assumption. The adversary (the third-party adversary or the malicious sender) could easily get the secret $\alpha$ hold by the receiver through solving the noisy polynomial interpolation problem. This attack has been considered in the recent version of OPE protocol [2]. When different assumptions related to polynomial reconstruction are used to build the OPE protocol, this kind of attack will not work successfully against it.

Although the security definition which considers both the privacy of the receiver and the privacy of the sender has been given in the papers [2][1], the work done later put their emphasis on the privacy of the receiver. The privacy of the sender faces the challenge of the third-party adversary and the dishonest receiver. In [2], the dishonest receiver is classified into two categories: semi-honest receiver and malicious receiver. Semi-honest receiver's operation is assumed to follow the behavior that it should take according to the protocol, but it also may try to deduce more information from the data it learns in the execution of the protocol. Malicious receiver can behave arbitrarily. This paper will focus on the attacks launched by the receiver. We will prove that the attacks launched by the receiver or the adversary who pretends to be the receiver can successfully break the security of the two OPE protocols, one of which is supposed to be secure against the malicious receiver by cheating the sender of his secret polynomial $P$.

## 4 The primitive of cryptanalysis of the OPE protocol

The basic idea of OPE protocol is to hide the receiver's secret $\alpha$ in a randomly chosen polynomial $S$ of degree $k$ by letting the equation $S(0) = \alpha$ hold. The privacy of the receiver is realized based on two intractability assumptions, which are both closely related to the Noisy Polynomial Reconstruction Problem [2]. In these two assumptions, $N$ random values $x_1, x_2, \cdots, x_N$ are generated at first, and then the polynomial $P$ is represented as $n$ different points $\{(x_i, y_i)\}_{i=j_1}^{j_n}$ which satisfy $y_i = S(x_i)$, and here $\{j_1, j_2, \cdots, j_n\}$ is a random set of $n$ indices in $[1, N]$. The polynomial then will be covered by the rest of the $N$ values by choosing their $y_i$ randomly which satisfies $y_i \neq S(x_i)$. We describe the basic idea in Fig.1.

For ease of exposition, we choose $k = 3, N = 8, S(x) = x^3 - 4 * x^2 + x + 8$, $n = 4$, and we have $\alpha = S(0) = 8$ since $S(0) = 8$. We randomly choose $\{(-1, 2), (2, 2), (4, 12), (5, 38)\}$ in the curve of this polynomial to represent $S(x)$, i.e. to represent the secret $\alpha = S(0)$ hold by the receiver. These four points will be covered by the other four points $\{(-4, -20), (1, -40), (7, -5), (9, -20)\}$ which are randomly chosen in the field $F$. However, these points can do more. If we randomly choose three of these four points such as $\{(-1, 2), (2, 2), (4, 12)\}$, then we can find a polynomial $S_1(x) = x^2 - x$ which fits these three points satisfying $S_1(0) = 0$. This means that we can use these points to represent another $\alpha_1$ which is equal to 0. If we use the other three points $\{(4, 12), (2, 2), (5, 38)\}$ we can get

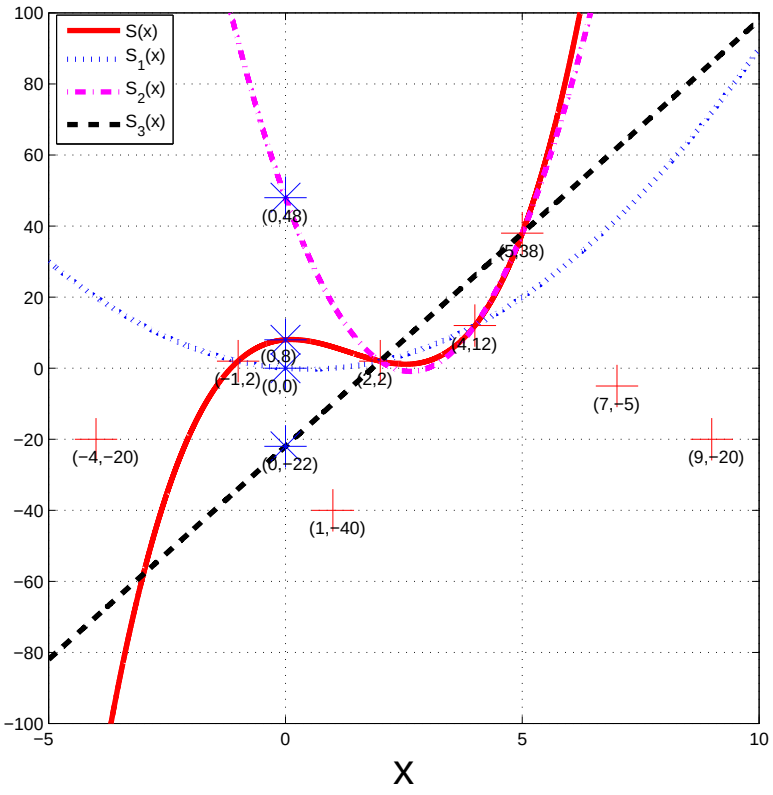


Figure 1: The polynomials hide secrets more than just one

another polynomial $S_2(x) = 7x^2 - 37x + 48$ which can represents another $\alpha_2 = S_2(0) = 48$. For there are $\binom{4}{3} = 4$ combinations of three points in these four points, we will have 4 polynomials to represent 4 secrets $\alpha_1, \alpha_2, \alpha_3, \alpha_4$. When the number of points used to represent a secret goes down, for example if the two of these four points are used to construct the polynomial, then the degree of the polynomial to represent $\alpha_i$ will go down too, but still we can easily get the polynomial satisfying $S_i(0) = \alpha_i$ in the same way mentioned above. For example, in Fig.1. if we use $\{(5, 38), (2, 2)\}$ to represent a secret, we will have a polynomial $S_3(x) = 12x - 22$ to represent the secret $\alpha = S_3(0) = -22$. That means if we use the two of these four points to represent a secret we can hide $\binom{4}{2} = 6$ secrets. To sum up, if the receiver does not care the number of points to represent the secrets, he will manage to hide $\binom{n}{n-1} + \binom{n}{n-2} + \cdots + \binom{n}{2} = 2^n - n - 2$ secrets in the set of $n$ points which are far more than to hide just one secret. This is exactly the primitive idea which will lead to our cryptanalysis. Although in [2] the OPE protocol were given, the authors obviously neglected the contraction between the definition of malicious receiver and what can be done by this kind of receiver. Since the malicious receiver could act arbitrarily, it's very likely for him to choose several 'secrets' which he doesn't care about in order to cheat the sender of his secret polynomial $P$. This also means the receiver does not need to consider his own privacy at all. Since it's the malicious receiver who is supposed to choose the $n$ points by himself, then he can choose these points so that they don't just represent one secret but several subsets of them can represent more 'secrets' than just one in the way mentioned above. That means the $N$ points sending from the receiver to the sender could hide the secret $S_1(0) = \alpha_1, S_2(0) = \alpha_2, \cdots, S_{d_P+1}(0) = \alpha_{d_P+1}$ rather than a single value $\alpha$ which it's supposed to

be. From the sender the receiver can get $\{Q[x_{i_j}, S_j(x_{i_j})]\}_{i_j \in I_j, j=1,2,\cdots,d_P+1}$, where $I_1, I_2, \cdots, I_{d_P+1}$ is the corresponding subsets which represent secrets $\alpha_1, \alpha_2, \cdots, \alpha_{d_P+1}$ in the way mentioned above. For the receiver, he is able to learn the polynomial $\{R_j(x) = Q[x, S_j(x)]\}_{j=1,2,\cdots,d_P+1}$ by using the corresponding points set $\{Q[x_{i_j}, S_j(x_{i_j})]\}_{i_j \in I_j, j=1,2,\cdots,d_P+1}$ to interpolate $\{R_j(x)\}_{j=1,2,\cdots,d_P+1}$. The receiver can easily learn $P(\alpha_1), P(\alpha_2), \cdots, P(\alpha_{d_P+1})$ Because $\{R_j(0) = Q(0, S_j(0)) = P(S_j(0)) = P(\alpha_j)\}_{j=1,2,\cdots,d_P+1}$ [2]. At last, the receiver could use these points $\{\alpha_j, P(\alpha_j)\}_{j=1}^{d_P+1}$ to interpolate the sender's polynomial $P$ of degree $d_P + 1$.

## 5 An attack on the generic protocol for OPE

As mentioned above, with $n$ points the receiver could actually hide $2^n - n - 2$ secrets which is far more than $d_P + 1$, hence, there are lots of ways for the receiver to choose the degree of polynomials $S_1(x), S_2(x), \cdots, S_{d_P+1}(x)$ and to choose the set of $n$ points in order to represent $d_P + 1$ different secrets. In this paper, for ease of exposition we let the degree of the polynomials be the same with $k - l$. In the description of this attack below the detailed way to partition the set will be given. Theorem 1 will show the requirement for $l$ in this situation. We also follow the notation of [2] here for ease of exposition.

---

**An attack on the generic protocol for OPE**

- **Step 1. The receiver hides secrets $\alpha_1, \alpha_2, \cdots, \alpha_{d_P+1}$ in several univariate polynomials:** *The receiver chooses $d_P + 1$ polynomials $S_1(x), S_2(x), \cdots, S_{d_P+1}(x)$ of degree $k - l$, such that*

$$\begin{array}{lcl} S_1(0) & = & \alpha_1, \\ S_2(0) & = & \alpha_2, \\ & \cdots, & \\ S_{d_P+1}(0) & = & \alpha_{d_P+1} \end{array}$$

  *and $\alpha_1, \alpha_2, \cdots, \alpha_{d_P+1}$ are randomly chosen to make sure that they are different from each other. The receiver's plan is to use the univariate polynomials $\{R_j(x) = Q[x, S_j(x)]\}_{j=1}^{d_P+1}$ to learn $\{P(\alpha_j)\}_{j=1}^{d_P+1}$: it holds that $R_j(0) = Q[0, S_j(0)] = P(S_j(0)) = P(\alpha_j)$ (see the second section) and, therefore, if the receiver is able to interpolate $R_j(x)$ she can learn $R_j(0) = P(\alpha_j)$. The degree of $R_j(x)$ is $d_{R_j(x)} = (k - l) * d_P$.*

- **Step 2. The receiver learns points of $\{R_j(x)\}_{j=1,2,\cdots,d_P+1}$:** *The receiver learns $[(k - l)d_p + 1] * (d_P + 1)$ values of the form $\{R_j(x_{i_j})\}_{i_j \in I_j, j=1,2,\cdots,d_P+1}$.*

- **Step 3. The receiver computes $P(x)$:** *The receiver uses the values of $\{R_j(x)\}_{j=1,2,\cdots,d_P+1}$ that it learned to interpolate $\{R_j(0) = P(\alpha_j)\}_{j=1,2,\cdots,d_P+1}$. Then the receiver can use $\{P(\alpha_j)\}_{j=1}^{d_P+1}$ to interpolate the polynomial $P(x)$.*

---

In [2], two detailed protocols based on two different assumptions were given. Since the pseudo-randomness assumption can be chosen by the receiver and pseudo-randomness assumption 2 is stronger than pseudo-randomness assumption 1 (the second problem is easier than the first one), we give a description of an attack on OPE based on assumption 1 for the receiver (In here, the receiver just uses the randomness assumption 1 to hide his secrets instead of following the assumption

exactly since the goal of him is to get the sender's polynomial and he doesn't care about his secrets actually).

**An attack on OPE based on assumption 1** *The attack is the attack on the generic protocol, where the second step is run as follows:*

- *The receiver sets* $n = d_R + 1 = d + 1 = kd_P + 1$ *(*$k$ *is the degree of the polynomial which is supposed to be used in the honest way to hide just one secret.) and choose* $N = nm$ *distinct random values* $x_1, \cdots, x_N \in F$*, all different from each other and 0.*

- *The receiver chooses a random set* $T$ *of* $n$ *indices* $1 \leq t_1, t_2, \cdots, t_n \leq N$*. The set* $T$ *will be partitioned into several subsets* $I_1, I_2, \cdots, I_{d_P+1}$ *satisfying* $|I_1| = |I_2| = \cdots = |I_{d_P+1}| = (k-l)d_P + 1$ *and another subset* $I_{d_P+2}$ *satisfying* $|I_{d_P+2}| = ld_P - d_P - kd_P^2 + ld_P^2$*. She then defines* $N$ *values* $y_i$*, for* $1 \leq i \leq N$*. The value* $y_i = S_j(x_i)$ *if* $i \in I_j, 1 \leq j \leq d_P + 1$*, and a random value if* $i$ *is not in any of the first* $d_P + 1$ *subsets.*

- *The receiver sends the* $N$ *points* $\{x_i, y_i\}_{i=1}^N$ *to the sender.*

- *The receiver and sender execute an* $n-out-of-N$ *oblivious transfer protocol, for the* $N$ *values* $Q(x_1, y_1), \cdots, Q(x_N, y_N)$*. The receiver will choose the corresponding* $Q(x_i, y_i)$ *for the indices* $i \in I_1, I_2, \cdots, I_{d_P+1}$*(since* $n > |I_1| + |I_2| + \cdots + |I_{d_P+1}|$*, the receiver could choose what he wants after the OT protocol), and then he will have* $\{Q[x_{i_j}, S_j(x_{i_j})]\}_{i_j \in I_j, j=1,2,\cdots,d_P+1}$*. Since in step 1* $\{R_j(x) = Q[x, S_j(x)]\}_{j=1}^{d_P+1}$*, That means the receiver has* $\{R_j(x_{i_j})\}_{i_j \in I_j, j=1,2,\cdots,d_P+1}$*.*

The realization of $n - out - of - N$ oblivious transfer protocol can be found in [1]. Since the way to use oblivious transfer protocol here isn't essentially different from that of [2], the receiver can get what he wants from the sender.

Theorem 2 in the latter section will show this attack will be successful, and the complexity of this attack will also be given in theorem 3.

# 6 An attack on the OPE protocol secure against malicious receivers

In [2], an OPE protocol secure against malicious receivers was given. This protocol was claimed to be secure against malicious receivers. In this section an attack against this protocol is given, and we'll show that this attack will successfully attack this protocol.

The attack on OPE protocol secure against malicious receivers is not significantly different from the attack on the generic protocol.

**An attack on OPE protocol secure against malicious receivers**

- *Step 1. The receiver hides secrets* $\alpha_1, \alpha_2, \cdots, \alpha_{d_P+1}$ *in several univariate polynomials: The receiver chooses* $d_P + 1$ *polynomials* $S_1(x), S_2(x), \cdots, S_{d_P+1}(x)$ *of degree* $k - l$*(Theorem 4 will show the requirement for* $l$*), such that* $S_1(0) = \alpha_1, S_2(0) = \alpha_2, \cdots, S_{d_P+1}(0) = \alpha_{d_P+1}$ *and* $\alpha_1, \alpha_2, \cdots, \alpha_{d_P+1}$ *are randomly chosen to make sure that they are different from each other. The receiver's plan is to use the univariate polynomials*

$$\begin{aligned} R_{j1}(x) &= Q_1(x, S_j(x))_{j=1}^{d_P+1}, \\ R_{j2}(x) &= Q_2(x, S_j(x))_{j=1}^{d_P+1}, \\ &\cdots \\ R_{jd_P}(x) &= Q_{d_P}(x, S_j(x))_{j=1}^{d_P+1} \end{aligned}$$

*to learn* $\{P_1(\alpha_j), P_2(\alpha_j), \cdots, P_{d_P}(\alpha_j)\}_{j=1}^{d_P+1}$ *since it holds that*

$$\begin{aligned} R_{j1}(0) &= P_1(\alpha_j)_{j=1}^{d_P+1}, \\ R_{j2}(0) &= P_2(\alpha_j)_{j=1}^{d_P+1}, \\ &\cdots \\ R_{jd_P}(0) &= P_{d_P}(\alpha_j)_{j=1}^{d_P+1} \end{aligned}$$

(see the second section).

*The degree of* $\{[R_{jm}(x)]|_{m=1}^{d_P}\}|_{j=1}^{d_P+1}$ *is* $(k-l)$ *since the polynomials* $P_1, P_2, \cdots, P_{d_P}$ *are all linear.*

- *Step 2. The receiver learns points of* $\{[R_{jm}(x)]|_{m=1}^{d_P}\}|_{j=1}^{d_P+1}$*: The receiver learns* $[(k-l)+1] * (d_P+1)$ *tuples of the form* $\{x_{i_j}, R_{j1}(x_{i_j}), R_{j2}(x_{i_j}), \cdots, R_{jd_P}(x_{i_j})\}_{i_j \in I_j, j=1,2,\cdots,d_P+1}$.

- *Step 3. The receiver computes* $P(x)$*: The receiver uses the values* $\{x_{i_j}, R_{j1}(x_{i_j}), R_{j2}(x_{i_j}), \cdots, R_{jd_P}(x_{i_j})\}_{i_j \in I_j, j=1,2,\cdots,d_P+1}$ *that it learned to interpolate* $\{[R_{jm}(x)]|_{m=1}^{d_P}\}|_{j=1}^{d_P+1}$*. Then the receiver can use* $\{P_1(\alpha_j), P_2(\alpha_j), \cdots, P_{d_P}(\alpha_j)\}_{j=1}^{d_P+1}$ *to compute* $\{P(\alpha_j)\}_{j=1}^{d_P+1}$ *by the method of Lemma 3.4 in [2]. The receiver can use* $\{P(\alpha_j)\}_{j=1}^{d_P+1}$ *to interpolate the polynomial* $P(x)$.

The realization of the second step is based on pseudo-randomness assumption 1. This attack is as follow:

**An attack on the OPE protocol secure against malicious receivers based on assumption 1** *The second step of this attack is run as follows:*

- *The receiver sets* $n = d_R + 1 = d + 1 = kd_P + 1$ *(*$k$ *is the degree of the polynomial which is supposed to be used in the honest way to hide just one secret.) and choose* $N = nm$ *distinct random values* $x_1, \cdots, x_N \in F$*, all different from each other and 0.*

- *The receiver chooses a random set* $T$ *of* $n$ *indices* $1 \leq t_1, t_2, \cdots, t_n \leq N$*. The set* $T$ *will be partitioned into several subsets* $I_1, I_2, \cdots, I_{d_P+1}$ *satisfying* $|I_1| = |I_2| = \cdots = |I_{d_P+1}| = (k-l)+1$ *and another subset* $I_{d_P+2}$ *satisfying* $|I_{d_P+2}| = ld_P - d_P - k + l$*. It then defines* $N$ *values* $y_i$*, for* $1 \leq i \leq N$*. The value* $y_i = S_j(x_i)$ *if* $i \in I_j, 1 \leq j \leq d_P + 1$*, and a random value if* $i$ *is not in any of the first* $d_P + 1$ *subsets.*

- *The receiver sends the* $N$ *points* $\{x_i, y_i\}_{i=1}^N$ *to the sender.*

- *The receiver and sender execute an* $n - out - of - N$ *oblivious transfer protocol, for the* $N$ *tuples* $\{Q_1(x_i, y_i), Q_2(x_i, y_i), \cdots, Q_{d_P}(x_i, y_i)\}_{i=1}^N$*. The receiver will choose the corresponding* $Q_1(x_i, y_i), Q_2(x_i, y_i), \cdots, Q_{d_P}(x_i, y_i)$ *for the indices* $i \in I_1, I_2, \cdots, I_{d_P+1}$*(since* $n > |I_1| + |I_2| + \cdots + |I_{d_P+1}|$*, the receiver could choose what he wants after the OT protocol), and then it will*

*have* $\{Q_1[x_{i_j}, S_j(x_{i_j})], Q_2[x_{i_j}, S_j(x_{i_j})], \cdots, Q_{d_P}[x_{i_j}, S_j(x_{i_j})]\}_{\ i_j \in I_j, j=1,2,\cdots,d_P+1}$. *Since it hold that*

$$\begin{aligned} R_{j1}(x) &= Q_1(x, S_j(x))_{j=1}^{d_P+1}, \\ R_{j2}(x) &= Q_2(x, S_j(x))_{j=1}^{d_P+1}, \\ &\cdots \\ R_{jd_P}(x) &= Q_{d_P}(x, S_j(x))_{j=1}^{d_P+1} \end{aligned}$$

*in step 1 of this attack. This means the receiver has* $\{[R_{jm}(x_{i_j})]|_{i_j \in I_j, j=1,2,\cdots,d_P+1}\}|_{m=1}^{d_P}$.

Theorem 5 in the latter section will show this attack will be successful, and the complexity of this attack will also be given in Theorem 6.

# 7 The properties of these attacks

**Theorem 1** *The requirement for $l$ to satisfy the requirement for the partition of the random set $T$ of $n$ indices is $\frac{kd_P+1}{d_P+1} \leq l < k$.*

**Proof 1** *Since the whole set will be partitioned into $d_P + 1$ subsets $I_1, I_2, \cdots, I_{d_P+1}$ satisfying $|I_1| = |I_2| = \cdots = |I_{d_P+1}| = (k-l)d_P + 1$ and the remainder subset $I_{d_P+2}$. It should be hold that*

$$(d_P + 1) \times [(k - l)d_P + 1] \leq kd_P + 1$$

*, and since we choose the degree of our polynomials as $k - l$, we have*

$$k - l > 0$$

*. Combine these two inequations together we have $\frac{kd_P+1}{d_P+1} \leq l < k$.*

**Theorem 2** *The attack on the generic OPE protocol based on pseudo-randomness assumption 1 will succeed.*

**Proof 2** *From the description of the attack we have $\{R_j(x_{i_j})\}_{i_j \in I_j, j=1,2,\cdots,d_P+1}$. Since these $x$'s are all chosen to be different from each other and from 0 and $|I_1| = |I_2| = \cdots = |I_{d_P+1}| = (k-l)d_P+1$, $R_j(x)$ of degree $d_{R_j(x)} = (k - l) * d_P$ can be easily computed. Since we have $R_j(0) = P(\alpha_j)$, then the receiver can have $\{P(\alpha_j)\}_{j=1}^{d_P+1}$ in step 3 to interpolate $P(x)$ of degree $d_P$. So the attack is successful.*

**Theorem 3** *The complexity of the attack against the generic OPE protocol based on pseudo-randomness assumption 1 is running a single invocation of $kd_P + 1 - out - of - (kd_P + 1)m$ oblivious transfer, i.e., $n - out - of - N]$ oblivious transfer.*

**Proof 3** *As we can see from the description of this attack, the whole execution of this attack is actually an execution of the generic OPE protocol except that the receiver needs to partition his indices of $x$ into several subsets and he needs to interpolate $P(x)$ after the execution of the protocol. However, if we measure the overhead of the attack in terms of the oblivious transfer stage as in [2] we can see that the overhead of this attack is the same with the OPE protocol based on Assumption 1. The complexity of this attack is running a single invocation of $kd_P + 1 - out - of - (kd_P + 1)m$ oblivious transfer.*

**Theorem 4** *The requirement for $l$ to satisfy the requirement for the partition of the random set $T$ of $n$ indices for the attack on the OPE secure against malicious receiver is $\frac{kd_P+1}{d_P+1} \leq l < k$.*

**Proof 4** *Since the whole set will be partitioned into $d_P + 1$ subsets $I_1, I_2, \cdots, I_{d_P+1}$ satisfying $|I_1| = |I_2| = \cdots = |I_{d_P+1}| = (k-l)+1$ and the remainder subset $I_{d_P+2}$. Then is should be hold that*

$$(d_P + 1) \times [(k - l) + 1] \leq kd_P + 1$$

*, and since we choose the degree of our polynomials as $k - l$, we have*

$$k - l > 0$$

*. Combine these two inequations together we have $\frac{k+d_P}{d_P+1} \leq l < k$.*

**Theorem 5** *The attack on the OPE protocol against malicious receiver based on pseudo-randomness assumption 1 will succeed.*

**Proof 5** *From the description of the attack we have $\{[R_{jm}(x_{i_j})]|_{i_j \in I_j, j=1,2,\cdots,d_P+1}\}|_{m=1}^{d_P}$. Since these $x$'s are all chosen to be different from each other and from 0 and $|I_1| = |I_2| = \cdots = |I_{d_P+1}| = (k-l)+1$, $\{[R_{jm}(x)]|_{m=1}^{d_P}\}|_{j=1}^{d_P+1}$ of degree $k - l$ can be easily computed. Then following step 3 in this attack the receiver will finally get the polynomial $P(x)$. So the attack is successful.*

**Theorem 6** *The complexity of the attack on the OPE protocol against malicious receiver based on pseudo-randomness assumption 1 is running a single invocation of $k + 1 - out - of - (k + 1)m$ oblivious transfer.*

**Proof 6** *The way to calculate the complexity of this attack is the same with the way to calculate the complexity of the attack on the generic OPE protocol based on pseudo-randomness assumption 1. It's straightforward to conclude that the complexity of this attack is $k + 1 - out - of - (k + 1)m$ since the complexity of this attack is the same with the OPE protocol against malicious receiver in terms of oblivious transfer and the complexity of OPE protocol against malicious receiver is the same with an OPE protocol of a linear polynomial.*

## 8 Remark

Chang and Lu [4] presented an oblivious polynomial evaluation protocol only based on oblivious transfer. The attacks presented in this paper won't work successfully on Chang and Lu's protocol. There are several applications of OPE protocol such as [8][7][6][3] . Since their protocols are all based on Naor and Pinkas's OPE protocol[2], then the attacks of this paper all work successfully on their protocols. To make these attacks ineffective, they only need to choose some other OPE protocols not based on polynomial reconstruction problem such as Chang and Lu's protocol[4].

## 9 Conclusion

In this paper, we present two attacks on the OPE protocols and prove these attacks will make the receiver successfully achieve the sender's secret polynomial $P$, which also make the OPE protocol based on polynomial reconstruction problem not a private OPE protocol any more by the definition 4. We also show that the complexity of these attacks is the same with that of the corresponding protocols attacked.

## References


[1] M. Naor and B. Pinkas, Oblivious Transfer and Polynomial Evaluation, In Proc. 31st Annual ACM Symposium on Theory of Computing, Atlanta, Georgia, pp. 245–254 May 1999.

[2] M. Naor and B. Pinkas, Oblivious Polynomial Evaluation, SIAM J. Comput. 35(5): 1254-1281, 2006.

[3] D. Bleichenbacher and P. Nguyen, Noisy polynomial interpolation and noisy chinese remaindering, In Proc.19th Eurocrypt'2000, Bruges, Belgium. Springer-Verlag, LNCS 1807, pp.53-69, 2000.

[4] Y. C. Chang and C. J. Lu, Oblivious polynomial evaluation and oblivious neural learning, In AsiaCrypt: Advances in Cryptology. LNCS, Springer-Verlag, 2001. 4.

[5] Y. Lindell and B. Pinkas, Privacy preserving data mining, In:Advances in Cryptology-Crypto.Berlin: Springer-Verlag, pp.36-54, 2000.

[6] W. Ogata and K. Kurosawa, Oblivious keyword search, J. Complexity 20(2-3): 356-371, 2004.

[7] S. Jha, L. Kruger and P. McDaniel, Privacy Preserving Clustering, In Proc. of the 10th ESORICS, 2005.

[8] N Gilboa, Two party rsa key generation In Advances in Cryptology (CRYPTO99), Santa Barbara, California, USA, pp.15-19, August 1999.