Beyond "MPC in the Head": Black-Box Constructions of Short Zero-Knowledge Proofs

Carmit Hazay — Bar-Ilan University

Muthuramakrishnan Venkitasubramaniam — Georgetown University

Mor Weiss — Bar-Ilan University

What This Talk Is About

- **Part (1): Beyond MPC in the Head**
  - New paradigm for ZKP design, extends [IKOS07]
  - Based on computations “in the head” from weak primitives
  - **Versatile:** applicable to many primitives (FHE, FE, FSS, HSS, RE, LFE) and protocols (IP, IOP), extends to commit-and-prove functionalities

What This Talk Is About

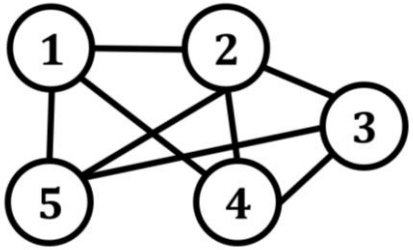
- **Part (1): Beyond MPC in the Head**
  - New paradigm for ZKP design, extends [IKOS07]
  - Based on computations "in the head" from weak primitives
  - **Versatile:** applicable to many primitives (FHE, FE, FSS, HSS, RE, LFE) and protocols (IP, IOP), extends to commit-and-prove functionalities
- **Part (2): Constructions of short (almost witness length) ZKPs**
  - New constructions for NC^1
  - Black-box alternatives to existing (non-BB) ZKPs for NC^1, NP and more
  - Casting some existing BB ZKPs as special cases of the paradigm

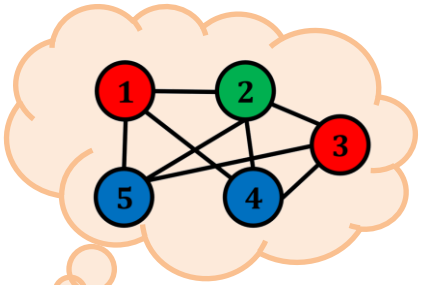
What This Talk Is About

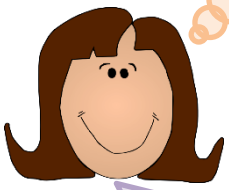
- **Part (1): Beyond MPC in the Head**
  - New paradigm for ZKP design, extends [IKOS07]
  - Based on computations "in the head" from weak primitives
  - **Versatile:** applicable to many primitives (FHE, FE, FSS, HSS, RE, LFE) and protocols (IP, IOP), extends to commit-and-prove functionalities
- **Part (2): Constructions of short (almost witness length) ZKPs**
  - New constructions for NC^1
  - Black-box alternatives to existing (non-BB) ZKPs for NC^1, NP and more
  - Casting some existing BB ZKPs as special cases of the paradigm

Scary table coming up!

Zero-Knowledge Proofs







Graph is 3-colorable!

I’m convinced!

Beyond MPC in the Head

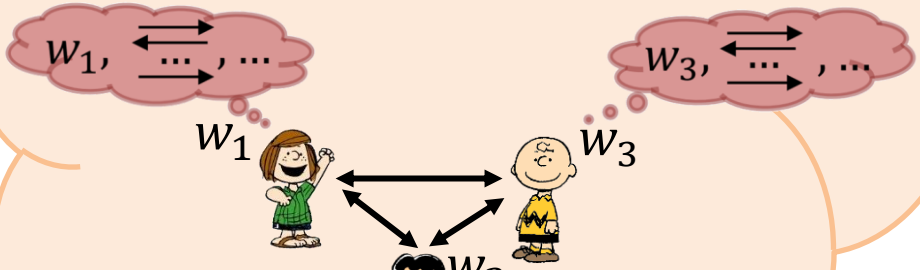
MPC in the Head 101 [IKOS07]

Claim: $C(w) = 1$ for some w

Witness Secret Share

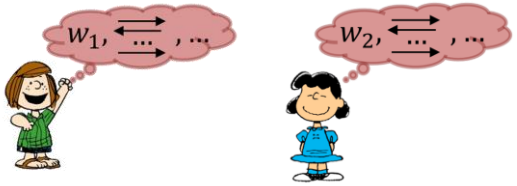
$$w = w_1 \oplus w_2 \oplus w_3$$

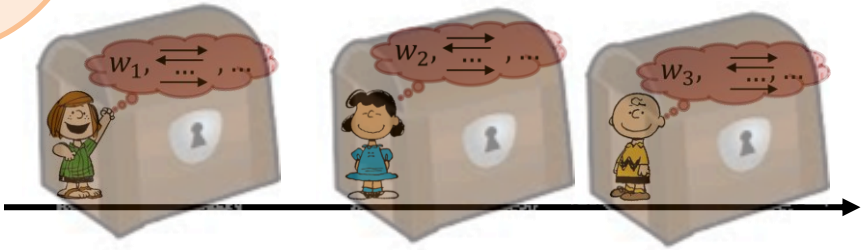
$$f(w_1, w_2, w_3) = C(w_1 \oplus w_2 \oplus w_3)$$

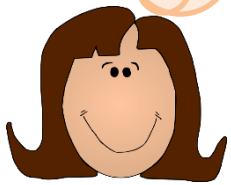


Check correctness of execution

$$i \neq j \leftarrow \{1,2,3\}$$







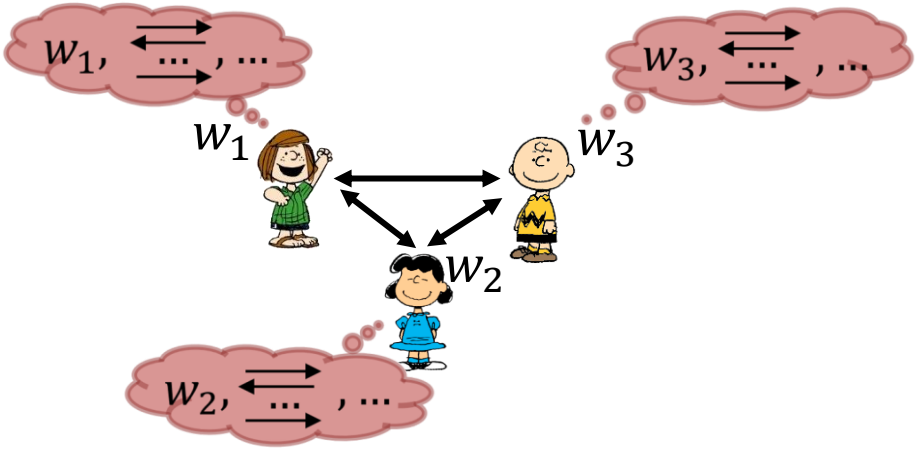
MPC in the Head 101 [IKOS07]

Claim: $C(w) = 1$ for some w

Witness Secret Share

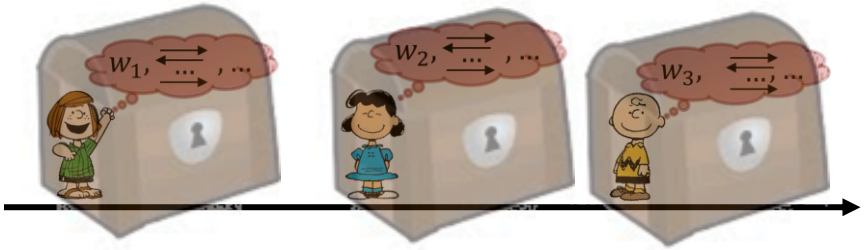
$w = w_1 \oplus w_2 \oplus w_3$

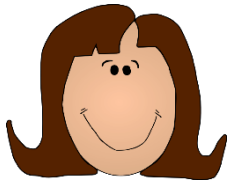
$f(w_1, w_2, w_3) = C(w_1 \oplus w_2 \oplus w_3)$



Check correctness of execution

$i \neq j \leftarrow \{1,2,3\}$





MPC in the Head 101 [IKOS07]

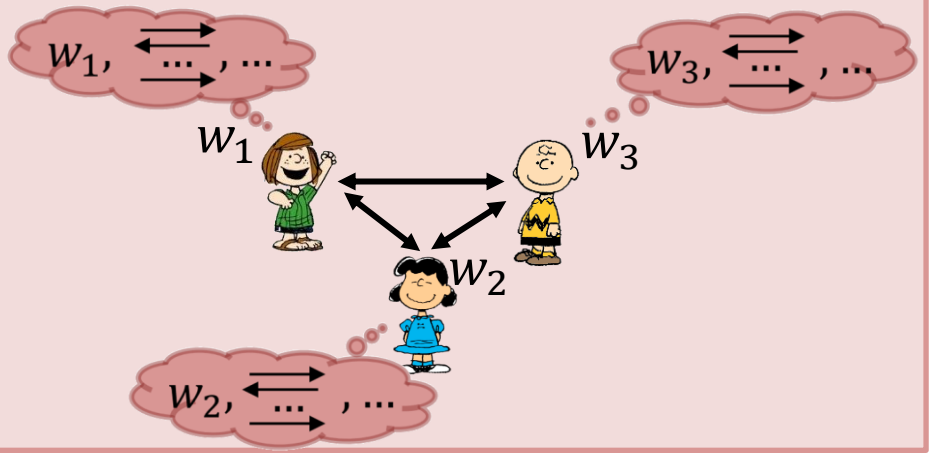
Claim: $C(w) = 1$ for some w

Witness Secret Share

$w = w_1 \oplus w_2 \oplus w_3$

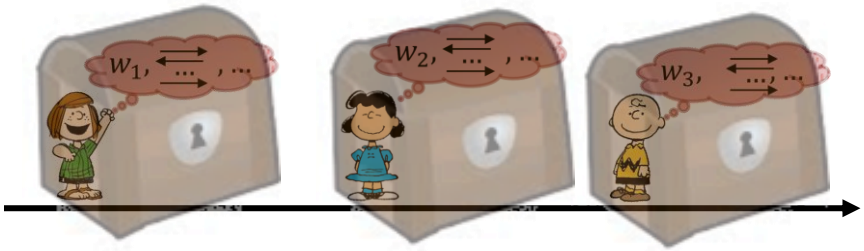
Eval

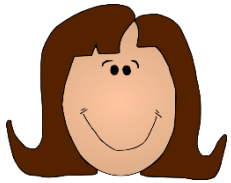
$f(w_1, w_2, w_3) = C(w_1 \oplus w_2 \oplus w_3)$



Check correctness of execution

$i \neq j \leftarrow \{1,2,3\}$





MPC in the Head 101 [IKOS07]

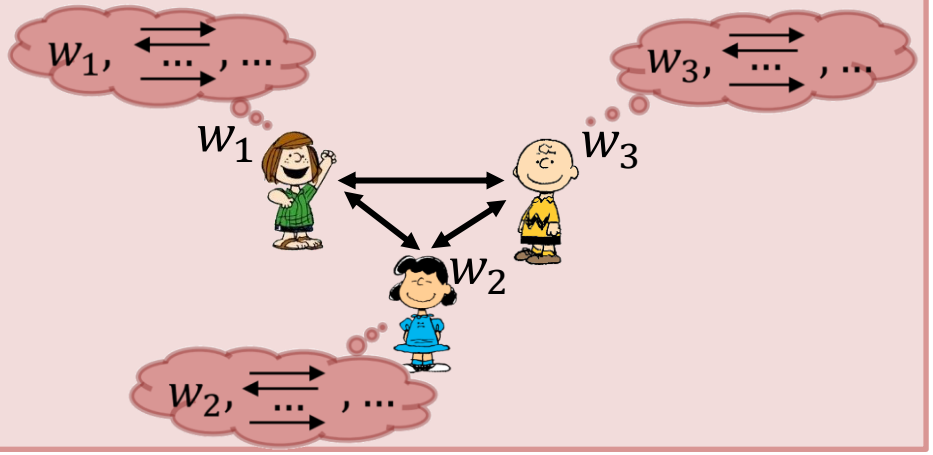
Claim: $C(w) = 1$ for some w

Witness Secret Share

$w = w_1 \oplus w_2 \oplus w_3$

Eval

$f(w_1, w_2, w_3) = C(w_1 \oplus w_2 \oplus w_3)$

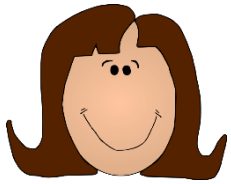


Check correctness of execution

$i \neq j \leftarrow \{1,2,3\}$

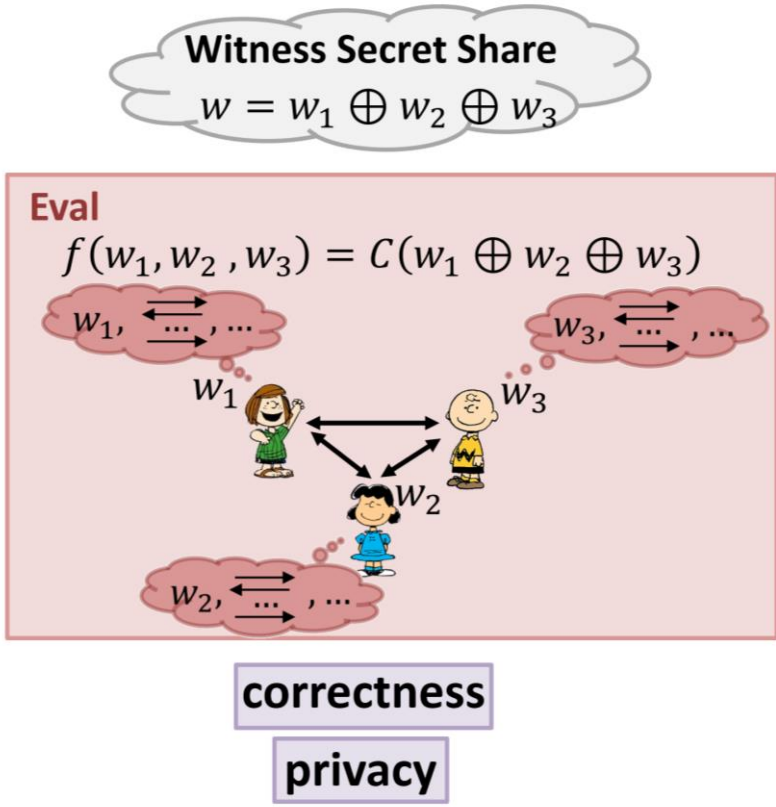
MPC correctness ⇒ soundness

MPC privacy ⇒ only 2 witness shares revealed ⇒ ZK



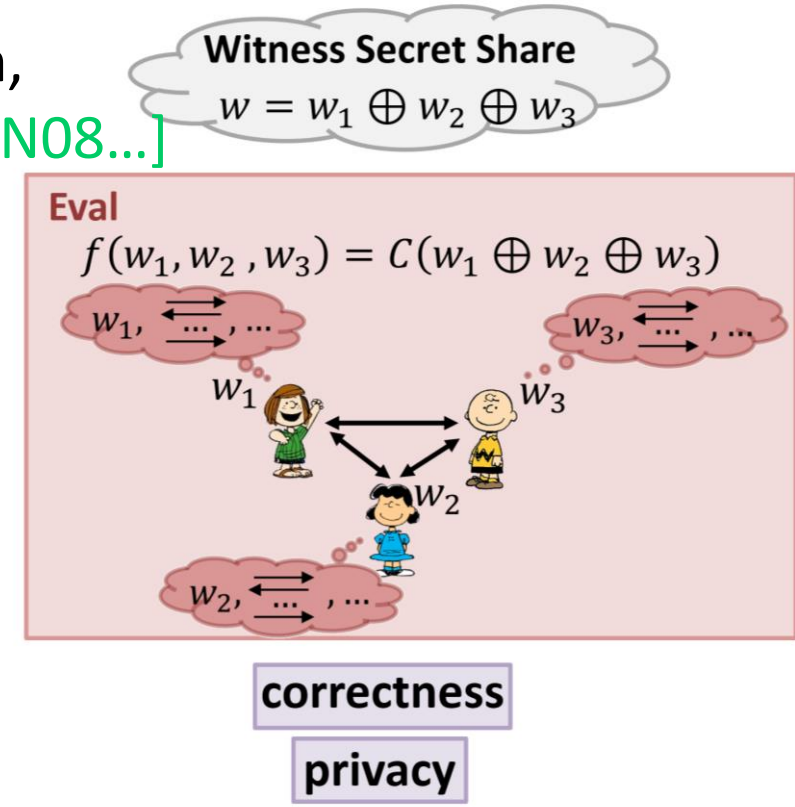
MPC in the Head: Digest

- Main ingredients:
  - Witness secret sharing
  - Evaluation of relation on secret shares
  - Correctness and privacy
    (full MPC security not needed!)



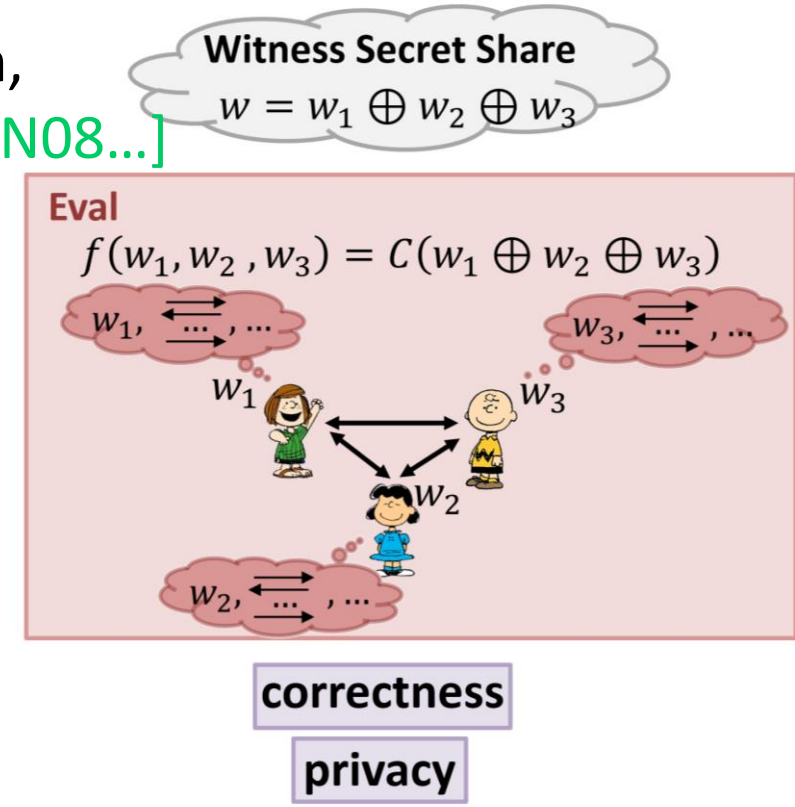
MPC in the Head: Digest

- Main ingredients:
  - Witness secret sharing
  - Evaluation of relation on secret shares
  - Correctness and privacy
    (full MPC security not needed!)
- Very influential and general paradigm, many applications [IKOS07,IPS08, HIKN08…]



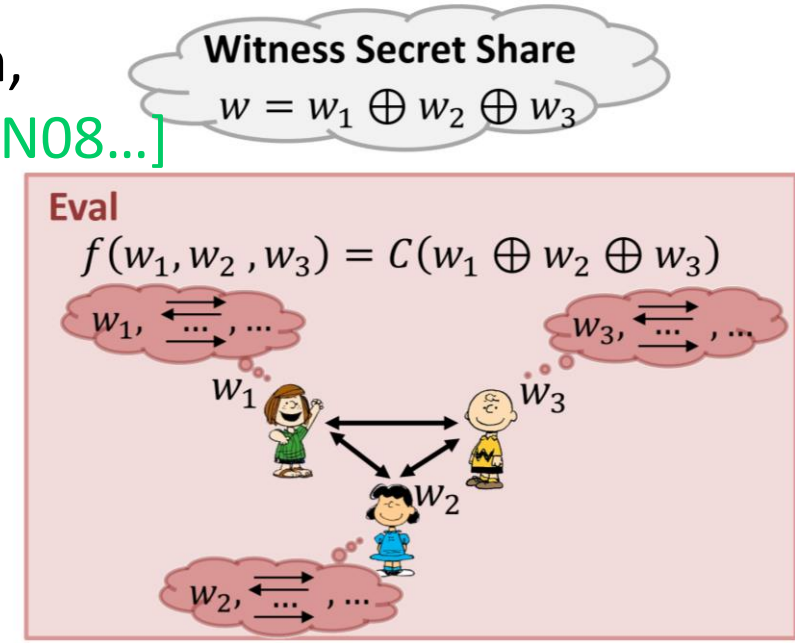
MPC in the Head: Digest

- Main ingredients:
  - Witness secret sharing
  - Evaluation of relation on secret shares
  - Correctness and privacy
    (full MPC security not needed!)
- Very influential and general paradigm, many applications [IKOS07,IPS08, HIKN08…]
- But all instantiations use *fully-secure* protocols



MPC in the Head: Digest

- Main ingredients:
  - Witness secret sharing
  - Evaluation of relation on secret shares
  - Correctness and privacy
    (full MPC security not needed!)
- Very influential and general paradigm, many applications [IKOS07,IPS08, HIKN08…]
- But all instantiations use *fully-secure* protocols



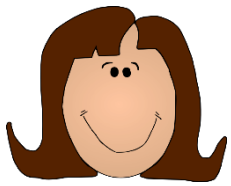
- **Today:** Beyond MPC in the Head
  - Generalized paradigm
  - From **game-based** primitives, enabling encrypting secrets and homomorphic computations

correctness

privacy

Example: ZKPs from FHE

Claim: $C(w) = 1$ for some w

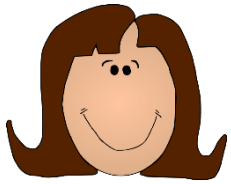


FHE: KeyGen, Enc, Eval, Dec

Example: ZKPs from FHE

Claim: $C(w) = 1$ for some w

Witness Secret Share
$w = w_1 \oplus w_2$



FHE: KeyGen, Enc, Eval, Dec

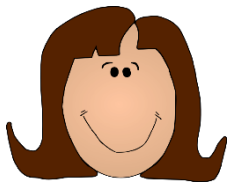
Example: ZKPs from FHE

Claim: $C(w) = 1$ for some w

Witness Secret Share
$w = w_1 \oplus w_2$

Key Gen

$(pk, sk) \leftarrow Gen(1^\kappa; r_G)$



FHE: KeyGen, Enc, Eval, Dec

Example: ZKPs from FHE

Claim: $C(w) = 1$ for some w

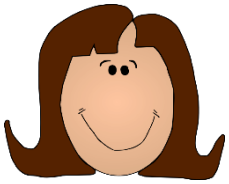
Witness Secret Share
$w = w_1 \oplus w_2$

Key Gen

$$(pk, sk) \leftarrow Gen(1^\kappa; r_G)$$

Witness Encryption

$$c \leftarrow Enc(sk, w_1; r_E)$$



FHE: KeyGen, Enc, Eval, Dec

Example: ZKPs from FHE

Claim: $C(w) = 1$ for some w

Witness Secret Share
$w = w_1 \oplus w_2$

Key Gen

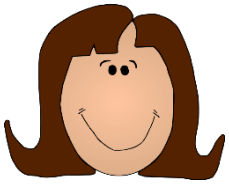
$$(pk, sk) \leftarrow Gen(1^\kappa; r_G)$$

Witness Encryption

$$c \leftarrow Enc(sk, w_1; r_E)$$

Eval

$$y \leftarrow Eval(pk, \tilde{C}, c; r_C)$$
$$\tilde{C}(u) := C(w_2 \oplus u)$$



FHE: KeyGen, Enc, Eval, Dec

Example: ZKPs from FHE

Claim: $C(w) = 1$ for some w

Witness Secret Share
$w = w_1 \oplus w_2$

Key Gen

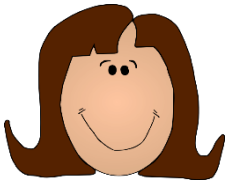
$$(pk, sk) \leftarrow Gen(1^\kappa; r_G)$$

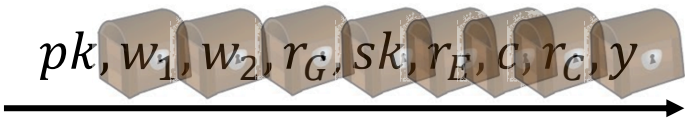
Witness Encryption

$$c \leftarrow Enc(sk, w_1; r_E)$$

Eval

$$y \leftarrow Eval(pk, \tilde{C}, c; r_C)$$
$$\tilde{C}(u) := C(w_2 \oplus u)$$





Example: ZKPs from FHE

Claim: $C(w) = 1$ for some w

Witness Secret Share
$w = w_1 \oplus w_2$

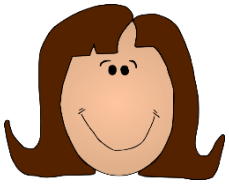
Check one phase of computation

Key Gen
$(pk, sk) \leftarrow Gen(1^\kappa; r_G)$

Witness Encryption
$c \leftarrow Enc(sk, w_1; r_E)$

Eval
$y \leftarrow Eval(pk, \tilde{C}, c; r_C)$
$\tilde{C}(u) := C(w_2 \oplus u)$

Key Gen
read sk, r_G



$pk, w_1, w_2, r_G, sk, r_E, c, r_C, y$

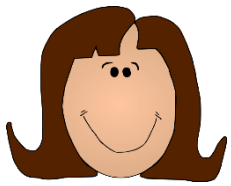
Example: ZKPs from FHE

Claim: $C(w) = 1$ for some w

Witness Secret Share
$w = w_1 \oplus w_2$

Check one phase of computation

Key Gen $(pk, sk) \leftarrow Gen(1^\kappa; r_G)$	**Key Gen** read sk, r_G
Witness Encryption $c \leftarrow Enc(sk, w_1; r_E)$	**Witness Encryption** read sk, w_1, r_E
Eval $y \leftarrow Eval(pk, \tilde{C}, c; r_C)$ $\tilde{C}(u) := C(w_2 \oplus u)$	



$pk, w_1, w_2, r_G, sk, r_E, c, r_C, y$

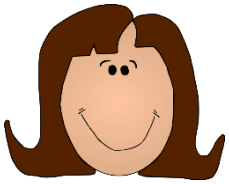
Example: ZKPs from FHE

Claim: $C(w) = 1$ for some w

Witness Secret Share
$w = w_1 \oplus w_2$

Check one phase of computation

Key Gen $(pk, sk) \leftarrow Gen(1^\kappa; r_G)$	**Key Gen** read sk, r_G
Witness Encryption $c \leftarrow Enc(sk, w_1; r_E)$	**Witness Encryption** read sk, w_1, r_E
Eval $y \leftarrow Eval(pk, \tilde{C}, c; r_C)$ $\tilde{C}(u) := C(w_2 \oplus u)$	**Eval** read y, w_2, c, r_C



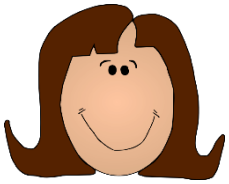
Example: ZKPs from FHE

Claim: $C(w) = 1$ for some w

Witness Secret Share
$w = w_1 \oplus w_2$

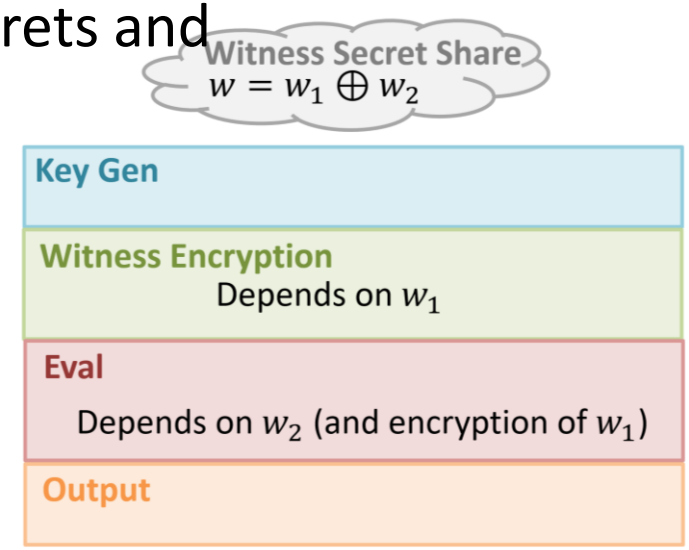
Check one phase of computation

Key Gen $(pk, sk) \leftarrow Gen(1^\kappa; r_G)$	**Key Gen** read sk, r_G
Witness Encryption $c \leftarrow Enc(sk, w_1; r_E)$	**Witness Encryption** read sk, w_1, r_E
Eval $y \leftarrow Eval(pk, \tilde{C}, c; r_C)$ $\tilde{C}(u) := C(w_2 \oplus u)$	**Eval** read y, w_2, c, r_C
	Output read sk, y



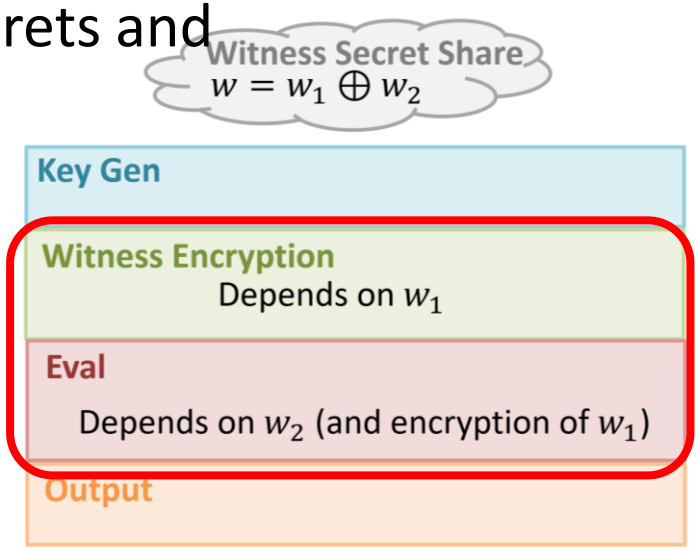
ZKPs from Game-Based Primitives (Blueprint)

- Instead of MPC, use weaker primitive
  - MPC execution replaced with executing primitive algorithms
- **Primitive syntax:** enables encrypting secrets and homomorphic computations



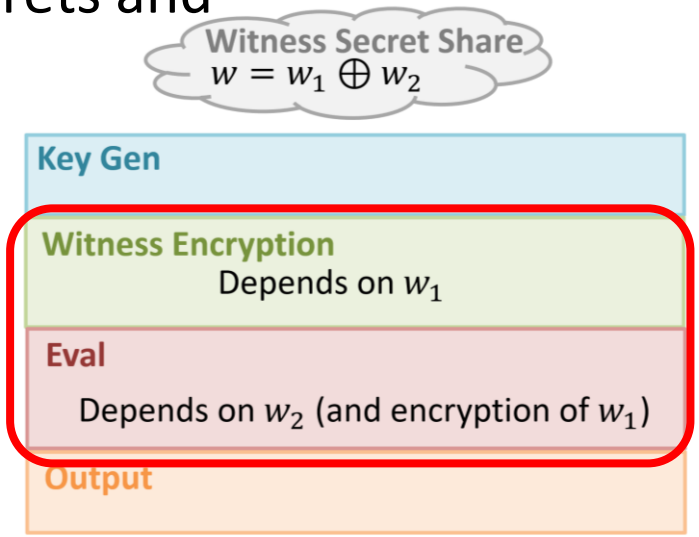
ZKPs from Game-Based Primitives (Blueprint)

- Instead of MPC, use weaker primitive
  - MPC execution replaced with executing primitive algorithms
- **Primitive syntax:** enables encrypting secrets and homomorphic computations
- **Primitive properties:**
  - Correctness
  - Input privacy: ciphertexts hide secrets
  - Function privacy: homomorphic eval hides evaluated function



ZKPs from Game-Based Primitives (Blueprint)

- Instead of MPC, use weaker primitive
  - MPC execution replaced with executing primitive algorithms
- **Primitive syntax:** enables encrypting secrets and homomorphic computations
- **Primitive properties:**
  - Correctness
  - Input privacy: ciphertexts hide secrets
  - Function privacy: homomorphic eval hides evaluated function
- **Versatile:** primitive can be
  - 1-party (FHE, FE, LFE, RE) or multi-party (HSS, FSS)
  - Interactive (IP, IOP)
  - Secret- or public-key
  - With imperfect correctness
- Extends also to commit-and-prove functionalities



New ZKP Constructions

- **ZKPs with $O(1)$ rounds and soundness error, BB in underlying primitive**

New ZKP Constructions

Circuit Class	Communication	Assumption	SotA (Same Params, NBB)
NC^1	$n \cdot poly(\kappa)$	DCR (BB in HSS)	AC^0, BB from OWFs [IKOS07]
NC^1	$n \cdot poly(\kappa)$	OWF	[GR20] ($O(\log n)$-round, NC [GKR08])
NP	$O(n + poly(\kappa))$ $O(\kappa \cdot S)$	FHE OWF	[GGIPSS15] [HV16] (BB in OWF)
$poly(m)$-size, $d(m)$-depth NP	$n \cdot poly(\kappa, d(m))$ $\boldsymbol{O(d_m)}$ **rounds**	OWF	[GKR08]
$poly(m)$-time, m^δ-space NP	$O(n) + m^\beta \cdot poly(\kappa)$	OWF	[NR22] (with communication $(1+\gamma)n + m^\beta \cdot poly(\kappa)$)

- **ZKPs with $\boldsymbol{O(1)}$ rounds and soundness error, BB in underlying primitive**
- n = witness length, κ = sec param, S = verification circuit size, m = instance length
- NC^1 results for poly-time uniform relations
- Depth-$d(m)$ result for logspace uniform relations

New ZKP Constructions

Circuit Class	Communication	Assumption	SotA (Same Params, NBB)
NC^1	$n \cdot poly(\kappa)$	DCR (BB in HSS)	AC^0, BB from OWFs [IKOS07]
NC^1	$n \cdot poly(\kappa)$	OWF	[GR20] ($O(\log n)$-round, NC [GKR08])

Round complexity depends on constant in circuit depth

- **ZKPs with $O(1)$ rounds and soundness error, BB in underlying primitive**
- n = witness length, κ = sec param, S = verification circuit size, m = instance length
- NC^1 results for poly-time uniform relations
- Depth-$d(m)$ result for logspace uniform relations

New ZKP Constructions

Circuit Class	Communication	Assumption	SotA (Same Params, NBB)
NC^1	$n \cdot poly(\kappa)$	DCR (BB in HSS)	AC^0, BB from OWFs [IKOS07]
NC^1	$n \cdot poly(\kappa)$	OWF	[GR20] ($O(\log n)$-round, NC [GKR08])
NP	$O(n + poly(\kappa))$ $O(\kappa \cdot S)$	FHE OWF	[GGIPSS15] [HV16] (BB in OWF)
$poly(m)$-size, $d(m)$-depth NP	$n \cdot poly(\kappa, d(m))$	OWF	[GKR08]
$poly(m)$-time, m^δ-space NP	$O(n) + m^\beta \cdot poly(\kappa)$	OWF	[NR22] (with communication $(1+\gamma)n + m^\beta \cdot poly(\kappa)$)

Round complexity depends on constant in circuit depth

$O(d_m)$ rounds

- **ZKPs with $O(1)$ rounds and soundness error, BB in underlying primitive**
- n = witness length, κ = sec param, S = verification circuit size, m = instance length
- NC^1 results for poly-time uniform relations
- Depth-$d(m)$ result for logspace uniform relations

Summary

- **Beyond MPC in the Head:** new paradigm for ZKP design using game-based primitives
  - With homomorphic computation properties
  - Only need correctness and privacy, not full security
  - Extends MPC in the Head of [IKOS07]

Summary

- **Beyond MPC in the Head:** new paradigm for ZKP design using game-based primitives
  - With homomorphic computation properties
  - Only need correctness and privacy, not full security
  - Extends MPC in the Head of [IKOS07]
- **Versatile:**
  - Applicable to many game-based primitives: FHE, FSS, HSS, RE, FE, LFE
  - Extend to Commit-and-Prove functionalities (can be applied also to IPs\IOPs)
  - See paper for details (ePrint 2023/1819)

Summary

- **Beyond MPC in the Head:** new paradigm for ZKP design using game-based primitives
  - With homomorphic computation properties
  - Only need correctness and privacy, not full security
  - Extends MPC in the Head of [IKOS07]
- **Versatile:**
  - Applicable to many game-based primitives: FHE, FSS, HSS, RE, FE, LFE
  - Extend to Commit-and-Prove functionalities (can be applied also to IPs\IOPs)
  - See paper for details (ePrint 2023/1819)
- **Applications:** new short (almost witness length) ZKPs:
  - New constant-round ZKPs for NC^1 (before: AC^0 [IKOS07])
  - BB alternatives to existing (non-BB) short ZKPs
  - Recasting some known ZKPs as special cases of the paradigm

Summary

- **Beyond MPC in the Head:** new paradigm for ZKP design using game-based primitives
  - With homomorphic computation properties
  - Only need correctness and privacy, not full security
  - Extends MPC in the Head of [IKOS07]
- **Versatile:**
  - Applicable to many game-based primitives: FHE, FSS, HSS, RE, FE, LFE
  - Extend to Commit-and-Prove functionalities (can be applied also to IPs\IOPs)
  - See paper for details (ePrint 2023/1819)
- **Applications:** new short (almost witness length) ZKPs:
  - New constant-round ZKPs for NC^1 (before: AC^0 [IKOS07])
  - BB alternatives to existing (non-BB) short ZKPs
  - Recasting some known ZKPs as special cases of the paradigm

Thank you!