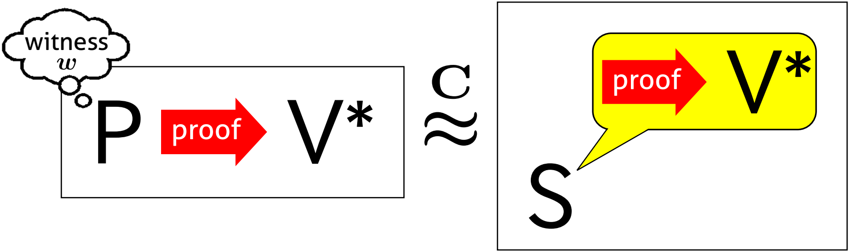
Constant-round Leakage-Resilient Zero-Knowledge from Collision Resistance

Susumu Kiyoshima

NTT, Japan.

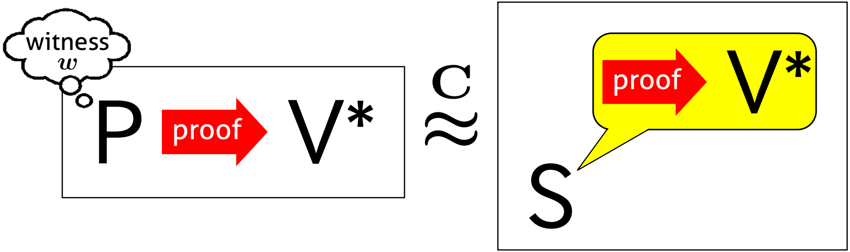
Zero-Knowledge

- ZK $\Leftrightarrow \forall$ verifier V^*, $\exists$ simulator $\mathcal{S}$ s.t.



Zero-Knowledge

- ZK $\Leftrightarrow \forall$ verifier V^*, $\exists$ simulator $\mathcal{S}$ s.t.



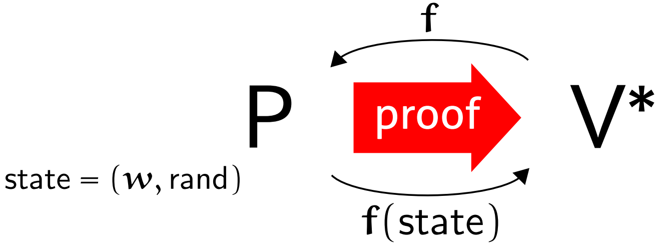
No security if P's state (w and randomness) is leaked!

$\Rightarrow$ No security against side-channel attack

Leakage-Resilient ZK (Informally)

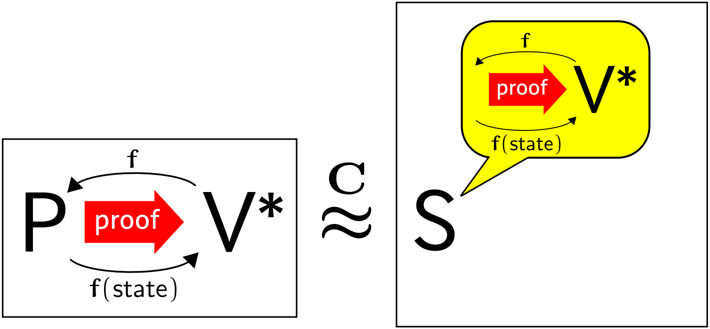
Leakage-resilient ZK [Garg-Jain-Sahai, 2011]
$\approx$ ZK against V^* who obtains leakage of P's state

where V^* who obtains leakage of P's state
$= V^*$ who makes any leakage queries



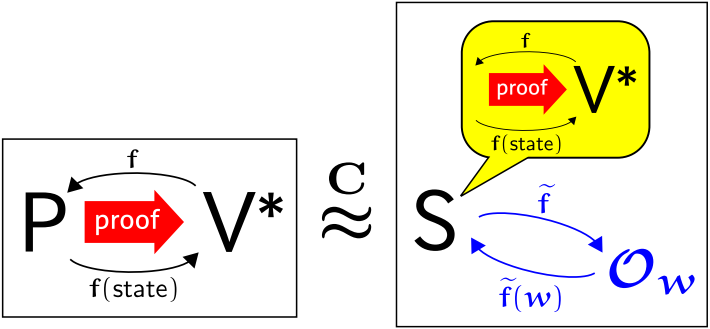
Leakage-Resilient ZK (More Formally)

- Leakage-resilient ZK $\Leftrightarrow \forall V^* \exists \mathcal{S}$ s.t.



Leakage-Resilient ZK (More Formally)

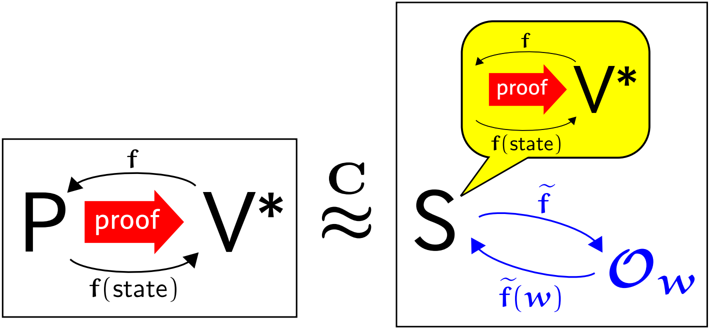
- Leakage-resilient ZK $\Leftrightarrow \forall V^* \exists \mathcal{S}$ s.t.



Note: $\mathcal{S}$ can obtain leakage of witness w from $\mathcal{O}_w$

Leakage-Resilient ZK (More Formally)

- Leakage-resilient ZK $\Leftrightarrow \forall V^* \exists \mathcal{S}$ s.t.



Note: $\mathcal{S}$ can obtain leakage of witness w from $\mathcal{O}_w$

Requirement: If V^* obtains ℓ-bit of leakage, $\mathcal{S}$ obtains at most ℓ-bit of leakage

Known Results

- [Garg-Jain-Sahai, 2011]
  - **Security:** Relaxed notion of leakage-resilient ZK
    (where $\mathcal{S}$ can obtain more leakage than V^*)
  - **# of Rounds**: $\geq \omega(\log n)$
  - **Assumption:** Existence of one-way functions

Known Results

- [Garg-Jain-Sahai, 2011]
  - **Security:** Relaxed notion of leakage-resilient ZK
    (where $\mathcal{S}$ can obtain more leakage than V^*)
  - **# of Rounds**: $\geq \omega(\log n)$
  - **Assumption:** Existence of one-way functions
- [Pandey, 2014]
  - **Security:** Leakage-resilient ZK ✔
  - **# of Rounds**: Constant ✔
  - **Assumption:** DDH assumption $+$ Existence of CR hash

Known Results

- [Garg-Jain-Sahai, 2011]
  - **Security:** Relaxed notion of leakage-resilient ZK (where $\mathcal{S}$ can obtain more leakage than V^*)
  - **# of Rounds**: $\geq \omega(\log n)$
  - **Assumption:** Existence of one-way functions
- [Pandey, 2014]
  - **Security:** Leakage-resilient ZK ✔
  - **# of Rounds**: Constant ✔
  - **Assumption:** DDH assumption $+$ Existence of CR hash

Is DDH really necessary?

Our Result

Our Result

Theorem

Assume existence of collision-resistant hash functions. There exists constant-round public-coin leakage-resilient ZK argument for NP.

Compared with previous work [Pandey, 2014]:

- **Security:** same
- **# of Rounds:** same (asymptotically)
- **Assumption:** DDH is no longer required!

Our Result

Theorem

Assume existence of collision-resistant hash functions. There exists constant-round public-coin leakage-resilient ZK argument for NP.

Additional Property: **Leakage-Resilient Soundness**

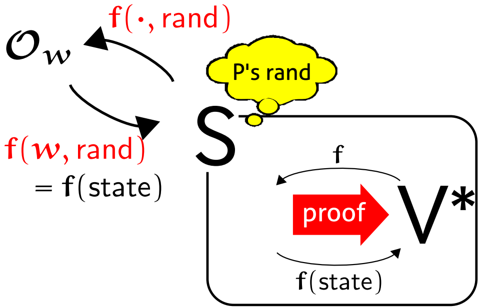
- Soundness for P^* who obtains unbounded amount of leakage
  (Previous leakage-resilient ZK is not sound in such a setting)
- Implied by public-coin property

Our Techniques

Simulator's Basic Strategy

It suffices for $\mathcal{S}$ to simulate P's msg and randomness

- **Recall:** $\mathcal{S}$'s goal is to simulate P's msg and leakage
- If $\mathcal{S}$ can simulate P's msg and randomness, then:



Road-map to Our Leakage-Resilient ZK

Step 1. Construct a tool:

Construct instance-based equivocal com with "nice" leakage-resilient property

- based on one-way functions
- possibly of independent interest

Step 2. Use the tool:

- Obtain leakage-resilient ZK by using it in "nice" way

NTT

Road-map to Our Leakage-Resilient ZK

Step 1. Construct a tool:

Construct instance-based equivocal com with "nice" leakage-resilient property

- based on one-way functions
- possibly of independent interest

Step 2. Use the tool:

- Obtain leakage-resilient ZK by using it in "nice" way

Recall: Instance-Based Equivocal Com

Commitment that is based on NP instance x

- **When x is false:**

binding

$$\xrightarrow{\mathrm{EquivCom}_x}$$

- **When x is true:**

equivocal
(given w)

$$\xrightarrow{\mathrm{EquivCom}_x}$$

Our Instance-Based Equivocal Com

We convert leakage-resilient ZK of [Garg-Jain-Sahai] to instance-based equivocal commitment

Fact 1: Leakage-resilient ZK of [Garg-Jain-Sahai] is based on Blum's Hamiltonicity ZK

Fact 2: Blum's Hamiltonicity ZK can be converted to instance-based equivocal commitment

[Feige-Shamir, Canetti-Lindell-Ostrovsky-Sahai, Lindell-Zarosim]

Our Instance-Based Equivocal Com

We convert leakage-resilient ZK of [Garg-Jain-Sahai]
to instance-based equivocal commitment

Fact 1: Leakage-resilient ZK of [Garg-Jain-Sahai] is based on Blum's Hamiltonicity ZK

Fact 2: Blum's Hamiltonicity ZK can be converted to instance-based equivocal commitment
[Feige-Shamir, Canetti-Lindell-Ostrovsky-Sahai, Lindell-Zarosim]

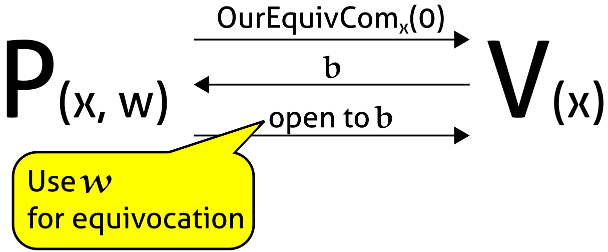
What property does OurEquivCom have?

Nice Property of OurEquivCom

Nice Property (Informal)

Given $b \in \{0, 1\}$, we can simulate P's msg/rand of **commit-then-equivocate-to-b**

commit-then-equivocate-to-b



Road-map to Our Leakage-Resilient ZK

Innovative R&D by NTT

Step 1. Construct a tool:

Construct instance-based equivocal com with "nice" leakage-resilient property

- based on one-way functions
- possibly of independent interest

Step 2. Use the tool:

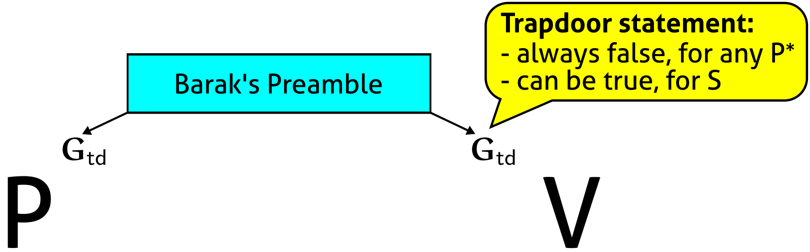
- Obtain leakage-resilient ZK by using it in "nice" way

NTT

Preliminary: Barak's Preamble

Preamble stage of Barak's non-BB ZK [Barak, 2001]

- P and V obtain trapdoor statement G_{td} such that:

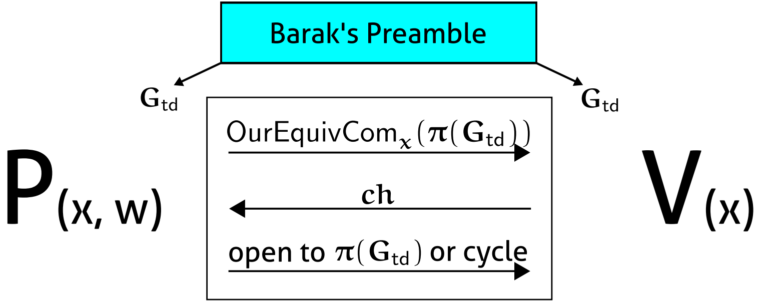


Note: Actually, we use a variant that is secure in leakage setting

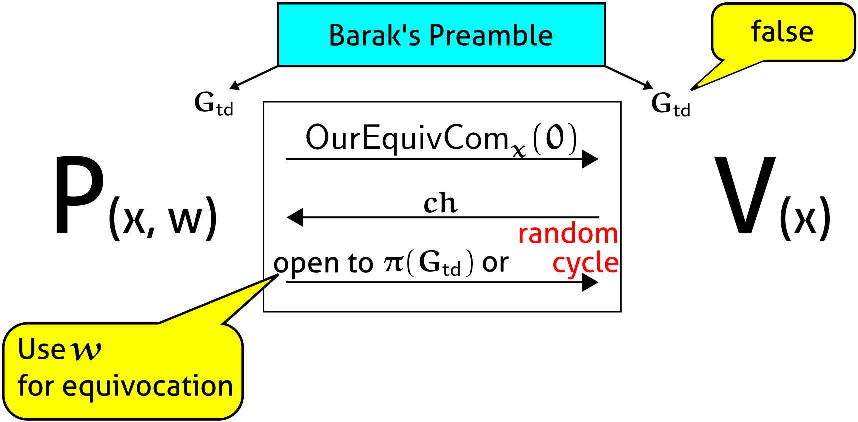
Our Leakage-Resilient ZK Protocol

We consider Hamiltonicity ZK s.t.

- $\mathrm{OurEquivCom}_{\chi}$ is used to commit to graph
- statement to be proven is trapdoor statement G_{td}

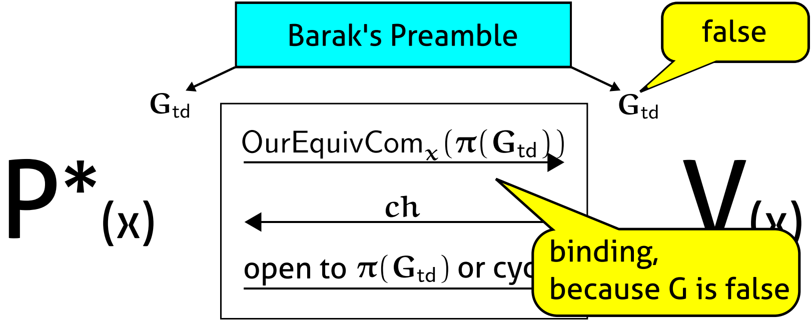


P can "simulate" Hamiltonicity ZK by equivocation



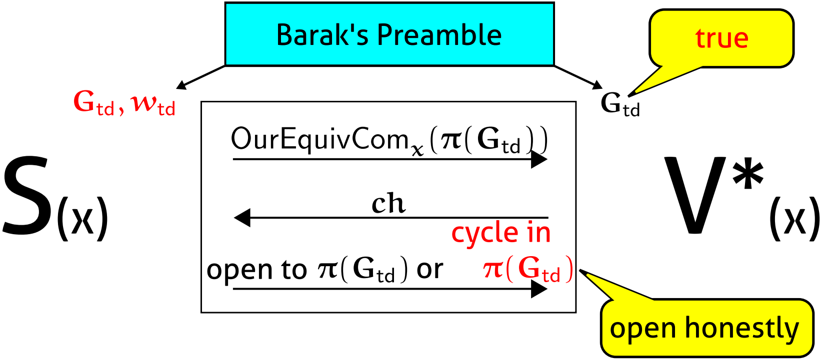
Soundness

Any P^* cannot prove G_{td} in Hamiltonicity ZK because of its soundness



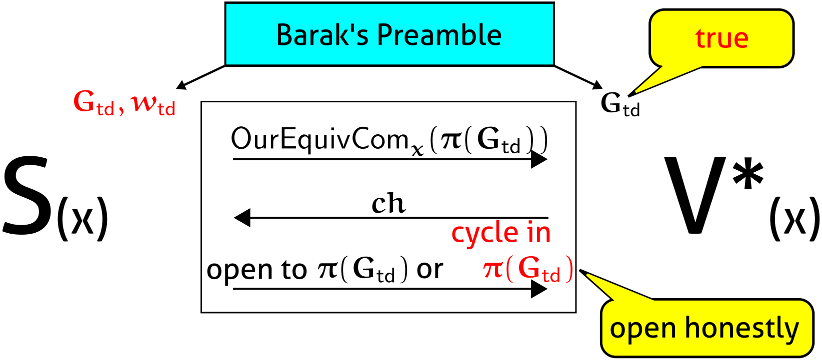
Warm-Up: Zero-Knowledge

$\mathcal{S}$ can prove G_{td} in Hamiltonicity ZK "honestly"



Warm-Up: Zero-Knowledge

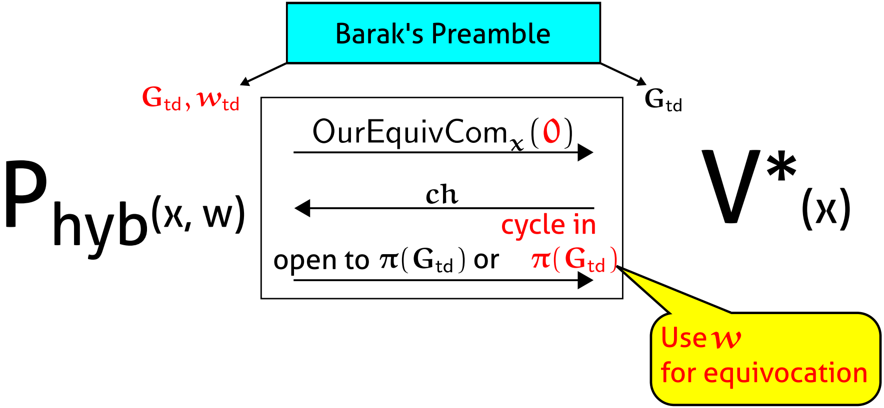
$\mathcal{S}$ can prove G_{td} in Hamiltonicity ZK "honestly"



Simulation of P's randomness?

Leakage-Resilient ZK

Consider hybrid experiment such that:



Leakage-Resilient ZK

P_{hyb} opens to $\pi(G_{td})$ or cycle in $\pi(G_{td})$

$\Rightarrow$ For each bit b in adjacent matrix of $\pi(G_{td})$, P_{hyb} does:

- Either **commit-then-equivocate-to-b**
- Or **commit-then-don't-open**

$\Rightarrow$ Use Nice Property! Q.E.D.

Nice Property of OurEquivCom:

Given $b \in \{0, 1\}$, we can simulate msg and randomness of **commit-then-equivocate-to-**b

Conclusion

Conclusion

Result

Using collision-resistant hash functions, we construct
leakage-resilient ZK argument for NP

(i.e., ZK argument that remains secure when honest party's state is leaked)

✔ We assume only the existence of CR hash functions
- Previous work additionally assumes DDH assumption

✔ Both ZKness and soundness hold in leakage setting
- Previous work doesn't sound under unbounded leakage

Thank you!