Noninteractive Zero Knowledge for NP from Learning With Errors

Chris Peikert   Sina Shiehian

TCS+
1 May 2019

Zero Knowledge [GoldwasserMicaliRackoff'85]

- Zero-knowledge (interactive) proof for language L: allows a prover P to convince a verifier V that some $x \in L$, while revealing nothing else.

Zero Knowledge [GoldwasserMicaliRackoff'85]

- Zero-knowledge (interactive) proof for language L: allows a prover P to convince a verifier V that some $x \in L$, while revealing nothing else.
- Example: 'cut-and-choose' protocol for Graph Isomorphism

$\underline{P(G_0, G_1, \pi)}$	$\underline{V(G_0, G_1)}$
$[G_0 = \pi(G_1)]$	

Zero Knowledge [GoldwasserMicaliRackoff'85]

- Zero-knowledge (interactive) proof for language L: allows a prover P to convince a verifier V that some $x \in L$, while revealing nothing else.
- Example: 'cut-and-choose' protocol for Graph Isomorphism

$\underline{P(G_0, G_1, \pi)}$		$\underline{V(G_0, G_1)}$
$[G_0 = \pi(G_1)]$		
$H = \rho(G_0)$	$\xrightarrow{\quad H \quad}$	

Zero Knowledge [GoldwasserMicaliRackoff'85]

- Zero-knowledge (interactive) proof for language L: allows a prover P to convince a verifier V that some $x \in L$, while revealing nothing else.
- Example: 'cut-and-choose' protocol for Graph Isomorphism

$\underline{P(G_0, G_1, \pi)}$		$\underline{V(G_0, G_1)}$
$[G_0 = \pi(G_1)]$		
$H = \rho(G_0)$	$\xrightarrow{\quad H \quad}$	
	$\xleftarrow{\quad b \leftarrow \{0,1\} \quad}$	("Prove $H \equiv G_b$")

Zero Knowledge [GoldwasserMicaliRackoff'85]

- Zero-knowledge (interactive) proof for language L: allows a prover P to convince a verifier V that some $x \in L$, while revealing nothing else.
- Example: 'cut-and-choose' protocol for Graph Isomorphism

$P(G_0, G_1, \pi)$		$V(G_0, G_1)$
$[G_0 = \pi(G_1)]$		
$H = \rho(G_0)$	H $\longrightarrow$	
	$b \leftarrow \{0,1\}$ $\longleftarrow$	("Prove $H \equiv G_b$")
	$\sigma = \rho \circ \pi^b$ $\longrightarrow$	check $H \stackrel{?}{=} \sigma(G_b)$

Zero Knowledge [GoldwasserMicaliRackoff'85]

- Zero-knowledge (interactive) proof for language L: allows a prover P to convince a verifier V that some $x \in L$, while revealing nothing else.
- Example: 'cut-and-choose' protocol for Graph Isomorphism

$\underline{P(G_0, G_1, \pi)}$		$\underline{V(G_0, G_1)}$
$[G_0 = \pi(G_1)]$		
$H = \rho(G_0)$	$\xrightarrow{H}$	
	$\xleftarrow{b \leftarrow \{0,1\}}$	("Prove $H \equiv G_b$")
	$\xrightarrow{\sigma = \rho \circ \pi^b}$	check $H \stackrel{?}{=} \sigma(G_b)$

1. Complete: if $G_0 \equiv G_1$, then P convinces V.

Zero Knowledge [GoldwasserMicaliRackoff'85]

- Zero-knowledge (interactive) proof for language L: allows a prover P to convince a verifier V that some $x \in L$, while revealing nothing else.
- Example: 'cut-and-choose' protocol for Graph Isomorphism

$P(G_0, G_1, \pi)$		$V(G_0, G_1)$
$[G_0 = \pi(G_1)]$		
$H = \rho(G_0)$	$\xrightarrow{H}$	
	$\xleftarrow{b \leftarrow \{0,1\}}$	("Prove $H \equiv G_b$")
	$\xrightarrow{\sigma = \rho \circ \pi^b}$	check $H \stackrel{?}{=} \sigma(G_b)$

1. Complete: if $G_0 \equiv G_1$, then P convinces V.
2. Sound: if $G_0 \not\equiv G_1$, cheating P^* convinces V with prob $\leq 1/2$.

Zero Knowledge [GoldwasserMicaliRackoff'85]

- Zero-knowledge (interactive) proof for language L: allows a prover P to convince a verifier V that some $x \in L$, while revealing nothing else.
- Example: 'cut-and-choose' protocol for Graph Isomorphism

$P(G_0, G_1, \pi)$		$V(G_0, G_1)$
$[G_0 = \pi(G_1)]$		
$H = \rho(G_0)$	$\xrightarrow{H}$	
	$\xleftarrow{b \leftarrow \{0,1\}}$	("Prove $H \equiv G_b$")
	$\xrightarrow{\sigma = \rho \circ \pi^b}$	check $H \stackrel{?}{=} \sigma(G_b)$

1. Complete: if $G_0 \equiv G_1$, then P convinces V.
2. Sound: if $G_0 \not\equiv G_1$, cheating P^* convinces V with prob $\leq 1/2$.
   Soundness error can be reduced exponentially by (parallel) repetition.

Zero Knowledge [GoldwasserMicaliRackoff'85]

- Zero-knowledge (interactive) proof for language L: allows a prover P to convince a verifier V that some $x \in L$, while revealing nothing else.
- Example: 'cut-and-choose' protocol for Graph Isomorphism

$\underline{P(G_0, G_1, \pi)}$ $\qquad$ $\underline{V(G_0, G_1)}$

$[G_0 = \pi(G_1)]$

$H = \rho(G_0)$ $\quad \xrightarrow{\;H\;}$

$\xleftarrow{\;b \leftarrow \{0,1\}\;}$ $\quad$ ("Prove $H \equiv G_b$")

$\xrightarrow{\;\sigma = \rho \circ \pi^b\;}$ $\quad$ check $H \stackrel{?}{=} \sigma(G_b)$

1. Complete: if $G_0 \equiv G_1$, then P convinces V.
2. Sound: if $G_0 \not\equiv G_1$, cheating P^* convinces V with prob $\leq 1/2$. Soundness error can be reduced exponentially by (parallel) repetition.
3. Zero Knowledge: can simulate (honest) V's view when $G_0 \equiv G_1$.

Zero Knowledge for NP

Theorem [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]

- Assuming OWFs, every NP language has a ZK proof/argument.

Zero Knowledge for NP

Theorem [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]

- Assuming OWFs, every NP language has a ZK proof/argument.
- Applications: identification, secure multiparty computation, ...

Zero Knowledge for NP

Theorem [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]

- Assuming OWFs, every NP language has a ZK proof/argument.
- Applications: identification, secure multiparty computation, ...

Cut-and-choose protocol for Hamiltonian Cycle [FeigeLapidotShamir'90]:

$P(G, \text{cycle } C)$ $V(G)$

Zero Knowledge for NP

Theorem [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]

- Assuming OWFs, every NP language has a ZK proof/argument.
- Applications: identification, secure multiparty computation, ...

Cut-and-choose protocol for Hamiltonian Cycle [FeigeLapidotShamir'90]:

$P(G, \text{cycle } C)$		$V(G)$
$H = \rho(G)$	$\xrightarrow{\{c_{i,j} \leftarrow \mathsf{Com}(h_{i,j})\}, \mathsf{Com}(\rho)}$	

Zero Knowledge for NP

Theorem [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]

- Assuming OWFs, every NP language has a ZK proof/argument.
- Applications: identification, secure multiparty computation, ...

Cut-and-choose protocol for Hamiltonian Cycle [FeigeLapidotShamir'90]:

$P(G, \text{cycle } C)$ $\qquad\qquad\qquad$ $V(G)$

$H = \rho(G)$ $\qquad \xrightarrow{\{c_{i,j} \leftarrow \mathsf{Com}(h_{i,j})\}, \mathsf{Com}(\rho)}$

$\xleftarrow{b \leftarrow \{0,1\}}$

Zero Knowledge for NP

Theorem [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]

- Assuming OWFs, every NP language has a ZK proof/argument.
- Applications: identification, secure multiparty computation, ...

Cut-and-choose protocol for Hamiltonian Cycle [FeigeLapidotShamir'90]:

$P(G, \text{cycle } C)$ $\qquad\qquad$ $V(G)$

$H = \rho(G)$ $\quad$ $\xrightarrow{\{c_{i,j} \leftarrow \mathsf{Com}(h_{i,j})\}, \mathsf{Com}(\rho)}$

$\xleftarrow{b \leftarrow \{0,1\}}$

$\xrightarrow{b = 0 : \text{open all } h_{i,j}, \rho}$ $\quad$ check $H = \rho(G)$

Zero Knowledge for NP

Theorem [GoldreichMicaliWigderson'86,NguyenOngVadhan'06]

- Assuming OWFs, every NP language has a ZK proof/argument.
- Applications: identification, secure multiparty computation, ...

Cut-and-choose protocol for Hamiltonian Cycle [FeigeLapidotShamir'90]:

$P(G, \text{cycle } C)$		$V(G)$
$H = \rho(G)$	$\{c_{i,j} \leftarrow \text{Com}(h_{i,j})\}, \text{Com}(\rho)$ $\longrightarrow$	
	$\longleftarrow$ $b \leftarrow \{0,1\}$	
	$b = 0$: open all $h_{i,j}, \rho$ $\longrightarrow$	check $H = \rho(G)$
	$b = 1$: open $h_{i,j}$ for $(i,j) \in \rho(C)$ $\longrightarrow$	check cycle

Noninteractive Zero Knowledge [BlumDeSantisMicaliPersiano'88]

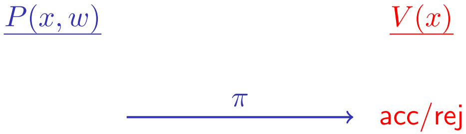
- Interaction is not always possible. What if. . . ?

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

$\xrightarrow{\pi}$ acc/rej

Noninteractive Zero Knowledge [BlumDeSantisMicaliPersiano'88]

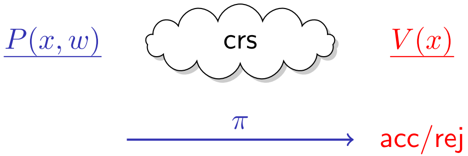
- Interaction is not always possible. What if. . . ?



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

Noninteractive Zero Knowledge [BlumDeSantisMicaliPersiano'88]

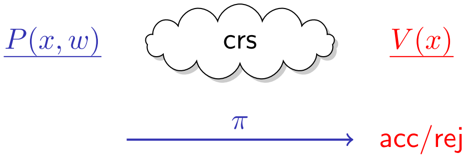
- Interaction is not always possible. What if. . . ?



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

Noninteractive Zero Knowledge [BlumDeSantisMicaliPersiano'88]

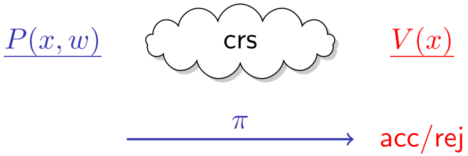
- Interaction is not always possible. What if. . . ?



- With common 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]

- Interaction is not always possible. What if...?

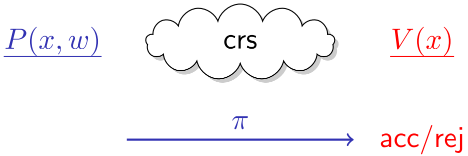


- With common random/reference string, NP $\subseteq$ NIZK assuming:
  - ⋆ quadratic residuosity/trapdoor permutations [BDMP'88,FLS'90]
  - ⋆ hard pairing-friendly groups [GrothOstrovskySahai'06]
  - ⋆ indistinguishability obfuscation [SahaiWaters'14]

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

Noninteractive Zero Knowledge [BlumDeSantisMicaliPersiano'88]

- Interaction is not always possible. What if...?



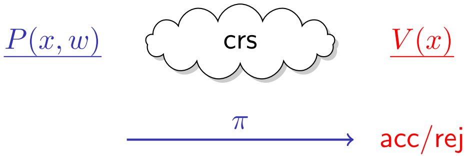
- With common random/reference string, NP $\subseteq$ NIZK assuming:
  - ⋆ quadratic residuosity/trapdoor permutations [BDMP'88,FLS'90]
  - ⋆ hard pairing-friendly groups [GrothOstrovskySahai'06]
  - ⋆ indistinguishability obfuscation [SahaiWaters'14]

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

- Open [PV'08]: a 'post-quantum' foundation like lattices/LWE [Regev'05]

Noninteractive Zero Knowledge [BlumDeSantisMicaliPersiano'88]

- Interaction is not always possible. What if. . . ?



- With common 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 [PV'08]: a 'post-quantum' foundation like lattices/LWE [Regev'05]

Our Main Theorem

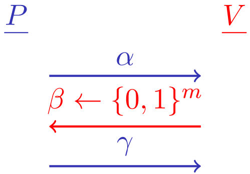
- NP $\subseteq$ NIZK assuming LWE/worst-case lattice problems are hard.

Fiat-Shamir Heuristic [FiatShamir'86]

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

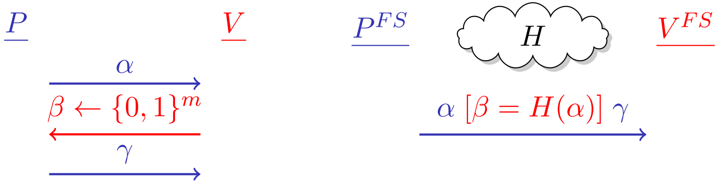
Fiat-Shamir Heuristic [FiatShamir'86]

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



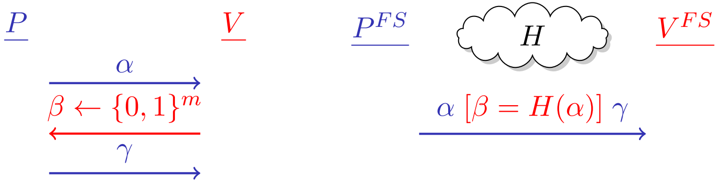
Fiat-Shamir Heuristic [FiatShamir'86]

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



Fiat-Shamir Heuristic [FiatShamir'86]

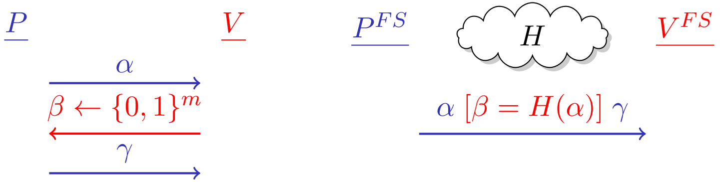
- A way to remove interaction from a public-coin protocol, via hashing:



- 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 Heuristic [FiatShamir'86]

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



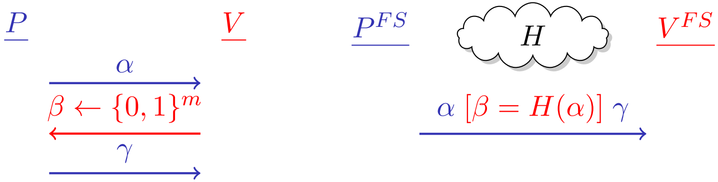
- Completeness and ZK (for honest V) are easy to preserve.
  For ZK, simulate α, β, γ; then 'program' H so that $H(\alpha) = \beta$.

Key Challenge: Soundness

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

Fiat-Shamir Heuristic [FiatShamir'86]

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

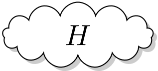


- Completeness and ZK (for honest V) are easy to preserve.
  For ZK, simulate α, β, γ; then 'program' H so that $H(\alpha) = \beta$.

Key Challenge: Soundness

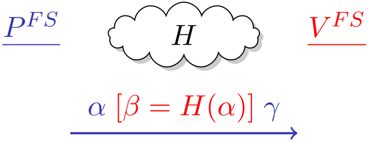
1. Are there α, γ with $\beta = H(\alpha)$ that fool V?
2. Can a cheating P^* find such values, given H? (Proof vs. argument.)

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

P^{FS}  V^{FS}

$$\xrightarrow{\alpha\ [\beta = H(\alpha)]\ \gamma}$$

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

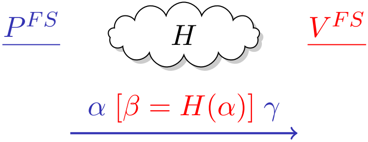


- Often, a correlation-intractable [CGH'98] hash family $\mathcal{H}$ suffices:

  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]



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

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

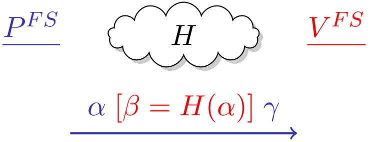
  Relation $R = \{(\alpha, \beta) : \exists\, \gamma$ that fools $V\}$.

Theorem [HL'18,CCH+'19]

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

$$R_C = \{(\alpha, C(\alpha))\},\ \ |C| \leq S = \text{poly}.$$

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



- Often, 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 $V\}$.

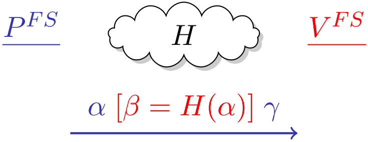
Theorem [HL'18,CCH+'19]

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

$$R_C = \{(\alpha, C(\alpha))\}, \;\; |C| \leq S = \text{poly}.$$

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

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



- Often, 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,CCH+'19]

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

$$R_C = \{(\alpha, C(\alpha))\},\ \ |C| \leq S = \text{poly}.$$

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

  Bad β is efficiently computable, using trapdoor for commitments in α.

Obtaining Correlation Intractability

[CCRR'18] CI for all sparse relations from 'exotic' assumptions, e.g., 'optimal' hardness of ad-hoc LWE variants.

Obtaining Correlation Intractability

[CCRR'18] CI for all sparse relations from 'exotic' assumptions, e.g., 'optimal' hardness of ad-hoc LWE variants.

[HL'18] CI for all sparse relations from (strong) obfuscation & more.

Obtaining Correlation Intractability

[CCRR'18] CI for all sparse relations from 'exotic' assumptions, e.g., 'optimal' hardness of ad-hoc LWE variants.

[HL'18] CI for all sparse relations from (strong) obfuscation & more.

[CCH+'19] CI for all bounded circuits from circularly secure FHE.

Obtaining Correlation Intractability

[CCRR'18] CI for all sparse relations from 'exotic' assumptions, e.g., 'optimal' hardness of ad-hoc LWE variants.

[HL'18] CI for all sparse relations from (strong) obfuscation & more.

[CCH+'19] CI for all bounded circuits from circularly secure FHE.

Seems tantalizingly close to LWE! But not known from LWE or worst-case lattice problems.

Obtaining Correlation Intractability

[CCRR'18] CI for all sparse relations from 'exotic' assumptions, e.g., 'optimal' hardness of ad-hoc LWE variants.

[HL'18] CI for all sparse relations from (strong) obfuscation & more.

[CCH+'19] CI for all bounded circuits from circularly secure FHE.
Seems tantalizingly close to LWE! But not known from LWE or worst-case lattice problems.

Our Main Construction

- A CI hash family for all bounded circuits C, from plain LWE

(for small poly approximation factors)

Obtaining Correlation Intractability

[CCRR'18] CI for all sparse relations from 'exotic' assumptions, e.g., 'optimal' hardness of ad-hoc LWE variants.

[HL'18] CI for all sparse relations from (strong) obfuscation & more.

[CCH+'19] CI for all bounded circuits from circularly secure FHE.
Seems tantalizingly close to LWE! But not known from LWE or worst-case lattice problems.

Our Main Construction

- A CI hash family for all bounded circuits C, from plain LWE
(for small poly approximation factors)
- As in [CCH+'19], our construction has two 'intractability modes':
  1. Computational: given $H \leftarrow \mathcal{H}$, hard to find α s.t. $H(\alpha) = C(\alpha)$.
  Yields statistically ZK argument in random-string model.

Obtaining Correlation Intractability

[CCRR'18] CI for all sparse relations from 'exotic' assumptions, e.g., 'optimal' hardness of ad-hoc LWE variants.

[HL'18] CI for all sparse relations from (strong) obfuscation & more.

[CCH+'19] CI for all bounded circuits from circularly secure FHE. Seems tantalizingly close to LWE! But not known from LWE or worst-case lattice problems.

Our Main Construction

- A CI hash family for all bounded circuits C, from plain LWE (for small poly approximation factors)
- As in [CCH+'19], our construction has two 'intractability modes':
  1. Computational: given $H \leftarrow \mathcal{H}$, hard to find α s.t. $H(\alpha) = C(\alpha)$. Yields statistically ZK argument in random-string model.
  2. Statistical: over $H \leftarrow \mathcal{H}_C \stackrel{c}{\approx} \mathcal{H}$, such α do not exist w/h.p. Yields computationally ZK proof in reference-string model.

Overview of Our Construction

1. A CI hash family for all NC^1 (log-depth) circuits from LWE/SIS

(for small poly approx factors)

Overview of Our Construction

1. A CI hash family for all NC^1 (log-depth) circuits from LWE/SIS

   (for small poly approx factors)

2. A CI 'bootstrapping' theorem, from (leveled) FHE decryption circuits in NC^1, to arbitrary bounded circuits, à la [Gentry'09,GGH+'13].

   (Such FHE can be based on LWE w/ small poly factors [BV'14].)

Overview of Our Construction

1. A CI hash family for all NC^1 (log-depth) circuits from LWE/SIS

   (for small poly approx factors)

2. A CI 'bootstrapping' theorem, from (leveled) FHE decryption circuits in NC^1, to arbitrary bounded circuits, à la [Gentry'09,GGH+'13].

   (Such FHE can be based on LWE w/ small poly factors [BV'14].)

- For NIZK we do not actually need bootstrapping, because the 'bad challenge' functions can be implemented in NC^1 [CCH+'19,Lombardi].

SIS and LWE [Ajtai'96,...,Regev'05,...]

- Fix integer modulus $q = \text{poly}(n)$ and dimensions $n, m \geq 2n\lceil \log q \rceil$.

SIS and LWE [Ajtai'96,...,Regev'05,...]

- Fix integer modulus $q = \text{poly}(n)$ and dimensions $n, m \geq 2n\lceil \log q \rceil$.

  SIS: given uniform $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, find 'short' nonzero $\mathbf{z} \in \mathbb{Z}^m$ s.t.

$$\left(\quad \mathbf{A} \quad \right) \begin{pmatrix} \\ \mathbf{z} \\ \\ \end{pmatrix} = \begin{pmatrix} \mathbf{0} \end{pmatrix} \in \mathbb{Z}_q^n.$$

SIS and LWE [Ajtai'96,...,Regev'05,...]

- Fix integer modulus $q = \text{poly}(n)$ and dimensions $n, m \geq 2n\lceil \log q \rceil$.

  SIS: given uniform $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, find 'short' nonzero $\mathbf{z} \in \mathbb{Z}^m$ s.t.

$$\begin{pmatrix} & \mathbf{A} & \end{pmatrix} \begin{pmatrix} \\ \mathbf{z} \\ \\ \end{pmatrix} = \begin{pmatrix} \mathbf{0} \end{pmatrix} \in \mathbb{Z}_q^n.$$

  LWE: distinguish uniform $\mathbf{A}$ from

$$\begin{pmatrix} \mathbf{A}' \\ \mathbf{s}^t \mathbf{A}' + \mathbf{e}^t \end{pmatrix}$$

  for uniform $\mathbf{A}' \in \mathbb{Z}_q^{(n-1) \times m}$ and 'short' (Gaussian) $\mathbf{s}, \mathbf{e} \in \mathbb{Z}^m$.

SIS and LWE [Ajtai'96,...,Regev'05,...]

- Fix integer modulus $q = \text{poly}(n)$ and dimensions $n, m \geq 2n\lceil \log q \rceil$.

  SIS: given uniform $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, find 'short' nonzero $\mathbf{z} \in \mathbb{Z}^m$ s.t.

$$\begin{pmatrix} & \mathbf{A} & \end{pmatrix} \begin{pmatrix} \\ \mathbf{z} \\ \\ \end{pmatrix} = \begin{pmatrix} \mathbf{0} \end{pmatrix} \in \mathbb{Z}_q^n.$$

  LWE: distinguish uniform $\mathbf{A}$ from

$$\begin{pmatrix} \mathbf{A}' \\ \mathbf{s}^t \mathbf{A}' + \mathbf{e}^t \end{pmatrix}$$

  for uniform $\mathbf{A}' \in \mathbb{Z}_q^{(n-1) \times m}$ and 'short' (Gaussian) $\mathbf{s}, \mathbf{e} \in \mathbb{Z}^m$.

Theorems

- Worst-case lattice problems reduce to average-case SIS/LWE.

SIS and LWE [Ajtai'96,...,Regev'05,...]

- Fix integer modulus $q = \text{poly}(n)$ and dimensions $n, m \geq 2n\lceil \log q \rceil$.

  SIS: given uniform $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, find 'short' nonzero $\mathbf{z} \in \mathbb{Z}^m$ s.t.

$$\begin{pmatrix} & \mathbf{A} & \end{pmatrix} \begin{pmatrix} \\ \mathbf{z} \\ \\ \end{pmatrix} = \begin{pmatrix} \mathbf{0} \end{pmatrix} \in \mathbb{Z}_q^n.$$

  LWE: distinguish uniform $\mathbf{A}$ from

$$\begin{pmatrix} \mathbf{A}' \\ \mathbf{s}^t \mathbf{A}' + \mathbf{e}^t \end{pmatrix}$$

  for uniform $\mathbf{A}' \in \mathbb{Z}_q^{(n-1) \times m}$ and 'short' (Gaussian) $\mathbf{s}, \mathbf{e} \in \mathbb{Z}^m$.

- Linear $G \colon \{0,1\}^m \to \mathbb{Z}_q^n$ and nonlinear $G^- \colon \mathbb{Z}_q^n \to \{0,1\}^m$ s.t.

$$G(G^-(\mathbf{u})) = \mathbf{u} \text{ for all } \mathbf{u} \in \mathbb{Z}_q^n.$$

Our Construction

- Goal: CI for size-S circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$, $m \geq 2n\lceil \log q \rceil$

Our Construction

- Goal: CI for size-S circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$, $m \geq 2n\lceil \log q \rceil$
- Uses LWE/SIS-based FH encryption/commitment [GSW'13,GVW'15]

Our Construction

- Goal: CI for size-S circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$, $m \geq 2n\lceil \log q \rceil$
- Uses LWE/SIS-based FH encryption/commitment [GSW'13,GVW'15]

Hash Key: commitment $\widehat{D}$ to 'dummy' circuit $D\colon \{0,1\}^\ell \to \{0,1\}^m$.

Our Construction

- Goal: CI for size-S circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$, $m \geq 2n\lceil \log q \rceil$
- Uses LWE/SIS-based FH encryption/commitment [GSW'13,GVW'15]

Hash Key: commitment $\widehat{D}$ to 'dummy' circuit $D\colon \{0,1\}^\ell \to \{0,1\}^m$.

Evaluation: on input $\alpha \in \{0,1\}^\ell$,

1. Homomorphically compute commitment $\widehat{D(\alpha)}$.

Our Construction

- Goal: CI for size-S circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$, $m \geq 2n\lceil \log q \rceil$
- Uses LWE/SIS-based FH encryption/commitment [GSW'13,GVW'15]

Hash Key: commitment $\widehat{D}$ to 'dummy' circuit $D\colon \{0,1\}^\ell \to \{0,1\}^m$.

Evaluation: on input $\alpha \in \{0,1\}^\ell$,

1. Homomorphically compute commitment $\widehat{D(\alpha)}$.
2. Homomorphically evaluate linear $G\colon \{0,1\}^m \to \mathbb{Z}_q^n$ to get 'inert commitment' $c_\alpha = \overline{G(D(\alpha))} \in \mathbb{Z}_q^n$.

Our Construction

- Goal: CI for size-S circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$, $m \geq 2n\lceil \log q \rceil$
- Uses LWE/SIS-based FH encryption/commitment [GSW'13,GVW'15]

Hash Key: commitment $\widehat{D}$ to 'dummy' circuit $D\colon \{0,1\}^\ell \to \{0,1\}^m$.

Evaluation: on input $\alpha \in \{0,1\}^\ell$,

1. Homomorphically compute commitment $\widehat{D(\alpha)}$.
2. Homomorphically evaluate linear $G\colon \{0,1\}^m \to \mathbb{Z}_q^n$ to get 'inert commitment' $c_\alpha = \overline{G(D(\alpha))} \in \mathbb{Z}_q^n$.
3. Output $G^{-}(c_\alpha) \in \{0,1\}^m$.

Our Construction

- Goal: CI for size-S circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$, $m \geq 2n\lceil \log q \rceil$
- Uses LWE/SIS-based FH encryption/commitment [GSW'13,GVW'15]

Hash Key: commitment $\widehat{D}$ to 'dummy' circuit $D\colon \{0,1\}^\ell \to \{0,1\}^m$.

([CCH+'19] uses FHE ciphertexts, also includes 'circular' $\widehat{sk}$.)

Evaluation: on input $\alpha \in \{0,1\}^\ell$,

1. Homomorphically compute commitment $\widehat{D(\alpha)}$.
2. Homomorphically evaluate linear $G\colon \{0,1\}^m \to \mathbb{Z}_q^n$ to get 'inert commitment' $c_\alpha = \overline{G(D(\alpha))} \in \mathbb{Z}_q^n$.
3. Output $G^-(c_\alpha) \in \{0,1\}^m$.

Our Construction

- Goal: CI for size-S circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$, $m \geq 2n\lceil \log q \rceil$
- Uses LWE/SIS-based FH encryption/commitment [GSW'13,GVW'15]

Hash Key: commitment $\widehat{D}$ to 'dummy' circuit $D\colon \{0,1\}^\ell \to \{0,1\}^m$.

([CCH+'19] uses FHE ciphertexts, also includes 'circular' $\widehat{sk}$.)

Evaluation: on input $\alpha \in \{0,1\}^\ell$,

1. Homomorphically compute commitment $\widehat{D(\alpha)}$.

   ([CCH+'19] does the same, but with ciphertexts.)

2. Homomorphically evaluate linear $G\colon \{0,1\}^m \to \mathbb{Z}_q^n$ to get 'inert commitment' $c_\alpha = \overline{G(D(\alpha))} \in \mathbb{Z}_q^n$.

3. Output $G^{-}(c_\alpha) \in \{0,1\}^m$.

Our Construction

- Goal: CI for size-S circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$, $m \geq 2n\lceil \log q \rceil$
- Uses LWE/SIS-based FH encryption/commitment [GSW'13,GVW'15]

Hash Key: commitment $\widehat{D}$ to 'dummy' circuit $D\colon \{0,1\}^\ell \to \{0,1\}^m$.

([CCH+'19] uses FHE ciphertexts, also includes 'circular' $\widehat{sk}$.)

Evaluation: on input $\alpha \in \{0,1\}^\ell$,

1. Homomorphically compute commitment $\widehat{D(\alpha)}$.

   ([CCH+'19] does the same, but with ciphertexts.)

2. Homomorphically evaluate linear $G\colon \{0,1\}^m \to \mathbb{Z}_q^n$ to get 'inert commitment' $c_\alpha = \overline{G(D(\alpha))} \in \mathbb{Z}_q^n$.

   ([CCH+'19] evaluates Dec_{sk} to get an FHE ciphertext.)

3. Output $G^{-}(c_\alpha) \in \{0,1\}^m$.

Our Construction

- Goal: CI for size-S circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$, $m \geq 2n\lceil \log q \rceil$
- Uses LWE/SIS-based FH encryption/commitment [GSW'13,GVW'15]

Hash Key: commitment $\widehat{D}$ to 'dummy' circuit $D\colon \{0,1\}^\ell \to \{0,1\}^m$.

([CCH+'19] uses FHE ciphertexts, also includes 'circular' $\widehat{sk}$.)

Evaluation: on input $\alpha \in \{0,1\}^\ell$,

1. Homomorphically compute commitment $\widehat{D(\alpha)}$.

   ([CCH+'19] does the same, but with ciphertexts.)

2. Homomorphically evaluate linear $G\colon \{0,1\}^m \to \mathbb{Z}_q^n$ to get 'inert commitment' $c_\alpha = \overline{G(D(\alpha))} \in \mathbb{Z}_q^n$.

   ([CCH+'19] evaluates Dec_{sk} to get an FHE ciphertext.)

3. Output $G^{-}(c_\alpha) \in \{0,1\}^m$.

Key Point: $c_\alpha \in \mathbb{Z}_q^n$ hides a $\mathbb{Z}_q^n$-value: lets us compare the two directly, not just reason about hidden values (as in [CCH+'19]).

Security Proof from SIS

Hash Key: commitment $\widehat{D}$.

Evaluation: $H(\alpha) := G^{-}(\overline{G(D(\alpha))})$

- Let $C \colon \{0,1\}^{\ell} \to \{0,1\}^{m}$ have size S.

Security Proof from SIS

Hash Key: commitment $\widehat{D}$.

Evaluation: $H(\alpha) := G^{-}(\overline{G(D(\alpha))}) = C(\alpha)$.

- Let $C\colon \{0,1\}^{\ell} \to \{0,1\}^{m}$ have size S.
- Suppose that $\mathcal{A}$, given hash key $\widehat{D}$, finds α s.t. $H(\alpha) = C(\alpha)$.

Security Proof from SIS

Hash Key: commitment $\widehat{C}$.

Evaluation: $H(\alpha) := G^{-}(\overline{G(C(\alpha))}) = C(\alpha)$.

- Let $C\colon \{0,1\}^\ell \to \{0,1\}^m$ have size S.
- Suppose that $\mathcal{A}$, given hash key $\widehat{D}$, finds α s.t. $H(\alpha) = C(\alpha)$.
- By commitment security, same holds for hash key $\widehat{C} = \mathsf{Com}(C; \mathbf{R}_C)$.

Security Proof from SIS

Hash Key: commitment $\widehat{C}$.

Evaluation: $H(\alpha) := G^{-}(\overline{G(C(\alpha))}) = C(\alpha)$.

- Let $C\colon \{0,1\}^\ell \to \{0,1\}^m$ have size S.
- Suppose that $\mathcal{A}$, given hash key $\widehat{D}$, finds α s.t. $H(\alpha) = C(\alpha)$.
- By commitment security, same holds for hash key $\widehat{C} = \mathsf{Com}(C; \mathbf{R}_C)$.
  Apply G to both sides:

$$c_\alpha = \overline{G(C(\alpha))} = G(C(\alpha)) \in \mathbb{Z}_q^n.$$

Security Proof from SIS

Hash Key: commitment $\widehat{C}$.

Evaluation: $H(\alpha) := G^{-}(\overline{G(C(\alpha))}) = C(\alpha)$.

- Let $C \colon \{0,1\}^{\ell} \to \{0,1\}^{m}$ have size S.
- Suppose that $\mathcal{A}$, given hash key $\widehat{D}$, finds α s.t. $H(\alpha) = C(\alpha)$.
- By commitment security, same holds for hash key $\widehat{C} = \mathsf{Com}(C; \mathbf{R}_C)$.
  Apply G to both sides:

$$c_{\alpha} = \overline{G(C(\alpha))} = G(C(\alpha)) \in \mathbb{Z}_q^n.$$

That is, the inert commitment c_{α} itself equals its 'contents.'

Security Proof from SIS

Hash Key: commitment $\widehat{C}$.

Evaluation: $H(\alpha) := G^{-}(\overline{G(C(\alpha))}) = C(\alpha)$.

- Let $C\colon \{0,1\}^\ell \to \{0,1\}^m$ have size S.
- Suppose that $\mathcal{A}$, given hash key $\widehat{D}$, finds α s.t. $H(\alpha) = C(\alpha)$.
- By commitment security, same holds for hash key $\widehat{C} = \mathsf{Com}(C; \mathbf{R}_C)$.

  Apply G to both sides:

$$c_\alpha = \overline{G(C(\alpha))} = G(C(\alpha)) \in \mathbb{Z}_q^n.$$

  That is, the inert commitment c_α itself equals its 'contents.'

Theorem

- From coins $\mathbf{R}_C$ for $\widehat{C}$ we can compute coins $\mathbf{r}_\alpha$ for c_α, solving SIS.

Security Proof from SIS

Hash Key: commitment $\widehat{C} = \mathsf{Com}(C; \mathbf{R}_C)$.

Evaluation: computes $c_\alpha = \overline{G(C(\alpha))} = G(C(\alpha))$.

Theorem

- From coins $\mathbf{R}_C$ for $\widehat{C}$ we can compute coins $\mathbf{r}_\alpha$ for c_α, solving SIS.

Security Proof from SIS

Hash Key: commitment $\widehat{C} = \mathsf{Com}(C; \mathbf{R}_C)$.

Evaluation: computes $c_\alpha = \overline{G(C(\alpha))} = G(C(\alpha))$.

Theorem

- From coins $\mathbf{R}_C$ for $\widehat{C}$ we can compute coins $\mathbf{r}_\alpha$ for c_α, solving SIS.

- Commitments are w.r.t. an SIS matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, w/ 'short' coins:

$$\widehat{C} = \mathbf{A} \cdot \mathbf{R}_C + \mathsf{encode}(C) \pmod{q}.$$

Security Proof from SIS

Hash Key: commitment $\widehat{C} = \mathsf{Com}(C; \mathbf{R}_C)$.

Evaluation: computes $c_\alpha = \overline{G(C(\alpha))} = G(C(\alpha))$.

Theorem

- From coins $\mathbf{R}_C$ for $\widehat{C}$ we can compute coins $\mathbf{r}_\alpha$ for c_α, solving SIS.

- Commitments are w.r.t. an SIS matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, w/ 'short' coins:

$$\widehat{C} = \mathbf{A} \cdot \mathbf{R}_C + \mathsf{encode}(C) \pmod q.$$

- From $\mathbf{R}_C$ we can compute coins $\mathbf{R}$ for $\widehat{C(\alpha)}$ [GVW'15]:

$$\widehat{C(\alpha)} = \mathbf{A} \cdot \mathbf{R} + \mathsf{encode}(C(\alpha)) \pmod q.$$

Security Proof from SIS

Hash Key: commitment $\widehat{C} = \mathsf{Com}(C; \mathbf{R}_C)$.

Evaluation: computes $c_\alpha = \overline{G(C(\alpha))} = G(C(\alpha))$.

Theorem

- From coins $\mathbf{R}_C$ for $\widehat{C}$ we can compute coins $\mathbf{r}_\alpha$ for c_α, solving SIS.

- Commitments are w.r.t. an SIS matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, w/ 'short' coins:

$$\widehat{C} = \mathbf{A} \cdot \mathbf{R}_C + \mathsf{encode}(C) \pmod q.$$

- From $\mathbf{R}_C$ we can compute coins $\mathbf{R}$ for $\widehat{C(\alpha)}$ [GVW'15]:

$$\widehat{C(\alpha)} = \mathbf{A} \cdot \mathbf{R} + \mathsf{encode}(C(\alpha)) \pmod q.$$

- From $\mathbf{R}$ we can compute coins $\mathbf{r}_\alpha$ for inert commitment c_α [this work]:

$$\overline{G(C(\alpha))} = \mathbf{A} \cdot \mathbf{r}_\alpha + G(C(\alpha)) \qquad \in \mathbb{Z}_q^n.$$

Security Proof from SIS

Hash Key: commitment $\widehat{C} = \mathsf{Com}(C; \mathbf{R}_C)$.

Evaluation: computes $c_\alpha = \overline{G(C(\alpha))} = G(C(\alpha))$.

Theorem

- From coins $\mathbf{R}_C$ for $\widehat{C}$ we can compute coins $\mathbf{r}_\alpha$ for c_α, solving SIS.

- Commitments are w.r.t. an SIS matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, w/ 'short' coins:

$$\widehat{C} = \mathbf{A} \cdot \mathbf{R}_C + \mathsf{encode}(C) \pmod q.$$

- From $\mathbf{R}_C$ we can compute coins $\mathbf{R}$ for $\widehat{C(\alpha)}$ [GVW'15]:

$$\widehat{C(\alpha)} = \mathbf{A} \cdot \mathbf{R} + \mathsf{encode}(C(\alpha)) \pmod q.$$

- From $\mathbf{R}$ we can compute coins $\mathbf{r}_\alpha$ for inert commitment c_α [this work]:

$$\overline{G(C(\alpha))} = \mathbf{A} \cdot \mathbf{r}_\alpha + G(C(\alpha)) = G(C(\alpha)) \in \mathbb{Z}_q^n.$$

Security Proof from SIS

Hash Key: commitment $\widehat{C} = \text{Com}(C; \mathbf{R}_C)$.
Evaluation: computes $c_\alpha = \overline{G(C(\alpha))} = G(C(\alpha))$.

Theorem

- From coins $\mathbf{R}_C$ for $\widehat{C}$ we can compute coins $\mathbf{r}_\alpha$ for c_α, solving SIS.

- Commitments are w.r.t. an SIS matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, w/ 'short' coins:

$$\widehat{C} = \mathbf{A} \cdot \mathbf{R}_C + \text{encode}(C) \pmod q.$$

- From $\mathbf{R}_C$ we can compute coins $\mathbf{R}$ for $\widehat{C(\alpha)}$ [GVW'15]:

$$\widehat{C(\alpha)} = \mathbf{A} \cdot \mathbf{R} + \text{encode}(C(\alpha)) \pmod q.$$

- From $\mathbf{R}$ we can compute coins $\mathbf{r}_\alpha$ for inert commitment c_α [this work]:

$$\overline{G(C(\alpha))} = \mathbf{A} \cdot \mathbf{r}_\alpha + G(C(\alpha)) = G(C(\alpha)) \in \mathbb{Z}_q^n.$$

- Thus $\mathbf{A} \cdot \mathbf{r}_\alpha = \mathbf{0}$, solving SIS!

Security Proof from SIS

Hash Key: commitment $\widehat{C} = \mathsf{Com}(C; \mathbf{R}_C)$.

Evaluation: computes $c_\alpha = \overline{G(C(\alpha))} = G(C(\alpha))$.

Theorem

- From coins $\mathbf{R}_C$ for $\widehat{C}$ we can compute coins $\mathbf{r}_\alpha$ for c_α, solving SIS.

- Commitments are w.r.t. an SIS matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, w/ 'short' coins:

$$\widehat{C} = \mathbf{A} \cdot \mathbf{R}_C + \mathsf{encode}(C) \pmod q.$$

- From $\mathbf{R}_C$ we can compute coins $\mathbf{R}$ for $\widehat{C(\alpha)}$ [GVW'15]:

$$\widehat{C(\alpha)} = \mathbf{A} \cdot \mathbf{R} + \mathsf{encode}(C(\alpha)) \pmod q.$$

- From $\mathbf{R}$ we can compute coins $\mathbf{r}_\alpha$ for inert commitment c_α [this work]:

$$\overline{G(C(\alpha))} = \mathbf{A} \cdot \mathbf{r}_\alpha + G(C(\alpha)) = G(C(\alpha)) \in \mathbb{Z}_q^n.$$

- Thus $\mathbf{A} \cdot \mathbf{r}_\alpha = \mathbf{0}$, solving SIS! (Also need $\mathbf{r}_\alpha \neq \mathbf{0}$, an easy tweak.)

Linear Homomorphism to an Inert Commitment

Given: commitment $\widehat{x}$ [and 'short' coins $\mathbf{R}$] for $x \in \{0,1\}^m$:

$$\widehat{x} = \mathbf{A} \cdot \mathbf{R} + \begin{pmatrix} x_1\mathbf{G} & \cdots & x_m\mathbf{G} \end{pmatrix} \pmod{q}.$$

Linear Homomorphism to an Inert Commitment

Given: commitment $\widehat{x}$ [and 'short' coins $\mathbf{R}$] for $x \in \{0,1\}^m$:

$$\widehat{x} = \mathbf{A} \cdot \mathbf{R} + \begin{pmatrix} x_1 \mathbf{G} & \cdots & x_m \mathbf{G} \end{pmatrix} \pmod{q}.$$

Goal: compute inert $\overline{L(x)}$ [and coins $\mathbf{r}$] for linear $L \colon \{0,1\}^m \to \mathbb{Z}_q^n$.

Linear Homomorphism to an Inert Commitment

Given: commitment $\widehat{x}$ [and 'short' coins $\mathbf{R}$] for $x \in \{0,1\}^m$:

$$\widehat{x} = \mathbf{A} \cdot \mathbf{R} + \begin{pmatrix} x_1\mathbf{G} & \cdots & x_m\mathbf{G} \end{pmatrix} \pmod q.$$

Goal: compute inert $\overline{L(x)}$ [and coins $\mathbf{r}$] for linear $L \colon \{0,1\}^m \to \mathbb{Z}_q^n$.

- Write $L(x) = \sum_i x_i \cdot \mathbf{c}_i$ for some $\mathbf{c}_i \in \mathbb{Z}_q^n$. Define short

$$\mathbf{v}_L := \begin{pmatrix} \mathbf{G}^{-1}(\mathbf{c}_1) \\ \vdots \\ \mathbf{G}^{-1}(\mathbf{c}_m) \end{pmatrix}.$$

Linear Homomorphism to an Inert Commitment

Given: commitment $\widehat{x}$ [and 'short' coins $\mathbf{R}$] for $x \in \{0,1\}^m$:

$$\widehat{x} = \mathbf{A} \cdot \mathbf{R} + \begin{pmatrix} x_1\mathbf{G} & \cdots & x_m\mathbf{G} \end{pmatrix} \pmod{q}.$$

Goal: compute inert $\overline{L(x)}$ [and coins $\mathbf{r}$] for linear $L \colon \{0,1\}^m \to \mathbb{Z}_q^n$.

- Write $L(x) = \sum_i x_i \cdot \mathbf{c}_i$ for some $\mathbf{c}_i \in \mathbb{Z}_q^n$. Define short

$$\mathbf{v}_L := \begin{pmatrix} \mathbf{G}^{-1}(\mathbf{c}_1) \\ \vdots \\ \mathbf{G}^{-1}(\mathbf{c}_m) \end{pmatrix}.$$

- Then

$$\begin{aligned} \widehat{x} \cdot \mathbf{v}_L &= \mathbf{A} \cdot \underbrace{\mathbf{R} \cdot \mathbf{v}_L}_{\mathbf{r}} + \sum_i x_i \cdot \mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{c}_i) \\ &= \mathbf{A} \cdot \mathbf{r} + L(x) = \overline{L(x)}. \end{aligned}$$

LWE-Based Construction

- SIS construction is computationally CI with uniform key $(\mathbf{A}, \widehat{D})$.

LWE-Based Construction

- SIS construction is computationally CI with uniform key $(\mathbf{A}, \widehat{D})$. Yields computationally sound, statistically ZK protocol.

LWE-Based Construction

- SIS construction is computationally CI with uniform key $(\mathbf{A}, \widehat{D})$. Yields computationally sound, statistically ZK protocol.
- An LWE-based statistically CI construction with non-uniform key:

LWE-Based Construction

- SIS construction is computationally CI with uniform key $(\mathbf{A}, \widehat{D})$. Yields computationally sound, statistically ZK protocol.
- An LWE-based statistically CI construction with non-uniform key:

Hash Key: commitment $\widehat{C}$ w.r.t. LWE matrix $\mathbf{A} = \begin{pmatrix} \mathbf{A}' \\ \mathbf{s}^t\mathbf{A}' + \mathbf{e}^t \end{pmatrix} \in \mathbb{Z}_q^{n \times m}$

LWE-Based Construction

- SIS construction is computationally CI with uniform key $(\mathbf{A}, \widehat{D})$.
  Yields computationally sound, statistically ZK protocol.

- An LWE-based statistically CI construction with non-uniform key:

Hash Key: commitment $\widehat{C}$ w.r.t. LWE matrix $\mathbf{A} = \begin{pmatrix} \mathbf{A}' \\ \mathbf{s}^t\mathbf{A}' + \mathbf{e}^t \end{pmatrix} \in \mathbb{Z}_q^{n \times m}$

Evaluation: computes $c_\alpha = \overline{G(C(\alpha))} - \begin{pmatrix} \mathbf{0} \\ q/2 \end{pmatrix} \in \mathbb{Z}_q^n$

LWE-Based Construction

- SIS construction is computationally CI with uniform key $(\mathbf{A}, \widehat{D})$.
  Yields computationally sound, statistically ZK protocol.

- An LWE-based statistically CI construction with non-uniform key:

Hash Key: commitment $\widehat{C}$ w.r.t. LWE matrix $\mathbf{A} = \left(\begin{smallmatrix} \mathbf{A}' \\ \mathbf{s}^t \mathbf{A}' + \mathbf{e}^t \end{smallmatrix} \right) \in \mathbb{Z}_q^{n \times m}$

Evaluation: computes $c_\alpha = \overline{G(C(\alpha))} - \left(\begin{smallmatrix} \mathbf{0} \\ q/2 \end{smallmatrix} \right) \in \mathbb{Z}_q^n$

- Now $H(\alpha) = C(\alpha)$ yields $\mathbf{A}\mathbf{r}_\alpha = \left(\begin{smallmatrix} \mathbf{0} \\ q/2 \end{smallmatrix} \right)$. So $\mathbf{A}'\mathbf{r}_\alpha = \mathbf{0}$ and

$$\tfrac{q}{2} = (\mathbf{s}^t \mathbf{A}' + \mathbf{e}^t) \cdot \mathbf{r}_\alpha = \mathbf{e}^t \cdot \mathbf{r}_\alpha \pmod q,$$

  but $\mathbf{e}, \mathbf{r}_\alpha$ are too small for this: contradiction!

Open Problems

1. CI beyond NC^1 from SIS (not LWE) w/poly factors?
   Currently we need bootstrapping, which brings in LWE.

Open Problems

1. CI beyond NC^1 from SIS (not LWE) w/poly factors?
   Currently we need bootstrapping, which brings in LWE.
2. Noninteractive Witness Indistinguishable (NIWI) proofs, plain model?

Open Problems

1. CI beyond NC^1 from SIS (not LWE) w/poly factors?
   Currently we need bootstrapping, which brings in LWE.
2. Noninteractive Witness Indistinguishable (NIWI) proofs, plain model?
   [GOS'06] gets NIWI from statistical soundness in random-string model.
   But we just have computational soundness there.

Open Problems

1. CI beyond NC^1 from SIS (not LWE) w/poly factors?
   Currently we need bootstrapping, which brings in LWE.
2. Noninteractive Witness Indistinguishable (NIWI) proofs, plain model?
   [GOS'06] gets NIWI from statistical soundness in random-string model.
   But we just have computational soundness there.
3. Compactness? Our hash key grows with the circuit size for CI, unlike those based on 'exotic' assumptions (e.g., obfuscation).

Open Problems

1. CI beyond NC^1 from SIS (not LWE) w/poly factors?
   Currently we need bootstrapping, which brings in LWE.
2. Noninteractive Witness Indistinguishable (NIWI) proofs, plain model?
   [GOS'06] gets NIWI from statistical soundness in random-string model.
   But we just have computational soundness there.
3. Compactness? Our hash key grows with the circuit size for CI, unlike those based on 'exotic' assumptions (e.g., obfuscation).
4. Succinct ZK arguments from LWE? Via Fiat-Shamir?

Open Problems

1. CI beyond NC^1 from SIS (not LWE) w/poly factors?
   Currently we need bootstrapping, which brings in LWE.
2. Noninteractive Witness Indistinguishable (NIWI) proofs, plain model?
   [GOS'06] gets NIWI from statistical soundness in random-string model.
   But we just have computational soundness there.
3. Compactness? Our hash key grows with the circuit size for CI, unlike those based on 'exotic' assumptions (e.g., obfuscation).
4. Succinct ZK arguments from LWE? Via Fiat-Shamir?

Thanks!