Verifiable and Private Oblivious Polynomial Evaluation

Manik Lal Das, Hardik Gajera, David Gérault, Matthieu Giraud and Pascal Lafourcade[*]

Abstract. It is a challenging problem to delegate the computation of a polynomial on encrypted data to a server in an oblivious and verifiable way. In this paper, we formally define *Verifiable and Private Oblivious Polynomial Evaluation* (VPOPE) scheme. We design a scheme called Verifiable IND-CFA Paillier based Private Oblivious Polynomial Evaluation (VIP-POPE). Using security properties of Private Polynomial Evaluation (PPE) schemes and Oblivious Polynomial Evaluation (OPE) schemes, we prove that our scheme is *proof unforgeability*, *indistinguishability against chosen function attack*, and *client privacy*-secure under the Decisional Composite Residuosity assumption in the random oracle model.

Keywords: Delegation of computation · Verifiable computation · Oblivious evaluation · Privacy.

1 Introduction

From harmless smart gardening [19] to critical applications such as forest fire detection [17], data monitoring through sensors is becoming pervasive. In particular, sensors for monitoring health-related data are more and more widely adopted, be it through smartwatches that track the heart rate, or sensors implemented in the patient's body [2]. This medical data can sometimes be used to assess the health status of an individual, by applying a single variable polynomial prediction function on it [7]. However, when it comes to medical data, extreme care must be taken in order to avoid any leakage. Recently, the leak of medical data of 1.5 million SingHealth users in Singapore strongly incentivized to improve the security and privacy surrounding medical data [1]. In this context, we consider the following problem:

How can a company use medical data recorded by clients to give them predictions about their health status in a private way?

For instance, this company may collect Fitbit data from its customers, and use it to predict things such as a risk factor for certain diseases. For economic reasons, this company keeps the polynomial secret: it invested time to build it,

[*] This research was conducted with the support of the FEDER program of 2014-2020, the region council of Auvergne-Rhône-Alpes.

and required to collect lots of data. Its economic model is based on the secrecy of the polynomial: the clients pay the company to obtain the polynomial's output on their medical data. If the polynomial was public, then the clients would directly compute it, and the company would cease to exist. However, as the company grows, it becomes difficult to treat all the computation requests, so that the company needs to delegate this computation to a cloud service. The company trusts the cloud service provider and gives the secret polynomial; however, the clients may not trust the server to produce correct results, so that the company would like the server to be able to prove the correctness of each prediction to the client, i.e., prove that its output is correct with regards to the secret prediction function.

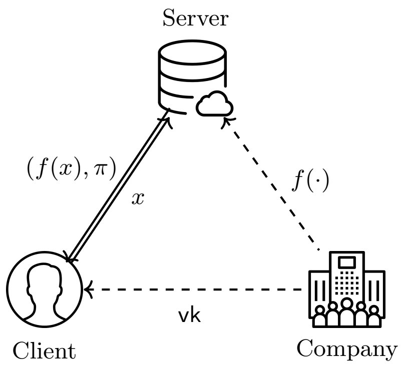

Fig. 1: Illustration of a PPE scheme.

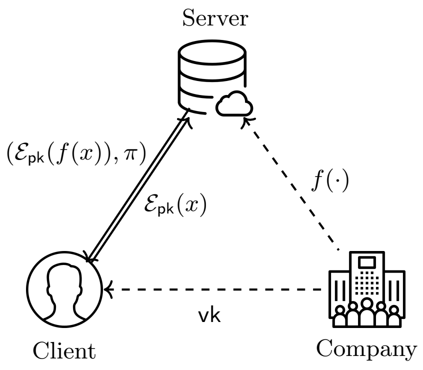
Server
$(\mathcal{E}_{\mathsf{pk}}(f(x)),\pi)$
$\mathcal{E}_{\mathsf{pk}}(x)$
$f(\cdot)$
Client
vk
Company

Fig. 2: Illustration of a VPOPE scheme.

In this scenario, the problem is how to delegate computations on a secret polynomial function to an external server in a verifiable way. This problem is solved by *Private Polynomial Evaluation* (PPE) schemes [14, 12, 4, 25] illustrated in Fig. 1. In a PPE scheme, the company outsources the secret polynomial function $f(\cdot)$ to an external server. Moreover, the company provides some public information vk called *verification key*. This verification key is used with the proof π generated by the server during the delegated computation of $f(x)$ to allow clients to verify the correctness of the result returned by the server.

However, PPE schemes do not protect the privacy of the clients: their data is handled in clear by the server. After the SingHealth hack, the company wants to be sure that even if an intruder hacks the server, he will not be able to steal the medical data of its clients. To solve this problem, we propose a new primitive called *Verifiable and Private Oblivious Polynomial Evaluation* (VPOPE). A VPOPE scheme is a private polynomial evaluation scheme, in which the data of the client cannot be read by the cloud server. More precisely, the client sends his encrypted data to the server, and the server never learns anything about x. We illustrate this new primitive in Fig. 2.

1.1 Related Works

VPOPE schemes are related to several research domains. The first one is the *Verifiable Computation* (VC) introduced by Gennaro *et al.* [13]. The aim of VC is to delegate a costly computation to an untrusted third party. This third party returns the result of the computation and a proof of correctness, which is easier to verify. Primitives where everyone can check the correctness of the computation are said to be *publicly verifiable* [23]. VC has given rise to a bunch of protocols [5, 9, 6, 22, 21]. Although VC is related to our paper; the difference is that in these works, the polynomial used by the server is not secret.

Another similar primitive is *Oblivious Polynomial Evaluation* (OPE) introduced by Naor and Pinkas [18]. OPE protocols are constituted of two parties. The first party, A, knows a secret function $f(\cdot)$ and the other one, B, has a secret element x. The aim of OPE is that B receives $f(x)$ in such a way A learns nothing about the value x sent by B, and that B learns nothing about the function $f(\cdot)$. OPE are used to solve different cryptographic problems as set membership, oblivious keyword search, and set intersection [16, 11, 10]. Although OPE and VPOPE are very similar; their difference lies in the fact that OPE do not consider the verifiability of the computation of $f(x)$, whereas it is a crucial point in VPOPE since the client does not trust the server.

Finally, the nature of VPOPE is very close to those of *Private Polynomial Evaluation* (PPE). To the best of our knowledge, only five papers [15, 14, 12, 4, 25] propose to hide a polynomial used by the server and allow a client to verify the returned results. Kate *et al.* [15] formally define a primitive called *commitments to polynomials* that can be used as a PPE scheme and propose the $\mathsf{PolyCommit}_{\mathsf{Ped}}$ scheme. In this primitive, the committer publishes some points (x, y) of the secret polynomial together with a proof that $y = f(x)$. Then, she can open the commitment *a posteriori* to reveal the secret polynomial. This is primitive is close to PPE and VPOPE schemes since the verification key used in PPE and VPOPE can be viewed as a commitment. However, this verification key is computed by a trusted party (the company) and computations are performed by an untrusted party (the server). Although the verification cost is in constant-time, it uses three pairing computations, and we show that, in practice, the verification cost of our VPOPE scheme is more efficient (see Section 5.2).

Independently of Kate *et al.* [15], Guo *et al.* [14] propose a scheme with similar security properties to delegate the computation of a secret health-related function on the users' health record. The polynomials are explicitly assumed to have low coefficients and degree, which significantly reduces their randomness. However, the authors give neither security models nor proofs. Later, Gajera *et al.* [12] show that any user can guess the polynomial using the Lagrange's interpolation on several points. They propose a scheme where the degree k is hidden and claim that it does not suffer from this kind of attack.

Following this work, Bultel *et al.* [4] show that hiding the degree k is useless and that no scheme can be secure when user query more than k points to the server. Moreover, they give a cryptanalysis of Guo *et al.* [14] PPE scheme and of Gajera *et al.* [12] PPE scheme which requires only one query to the server and present the first security model for PPE schemes. A PPE scheme must satisfy the following properties: (i) *proof unforgeability* (UNF) requires that the server cannot provide a valid proof to the client for a point that is not a point of the secret polynomial; (ii) *indistinguishability against chosen function attack* ($\mathsf{IND\text{-}CFA}$) requires that the client cannot distinguish which of two polynomials of her choice has been evaluated by the server. Bultel *et al.*show that $\mathsf{PolyCommit}_{\mathsf{Ped}}$ scheme from Kate *et al.* [15] satisfies these security properties. Moreover, Bultel *et al.*design a PPE scheme called PIPE that is $\mathsf{IND\text{-}CFA}$ secure and solves an open problem described by Kate *et al.*concerning the design of a scheme with a weaker assumption than t-SDH. Despite having the additional property that it protects the privacy of the client, we show that the verification of our VPOPE scheme is more efficient than for PIPE.

More recently, Xia *et al.* [25] proposed a new efficient PPE scheme. As PIPE, their scheme satisfies the required security properties defined in [4]. Their scheme is based on the Pedersen's Verifiable Secret Sharing [24] and does not depend on NIZKP to allow the client to verify the correctness of the result contrary to Bultel *et al.*[4]. In addition to have computational advantages over previous PPE schemes, Xia *et al.*'s scheme relies only on the *Discrete Logarithm* assumption. However, the verification cost of Xia *et al.*'s scheme also requires k exponentiations where k is the degree of the secret polynomial, which makes it costlier than our scheme that needs only three exponentiations, one Paillier decryption, and k multiplications.

1.2 Contributions

The contributions of this paper are summarized as follows:

- We formally define the VPOPE schemes and give security framework based on those of PPE and *Oblivious Polynomial Evaluation* (OPE) schemes.
- We design VIP-POPE (for *Verifiable $\mathsf{IND\text{-}CFA}$ Paillier based Private Oblivious Polynomial Evaluation*), an efficient and secure VPOPE scheme. This scheme uses homomorphic properties of Paillier's encryption scheme [20] in order to achieve encrypted polynomial evaluation.
- We also formally prove its security in the random oracle model and compare its efficiency for the verification cost with the existing PPE schemes. We show that VIP-POPE is more efficient for the verification part than PPE schemes presented in [15, 4, 25].

1.3 Outline

In the next section, we recall the cryptographic notions used in this paper. In Section 3, we give the PPE and OPE security model for VPOPE schemes. Then, we present in Section 4, our VPOPE scheme called VIP-POPE. Before to conclude, we prove in Section 5 that VIP-POPE satisfies the security properties for VPOPE schemes and compare its verification cost with other PPE schemes of the literature.

2 Preliminaries

We start by recalling the definition of the cryptographic tools used in this paper. In the rest of the paper, we denote by $\textsc{poly}(\eta)$ the set of probabilistic polynomial time algorithms with respect to the security parameter η.

2.1 Paillier Cryptosystem

We now recall the generation, the encryption and decryption algorithms of the Paillier's public key encryption scheme [20] used in our scheme.
Key Generation. We denote by $\mathbb{Z}_n$, the ring of integers modulo n and by $\mathbb{Z}_n^\star$ the set of invertible elements of $\mathbb{Z}_n$. The public key pk of Paillier's encryption scheme is (n, g), where $g \in \mathbb{Z}_{n^2}^\star$ and $n = pq$ is the product of two prime numbers.

The corresponding secret key sk is (λ, μ), where λ is the least common multiple of $p-1$ and $q-1$ and $\mu = (L(g^\lambda \mod n^2))^{-1} \mod n$, where $L(x) = \frac{x-1}{n}$.
Encryption Algorithm. Let m be a message such that $m \in \mathbb{Z}_n$. Let r be a random element of $\mathbb{Z}_n^\star$. We denote by $\mathcal{E}_{\mathsf{pk}}$ the encryption algorithm that produces the ciphertext c from a given plaintext m with the public key $\mathsf{pk} = (n, g)$ as follows: $c = \mathcal{E}_{\mathsf{pk}}(m) = g^m r^n \mod n^2$.
Decryption Algorithm. Let c be the ciphertext such that $c \in \mathbb{Z}_{n^2}$. We denote by $\mathcal{D}_{\mathsf{sk}}$ the decryption function of the plaintext c with the secret key $\mathsf{sk} = (\lambda, \mu)$ defined as follows: $m = \mathcal{D}_{\mathsf{sk}}(c) = L\left(c^\lambda \mod n^2\right) \cdot \mu \mod n$.

Paillier's cryptosystem is a partial homomorphic encryption scheme. Let m_1 and m_2 be two plaintexts in $\mathbb{Z}_n$. The product of the two associated ciphertexts with the public key $\mathsf{pk} = (n, g)$, denoted $c_1 = \mathcal{E}_{\mathsf{pk}}(m_1) = g^{m_1} r_1^n \mod n^2$ and $c_2 = \mathcal{E}_{\mathsf{pk}}(m_2) = g^{m_2} r_2^n \mod n^2$, is the encryption of the sum of m_1 and m_2. We also remark that: $\mathcal{E}_{\mathsf{pk}}(m_1) \cdot \mathcal{E}_{\mathsf{pk}}(m_2)^{-1} = \mathcal{E}_{\mathsf{pk}}(m_1 - m_2)$ and $\mathcal{E}_{\mathsf{pk}}(m_1)^{m_2} = \mathcal{E}_{\mathsf{pk}}(m_1 m_2)$.

Theorem 1. *Paillier's cryptosystem is IND-CPA-secure if and only if the Decisional Composite Residuosity Assumption holds [20].*

To present our scheme, we first claim the following property on Paillier ciphertexts.

Property 1. Let n be the product of two prime numbers, $x \in \mathbb{Z}_n$, and $g \in \mathbb{Z}_{n^2}^\star$. We set $\mathsf{pk} = (n, g)$ a Paillier public key. Let $\{t_i\}_{i=1}^k$ such that for all $i \in \{1, \dots, k\}$, we have $t_i = t_{i-1}^x \cdot r_i^n$ with $t_0 = g$, and $r_i \in \mathbb{Z}_{n^2}^\star$. Then for all $i \in \{1, \dots, k\}$, $t_i = \mathcal{E}_{\mathsf{pk}}(x^i)$.

2.2 Zero-Knowledge Proof

We use the ZKP given by Baudron *et al.* [3] to prove the plaintexts equality of $k \in \mathbb{N}$ Paillier ciphertexts. Let $\mathbb{Z}_{n^2}^\star$ be a multiplicative group where n is the product of two prime numbers p and q. The language is the set of all statements $(t_1, \dots, t_k) \in (\mathbb{Z}_{n^2}^\star)^k$ for $k \in \mathbb{Z}_{\geq 2}$ such that for all $i \in \{1, \dots, k\}$, $t_i = t_{i-1}^x \cdot r_i^n \mod n^2$ where $t_0 \in \mathbb{Z}_{n^2}^\star$ and $r_i \in \mathbb{Z}_{n^2}^\star$.

Since the ZKP given by Baudron *et al.* [3] is a sigma protocol, we can use the *Fiat-Shamir Transformation* [8] to obtain a NIZKP. We formally define this NIZKP called $\mathsf{DecPaillierEq}$.

Definition 1 ($\mathsf{DecPaillierEq}$ [3])**.** *Let n be the product of two prime numbers p and q and H be a hash function, $\mathcal{L}$ be the set of all $(t_1, \dots, t_k) \in (\mathbb{Z}_{n^2}^\star)^k$ such that for all $i \in \{1, \dots, k\}$, $t_i = t_{i-1}^x \cdot r_i^n \mod n^2$ where $t_0 \in \mathbb{Z}_{n^2}^\star$ and $r_i \in \mathbb{Z}_{n^2}^\star$. We define the NIZKP* $\mathsf{DecPaillierEq} = (\mathsf{Prove}, \mathsf{Verify})$ *for $\mathcal{L}$ as follow:*

– $\mathsf{Prove}((t_1,\ldots,t_k),\omega)$: *Using the witness* $\omega = (x, t_0, \{r_i\}_{i=1}^k)$, *it picks* $\rho \xleftarrow{\$} [0, 2^{\log(n)}]$ *and* $s_i \in \mathbb{Z}_n^*$ *for* $1 \le i \le k$, *and computes* $u_i = t_{i-1}^{\rho} \cdot s_i^n \mod n^2$ *for* $1 \le i \le k$. *Moreover, it computes* $w = \rho + x \cdot H(t)$ *and sets* $v_i = s_i \cdot r_i^{H(t)} \mod n$ *for* $1 \le i \le k$. *Finally, it outputs* $\pi_t = (w, \{u_i\}_{i=1}^k, \{v_i\}_{i=1}^k)$.
– $\mathsf{Verify}((t_1,\ldots,t_k),\pi_t)$: *Using* $\pi_t = (w, \{u_i\}_{i=1}^k, \{v_i\}_{i=1}^k)$, *it verifies if* $w \in [0, 2^{\log(n)}]$, *and if* $t_{i-1}^{w} \cdot v_i^n = u_i \cdot t_i^{H(t)} \mod n^2$ *for* $1 \le i \le k$. *Then it outputs* 1, *else* 0.

Moreover, Baudron *et al.* [3] prove the following theorem.

Theorem 2. DecPaillierEq *is unconditionally complete, sound and zero-knowledge in the random oracle model.*

3 Definition and Security Model

Before we present our security model, we first formally define a Private Oblivious Polynomial Evaluation scheme.

Definition 2. *A* Verifiable and Private Oblivious Polynomial Evaluation *(VPOPE) scheme is composed of eight algorithms* (setup, init, keyGen, queryGen, queryDec, compute, decrypt, verif) *defined as follows:*

– $\mathsf{setup}(\eta)$: *Using the security parameter* η, *this algorithm generates a ring* F, *public parameters* pub *and secret parameters* sec. *It returns* $(\mathsf{pub}, F, \mathsf{sec})$.
– $\mathsf{init}(F, f, \mathsf{sec})$: *Using* F, *the secret polynomial* f, *and parameters* sec, *this algorithm returns a verification key* vk *and a server key* sk_f *associated to the secret polynomial* f.
– $\mathsf{keyGen}(\eta, \mathsf{pub}, k)$: *Using the security parameter* η *and public parameters* pub, *this algorithm generates and returns a client's key pair* $(\mathsf{pk}_c, \mathsf{sk}_c)$.
– $\mathsf{queryGen}(\mathsf{pk}_c, x)$: *Using a public key* pk_c *and an input* x, *this algorithm generates an encrypted query* t *associated to* x, *a proof* π_t *proving that* t *is a valid encrypted query, and returns* (t, π_t).
– $\mathsf{queryDec}(\mathsf{sk}_c, t)$: *Using a secret key* sk_c *and an encrypted request* t, *this algorithm outputs* x *if* t *is a valid request of* x, $\perp$ *otherwise.*
– $\mathsf{compute}(t, \pi_t, f, \mathsf{sk}_f, F)$: *Using* t, π_t, f, sk_f, *and* F, *this algorithm returns an encrypted value* d *along with a proof* π_d *proving that* d *is an encryption of* $f(x)$ *if the proof* π_t *is "accepted". Else it returns* $\perp$.
– $\mathsf{decrypt}(\mathsf{sk}_c, d)$: *Using a secret key* sk_c *and the encrypted value* d, *this algorithm returns* y, *the decryption of* d.
– $\mathsf{verif}(x, \mathsf{sk}_c, \mathsf{pub}, y, \pi_d, \mathsf{vk})$: *This algorithm returns* 1 *if the proof* π_d *is "accepted",* 0 *otherwise.*

3.1 Security Models

We use security notions of *PPE* schemes formalized by Bultel *et al.* [4], namely *Unforgeability* (UNF), and *Indistinguishability against Chosen Function Attack* (IND-CFA), and adapt them to VPOPE

```
$\mathbf{Exp}^{\mathsf{CPI}}_{\Pi,\mathcal{A}}(\eta)$:
$b \xleftarrow{\$} \{0,1\}$ ;
$(\mathsf{pub}, F, \mathsf{sec}) \leftarrow \mathsf{setup}(\eta)$ ;
$f \xleftarrow{\$} F[X]^k$ ;
$(\mathsf{vk}, \mathsf{sk}_f) \leftarrow \mathsf{init}(F, f, \mathsf{sec})$ ;
$(\mathsf{pk}_c, \mathsf{sk}_c) \leftarrow \mathsf{keyGen}(\eta, \mathsf{pub}, k)$ ;
$(x_0, x_1, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pk}_c, \mathsf{pub}, F)$ ;
$(t, \pi_t) \leftarrow \mathsf{queryGen}(\mathsf{pk}_c, x_b)$ ;
$b_* \leftarrow \mathcal{A}_2^{\mathsf{CO}_{\mathsf{CPI}}(\cdot)}(t, f, \mathsf{sk}_f, F, \mathsf{st})$ ;
return $(b = b_*)$ .
$\mathsf{CO}_{\mathsf{CPI}}(x)$:
$(t, \pi_t) \leftarrow \mathsf{queryGen}(\mathsf{pk}_c, x)$ ;
return $t$ .
```

Fig. 3: CPI experiment.

schemes. The security model IND-CFA ensure secrecy of the polynomial, the security model UNF ensures validity of the verification process. Since VPOPE schemes consider encrypted data on client side, we recall the *Client's Privacy - Indistinguishability* (CPI) security property defined by Naor and Pinkas [18] to include the privacy on data client. Moreover, we define the *Query Soundness* (QS) notion in order to prove that a client cannot have other information than points that she queried. In all the security models, we denote by $F[x]^k$, the set of all polynomials of degree k over a finite field F.

Client's Privacy - Indistinguishability

We first recall the *Client's Privacy - Indistinguishability* (CPI) security for VPOPE schemes introduced by Naor and Pinkas [18]. In this model the adversary chooses two queries (x_0, x_1) and tries to guess the evaluation x_b asked by the client. The adversary has access to the ciphertext oracle $\mathsf{CO}_{\mathsf{CPI}}(\cdot)$ taking x as input and returns the encrypted query t. A VPOPE scheme is CPI-secure if no adversary can output the query chosen by the client with a better probability than by guessing.

Definition 3 (Client's privacy - indistinguishability.). *Let Π be a VPOPE, $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2) \in \text{POLY}(\eta)^2$ be a two-party adversary. The* client's privacy - indistinguishability *(CPI) experiment for $\mathcal{A}$ against Π is defined in Fig. 3, where $\mathcal{A}$ has access to the oracle $\mathsf{CO}_{\mathsf{CPI}}(\cdot)$. The advantage of the adversary $\mathcal{A}$ against the CPI experiment is given by:*

$$\mathsf{Adv}^{\mathit{CPI}}_{\Pi,\mathcal{A}}(\eta) = \left| \frac{1}{2} - \Pr\left[1 \leftarrow \mathit{Exp}^{\mathit{CPI}}_{\Pi,\mathcal{A}}(\eta)\right] \right| .$$

A scheme Π is CPI-secure if this advantage is negligible for any $\mathcal{A} \in \text{POLY}(\eta)^2$.

Chosen Function Attack

We recall the model for *k-Indistinguishability against Chosen Function Attack* (k-IND-CFA). In this model, the adversary chooses two polynomials (f_0, f_1) and tries to guess the polynomial f_b used by the server, where $b \in \{0, 1\}$. The adversary has access to a server oracle $\mathsf{CO}_{\mathsf{CFA}}(\cdot)$ and sends to her an

```
$\mathbf{Exp}^{k\text{-}\mathbf{IND\text{-}CFA}}_{\Pi,\mathcal{A}}(\eta)$:
$b \xleftarrow{\$} \{0,1\}$ ;
$(\mathsf{pub}, F, \mathsf{sec}) \leftarrow \mathsf{setup}(\eta)$ ;
$(\mathsf{pk}_c, \mathsf{sk}_c) \leftarrow \mathsf{keyGen}(\eta, \mathsf{pub}, k)$ ;
$(f_0, f_1, \mathsf{st}_2) \leftarrow \mathcal{A}_1(\mathsf{pk}_c, \mathsf{pub}, F, k)$ ;
$c \leftarrow 0$ ;
$(\mathsf{vk}, \mathsf{sk}_f) \leftarrow \mathsf{init}(F, f_b, \mathsf{sec})$ ;
$b_* \leftarrow \mathcal{A}_2^{\mathsf{CO}_{\mathsf{CFA}}(\cdot)}(\mathsf{pk}_c, \mathsf{pub}, F, \mathsf{vk}, k, \mathsf{st})$ ;
if $f_0 \notin F[X]^k$ or $f_1 \notin F[X]^k$:
then return $\perp$ ;
else return $(b = b_*)$ .
$\mathsf{CO}_{\mathsf{CFA}}(t, \pi_t)$:
$(d, \pi_d) \leftarrow \mathsf{compute}(t, \pi_t, f_b, \mathsf{sk}_f, F)$ ;
if $x \leftarrow \mathsf{queryDec}(t, \mathsf{sk}_c)$ and $x \neq \perp$ and $f_0(x) = f_1(x)$:
then return $(d, \pi_d)$ ;
else return $\perp$ .
```

Fig. 4: IND-CFA experiment.

encrypted query t associated to her data x along with a proof π_t. The oracle decrypts the query t and obtains x if t is valid. If $f_0(x) = f_1(x)$, the oracle returns d i.e. the encrypted value of $f_b(x)$, along with a proof π_d. If $f_0(x) \neq f_1(x)$, then the server returns nothing. In practice, an adversary chooses (f_0, f_1) such that $f_0 \neq f_1$, but with k points (x_i, y_i) such that $f_0(x_i) = f_1(x_i)$. It allows the adversary to maximize his oracle calls in order to increase his chances of success.

Definition 4. (k-IND-CFA)**.** *Let Π be a VPOPE, $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2) \in \textsc{poly}(\eta)$ be a two-party adversary and k be an integer. The k-IND-CFA experiment for $\mathcal{A}$ against Π is defined in Fig. 4, where $\mathcal{A}$ has access to the* server oracle $\mathsf{CO}_{\mathit{CFA}}(\cdot)$*. The advantage of the adversary $\mathcal{A}$ against the k-IND-CFA experiment is given by:*

$$\mathsf{Adv}^{k\text{-}\mathit{IND}\text{-}\mathit{CFA}}_{\Pi,\mathcal{A}}(\eta) = \left| \frac{1}{2} - \Pr\left[1 \leftarrow \mathit{Exp}^{k\text{-}\mathit{IND}\text{-}\mathit{CFA}}_{\Pi,\mathcal{A}}(\eta)\right] \right| .$$

A scheme Π is k-IND-CFA-secure if this advantage is negligible for any $\mathcal{A} \in \textsc{poly}(\eta)^2$.

Query Soundness

We now define a model for *Query Soundness* (QS). In this model, the adversary tries to learn other information than points of the secret polynomial that she queried by sending a particular query t along with a proof π_t to the server.

```
$\mathbf{Exp}^{\mathsf{QS}}_{\Pi,\mathcal{A}}(\eta)$:
$(\mathsf{pub}, F, \mathsf{sec}) \leftarrow \mathsf{setup}(\eta)$ ;
$f \xleftarrow{\$} F[X]^k$ ;
$(\mathsf{vk}, \mathsf{sk}_f) \leftarrow \mathsf{init}(F, f, \mathsf{sec})$ ;
$(\mathsf{pk}_c, \mathsf{sk}_c) \leftarrow \mathsf{keyGen}(\eta, \mathsf{pub}, k)$ ;
$(t, \pi_t) \leftarrow \mathcal{A}((\mathsf{pk}_c, \mathsf{sk}_c), \mathsf{pub}, F)$ ;
if $\mathsf{queryDec}(t) \neq \bot$ and $\mathsf{compute}(t, \pi_t, f, \mathsf{sk}_f, F) \neq \bot$
    and $f(\mathsf{queryDec}(\mathsf{sk}_c, t)) \neq \mathsf{decrypt}(\mathsf{sk}_c, d)$ such that
    $(d, \pi_d) \leftarrow \mathsf{compute}(t, \pi_t, f, \mathsf{sk}_f, F)$:
then return 1 ;
else return 0 .
```

Fig. 5: QS experiment.

Definition 5 (Query Soundness). *Let Π be a VPOPE, and $\mathcal{A} \in \textsc{poly}(\eta)$ be an adversary. The* Query Soundness *(*QS*) experiment for $\mathcal{A}$ against Π is defined in Fig. 5. The advantage of the adversary $\mathcal{A}$ against the* QS *experiment is given by:*

$$\mathsf{Adv}^{\mathsf{QS}}_{\Pi,\mathcal{A}}(\eta) = \Pr\left[1 \leftarrow \mathit{Exp}^{\mathsf{QS}}_{\Pi,\mathcal{A}}(\eta)\right] .$$

A scheme Π is QS*-secure if this advantage is negligible for any $\mathcal{A} \in \textsc{poly}(\eta)$.*

Unforgeability

Finally, we recall the unforgeability property. A VPOPE is unforgeable when a dishonest server cannot produce a valid proof for a point (x, y) such that $y \neq f(x)$. In this model, the secret polynomial f is chosen by the server.

```
$\mathbf{Exp}^{\mathsf{UNF}}_{\Pi,\mathcal{A}}(\eta)$:
$(\mathsf{pub}, F, \mathsf{sec}) \leftarrow \mathsf{setup}(\eta)$ ;
$(f, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{sec})$ ;
$(\mathsf{vk}, \mathsf{sk}_f) \leftarrow \mathsf{init}(F, f, \mathsf{sec})$ ;
$(\mathsf{pk}_c, \mathsf{sk}_c) \leftarrow \mathsf{keyGen}(\eta, \mathsf{pub}, k)$ ;
$(x_*, y_*, \pi_*) \leftarrow \mathcal{A}_2(\mathsf{pub}, \mathsf{sk}_f, \mathsf{vk}, F, f, \mathsf{st})$ ;
if $f(x_*) \neq y_*$ and $\mathsf{verif}(x_*, \mathsf{sk}_c, \mathsf{pub}, y_*, \pi_*, \mathsf{vk}) = 1$:
then return 1 ;
else return 0 .
```

Fig. 6: UNF experiment.

Definition 6 (Unforgeability). *Let Π be a VPOPE, $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2) \in \text{POLY}(\eta)$ be a two-party adversary. The* unforgeability *(UNF) experiment for $\mathcal{A}$ against Π is defined in Fig. 6. We define the advantage of the adversary $\mathcal{A}$ against the UNF experiment by:*

$$\mathsf{Adv}^{\mathsf{UNF}}_{\Pi,\mathcal{A}}(\eta) = \Pr\left[1 \leftarrow \mathsf{Exp}^{\mathsf{UNF}}_{\Pi,\mathcal{A}}(\eta)\right] .$$

A scheme Π is UNF-secure if this advantage is negligible for any $\mathcal{A} \in \text{POLY}(\eta)^2$.

3.2 Security Against Collusion Attacks

There are two possible collusion scenarios: collusion of a client and the server, and collusion of two or more clients.

Scenario 1: In collusion of a client and the server, the server can provide the secret polynomial to the client. This is inherent problem and cannot be prevented. The client can share public parameters and verification keys with the server but these parameters are already public and known to the server. The collusion does not give any advantage to the server to forge fake proof of computation.

Scenario 2: In collusion of two or more clients, sharing Paillier secret key with each other does not provide any information about the secret polynomial. All the verification keys and public parameters are same for each client. The inherent limitation is that the collusion of clients can share their evaluated points with each other and if the total number of points is more than k, where k is the degree of the secret polynomial, then clients can derive the polynomial. This problem exists in any polynomial computation and cannot be prevented.

4 VIP-POPE Description

In our scheme, we assume that the server is not trusted with the computation result and clients are curious to learn about the secret polynomial. A client may forge an encrypted query to gain more information about the secret polynomial. We first give the intuition of our scheme VIP-POPE and then give its formal definition.

We use homomorphic properties of Paillier's cryptosystem to design our scheme called VIP-POPE. The key idea is to use the fact that a client can generate an encrypted query $t = \{t_i\}_{i=1}^{k}$ where $t_i = \mathcal{E}_{\mathsf{pk}}(x^i)$ and k is the degree of the secret polynomial $f(\cdot)$ to allow the server to compute $\mathcal{E}_{\mathsf{pk}}(f(x))$. Since the server knows coefficients $\{a_i\}_{i=0}^{k}$ of $f(\cdot)$, it computes $\mathcal{E}_{\mathsf{pk}}(f(x))$ as follows:

$$\mathcal{E}_{\mathsf{pk}}(a_0) \cdot \prod_{i=1}^{i=k} \mathcal{E}_{\mathsf{pk}}(x^i)^{a_i} = \prod_{i=0}^{i=k} \mathcal{E}_{\mathsf{pk}}(a_i x^i) = \mathcal{E}_{\mathsf{pk}}\left(\sum_{i=0}^{i=k} a_i x^i\right) = \mathcal{E}_{\mathsf{pk}}(f(x)) \ .$$

The client may forges an untrustworthy encrypted query to learn more than a point on the polynomial. To avoid this kind of attack, the client must provide a proof of validity π_t for each query $t = \{t_i\}_{i=1}^{k}$ that she sends to the server, i.e., a proof that $t_i = \mathcal{E}_{\mathsf{pk}}(x^i)$ for all $i \in \{1, \ldots, k\}$. Based on Property 1, such a proof can be built using the NIZKP $\mathsf{DecPaillierEq}$ presented in Definition 1.

4.1 Formal Definition of VIP-POPE

We now give the formal definition of our scheme VIP-POPE. The algorithms setup and init are run by the company, the algorithm $\mathsf{compute}$ is run by the server and the algorithms keyGen, $\mathsf{queryGen}$, $\mathsf{decrypt}$ and verif are run by a client.

Definition 7. *Let VIP-POPE* = (setup, init, keyGen, $\mathsf{queryGen}$, $\mathsf{queryDec}$, $\mathsf{compute}$, $\mathsf{decrypt}$, verif) *be a scheme defined by:*

- $\mathsf{setup}(\eta)$: *Using the security parameter η, this algorithm first generates a prime number q. It selects a multiplicative group G of order q and generated by h. It picks $(s_1, s_2) \leftarrow (\mathbb{Z}_q^\star)^2$ and sets $\mathsf{pub} = (h^{s_1}, h^{s_2}, h, q)$, $\mathsf{sec} = (s_1, s_2)$, and $F = \mathbb{Z}_q$. Finally, it outputs pub, F, and sec.*
- $\mathsf{init}(F, f, \mathsf{sec})$: *We set $f(x) = \sum_{i=0}^{i=k} a_i \cdot x^i$ where $a_i \in \mathbb{Z}_q$. For all $i \in \{0, \ldots, k\}$, it picks $r_i \in \mathbb{Z}_q^\star$ and computes $\alpha_i = (a_i + r_i) \cdot s_1$ and $\gamma_i = s_1 \cdot s_2^{-1} \cdot r_i$. Finally, it sets $\mathsf{vk} = \{\gamma_i\}_{i=0}^{k}$, $\mathsf{sk}_f = \{\alpha_i\}_{i=0}^{k}$, and returns $(\mathsf{vk}, \mathsf{sk}_f)$.*
- $\mathsf{keyGen}(\eta, \mathsf{pub}, k)$: *For a client c, it picks two primes p_c and q_c such that $(k+1)q^2 < p_c q_c$ and $p_c \approx q_c$. It sets $n_c = p_c q_c$. According to n_c, it generates a Paillier key pair such that $\mathsf{pk}_c = (n_c, g_c)$ and $\mathsf{sk}_c = (\lambda_c, \mu_c)$ as described in Section 2. It outputs $(\mathsf{pk}_c, \mathsf{sk}_c)$.*
- $\mathsf{queryGen}(\mathsf{pk}_c, x)$: *Using x and the Paillier public key pk_c, this algorithm computes, for all $i \in \{1, \ldots, k\}$, $t_i = \mathcal{E}_{\mathsf{pk}}(x^i)$ and returns the encrypted query $t = (\mathsf{pk}_c, \{t_i\}_{i=1}^{k})$ along with a proof π_t of equality of plaintexts using $\mathsf{proof}^{\mathsf{PaillierEq}}$.*
- $\mathsf{queryDec}(\mathsf{sk}_c, t)$: *First this algorithm parses t as $(\mathsf{pk}_c, \{t_i\}_{i=1}^{k})$. Using the Paillier secret key sk_c, this algorithm sets $x = \mathcal{D}_{\mathsf{sk}_c}(t_1)$. If $\mathcal{D}_{\mathsf{sk}_c}(t_i) = x^i$ for $2 \leq i \leq k$, it outputs x, $\perp$ otherwise.*
- $\mathsf{compute}(t, \pi_t, f, \mathsf{sk}_f, F)$: *If π_t is accepted by $\mathsf{verify}^{\mathsf{PaillierEq}}$, this algorithm uses $\{t_i\}_{i=1}^{k}$ from t, coefficients $\{a_i\}_{i=0}^{k}$ of the polynomial function $f(\cdot)$, and $\{\alpha_i\}_{i=0}^{k}$ from the server secret key sk_f to compute:*

$$d = \mathcal{E}_{\mathsf{pk}_c}(a_0) \cdot \prod_{i=1}^{i=k} t_i^{a_i} \quad \text{and} \quad \pi_d = \mathcal{E}_{\mathsf{pk}_c}(\alpha_0) \cdot \prod_{i=1}^{i=k} t_i^{\alpha_i} \ ,$$

  and returns (d, π_d), else it returns $\perp$.

– $\mathsf{decrypt}(\mathsf{sk}_c, d)$: *Using the Paillier secret key* sk_c *which is equal to* (λ_c, μ_c)*, this algorithm returns* $y = \mathcal{D}_{\mathsf{sk}_c}(d) \ mod \ q$.
– $\mathsf{verif}(x, \mathsf{sk}_c, \mathsf{pub}, y, \pi_d, \mathsf{vk})$: *Using* x*,* sk_c*,* vk*, and the proof* π_d*, this algorithm computes:*

$$y' = \mathcal{D}_{\mathsf{sk}_c}(\pi_d) \ mod \ q \quad and \quad z = \sum_{i=0}^{i=k} \gamma_i \cdot x^i \ .$$

If $(h^{s_1})^y \cdot (h^{s_2})^z = h^{y'}$*, then the algorithm returns* 1*, else it returns* 0.

Parameter Selection. First, consider the additive group $F = \mathbb{Z}_q$ of order q. The size of the prime q must be at least 1024 bits to make the discrete logarithm problem hard in the group G. We recall that the polynomial $f(\cdot)$ is equal to $\sum_{i=0}^{i=k} a_i \cdot x^i$ where $a_i \in \mathbb{Z}_q$ for all $i \in \{0, \dots, k\}$. Hence, all evaluations are in $\mathbb{Z}_q$; thus we assume that for all $i \in \{0, \dots, k\}$, we have $0 \leq x^i < q$, and that $0 \leq f(x) < q$. Moreover, the client encrypts for all $i \in \{1, \dots, k\}$ the value x^i. The evaluation performed by the server is done over encrypted values, i.e., $\mathcal{E}_{\mathsf{pk}_c}(a_0) \cdot \prod_{i=1}^{i=k} \mathcal{E}_{\mathsf{pk}_c}(x^i)^{a_i} = \mathcal{E}_{\mathsf{pk}_c}(a_0 + a_1 \cdot x + \dots + a_k \cdot x^k)$; then, we need to have $\sum_{i=0}^{i=k} a_i \cdot x^i < n_c = p_c \cdot q_c$ for successful decryption due to Paillier cryptosystem properties, where $\mathbb{Z}_{n_c}$ is the plaintext space of Paillier cryptosystem, p_c and q_c are two prime numbers. Since $0 \leq a_i < q$ and $0 \leq x^i < q$, we have $a_i \cdot x^i < q^2$ for each $i \in \{0, \dots, k\}$ that gives us $a_0 + a_1 \cdot x + \dots + a_k \cdot x^k < (k+1) \cdot q^2$. Hence, we need to have $(k + 1) \cdot q^2 < n_c$ to always have successful decryption. Moreover, we recommend the size of each prime p_c and q_c to be at least 1024 bits to make the factorization of n_c hard.

5 Security and Performance Analysis

We first give a theorem on the security of VIP-POPE. Then we provide some comparisons with PPE schemes of the literature [15, 4, 25].

5.1 Security proofs

We present the security proofs of VIP-POPE in our security model.

Theorem 3. *VIP-POPE is a CPI-secure scheme under the DCR assumption.*

Proof. We assume there exists $\mathcal{A} \in \text{POLY}(\eta)^2$ such that $\mathsf{Adv}^{\mathsf{CPI}}_{\text{VIP-POPE},\mathcal{A}}(\eta)$ is non-negligible and we show there exists an algorithm $\mathcal{B} \in \text{POLY}(\eta)$ such that $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\text{Paillier},\mathcal{B}}(\eta)$ is non-negligible. We build $\mathcal{B}$ as follows:

– $\mathcal{B}$ receives $\mathbb{Z}_q, \mathsf{sec}$ from $\mathsf{setup}(\eta)$ and pk_c from $\mathsf{keyGen}(\eta, \mathsf{pub}, k)$.
– $\mathcal{B}$ runs $(x_0, x_1, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pk}_c)$.
– $\mathcal{B}$ picks $f \xleftarrow{\$} \mathbb{Z}_q[X]^k$ and runs $\mathsf{init}(\mathbb{Z}_q, f, \mathsf{sec})$ to obtain vk and sk_f.
– $\mathcal{B}$ runs the oracle $\mathcal{E}_{\mathsf{pk}}(\mathsf{LR}_b(\cdot, \cdot))$ on (x_0^i, x_1^i) for $i \in \{1, \dots, k\}$ and obtains $t = \{t_i\}_{i=1}^k$, Paillier ciphertexts of x_b^i.
– $\mathcal{B}$ runs $b_* \leftarrow \mathcal{A}_2(t, f, \mathsf{sk}_f, \mathbb{Z}_q, \mathsf{st})$. To simulate the oracle $\mathsf{CO}_{\mathsf{CPI}}(\cdot)$ on x to $\mathcal{A}$, $\mathcal{B}$ computes $t = \{\mathcal{E}_{pk_c}(x^i)\}_{i=1}^k$.
– Finally, $\mathcal{B}$ outputs b_*.

We remark that:

1. The experiment CPI is perfectly simulated for $\mathcal{A}$.
2. $\mathcal{B}$ wins the $\mathsf{IND\text{-}CPA}$ experiment if and only if $\mathcal{A}$ wins the CPI experiment.

Since $\mathsf{Adv}^{\mathsf{CPI}}_{\text{VIP-POPE},\mathcal{A}}(\eta)$ is non-negligible, then $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\text{Paillier},\mathcal{B}}(\eta)$ is non-negligible. However, Paillier cryptosystem is $\mathsf{IND\text{-}CPA}$ under the DCR assumption, then $\mathcal{B}$ can be used to break the DCR assumption, which contradicts our hypothesis and concludes the proof. $\square$

Theorem 4. *For any $k \in \mathbb{N}$, VIP-POPE is a k-IND-CFA-secure scheme.*

Proof. Let $\mathcal{A} \in \text{POLY}(\eta)$ be an algorithm. We show that there exists an algorithm $\mathcal{B} \in \text{POLY}(\eta)$ simulating the experiment $\mathsf{Exp}^{k\text{-}\mathsf{IND\text{-}CFA}}_{\text{VIP-POPE},\mathcal{A}}(\eta)$ to $\mathcal{A}$. We build $\mathcal{B}$ as follows:

– $\mathcal{B}$ picks $b \xleftarrow{\$} \{0,1\}$.
– $\mathcal{B}$ generates $(\mathsf{pub}, \mathbb{Z}_q, \mathsf{sec}) \leftarrow \mathsf{setup}(\eta)$, where $\mathsf{pub} = (h^{s_1}, h^{s_2}, h)$, and $\mathsf{sec} = (s_1, s_2) \in \mathbb{Z}_q^\star$.
– $\mathcal{B}$ runs $(f_0, f_1, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathbb{Z}_q, k)$, and it sets $f_0(x) = \sum_{i=0}^{i=k} a_{0,i} \cdot x^i$ and $f_1(x) = \sum_{i=0}^{i=k} a_{1,i} \cdot x^i$.
– $\mathcal{B}$ picks $r \xleftarrow{\$} \mathbb{Z}_q^\star$. For all $i \in \{0, \ldots, k\}$, it picks $r_i \xleftarrow{\$} \mathbb{Z}_q^\star$, and sets $\alpha_i = (a_{b,i} + r_i) \cdot s_1$, and $\gamma_i = s_1 \cdot s_2^{-1} \cdot r_i$. Finally, it sets $f'(x) = \sum_{i=0}^{i=k} \alpha_i \cdot x^i$, and returns $\mathsf{vk} = \{\gamma_i\}_{i=0}^k$.
– $\mathcal{B}$ generates $(\mathsf{pk}_c, sk_c) \leftarrow \mathsf{keyGen}(\eta, \mathsf{pub}, k)$.
– $\mathcal{B}$ runs $b_* \leftarrow \mathcal{A}_2((\mathsf{pk}_c, \mathsf{sk}_c), \mathsf{pub}, \mathbb{Z}_q^\star, \mathsf{vk}, k, \mathsf{st})$. To simulate the oracle $\mathsf{CO}_{\mathsf{CFA}}(\cdot)$ to $\mathcal{A}$ on $t = \{\mathcal{E}_{\mathsf{pk}}(x^i)\}_{i=1}^k$, $\mathcal{B}$ first verifies if $f_0(\mathcal{D}_{\mathsf{sk}_c}(\mathcal{E}_{\mathsf{pk}_c}(x))) = f_1(\mathcal{D}_{\mathsf{sk}_c}(\mathcal{E}_{\mathsf{pk}_c}(x))$ then computes:
$$d = \mathcal{E}_{\mathsf{pk}_c}(a_{b,0}) \cdot \prod_{i=1}^{i=k} \mathcal{E}_{\mathsf{pk}}(x_j^i)^{a_{b,i}} , \qquad \pi_d = \mathcal{E}_{\mathsf{pk}_c}(\alpha_0) \cdot \prod_{i=1}^{i=k} \mathcal{E}_{\mathsf{pk}}(x_j^i)^{\alpha_i} ,$$
and returns (d, π_d). Else, it returns $\perp$.
– Finally, $\mathcal{B}$ outputs b_*.

We remark that r and r_i (for $0 \leq i \leq k$) are chosen in the uniform distribution of $\mathbb{Z}_q^\star$, then each element of vk comes from the uniform distribution on $\mathbb{Z}_q^\star$. Finally, we have:
$$\begin{aligned}(h^{s_1})^{f(x)} \cdot (h^{s_2})^{Z(x)} &= h^{s_1 \cdot \sum_{i=0}^{i=k} a_i \cdot x^i + s_2 \cdot \sum_{i=0}^{i=k} \gamma_i \cdot x^i} \\ &= h^{s_1 \cdot \sum_{i=0}^{i=k} a_i \cdot x^i + s_2 \cdot \sum_{i=0}^{i=k} s_1 \cdot s_2^{-1} \cdot r_i \cdot x^i} \\ &= h^{\sum_{i=0}^{i=k} (a_i + r_i) \cdot s_1 \cdot x^i} = h^{\sum_{i=0}^{i=k} \alpha_i \cdot x^i} = h^{f'(x)}.\end{aligned}$$

We deduce that the experiment k-$\mathsf{IND\text{-}CFA}$ is perfectly simulated for $\mathcal{A}$. Then $\mathcal{A}$ cannot do better than the random to guess the value of the chosen b. Hence, we have:
$$\Pr[1 \leftarrow \mathsf{Exp}^{k\text{-}\mathsf{IND\text{-}CFA}}_{\text{VIP-POPE},\mathcal{A}}(\eta)] = 1/2 ,$$
and so $\mathsf{Adv}^{k\text{-}\mathsf{IND\text{-}CFA}}_{\text{VIP-POPE},\mathcal{A}}(\eta)$ is negligible which concludes the proof. $\square$

Theorem 5. *For any $k \in \mathbb{N}$, VIP-POPE is QS-secure in the random oracle model.*

Proof. The proof π_t is computed as in $\mathsf{DecPaillierEq}$ (Definition 1). This NIZKP is unconditionally *sound*, then there exists no probabilistic polynomial time algorithm that forges a valid proof on a false statement with non-negligible probability, i.e., a statement $(t_1, \ldots, t_k)$ where there exists $1 \le i \le k$ such that $t_i \ne t_{i-1}^x \cdot r_i^n$ where $n = p \cdot q$ and p, q are two prime numbers, $t_0 \in \mathbb{Z}_{n^2}^*$ $r_i \in \mathbb{Z}_{n^2}^*$, and $x \in \mathbb{Z}_{n^2}^*$.

We show that if there exists $\mathcal{A} \in \text{POLY}(\eta)^2$ such that $\mathsf{Adv}_{\text{VIP-POPE},\mathcal{A}}^{\mathsf{QS}}(\eta)$ is non-negligible, then there exists $\mathcal{B} \in \text{POLY}(\eta)$ that forges a valid proof of an instance where $t_i \ne t_{i-1}^x \cdot r_i^n$. It contradicts the soundness of $\mathsf{DecPaillierEq}$ which concludes the proof. $\mathcal{B}$ works as follows:

- $\mathcal{B}$ runs $(\mathsf{pub}, F, \mathsf{sec}) \leftarrow \mathsf{setup}(\eta)$, $(\mathsf{pk}_c, sk_c) \leftarrow \mathsf{keyGen}(\eta, \mathsf{pub}, k)$, $(f, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pk}_c, \mathsf{pub}, F)$, $(\mathsf{vk}, \mathsf{sk}_f) \leftarrow \mathsf{init}(F, f, \mathsf{sec})$, and $(t, \pi_t) \leftarrow \mathcal{A}_2(\mathsf{pk}_c, \mathsf{pub}, F, \mathsf{vk})$ where $\pi_t = (w, \{u_i\}_{i=1}^k, \{v_i\}_{i=1}^k)$.
- $\mathcal{B}$ returns t as a statement together with the proof π_t.

We observe that since $\mathsf{Adv}_{\text{VIP-POPE},\mathcal{A}}^{\mathsf{QS}}(\eta)$ is non-negligible, then the probability that $f(\mathsf{queryDec}(\mathsf{sk}_c, t)) \ne \mathsf{decrypt}(\mathsf{sk}_c, d)$ and $\mathsf{compute}(t, \pi_t, f, \mathsf{sk}_f, F) \ne \perp$ is non-negligible. Moreover:

- $f(\mathsf{queryDec}(\mathsf{sk}_c, t)) \ne \mathsf{decrypt}(\mathsf{sk}_c, d) \Rightarrow f(x) \ne y$, that means there exists $1 \le i_0 \le k$ such that $\mathcal{D}_{\mathsf{sk}_c}(t_i) \ne x^i$.
- $\mathsf{compute}(t, \pi_t, f, \mathsf{sk}_f, F) \ne \perp \Rightarrow t_{i-1}^w \cdot v_i^n = u_i \cdot t_i^{H(t)} \mod n^2$ for $1 \le i \le k$. Then π_t is a valid proof.

$\mathcal{B}$ returns a valid proof of a false instance with non-negligible probability. □

Theorem 6. *For any $k \in \mathbb{N}$, VIP-POPE is UNF-secure under the DL assumption.*

Proof. We assume there exists $\mathcal{A} \in \text{POLY}(\eta)^2$ such that $\mathsf{Adv}_{\text{VIP-POPE},\mathcal{A}}^{\mathsf{UNF}}(\eta)$ is non-negligible. We show that $\mathcal{A}$ can be used to construct an algorithm $\mathcal{B}$ that computes $\log_h(h^{s_1})$.

First, we note that if $y_* \ne f(x_*)$, then we also have $y'_* \ne y'$ where we denote $y'_* = \mathcal{D}_{\mathsf{sk}_c}(\pi_*)$ and $y' = \sum_{i=0}^{i=k} \alpha_i \cdot x_*^i$. It is easy to check this condition. Therefore, we must have both inequalities $y_* \ne f(x_*)$ and $y'_* \ne y'$ hold. We show that there exists an algorithm $\mathcal{B} \in \text{POLY}(\eta)$ that breaks the DL assumption by computing $\log_h(h^{s_1})$ using $\mathcal{A}$. $\mathcal{B}$ works as follows:

- $\mathcal{B}$ obtains $(\mathsf{pub}, F, \mathsf{sec}) \leftarrow \mathsf{setup}(\eta)$ and $(\mathsf{pk}_c, sk_c) \leftarrow \mathsf{keyGen}(\eta, \mathsf{pub}, k)$.
- $\mathcal{B}$ receives $(f, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pk}_c, \mathsf{pub}, F)$.
- $\mathcal{B}$ runs $(\mathsf{vk}, \mathsf{sk}_f) \leftarrow \mathsf{init}(F, f, \mathsf{sec})$ where $\mathsf{sk}_f = \{\alpha_i\}_{i=0}^k$, then obtains $(x_*, y_*, \pi_*) \leftarrow \mathcal{A}_2(\mathsf{pub}_c, \mathsf{sk}_f, \mathsf{vk}, F, f, \mathsf{st})$.
- $\mathcal{B}$ computes:
$$\log_h(h^{s_1}) = \frac{\mathcal{D}_{\mathsf{sk}_c}(\pi_*) - \sum_{i=0}^{i=k} \alpha_i \cdot x_*^i}{y_* - f(x_*)} \,.$$

Since we have proved that $y_* \neq f(x_*)$, the discrete logarithm $\log_h(h^{s_1})$ can be computed with the same probability as $\mathcal{A}$ wins the UNF experiment. Therefore, based on the DL assumption, there cannot exist an adversary $\mathcal{A}$ such that $\mathsf{Adv}^{\mathsf{UNF}}_{\text{VIP-POPE},\mathcal{A}}(\eta)$ is non-negligible. □

5.2 Comparison with other PPE schemes

In Table 1, we provide comparison of our scheme with $\mathsf{PolyCommit}_{\mathsf{Ped}}$ [15], PIPE [4] and Xia *et al.*'s scheme [25]. We observe that the verification key size and verification cost are constant in $\mathsf{PolyCommit}_{\mathsf{Ped}}$ while in all other schemes it depends on the degree k. The verification equation in $\mathsf{PolyCommit}_{\mathsf{Ped}}$ involves several bilinear pairing which is costly compared to other operations. The verification key size and verification cost are not constant in our scheme but our scheme is pairing free and efficient as compared to other pairing free schemes. Moreover, our scheme VIP-POPE provides client's data privacy while other three schemes do not provide any privacy. To support our claim about efficiency, we implement all these schemes for different values of degrees with realistic parameters.

In our scheme, the verification of the result obtained from the server is done by a client. In such a case, the verification cost becomes important aspect of the scheme. We claim that our scheme is most efficient so far in terms of verification cost. We implement VIP-POPE, PIPE and Xia's scheme in SageMath 8.1

Table 1: Comparison of VIP-POPE with other PPE schemes. We denote by D the constant cost of one Paillier decryption.

Schemes	Setup size	Key size	Verif. cost	Pairing	Assumption	Model	Privacy
$\mathsf{PolyCommit}_{\mathsf{Ped}}$ [15]	$\mathcal{O}(k)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Yes	t-SDH	Standard	No
PIPE [4]	$\mathcal{O}(1)$	$\mathcal{O}(k)$	$\mathcal{O}(k \cdot \log(q))$	No	DDH	ROM	No
Xia *et al.*'s [25]	$\mathcal{O}(1)$	$\mathcal{O}(k)$	$\mathcal{O}(k \cdot \log(q))$	No	DL	Standard	No
VIP-POPE	$\mathcal{O}(1)$	$\mathcal{O}(k)$	$\mathcal{O}(3 \cdot \log(q) + k) + D$	No	DL/DCR	ROM	Yes

on 64-bit PC with Intel Core i5 - 6500 CPU @ 3.2 GHz and 4 GiB RAM. The new scheme, VIP-POPE, provides privacy of client's data while the other two schemes, PIPE and Xia's scheme, do not provide privacy of client's data. To keep the comparison as fair as possible, we implement all three schemes with the same realistic parameters. For our scheme, we choose a 1024 bit prime q and 160 bit prime q_1 such that $q' = 2q_1q + 1$ is a prime. We choose another 1024 bit prime p and set $n = pq'$. The coefficients of the polynomial $f(x)$, the secret values (s_1, s_2) and $\{r_i\}_{i=0}^{k}$ are all selected uniformly at random from $\mathbb{Z}_q^\star$. For Xia's scheme and PIPE, we keep the value of q, the polynomial $f(x)$ and $\{r_i\}_{i=0}^{k}$ same as in VIP-POPE. We compare the cost of only verification equation in all the three schemes.

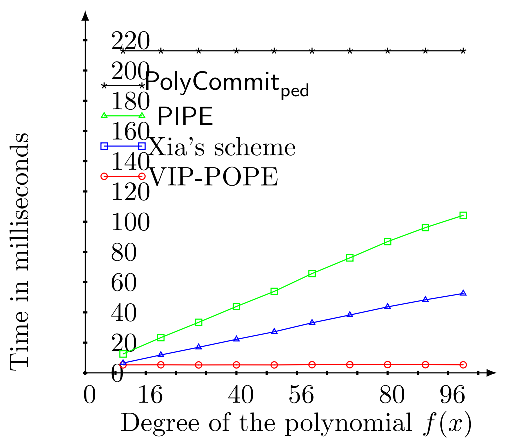

Fig. 7: Verification cost comparison.

For different values of the degree of the polynomial $f(x)$, we ran each scheme for 100 new instances and each instance for 10 times. We then averaged out the total time for the verification equation in each scheme. In Fig. 7, we observe that VIP-POPE takes almost constant time while the cost of verification equation in PIPE and Xia's scheme increases linearly with respect to the degree k. Moreover, our scheme takes only around $5-6$ milliseconds for verification equation even for $k=100$ which makes it practically feasible for real applications.

6 Conclusion

In this paper, we gave a formal definition of new primitive called VPOPE (for Verifiable and Private Oblivious Polynomial Evaluation). This primitive allows a company to delegate the computation of a secret polynomial $f(\cdot)$ to an external server on client's encrypted data in a verifiable way. In other terms, a client sends an encrypted query to a server associated to her secret data x using her own public key pk. Then, the client receives d with a proof that $d = \mathcal{E}_{\mathsf{pk}}(f(x))$. We design the first VPOPE scheme called VIP-POPE (for Verifiable IND-CFA Paillier based Private Oblivious IND-CFA Polynomial Evaluation) and prove that it satisfies the required security properties, i.e., VIP-POPE is CPI-, IND-CFA-, QS-, UNF-secure in the random oracle model. Moreover, we compare our scheme to other existing PPE schemes of the literature and show that its computational verification cost is less as compared others.

References

1. Personal info of 1.5m SingHealth patients, including PM Lee, stolen in Singapore's worst cyber attack. https://www.straitstimes.com/singapore/personal-info-of-15m-singhealth-patients-including-pm-lee-stolen-in-singapores-most, accessed: 2019-08-20
2. Amin, R., Islam, S.H., Biswas, G., Khan, M.K., Kumar, N.: A robust and anonymous patient monitoring system using wireless medical sensor networks. Future Generation Computer Systems **80**, 483–495 (2018)
3. Baudron, O., Fouque, P., Pointcheval, D., Stern, J., Poupard, G.: Practical multi-candidate election system. In: Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing, PODC 2001, Newport, Rhode Island, USA. pp. 274–283 (2001)
4. Bultel, X., Das, M.L., Gajera, H., Gérault, D., Giraud, M., Lafourcade, P.: Verifiable Private Polynomial Evaluation. In: Proceedings of Provable Security - 11th International Conference, ProvSec 2017, Xi'an, China. pp. 487–506 (2017)
5. Canetti, R., Riva, B., Rothblum, G.N.: Two Protocols for Delegation of Computation. In: Proceedings of Information Theoretic Security - 6th International Conference, ICITS, Montreal, QC, Canada. pp. 37–61 (2012)
6. Choi, S.G., Katz, J., Kumaresan, R., Cid, C.: Multi-Client Non-interactive Verifiable Computation. In: Proceedings of Theory of Cryptography - 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan. pp. 499–518 (2013)
7. De Muth, J.E.: Basic statistics and pharmaceutical statistical applications. Chapman and Hall/CRC (2014)

8. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Proceedings of Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA. pp. 186–194 (1986)
9. Fiore, D., Gennaro, R.: Publicly verifiable delegation of large polynomials and matrix computations, with applications. In: Proceedings of the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA. pp. 501–512 (2012)
10. Freedman, M.J., Ishai, Y., Pinkas, B., Reingold, O.: Keyword Search and Oblivious Pseudorandom Functions. In: Proceedings of Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA. pp. 303–324 (2005)
11. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient Private Matching and Set Intersection. In: Proceedings of Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland. pp. 1–19 (2004)
12. Gajera, H., Naik, S., Das, M.L.: On the Security of "Verifiable Privacy-Preserving Monitoring for Cloud-Assisted mHealth Systems". In: Proceedings of Information Systems Security - 12th International Conference, ICISS, Jaipur, India. pp. 324–335 (2016)
13. Gennaro, R., Gentry, C., Parno, B.: Non-interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers. In: Proceedings of Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA. pp. 465–482 (2010)
14. Guo, L., Fang, Y., Li, M., Li, P.: Verifiable privacy-preserving monitoring for cloud-assisted mHealth systems. In: Proceedings of IEEE Conference on Computer Communications, INFOCOM, Kowloon, Hong Kong. pp. 1026–1034 (2015)
15. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-Size Commitments to Polynomials and Their Applications. In: Proceedings of Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore. pp. 177–194 (2010)
16. Lindell, Y., Pinkas, B.: Privacy Preserving Data Mining. Journal of Cryptology **15**(3), 177–206 (2002)
17. Lloret, J., Garcia, M., Bri, D., Sendra, S.: A wireless sensor network deployment for rural and forest fire detection and verification. Sensors **9**(11), 8722–8747 (2009)
18. Naor, M., Pinkas, B.: Oblivious Transfer and Polynomial Evaluation. In: Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, Atlanta, Georgia, USA. pp. 245–254 (1999)
19. Okayama, T.: Future gardening system—smart garden. Journal of Developments in Sustainable Agriculture **9**(1), 47–50 (2014)
20. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Proceedings of Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic. pp. 223–238 (1999)
21. Papamanthou, C., Shi, E., Tamassia, R.: Signatures of Correct Computation. In: Proceedings of Theory of Cryptography - 10th Theory of Cryptography Conference, TCC, Tokyo, Japan. pp. 222–242 (2013)
22. Parno, B., Howell, J., Gentry, C., Raykova, M.: Pinocchio: Nearly Practical Verifiable Computation. In: Proceedings of IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA. pp. 238–252 (2013)

23. Parno, B., Raykova, M., Vaikuntanathan, V.: How to delegate and verify in public: Verifiable computation from attribute-based encryption. In: Proceedings of Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy. pp. 422–439 (2012)
24. Pedersen, T.P.: Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In: Proceedings of Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA. pp. 129–140 (1991)
25. Xia, Z., Yang, B., Zhang, M., Mu, Y.: An Efficient and Provably Secure Private Polynomial Evaluation Scheme. In: Proceedings of 14th International Conference on Information Security Practice and Experience, ISPEC, Tokyo, Japan. pp. 595–609 (2018)