On the Existence of Three Round Zero-Knowledge Proofs

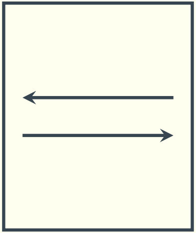
Nils Fleischhacker, Vipul Goyal, Abhishek Jain

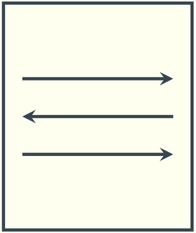
Tel Aviv, May 2, 2018

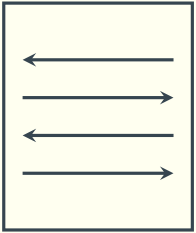
Johns Hopkins University

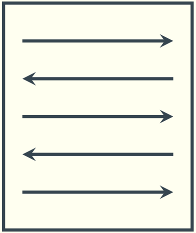
Carnegie Mellon University

Round-Complexity of ZK-Proofs for NP

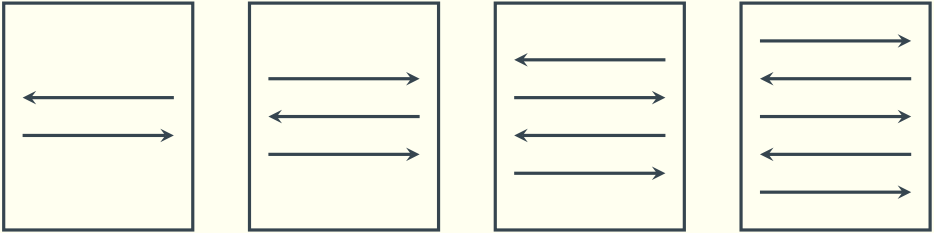




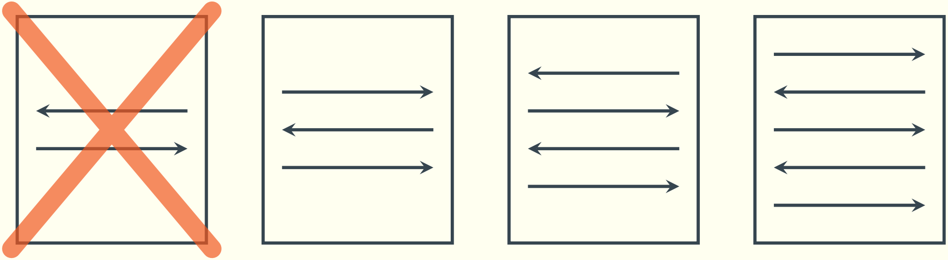




Round-Complexity of ZK-Proofs for NP

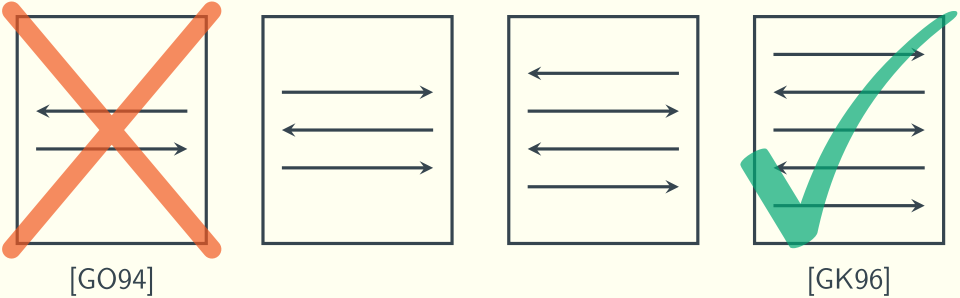


Round-Complexity of ZK-Proofs for NP

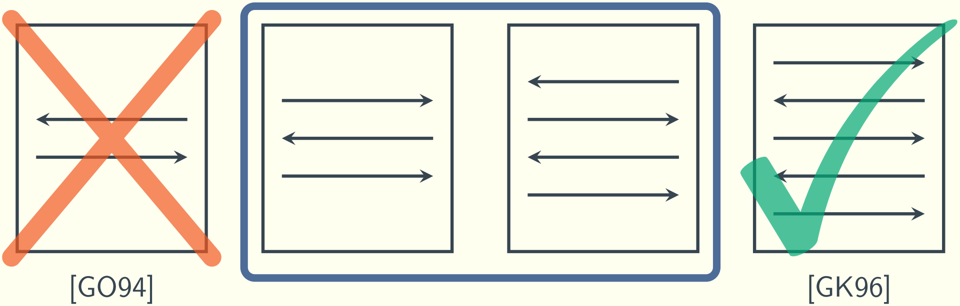


[GO94]

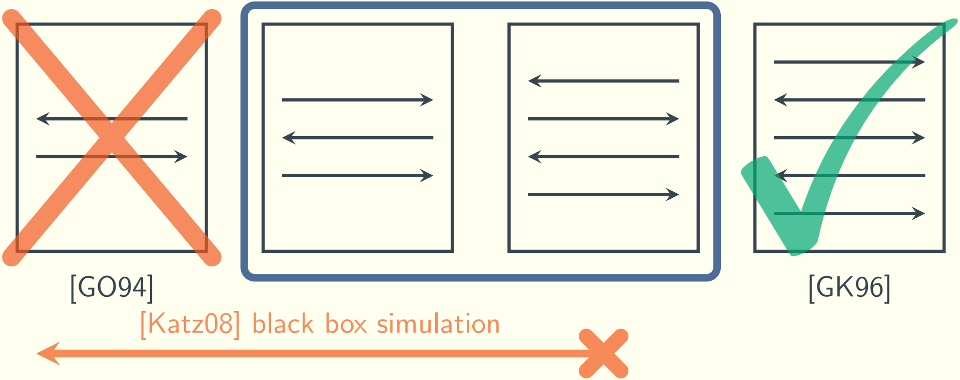
Round-Complexity of ZK-Proofs for NP



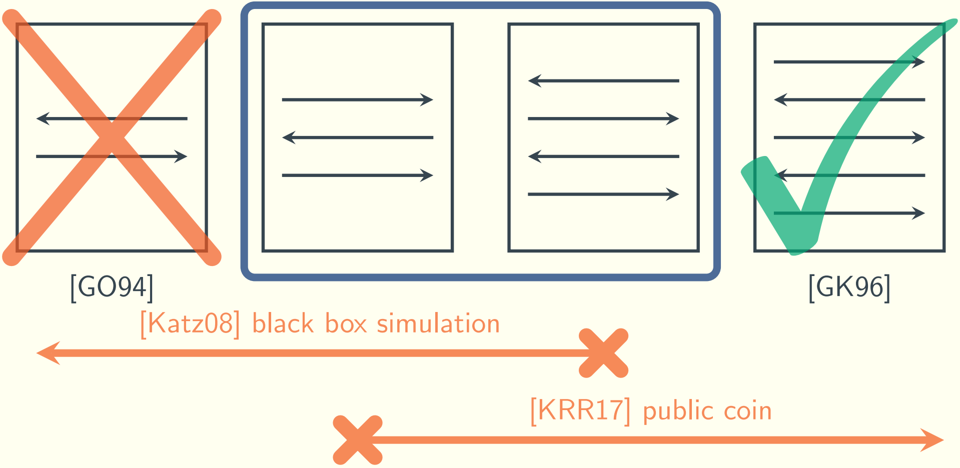
Round-Complexity of ZK-Proofs for NP



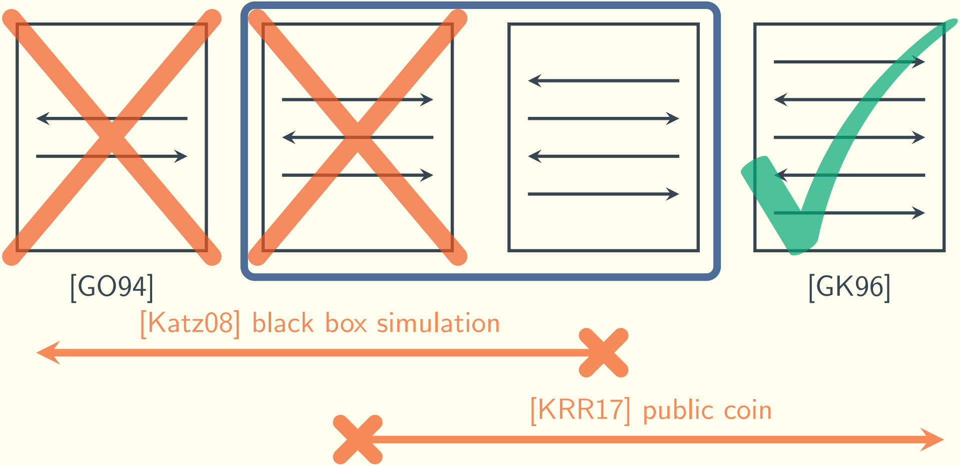
Round-Complexity of ZK-Proofs for NP



Round-Complexity of ZK-Proofs for NP



Round-Complexity of ZK-Proofs for NP



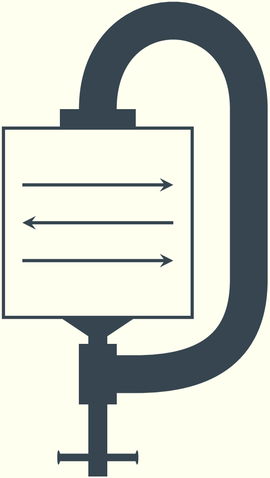
The Result

Assuming sub-exponentially secure iO and sub-exponentially secure PRFs as well as exponentially secure input-hiding obfuscation for multi-bit point functions, even private coin three round zero-knowledge proofs can only exist for languages in BPP.

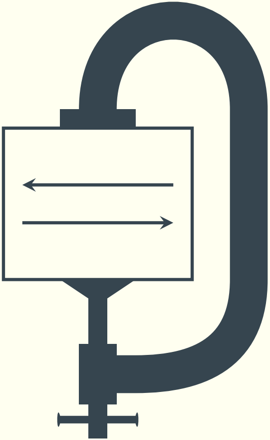
What About Four Rounds?

- We do not expect our technique to easily extend to four rounds.
- Our result extends to a weaker notion of ϵ-ZK.
- For ϵ-ZK, four round private coin protocols exist based on keyless multi-collision resistant hash functions (MCRH). [BKP17]

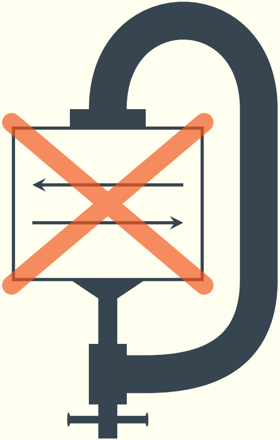
Compressing Proofs



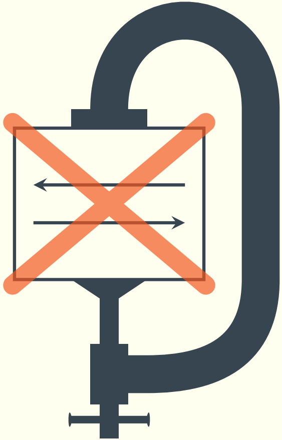
Compressing Proofs



Compressing Proofs

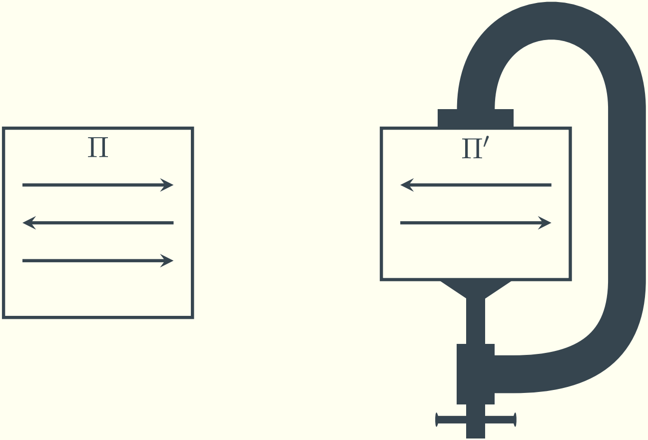


Compressing Proofs



Sadly, it's not that simple.

Proofs vs. Arguments



We lose statistical soundness. Π' is only an argument.

Π Sound $\implies$ Π' Sound $\implies$ Π not ZK

How to Compress Proofs

$\alpha \leftarrow \mathsf{P}_1(x, w)$

$\xrightarrow{\alpha}$

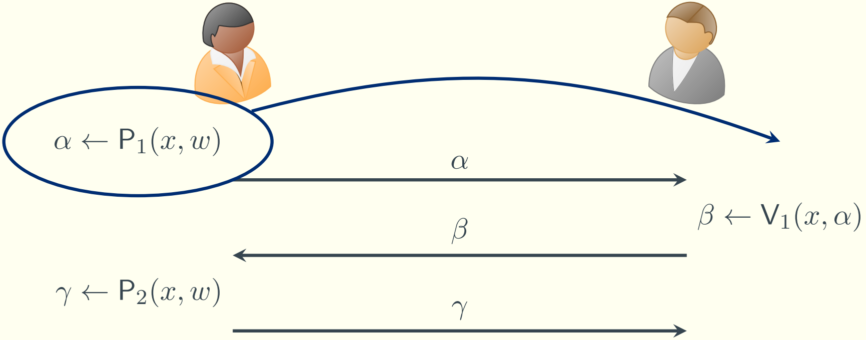
$\beta \leftarrow \mathsf{V}_1(x, \alpha)$

$\xleftarrow{\beta}$

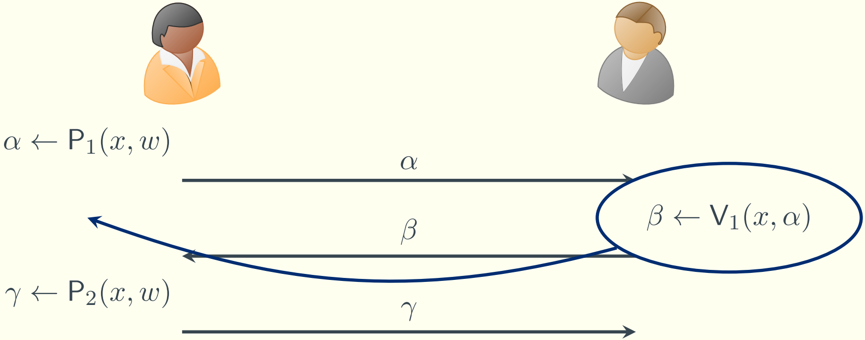
$\gamma \leftarrow \mathsf{P}_2(x, w)$

$\xrightarrow{\gamma}$

How to Compress Proofs



How to Compress Proofs



How to Compress Proofs

$\alpha \leftarrow \mathsf{P}_1(x, w)$

$\xrightarrow{\quad \alpha \quad}$

~~$\beta \leftarrow \mathsf{V}_1(x, \alpha)$~~

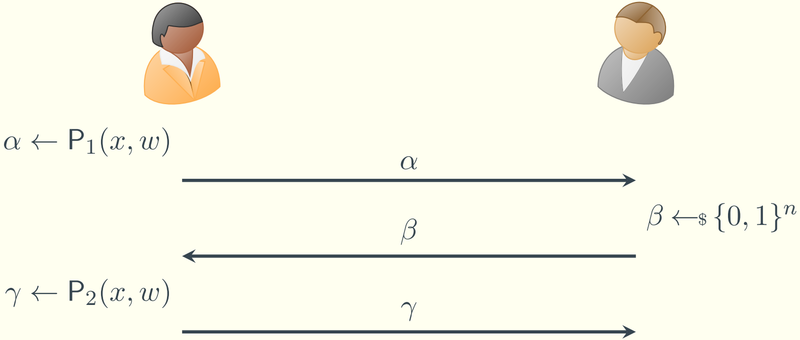
$\xleftarrow{\quad \beta \quad}$

$\beta \leftarrow_{\$} \{0,1\}^n$

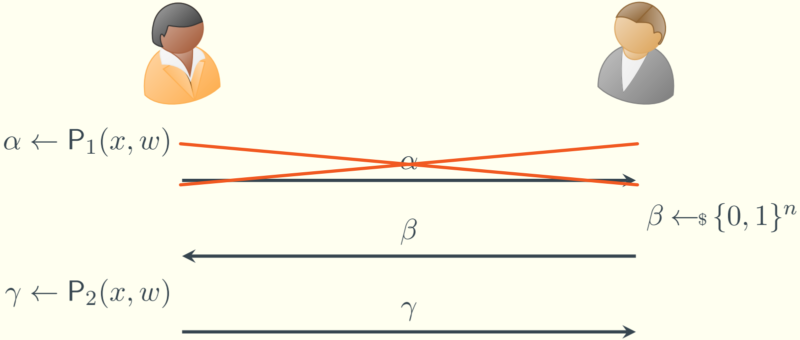
$\gamma \leftarrow \mathsf{P}_2(x, w)$

$\xrightarrow{\quad \gamma \quad}$

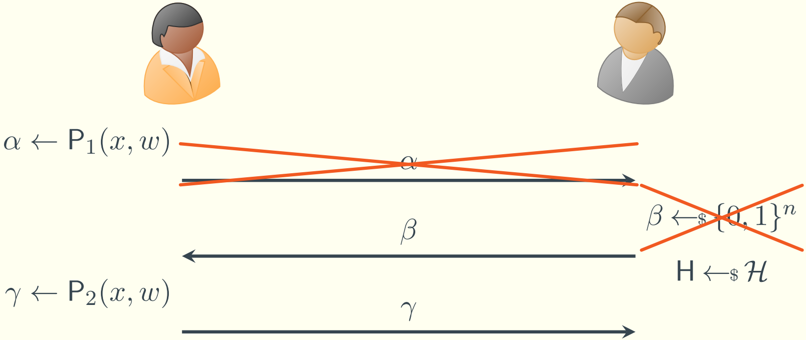
The Public Coin Case



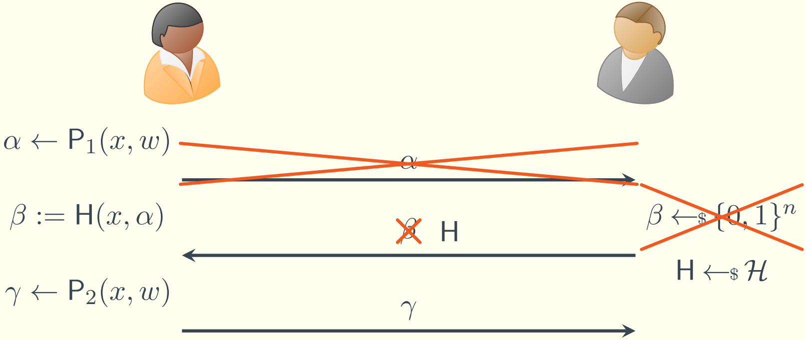
The Public Coin Case



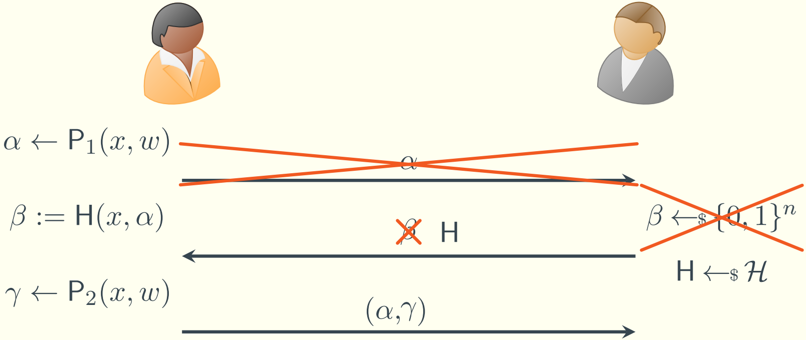
The Public Coin Case



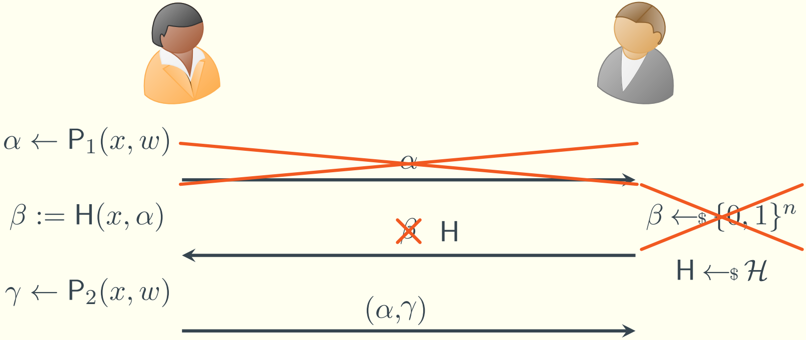
The Public Coin Case



The Public Coin Case



The Public Coin Case



[KRR17]: $\mathsf{H} := \mathsf{iO}(\mathsf{PRF}_k(\cdot))$

But What About Private Coin?

$\alpha \leftarrow \mathsf{P}_1(x, w)$

$\xrightarrow{\quad \alpha \quad}$

$\beta \leftarrow \mathsf{V}_1(x, \alpha)$

$\xleftarrow{\quad \beta \quad}$

$\gamma \leftarrow \mathsf{P}_2(x, w)$

$\xrightarrow{\quad \gamma \quad}$

But What About Private Coin?

$\mathsf{C_V}[k, x](\alpha)$

$s := \mathsf{PRF}_k(\alpha)$

$\beta := \mathsf{V}_1(x, \alpha; s)$

return β

$\alpha \leftarrow \mathsf{P}_1(x, w)$

α →

$\beta \leftarrow \mathsf{V}_1(x, \alpha)$

← β

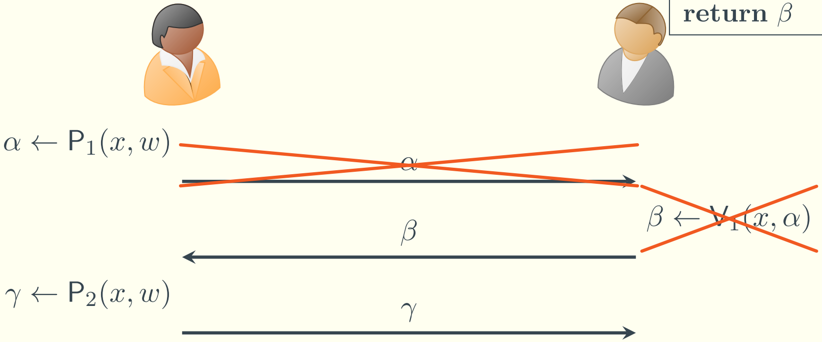
$\gamma \leftarrow \mathsf{P}_2(x, w)$

γ →

But What About Private Coin?

$\mathsf{C_V}[k, x](\alpha)$

$s := \mathsf{PRF}_k(\alpha)$
$\beta := \mathsf{V}_1(x, \alpha; s)$
return β



But What About Private Coin?

$\mathsf{C_V}[k, x](\alpha)$

$s := \mathsf{PRF}_k(\alpha)$

$\beta := \mathsf{V}_1(x, \alpha; s)$

return β

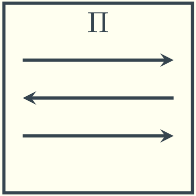
$\alpha \leftarrow \mathsf{P}_1(x, w)$ — ~~α~~ →

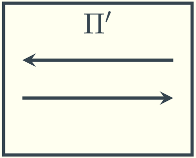
$\beta := \mathbf{B}(\alpha)$ ← ~~β~~ $\mathbf{B}$ — ~~$\beta \leftarrow \mathsf{V}_1(x, \alpha)$~~

$\mathbf{B} \leftarrow \mathsf{iO}(\mathsf{C_V}[k, x])$

$\gamma \leftarrow \mathsf{P}_2(x, w)$ — (α, γ) →

How to Prove it.

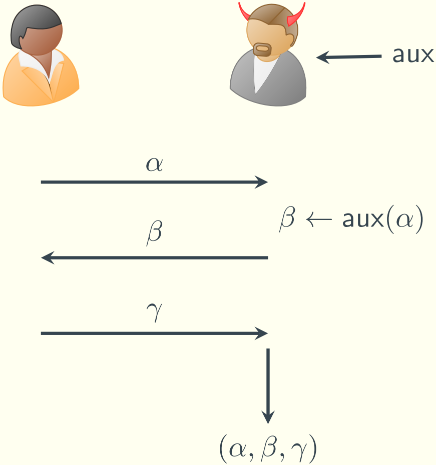




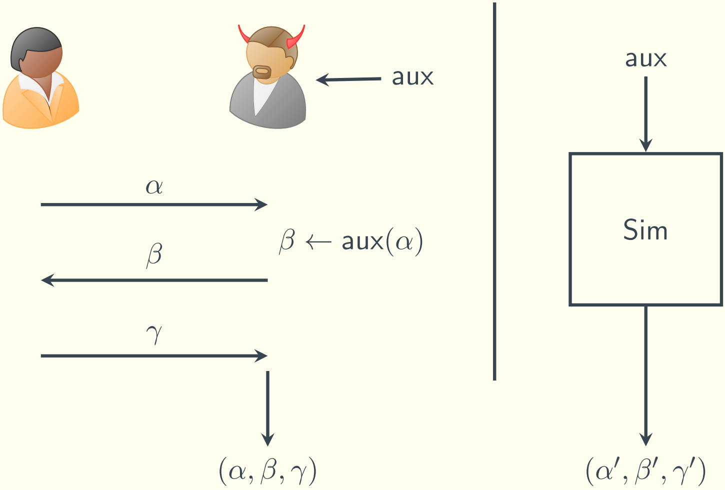
We need to prove two things:

1. If Π' is sound then Π is not zero knowledge.
2. The compression preserves soundness. I.e., if Π is sound then Π' is also sound.

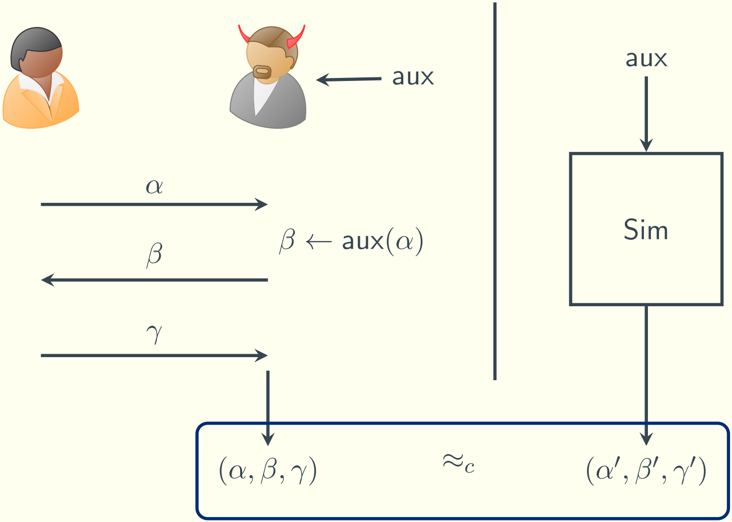
Π' sound $\implies$ Π' not ZK [GO94]



Π' sound $\implies$ Π' not ZK [GO94]



Π' sound $\implies$ Π' not ZK [GO94]



Π' sound $\implies$ Π' not ZK

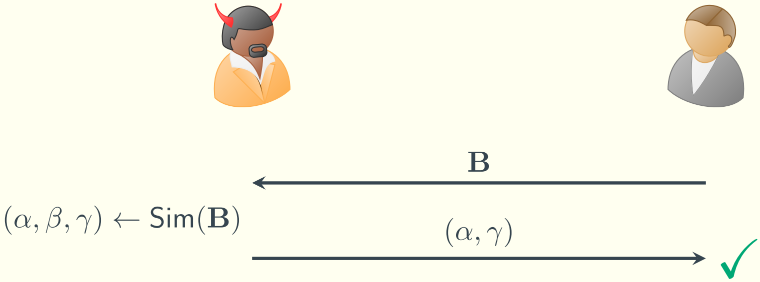
$\mathbf{B}$ ←

$(\alpha, \beta, \gamma) \leftarrow \mathsf{Sim}(\mathbf{B})$

(α, γ) → ✓

$(x^* \in \mathcal{L}) \approx_c (x^* \notin \mathcal{L})$ unless $\mathcal{L} \in \mathsf{BPP}$

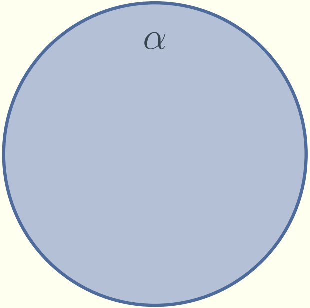
Π' **sound** $\implies$ Π' **not ZK**



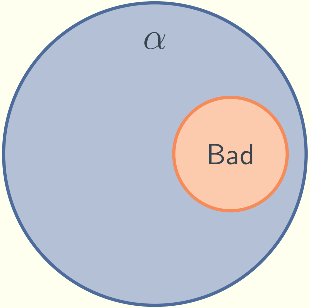
$(x^* \in \mathcal{L}) \approx_c (x^* \notin \mathcal{L})$ unless $\mathcal{L} \in \mathrm{BPP}$

But is it sound?

How Can a Prover Cheat? Defining Bad Alphas.

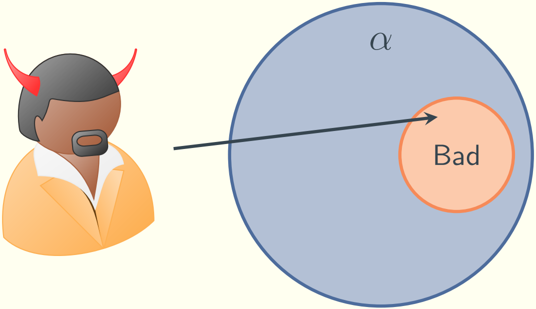


How Can a Prover Cheat? Defining Bad Alphas.



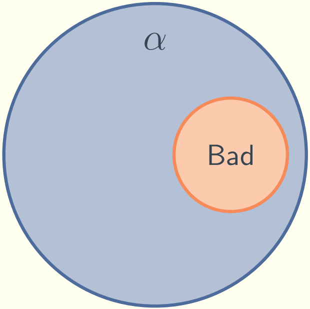
1. Specify a set of bad α's.

How Can a Prover Cheat? Defining Bad Alphas.



1. Specify a set of bad α's.
2. Prove that a cheating prover must use a bad α to cheat.

How Can a Prover Cheat? Defining Bad Alphas.

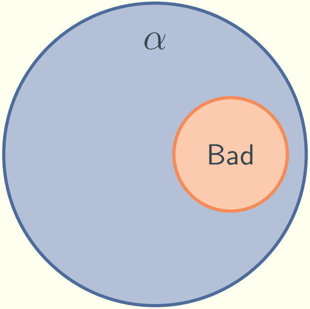


1. Specify a set of **bad** α's.

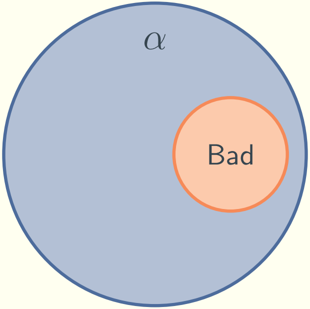
2. Prove that a cheating prover must use a bad α to cheat.

3. Prove that bad α's remain hidden by the obfuscation.

How Can a Prover Cheat? Defining Bad Alphas.

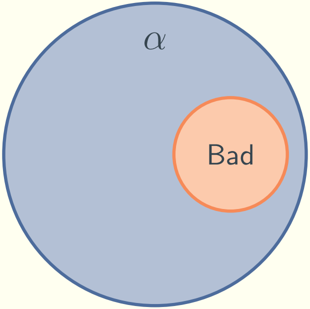


How Can a Prover Cheat? Defining Bad Alphas.



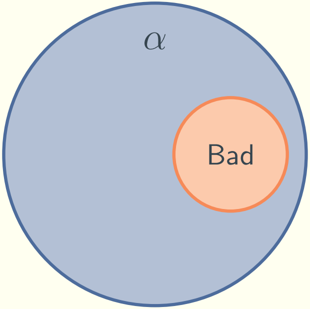
- In the public coin case, defining bad α's is trivial: Any α, such that for $\beta := \mathsf{PRF}_k(\alpha)$ there exists an accepting γ.

How Can a Prover Cheat? Defining Bad Alphas.



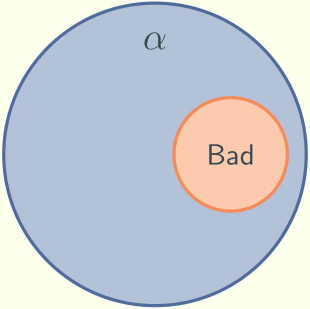
- In the public coin case, defining bad α's is trivial: Any α, such that for $\beta := \mathsf{PRF}_k(\alpha)$ there exists an accepting γ.
- In the private coin case, however there may always be accepting γ's.

How Can a Prover Cheat? Defining Bad Alphas.



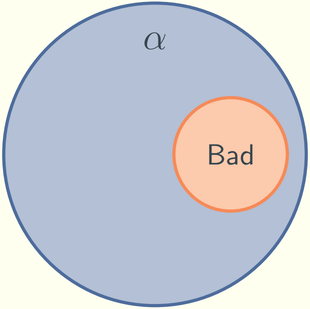
- In the public coin case, defining bad α's is trivial: Any α, such that for $\beta := \mathsf{PRF}_k(\alpha)$ there exists an accepting γ.
- In the private coin case, however there may always be accepting γ's.
- But, those γ's depend on which consistent random tape was used.

How Can a Prover Cheat? Defining Bad Alphas.



- In the public coin case, defining bad α's is trivial: Any α, such that for $\beta := \mathsf{PRF}_k(\alpha)$ there exists an accepting γ.
- In the private coin case, however there may always be accepting γ's.
- But, those γ's depend on which consistent random tape was used.
- Security of iO and puncturable PRF hide which random tape was used.

Bad Alphas in the Private Coin Case.



- An α is bad if the random tape $s := \mathsf{PRF}_k(\alpha)$ leads to a β such that for (α, β) there exists γ that will be accepted by the verifier with high probability over all consistent random tapes.

Hiding Bad Alphas.

- A cheating prover will output a bad α with high probability.

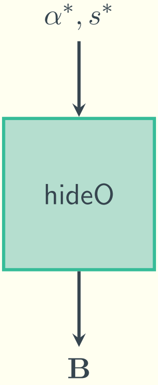
Hiding Bad Alphas.

- A cheating prover will output a bad α with high probability.
- This can be lead to a direct contradiction with the soundness of Π but incurs an exponential loss.

Hiding Bad Alphas.

- A cheating prover will output a bad α with high probability.
- This can be lead to a direct contradiction with the soundness of Π but incurs an exponential loss.
- We follow the approach of [KRR17] and “transfer” the loss to a seperate primitive.

Input Hiding Obfuscation of Multi-Bit Point Functions



Correctness: $\mathbf{B}(\alpha^*) = s^*$

$\forall \alpha \neq \alpha^* : \mathbf{B}(\alpha) = \perp$

Security: $\Pr[\mathcal{A}(\mathbf{B}, 1^n) = \alpha^*] \leq 2^{-n}$

Input Hiding Obfuscation of Multi-Bit Point Functions

Correctness: $\mathbf{B}(\alpha^*) = s^*$

$\forall \alpha \neq \alpha^* : \mathbf{B}(\alpha) = \bot$

Security: $\Pr[\mathcal{A}(\mathbf{B}, 1^n) = \alpha^*] \leq 2^{-n}$

Can be instantiated in the generic group model by [CD08] as shown in [BC10] based on a strong variant of DDH.

Transferring the Loss

Transferring the Loss

$\mathbf{C}_{\mathsf{pct}}[k, \alpha^*, \beta^*](\alpha)$

if $\alpha \stackrel{?}{=} \alpha^*$

$\quad \beta := \beta^*$

else

$\quad s := \mathsf{PRF}_k(\alpha)$

$\quad \beta := \mathsf{V}_1(x, \alpha; s)$

return β

Transferring the Loss

$\mathbf{C}_{\mathsf{pct}}[k, \alpha^*, \beta^*](\alpha)$

if $\alpha \stackrel{?}{=} \alpha^*$

$\quad \beta := \beta^*$

else

$\quad s := \mathsf{PRF}_k(\alpha)$

$\quad \beta := \mathsf{V}_1(x, \alpha; s)$

return β

Conditioned on α^* being bad we get that

$$\Pr_{k,\alpha^*,s^*,\mathsf{iO},\mathcal{A}}\left[\mathsf{P}^*\Big(\mathsf{iO}\big(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\}, \alpha^*, \mathsf{V}_1(x^*, \alpha; s^*)]\big)\Big) = (\alpha^*, \gamma)\right]$$

is slightly higher than random chance.

Transferring the Loss

$\mathbf{C}_{\mathsf{pct}}[k, \alpha^*, \beta^*](\alpha)$

if $\alpha \stackrel{?}{=} \alpha^*$
  $\beta := \beta^*$
else
  $s := \mathsf{PRF}_k(\alpha)$
  $\beta := \mathsf{V}_1(x, \alpha; s)$
return β

$\mathbf{C}_{\mathsf{hide}}[k, \mathbf{B}](\alpha)$

$s := \mathbf{B}(\alpha)$
if $s = \perp$
  $s := \mathsf{PRF}_k(\alpha)$
$\beta := \mathsf{V}_1(x^*, \alpha; s)$
return β

Conditioned on α^* being bad we get that

$$\Pr_{k,\alpha^*,s^*,\mathsf{iO},\mathcal{A}}\left[\mathsf{P}^*\Big(\mathsf{iO}\big(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\}, \alpha^*, \mathsf{V}_1(x^*, \alpha; s^*)]\big)\Big) = (\alpha^*, \gamma)\right]$$

is slightly higher than random chance.

Conclusion

Assuming sub-exponentially secure iO and sub-exponentially secure PRFs as well as exponentially secure input-hiding obfuscation for multi-bit point functions, three round zero-knowledge proofs can only exist for languages in BPP.

Thanks!

`ia.cr/2018/167`