A New Generic Transform from Multi-Round Interactive Proof to NIZK

Pierre-Alain Fouque, Adela Georgescu, Chen Qian, Adeline Roux-Langlois, Weiqiang Wen

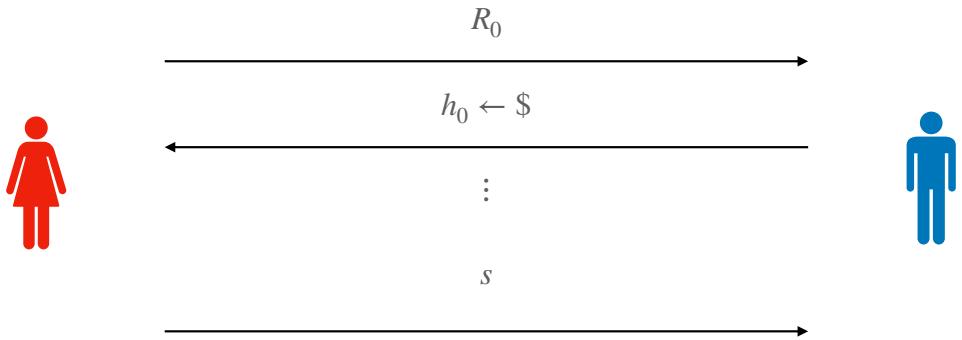
Content

- Introduction
  - Muti-round PCIP and NIZK
  - Existing transforms: Fiat-Shamir and CPSV@TCC16
- CPSV@TCC16 transform and OR-proofs
- Our new OR-proof technique and new transform
- Conclusion

Content

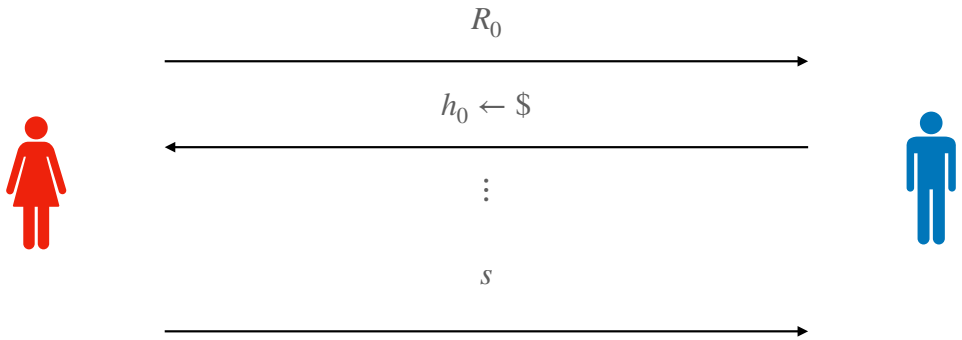
- Introduction
  - Muti-round PCIP and NIZK
  - Existing transforms: Fiat-Shamir CPSV@TCC16
- CPSV@TCC16 transform and OR-proofs
- Our new OR-proof technique and new transform
- Conclusion

Multi-Round (Public-Coin) Interactive Proof



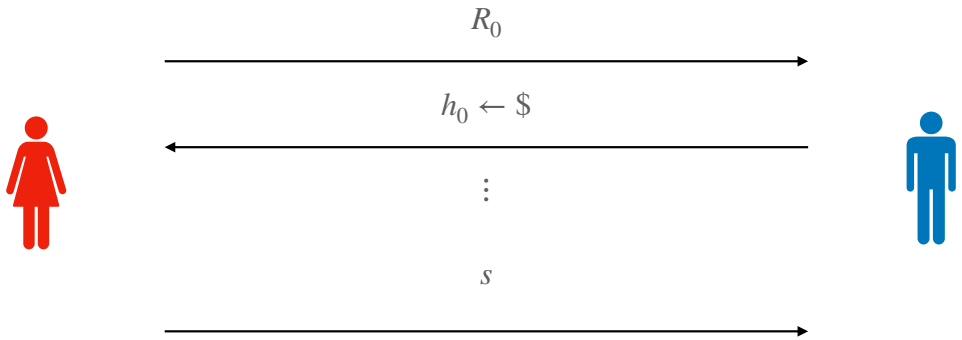
Multi-Round (Public-Coin) Interactive Proof

- Goal: Prove $x \in_w \mathscr{L}$ without leak w.



Multi-Round (Public-Coin) Interactive Proof

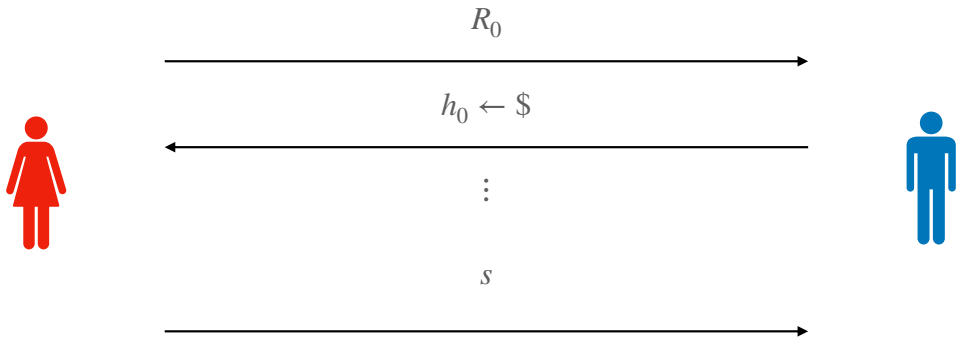
- Goal: Prove $x \in_w \mathscr{L}$ without leak w.



- Correctness: Honestly generated proof should be verifiable.

Multi-Round (Public-Coin) Interactive Proof

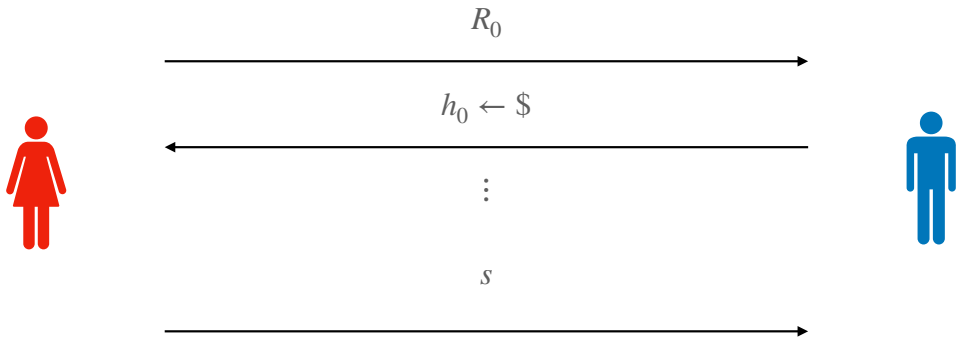
- Goal: Prove $x \in_w \mathscr{L}$ without leak w.



- Correctness: Honestly generated proof should be verifiable.
- Soundness: If $x \notin \mathscr{L}$, it is computationally hard to produce a valid proof.

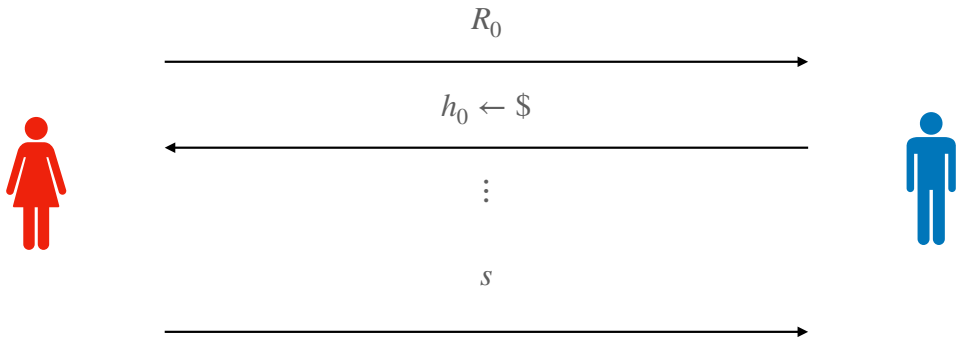
Multi-Round (Public-Coin) Interactive Proof

- Goal: Prove $x \in_w \mathscr{L}$ without leak w.



- Correctness: Honestly generated proof should be verifiable.
- Soundness: If $x \notin \mathscr{L}$, it is computationally hard to produce a valid proof.
- (Honest Verifier) Zero-knowledge: The transcript does not leak information more than $x \in \mathscr{L}$.

Multi-Round (Public-Coin) Interactive Proof



- Soundness: If $x \notin \mathscr{L}$, it is computationally hard to produce a valid proof.
  - Round-By-Round: Implied by negligible soundness.
  - Optimal: Generalization of special soundness.

Our contribution: Transform from PCIP to NIZK

Our contribution: Transform from PCIP to NIZK

	Multi-round?	Soudness	Zero-Knowledge
Fiat-Shamir	Yes	Random Oracle	Random Oracle
CPSV@TCC16	No	Non-programmable RO	standard model
This work	Yes	Non-programmable RO	standard model

Our contribution: Transform from PCIP to NIZK

	Multi-round?	Soudness	Zero-Knowledge
Fiat-Shamir	Yes	Random Oracle	Random Oracle
CPSV@TCC16	No	Non-programmable RO	standard model
This work	Yes	Non-programmable RO	standard model

- RO vs NPRO: NPRO is a weaker model -> theoretical interest

Our contribution: Transform from PCIP to NIZK

	Multi-round?	Soudness	Zero-Knowledge
Fiat-Shamir	Yes	Random Oracle	Random Oracle
CPSV@TCC16	No	Non-programmable RO	standard model
This work	Yes	Non-programmable RO	standard model

- RO vs NPRO: NPRO is a weaker model -> theoretical interest
- (Zero-knowledge) RO vs standard model: Long term privacy for high level protocol, like privacy in e-voting…

Our contribution: Transform from PCIP to NIZK

	Multi-round?	Soudness	Zero-Knowledge
Fiat-Shamir	Yes	Random Oracle	Random Oracle
CPSV@TCC16	No	Non-programmable RO	standard model
This work	Yes	Non-programmable RO	standard model

- RO vs NPRO: NPRO is a weaker model -> theoretical interest
- (Zero-knowledge) RO vs standard model: Long term privacy for high level protocol, like privacy in e-voting…
- multi-round: bulletproof, new lattice-based exact proofs…

Content

- Introduction
  - Muti-round PCIP and NIZK
  - Existing transforms: Fiat-Shamir CPSV@TCC16
- CPSV@TCC16 transform and OR-proofs
- Our new OR-proof technique and new transform
- Conclusion

Starting point: CPSV@TCC16

Starting point: CPSV@TCC16

- Goal: transform from Σ-protocol to NIZK with proof language $x \in_w \mathscr{L}$

Starting point: CPSV@TCC16

- Goal: transform from Σ-protocol to NIZK with proof language $x \in_w \mathscr{L}$
- High level construction:

Starting point: CPSV@TCC16

- Goal: transform from Σ-protocol to NIZK with proof language $x \in_w \mathscr{L}$
- High level construction:
  - Use another NP hard membership problem $\mathscr{L}'$

Starting point: CPSV@TCC16

- Goal: transform from Σ-protocol to NIZK with proof language $x \in_w \mathscr{L}$
- High level construction:
  - Use another NP hard membership problem $\mathscr{L}'$
  - The common reference string of NIZK is $x' \notin \mathscr{L}'$

Starting point: CPSV@TCC16

- Goal: transform from Σ-protocol to NIZK with proof language $x \in_w \mathscr{L}$
- High level construction:
  - Use another NP hard membership problem $\mathscr{L}'$
  - The common reference string of NIZK is $x' \notin \mathscr{L}'$
  - A NIZK proof is of the form $x \in \mathscr{L} \vee x' \in \mathscr{L}'$

Starting point: CPSV@TCC16

- Goal: transform from Σ-protocol to NIZK with proof language $x \in_w \mathscr{L}$
- High level construction:
  - Use another NP hard membership problem $\mathscr{L}'$
  - The common reference string of NIZK is $x' \notin \mathscr{L}'$
  - A NIZK proof is of the form $x \in \mathscr{L} \vee x' \in \mathscr{L}'$
- Proof intuition:

Starting point: CPSV@TCC16

- Goal: transform from Σ-protocol to NIZK with proof language $x \in_w \mathscr{L}$
- High level construction:
  - Use another NP hard membership problem $\mathscr{L}'$
  - The common reference string of NIZK is $x' \notin \mathscr{L}'$
  - A NIZK proof is of the form $x \in \mathscr{L} \vee x' \in \mathscr{L}'$
- Proof intuition:
  - Soundness: $x' \notin \mathscr{L}'$

Starting point: CPSV@TCC16

- Goal: transform from Σ-protocol to NIZK with proof language $x \in_w \mathscr{L}$
- High level construction:
  - Use another NP hard membership problem $\mathscr{L}'$
  - The common reference string of NIZK is $x' \notin \mathscr{L}'$
  - A NIZK proof is of the form $x \in \mathscr{L} \vee x' \in \mathscr{L}'$
- Proof intuition:
  - Soundness: $x' \notin \mathscr{L}'$
  - Zero-knowledge: switch CRS with $x' \in_{w'} \mathscr{L}'$, then the simulator prove with w'

Main difficulty for Multi-Round PCIP

Main difficulty for Multi-Round PCIP

- The core part of CPSV@TCC16 is an OR-proof of Σ-protocol

Main difficulty for Multi-Round PCIP

- The core part of CPSV@TCC16 is an OR-proof of Σ-protocol
- Existing OR-proof technique cannot be used on multi-round PCIP

Main difficulty for Multi-Round PCIP

- The core part of CPSV@TCC16 is an OR-proof of Σ-protocol
- Existing OR-proof technique cannot be used on multi-round PCIP
  - Parallel OR-proof: CDS@Crypto94

Main difficulty for Multi-Round PCIP

- The core part of CPSV@TCC16 is an OR-proof of Σ-protocol
- Existing OR-proof technique cannot be used on multi-round PCIP
  - Parallel OR-proof: CDS@Crypto94
  - Sequential OR-proof: AOS@Asiacrypt02

Parallel OR-Proof

CDS@Crypto94

- Proof of the form: $\pi = (R_0, R_1, h_0, h_1, s_0, s_1)$
- $h_0 \oplus h_1 = H(R_0, R_1)$: The adversary can freely choose h_0 or h_1.

Counter Example

Parallel OR-proof does not work for multi-round

Counter Example

Parallel OR-proof does not work for multi-round

- CDS@Crypto94 does not work for multi-round PCIP

Counter Example

Parallel OR-proof does not work for multi-round

- CDS@Crypto94 does not work for multi-round PCIP

- Remarks:

Counter Example

Parallel OR-proof does not work for multi-round

- CDS@Crypto94 does not work for multi-round PCIP

- Remarks:
  - A natural extension of CDS@Crypto94 leads to $h_0 \oplus \hat{h}_1 = H(R_0, \hat{R}_1)$ and $\hat{h}_0 \oplus h_1 = H(R_0, \hat{R}_1, \hat{R}_0, R_1)$.

Counter Example

Parallel OR-proof does not work for multi-round

- CDS@Crypto94 does not work for multi-round PCIP

- Remarks:
  - A natural extension of CDS@Crypto94 leads to $h_0 \oplus \hat{h}_1 = H(R_0, \hat{R}_1)$ and $\hat{h}_0 \oplus h_1 = H(R_0, \hat{R}_1, \hat{R}_0, R_1)$.
  - The adversary can freely choose both h_0 and h_1.

Sequential OR-Proof

AOS@Asiacrypt02

- Proof of the form: $\pi = (R_0, R_1, h_0, h_1, s_0, s_1)$
- $h_0 = H(R_1)$ and $h_1 = H(R_0)$: The adversary can freely choose h_0 or h_1 depending on the order.

Content

- Introduction
  - Muti-round PCIP and NIZK
  - Existing transforms: Fiat-Shamir CPSV@TCC16
- CPSV@TCC16 transform and OR-proofs
- Our new OR-proof technique and new transform
- Conclusion

Our intuition:

New OR-proof

Our intuition:

New OR-proof

- Need to combine all the challenges of one side together!

Our intuition:

New OR-proof

- Need to combine all the challenges of one side together!
- Need to separate the challenges from each sides

First attempt:
Idea from parallel OR-proof

First attempt:

Idea from parallel OR-proof

- Combine all the challenges of the same side together by an offset fixed in the hash input.

First attempt:

Idea from parallel OR-proof

- Combine all the challenges of the same side together by an offset fixed in the hash input.
- Given an offset $\mathsf{A}_b = (a_{1,b}, \ldots, a_{n,b})$ and the first i rounds commitments $(R_{1,b}, \ldots, R_{i,b})$, the i-th round challenge is fixed.

First attempt:

Idea from parallel OR-proof

- Combine all the challenges of the same side together by an offset fixed in the hash input.
- Given an offset $\mathsf{A}_b = (a_{1,b}, \ldots, a_{n,b})$ and the first i rounds commitments $(R_{1,b}, \ldots, R_{i,b})$, the i-th round challenge is fixed.
- First attempt:

First attempt:

Idea from parallel OR-proof

- Combine all the challenges of the same side together by an offset fixed in the hash input.
- Given an offset $\mathsf{A}_b = (a_{1,b}, \ldots, a_{n,b})$ and the first i rounds commitments $(R_{1,b}, \ldots, R_{i,b})$, the i-th round challenge is fixed.
- First attempt:
  - $h_{i,0} = H(\{R_{j,0}\}_{j=1}^{i}) \oplus a_{i,0}$, $h_{i,1} = H(\{R_{j,1}\}_{j=1}^{i}) \oplus a_{i,1}$

Second attempt

Idea from sequential OR-proof

Second attempt
Idea from sequential OR-proof

- Second attempt:

Second attempt

Idea from sequential OR-proof

- Second attempt:
  - $h_{i,0} = H(\{R_{j,0}\}_{j=1}^{i}, \mathsf{A}_1) \oplus a_{i,0}$, $h_{i,1} = H(\{R_{j,1}\}_{j=1}^{i}, \mathsf{A}_0) \oplus a_{i,1}$

Second attempt

Idea from sequential OR-proof

- Second attempt:
  - $h_{i,0} = H(\{R_{j,0}\}_{j=1}^{i}, \mathsf{A}_1) \oplus a_{i,0}$, $h_{i,1} = H(\{R_{j,1}\}_{j=1}^{i}, \mathsf{A}_0) \oplus a_{i,1}$
- How to generate a proof with only w_b:

Second attempt

Idea from sequential OR-proof

- Second attempt:
  - $h_{i,0} = H(\{R_{j,0}\}_{j=1}^{i}, \mathsf{A}_1) \oplus a_{i,0}$, $h_{i,1} = H(\{R_{j,1}\}_{j=1}^{i}, \mathsf{A}_0) \oplus a_{i,1}$
- How to generate a proof with only w_b:
  - Generate a random polynomial A_b.

Second attempt

Idea from sequential OR-proof

- Second attempt:
  - $h_{i,0} = H(\{R_{j,0}\}_{j=1}^{i}, \mathsf{A}_1) \oplus a_{i,0}$, $h_{i,1} = H(\{R_{j,1}\}_{j=1}^{i}, \mathsf{A}_0) \oplus a_{i,1}$
- How to generate a proof with only w_b:
  - Generate a random polynomial A_b.
  - Using HVZK to simulate $\pi_{1-b} = (R_{1,1-b}, h_{1,1-b}, \ldots, R_{n,1-b}, h_{n,1-b}, s_{1-b})$

Second attempt

Idea from sequential OR-proof

- Second attempt:
  - $h_{i,0} = H(\{R_{j,0}\}_{j=1}^{i}, \mathsf{A}_1) \oplus a_{i,0}$, $h_{i,1} = H(\{R_{j,1}\}_{j=1}^{i}, \mathsf{A}_0) \oplus a_{i,1}$
- How to generate a proof with only w_b:
  - Generate a random polynomial A_b.
  - Using HVZK to simulate $\pi_{1-b} = (R_{1,1-b}, h_{1,1-b}, \ldots, R_{n,1-b}, h_{n,1-b}, s_{1-b})$
  - Compute $a_{i,1-b} = H(\{R_{j,1-b}\}_{j=1}^{i}, \mathsf{A}_b) \oplus h_{i,1-b}$ for all $\mathsf{i} \in [\mathsf{n}]$

Second attempt

Idea from sequential OR-proof

- Second attempt:
  - $h_{i,0} = H(\{R_{j,0}\}_{j=1}^{i}, \mathsf{A}_1) \oplus a_{i,0}$, $h_{i,1} = H(\{R_{j,1}\}_{j=1}^{i}, \mathsf{A}_0) \oplus a_{i,1}$
- How to generate a proof with only w_b:
  - Generate a random polynomial A_b.
  - Using HVZK to simulate $\pi_{1-b} = (R_{1,1-b}, h_{1,1-b}, \ldots, R_{n,1-b}, h_{n,1-b}, s_{1-b})$
  - Compute $a_{i,1-b} = H(\{R_{j,1-b}\}_{j=1}^{i}, \mathsf{A}_b) \oplus h_{i,1-b}$ for all $\mathsf{i} \in [\mathsf{n}]$
  - Let $\mathsf{A}_{1-b} = (a_{1,1-b}, \ldots, a_{n,1-b})$

Second attempt

Idea from sequential OR-proof

- Second attempt:
  - $h_{i,0} = H(\{R_{j,0}\}_{j=1}^{i}, \mathsf{A}_1) \oplus a_{i,0}$, $h_{i,1} = H(\{R_{j,1}\}_{j=1}^{i}, \mathsf{A}_0) \oplus a_{i,1}$
- How to generate a proof with only w_b:
  - Generate a random polynomial A_b.
  - Using HVZK to simulate $\pi_{1-b} = (R_{1,1-b}, h_{1,1-b}, \ldots, R_{n,1-b}, h_{n,1-b}, s_{1-b})$
  - Compute $a_{i,1-b} = H(\{R_{j,1-b}\}_{j=1}^{i}, \mathsf{A}_b) \oplus h_{i,1-b}$ for all i ∈ [n]
  - Let $\mathsf{A}_{1-b} = (a_{1,1-b}, \ldots, a_{n,1-b})$
  - Compute π_b using A_b and A_{1-b}

Security and QROM

Security and QROM

- Classical Setting:

Security and QROM

- Classical Setting:
  - Our transform in NPROM is suitable for optimal soundness (->special soundness) and round-by-round soundness (-> negligible soundness).

Security and QROM

- Classical Setting:
  - Our transform in NPROM is suitable for optimal soundness (->special soundness) and round-by-round soundness (-> negligible soundness).
- Quantum setting:

Security and QROM

- Classical Setting:
  - Our transform in NPROM is suitable for optimal soundness (->special soundness) and round-by-round soundness (-> negligible soundness).
- Quantum setting:
  - QROM based proof with security loss Q_H^4 instead of Q_H^{2n}.

Content

- Introduction
  - Muti-round PCIP and NIZK
  - Existing transforms: Fiat-Shamir CPSV@TCC16
- CPSV@TCC16 transform and OR-proofs
- Our new OR-proof technique and new transform
- Conclusion

Conclusion

Our contribution and open problems

Conclusion

Our contribution and open problems

- Contribution:

Conclusion

Our contribution and open problems

- Contribution:
  - First OR-proof for multi-round PCIP protocols

Conclusion

Our contribution and open problems

- Contribution:
  - First OR-proof for multi-round PCIP protocols
  - First generic transform from multi-round PCIP to NIZK with soundness in NPROM and zero-knowledge in the standard model

Conclusion

Our contribution and open problems

- Contribution:
  - First OR-proof for multi-round PCIP protocols
  - First generic transform from multi-round PCIP to NIZK with soundness in NPROM and zero-knowledge in the standard model
  - First transform in QROM with security loss Q_H^4 instead of Q_H^{2n}. (See the paper for more details)

Conclusion

Our contribution and open problems

- Contribution:
  - First OR-proof for multi-round PCIP protocols
  - First generic transform from multi-round PCIP to NIZK with soundness in NPROM and zero-knowledge in the standard model
  - First transform in QROM with security loss Q_H^4 instead of Q_H^{2n}. (See the paper for more details)
- Open problem:

Conclusion

Our contribution and open problems

- Contribution:
  - First OR-proof for multi-round PCIP protocols
  - First generic transform from multi-round PCIP to NIZK with soundness in NPROM and zero-knowledge in the standard model
  - First transform in QROM with security loss Q_H^4 instead of Q_H^{2n}. (See the paper for more details)
- Open problem:
  - The proof in the QROM require programmability, does there exist a transform in QROM without programmability?