Triply Adaptive UC-NIZK

Ran Canetti
Boston University

Pratik Sarkar
Boston University

Xiao Wang
Northwestern University

eprint: 2020/1212

Chapter I:

Introduction

Non-interactive Zero Knowledge

Prover P(x, w)

Setup: crs

Proof π

Verifier V(x)

Input: NP statement x, witness w

Output: Proof π

Input: NP statement x

Output: 0/1

Correctness: If x $\in L$ and w is a valid witness then V outputs 1

Soundness: If x $\notin L$, then V outputs 0 with high probability

(Non-Adaptive) Zero Knowledge

(Non-Adaptive) Zero Knowledge Game

Simulator Sim(x)

Setup: crs

Simulated Proof π'

Corrupt Verifier V(x)

Input: NP statement x
Samples (crs, td) =Setup.Gen(1^κ)
Output: Simulated Proof π' = Sim(x)

Input: NP statement x

(Non-Adaptive) Zero Knowledge Game

Simulator Sim(x)

Setup: crs

Simulated Proof π'

Corrupt Verifier V(x)

Input: NP statement x
Samples (crs, td) =Setup.Gen(1^{κ})
Output: Simulated Proof π' = Sim(x)

Input: NP statement x

(Non-Adaptive) Zero Knowledge: ∃PPT algorithm Sim, such that the simulated proof is indistinguishable from real proof:

$$\{crs, P(x, w)\} \approx \{crs, Sim(x)\}$$

Non-interactive Zero Knowledge

Prover P(x, w)

Setup: crs

Verifier V(x)

Proof π

Input: NP statement x, witness w

Output: Proof π

Correctness

Soundness

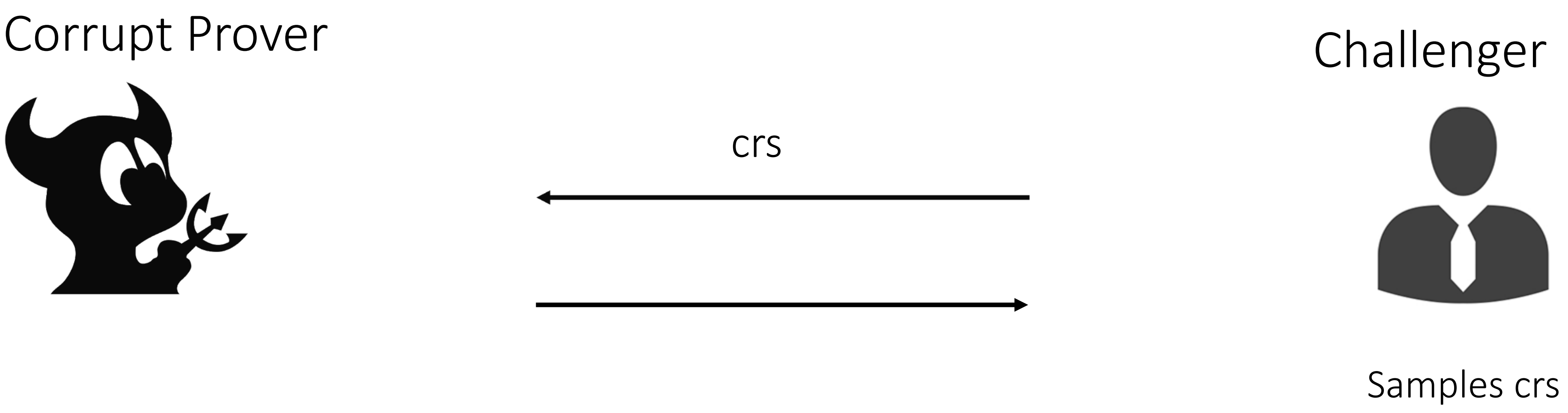
(Non-Adaptive) Zero Knowledge

Input: NP statement x

Triply Adaptive NIZK

Adaptive Soundness   Adaptive Zero Knowledge   Adaptive Security

Adaptive Soundness Game



Adaptive Soundness Game

Corrupt Prover

crs ←

Statement x, Proof π →

Challenger

Samples crs
Outputs V(x, π; crs)

Adaptive Soundness: If x $\notin L$, then Challenger outputs 0 with high probability

Adaptive Soundness Game

Corrupt Prover

Challenger

crs ←

Statement x, Proof π →

Samples crs

Outputs V(x, π; crs)

Adaptive Soundness: If x $\notin L$, then Challenger outputs 0 with high probability

Adaptive Soundness is stronger than soundness. [GroOsSah12] is sound but not adaptively sound

Triply Adaptive NIZK

Adaptive Soundness	Adaptive Zero Knowledge	Adaptive Security
Corrupt prover chooses statement **x** after seeing **crs** Soundness preserved		

Adaptive Zero Knowledge Game

Simulator Sim(x)

Sim samples (crs, td)

Setup: crs

Statement x

Corrupt Verifier V(x)

Samples (x, w) $\in L$ after obtaining crs

Adaptive Zero Knowledge Game

Simulator Sim(x)

Setup: crs

Statement x

Simulated Proof π'

Corrupt Verifier V(x)

Input: NP statement x
Sim samples (crs, td)
Output: Simulated Proof π' = Sim(x)

Samples (x, w) ∈ L after obtaining crs

Adaptive Zero Knowledge Game

Simulator Sim(x)

Setup: crs

Corrupt Verifier V(x)

Statement x

Simulated Proof π'

Input: NP statement x
Sim samples (crs, td)
Output: Simulated Proof π' = Sim(x)

Samples (x, w) ∈ L after obtaining crs

Adaptive Zero Knowledge: ∃PPT algorithm Sim, such that the simulated proof is indistinguishable from real proof:

$$\{crs, P(x, w)\} \approx \{crs, Sim(x)\}$$

Triply Adaptive NIZK

Adaptive Soundness	Adaptive Zero Knowledge	Adaptive Security
Corrupt prover chooses statement **x** after seeing **crs**	Corrupt verifier who chooses statement **x** after seeing **crs**	
Soundness preserved	Zero-Knowledge preserved	

Security against Adaptive Corruptions

Simulator Sim(x)

Setup: crs

Statement x

Corrupt Verifier V(x)

Sim samples (crs, td)

Samples (x, w) ∈ L after obtaining crs

Security against Adaptive Corruptions

Simulator Sim(x)

Setup: crs

Statement x

Simulated Proof π'

Corrupt Verifier V(x)

Input: NP statement x
Sim samples (crs, td)
Output: Simulated Proof π'= Sim_1(x; r')

Samples (x, w) ∈ L after obtaining crs

Zero Knowledge: ∃PPT algorithm Sim_1, such that the simulated proof is indistinguishable from real proof:

$$\{\text{crs}, P(x, w; r)\} \approx \{\text{crs}, \text{Sim}_1(x; r')\}$$

Security against Adaptive Corruptions

Corrupt Prover P(x)

Setup: crs

Statement x

Simulated Proof π'

Corrupt Verifier V(x)

Input: NP statement x
Sim samples (crs, td)
Output: Simulated Proof π'= Sim_1(x; r')

Samples (x, w) ∈ L after obtaining crs

Internal State: Randomness Sim_2(w, r')

Zero Knowledge: ∃PPT algorithm Sim_1, such that the simulated proof is indistinguishable from real proof:

$$\{\mathrm{crs}, P(x, w; r)\} \approx \{\mathrm{crs}, \mathrm{Sim}_1(x; r')\}$$

Security against Adaptive Corruptions

Corrupt Prover P(x)

Setup: crs

Statement x

Simulated Proof π'

Corrupt Verifier V(x)

Input: NP statement x
Sim samples (crs, td)
Output: Simulated Proof π'= Sim_1(x; r’)

Samples (x, w) ∈ L after obtaining crs

Internal State: Randomness Sim_2(w, r’)

Zero Knowledge: ∃PPT algorithm Sim_1, such that the simulated proof is indistinguishable from real proof:

$$\{\mathrm{crs}, P(x, w; r)\} \approx \{\mathrm{crs}, \mathrm{Sim}_1(x; r')\}$$

Security against Adaptive Corruption: ∃PPT algorithm Sim_2, such that:

$$\{\mathrm{crs}, P(x, w; r), r\} \approx \{\mathrm{crs}, \mathrm{Sim}_1(x; r'), \mathrm{Sim}_2(w, r')\}$$

Triply Adaptive NIZK

Adaptive Soundness	Adaptive Zero Knowledge	Adaptive Security
Corrupt prover chooses statement **x** after seeing **crs**	Corrupt verifier who chooses statement **x** after seeing **crs**	Security against adaptive corruption of prover
Soundness preserved	Zero-Knowledge preserved	

Triply Adaptive NIZK

Adaptive Soundness	Adaptive Zero Knowledge	Adaptive Security
Corrupt prover chooses statement **x** after seeing **crs**	Corrupt verifier who chooses statement **x** after seeing **crs**	Security against adaptive corruption of prover
Soundness preserved	Zero-Knowledge preserved	

Realistic Security Guarantees: The Prover uses the same crs to prove <u>adaptively chosen</u> statements
Security against adaptive corruptions, useful for MPC protocols

Triply Adaptive NIZK

Adaptive Soundness	Adaptive Zero Knowledge	Adaptive Security
Corrupt prover chooses statement **x** after seeing **crs**	Corrupt verifier who chooses statement **x** after seeing **crs**	Security against adaptive corruption of prover
Soundness preserved	Zero-Knowledge preserved	

Realistic Security Guarantees: The Prover uses the same crs to prove <u>adaptively chosen</u> statements
Security against adaptive corruptions, useful for MPC protocols

UC-Security: Extendable to the provide UC security and reusable crs model across multiple sessions between different parties

State-of-the-art and Our Main Result

Protocols	Adaptive Soundness	Adaptive Zero Knowledge	Adaptive Security (against adaptive corruptions)	Assumptions
[GroOstSah06]*	✗	✓	✓	Pairings

*Achieves Adaptive culpable soundness which is weaker than adaptive soundness

State-of-the-art and Our Main Result

Protocols	Adaptive Soundness	Adaptive Zero Knowledge	Adaptive Security (against adaptive corruptions)	Assumptions
[GroOstSah06]*	✗	✓	✓	Pairings
[KatNisYamYam19, KatNisYayYam20]*	✗	✓	✓	Pairings

*Achieves Adaptive culpable soundness which is weaker than adaptive soundness

State-of-the-art and Our Main Result

Protocols	Adaptive Soundness	Adaptive Zero Knowledge	Adaptive Security (against adaptive corruptions)	Assumptions
[GroOstSah06]*	✗	✓	✓	Pairings
[KatNisYamYam19, KatNisYayYam20]*	✗	✓	✓	Pairings
[AbeFeh07]	✓	✓	✓	Knowledge Assumptions

*Achieves Adaptive culpable soundness which is weaker than adaptive soundness

State-of-the-art and Our Main Result

Protocols	Adaptive Soundness	Adaptive Zero Knowledge	Adaptive Security (against adaptive corruptions)	Assumptions
[GroOstSah06]*	✗	✓	✓	Pairings
[KatNisYamYam19, KatNisYayYam20]*	✗	✓	✓	Pairings
[AbeFeh07]	✓	✓	✓	Knowledge Assumptions
CI-based Protocols [CCH+19,PS19,BKM20]	✓	✓	✗	LWE/ DDH+LPN

*Achieves Adaptive culpable soundness which is weaker than adaptive soundness

State-of-the-art and Our Main Result

Protocols	Adaptive Soundness	Adaptive Zero Knowledge	Adaptive Security (against adaptive corruptions)	Assumptions
[GroOstSah06]*	✗	✓	✓	Pairings
[KatNisYamYam19, KatNisYayYam20]*	✗	✓	✓	Pairings
[AbeFeh07]	✓	✓	✓	Knowledge Assumptions
CI-based Protocols [CCH+19,PS19,BKM20]	✓	✓	✗	LWE/ DDH+LPN
Ours	✓	✓	✓	LWE/ DDH+LPN

*Achieves Adaptive culpable soundness which is weaker than adaptive soundness

Challenges and Ideas

Correlation Intractability | Adaptive Soundness | Ideas

Correlation Intractability (CI) based Protocols require the initial interactive protocol to be statistically sound

This contradicts adaptive security as statistically sound protocols cannot be equivocated upon adaptive corruption

Challenges and Ideas

Correlation Intractability	Adaptive Soundness	Ideas
Correlation Intractability (CI) based Protocols require the initial interactive protocol to be statistically sound This contradicts adaptive security as statistically sound protocols cannot be equivocated upon adaptive corruption	Previous adaptively secure NIZKs [GOS12] (with non-adaptive soundness) switch the crs mode to perform equivocation Adaptive soundness prevents us from switching the mode of crs	

Challenges and Ideas

Correlation Intractability	Adaptive Soundness	Ideas
Correlation Intractability (CI) based Protocols require the initial interactive protocol to be statistically sound This contradicts adaptive security as statistically sound protocols cannot be equivocated upon adaptive corruption	Previous adaptively secure NIZKs [GOS12] (with non-adaptive soundness) switch the crs mode to perform equivocation Adaptive soundness prevents us from switching the mode of crs	Perform Fiat-Shamir for interactive arguments - Rely on CI in the hybrids

Challenges and Ideas

Correlation Intractability	Adaptive Soundness	Ideas
Correlation Intractability (CI) based Protocols require the initial interactive protocol to be statistically sound	Previous adaptively secure NIZKs [GOS12] (with non-adaptive soundness) switch the crs mode to perform equivocation	Perform Fiat-Shamir for interactive arguments - Rely on CI in the hybrids
This contradicts adaptive security as statistically sound protocols cannot be equivocated upon adaptive corruption	Adaptive soundness prevents us from switching the mode of crs	Underlying argument is only computationally binding and hence equivocal

Challenges and Ideas

Correlation Intractability	Adaptive Soundness	Ideas
Correlation Intractability (CI) based Protocols require the initial interactive protocol to be statistically sound	Previous adaptively secure NIZKs [GOS12] (with non-adaptive soundness) switch the crs mode to perform equivocation	Perform Fiat-Shamir for interactive arguments - Rely on CI in the hybrids
This contradicts adaptive security as statistically sound protocols cannot be equivocated upon adaptive corruption		Underlying argument is only computationally binding and hence equivocal
	Adaptive soundness prevents us from switching the mode of crs	Perform the soundness argument without switching crs mode – enables adaptive soundness

Our Contributions

Non-interactive UC-Commitment Functionality $\mathcal{F}_{\text{NICOM}}$

Parties access $\mathcal{F}_{\text{NICOM}}$ locally for Commitment generation and verification

Functionality outputs commitment string during Commit phase

Protocol Friendly

Our Contributions

Non-interactive UC-Commitment Functionality $\mathcal{F}_{\text{NICOM}}$	Triply Adaptive NIZK
Parties access $\mathcal{F}_{\text{NICOM}}$ locally for Commitment generation and verification	**Triply adaptive** Sigma protocol in $\mathcal{F}_{\text{NICOM}}$ model
Functionality outputs commitment string during Commit phase	Compile the above Sigma protocol to obtain Triply adaptive NIZK
Protocol Friendly	Apply Correlation Intractability for NIZK arguments

Our Contributions

Non-interactive UC-Commitment Functionality $\mathcal{F}_{\text{NICOM}}$	Triply Adaptive NIZK	Instantiations
Parties access $\mathcal{F}_{\text{NICOM}}$ locally for Commitment generation and verification	**Triply adaptive** Sigma protocol in $\mathcal{F}_{\text{NICOM}}$ model	Most Sigma protocols are **Triply adaptive** in $\mathcal{F}_{\text{NICOM}}$ model
Functionality outputs commitment string during Commit phase	Compile the above Sigma protocol to obtain Triply adaptive NIZK	Implement $\mathcal{F}_{\text{NICOM}}$ with [CanFis01] commitment scheme
Protocol Friendly	Apply Correlation Intractability for NIZK arguments	

Our Contributions

Non-interactive UC-Commitment Functionality $\mathcal{F}_{\text{NICOM}}$	Triply Adaptive NIZK	Instantiations
Parties access $\mathcal{F}_{\text{NICOM}}$ locally for Commitment generation and verification	**Triply adaptive** Sigma protocol in $\mathcal{F}_{\text{NICOM}}$ model	Most Sigma protocols are **Triply adaptive** in $\mathcal{F}_{\text{NICOM}}$ model
Functionality outputs commitment string during Commit phase	Compile the above Sigma protocol to obtain Triply adaptive NIZK	Implement $\mathcal{F}_{\text{NICOM}}$ with [CanFis01] commitment scheme
Protocol Friendly	Apply Correlation Intractability for NIZK arguments	

UC-Security: Obtain UC-security using standard tricks [GosOstSah12]

Our Contributions

Non-interactive UC-Commitment Functionality $\mathcal{F}_{\text{NICOM}}$	**Triply Adaptive NIZK**	**Instantiations**
Parties access $\mathcal{F}_{\text{NICOM}}$ locally for Commitment generation and verification	**Triply adaptive** Sigma protocol in $\mathcal{F}_{\text{NICOM}}$ model	Most Sigma protocols are **Triply adaptive** in $\mathcal{F}_{\text{NICOM}}$ model
Functionality outputs commitment string during Commit phase	Compile the above Sigma protocol to obtain Triply adaptive NIZK	Implement $\mathcal{F}_{\text{NICOM}}$ with [CanFis01] commitment scheme
Protocol Friendly	Apply Correlation Intractability for NIZK arguments	

UC-Security: Obtain UC-security using standard tricks [GosOstSah12]

Chapter II:

Non-interactive
UC commitment functionality

Non-interactive Commitment Functionality $\mathcal{F}_{\text{NICOM}}$

Ideal world Adversary

Initialize $\mathcal{F}_{\text{NICOM}}$ with S

$\mathcal{F}_{\text{NICOM}} \longleftarrow$

Non-interactive Commitment Functionality $\mathcal{F}_{\text{NICOM}}$

Ideal world Adversary

Initialize $\mathcal{F}_{\text{NICOM}}$ with $\mathcal{S}$

Commiter (m)

$\mathcal{F}_{\text{NICOM}}$

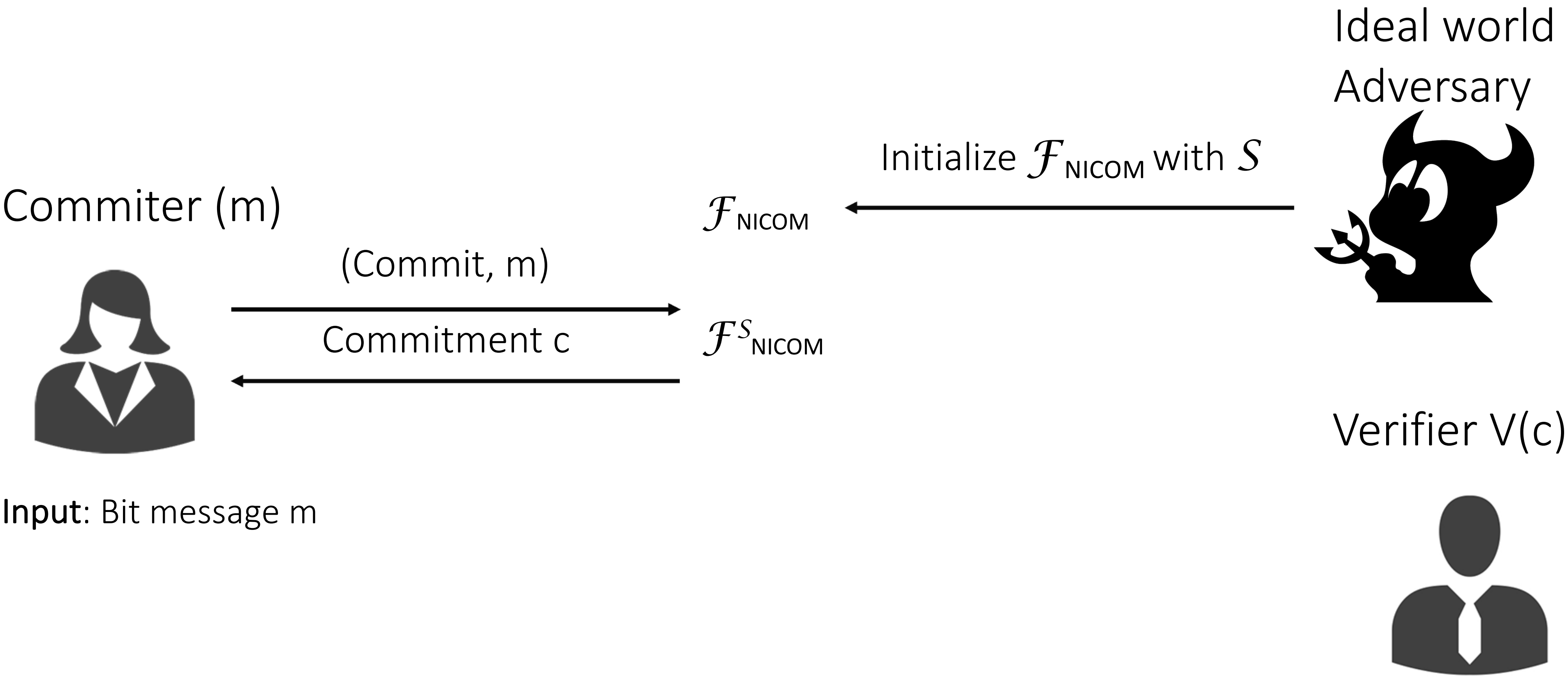
(Commit, m)

Commitment c

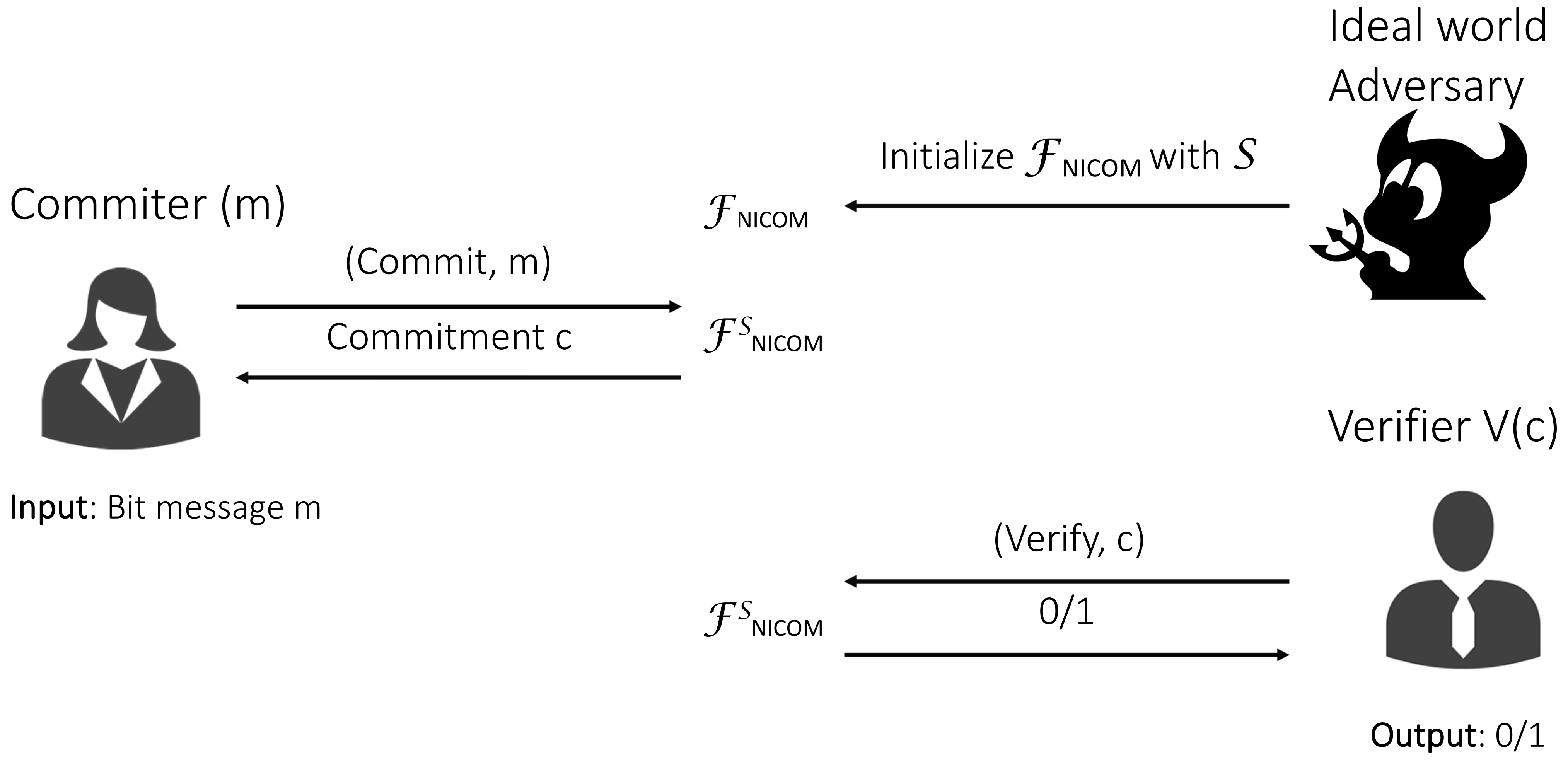
$\mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$

Input: Bit message m

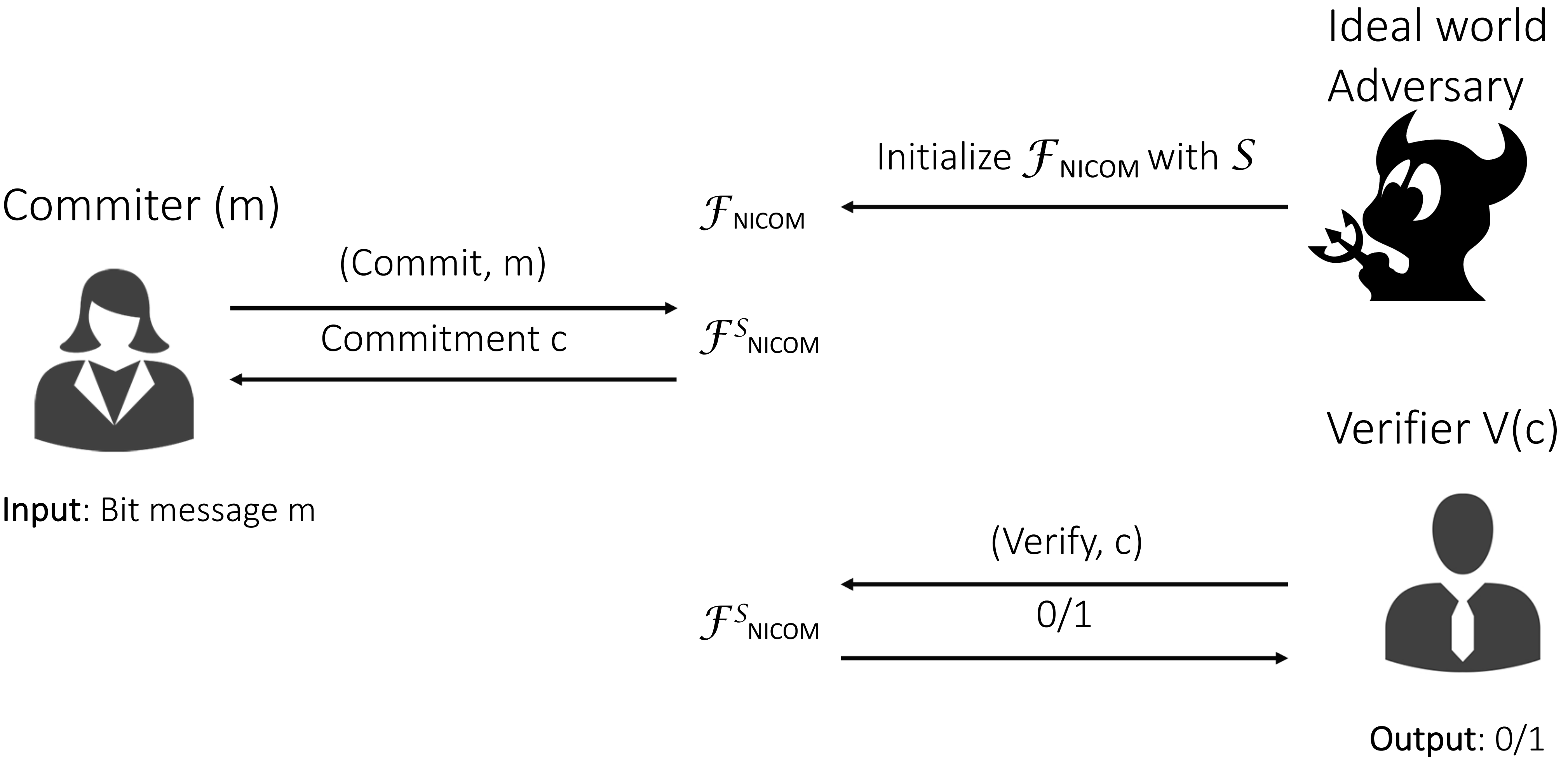
Non-interactive Commitment Functionality $\mathcal{F}_{\text{NICOM}}$



Non-interactive Commitment Functionality $\mathcal{F}_{\text{NICOM}}$



Non-interactive Commitment Functionality $\mathcal{F}_{\text{NICOM}}$



[CanFis01]: If there exists an **equivocal commitment scheme** and a **CCA-2 secure public key encryption scheme with oblivious ciphertext sampling**, then there exists a commitment scheme implementing $\mathcal{F}_{\text{NICOM}}$

Our Contributions

Non-interactive UC-Commitment Functionality $\mathcal{F}_{\text{NICOM}}$	Triply Adaptive NIZK	Instantiations
Parties access $\mathcal{F}_{\text{NICOM}}$ locally for Commitment generation and verification	**Triply adaptive** Sigma protocol in $\mathcal{F}_{\text{NICOM}}$ model	Most Sigma protocols are **Triply adaptive** in $\mathcal{F}_{\text{NICOM}}$ model
Functionality outputs commitment string during Commit phase	Compile the above Sigma protocol to obtain Triply adaptive NIZK	Implement $\mathcal{F}_{\text{NICOM}}$ with [CanFis01] commitment scheme
Protocol Friendly	Apply Correlation Intractability for NIZK arguments	

UC-Security: Obtain UC-security using standard tricks [GosOstSah12]

Chapter III:

Adaptively Secure Sigma Protocol

Sigma Protocol

Prover P(x, w)

Verifier V(x)

$$\text{Prover} \xrightarrow{a} \text{Verifier}$$
$$\text{Prover} \xleftarrow{e} \text{Verifier}$$
$$\text{Prover} \xrightarrow{z} \text{Verifier}$$

Input: NP statement x, witness w

Output: Proof π

Input: NP statement x

Samples challenge e ∈ C

Output: 0/1

Correctness: If x ∈ L and w is a valid witness then V(x, a, e, z) outputs 1

Special Soundness: If a corrupt prover outputs two accepting proofs (a, e, z) and (a, e’, z’) then there exists PPT witness extractor algorithm :

$$\text{Ext}(x, a, e, e', z, z') = w \text{ if } V(x, a, e, z) = V(x, a, e', z') = 1 \text{ for } e \neq e'$$

Honest Verifier Zero Knowledge: ∃PPT algorithm Sim, such that HVZK proof is indistinguishable from real proof:

$$P(x, w) \approx \text{Sim}(x, e)$$

(where e ∈ C is a random challenge)

Adaptively Secure Sigma Protocol in $\mathcal{F}^{S}{}_{\text{NICOM}}$ model

Prover P(x, w)

$\longleftrightarrow \mathcal{F}^{S}{}_{\text{NICOM}}$

Input: NP statement x,
witness w

Output: Proof π

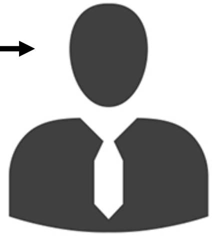
$$\xrightarrow{\quad a \quad}$$

$$\xleftarrow{\quad e \quad}$$

$$\xrightarrow{\quad z \quad}$$

Verifier V(x)

$\mathcal{F}^{S}{}_{\text{NICOM}} \longleftrightarrow$



Input: NP statement x

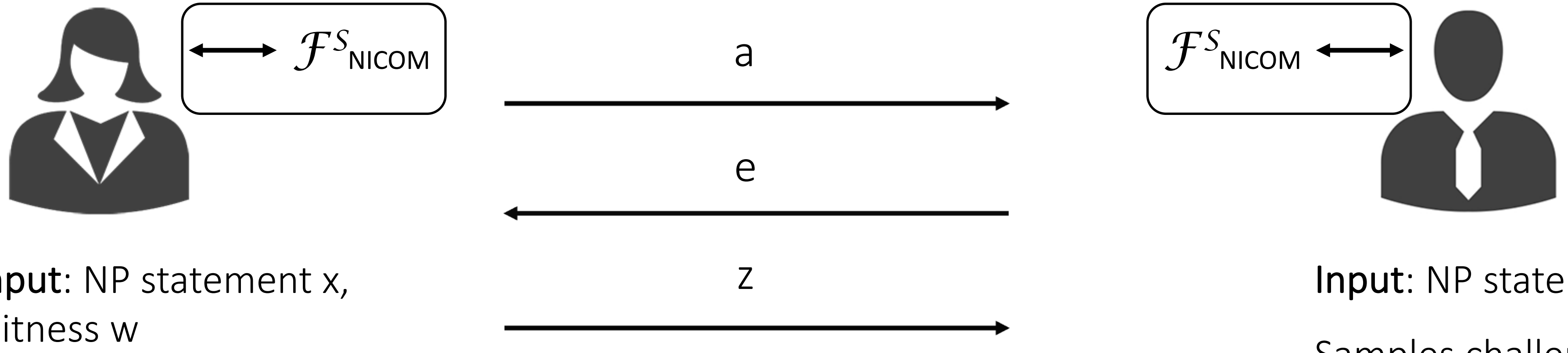
Samples challenge e $\in \mathcal{C}$

Output: 0/1

Adaptively Secure Sigma Protocol in $\mathcal{F}^{S}{}_{\text{NICOM}}$ model

Prover P(x, w)

Verifier V(x)



Input: NP statement x, witness w

Output: Proof π

Input: NP statement x

Samples challenge e $\in \mathcal{C}$

Output: 0/1

Adaptively Secure Sigma Protocol in $\mathcal{F}^{S}_{\text{NICOM}}$ model

Prover P(x, w)

Verifier V(x)

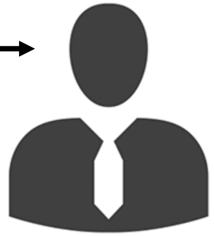
$\longleftrightarrow \mathcal{F}^{S}_{\text{NICOM}}$

$$\xrightarrow{\quad a \quad}$$

$$\xleftarrow{\quad e \quad}$$

$$\xrightarrow{\quad z \quad}$$

$\mathcal{F}^{S}_{\text{NICOM}} \longleftrightarrow$



Input: NP statement x, witness w

Output: Proof π

Input: NP statement x

Samples challenge e $\in \mathcal{C}$

Output: 0/1

Correctness, Special Soundness: Same as Sigma protocol

Adaptively Secure Sigma Protocol in $\mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$ model

Prover P(x, w)

Verifier V(x)

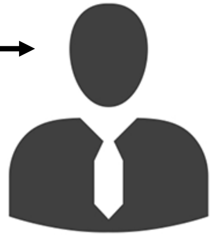
$\longleftrightarrow \mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$

$\xrightarrow{\quad a \quad}$

$\xleftarrow{\quad e \quad}$

$\xrightarrow{\quad z \quad}$

$\mathcal{F}^{\mathcal{S}}_{\text{NICOM}} \longleftrightarrow$



Input: NP statement x, witness w

Output: Proof π

Input: NP statement x

Samples challenge e ∈ $\mathcal{C}$

Output: 0/1

Correctness, Special Soundness: Same as Sigma protocol

Honest Verifier Zero Knowledge: ∃PPT algorithm $\text{Sim}_1^{\mathcal{S}}$, such that HVZK proof is indistinguishable from real proof:

$$P(x, w; r) \approx \text{Sim}_1^{\mathcal{S}}(x, e; r')$$

(where e ∈ $\mathcal{C}$ is a random challenge, $\mathcal{S}$ is the Simulator for $\mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$)

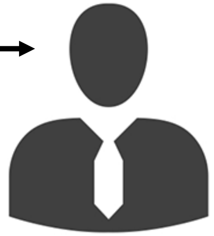
Adaptively Secure Sigma Protocol in $\mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$ model

Prover P(x, w)

$\longleftrightarrow \mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$

Verifier V(x)

$\mathcal{F}^{\mathcal{S}}_{\text{NICOM}} \longleftrightarrow$



$$\xrightarrow{\quad a \quad}$$

$$\xleftarrow{\quad e \quad}$$

$$\xrightarrow{\quad z \quad}$$

Input: NP statement x, witness w

Output: Proof π

Input: NP statement x

Samples challenge e ∈ $\mathcal{C}$

Output: 0/1

Correctness, Special Soundness: Same as Sigma protocol

Honest Verifier Zero Knowledge: ∃PPT algorithm $\text{Sim}_1^{\mathcal{S}}$, such that HVZK proof is indistinguishable from real proof:

$$P(x, w; r) \approx \text{Sim}_1^{\mathcal{S}}(x, e; r')$$

(where e ∈ $\mathcal{C}$ is a random challenge, $\mathcal{S}$ is the Simulator for $\mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$)

Adaptive Security: ∃PPT algorithm $\text{Sim}_2^{\mathcal{S}}$, such that:

$$\{\text{crs}, P(x, w; r), r\} \approx \{\text{crs}, \text{Sim}_1^{\mathcal{S}}(x, e; r'), \text{Sim}_2^{\mathcal{S}}(w, r')\}$$

Adaptively Secure Sigma Protocol in $\mathcal{F}^{S}_{\text{NICOM}}$ model

Prover P(x, w)

Verifier V(x)

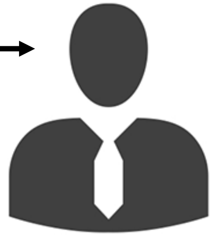
$\longleftrightarrow \mathcal{F}^{S}_{\text{NICOM}}$

$\xrightarrow{\quad a \quad}$

$\xleftarrow{\quad e \quad}$

$\xrightarrow{\quad z \quad}$

$\mathcal{F}^{S}_{\text{NICOM}} \longleftrightarrow$



Input: NP statement x, witness w

Output: Proof π

Input: NP statement x

Samples challenge e ∈ C

Output: 0/1

Correctness, Special Soundness: Same as Sigma protocol

Adaptive Secure Honest Verifier Zero Knowledge: ∃PPT algorithm ($\text{Sim}_1^{\mathcal{S}}$, $\text{Sim}_2^{\mathcal{S}}$) such that HVZK proof is indistinguishable from real proof:

$$\{\text{crs}, P(x, w; r), r\} \approx \{\text{crs}, \text{Sim}_1^{S}(x, e; r'), \text{Sim}_2^{S}(w, r')\}$$

(where e ∈ C is a random challenge, $\mathcal{S}$ is the Simulator for $\mathcal{F}^{S}_{\text{NICOM}}$)

Adaptively Secure Sigma Protocol in $\mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$ model

Prover P(x, w)

$\longleftrightarrow \mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$

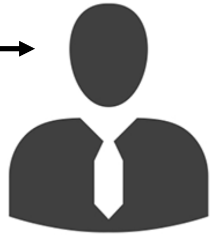
$\xrightarrow{\quad a \quad}$

$\xleftarrow{\quad e \quad}$

$\xrightarrow{\quad z \quad}$

Verifier V(x)

$\mathcal{F}^{\mathcal{S}}_{\text{NICOM}} \longleftrightarrow$



Input: NP statement x,
witness w

Output: Proof π

Input: NP statement x

Samples challenge e $\in \mathcal{C}$

Output: 0/1

Correctness, Special Soundness: Same as Sigma protocol

Adaptive Secure Honest Verifier Zero Knowledge: ∃PPT algorithm ($\text{Sim}_1^{\mathcal{S}}$, $\text{Sim}_2^{\mathcal{S}}$) such that HVZK proof is indistinguishable from real proof:

$$\{\text{crs}, P(x, w; r), r\} \approx \{\text{crs}, \text{Sim}_1^{S}(x, e; r'), \text{Sim}_2^{S}(w, r')\}$$

(where e $\in \mathcal{C}$ is a random challenge, $\mathcal{S}$ is the Simulator for $\mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$)

Next Step: Compile to an adaptively secure NIZK

Our Contributions

Non-interactive UC-Commitment Functionality $\mathcal{F}_{\text{NICOM}}$	Triply Adaptive NIZK	Instantiations
Parties access $\mathcal{F}_{\text{NICOM}}$ locally for Commitment generation and verification	**Triply adaptive** Sigma protocol in $\mathcal{F}_{\text{NICOM}}$ model	Most Sigma protocols are **Triply adaptive** in $\mathcal{F}_{\text{NICOM}}$ model
Functionality outputs commitment string during Commit phase	Compile the above Sigma protocol to obtain Triply adaptive NIZK	Implement $\mathcal{F}_{\text{NICOM}}$ with [CanFis01] commitment scheme
Protocol Friendly	Apply Correlation Intractability for NIZK arguments	

UC-Security: Obtain UC-security using standard tricks [GosOstSah12]

Chapter IV:

Preliminaries for NIZK

Fiat Shamir Transform

Sigma Protocol

Prover P(x, w)

Input: NP statement x, witness w

Output: Proof π = (a, e, z)

$\xrightarrow{\quad a \quad}$

$\xleftarrow{\quad e \quad}$

$\xrightarrow{\quad z \quad}$

Verifier V(x)

Input: NP statement x

Output: V(x, a, e, z)

Fiat Shamir Transform

Sigma Protocol → NIZK

Prover P(x, w)

Setup: Hash function h

Verifier V(x)

$a \rightarrow$

$e \leftarrow$

$z \rightarrow$

a, z $\rightarrow$

Input: NP statement x, witness w

Output: Proof π = (a, e, z)

e = h(a)

Input: NP statement x

Compute e = h(a)

Output: V(x, a, e, z)

Correlation Intractability [CCH+19, PS19, BKM20]

A hash family H is *correlation intractable* for a sparse relation R if:

Given $\mathrm{h} \in_R \mathrm{H}$, infeasible to find x s.t. (x, h(x)) ∈ R

∀PPT adversaries A,

$$\Pr_{\substack{h \leftarrow H \\ x \leftarrow A(h)}} [(\mathrm{x}, \mathrm{h}(\mathrm{x})) \in R] = negli(\kappa)$$

Example: for a function f, let $R_f = \{(\mathrm{x}, \mathrm{f}(\mathrm{x}))\}$

Fiat Shamir Transform : CI-based Instantiation

Sigma Protocol → NIZK

Prover P(x, w)

Setup: CI-Hash h for R_Π

a, z

Verifier V(x)

Input: NP statement x, witness w

e = h(a)

Output: Proof π = (a, e, z)

Input: NP statement x

Compute e = h(a)

Output: V(x, a, e, z)

Consider $R_\Pi = \{(\mathrm{a}, \mathrm{e}) : \exists \mathrm{z} \text{ s.t. Verifier accepts } (\mathrm{x}, \mathrm{a}, \mathrm{e}, \mathrm{z})\}$

Fiat Shamir Transform : CI-based Instantiation

Sigma Protocol → NIZK

Prover P(x, w)

Setup: CI-Hash h for R_Π

a, z

$\longrightarrow$

Verifier V(x)

Input: NP statement x, witness w

e = h(a)

Output: Proof π = (a, e, z)

Input: NP statement x

Compute e = h(a)

Output: V(x, a, e, z)

Consider $R_\Pi = \{(\mathrm{a}, \mathrm{e}) : \exists \mathrm{z}\ \mathrm{s.t.\ Verifier\ accepts}\ (\mathrm{x}, \mathrm{a}, \mathrm{e}, \mathrm{z}))\}$

Correctness: If x ∈ L and w is a valid witness then V(x, a, e, z) outputs 1
Soundness: If x ∉ L, then V(x, a, z) outputs 0 for a PPT Prover P
Zero Knowledge: ∃PPT algorithm Sim, such that the simulated proof is indistinguishable from real proof:

P(x, w) ≈ Sim(x), (where h is sampled by Sim in ideal world)

Chapter V:

Triply Adaptively Secure NIZK Protocol

Adaptively Secure Sigma protocol Σ ➡ Adaptively Secure NIZK

Prover P(x, w)

Setup: CI-Hash h^S for R_{f}

Verifier V(x)

$\longleftrightarrow \mathcal{F}^{S}_{\text{NICOM}}$

α, γ $\longrightarrow$

$\mathcal{F}^{S}_{\text{NICOM}} \longleftrightarrow$

Input: NP statement x, witness w

Output: Proof π = (α, γ)

Input: NP statement x

Samples challenge e ∈ C

Output: 0/1

Adaptively Secure Sigma protocol $\Sigma \longrightarrow$ Adaptively Secure NIZK

Prover P(x, w)

Setup: CI-Hash h^S for R_f

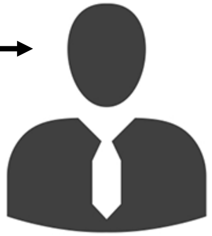
Verifier V(x)

$\longleftrightarrow \mathcal{F}^S_{\text{NICOM}}$

α, γ

$\longrightarrow$

$\mathcal{F}^S_{\text{NICOM}} \longleftrightarrow$



Input: NP statement x,
witness w

Output: Proof $\pi = (\alpha, \gamma)$

Compute two transcripts (a, c_0, z_0), (a, c_1, z_1) for the same first message for prover chosen challenges $c_0 \neq c_1 \in \mathcal{C}$:

$(a, c_0, c_1, z_0, z_1) = \Sigma.P(x, w; r)$

Input: NP statement x

Samples challenge $e \in \mathcal{C}$

Output: 0/1

Adaptively Secure Sigma protocol $\Sigma \longrightarrow$ Adaptively Secure NIZK

Prover P(x, w)

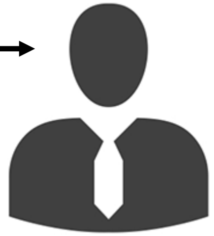
Setup: CI-Hash h^{S} for R_{f}

Verifier V(x)

$\leftrightarrow \mathcal{F}^{S}_{\text{NICOM}}$

α, γ $\longrightarrow$

$\mathcal{F}^{S}_{\text{NICOM}} \leftrightarrow$



Input: NP statement x,
witness w

Output: Proof π = (α, γ)

Compute two transcripts (a, c_0, z_0), (a, c_1, z_1) for the same first message for prover chosen challenges $c_0 \neq c_1 \in \mathcal{C}$:

(a, c_0, c_1, z_0, z_1) = Σ.P(x, w; r)

Commit to challenge as (C, δ^{c}) = $\mathcal{F}^{S}_{\text{NICOM}}$ (c_0, c_1)

Commit to responses as (Z_0 , δ_0) = $\mathcal{F}^{S}_{\text{NICOM}}$($z_0$), ($Z_1$, δ_1) = $\mathcal{F}^{S}_{\text{NICOM}}$($z_1$)

Input: NP statement x

Samples challenge e $\in \mathcal{C}$

Output: 0/1

Adaptively Secure Sigma protocol Σ $\longrightarrow$ Adaptively Secure NIZK

Prover P(x, w)

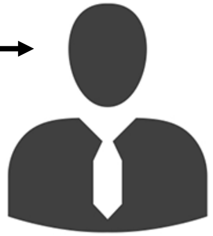
Setup: CI-Hash h^{S} for R_f

Verifier V(x)

$\longleftrightarrow \mathcal{F}^{S}_{\text{NICOM}}$

$$\xrightarrow{\alpha, \gamma}$$

$\mathcal{F}^{S}_{\text{NICOM}} \longleftrightarrow$



Input: NP statement x,
witness w

Output: Proof π = (α, γ)

Compute two transcripts (a, c_0, z_0), (a, c_1, z_1) for the same first message for prover chosen challenges $c_0 \neq c_1 \in \mathcal{C}$:

$(a, c_0, c_1, z_0, z_1) = \Sigma.P(x, w; r)$

Commit to challenge as $(C, \delta^{c}) = \mathcal{F}^{S}_{\text{NICOM}}(c_0, c_1)$

Commit to responses as $(Z_0, \delta_0) = \mathcal{F}^{S}_{\text{NICOM}}(z_0)$, $(Z_1, \delta_1) = \mathcal{F}^{S}_{\text{NICOM}}(z_1)$

Construct first message $\alpha = (a, C, Z_0, Z_1)$

Input: NP statement x

Samples challenge $e \in \mathcal{C}$

Output: 0/1

Adaptively Secure Sigma protocol $\Sigma \longrightarrow$ Adaptively Secure NIZK

Prover P(x, w)

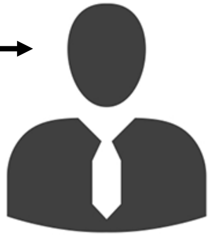
Setup: CI-Hash $h^{\mathcal{S}}$ for R_f

Verifier V(x)

$\longleftrightarrow \mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$

$\xrightarrow{\alpha,\, \gamma}$

$\mathcal{F}^{\mathcal{S}}_{\text{NICOM}} \longleftrightarrow$



f(α) = 0 iff V(x, a, c_0, z_0) = 1
where c_0, z_0 are extracted
from α using $\mathcal{S}$ algorithm

Input: NP statement x,
witness w

Output: Proof $\pi = (\alpha, \gamma)$

Compute two transcripts (a, c_0, z_0), (a, c_1, z_1) for the same first message for prover chosen challenges $c_0 \neq c_1 \in \mathcal{C}$:

(a, c_0, c_1, z_0, z_1) = Σ.P(x, w; r)

Commit to challenge as (C, δ^c) = $\mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$ (c_0, c_1)

Commit to responses as (Z_0, δ_0) = $\mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$($z_0$), ($Z_1$, δ_1) = $\mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$($z_1$)

Construct first message α = (a, C, Z_0, Z_1)

Construct challenge e = $h^{\mathcal{S}}(\alpha)$

Input: NP statement x

Samples challenge e $\in \mathcal{C}$

Output: 0/1

Adaptively Secure Sigma protocol $\Sigma \longrightarrow$ Adaptively Secure NIZK

Prover P(x, w)

Setup: CI-Hash $h^{\mathcal{S}}$ for R_f

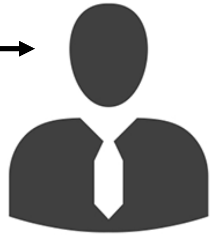
Verifier V(x)

$\longleftrightarrow \mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$

α, γ

$\longrightarrow$

$\mathcal{F}^{\mathcal{S}}_{\text{NICOM}} \longleftrightarrow$



Input: NP statement x, witness w

f(α) = 0 iff V(x, a, c_0, z_0) = 1
where c_0, z_0 are extracted
from α using $\mathcal{S}$ algorithm

Input: NP statement x

Samples challenge e $\in \mathcal{C}$

Output: 0/1

Output: Proof π = (α, γ)

Compute two transcripts (a, c_0, z_0), (a, c_1, z_1) for the same first message for prover chosen challenges $c_0 \neq c_1 \in \mathcal{C}$:

(a, c_0, c_1, z_0, z_1) = Σ.P(x, w; r)

Commit to challenge as (C, δ^c) = $\mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$ (c_0, c_1)

Commit to responses as (Z_0 , δ_0) = $\mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$($z_0$), ($Z_1$, δ_1) = $\mathcal{F}^{\mathcal{S}}_{\text{NICOM}}$($z_1$)

Construct first message α = (a, C, Z_0, Z_1)

Construct challenge e = $h^{\mathcal{S}}(\alpha)$

Construct response γ = (c_0, c_1, δ^c, z_e, δ_e)

Adaptively Secure Sigma protocol $\Sigma \longrightarrow$ Adaptively Secure NIZK

Prover P(x, w)

Setup: CI-Hash h^{S} for R_{f}

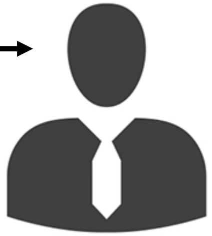
Verifier V(x)

$\longleftrightarrow \mathcal{F}^{S}_{\text{NICOM}}$

α, γ

$\longrightarrow$

$\mathcal{F}^{S}_{\text{NICOM}} \longleftrightarrow$



Input: NP statement x, witness w

f(α) = 0 iff V(x, a, c_0, z_0) = 1 where c_0, z_0 are extracted from α using $\mathcal{S}$ algorithm

Input: NP statement x

Samples challenge e ∈ $\mathcal{C}$

Output: 0/1

Output: Proof π = (α, γ)

Compute two transcripts (a, c_0, z_0), (a, c_1, z_1) for the same first message for prover chosen challenges $c_0 \neq c_1 \in \mathcal{C}$:

(a, c_0, c_1, z_0, z_1) = Σ.P(x, w; r)

Commit to challenge as (C, δ^c) = $\mathcal{F}^{S}_{\text{NICOM}}$ (c_0, c_1)

Commit to responses as (Z_0 , δ_0) = $\mathcal{F}^{S}_{\text{NICOM}}$($z_0$), ($Z_1$, δ_1) = $\mathcal{F}^{S}_{\text{NICOM}}$($z_1$)

Construct first message α = (a, C, Z_0, Z_1)

Construct challenge e = h^{S}(α)

Construct response γ = (c_0, c_1, δ^c, z_e, δ_e)

Compute e = H(α)

Verify Decommitments to c_0, c_1, z_e in γ

Verify $c_0 \neq c_1$

Adaptively Secure Sigma protocol Σ ⟶ Adaptively Secure NIZK

Setup: CI-Hash h^S for R_f

Prover P(x, w)

$\leftrightarrow \mathcal{F}^S_{\text{NICOM}}$

Input: NP statement x, witness w

Output: Proof $\pi = (\alpha, \gamma)$

Compute two transcripts (a, c_0, z_0), (a, c_1, z_1) for the same first message for prover chosen challenges $c_0 \neq c_1 \in \mathcal{C}$:

$(a, c_0, c_1, z_0, z_1) = \Sigma.P(x, w; r)$

Commit to challenge as $(C, \delta^c) = \mathcal{F}^S_{\text{NICOM}}(c_0, c_1)$

Commit to responses as $(Z_0, \delta_0) = \mathcal{F}^S_{\text{NICOM}}(z_0)$, $(Z_1, \delta_1) = \mathcal{F}^S_{\text{NICOM}}(z_1)$

Construct first message $\alpha = (a, C, Z_0, Z_1)$

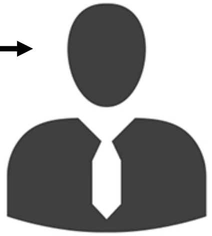
Construct challenge $e = h^S(\alpha)$

Construct response $\gamma = (c_0, c_1, \delta^c, z_e, \delta_e)$

α, γ ⟶

$f(\alpha) = 0$ iff $V(x, a, c_0, z_0) = 1$
where c_0, z_0 are extracted
from α using $\mathcal{S}$ algorithm

Verifier V(x)



$\mathcal{F}^S_{\text{NICOM}} \leftrightarrow$

Input: NP statement x

Samples challenge $e \in \mathcal{C}$

Output: 0/1

Compute $e = H(\alpha)$

Verify Decommitments to c_0, c_1, z_e in γ

Verify $c_0 \neq c_1$

Output $\Sigma.V(x, a, c_e, z_e)$

Adaptively Secure Sigma protocol $\Sigma \longrightarrow$ Adaptively Secure NIZK

Prover P(x, w)

$\longleftrightarrow \mathcal{F}^{S}_{\mathrm{NICOM}}$

Input: NP statement x, witness w

Output: Proof $\pi = (\alpha, \gamma)$

Compute two transcripts (a, c_0, z_0), (a, c_1, z_1) for the same first message for prover chosen challenges $c_0 \neq c_1 \in \mathcal{C}$:

$(a, c_0, c_1, z_0, z_1) = \Sigma.\mathrm{P}(x, w; r)$

Commit to challenge as $(C, \delta^c) = \mathcal{F}^{S}_{\mathrm{NICOM}}(c_0, c_1)$

Commit to responses as $(Z_0, \delta_0) = \mathcal{F}^{S}_{\mathrm{NICOM}}(z_0)$, $(Z_1, \delta_1) = \mathcal{F}^{S}_{\mathrm{NICOM}}(z_1)$

Construct first message $\alpha = (a, C, Z_0, Z_1)$

Construct challenge $e = h^{S}(\alpha)$

Construct response $\gamma = (c_0, c_1, \delta^c, z_e, \delta_e)$

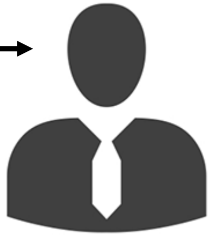
Setup: CI-Hash h^{S} for R_{f}

α, γ $\longrightarrow$

$f(\alpha) = 0$ iff $V(x, a, c_0, z_0) = 1$
where c_0, z_0 are extracted
from α using S algorithm

Verifier V(x)

$\mathcal{F}^{S}_{\mathrm{NICOM}} \longleftrightarrow$



Input: NP statement x

Samples challenge $e \in \mathcal{C}$

Output: 0/1

Compute $e = H(\alpha)$

Verify Decommitments to c_0, c_1, z_e in γ

Verify $c_0 \neq c_1$

Output $\Sigma.\mathrm{V}(x, a, c_e, z_e)$

Adaptive Security and adaptive ZK of NIZK follows from Adaptive Security of Sigma protocol in $\mathcal{F}^{S}_{\mathrm{NICOM}}$ - model

Adaptively Secure Sigma protocol Σ ➡ Adaptively Secure NIZK

Prover P(x, w)

$\leftrightarrow \mathcal{F}^{S}_{\text{NICOM}}$

Input: NP statement x, witness w

Output: Proof π = (α, γ)

Compute two transcripts (a, c_0, z_0), (a, c_1, z_1) for the same first message for prover chosen challenges $c_0 \neq c_1 \in \mathcal{C}$:

(a, c_0, c_1, z_0, z_1) = Σ.P(x, w; r)

Commit to challenge as (C, δ^c) = $\mathcal{F}^{S}_{\text{NICOM}}$ (c_0, c_1)

Commit to responses as (Z_0 , δ_0) = $\mathcal{F}^{S}_{\text{NICOM}}$($z_0$), ($Z_1$, δ_1) = $\mathcal{F}^{S}_{\text{NICOM}}$($z_1$)

Construct first message α = (a, C, Z_0, Z_1)

Construct challenge e = $h^{S}(\alpha)$

Construct response γ = (c_0, c_1, δ^c, z_e, δ_e)

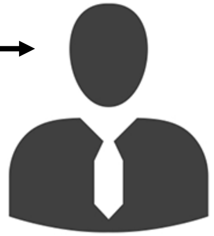
Setup: CI-Hash h^{S} for R_f

$\xrightarrow{\alpha, \gamma}$

f(α) = 0 iff V(x, a, c_0, z_0) = 1
where c_0, z_0 are extracted
from α using S algorithm

Verifier V(x)

$\mathcal{F}^{S}_{\text{NICOM}} \leftrightarrow$



Input: NP statement x

Samples challenge e $\in \mathcal{C}$

Output: 0/1

Compute e = H(α)

Verify Decommitments to c_0, c_1, z_e in γ

Verify $c_0 \neq c_1$

Output Σ.V(x, a, c_e, z_e)

Adaptive Security and Adaptive ZK

Soundness relies on
Special soundness of Sigma protocol in
$\mathcal{F}^{S}_{\text{NICOM}}$ - model + CI for R_f

Our Contributions

Non-interactive UC-Commitment Functionality $\mathcal{F}_{\text{NICOM}}$	Triply Adaptive NIZK	Instantiations
Parties access $\mathcal{F}_{\text{NICOM}}$ locally for Commitment generation and verification	**Triply adaptive** Sigma protocol in $\mathcal{F}_{\text{NICOM}}$ model	Most Sigma protocols are **Triply adaptive** in $\mathcal{F}_{\text{NICOM}}$ model
Functionality outputs commitment string during Commit phase	Compile the above Sigma protocol to obtain Triply adaptive NIZK	Implement $\mathcal{F}_{\text{NICOM}}$ with [CanFis01] commitment scheme
Protocol Friendly	Apply Correlation Intractability for NIZK arguments	

UC-Security: Obtain UC-security using standard tricks [GosOstSah12]

Chapter VI:

Instantiations

Implementing Adaptively Secure Sigma Protocols in $\mathcal{F}_{\text{NICOM}}$ model

Schnorr type Protocols

Garbled circuit-based protocol of [HazVen16] (Avoids expensive Karp reductions)

Protocols for Graph Hamiltonicity by [FeiLapSha99] and [Blum86]

Implementing $\mathcal{F}_{\text{NICOM}}$ model

Implemented using [CanFis01] commitment

Based on equivocal commitments+ CCA-2 public key encryption with oblivious ciphertext sampling

Can be instantiated from LWE/ DDH

Implementing $\mathcal{F}_{\text{NICOM}}$ model

Implemented using [CanFis01] commitment

Based on equivocal commitments+ CCA-2 public key encryption with oblivious ciphertext sampling

Can be instantiated from LWE/ DDH

Note: For adaptive soundness we need the crs distribution of real and ideal world to be identical/statistically close for the commitment

Summary

Non-interactive UC-Commitment Functionality $\mathcal{F}_{\text{NICOM}}$	Triply Adaptive UC-NIZK	Instantiations
Proposed a new UC commitment functionality which is Protocol Friendly	Proposed the definition and provided a generic UC-NIZK compiler with triple adaptivity	Instantiated $\mathcal{F}_{\text{NICOM}}$ from [CF01] Instantiated NIZK compiler based on LWE/DDH+LPN by instantiating the CI hash

Thank you

2020/1212

pratik93@bu.edu