Verifiable and Private Oblivious Polynomial Evaluation

Hardik Gajera [2] Matthieu Giraud [1] David Gérault [1]
Manik Lal Das [2] **Pascal Lafourcade** [1]

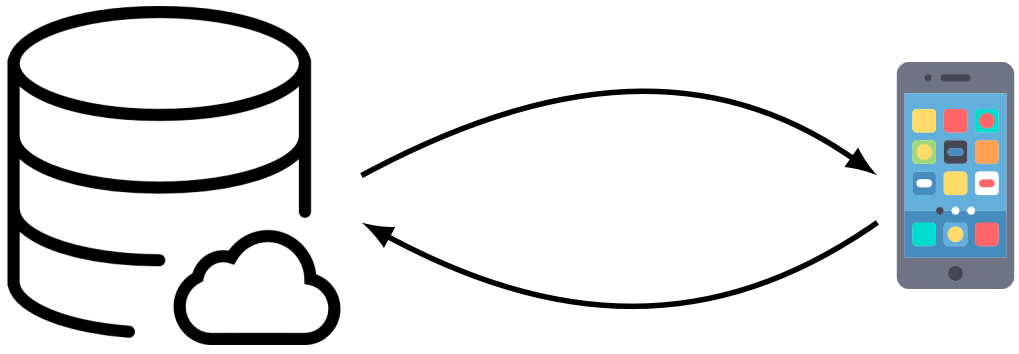
[1]LIMOS, Université Clermont Auvergne, Clermont-Ferrand, France
[2]DA-IICT, Gandhinagar, India

WISTP 2019, Paris, France

December 11, 2019

LABORATOIRE D'INFORMATIQUE,
DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

Cloud Computing

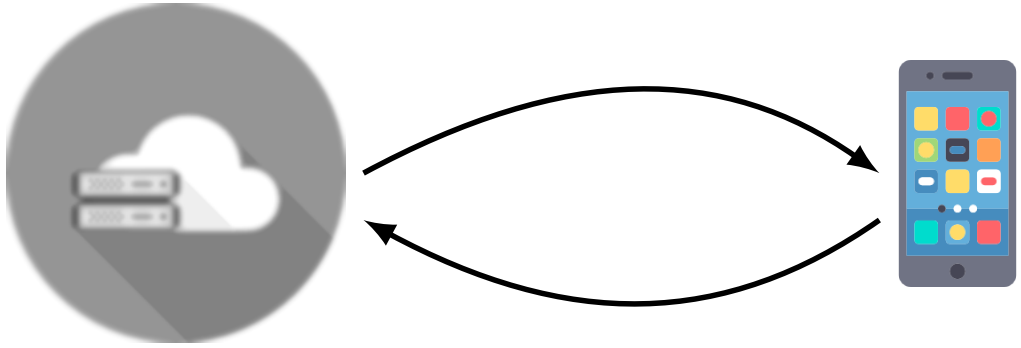


- Small devices outsourcing complex computations
- No need maintain hardware

Private Oblivious Evaluation

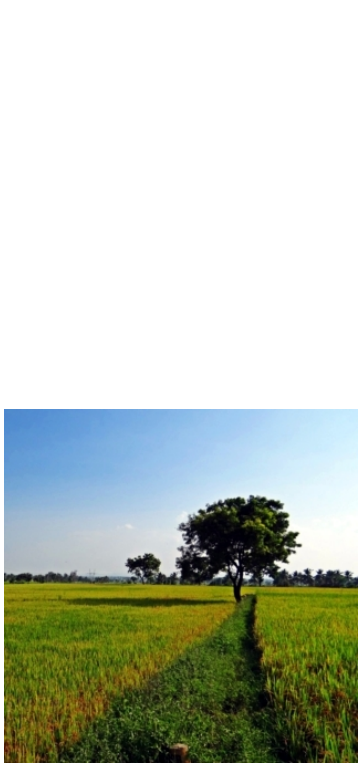
Private Computation

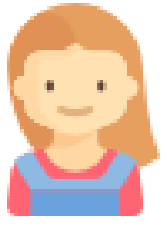
- Health service
- Financial service



Issue: correctness of result? privacy of data?

Private Polynomial Evaluation Scheme (PPE)





Alice

Server

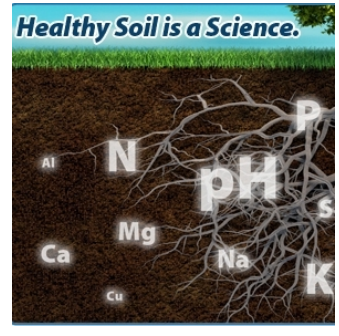
x

$(f(x), \pi)$

f

vk

Company

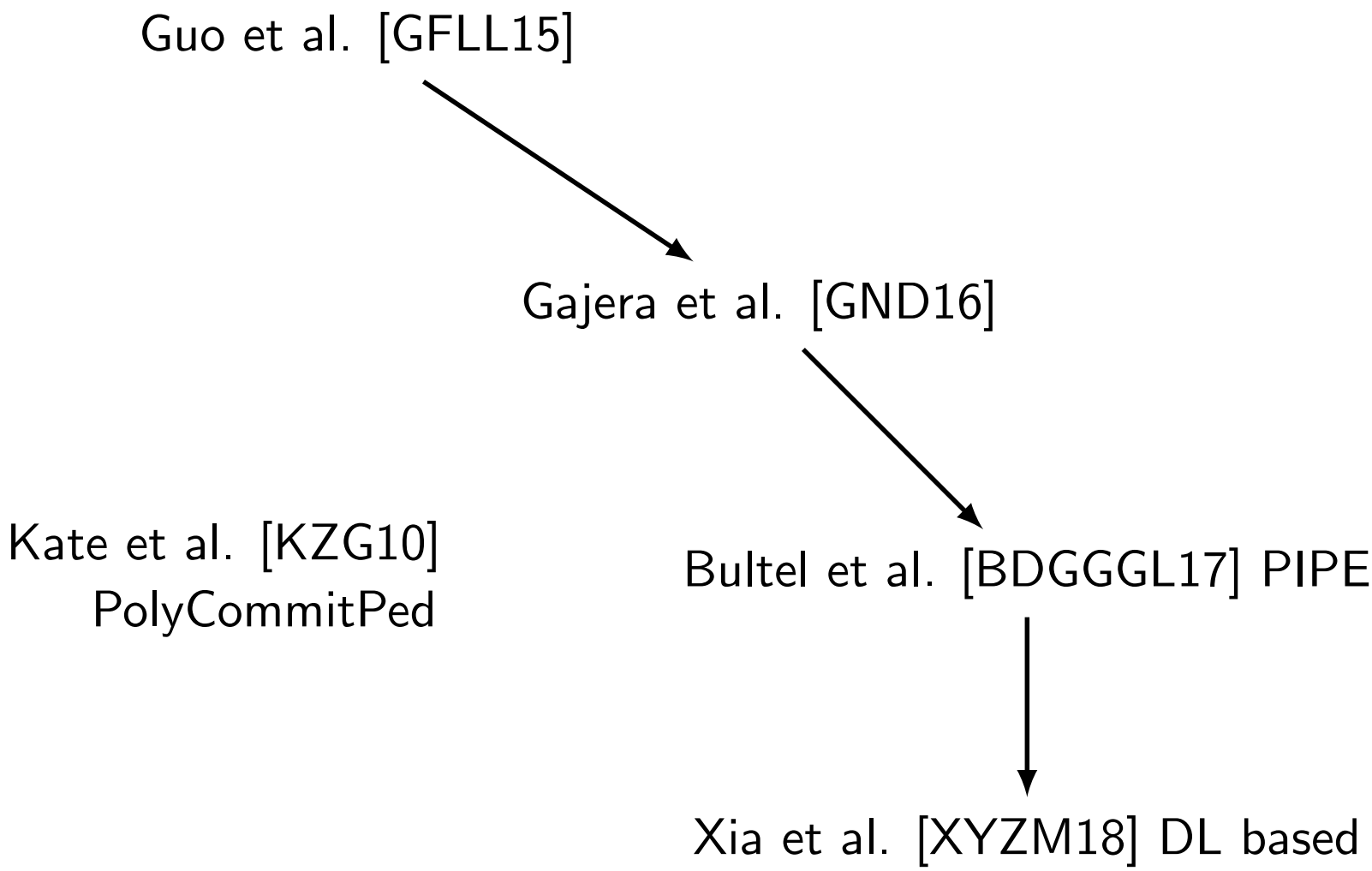


Contributions

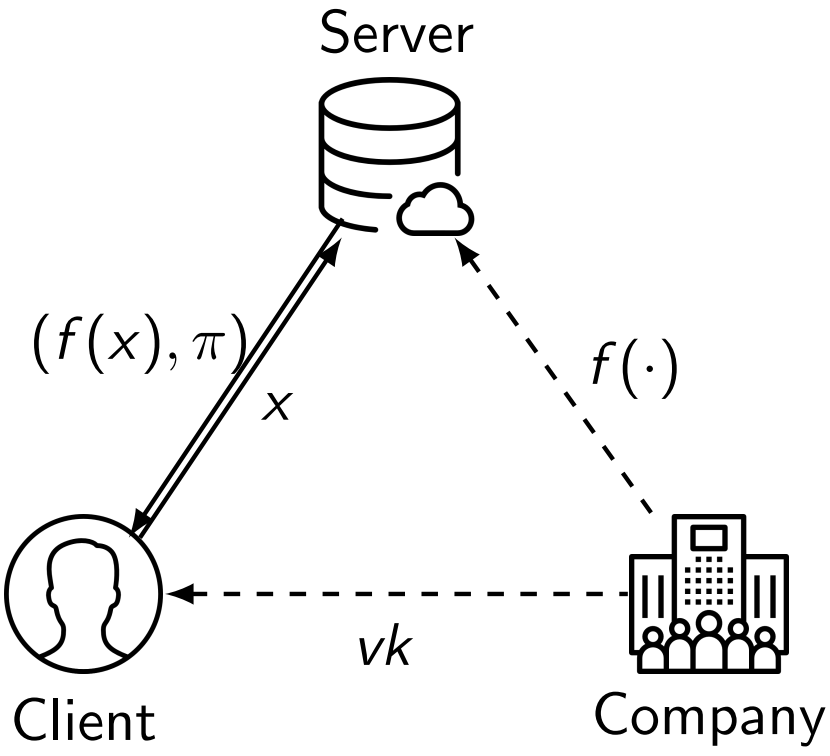
Verifiable and Private Oblivious Polynomial Evaluation

1. Formal definition of VPOPE scheme and security framework
2. Design of a VPOPE scheme called **VIP-POPE**
3. Security proofs and comparison with existing PPE schemes

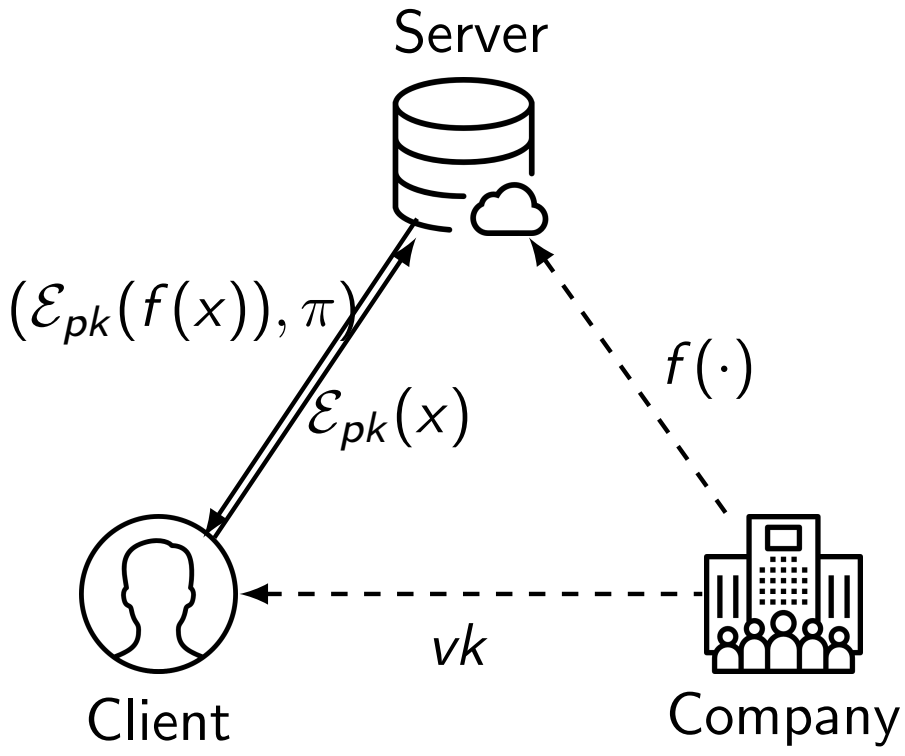
Related Work



PIPE vs VPOPE



PIPE scheme



VPOPE scheme

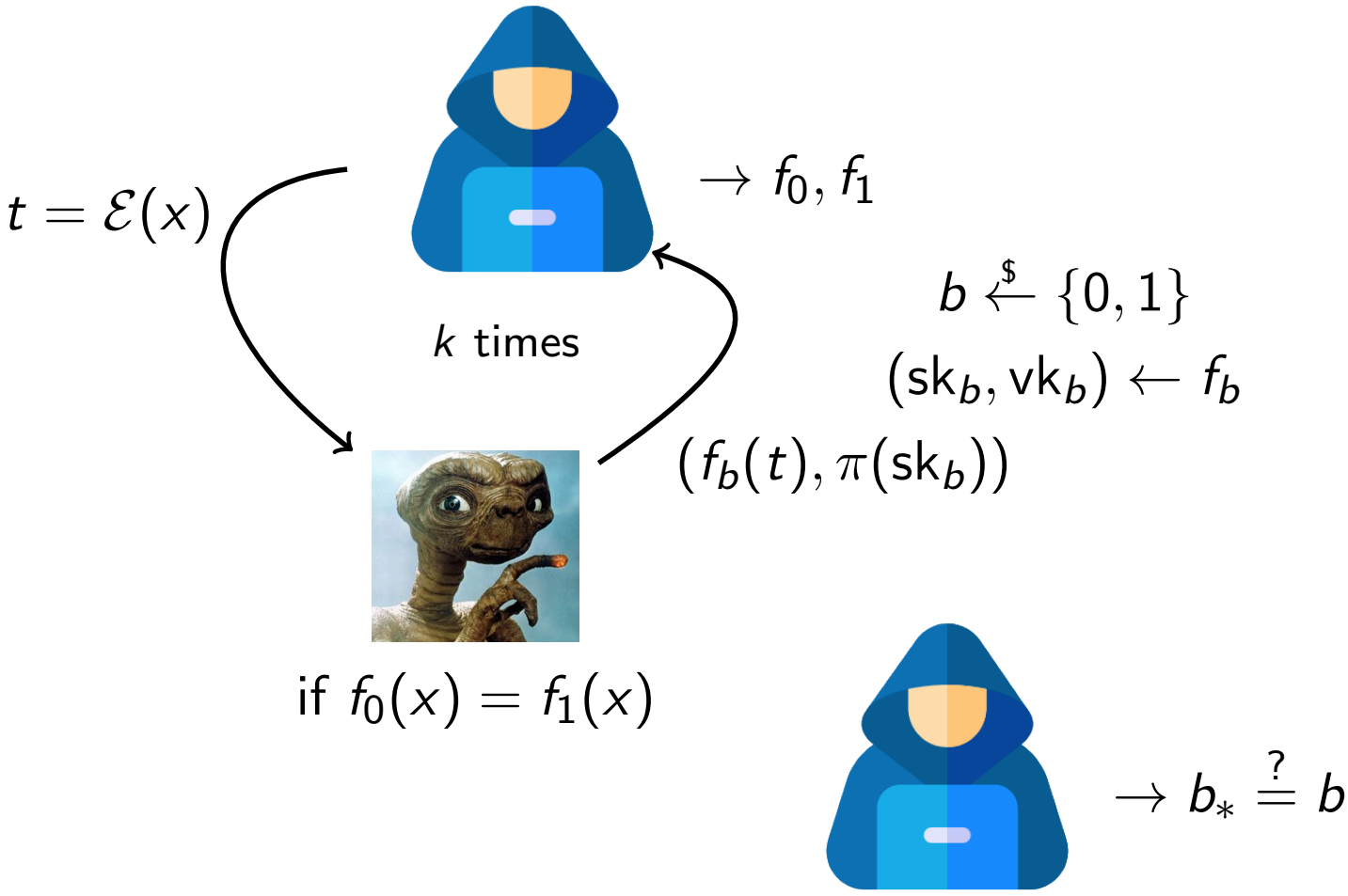
Comparison

	VIP-POPE	PolyCommit$_{\text{Ped}}$	PIPE	Xia *et al.*
Setup size	$\mathcal{O}(1)$	$\mathcal{O}(k)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
Key size	$\mathcal{O}(k)$	$\mathcal{O}(1)$	$\mathcal{O}(k)$	$\mathcal{O}(k)$
Verif. cost	$\mathcal{O}(3 \cdot \log(q)) + D$	$\mathcal{O}(1)$	$\mathcal{O}(k \cdot \log(q))$	$\mathcal{O}(k \cdot \log(q))$
Pairing	Pairing free	Pairing based	Pairing free	Pairing free
Assumption	DL/DCR	t-DDH	DDH	DL
Model	ROM	Standard	ROM	Standard
Privacy	Yes	No	No	No

Outline

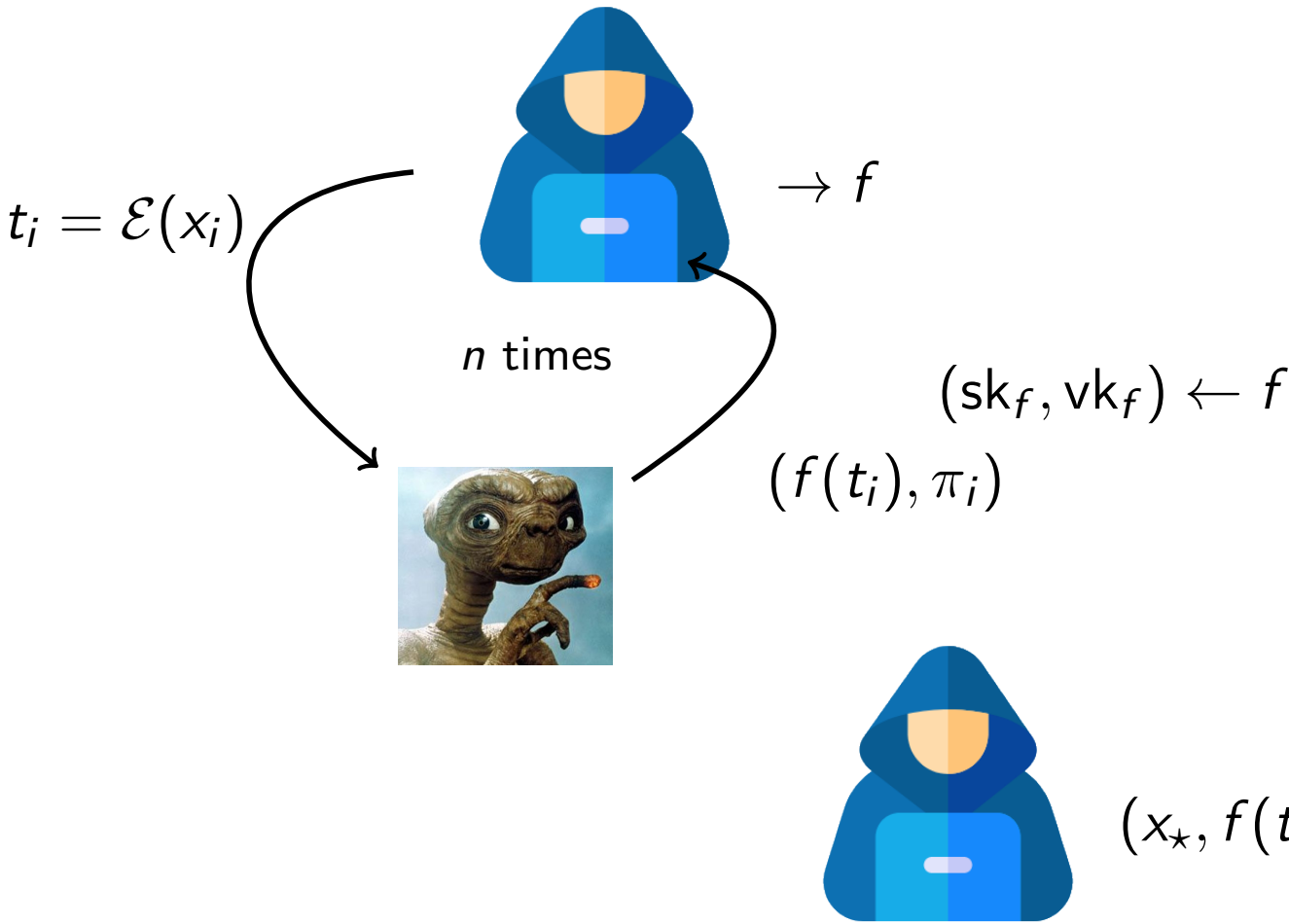
IND-CFA: IND against Chosen Function Attack

IND-CFA Experiment



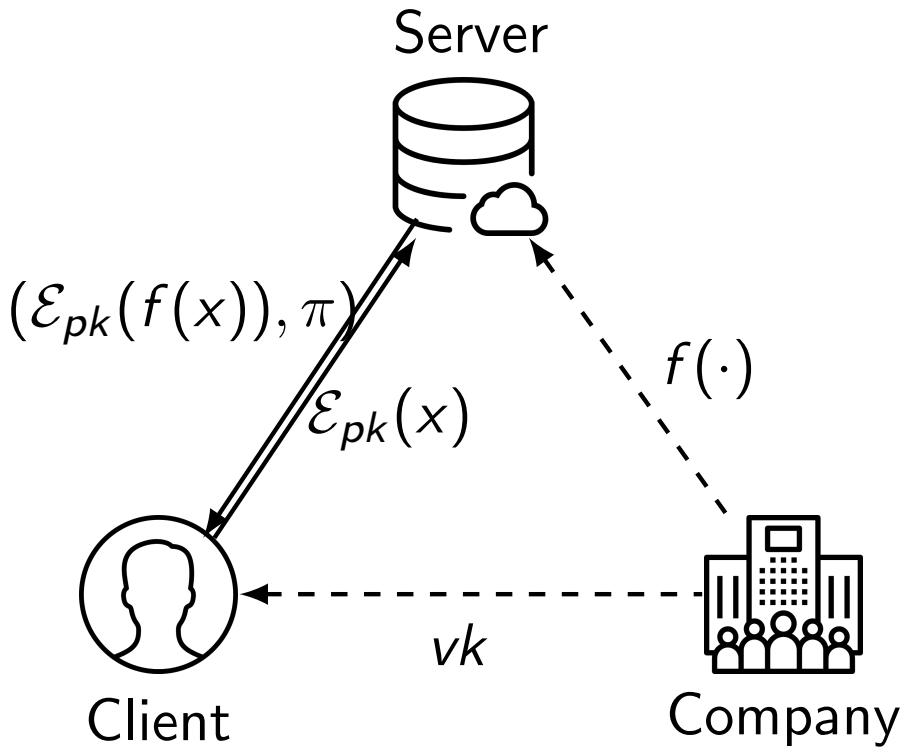
Unforgeability

UNF Experiment



$\mathcal{D}(f(t_\star)) \neq f(x_\star)$ and $\mathit{verif}(x_\star, f(t_\star), \pi_\star) = 1$

Client's Privacy-Indistinguishability (CPI)



An adversary cannot distinguish two queries

Outline

Recall

Paillier Encryption

- KeyGen
  - $sk = (\lambda, \mu)$ where $\lambda = \text{lcm}(p-1, q-1)$, $\mu = (L(g^\lambda \bmod n^2)^{-1})$ and $L(x) = \frac{x-1}{n}$
  - $pk = (G, g, n)$ where $G = \langle g \rangle, |G| = n = pq$
- $\text{Enc}_{pk}(m) = g^m r^n \bmod n^2$ where $r \xleftarrow{\$} \mathbb{Z}_n^\star$
- $\text{Dec}_{sk}(c) = L(c^\lambda \bmod n^2) \cdot \mu \bmod n$

Homomorphic Properties

$$\mathcal{E}_{pk}(m_1) \cdot \mathcal{E}_{pk}(m_2) = \mathcal{E}_{pk}(m_1 + m_2)$$

$$\mathcal{E}_{pk}(m_1)^{m_2} = \mathcal{E}_{pk}(m_1 m_2)$$

Plaintext Equality of k Paillier Ciphertexts

$t = (t_1, \ldots, t_k)$ where $t_i = \mathcal{E}_{pk}(x^i) = t_{i-1}^x r_i^n \bmod n^2$ and $t_0 = g$.

DecPaillierEq

1. **Prove:** Pick $\rho \in \mathbb{Z}_n^\star$ and $s_i \in \mathbb{Z}_n^\star$.

   $$u_i = t_{i-1}^\rho \cdot s_i^n \bmod n^2 \qquad \text{and} \qquad v_i = s_i \cdot r_i^{H(t)} \bmod n$$

   Witness $w = \rho + x \cdot H(t)$ and proof $\pi_t = (w, \{u_i, v_i\}_{i=1}^k)$.

Plaintext Equality of k Paillier Ciphertexts

$t = (t_1, \ldots, t_k)$ where $t_i = \mathcal{E}_{pk}(x^i) = t_{i-1}^x r_i^n \bmod n^2$ and $t_0 = g$.

DecPaillierEq

1. **Prove:** Pick $\rho \in \mathbb{Z}_n^\star$ and $s_i \in \mathbb{Z}_n^\star$.

   $$u_i = t_{i-1}^\rho \cdot s_i^n \bmod n^2 \qquad \text{and} \qquad v_i = s_i \cdot r_i^{H(t)} \bmod n$$

   Witness $w = \rho + x \cdot H(t)$ and proof $\pi_t = (w, \{u_i, v_i\}_{i=1}^k)$.
2. **Verify:** For each $1 \leq i \leq k$, check

   $$t_{i-1}^w \cdot v_i^n = u_i \cdot t_i^{H(t)} \bmod n^2$$

VIP-POPE: **V**erifiable **I**ND-CFA **P**rivate **O**blivious **P**olynomial **E**valution

Idea

$f(x) = \sum_{i=0}^{k} a_i \cdot x^i$
The verification and secret keys corresponding to f

$$sk_f : \{\alpha_i = (a_i + r_i) \cdot s_1\}_{0 \leq i \leq k}$$
$$vk_f : \{\gamma_i = s_1 \cdot s_2^{-1} \cdot r_i\}_{0 \leq i \leq k}$$

where $r_i \xleftarrow{\$} \mathbb{Z}_q^\star$ and $(s_1, s_2) \in (\mathbb{Z}_q^\star)^2$.

The secret (s_1, s_2) is known to the company only.

VIP-POPE: **V**erifiable **I**ND-CFA **P**rivate **O**blivious **P**olynomial **E**valution

Idea

$f(x) = \sum_{i=0}^{k} a_i \cdot x^i$
The verification and secret keys corresponding to f

$$sk_f : \{\alpha_i = (a_i + r_i) \cdot s_1\}_{0 \leq i \leq k}$$
$$vk_f : \{\gamma_i = s_1 \cdot s_2^{-1} \cdot r_i\}_{0 \leq i \leq k}$$

where $r_i \xleftarrow{\$} \mathbb{Z}_q^\star$ and $(s_1, s_2) \in (\mathbb{Z}_q^\star)^2$.

The secret (s_1, s_2) is known to the company only.

IND-CFA $\Rightarrow$ attacker does not know if $\{(\alpha_i, \gamma_i)\}_{0 \leq i \leq k}$ is related to f_0 or f_1

Idea

- Computation

$$y = \sum_{i=0}^{k} a_i \cdot x^i$$

$$y' = \sum_{i=0}^{k} \alpha_i \cdot x^i$$

- Verification

$$(h^{s_1})^y \cdot (h^{s_2})^{\sum_{i=0}^{k} \gamma_i \cdot x^i} = h^{y'}$$

Recall : $\{\alpha_i = (a_i + r_i) \cdot s_1\}_{0 \leq i \leq k}$ and $\{\gamma_i = s_1 \cdot s_2^{-1} \cdot r_i\}_{0 \leq i \leq k}$

VIP-POPE Scheme

A PPE scheme is composed of following algorithms:

1. setup
2. init
3. keyGen
4. queryGen
5. compute
6. decrypt
7. verif

VIP-POPE Scheme

setup(η)

- Group G of prime order q with a generator h
- $(s_1, s_2) \xleftarrow{\$} (\mathbb{Z}_q^\star)^2$
- Return pub $= (G, q, h, h^{s_1}, h^{s_2})$, sec $= (s_1, s_2)$

init($\mathbb{Z}_q, f$, sec)

- $f(x) = \sum_{i=0}^{k} a_i \cdot x^i$ where $a_i \in \mathbb{Z}_q^*$
- For $0 \leq i \leq k$, pick $r_i \xleftarrow{\$} \mathbb{Z}_q^*$ and compute

$$\alpha_i = (a_i + r_i) \cdot s_1 \quad \text{and} \quad \gamma_i = s_1 \cdot s_2^{-1} \cdot r_i$$

- Return $(sk_f : \{\alpha_i\}_{0 \leq i \leq k}, vk_f : \{\alpha_i\}_{0 \leq i \leq k})$

VIP-POPE Scheme

keyGen(η, pub, k)

- Returns Paillier key pairs (pk_c, sk_c) for the user c

queryGen(pk_c, x, k)

- For $0 \leq i \leq k$, computes $t_i = \mathcal{E}_{pk_c}(x^i)$
- Returns $t = (pk_c, \{t_i\}_{i \leq 0 \leq k})$ along with the proof π_t of equality of plaintext using proof$^{\text{PaillierEq}}$

compute(t, π_t, f, sk_f, pub)

- Verify π_t using verify$^{\text{PaillierEq}}$
- Compute $d = \prod_{i=0}^{k} t_i^{a_i}$ and $\imath_d = \prod_{i=0}^{k} t_i^{\alpha_i}$
- Return (d, π_d)

VIP-POPE Scheme

decrypt(sk_c, d)

- Decrypt d as $y = \mathcal{D}_{sk_c}(d) \bmod q$
- Return y

verif($x, sk_c, \text{pub}, y, \pi_d, vk_f$)

- Compute $y' = \mathcal{D}_{sk_c}(\pi_d) \bmod q$ and $z = \sum_{i=0}^{k} \gamma_i \cdot x^i$
- Verify

$$(h^{s_1})^y \cdot (h^{s_2})^z = h^{y'}$$

PIPE Scheme

Theorem

VIP-POPE scheme is IND-CFA-secure scheme.

Theorem

VIP-POPE scheme is CPI-secure under Decisional Composite Residute DCR assumption.

Theorem

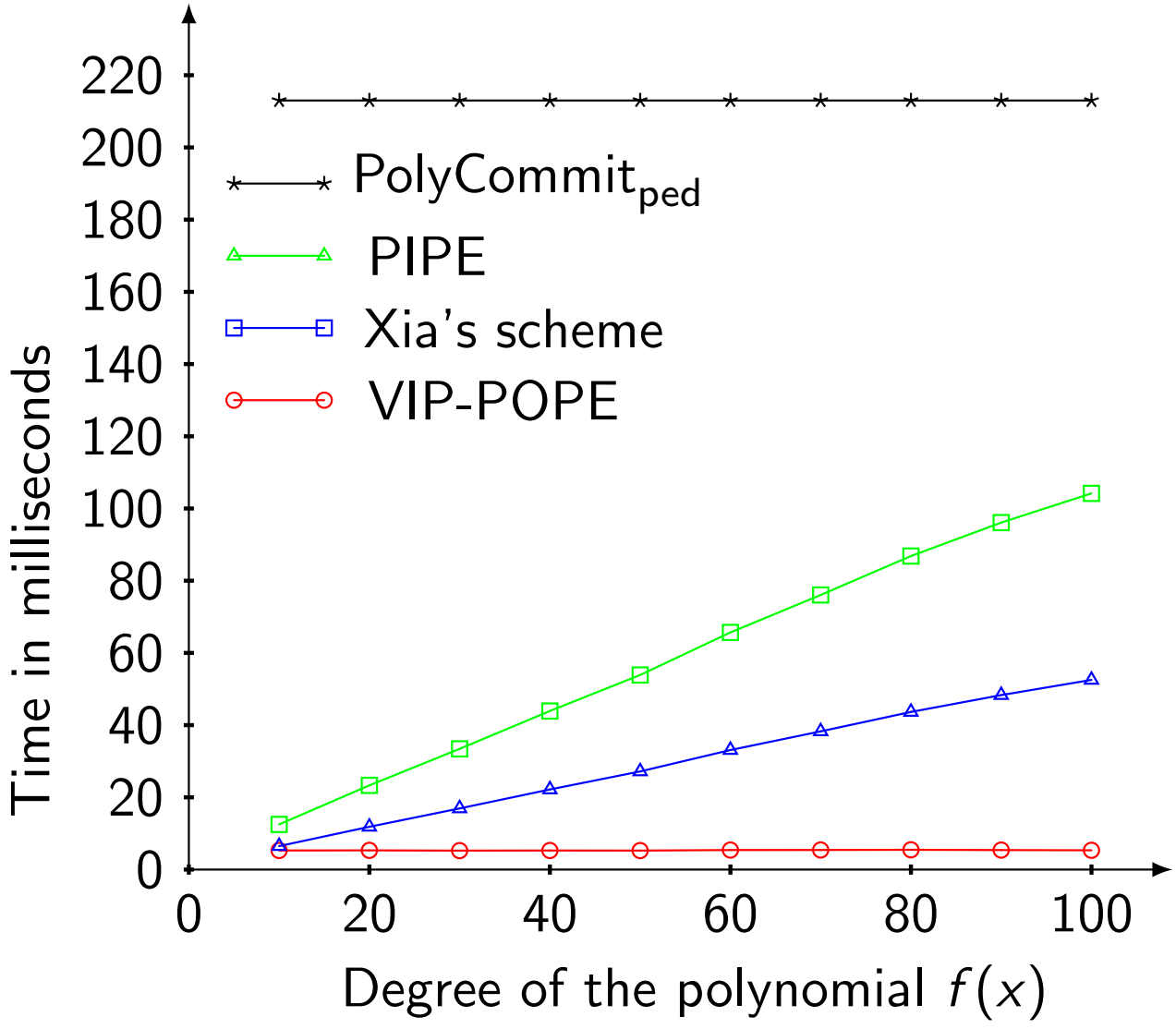
VIP-POPE scheme is UNF-secure under the DL assumption.

Outline

Experiment

- **System:** 64-bit PC with Intel Core i5-6500 CPU @ 3.2 GHz and 4 GB RAM
- **Platform:** SageMath 8.1
- **Parameters:** p, q are 1024 bit primes and q_1 is 160 bit prime such that $q' = 2q_1q + 1$ is a prime
  $n = pq'$ and $f \in \mathbb{Z}_q[X]$

Experiment : Verification cost comparison



Outline

Contributions

Verifiable and Private Oblivious Polynomial Evaluation

1. Formal definition of VPOPE scheme and security framework
2. Design of a VPOPE scheme called **VIP-POPE**
3. Security proofs
4. Better than existing schemes

Future Works

- Multivariate Polynomial with inputs from different users
- Generalization to other functions

Thank you for your attention.

Any questions?

pascal.lafourcade@uca.fr