Functional Encryption for Inner Product with Full Function Privacy

by

Sourav Mukhopadhyay

joint work with

Pratish Datta and **Ratna Dutta**

Department of Mathematics
Indian Institute of Technology Kharagpur
Kharagpur-721302
India

PKC 2016
6–9th March, 2016

Outline

Functional Encryption and Secure Delegation of Computation

- In a functional encryption (FE) scheme for certain function family $\mathcal{F}$, it is possible to derive functional keys SK_f for any function $f \in \mathcal{F}$ from a master secret key.
- Any party given such a functional key SK_f and a ciphertext CT_z encrypting some message z, should be able to learn $f(z)$ and nothing beyond that about z.
- FE enables secure computation on private sensitive data outsourced to untrusted servers by remotely querying the server.

Need of Function Privacy in Functional Encryption

- Assume that a health organization subscribes to a cloud service provider to store medical records of its patients.
- To ensure data confidentiality, the organization encrypts those records locally using an FE scheme prior to uploading them to the cloud server.
- Now, the health organization gives the cloud a functional key corresponding to the function that determines the names of the patients who are receiving treatment for some chronic disease.
- Say, after performing the assigned computation on the encrypted records using the given functional key, the cloud server obtains a list of patients that includes the name of a certain celebrity.
- If the cloud server also comes to know the functionality it has computed on the encrypted records yielding that list, it would at once understand that the particular celebrity is suffering from such a chronic disease and it might leak this information to the media, possibly for financial gain.

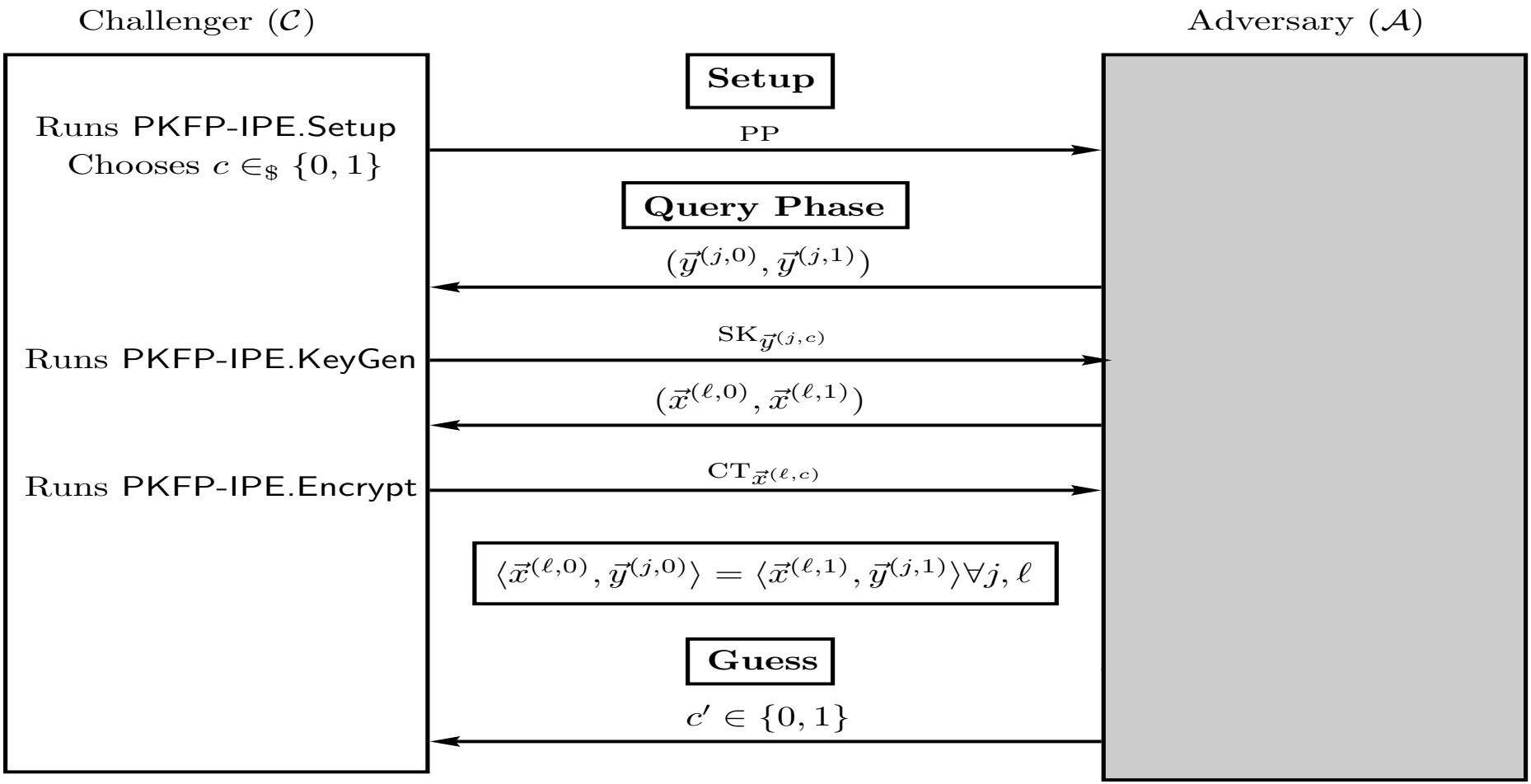
Inner Product Functionality and its Applications

- A function $\text{IP}_{\vec{y}} \in \mathcal{IP}_p$ is associated with a vector $\vec{y} \in \mathbb{Z}_p^n$ over the finite field $\mathbb{Z}_p$, where p is a prime integer.
- On a message $\vec{x} \in \mathbb{Z}_p^n$, $\text{IP}_{\vec{y}}(\vec{x}) = \langle \vec{x}, \vec{y} \rangle$ modulo p.
- Inner product is extremely useful functionality in the context of descriptive statistics, e.g., to compute the weighted mean of a collection of informations.
- Inner product enables computation of conjunctions, disjunctions, polynomial evaluations, and exact thresholds.

Syntax of Private Key Function-Private Inner Product Encryption (PKFP-IPE)

- PKFP-IPE.Setup$(1^\lambda, n) \rightarrow \mathrm{MSK}, \mathrm{PP}$
- PKFP-IPE.Encrypt$(\mathrm{MSK}, \mathrm{PP}, \vec{x} \in \mathbb{Z}_p^n \backslash \{\vec{0}\}) \rightarrow \mathrm{CT}_{\vec{x}}$
- PKFP-IPE.KeyGen$(\mathrm{MSK}, \mathrm{PP}, \vec{y} \in \mathbb{Z}_p^n \backslash \{\vec{0}\}) \rightarrow \mathrm{SK}_{\vec{y}}$
- PKFP-IPE.Decrypt$(\mathrm{PP}, \mathrm{CT}_{\vec{x}}, \mathrm{SK}_{\vec{y}}) \rightarrow \langle \vec{x}, \vec{y} \rangle$

Full-Hiding Security Model for PKFP-IPE



$$\mathsf{Adv}_{\mathcal{A}}^{\text{PKFP-IPE}}(\lambda) = |\Pr[c' = c] - 1/2| \leq \mathsf{negl}(\lambda)$$

Motivation

- The security framework of [BJK15] assumes that for all $(\vec{y}^{(j,0)}, \vec{y}^{(j,1)})$ and $(\vec{x}^{(\ell,0)}, \vec{x}^{(\ell,1)})$ with which the adversaries query the functional key generation and encryption oracles respectively, it holds that

$$\langle \vec{x}^{(\ell,0)}, \vec{y}^{(j,0)} \rangle = \boxed{\langle \vec{x}^{(\ell,0)}, \vec{y}^{(j,1)} \rangle = \langle \vec{x}^{(\ell,1)}, \vec{y}^{(j,0)} \rangle} = \langle \vec{x}^{(\ell,1)}, \vec{y}^{(j,1)} \rangle$$

  which is a *stronger* requirement than the restriction imposed in full-hiding security model.
- Our goal is to develop function-private PKFP-IPE scheme whose security *does not require* any such *extra restriction* beyond that specified in the full-hiding security model.
- We attempt to build PKFP-IPE which is *non-generic* and uses *efficient* and *standard* primitives.

[BJK15]: Allison Bishop, Abhishek Jain, and Lucas Kowalczyk. ASIACRYPT 2015.

Asymmetric Bilinear Pairing Group

An asymmetric bilinear pairing group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \xleftarrow{\$} \mathcal{G}_{\mathsf{ABPG}}(1^\lambda)$ is a tuple of

- a prime integer p;
- cyclic multiplicative groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of order p each with polynomial-time computable group operations;
- generators $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$;
- a polynomial-time computable pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ that satisfies
  - (*bilinearity*) $e(g_1^s, g_2^{\breve{s}}) = e(g_1, g_2)^{s\breve{s}}$ for all $s, \breve{s} \in \mathbb{Z}_p$ and
  - (*non-degeneracy*) $e(g_1, g_2) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ denotes the identity element of the group $\mathbb{G}_T$.

Dual Pairing Vector Spaces (DPVS)

A DPVS $(p, \mathbb{V}_1, \mathbb{V}_2, \mathbb{G}_T, \mathbb{A}_1, \mathbb{A}_2, E) \leftarrow \mathcal{G}_{\mathsf{DPVS}}(n, (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e))$ is a tuple of

- a prime integer p;
- n-dimensional vector space $\mathbb{V}_h = \mathbb{G}_h^n$ over $\mathbb{Z}_p$ under $g_h^{\vec{v}} \oplus g_h^{\vec{w}} = g_h^{\vec{v}+\vec{w}}$ and $a \otimes g_h^{\vec{v}} = g_h^{a\vec{v}}$, for $h = 1, 2$, where $\vec{v}, \vec{w} \in \mathbb{Z}_p^n$, and $a \in \mathbb{Z}_p$;
- canonical bases $\mathbb{A}_h = \{g_h^{\vec{e}_i}\}_{i=1,\ldots,n}$ of $\mathbb{V}_h$, for $h = 1, 2$, where $\vec{e}_i = (\overbrace{0, \ldots, 0}^{i-1}, 1, \overbrace{0, \ldots, 0}^{n-i}) \in \mathbb{Z}_p^n$;
- a pairing $E : \mathbb{V}_1 \times \mathbb{V}_2 \rightarrow \mathbb{G}_T$ defined by

$$E(g_1^{\vec{v}}, g_2^{\vec{w}}) = \prod_{i=1}^{n} e(g_1^{v_i}, g_2^{w_i}) = e(g_1, g_2)^{\langle \vec{v}, \vec{w} \rangle} \in \mathbb{G}_T,$$

where $\vec{v}, \vec{w} \in \mathbb{Z}_p^n$, that satisfies

- (*bilinearity*) $E(s \otimes g_1^{\vec{v}}, \breve{s} \otimes g_2^{\vec{w}}) = E(g_1^{s\vec{v}}, g_2^{\breve{s}\vec{w}}) = E(g_1^{\vec{v}}, g_2^{\vec{w}})^{s\breve{s}}$ for $s, \breve{s} \in \mathbb{Z}_p$, $\vec{v}, \vec{w} \in \mathbb{Z}_p^n$ and
- (*non-degeneracy*) if $E(g_1^{\vec{v}}, g_2^{\vec{w}}) = 1_{\mathbb{G}_T}$ for all $\vec{w} \in \mathbb{Z}_p^n$, then $\vec{v} = \vec{0}$.

Dual orthonormal basis generator $\mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^n)$

1. Choose $\boldsymbol{B} = (b_{i,j})_{i,j=1,\ldots,n} \xleftarrow{\$} \mathsf{GL}(n, \mathbb{Z}_p)$.
2. Compute $\boldsymbol{B}^* = (b^*_{i,j})_{i,j=1,\ldots,n} = (\boldsymbol{B}^\intercal)^{-1}$.
3. Let, $\vec{b}_i$ and $\vec{b}^*_i$ represent the i-th rows of $\boldsymbol{B}$ and $\boldsymbol{B}^*$ respectively, for $i = 1, \ldots, n$.
4. Set $\mathbb{B} = \{\vec{b}_1, \ldots, \vec{b}_n\}$ and $\mathbb{B}^* = \{\vec{b}^*_1, \ldots, \vec{b}^*_n\}$.
5. $(\mathbb{B}, \mathbb{B}^*)$ are dual orthonormal in the sense that for $i, i' = 1, \ldots, n$,

$$\langle \vec{b}_i, \vec{b}^*_{i'} \rangle = \begin{cases} 1, & \text{if } i = i' \\ 0, & \text{otherwise} \end{cases}$$

6. Return $(\mathbb{B}, \mathbb{B}^*)$.

Construction

PKFP-IPE.Setup$(1^\lambda, n)$

1. $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \xleftarrow{\$} \mathcal{G}_{\mathsf{ABPG}}(1^\lambda)$.

2. $(p, \mathbb{V}_1, \mathbb{V}_2, \mathbb{G}_T, \mathbb{A}_1, \mathbb{A}_2, E) \leftarrow \mathcal{G}_{\mathsf{DPVS}}\big(4n+2, (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)\big)$,
$(p, \mathbb{V}'_1, \mathbb{V}'_2, \mathbb{G}_T, \mathbb{A}'_1, \mathbb{A}'_2, E') \leftarrow \mathcal{G}_{\mathsf{DPVS}}\big(6, (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)\big)$.

3. $\big(\mathbb{B} = \{\vec{b}_1, \ldots, \vec{b}_{4n+2}\},\ \mathbb{B}^* = \{\vec{b}^*_1, \ldots, \vec{b}^*_{4n+2}\}\big) \xleftarrow{\$} \mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^{4n+2})$,
$\big(\mathbb{D} = \{\vec{d}_1, \ldots, \vec{d}_6\},\ \mathbb{D}^* = \{\vec{d}^*_1, \ldots, \vec{d}^*_6\}\big) \xleftarrow{\$} \mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^6)$.

4. Define $\widehat{\mathbb{B}} = \{\vec{b}_1, \ldots, \vec{b}_n, \vec{b}_{4n+2}\},\ \widehat{\mathbb{B}}^* = \{\vec{b}^*_1, \ldots, \vec{b}^*_n, \vec{b}^*_{4n+1}\}$,
$\widehat{\mathbb{D}} = \{\vec{d}_1, \vec{d}_6\}$, $\widehat{\mathbb{D}}^* = \{\vec{d}^*_1, \vec{d}^*_5\}$.

5. Keep $\mathrm{MSK} = (\widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \widehat{\mathbb{D}}, \widehat{\mathbb{D}}^*)$.
Publish $\mathrm{PP} = \big(p, \{\mathbb{V}_h, \mathbb{V}'_h\}_{h=1,2}, \mathbb{G}_T, \{\mathbb{A}_h, \mathbb{A}'_h\}_{h=1,2}, E, E'\big)$.

Construction

PKFP-IPE.Encrypt($\mathrm{MSK}, \mathrm{PP}, \vec{x} \in \mathbb{Z}_p^n \backslash \{\vec{0}\}$)

1. Select $\alpha, \xi, \xi_0 \xleftarrow{\$} \mathbb{Z}_p$ and compute

$$\begin{aligned} \boldsymbol{c}_1 &= g_1^{\alpha \sum_{i=1}^{n} x_i \vec{b}_i + \xi \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2 &= g_1^{\alpha \vec{d}_1 + \xi_0 \vec{d}_6} \end{aligned}$$

   utilizing $\widehat{\mathbb{B}}$ and $\widehat{\mathbb{D}}$ respectively from MSK.

2. Output $\mathrm{CT}_{\vec{x}} = (\boldsymbol{c}_1, \boldsymbol{c}_2)$.

Construction

PKFP-IPE.KeyGen($\mathrm{MSK}, \mathrm{PP}, \vec{y} \in \mathbb{Z}_p^n \backslash \{\vec{0}\}$)

1. Pick $\gamma, \eta, \eta_0 \xleftarrow{\$} \mathbb{Z}_p$ and compute

$$\begin{aligned} \boldsymbol{k}_1^* &= g_2^{\gamma \sum_{i=1}^{n} y_i \vec{b}_i^* + \eta \vec{b}_{4n+1}^*}, \\ \boldsymbol{k}_2^* &= g_2^{\gamma \vec{d}_1^* + \eta_0 \vec{d}_5^*} \end{aligned}$$

   utilizing $\widehat{\mathbb{B}}^*$ and $\widehat{\mathbb{D}}^*$ respectively from MSK.

2. Provide $\mathrm{SK}_{\vec{y}} = (\boldsymbol{k}_1^*, \boldsymbol{k}_2^*)$ to a legitimate decrypter.

Construction

PKFP-IPE.Decrypt$\left(\mathrm{PP}, \mathrm{CT}_{\vec{x}} = (\boldsymbol{c}_1, \boldsymbol{c}_2), \mathrm{SK}_{\vec{y}} = (\boldsymbol{k}_1^*, \boldsymbol{k}_2^*)\right)$

1. It computes

$$\begin{aligned} T_1 &= E(\boldsymbol{c}_1, \boldsymbol{k}_1^*), \\ T_2 &= E'(\boldsymbol{c}_2, \boldsymbol{k}_2^*). \end{aligned}$$

2. Attempt to determine a value $m \in \mathbb{Z}_p$ such that $T_2^m = T_1$ as elements of $\mathbb{G}_T$ by checking a specified polynomial-size range of possible values. If successful, output m. Otherwise output $\perp$.

Remark: The polynomial running time of our decryption algorithm is guaranteed by restricting the output to lie within a fixed polynomial-size range.

Correctness

- For any $\mathrm{CT}_{\vec{x}} = (\boldsymbol{c}_1, \boldsymbol{c}_2)$ and any $\mathrm{SK}_{\vec{y}} = (\boldsymbol{k}_1^*, \boldsymbol{k}_2^*)$, we have

$$\begin{aligned} T_1 = \; E(\boldsymbol{c}_1, \boldsymbol{k}_1^*) = \; e(g_1, g_2)^{\alpha\gamma\langle\vec{x},\vec{y}\rangle}, \\ T_2 = \; E'(\boldsymbol{c}_2, \boldsymbol{k}_2^*) = \; e(g_1, g_2)^{\alpha\gamma}. \end{aligned}$$

- This follows from the expressions of $\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{k}_1^*, \boldsymbol{k}_2^*$ together with the fact that $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{D}, \mathbb{D}^*)$ are *dual orthonormal* bases.

- Thus if $\langle\vec{x}, \vec{y}\rangle$ is contained in the specified polynomial-size range of possible values that the decryption algorithm checks, it would output $\langle\vec{x}, \vec{y}\rangle$ as desired.

Security Statement

Theorem

Our PKFP-IPE *scheme is secure as per the strongest indistinguishability-based function-privacy model of Brakerski and Segev (TCC 2014) under the* SXDH *assumption.*

Symmetric External Diffie-Hellman (SXDH) Assumption

- It is hard to distinguish between the distributions

$$\varrho_\beta = ((p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e), g_1^\mu, g_1^\nu, \Re_\beta,) \text{ for } \beta \in \{0, 1\}$$

such that

  - $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \xleftarrow{\$} \mathcal{G}_{\mathsf{ABPG}}(1^\lambda)$,
  - $\mu, \nu \xleftarrow{\$} \mathbb{Z}_p$,
  - $\Re_\beta = g_1^{\mu\nu+r}$ where $r = 0$ or $r \xleftarrow{\$} \mathbb{Z}_p$ according as $\beta = 0$ or 1 respectively.

- The same is true for the analogous distributions obtained from switching the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$.

Our Proof Idea

- We design our hybrid argument in a non-trivial way to use the following information theoretic property of DPVS:

Lemma (Okamoto and Takashima (ASIACRYPT 2012))

For $\tau \in \mathbb{Z}_p$, *let* $\mathbb{S}_\tau = \{(\vec{\chi}, \vec{\vartheta}) \mid \langle \vec{\chi}, \vec{\vartheta} \rangle = \tau\} \subset \mathbb{Z}_p^n \times \mathbb{Z}_p^n$, *where* p *is a prime integer and* n *is some positive integer. For all* $(\vec{\chi}, \vec{\vartheta}) \in \mathbb{S}_\tau$, *for all* $(\vec{\zeta}, \vec{v}) \in \mathbb{S}_\tau$,

$$\Pr[\vec{\chi} \cdot \boldsymbol{F} = \vec{\zeta} \bigwedge \vec{\vartheta} \cdot \boldsymbol{F}^* = \vec{v}] = \Pr[\vec{\chi} \cdot \boldsymbol{F}^* = \vec{\zeta} \bigwedge \vec{\vartheta} \cdot \boldsymbol{F} = \vec{v}] = 1/\sharp\mathbb{S}_\tau,$$

where $\boldsymbol{F} \xleftarrow{\$} \mathsf{GL}(n, \mathbb{Z}_p)$, $\boldsymbol{F}^* = (\boldsymbol{F}^\intercal)^{-1}$, *and for any set* A, $\sharp A$ *denotes the cardinality of the set* A.

- We begin our hybrid game transition by changing the form of the queried ciphertexts and instead of finishing it off completely, at some appropriate point, we initiate change in the queried functional keys.
- Since then functional keys and ciphertexts change hand in hand.

Communication and Storage Comparison

PKFP-IPE	Security	Complexity Assumption	$\lvert\text{MSK}\rvert$	$\lvert\text{CT}_{\vec{x}}\rvert$	$\lvert\text{SK}_{\vec{y}}\rvert$
[BJK15]	weak function-hiding	SXDH	$8n^2 + 8$ in $\mathbb{Z}_p$	$2n + 2$ in $\mathbb{G}_1$	$2n + 2$ in $\mathbb{G}_2$
Ours	strong function-hiding	SXDH	$8n^2 + 12n + 28$ in $\mathbb{Z}_p$	$4n + 8$ in $\mathbb{G}_1$	$4n + 8$ in $\mathbb{G}_2$

[BJK15]: Allison Bishop, Abhishek Jain, and Lucas Kowalczyk. ASIACRYPT 2015.

Computation Comparison

PKFP-IPE	PKFP-IPE.Encrypt	PKFP-IPE.KeyGen	PKFP-IPE.Decrypt
[BJK15]	$2n+2$ exp. in $\mathbb{G}_1$	$2n+2$ exp. in $\mathbb{G}_2$	$2n+2$ pairings
Ours	$4n+8$ exp. in $\mathbb{G}_1$	$4n+8$ exp. in $\mathbb{G}_2$	$4n+8$ pairings

[BJK15]: Allison Bishop, Abhishek Jain, and Lucas Kowalczyk. ASIACRYPT 2015.

Summary and Future Scope

- We have presented the *first non-generic* private key FE scheme for the inner product functionality achieving the *strongest indistinguishability-based* notion of function privacy, namely, the *full-hiding security*.
- Our construction has utilized the standard asymmetric bilinear pairing group of prime order and has derived its security from the SXDH assumption.
- A significant future direction of research in this area would be to explore *simulation-based* notion of function privacy in the context of IPE in the private key setting.

Thanking Note