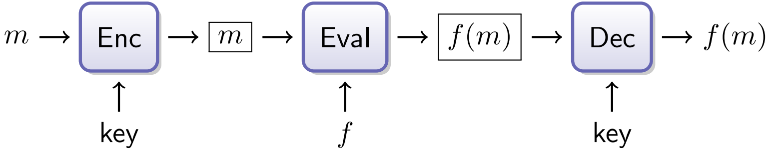
Unexpected Applications of Fully Homomorphic Encryption

Chris Peikert
University of Michigan

Public Key Cryptography
8 May 2023

Fully Homomorphic Encryption [RAD'78,Gentry'09,...]

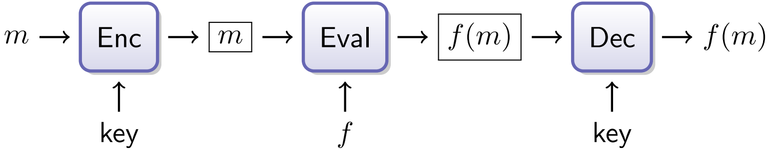
- FHE lets us do this:



Compact: $\boxed{f(m)} \ll |f|$.

Fully Homomorphic Encryption [RAD'78,Gentry'09,...]

- FHE lets us do this:

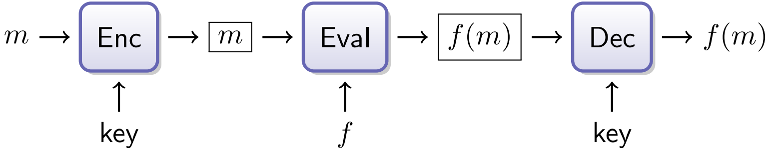


Compact: $\boxed{f(m)} \ll |f|$.

First solved by [Gentry'09], followed by [vDGHV'10,BV'11a,BV'11b,BGV'12,B'12,GSW'13,CKKS'17,...]

Fully Homomorphic Encryption [RAD'78,Gentry'09,. . .]

- FHE lets us do this:



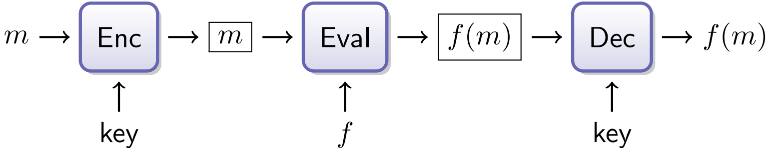
Compact: $\boxed{f(m)} \ll |f|$.

First solved by [Gentry'09], followed by
[vDGHV'10,BV'11a,BV'11b,BGV'12,B'12,GSW'13,CKKS'17,. . .]

A cryptographic "holy grail" with countless applications. . .

Fully Homomorphic Encryption [RAD'78,Gentry'09,. . .]

- FHE lets us do this:



Compact: $\boxed{f(m)} \ll |f|$.

First solved by [Gentry'09], followed by
[vDGHV'10,BV'11a,BV'11b,BGV'12,B'12,GSW'13,CKKS'17,. . .]

A cryptographic "holy grail" with countless applications. . .
some more surprising than others!

Applications of FHE

Less Surprising

- Private cloud computation
- Low-communication MPC
- Code obfuscation
- Quantum FHE, etc. etc.

Applications of FHE

Less Surprising

- Private cloud computation
- Low-communication MPC
- Code obfuscation
- Quantum FHE, etc. etc.

Unexpected (to me at least)

1. Functional commitments for all functions [PPS'21,dCP'23]
2. Instantiating Fiat-Shamir & noninteractive ZK [CCHLRRW'19,PS'19]
3. Attribute-based encryption & much more [BGGHNSVV'14,. . .]

Applications of FHE

Less Surprising

- Private cloud computation
- Low-communication MPC
- Code obfuscation
- Quantum FHE, etc. etc.

Unexpected (to me at least)

1. Functional commitments for all functions [PPS'21,dCP'23]
2. Instantiating Fiat-Shamir & noninteractive ZK [CCHLRRW'19,PS'19]
3. Attribute-based encryption & much more [BGGHNSVV'14,...]

Why? no (computation on) *hidden* data, and/or no *decryption* of it.

Applications of FHE

Less Surprising

- Private cloud computation
- Low-communication MPC
- Code obfuscation
- Quantum FHE, etc. etc.

Unexpected (to me at least)

1. Functional commitments for all functions [PPS'21,dCP'23]
2. Instantiating Fiat-Shamir & noninteractive ZK [CCHLRRW'19,PS'19]
3. Attribute-based encryption & much more [BGGHNSVV'14,...]

Why? no (computation on) *hidden* data, and/or no *decryption* of it.

Instead, *compactness* and *special structure* of FHE scheme are essential!

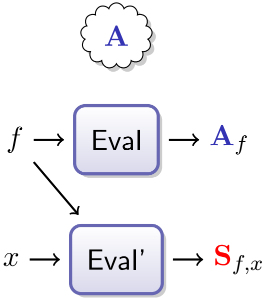
Background and the Central Equation

Homomorphic Computation [GentrySahaiWaters'13,...,deCastroP'23]

Theorem

- For *any* matrix $\mathbf{A}$ and (Boolean) function f, can compute $\mathbf{A}_f$. Then for *any* input x, can compute "short" matrix $\mathbf{S}_{f,x}$ satisfying

$$(\mathbf{A} - \mathsf{Encode}(x)) \cdot \mathbf{S}_{f,x} = \mathbf{A}_f - \mathsf{Encode}(f(x)).$$

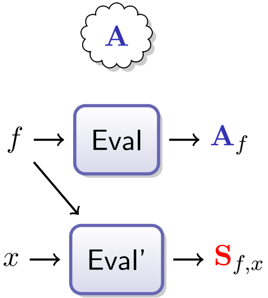


Homomorphic Computation [GentrySahaiWaters'13,...,deCastroP'23]

Theorem

- For *any* matrix $\mathbf{A}$ and (Boolean) function f, can compute $\mathbf{A}_f$. Then for *any* input x, can compute "short" matrix $\mathbf{S}_{f,x}$ satisfying

$$(\mathbf{A} - \text{Encode}(x)) \cdot \mathbf{S}_{f,x} = \mathbf{A}_f - \text{Encode}(f(x)).$$



Implies LWE-Based FHE

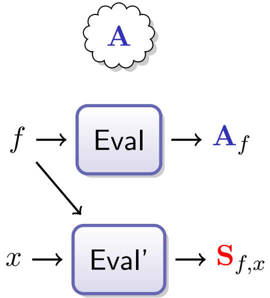
- Ciphertext $\mathbf{A} = \mathbf{B} + \text{Encode}(x)$ where $\mathbf{sB} \approx \mathbf{0}$. Hides x by LWE.

Homomorphic Computation [GentrySahaiWaters'13,...,deCastroP'23]

Theorem

- For *any* matrix $\mathbf{A}$ and (Boolean) function f, can compute $\mathbf{A}_f$. Then for *any* input x, can compute "short" matrix $\mathbf{S}_{f,x}$ satisfying

$$(\mathbf{A} - \mathsf{Encode}(x)) \cdot \mathbf{S}_{f,x} = \mathbf{A}_f - \mathsf{Encode}(f(x)).$$



Implies LWE-Based FHE

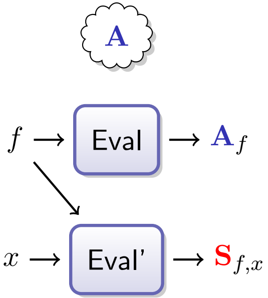
- Ciphertext $\mathbf{A} = \mathbf{B} + \mathsf{Encode}(x)$ where $\mathbf{sB} \approx \mathbf{0}$. Hides x by LWE.
- Homomorphic evaluation of f is $\mathbf{A}_f = \mathbf{B} \cdot \mathbf{S}_{f,x} + \mathsf{Encode}(f(x))$.

Homomorphic Computation [GentrySahaiWaters'13,...,deCastroP'23]

Theorem

- For *any* matrix $\mathbf{A}$ and (Boolean) function f, can compute $\mathbf{A}_f$. Then for *any* input x, can compute "short" matrix $\mathbf{S}_{f,x}$ satisfying

$$(\mathbf{A} - \mathsf{Encode}(x)) \cdot \mathbf{S}_{f,x} = \mathbf{A}_f - \mathsf{Encode}(f(x)).$$



Implies LWE-Based FHE

- Ciphertext $\mathbf{A} = \mathbf{B} + \mathsf{Encode}(x)$ where $\mathbf{s}\mathbf{B} \approx \mathbf{0}$. Hides x by LWE.
- Homomorphic evaluation of f is $\mathbf{A}_f = \mathbf{B} \cdot \mathbf{S}_{f,x} + \mathsf{Encode}(f(x))$.
- Decryption:

$$\mathbf{s}\mathbf{A}_f = \mathbf{s}\mathbf{B} \cdot \mathbf{S}_{f,x} + \mathbf{s} \cdot \mathsf{Encode}(f(x)) \approx \mathbf{s} \cdot \mathsf{Encode}(f(x)).$$

Homomorphic Computation Internals

Goal

$$(\mathbf{A} - \mathsf{Encode}(x)) \cdot \mathbf{S}_{f,x} = \mathbf{A}_f - \mathsf{Encode}(f(x))$$

Homomorphic Computation Internals

Goal

$$(\mathbf{A} - \mathsf{Encode}(x)) \cdot \mathbf{S}_{f,x} = \mathbf{A}_f - \mathsf{Encode}(f(x))$$

How It's Done

- $\mathsf{Encode}(x) = \mathbf{x} \otimes \mathbf{G}$ where $\mathbf{G}^{-1}(\mathbf{Z})$ is short and $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{Z}) = \mathbf{Z}, \forall \mathbf{Z}$.

  By composition, suffices to handle negation, $+$, $\times$.

Homomorphic Computation Internals

Goal

$$(\mathbf{A} - \text{Encode}(x)) \cdot \mathbf{S}_{f,x} = \mathbf{A}_f - \text{Encode}(f(x))$$

How It's Done

- $\text{Encode}(x) = \mathbf{x} \otimes \mathbf{G}$ where $\mathbf{G}^{-1}(\mathbf{Z})$ is short and $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{Z}) = \mathbf{Z}, \forall \mathbf{Z}$.
  By composition, suffices to handle negation, $+$, $\times$.
- **Negation:** define $\mathbf{S}_{\text{neg}} = -\mathbf{I}$ and $\mathbf{A}_{\text{neg}} = \mathbf{A} \cdot \mathbf{S}_{\text{neg}} = -\mathbf{A}$.

Homomorphic Computation Internals

Goal

$$(\mathbf{A} - \mathsf{Encode}(x)) \cdot \mathbf{S}_{f,x} = \mathbf{A}_f - \mathsf{Encode}(f(x))$$

How It's Done

- $\mathsf{Encode}(x) = \mathbf{x} \otimes \mathbf{G}$ where $\mathbf{G}^{-1}(\mathbf{Z})$ is short and $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{Z}) = \mathbf{Z}, \forall \mathbf{Z}$.
  By composition, suffices to handle negation, $+$, $\times$.
- **Negation:** define $\mathbf{S}_{\mathsf{neg}} = -\mathbf{I}$ and $\mathbf{A}_{\mathsf{neg}} = \mathbf{A} \cdot \mathbf{S}_{\mathsf{neg}} = -\mathbf{A}$.
- **Addition:** define $\mathbf{S}_+ = \left[\begin{smallmatrix}\mathbf{I}\\ \mathbf{I}\end{smallmatrix}\right]$ and $\mathbf{A}_+ = \mathbf{A} \cdot \mathbf{S}_+ = \mathbf{A}_1 + \mathbf{A}_2$. Then

$$([\mathbf{A}_1 \mid \mathbf{A}_2] - [x_1\mathbf{G} \mid x_2\mathbf{G}]) \cdot \mathbf{S}_+ = \mathbf{A}_+ - (x_1 + x_2)\mathbf{G}.$$

Homomorphic Computation Internals

Goal

$$(\mathbf{A} - \text{Encode}(x)) \cdot \mathbf{S}_{f,x} = \mathbf{A}_f - \text{Encode}(f(x))$$

How It's Done

- $\text{Encode}(x) = \mathbf{x} \otimes \mathbf{G}$ where $\mathbf{G}^{-1}(\mathbf{Z})$ is short and $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{Z}) = \mathbf{Z}, \forall \mathbf{Z}$.
  By composition, suffices to handle negation, $+$, $\times$.
- **Negation:** define $\mathbf{S}_{\text{neg}} = -\mathbf{I}$ and $\mathbf{A}_{\text{neg}} = \mathbf{A} \cdot \mathbf{S}_{\text{neg}} = -\mathbf{A}$.
- **Addition:** define $\mathbf{S}_{+} = \left[\begin{smallmatrix} \mathbf{I} \\ \mathbf{I} \end{smallmatrix}\right]$ and $\mathbf{A}_{+} = \mathbf{A} \cdot \mathbf{S}_{+} = \mathbf{A}_1 + \mathbf{A}_2$. Then

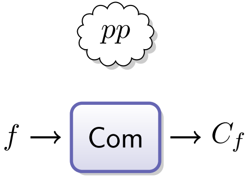
$$([\mathbf{A}_1 \mid \mathbf{A}_2] - [x_1\mathbf{G} \mid x_2\mathbf{G}]) \cdot \mathbf{S}_{+} = \mathbf{A}_{+} - (x_1 + x_2)\mathbf{G}.$$

- **Multiplication:** define $\mathbf{S}_{\times, x_1} = \left[\begin{smallmatrix} \mathbf{G}^{-1}(\mathbf{A}_2) \\ x_1 \mathbf{I} \end{smallmatrix}\right]$ and $\mathbf{A}_{\times} = \mathbf{A}_1 \cdot \mathbf{G}^{-1}(\mathbf{A}_2)$:

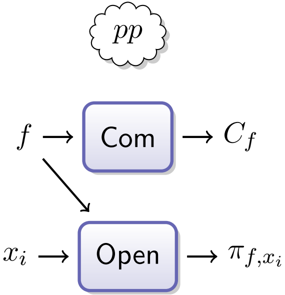
$$([\mathbf{A}_1 \mid \mathbf{A}_2] - [x_1\mathbf{G} \mid x_2\mathbf{G}]) \cdot \mathbf{S}_{\times, x_1} = \mathbf{A}_{\times} - x_1 x_2 \mathbf{G}.$$

Functional Commitments

Functional Commitments [LibertRamannaYung'16]

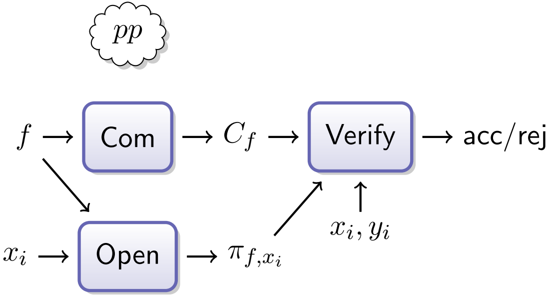


Functional Commitments [LibertRamannaYung'16]



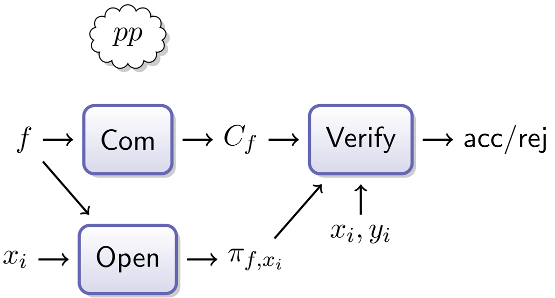
$$|C_f|, |\pi_{f,x_i}| \ll |f|$$

Functional Commitments [LibertRamannaYung'16]



$|C_f|, |\pi_{f,x_i}| \ll |f|$

Functional Commitments [LibertRamannaYung'16]

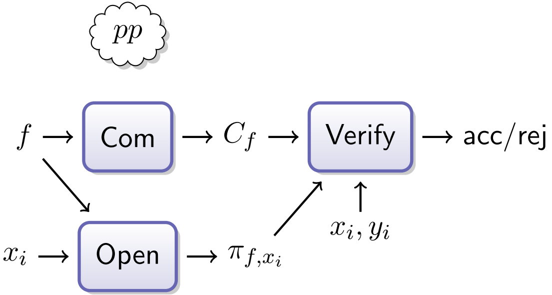


$|C_f|, |\pi_{f,x_i}| \ll |f|$

Applications

- Specializations: vector/key-value/polynomial/linear commitments [LY'10,KZG'10,LRY'16,BBF'19]
- Verifiable outsourced storage/data structures [BGV'11,PSTY'13]
- Accumulators, updateable ZK sets/databases [BdM'93,MRK'03,Lis'05]
- Outsourced committed programs [GSW'23]
- And much more... [CPSZ'18,BFS'20,BDFG'21,...]

Functional Commitments [LibertRamannaYung'16]

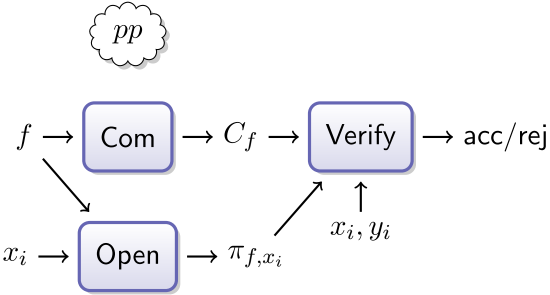


$|C_f|, |\pi_{f,x_i}| \ll |f|$

Basic Security Properties

- **Evaluation binding:** infeasible to find $C^*, x^*, y_0^* \neq y_1^*, \pi_0^*, \pi_1^*$ s.t. Verify$(pp, C^*, x^*, y_b^*, \pi_b^*) =$ acc for $b \in \{0, 1\}$. (No hiding required!)

Functional Commitments [LibertRamannaYung'16]

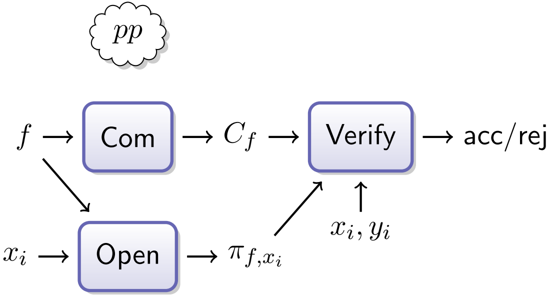


$|C_f|, |\pi_{f,x_i}| \ll |f|$

Basic Security Properties

- **Evaluation binding:** infeasible to find $C^*, x^*, y_0^* \neq y_1^*, \pi_0^*, \pi_1^*$ s.t. Verify$(pp, C^*, x^*, y_b^*, \pi_b^*) =$ acc for $b \in \{0, 1\}$. (No hiding required!)
- **Target binding:** same, but for honestly generated C_f.

Functional Commitments [LibertRamannaYung'16]

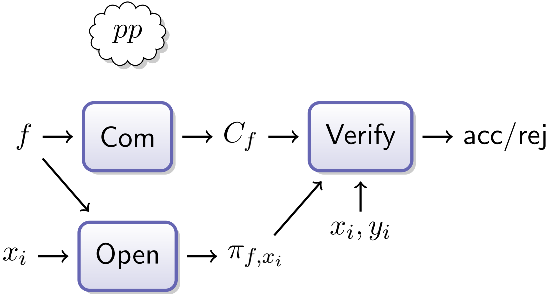


$|C_f|, |\pi_{f,x_i}| \ll |f|$

Basic Security Properties

- **Evaluation binding:** infeasible to find $C^*, x^*, y_0^* \neq y_1^*, \pi_0^*, \pi_1^*$ s.t. Verify$(pp, C^*, x^*, y_b^*, \pi_b^*) =$ acc for $b \in \{0, 1\}$. (No hiding required!)
- **Target binding:** same, but for honestly generated C_f.
- **Zero knowledge:** C_f and π_{f,x_i} reveal nothing except for $x_i, f(x_i)$.

Functional Commitments [LibertRamannaYung'16]

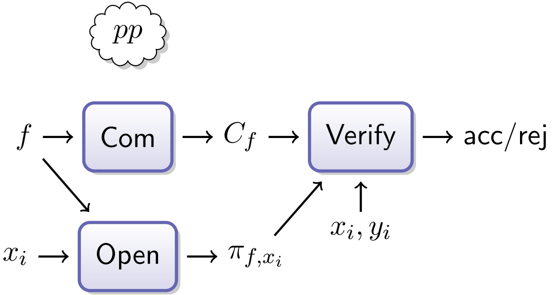


$|C_f|, |\pi_{f,x_i}| \ll |f|$

Constructions

- Were limited to 'linearizable' functions, or relied on non-falsifiable assumptions (SNARGs for NP)

Functional Commitments [LibertRamannaYung'16]

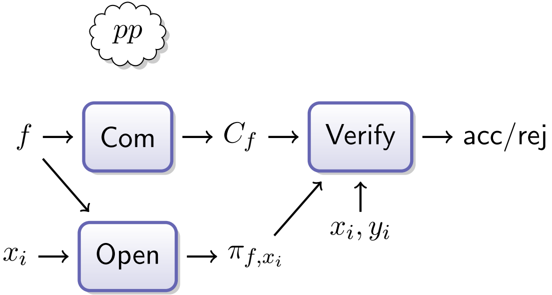


$|C_f|, |\pi_{f,x_i}| \ll |f|$

Constructions

- Were limited to 'linearizable' functions, or relied on non-falsifiable assumptions (SNARGs for NP)
- All functions from SIS, but needs online authority to generate 'opening keys' using trapdoor for pp [PPS'21]

Functional Commitments [LibertRamannaYung'16]

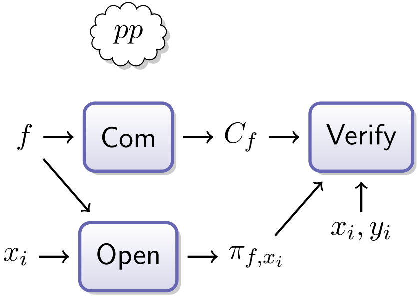


$$|C_f|, |\pi_{f,x_i}| \ll |f|$$

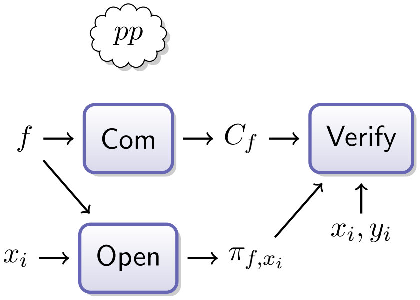
Constructions

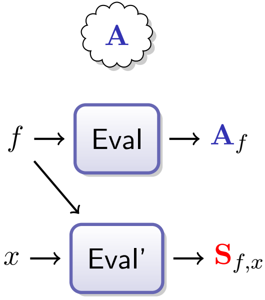
- Were limited to 'linearizable' functions, or relied on non-falsifiable assumptions (SNARGs for NP)
- All functions from SIS, but needs online authority to generate 'opening keys' using trapdoor for pp [PPS'21]
- All functions from SIS, with *transparent* setup: public-coin pp [dCP'23]

Functional Commitments from SIS [deCastroPeikert'23]

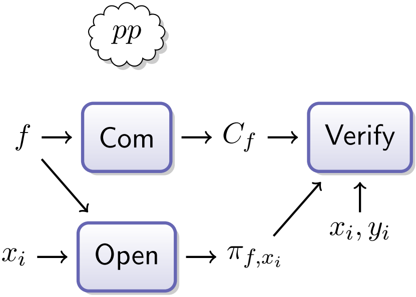


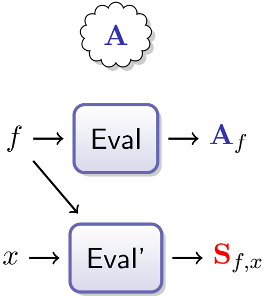
Functional Commitments from SIS [deCastroPeikert'23]





Functional Commitments from SIS [deCastroPeikert'23]

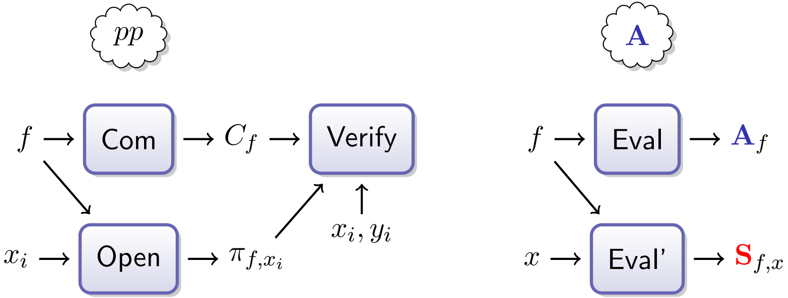




Verification ≡ Central Equation

$$(\mathbf{A} - \text{Encode}(x^*)) \cdot \mathbf{S}^* \stackrel{?}{=} \mathbf{A}^* - \text{Encode}(y^*)$$

Functional Commitments from SIS [deCastroPeikert'23]



Verification $\equiv$ Central Equation

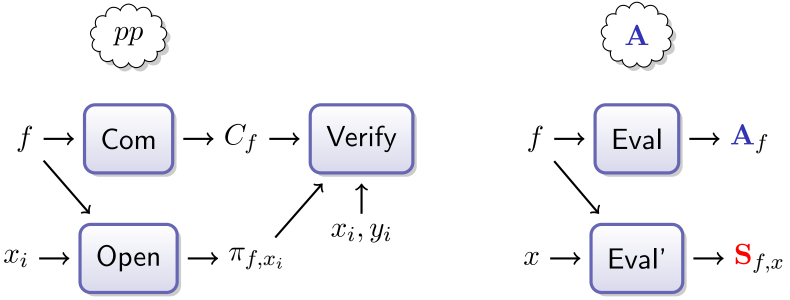
$$(\mathbf{A} - \text{Encode}(x^*)) \cdot \mathbf{S}^* \stackrel{?}{=} \mathbf{A}^* - \text{Encode}(y^*)$$

Evaluation Binding from SIS

- For commitment $\mathbf{A}^*$, valid proofs at x^* for $y_0^* \neq y_1^*$ imply:

$$(\mathbf{A} - \text{Encode}(x^*)) \cdot (\mathbf{S}_0^* - \mathbf{S}_1^*) = \text{Encode}(y_0^* - y_1^*).$$

Functional Commitments from SIS [deCastroPeikert'23]



Verification ≡ Central Equation

$$(\mathbf{A} - \mathrm{Encode}(x^*)) \cdot \mathbf{S}^* \stackrel{?}{=} \mathbf{A}^* - \mathrm{Encode}(y^*)$$

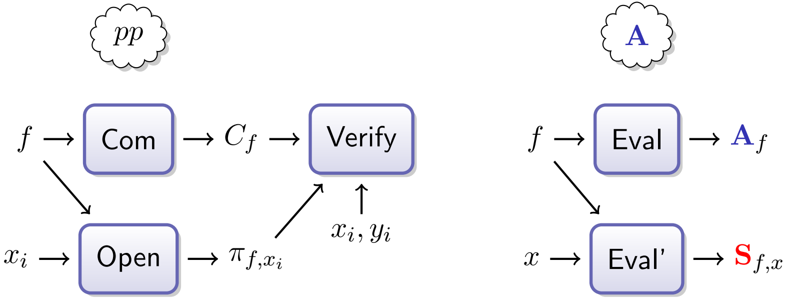
Evaluation Binding from SIS

- For commitment $\mathbf{A}^*$, valid proofs at x^* for $y_0^* \neq y_1^*$ imply:

$$(\mathbf{A} - \mathrm{Encode}(x^*)) \cdot (\mathbf{S}_0^* - \mathbf{S}_1^*) = \mathrm{Encode}(y_0^* - y_1^*).$$

- RHS has short nonzero column $\implies$ solves SIS for $\mathbf{A} - \mathrm{Encode}(x^*)$.

Functional Commitments from SIS [deCastroPeikert'23]

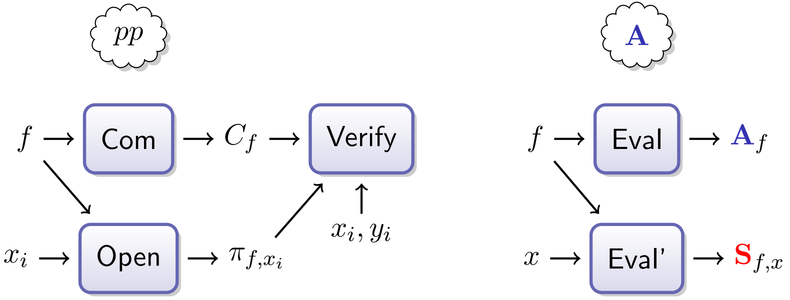


Bonus Features

- Efficient specializations to vector/key-value/linear/polynomial commitments via precomputation and linearity:

$$f(x) = \sum_{\bar{x}} f(\bar{x}) \cdot \mathsf{Eq}_{\bar{x}}(x).$$

Functional Commitments from SIS [deCastroPeikert'23]



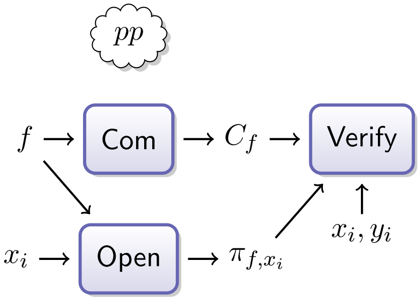
Bonus Features

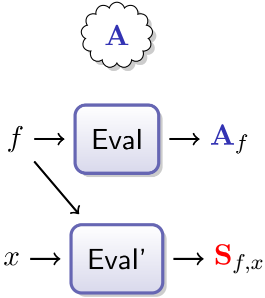
- Efficient specializations to vector/key-value/linear/polynomial commitments via precomputation and linearity:

$$f(x) = \sum_{\bar{x}} f(\bar{x}) \cdot \mathsf{Eq}_{\bar{x}}(x).$$

- Stateless updates by composition: $\mathbf{A}_f \to \mathbf{A}_{g \circ f}$, $\mathbf{S}_{f,x} \cdot \mathbf{S}_{g,f(x)} = \mathbf{S}_{g \circ f,x}$

Functional Commitments from SIS [deCastroPeikert'23]





Bonus Features

- Efficient specializations to vector/key-value/linear/polynomial commitments via precomputation and linearity:

$$f(x) = \sum_{\bar{x}} f(\bar{x}) \cdot \mathsf{Eq}_{\bar{x}}(x).$$

- Stateless updates by composition: $\mathbf{A}_f \to \mathbf{A}_{g \circ f}$, $\mathbf{S}_{f,x} \cdot \mathbf{S}_{g,f(x)} = \mathbf{S}_{g \circ f,x}$
- ZK (w/target binding) via Eval privacy and preimage sampling.

Functional Commitments: Final Thoughts

- Unlike FHE, no hiding or 'structure' needed: public f and x, no sk, unstructured $pp = \mathbf{A}$.

Functional Commitments: Final Thoughts

- Unlike FHE, no hiding or 'structure' needed: public f and x, no sk, unstructured $pp = \mathbf{A}$.

- Compactness is key: single small $\mathbf{A}_f = \mathsf{Eval}(\mathbf{A}, f)$ supports many solutions $\mathbf{S}_{f,x} = \mathsf{Eval'}(\mathbf{A}, f, x)$ to

$$(\mathbf{A} - \mathsf{Encode}(x)) \cdot \mathbf{S}_{f,x} = \mathbf{A}_f - \mathsf{Encode}(f(x)).$$

Functional Commitments: Final Thoughts

- Unlike FHE, no hiding or 'structure' needed: public f and x, no sk, unstructured $pp = \mathbf{A}$.
- Compactness is key: single small $\mathbf{A}_f = \text{Eval}(\mathbf{A}, f)$ supports many solutions $\mathbf{S}_{f,x} = \text{Eval'}(\mathbf{A}, f, x)$ to

$$(\mathbf{A} - \text{Encode}(x)) \cdot \mathbf{S}_{f,x} = \mathbf{A}_f - \text{Encode}(f(x)).$$

- Similar ideas in [WeeWu'23] FCs, but:
  - ⋆ structured CRS (private-key setup);
  - ⋆ swapped Prove/Verify burden;
  - ⋆ smaller proofs;
  - ⋆ based on new, ad-hoc BASIS assumption.

Instantiating Fiat-Shamir and Noninteractive Zero Knowledge

(Noninteractive) Zero Knowledge [BlumDeSantisMicaliPersiano'88]

- Assuming OWFs, every NP language has a ZK proof/argument. [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]

(Noninteractive) Zero Knowledge [BlumDeSantisMicaliPersiano'88]

- Assuming OWFs, every NP language has a ZK proof/argument. [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]
- Interaction is undesirable. What if. . . ?

$\underline{P(x, w)}$ $\qquad\qquad$ $\underline{V(x)}$

$\xrightarrow{\pi}$ acc/rej

(Noninteractive) Zero Knowledge [BlumDeSantisMicaliPersiano'88]

- Assuming OWFs, every NP language has a ZK proof/argument. [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]
- Interaction is undesirable. What if. . . ?

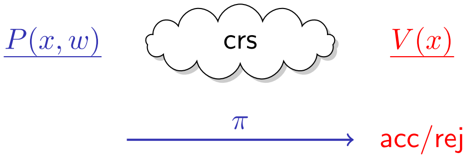
$\underline{P(x, w)}$ $\qquad\qquad$ $\underline{V(x)}$

$\xrightarrow{\pi}$ acc/rej

- In 'plain' model, NIZK = BPP (trivial).

(Noninteractive) Zero Knowledge [BlumDeSantisMicaliPersiano'88]

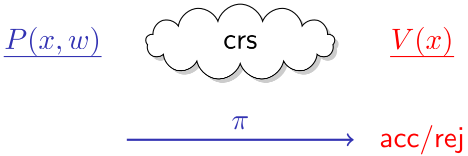
- Assuming OWFs, every NP language has a ZK proof/argument. [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]
- Interaction is undesirable. What if. . . ?



- With random/reference string, NP $\subseteq$ NIZK assuming:

(Noninteractive) Zero Knowledge [BlumDeSantisMicaliPersiano'88]

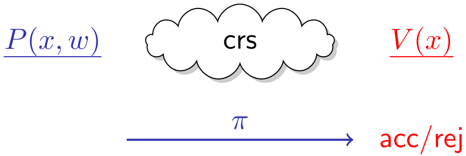
- Assuming OWFs, every NP language has a ZK proof/argument. [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]
- Interaction is undesirable. What if. . . ?



- With random/reference string, NP $\subseteq$ NIZK assuming:
  - ⋆ quadratic residuosity/trapdoor permutations [BDMP'88,FLS'90]
  - ⋆ hard pairing-friendly groups [GrothOstrovskySahai'06]
  - ⋆ indistinguishability obfuscation [SahaiWaters'14]

(Noninteractive) Zero Knowledge [BlumDeSantisMicaliPersiano'88]

- Assuming OWFs, every NP language has a ZK proof/argument. [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]
- Interaction is undesirable. What if...?

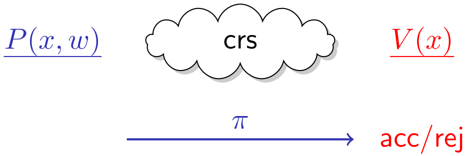


- With random/reference string, NP $\subseteq$ NIZK assuming:
  - ⋆ quadratic residuosity/trapdoor permutations [BDMP'88,FLS'90]
  - ⋆ hard pairing-friendly groups [GrothOstrovskySahai'06]
  - ⋆ indistinguishability obfuscation [SahaiWaters'14]

  Apps: signatures, CCA-secure encryption, cryptocurrencies, ...

(Noninteractive) Zero Knowledge [BlumDeSantisMicaliPersiano'88]

- Assuming OWFs, every NP language has a ZK proof/argument. [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]
- Interaction is undesirable. What if...?

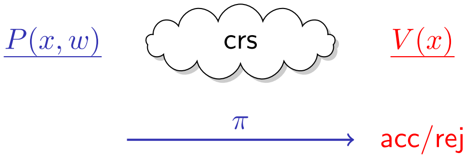


- With random/reference string, NP $\subseteq$ NIZK assuming:
  - ⋆ quadratic residuosity/trapdoor permutations [BDMP'88,FLS'90]
  - ⋆ hard pairing-friendly groups [GrothOstrovskySahai'06]
  - ⋆ indistinguishability obfuscation [SahaiWaters'14]

  Apps: signatures, CCA-secure encryption, cryptocurrencies, ...
- Open [PW'08,PV'08]: 'post-quantum' foundation like lattices/LWE

(Noninteractive) Zero Knowledge [BlumDeSantisMicaliPersiano'88]

- Assuming OWFs, every NP language has a ZK proof/argument. [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]
- Interaction is undesirable. What if. . . ?



- With random/reference string, NP $\subseteq$ NIZK assuming:
  - ⋆ quadratic residuosity/trapdoor permutations [BDMP'88,FLS'90]
  - ⋆ hard pairing-friendly groups [GrothOstrovskySahai'06]
  - ⋆ indistinguishability obfuscation [SahaiWaters'14]

  Apps: signatures, CCA-secure encryption, cryptocurrencies, . . .
- Open [PW'08,PV'08]: 'post-quantum' foundation like lattices/LWE

Theorem [CCHLRRW'19,PS'19]

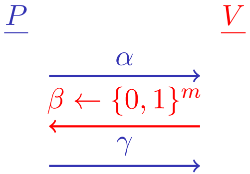
- NP $\subseteq$ NIZK assuming LWE.

Fiat-Shamir Transform [FiatShamir'86]

- A way to remove interaction from a public-coin protocol, via hashing:

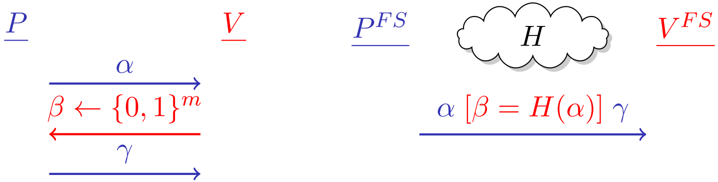
Fiat-Shamir Transform [FiatShamir'86]

- A way to remove interaction from a public-coin protocol, via hashing:



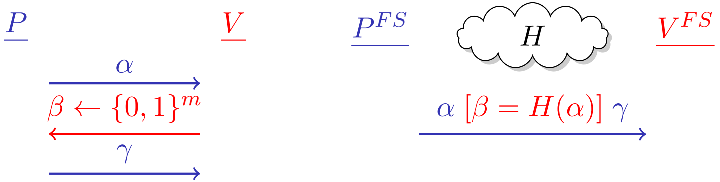
Fiat-Shamir Transform [FiatShamir'86]

- A way to remove interaction from a public-coin protocol, via hashing:



Fiat-Shamir Transform [FiatShamir'86]

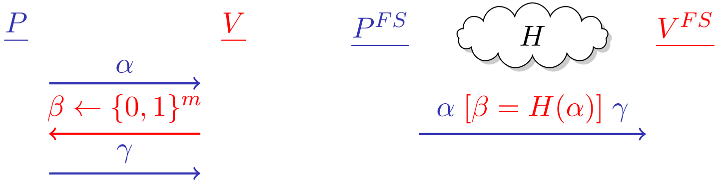
- A way to remove interaction from a public-coin protocol, via hashing:



- Completeness and ZK (for honest V) are easy to preserve.
  For ZK, simulate α, β, γ; then 'program' H so that $H(\alpha) = \beta$.

Fiat-Shamir Transform [FiatShamir'86]

- A way to remove interaction from a public-coin protocol, via hashing:



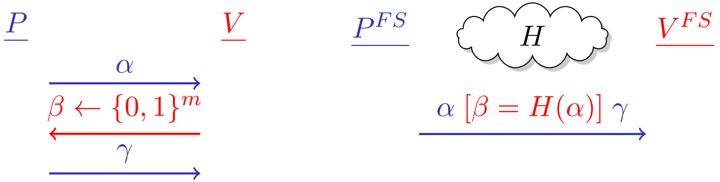
- Completeness and ZK (for honest V) are easy to preserve.
  For ZK, simulate α, β, γ; then 'program' H so that $H(\alpha) = \beta$.

Key Challenge: Soundness

1. Are there α, γ with $\beta = H(\alpha)$ that fool V?

Fiat-Shamir Transform [FiatShamir'86]

- A way to remove interaction from a public-coin protocol, via hashing:



- Completeness and ZK (for honest V) are easy to preserve. For ZK, simulate α, β, γ; then 'program' H so that $H(\alpha) = \beta$.

Key Challenge: Soundness

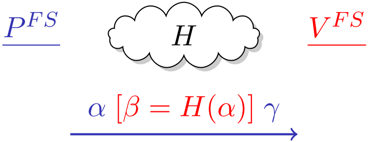
1. Are there α, γ with $\beta = H(\alpha)$ that fool V?
2. Can a cheating P^* find such values, given H? (Proof vs. argument.)

Fiat-Shamir, Soundly [KRR'17,CCRR'18,HL'18,CCHLRRW'19]

$\underline{P^{FS}}$ H $\underline{V^{FS}}$

$\alpha\ [\beta = H(\alpha)]\ \gamma$ $\longrightarrow$

Fiat-Shamir, Soundly [KRR'17,CCRR'18,HL'18,CCHLRRW'19]

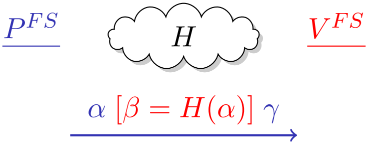


- A correlation-intractable [CGH'98] hash family $\mathcal{H}$ suffices:

  Given $H \leftarrow \mathcal{H}$, hard/impossible to find α s.t. $(\alpha, H(\alpha)) \in R$.

  Relation $R = \{(\alpha, \beta) : \exists\, \gamma$ that fools $V\}$.

Fiat-Shamir, Soundly [KRR'17,CCRR'18,HL'18,CCHLRRW'19]



- A correlation-intractable [CGH'98] hash family $\mathcal{H}$ suffices:

  Given $H \leftarrow \mathcal{H}$, hard/impossible to find α s.t. $(\alpha, H(\alpha)) \in R$.

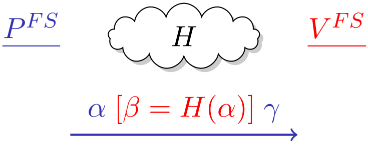
  Relation $R = \{(\alpha, \beta) : \exists\, \gamma$ that fools $V\}$.

Theorem [HL'18,CCHLRRW'19]

- NP $\subseteq$ NIZK assuming a hash family that is CI for all bounded circuits:

  can't find α s.t. $H(\alpha) = C(\alpha)$, $|C| \leq S :=$ poly.

Fiat-Shamir, Soundly [KRR'17,CCRR'18,HL'18,CCHLRRW'19]



- A correlation-intractable [CGH'98] hash family $\mathcal{H}$ suffices:

  Given $H \leftarrow \mathcal{H}$, hard/impossible to find α s.t. $(\alpha, H(\alpha)) \in R$.

  Relation $R = \{(\alpha, \beta) : \exists\, \gamma$ that fools $V\}$.

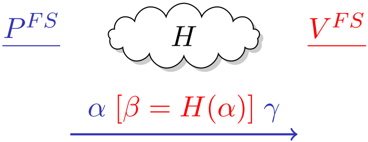
Theorem [HL'18,CCHLRRW'19]

- NP $\subseteq$ NIZK assuming a hash family that is CI for all bounded circuits:

  can't find α s.t. $H(\alpha) = C(\alpha)$, $|C| \leq S :=$ poly.

- Proof idea: for HamCycle^m protocol [FLS'90], each potential α has ≤ 1 'fooling challenge' $\beta \in \{0,1\}^m$ for which V can be fooled.

Fiat-Shamir, Soundly [KRR'17,CCRR'18,HL'18,CCHLRRW'19]



- A correlation-intractable [CGH'98] hash family $\mathcal{H}$ suffices:

  Given $H \leftarrow \mathcal{H}$, hard/impossible to find α s.t. $(\alpha, H(\alpha)) \in R$.

  Relation $R = \{(\alpha, \beta) : \exists\, \gamma$ that fools $V\}$.

Theorem [HL'18,CCHLRRW'19]

- NP $\subseteq$ NIZK assuming a hash family that is CI for all bounded circuits:

  can't find α s.t. $H(\alpha) = C(\alpha)$, $|C| \leq S :=$ poly.

- Proof idea: for HamCycle^m protocol [FLS'90], each potential α has ≤ 1 'fooling challenge' $\beta \in \{0,1\}^m$ for which V can be fooled.

  Such $\beta = C_{sk}(\alpha)$ using a trapdoor sk for decrypting α.

Obtaining Correlation Intractability [CCRR'18,HL'18,CCH+'19,PS'19]

CI Hash Family Construction [PS'19]

- CI for all bounded circuits C via homomorphic computation, assuming SIS/LWE

Obtaining Correlation Intractability [CCRR'18,HL'18,CCH+'19,PS'19]

CI Hash Family Construction [PS'19]

- CI for all bounded circuits C via homomorphic computation, assuming SIS/LWE
- As in [CCH+'19], two 'intractability modes':
  1. Computational (SIS): given $H \leftarrow \mathcal{H}$, hard to find α s.t. $H(\alpha) = C(\alpha)$. Yields statistically ZK argument in uniform random string model.

Obtaining Correlation Intractability [CCRR'18,HL'18,CCH+'19,PS'19]

CI Hash Family Construction [PS'19]

- CI for all bounded circuits C via homomorphic computation, assuming SIS/LWE
- As in [CCH+'19], two 'intractability modes':
  1. Computational (SIS): given $H \leftarrow \mathcal{H}$, hard to find α s.t. $H(\alpha) = C(\alpha)$.
     Yields statistically ZK argument in uniform random string model.
  2. Statistical (LWE): over $H \leftarrow \mathcal{H}_C \overset{c}{\approx} \mathcal{H}$, such α do not exist w.h.p.
     Yields computationally ZK proof in structured reference string model.

CI Hashing from Homomorphic Computation

- Goal: CI for size-S circuits with vector outputs

CI Hashing from Homomorphic Computation

- Goal: CI for size-S circuits with vector outputs

Hash Key: uniformly random matrix $\mathbf{A}$ (that can 'hide' a circuit C)

CI Hashing from Homomorphic Computation

- Goal: CI for size-S circuits with vector outputs

Hash Key: uniformly random matrix $\mathbf{A}$ (that can 'hide' a circuit C)

Evaluation: on input α,

1. Compute $\mathbf{A}_\alpha = \mathsf{Eval}(\mathbf{A}, U_\alpha)$ where $U_\alpha(C) := C(\alpha)$.

CI Hashing from Homomorphic Computation

- Goal: CI for size-S circuits with vector outputs

Hash Key: uniformly random matrix $\mathbf{A}$ (that can 'hide' a circuit C)

Evaluation: on input α,

1. Compute $\mathbf{A}_\alpha = \mathsf{Eval}(\mathbf{A}, U_\alpha)$ where $U_\alpha(C) := C(\alpha)$.
2. 'Inertify': let $\mathbf{a}_\alpha = \mathbf{A}_\alpha \cdot \mathbf{s}^*$, where $\mathsf{Encode}(\mathbf{y}) \cdot \mathbf{s}^* = \mathbf{y}$ for all $\mathbf{y}$.

CI Hashing from Homomorphic Computation

- Goal: CI for size-S circuits with vector outputs

Hash Key: uniformly random matrix $\mathbf{A}$ (that can 'hide' a circuit C)

Evaluation: on input α,

1. Compute $\mathbf{A}_\alpha = \mathsf{Eval}(\mathbf{A}, U_\alpha)$ where $U_\alpha(C) := C(\alpha)$.
2. 'Inertify': let $\mathbf{a}_\alpha = \mathbf{A}_\alpha \cdot \mathbf{s}^*$, where $\mathsf{Encode}(\mathbf{y}) \cdot \mathbf{s}^* = \mathbf{y}$ for all $\mathbf{y}$.
3. Output $\mathbf{a}_\alpha$.

CI Hashing from Homomorphic Computation

- Goal: CI for size-S circuits with vector outputs

Hash Key: uniformly random matrix $\mathbf{A}$ (that can 'hide' a circuit C)

Evaluation: on input α,

1. Compute $\mathbf{A}_\alpha = \mathsf{Eval}(\mathbf{A}, U_\alpha)$ where $U_\alpha(C) := C(\alpha)$.
2. 'Inertify': let $\mathbf{a}_\alpha = \mathbf{A}_\alpha \cdot \mathbf{s}^*$, where $\mathsf{Encode}(\mathbf{y}) \cdot \mathbf{s}^* = \mathbf{y}$ for all $\mathbf{y}$.
3. Output $\mathbf{a}_\alpha$.

Key Point: $\mathbf{a}_\alpha$ can 'hide' a circuit output $\mathbf{y}$ from the same domain, letting the two values 'mix'/cancel out.
Can reason about more than the hidden $\mathbf{y}$ alone.

Proof of Correlation Intractability from SIS/LWE

Hash Key: uniformly random matrix $\mathbf{A}$.

Evaluation: $H(\alpha) := \mathbf{A}_\alpha \cdot \mathbf{s}^* = \mathbf{a}_\alpha$

1. Consider any size-S circuit C with vector output.

Proof of Correlation Intractability from SIS/LWE

Hash Key: uniformly random matrix $\mathbf{A}$.

Evaluation: $H(\alpha) := \mathbf{A}_\alpha \cdot \mathbf{s}^* = \mathbf{a}_\alpha = C(\alpha)$.

1. Consider any size-S circuit C with vector output.
2. Suppose that $\mathcal{A}$, given hash key $\mathbf{A}$, finds α s.t. $H(\alpha) = C(\alpha)$.

Proof of Correlation Intractability from SIS/LWE

Hash Key: uniformly random matrix $\mathbf{A} = \mathbf{B} + \text{Encode}(C)$.

Evaluation: $H(\alpha) := \mathbf{A}_\alpha \cdot \mathbf{s}^* = \mathbf{a}_\alpha = C(\alpha)$.

1. Consider any size-S circuit C with vector output.
2. Suppose that $\mathcal{A}$, given hash key $\mathbf{A}$, finds α s.t. $H(\alpha) = C(\alpha)$.
3. Same holds for hash key $\mathbf{A} = \mathbf{B} + \text{Encode}(C)$, for uniform $\mathbf{B}$.

Proof of Correlation Intractability from SIS/LWE

Hash Key: uniformly random matrix $\mathbf{A} = \mathbf{B} + \text{Encode}(C)$.

Evaluation: $H(\alpha) := \mathbf{A}_\alpha \cdot \mathbf{s}^* = \mathbf{a}_\alpha = C(\alpha)$.

1. Consider any size-S circuit C with vector output.
2. Suppose that $\mathcal{A}$, given hash key $\mathbf{A}$, finds α s.t. $H(\alpha) = C(\alpha)$.
3. Same holds for hash key $\mathbf{A} = \mathbf{B} + \text{Encode}(C)$, for uniform $\mathbf{B}$.

   Let $\mathbf{S}_{\alpha,C} = \text{Eval'}(\mathbf{A}, U_\alpha, C)$. By the Central Equation,

$$\begin{aligned}\mathbf{B} \cdot \mathbf{S}_{\alpha,C} \cdot \mathbf{s}^* &= (\mathbf{A} - \text{Encode}(C)) \cdot \mathbf{S}_{\alpha,C} \cdot \mathbf{s}^* \\ &= (\mathbf{A}_\alpha - \text{Encode}(C(\alpha))) \cdot \mathbf{s}^* \\ &= \mathbf{a}_\alpha - C(\alpha) = \mathbf{0}.\end{aligned}$$

   This solves SIS for instance $\mathbf{B}$!

Proof of Correlation Intractability from SIS/LWE

Hash Key: uniformly random matrix $\mathbf{A} = \mathbf{B} + \text{Encode}(C)$.

Evaluation: $H(\alpha) := \mathbf{A}_\alpha \cdot \mathbf{s}^* = \mathbf{a}_\alpha = C(\alpha)$.

1. Consider any size-S circuit C with vector output.
2. Suppose that $\mathcal{A}$, given hash key $\mathbf{A}$, finds α s.t. $H(\alpha) = C(\alpha)$.
3. Same holds for hash key $\mathbf{A} = \mathbf{B} + \text{Encode}(C)$, for uniform $\mathbf{B}$.

   Let $\mathbf{S}_{\alpha,C} = \text{Eval'}(\mathbf{A}, U_\alpha, C)$. By the Central Equation,

$$\begin{aligned}\mathbf{B} \cdot \mathbf{S}_{\alpha,C} \cdot \mathbf{s}^* &= (\mathbf{A} - \text{Encode}(C)) \cdot \mathbf{S}_{\alpha,C} \cdot \mathbf{s}^* \\ &= (\mathbf{A}_\alpha - \text{Encode}(C(\alpha))) \cdot \mathbf{s}^* \\ &= \mathbf{a}_\alpha - C(\alpha) = \mathbf{0}.\end{aligned}$$

   This solves SIS for instance $\mathbf{B}$!

   (Tweak: can make $H(\alpha) = C(\alpha)$ impossible using LWE matrix $\mathbf{B}$.)

CI Hashing: Final Thoughts

- In security proof, hash key hides a trapdoor sk for homomorphically computing the 'fooling challenge' $\beta = C_{sk}(\alpha)$ in the ZK protocol.

CI Hashing: Final Thoughts

- In security proof, hash key hides a trapdoor sk for homomorphically computing the 'fooling challenge' $\beta = C_{sk}(\alpha)$ in the ZK protocol.

  Yet more power of homomorphic decryption! (Cf. 'bootstrapping')

CI Hashing: Final Thoughts

- In security proof, hash key hides a trapdoor sk for homomorphically computing the 'fooling challenge' $\beta = C_{sk}(\alpha)$ in the ZK protocol.

  Yet more power of homomorphic decryption! (Cf. 'bootstrapping')

- Hidden/computed data is never 'opened' in the construction!

CI Hashing: Final Thoughts

- In security proof, hash key hides a trapdoor sk for homomorphically computing the 'fooling challenge' $\beta = C_{sk}(\alpha)$ in the ZK protocol.

  Yet more power of homomorphic decryption! (Cf. 'bootstrapping')

- Hidden/computed data is never 'opened' in the construction!

  Breaking CI
  $\Rightarrow$ equating (public) hash value and (hidden) computed value
  $\Rightarrow$ cancellation solves SIS via Eval'.

More Applications of Homomorphic Computation

- Attribute-based encryption [BGGHNSVV'14]
  - Homomorphic computation on the public attributes

More Applications of Homomorphic Computation

- Attribute-based encryption [BGGHNSVV'14]
  - Homomorphic computation on the public attributes
- Predicate encryption ('hidden-attribute ABE') [GVW'15]
  - Two layers: HC on (public) FHE-encrypted attributes

More Applications of Homomorphic Computation

- Attribute-based encryption [BGGHNSVV'14]
  - Homomorphic computation on the public attributes
- Predicate encryption ('hidden-attribute ABE') [GVW'15]
  - Two layers: HC on (public) FHE-encrypted attributes
- Fully homomorphic signatures [GVW'15]
  - Homomorphic computation on the public signed data

More Applications of Homomorphic Computation

- Attribute-based encryption [BGGHNSVV'14]
  - Homomorphic computation on the public attributes
- Predicate encryption ('hidden-attribute ABE') [GVW'15]
  - Two layers: HC on (public) FHE-encrypted attributes
- Fully homomorphic signatures [GVW'15]
  - Homomorphic computation on the public signed data
- Privately constrained PRFs [BKM'17,CC'17,BTVW'17,PS'18]
  - Homomorphic computation on the public PRF input

More Applications of Homomorphic Computation

- Attribute-based encryption [BGGHNSVV'14]
  - ⋆ Homomorphic computation on the public attributes
- Predicate encryption ('hidden-attribute ABE') [GVW'15]
  - ⋆ Two layers: HC on (public) FHE-encrypted attributes
- Fully homomorphic signatures [GVW'15]
  - ⋆ Homomorphic computation on the public signed data
- Privately constrained PRFs [BKM'17,CC'17,BTVW'17,PS'18]
  - ⋆ Homomorphic computation on the public PRF input
- . . . your next great idea!

More Applications of Homomorphic Computation

- Attribute-based encryption [BGGHNSVV'14]
  - Homomorphic computation on the public attributes
- Predicate encryption ('hidden-attribute ABE') [GVW'15]
  - Two layers: HC on (public) FHE-encrypted attributes
- Fully homomorphic signatures [GVW'15]
  - Homomorphic computation on the public signed data
- Privately constrained PRFs [BKM'17,CC'17,BTVW'17,PS'18]
  - Homomorphic computation on the public PRF input
- . . . your next great idea!

Thanks! Questions?