On the Black-Box Complexity of Private-Key IPFE

Mohammad Hajiabadi[1] Roman Langrehr[2] Adam O'Neill[3] Mingyuan Wang[4]

[1]University of Waterloo

[2]ETH Zurich

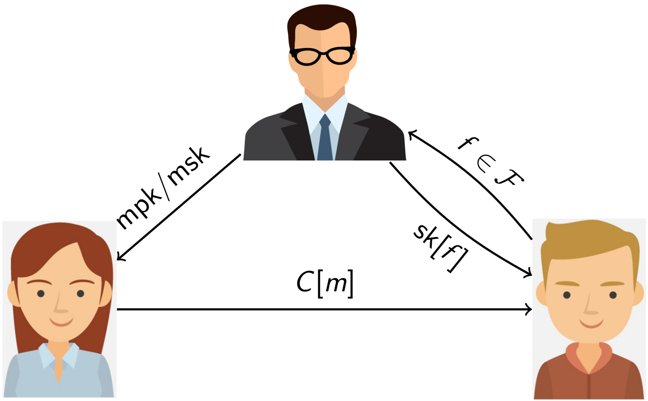
[3]Manning CICS, University of Massachusetts Amherst

[4]NYU Shanghai

2024-12-05

Functional Encryption [SW05, O'N10, BSW11]

$$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{MKGen}(1^\kappa)$$
$$\mathsf{sk}[f] \leftarrow \mathsf{KGen}(\mathsf{msk}, f)$$

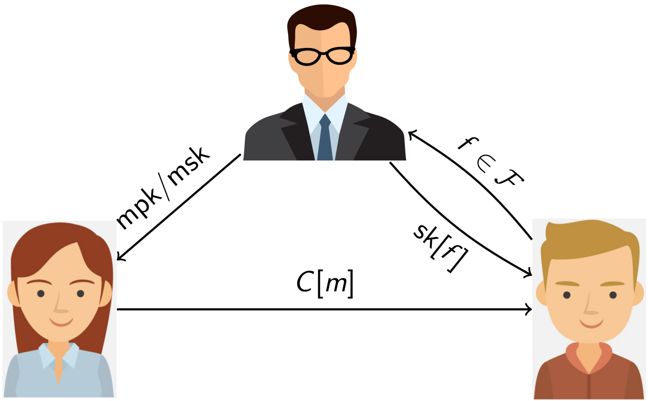


$C[m] \leftarrow \mathsf{Enc}(\mathsf{mpk}/\mathsf{msk}, m)$

$f(m) \leftarrow \mathsf{Dec}(\mathsf{sk}[f], C[m])$

Functional Encryption [SW05, O'N10, BSW11]

$$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{MKGen}(1^\kappa)$$
$$\mathsf{sk}[f] \leftarrow \mathsf{KGen}(\mathsf{msk}, f)$$



$\mathcal{A}$ gets

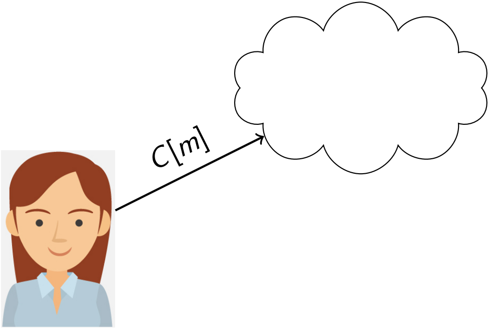
- mpk or Enc oracle,
- $C(m^*)$,
- $\mathsf{sk}[f_1], \ldots, \mathsf{sk}[f_n]$

and learns only $f_1(m^*), \ldots, f_n(m^*)$.

$$C[m] \leftarrow \mathsf{Enc}(\mathsf{mpk}/\mathsf{msk}, m)$$

$$f(m) \leftarrow \mathsf{Dec}(\mathsf{sk}[f], C[m])$$

Application of sk-FE



$$\mathsf{msk} \leftarrow \mathsf{MKGen}(1^\kappa)$$
$$C[m] \leftarrow \mathsf{Enc}(\mathsf{msk}, m)$$

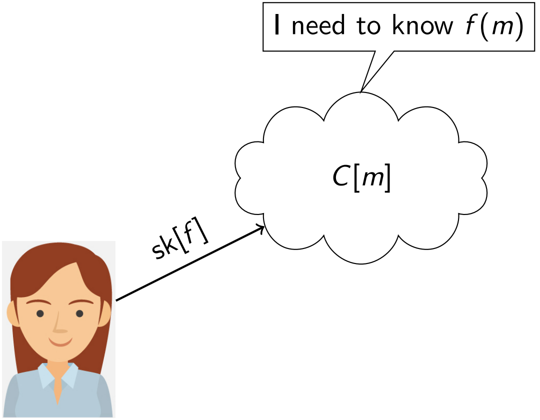
Application of sk-FE

I need to know $f(m)$

$C[m]$

msk

Application of sk-FE

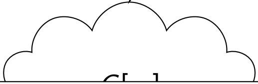


msk

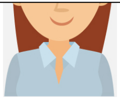
$\mathsf{sk}[f] \leftarrow \mathsf{KGen}(\mathsf{msk}, f)$

Application of sk-FE

I need to know $f(m)$



Is sk-FE easier to build than pk-FE?



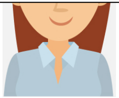
msk

$\mathsf{sk}[f] \leftarrow \mathsf{KGen}(\mathsf{msk}, f)$

Application of sk-FE

I need to know $f(m)$

Is sk-FE easier to build than pk-FE?
What are the minimal assumptions for sk-FE?



msk

$\mathsf{sk}[f] \leftarrow \mathsf{KGen}(\mathsf{msk}, f)$

Bounded-collusion

There is an a-priori bound on the number of functional secret keys that $\mathcal{A}$ can get.

$$\text{IND-CPA secure PKE} \iff \text{bounded-collusion pk-FE [SS10, GVW12, AV19]}$$

Bounded-collusion

There is an a-priori bound on the number of functional secret keys that $\mathcal{A}$ can get.

$$\text{IND-CPA secure PKE} \Longleftrightarrow \text{bounded-collusion pk-FE [SS10, GVW12, AV19]}$$

$$\text{IND-CPA secure SKE} \Longleftrightarrow \text{bounded-collusion sk-FE [AV19]}$$

pk vs. sk

For general functions: sk-FE $\Longleftrightarrow$ pk-FE ($\Longleftrightarrow$ iO)[BV15, AJ15]

pk vs. sk

$$\text{For general functions: sk-FE} \iff \text{pk-FE } (\iff \text{iO})\text{[BV15, AJ15]}$$

$$\text{sk additively HE} \iff \text{pk additively HE [Rot11]}$$

$$\text{IND-CPA PKE} + \text{circular SKE} \implies \text{KDM-secure PKE [KM20]}$$

pk vs. sk

For general functions: sk-FE $\Longleftrightarrow$ pk-FE ($\Longleftrightarrow$ iO)[BV15, AJ15]

sk additively HE $\Longleftrightarrow$ pk additively HE [Rot11]

IND-CPA PKE $+$ circular SKE $\Longrightarrow$ KDM-secure PKE [KM20]

What about FE for specific functionalities?

Functionalities

	Point functions (IBE) $f_{\mathrm{id}'}(\mathrm{id}, m) = \begin{cases} m & \text{if } id = id' \\ \perp & \text{otherwise} \end{cases}$	
pk	✓ pairings, lattices, and more ✗ generic groups [PRV12, SS21, Zha22] ✗ TDP (BB use) [BPR$^+$08]	
sk		

Functionalities

	Point functions (IBE) $$f_{\mathrm{id}'}(\mathrm{id}, m) = \begin{cases} m & \text{if } id = id' \\ \perp & \text{otherwise} \end{cases}$$	
pk	✓ pairings, lattices, and more ✗ generic groups [PRV12, SS21, Zha22] ✗ TDP (BB use) [BPR+08]	
sk	✓ OWF (trivial)	

Functionalities

	Point functions (IBE) $f_{\mathrm{id}'}(\mathrm{id}, m) = \begin{cases} m & \text{if } id = id' \\ \perp & \text{otherwise} \end{cases}$	Inner product (IPFE) $f_{\mathbf{y}}(\mathbf{x}) = \langle \mathbf{y}, \mathbf{x} \rangle \bmod q$
pk	✓ pairings, lattices, and more ✗ generic groups [PRV12, SS21, Zha22] ✗ TDP (BB use) [BPR+08]	
sk	✓ OWF (trivial)	

Functionalities

	Point functions (IBE) $f_{\mathrm{id}'}(\mathrm{id}, m) = \begin{cases} m & \text{if } id = id' \\ \perp & \text{otherwise} \end{cases}$	Inner product (IPFE) $f_{\mathbf{y}}(\mathbf{x}) = \langle \mathbf{y}, \mathbf{x} \rangle \bmod q$
pk	✓ pairings, lattices, and more ✗ generic groups [PRV12, SS21, Zha22] ✗ TDP (BB use) [BPR+08]	pk-IPFE: ✓ groups*, lattices* [ABDP15], class groups [CLT18]
sk	✓ OWF (trivial)	

*for restricted $\mathbf{x}/\mathbf{y}$

Functionalities

	Point functions (IBE) $f_{\mathrm{id}'}(\mathrm{id}, m) = \begin{cases} m & \text{if } id = id' \\ \perp & \text{otherwise} \end{cases}$	Inner product (IPFE) $f_{\mathbf{y}}(\mathbf{x}) = \langle \mathbf{y}, \mathbf{x} \rangle \bmod q$
pk	✓ pairings, lattices, and more ✗ generic groups [PRV12, SS21, Zha22] ✗ TDP (BB use) [BPR+08]	pk-IPFE: ✓ groups*, lattices* [ABDP15], class groups [CLT18]
sk	✓ OWF (trivial)	Trivial from pk-IPFE function hiding: pairings [BJK15]

*for restricted $\mathbf{x}/\mathbf{y}$

Functionalities

	Point functions (IBE) $f_{\text{id}'}(\text{id}, m) = \begin{cases} m & \text{if } id = id' \\ \perp & \text{otherwise} \end{cases}$	Inner product (IPFE) $f_{\mathbf{y}}(\mathbf{x}) = \langle \mathbf{y}, \mathbf{x} \rangle \bmod q$
pk	✓ pairings, lattices, and more ✗ generic groups [PRV12, SS21, Zha22] ✗ TDP (BB use) [BPR+08]	pk-IPFE: ✓ groups*, lattices* [ABDP15], class groups [CLT18]
sk	✓ OWF (trivial)	Trivial from pk-IPFE function hiding: pairings [BJK15] Our result: ✗ OWF (BB use)

*for restricted $\mathbf{x}/\mathbf{y}$

Predicate encryption (PE): FE for “all-or-nothing” functionalities

$$
\begin{tikzcd}
\text{Generic groups} \arrow[r, "{[\text{PRV12, SS21, Zha22}]}", "/" marking] & \text{pk-IBE} \arrow[r, "{[\text{GKLM12}]}", "/" marking] & \text{pk-PE for threshold predicates} \\
\text{TDP} \arrow[ur, "{[\text{BPR}^{+}\text{08}]}"', "/" marking] \arrow[rr, "{[\text{KY09}]}"', "/" marking] & & \text{pk-PE for certain predicates}
\end{tikzcd}
$$

Previous FE impossibility results

Predicate encryption (PE): FE for "all-or-nothing" functionalities

$$
\begin{tikzcd}[column sep=huge]
\text{Generic groups} \arrow[r, "/" marking, "{[\mathrm{PRV12},\ \mathrm{SS21},\ \mathrm{Zha22}]}"] & \text{pk-IBE} \arrow[r, "/" marking, "{[\mathrm{GKLM12}]}"] & \text{pk-PE for threshold predicates} \\
\text{TDP} \arrow[ur, "/" marking, "{[\mathrm{BPR}^{+}08]}"'] \arrow[rr, "/" marking, "{[\mathrm{KY09}]}"'] & & \text{pk-PE for certain predicates}
\end{tikzcd}
$$

- Idea: Compressing $\mathsf{pk}_1, \ldots, \mathsf{pk}_{2^\kappa}$ into mpk is impossible

Previous FE impossibility results

Predicate encryption (PE): FE for "all-or-nothing" functionalities

$$
\begin{tikzcd}
\text{Generic groups} \arrow[r, "/" marking, "{[\text{PRV12, SS21, Zha22}]}"] & \text{pk-IBE} \arrow[r, "/" marking, "{[\text{GKLM12}]}"] & \text{pk-PE for threshold predicates} \\
\text{TDP} \arrow[ur, "/" marking, "{[\text{BPR}^{+}\text{08}]}"'] \arrow[rr, "/" marking, "{[\text{KY09}]}"'] & & \text{pk-PE for certain predicates}
\end{tikzcd}
$$

- Idea: Compressing $\mathsf{pk}_1, \ldots, \mathsf{pk}_{2^\kappa}$ into mpk is impossible
  - Possible for secret keys: Generate $\mathsf{sk}_1, \ldots, \mathsf{sk}_{2^\kappa}$ with a PRF $\Rightarrow$ This idea cannot be applied to sk-FE

Previous FE impossibility results

Predicate encryption (PE): FE for "all-or-nothing" functionalities

$$
\begin{tikzcd}
\text{Generic groups} \arrow[r, "/" marking, "{[\text{PRV12, SS21, Zha22}]}"] & \text{pk-IBE} \arrow[r, "/" marking, "{[\text{GKLM12}]}"] & \text{pk-PE for threshold predicates} \\
\text{TDP} \arrow[ur, "/" marking, "{[\text{BPR}^{+}\text{08}]}"'] \arrow[rr, "/" marking, "{[\text{KY09}]}"'] & & \text{pk-PE for certain predicates}
\end{tikzcd}
$$

- Idea: Compressing $\mathsf{pk}_1, \ldots, \mathsf{pk}_{2^\kappa}$ into mpk is impossible
  - Possible for secret keys: Generate $\mathsf{sk}_1, \ldots, \mathsf{sk}_{2^\kappa}$ with a PRF $\Rightarrow$ This idea cannot be applied to sk-FE
- Only for predicate encryption

Our result

Theorem

There exists no unbounded-collusion sk-IPFE in the random oracle model.*

*against unbounded adversaries that can make only polynomially many ROM queries

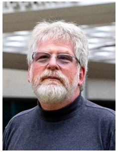
Our result

Theorem

There exists no unbounded-collusion sk-IPFE in the random oracle model.*

*against unbounded adversaries that can make only polynomially many ROM queries

$\Downarrow$ [IR89]



Impagliazzo

Rudich

There exists no unbounded-collusion black-box construction of sk-IPFE from OWFs/CRHFs

Proof overview

Goal: Learn $\langle \mathbf{v}^\star, \mathbf{w} \rangle$

Given: $C[\mathbf{w}] \xleftarrow{\$} \mathsf{Enc}^{O}(\mathsf{msk}, \mathbf{w})$, $\mathsf{sk}[\mathbf{v}_i] \xleftarrow{\$} \mathsf{KGen}^{O}(\mathsf{msk}, \mathbf{v}_i)$ with $\mathbf{v}^\star \notin \mathrm{Span}\left(\mathbf{v}_1, \ldots, \mathbf{v}_t\right)$

Proof overview

Goal: Learn $\langle \mathbf{v}^\star, \mathbf{w} \rangle$

Given: $C[\mathbf{w}] \xleftarrow{\$} \mathsf{Enc}^{O}(\mathsf{msk}, \mathbf{w})$, $\mathsf{sk}[\mathbf{v}_i] \xleftarrow{\$} \mathsf{KGen}^{O}(\mathsf{msk}, \mathbf{v}_i)$ with $\mathbf{v}^\star \notin \mathrm{Span}\left(\mathbf{v}_1, \ldots, \mathbf{v}_t\right)$

$\mathcal{A}$ brute-force searches for $\mathsf{sk}[\mathbf{v}^\star]$

- Compute $\mathsf{Dec}^{O}(\mathsf{sk}[\mathbf{v}^\star], C[\mathbf{w}])$ and check the solution

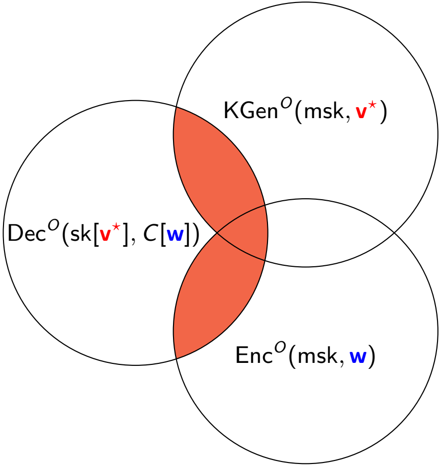
Proof overview

Goal: Learn $\langle \mathbf{v}^\star, \mathbf{w} \rangle$

Given: $C[\mathbf{w}] \xleftarrow{\$} \mathrm{Enc}^{O}(\mathrm{msk}, \mathbf{w})$, $\mathrm{sk}[\mathbf{v}_i] \xleftarrow{\$} \mathrm{KGen}^{O}(\mathrm{msk}, \mathbf{v}_i)$ with $\mathbf{v}^\star \notin \mathrm{Span}\left(\mathbf{v}_1, \ldots, \mathbf{v}_t\right)$

$\mathcal{A}$ brute-force searches for $\mathrm{sk}[\mathbf{v}^\star]$

- Compute $\mathrm{Dec}^{O}(\mathrm{sk}[\mathbf{v}^\star], C[\mathbf{w}])$ and check the solution
  - Self-simulate all ROM queries consistently!



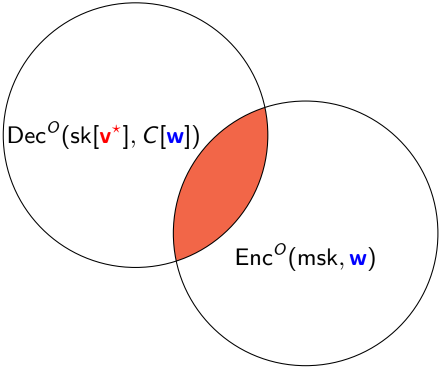
Proof overview

Goal: Learn $\langle \mathbf{v}^\star, \mathbf{w} \rangle$

Given: $C[\mathbf{w}] \xleftarrow{\$} \mathsf{Enc}^{O}(\mathsf{msk}, \mathbf{w})$, $\mathsf{sk}[\mathbf{v}_i] \xleftarrow{\$} \mathsf{KGen}^{O}(\mathsf{msk}, \mathbf{v}_i)$ with $\mathbf{v}^\star \notin \mathsf{Span}(\mathbf{v}_1, \ldots, \mathbf{v}_t)$

$\mathcal{A}$ brute-force searches for $\mathsf{sk}[\mathbf{v}^\star]$

- Compute $\mathsf{Dec}^{O}(\mathsf{sk}[\mathbf{v}^\star], C[\mathbf{w}])$ and check the solution
  - Self-simulate all ROM queries consistently!
  - We ignore $\mathsf{KGen}^{O}(\mathsf{msk}, \mathbf{v}^\star)$ ROM queries here



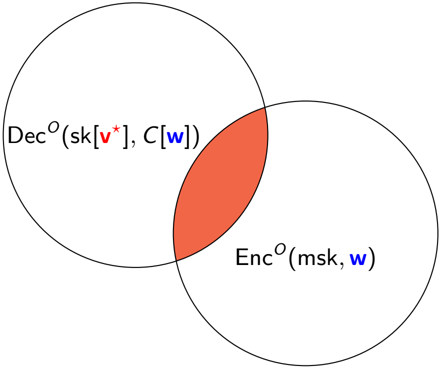
Proof overview

Goal: Learn $\langle \mathbf{v}^\star, \mathbf{w} \rangle$

Given: $C[\mathbf{w}] \xleftarrow{\$} \mathsf{Enc}^{O}(\mathsf{msk}, \mathbf{w})$, $\mathsf{sk}[\mathbf{v}_i] \xleftarrow{\$} \mathsf{KGen}^{O}(\mathsf{msk}, \mathbf{v}_i)$ with $\mathbf{v}^\star \notin \mathrm{Span}\left(\mathbf{v}_1, \ldots, \mathbf{v}_t\right)$

$\mathcal{A}$ brute-force searches for $\mathsf{sk}[\mathbf{v}^\star]$

- Compute $\mathsf{Dec}^{O}(\mathsf{sk}[\mathbf{v}^\star], C[\mathbf{w}])$ and check the solution
  - Self-simulate all ROM queries consistently!
  - We ignore $\mathsf{KGen}^{O}(\mathsf{msk}, \mathbf{v}^\star)$ ROM queries here
  - ROM queries made during $\mathsf{Enc}^{O}(\mathsf{msk}, \mathbf{w})$:



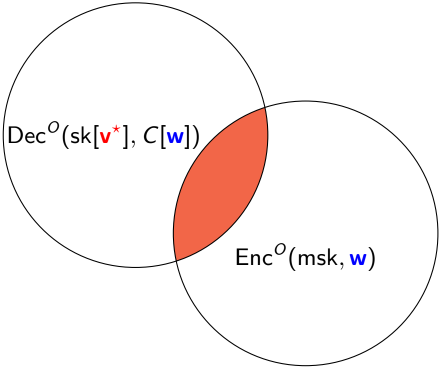
Proof overview

Goal: Learn $\langle \mathbf{v}^\star, \mathbf{w} \rangle$

Given: $C[\mathbf{w}] \xleftarrow{\$} \mathsf{Enc}^{\mathcal{O}}(\mathsf{msk}, \mathbf{w})$, $\mathsf{sk}[\mathbf{v}_i] \xleftarrow{\$} \mathsf{KGen}^{\mathcal{O}}(\mathsf{msk}, \mathbf{v}_i)$ with $\mathbf{v}^\star \notin \mathsf{Span}(\mathbf{v}_1, \ldots, \mathbf{v}_t)$

$\mathcal{A}$ brute-force searches for $\mathsf{sk}[\mathbf{v}^\star]$

- Compute $\mathsf{Dec}^{\mathcal{O}}(\mathsf{sk}[\mathbf{v}^\star], C[\mathbf{w}])$ and check the solution
    - Self-simulate all ROM queries consistently!
    - We ignore $\mathsf{KGen}^{\mathcal{O}}(\mathsf{msk}, \mathbf{v}^\star)$ ROM queries here
    - ROM queries made during $\mathsf{Enc}^{\mathcal{O}}(\mathsf{msk}, \mathbf{w})$:
        - Decrypt $C[\mathbf{w}]$ with many known secret keys $\mathsf{sk}[\mathbf{v}_i]$ using the real RO.



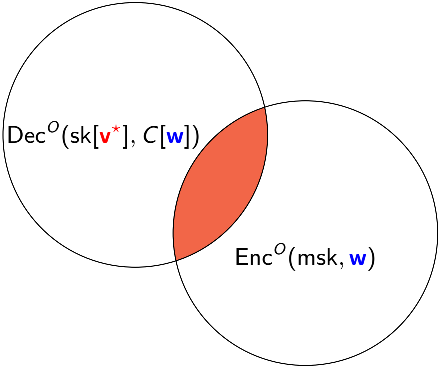
Proof overview

Goal: Learn $\langle \mathbf{v}^\star, \mathbf{w} \rangle$

Given: $C[\mathbf{w}] \xleftarrow{\$} \mathsf{Enc}^O(\mathsf{msk}, \mathbf{w})$, $\mathsf{sk}[\mathbf{v}_i] \xleftarrow{\$} \mathsf{KGen}^O(\mathsf{msk}, \mathbf{v}_i)$ with $\mathbf{v}^\star \notin \mathsf{Span}(\mathbf{v}_1, \ldots, \mathbf{v}_t)$

$\mathcal{A}$ brute-force searches for $\mathsf{sk}[\mathbf{v}^\star]$

- Compute $\mathsf{Dec}^O(\mathsf{sk}[\mathbf{v}^\star], C[\mathbf{w}])$ and check the solution
  - Self-simulate all ROM queries consistently!
  - We ignore $\mathsf{KGen}^O(\mathsf{msk}, \mathbf{v}^\star)$ ROM queries here
  - ROM queries made during $\mathsf{Enc}^O(\mathsf{msk}, \mathbf{w})$:
    - Decrypt $C[\mathbf{w}]$ with many known secret keys $\mathsf{sk}[\mathbf{v}_i]$ using the real RO.
    - Use the RO queries that appeared here.



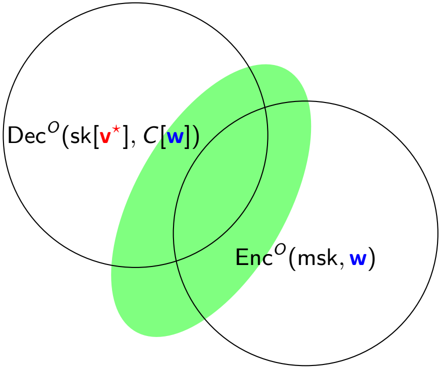
Proof overview

Goal: Learn $\langle \mathbf{v}^\star, \mathbf{w} \rangle$

Given: $C[\mathbf{w}] \xleftarrow{\$} \mathsf{Enc}^{\mathcal{O}}(\mathsf{msk}, \mathbf{w})$, $\mathsf{sk}[\mathbf{v}_i] \xleftarrow{\$} \mathsf{KGen}^{\mathcal{O}}(\mathsf{msk}, \mathbf{v}_i)$ with $\mathbf{v}^\star \notin \mathsf{Span}(\mathbf{v}_1, \ldots, \mathbf{v}_t)$

$\mathcal{A}$ brute-force searches for $\mathsf{sk}[\mathbf{v}^\star]$

- Compute $\mathsf{Dec}^{\mathcal{O}}(\mathsf{sk}[\mathbf{v}^\star], C[\mathbf{w}])$ and check the solution
  - Self-simulate all ROM queries consistently!
  - We ignore $\mathsf{KGen}^{\mathcal{O}}(\mathsf{msk}, \mathbf{v}^\star)$ ROM queries here
  - ROM queries made during $\mathsf{Enc}^{\mathcal{O}}(\mathsf{msk}, \mathbf{w})$:
    - Decrypt $C[\mathbf{w}]$ with many known secret keys $\mathsf{sk}[\mathbf{v}_i]$ using the real RO.
    - Use the RO queries that appeared here.



The Combinatorial Lemma

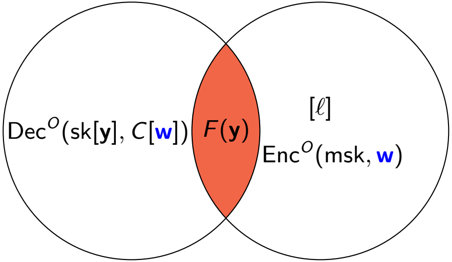
Lemma

For every $F : \mathbb{Z}_q^n \to 2^{[\ell]}$ with polynomial ℓ there exists a polynomial t such that with overwhelming probability

$$F(\mathbf{y}^*) \subseteq \bigcup_{i=1}^{t} F(\mathbf{y}_i),$$

for $\mathbf{y}^*, \mathbf{y}_1, \ldots, \mathbf{y}_t \leftarrow \mathbb{Z}_q^n$ subject to $\mathbf{y}^* \notin \mathrm{Span}(\mathbf{y}_1, \ldots, \mathbf{y}_t)$.

Intuition:



The Combinatorial Lemma

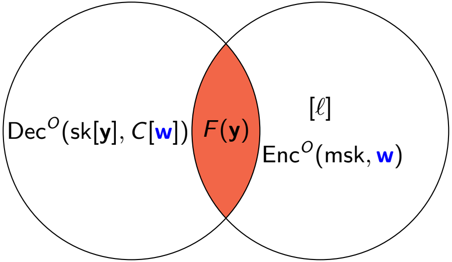
Lemma

For every $F : \mathbb{Z}_q^n \to 2^{[\ell]}$ *with polynomial* ℓ *there exists a polynomial* t *such that with overwhelming probability*

$$F(\mathbf{y}^*) \subseteq \bigcup_{i=1}^{t} F(\mathbf{y}_i),$$

for $\mathbf{y}^*, \mathbf{y}_1, \ldots, \mathbf{y}_t \leftarrow \mathbb{Z}_q^n$ *subject to* $\mathbf{y}^* \notin \mathrm{Span}\left(\mathbf{y}_1, \ldots, \mathbf{y}_t\right)$.

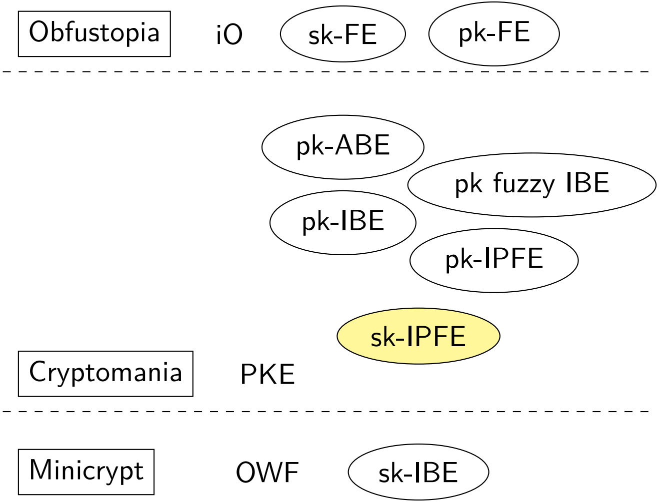
Intuition:



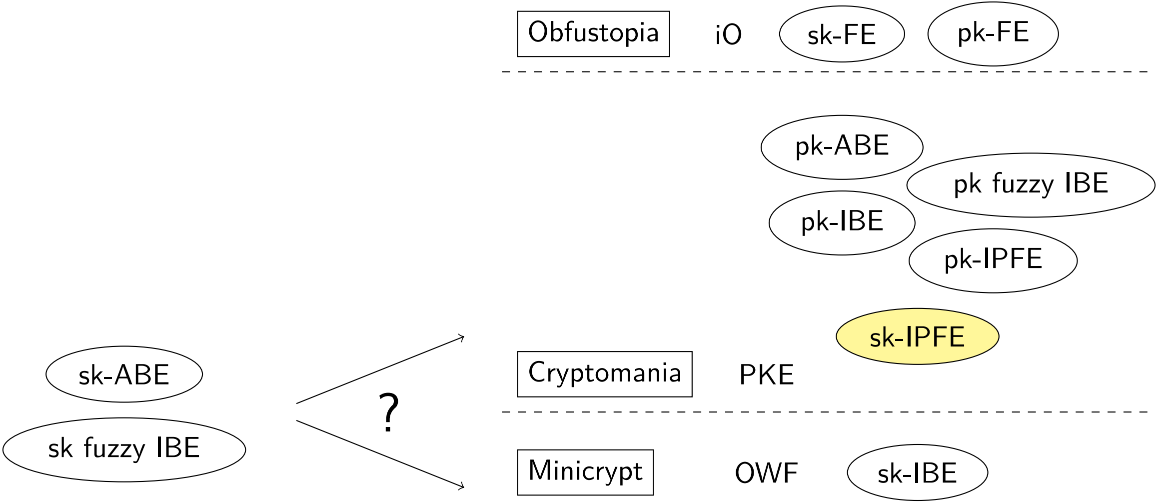
Proofs:

	$q = 2$	arb. prime q
Existential variant (weaker)	Double-counting	
Average-case variant (this)	Fourier	Cauchy-Schwarz

Conclusion & Open problems



Conclusion & Open problems



References I

Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval.
Simple functional encryption schemes for inner products.
In Jonathan Katz, editor, PKC 2015: 18th International Conference on Theory and Practice of Public Key Cryptography, volume 9020 of Lecture Notes in Computer Science, pages 733–751, Gaithersburg, MD, USA, March 30 – April 1, 2015. Springer, Berlin, Heidelberg, Germany.
doi:10.1007/978-3-662-46447-2_33.

Prabhanjan Ananth and Abhishek Jain.
Indistinguishability obfuscation from compact functional encryption.
In Rosario Gennaro and Matthew J. B. Robshaw, editors, Advances in Cryptology – CRYPTO 2015, Part I, volume 9215 of Lecture Notes in Computer Science, pages 308–326, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Berlin, Heidelberg, Germany.
doi:10.1007/978-3-662-47989-6_15.

References II

Prabhanjan Ananth and Vinod Vaikuntanathan.
Optimal bounded-collusion secure functional encryption.
In Dennis Hofheinz and Alon Rosen, editors, TCC 2019: 17th Theory of Cryptography Conference, Part I, volume 11891 of Lecture Notes in Computer Science, pages 174–198, Nuremberg, Germany, December 1–5, 2019. Springer, Cham, Switzerland.
doi:10.1007/978-3-030-36030-6_8.

Allison Bishop, Abhishek Jain, and Lucas Kowalczyk.
Function-hiding inner product encryption.
In Tetsu Iwata and Jung Hee Cheon, editors, Advances in Cryptology – ASIACRYPT 2015, Part I, volume 9452 of Lecture Notes in Computer Science, pages 470–491, Auckland, New Zealand, November 30 – December 3, 2015. Springer, Berlin, Heidelberg, Germany.
doi:10.1007/978-3-662-48797-6_20.

Dan Boneh, Periklis A. Papakonstantinou, Charles Rackoff, Yevgeniy Vahlis, and Brent Waters.
On the impossibility of basing identity based encryption on trapdoor permutations.
In 49th Annual Symposium on Foundations of Computer Science, pages 283–292, Philadelphia, PA, USA, October 25–28, 2008. IEEE Computer Society Press.
doi:10.1109/FOCS.2008.67.

Dan Boneh, Amit Sahai, and Brent Waters.
Functional encryption: Definitions and challenges.
In Yuval Ishai, editor, TCC 2011: 8th Theory of Cryptography Conference, volume 6597 of Lecture Notes in Computer Science, pages 253–273, Providence, RI, USA, March 28–30, 2011. Springer, Berlin, Heidelberg, Germany.
doi:10.1007/978-3-642-19571-6_16.

References IV

Nir Bitansky and Vinod Vaikuntanathan.
Indistinguishability obfuscation from functional encryption.
In Venkatesan Guruswami, editor, 56th Annual Symposium on Foundations of Computer Science, pages 171–190, Berkeley, CA, USA, October 17–20, 2015. IEEE Computer Society Press.
doi:10.1109/FOCS.2015.20.

Guilhem Castagnos, Fabien Laguillaumie, and Ida Tucker.
Practical fully secure unrestricted inner product functional encryption modulo p.
In Thomas Peyrin and Steven Galbraith, editors, Advances in Cryptology – ASIACRYPT 2018, Part II, volume 11273 of Lecture Notes in Computer Science, pages 733–764, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Cham, Switzerland.
doi:10.1007/978-3-030-03329-3_25.

References V

Vipul Goyal, Virendra Kumar, Satyanarayana V. Lokam, and Mohammad Mahmoody.
On black-box reductions between predicate encryption schemes.
In Ronald Cramer, editor, TCC 2012: 9th Theory of Cryptography Conference, volume 7194 of Lecture Notes in Computer Science, pages 440–457, Taormina, Sicily, Italy, March 19–21, 2012. Springer, Berlin, Heidelberg, Germany.
doi:10.1007/978-3-642-28914-9_25.

Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee.
Functional encryption with bounded collusions via multi-party computation.
In Reihaneh Safavi-Naini and Ran Canetti, editors, Advances in Cryptology – CRYPTO 2012, volume 7417 of Lecture Notes in Computer Science, pages 162–179, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Berlin, Heidelberg, Germany.
doi:10.1007/978-3-642-32009-5_11.

References VI

Russell Impagliazzo and Steven Rudich.
Limits on the provable consequences of one-way permutations.
In 21st Annual ACM Symposium on Theory of Computing, pages 44–61, Seattle, WA, USA, May 15–17, 1989. ACM Press.
doi:10.1145/73007.73012.

Fuyuki Kitagawa and Takahiro Matsuda.
Circular security is complete for KDM security.
In Shiho Moriai and Huaxiong Wang, editors, Advances in Cryptology – ASIACRYPT 2020, Part I, volume 12491 of Lecture Notes in Computer Science, pages 253–285, Daejeon, South Korea, December 7–11, 2020. Springer, Cham, Switzerland.
doi:10.1007/978-3-030-64837-4_9.

References VII

Jonathan Katz and Arkady Yerukhimovich.
On black-box constructions of predicate encryption from trapdoor permutations.
In Mitsuru Matsui, editor, Advances in Cryptology – ASIACRYPT 2009, volume 5912 of Lecture Notes in Computer Science, pages 197–213, Tokyo, Japan, December 6–10, 2009. Springer, Berlin, Heidelberg, Germany.
doi:10.1007/978-3-642-10366-7_12.

Adam O'Neill.
Definitional issues in functional encryption.
Cryptology ePrint Archive, Report 2010/556, 2010.
URL: https://eprint.iacr.org/2010/556.

References VIII

Periklis A. Papakonstantinou, Charles W. Rackoff, and Yevgeniy Vahlis.
How powerful are the DDH hard groups?
Cryptology ePrint Archive, Report 2012/653, 2012.
URL: https://eprint.iacr.org/2012/653.

Ron Rothblum.
Homomorphic encryption: From private-key to public-key.
In Yuval Ishai, editor, TCC 2011: 8th Theory of Cryptography Conference, volume 6597 of Lecture Notes in Computer Science, pages 219–234, Providence, RI, USA, March 28–30, 2011. Springer, Berlin, Heidelberg, Germany.
doi:10.1007/978-3-642-19571-6_14.

References IX

Amit Sahai and Hakan Seyalioglu.
Worry-free encryption: functional encryption with public keys.
In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, ACM CCS 2010: 17th Conference on Computer and Communications Security, pages 463–472, Chicago, Illinois, USA, October 4–8, 2010. ACM Press.
doi:10.1145/1866307.1866359.

Gili Schul-Ganz and Gil Segev.
Generic-group identity-based encryption: A tight impossibility result.
In Stefano Tessaro, editor, 2nd Conference on Information-Theoretic Cryptography, ITC 2021, July 23-26, 2021, Virtual Conference, volume 199 of LIPIcs, pages 26:1–26:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

References X

Amit Sahai and Brent R. Waters.
Fuzzy identity-based encryption.
In Ronald Cramer, editor, Advances in Cryptology – EUROCRYPT 2005, volume 3494 of Lecture Notes in Computer Science, pages 457–473, Aarhus, Denmark, May 22–26, 2005. Springer, Berlin, Heidelberg, Germany.
doi:10.1007/11426639_27.

Mark Zhandry.
Augmented random oracles.
In Yevgeniy Dodis and Thomas Shrimpton, editors, Advances in Cryptology – CRYPTO 2022, Part III, volume 13509 of Lecture Notes in Computer Science, pages 35–65, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Cham, Switzerland.
doi:10.1007/978-3-031-15982-4_2.

Pictures

Alice, Bob, and other faces: freepik.com
Devil face: vecteezy.com
Russell Impagliazzo: quantamagazine.org
Steven Rudich: commons.wikimedia.org, Andrej Bauer, CC BY-SA 2.5 SI

Our result (more detail)

Theorem

There exists no unbounded-collusion sk-IPFE for dimension n and modulus q if

- $n \geqslant 3$
- *and q^n is super-polynomial in the security parameter*

in the random oracle model.*

*against unbounded adversaries that can make only polynomially many ROM queries

Our result (more detail)

Theorem

There exists no unbounded-collusion sk-IPFE for dimension n and modulus q if

- $n \geqslant 3$
- *and q^n is super-polynomial in the security parameter*

in the random oracle model.*

*against unbounded adversaries that can make only polynomially many ROM queries

- sk-IPFE for $n = 1$ is just plain SKE

Our result (more detail)

Theorem

There exists no unbounded-collusion sk-IPFE for dimension n and modulus q if

- $n \geqslant 3$
- *and q^n is super-polynomial in the security parameter*

in the random oracle model.*

*against unbounded adversaries that can make only polynomially many ROM queries

- sk-IPFE for $n = 1$ is just plain SKE
- if q^n is polynomial, then we can use $\mathsf{Enc}^{O}(\mathsf{msk}, \mathbf{x}) = (\mathsf{SKE.Enc}^{O}(\mathsf{PRF}(\mathbf{y}), \langle \mathbf{x}, \mathbf{y} \rangle))_{\mathbf{y} \in \mathbb{Z}_q^n}$

Our result (more detail)

Theorem

There exists no unbounded-collusion sk-IPFE for dimension n and modulus q if

- $n \geqslant 3$
- *and q^n is super-polynomial in the security parameter*

in the random oracle model.*

*against unbounded adversaries that can make only polynomially many ROM queries

- sk-IPFE for $n = 1$ is just plain SKE
- if q^n is polynomial, then we can use $\mathsf{Enc}^{O}(\mathsf{msk}, \mathbf{x}) = (\mathsf{SKE.Enc}^{O}(\mathsf{PRF}(\mathbf{y}), \langle \mathbf{x}, \mathbf{y} \rangle))_{\mathbf{y} \in \mathbb{Z}_q^n}$
- bounded-collusion sk-IPFE can be built from OWF [AV19]

Our security game

Challenger		Adversary
$\mathsf{msk} \leftarrow \mathsf{MKGen}(1^\kappa)$		
$b \xleftarrow{\$} \{0,1\}$		
	$\leftarrow$ $\mathbf{v}^\star, \mathbf{v}_1, \ldots, \mathbf{v}_t$ with $\mathbf{v}^\star \notin \mathsf{Span}(\mathbf{v}_1, \ldots, \mathbf{v}_t)$	$t = \mathsf{poly}(\kappa)$
$\mathsf{sk}[\mathbf{v}_i] \leftarrow \mathsf{KGen}^{O}(\mathsf{msk}, \mathbf{v}_i)$ for all $i \in [t]$	$\rightarrow$ $\mathsf{sk}[\mathbf{v}_1], \ldots, \mathsf{sk}[\mathbf{v}_t]$	
$\mathbf{w}_i \xleftarrow{\$} \mathbb{Z}_q^n$		
$C[\mathbf{w}_i] \xleftarrow{\$} \mathsf{Enc}^{O}(\mathsf{msk}, \mathbf{w}_i)$	$\leftarrow$	$\eta = \mathsf{poly}'(\kappa)$ queries
If $b = 0$: $m_i^* := \langle \mathbf{w}_i, \mathbf{v}^\star \rangle$	$\rightarrow$ $C[\mathbf{w}_i], m_i^*$	
If $b = 1$: $m_i^* \xleftarrow{\$} \mathbb{Z}_q$		
	$\leftarrow$ b'	
$b \stackrel{?}{=} b'$		

Proof overview (complete)

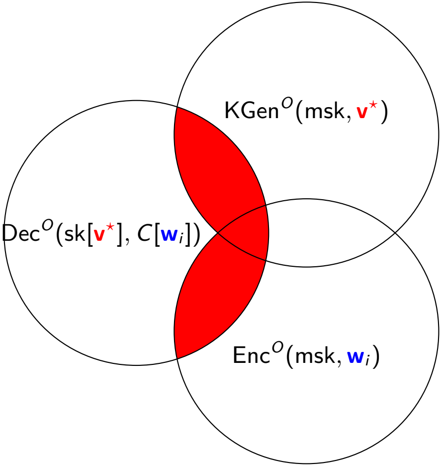
$\mathcal{A}$ brute-force searches for sk[$\mathbf{v}^\star$]

- Check if $\mathsf{Dec}^{O}(\mathsf{sk}[\mathbf{v}^\star], C[\mathbf{w}_i]) = m_i$ for all $i \in [\eta]$
- If such a secret key is found, return $b' = 0$, otherwise $b' = 1$

Proof overview (complete)

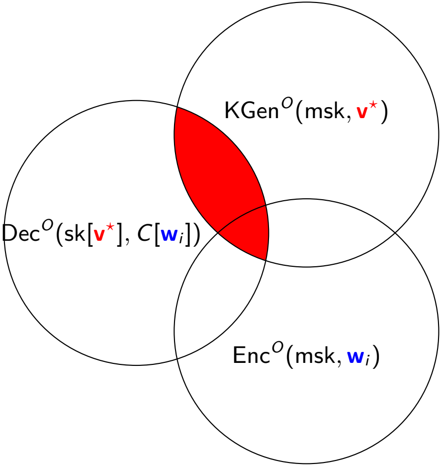
$\mathcal{A}$ brute-force searches for sk[$\mathbf{v}^\star$]

- Check if $\mathsf{Dec}^O(\mathsf{sk}[\mathbf{v}^\star], C[\mathbf{w}_i]) = m_i$ for all $i \in [\eta]$
  - Self-simulate all ROM queries consistently!
- If such a secret key is found, return $b' = 0$, otherwise $b' = 1$



$\mathcal{A}$ brute-force searches for sk[$\mathbf{v}^\star$]

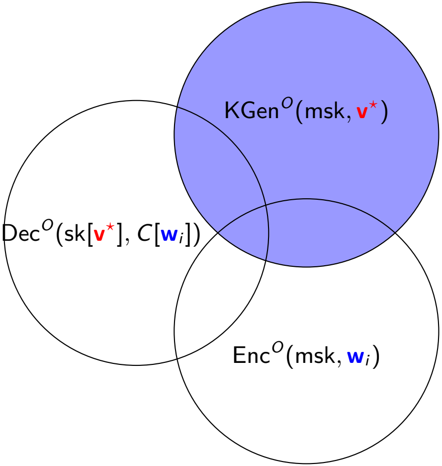
- Check if $\mathsf{Dec}^O(\mathsf{sk}[\mathbf{v}^\star], C[\mathbf{w}_i]) = m_i$ for all $i \in [\eta]$
    - Self-simulate all ROM queries consistently!
    - ROM queries made during $\mathsf{KGen}^O(\mathsf{msk}, \mathbf{v}^\star)$:
- If such a secret key is found, return $b' = 0$, otherwise $b' = 1$



Proof overview (complete)

$\mathcal{A}$ brute-force searches for $\mathsf{sk}[\mathbf{v}^\star]$ and RO Q-A pairs made during $\mathsf{KGen}^O(\mathsf{msk}, \mathbf{v}^\star)$

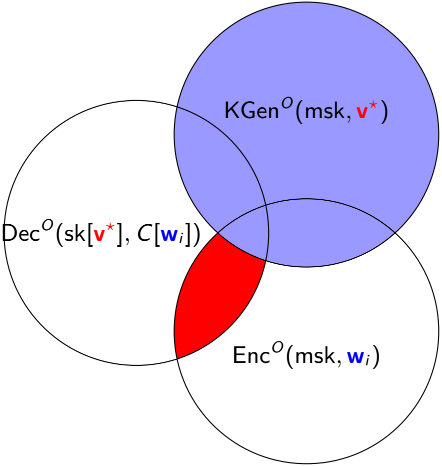
- Check if $\mathsf{Dec}^O(\mathsf{sk}[\mathbf{v}^\star], C[\mathbf{w}_i]) = m_i$ for all $i \in [\eta]$
  - Self-simulate all ROM queries consistently!
  - ROM queries made during $\mathsf{KGen}^O(\mathsf{msk}, \mathbf{v}^\star)$:
    - Just brute-force them, too.
- If such a secret key is found, return $b' = 0$, otherwise $b' = 1$



Proof overview (complete)

$\mathcal{A}$ brute-force searches for $\mathsf{sk}[\mathbf{v}^\star]$ and RO Q-A pairs made during $\mathsf{KGen}^O(\mathsf{msk}, \mathbf{v}^\star)$

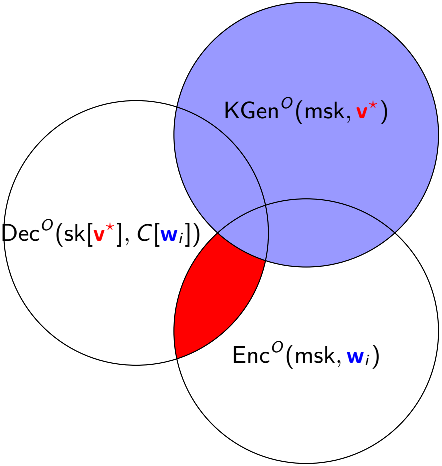
- Check if $\mathsf{Dec}^O(\mathsf{sk}[\mathbf{v}^\star], C[\mathbf{w}_i]) = m_i$ for all $i \in [\eta]$
  - Self-simulate all ROM queries consistently!
  - ROM queries made during $\mathsf{KGen}^O(\mathsf{msk}, \mathbf{v}^\star)$:
    - Just brute-force them, too.
  - ROM queries made during $\mathsf{Enc}^O(\mathsf{msk}, \mathbf{w}_i)$:
- If such a secret key is found, return $b' = 0$, otherwise $b' = 1$



Proof overview (complete)

$\mathcal{A}$ brute-force searches for $\mathsf{sk}[\mathbf{v}^\star]$ and RO Q-A pairs made during $\mathsf{KGen}^O(\mathsf{msk}, \mathbf{v}^\star)$

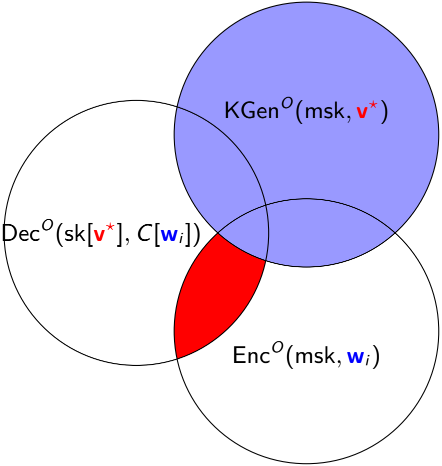
- Check if $\mathsf{Dec}^O(\mathsf{sk}[\mathbf{v}^\star], C[\mathbf{w}_i]) = m_i$ for all $i \in [\eta]$
  - Self-simulate all ROM queries consistently!
  - ROM queries made during $\mathsf{KGen}^O(\mathsf{msk}, \mathbf{v}^\star)$:
    - Just brute-force them, too.
  - ROM queries made during $\mathsf{Enc}^O(\mathsf{msk}, \mathbf{w}_i)$:
    - Decrypt $C[\mathbf{w}_i]$ with many known secret keys $\mathsf{sk}[\mathbf{v}_i]$ using the real RO.
- If such a secret key is found, return $b' = 0$, otherwise $b' = 1$



Proof overview (complete)

$\mathcal{A}$ brute-force searches for sk[$\mathbf{v}^\star$] and RO Q-A pairs made during $\mathsf{KGen}^O(\mathsf{msk}, \mathbf{v}^\star)$

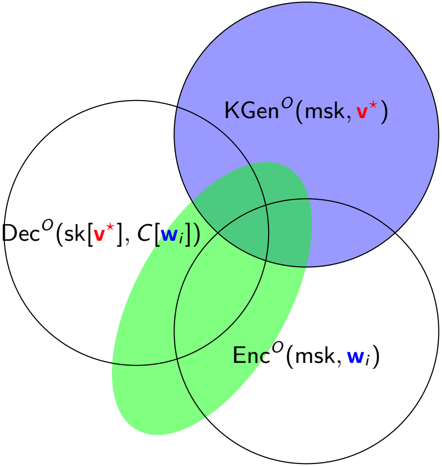
- Check if $\mathsf{Dec}^O(\mathsf{sk}[\mathbf{v}^\star], C[\mathbf{w}_i]) = m_i$ for all $i \in [\eta]$
  - Self-simulate all ROM queries consistently!
  - ROM queries made during $\mathsf{KGen}^O(\mathsf{msk}, \mathbf{v}^\star)$:
    - Just brute-force them, too.
  - ROM queries made during $\mathsf{Enc}^O(\mathsf{msk}, \mathbf{w}_i)$:
    - Decrypt $C[\mathbf{w}_i]$ with many known secret keys sk[$\mathbf{v}_i$] using the real RO.
    - Use the RO queries that appeared here.
- If such a secret key is found, return $b' = 0$, otherwise $b' = 1$



Proof overview (complete)

$\mathcal{A}$ brute-force searches for sk[$\mathbf{v}^\star$] and RO Q-A pairs made during $\mathsf{KGen}^O(\mathsf{msk}, \mathbf{v}^\star)$

- Check if $\mathsf{Dec}^O(\mathsf{sk}[\mathbf{v}^\star], C[\mathbf{w}_i]) = m_i$ for all $i \in [\eta]$
  - Self-simulate all ROM queries consistently!
  - ROM queries made during $\mathsf{KGen}^O(\mathsf{msk}, \mathbf{v}^\star)$:
    - Just brute-force them, too.
  - ROM queries made during $\mathsf{Enc}^O(\mathsf{msk}, \mathbf{w}_i)$:
    - Decrypt $C[\mathbf{w}_i]$ with many known secret keys sk[$\mathbf{v}_i$] using the real RO.
    - Use the RO queries that appeared here.
- If such a secret key is found, return $b' = 0$, otherwise $b' = 1$



The Combinatorial Lemma (formal)

Lemma

Fix $n = n(\kappa)$ *and suppose* $q^{-n} \in \text{negl}(\kappa)$ *and* $n \geqslant 3$. *Let* $\ell = \text{poly}(\kappa)$, q *be a prime number and* $F : \mathbb{Z}_q^n \to 2^{[\ell]}$. *Fix a constant* c. *Then, there exists a polynomial* $t = t(\kappa)$ *such that with probability at least* $1 - \kappa^{-c}$

$$F(\mathbf{y}^*) \subseteq \bigcup_{i=1}^{t} F(\mathbf{y}_i),$$

where $\mathbf{y}^* \leftarrow \mathbb{Z}_q^n$, *and we sample a random* $(n-1)$*-dimensional subspace* V *subject to* $\mathbf{y}^* \notin V$ *and we sample* $\mathbf{y}_1, \ldots, \mathbf{y}_t$ *all uniformly at random from* V.