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

Susumu Kiyoshima

NTT, Japan.
Zero-Knowledge

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

```
\text{witness } w \\
P \xrightarrow{\text{proof}} V^* \\
\cong C \\
S \xrightarrow{\text{proof}} V^*
```
Zero-Knowledge

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

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]

\[ \approx ZK \text{ against } V^* \text{ who obtains leakage of } P\text{'s state} \]

where \( V^* \) who obtains leakage of \( P\text{'s state} \)

\[ = V^* \text{ who makes any leakage queries} \]

\[
\begin{array}{ccc}
\text{P} & \overset{\text{proof}}{\rightarrow} & \text{V*} \\
\text{state} = (w, \text{rand}) & \rightarrow & f(\text{state}) \\
\end{array}
\]
Leakage-Resilient ZK (More Formally)

- Leakage-resilient ZK $\iff \forall V^* \exists S$ s.t.

\[ f \text{ proof } V^* \quad \text{and} \quad f(\text{state}) \]
Leakage-Resilient ZK (More Formally)

- Leakage-resilient ZK $\iff \forall V^* \exists S$ s.t.

\[ P \xymatrix@C=10pt{ \ar[r]^-{f} & \ar[r]^-{f(\text{state})} & V^* \ar[r]^-{\text{proof}} & S \ar[r]^-{\tilde{f}} & \tilde{f}(w) \ar[r]^-{O_w} & V^* } \]

Note: $S$ can obtain leakage of witness $w$ from $O_w$
Leakage-Resilient ZK (More Formally)

- Leakage-resilient ZK ⇔ ∀V* ∃S s.t.

Note: $S$ can obtain leakage of witness $w$ from $O_w$

Requirement: If $V^*$ obtains $\ell$-bit of leakage, $S$ obtains at most $\ell$-bit of leakage
Known Results

- **[Garg-Jain-Sahai, 2011]**
  - **Security:** Relaxed notion of leakage-resilient ZK (where $S$ 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 $S$ 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 $S$ 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 $S$ to simulate $P$'s msg and randomness

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

\[
O_w \xrightarrow{f(\cdot, \text{rand})} S
\]

\[
f(w, \text{rand}) = f(\text{state})
\]

\[
P'\text{s rand}
\]

\[
V^*
\]

\[
f(\text{state})
\]

\[
\text{proof}
\]
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
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 Commitment that is based on NP instance $\chi$

- **When $\chi$ is false:**

  $$\text{binding}$$

  $\text{EquivCom}_{\chi}$

- **When $\chi$ is true:**

  $\text{equivocal (given } \omega \text{)}$

  $\text{EquivCom}_{\chi}$
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 (Informal)

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

\[
P(x, w) \xrightarrow{\text{commit-then-equivocate-to-$b$}} \text{OurEquivCom}_x(0) \xrightarrow{b} \text{open to } b \xrightarrow{\text{use } w \text{ for equivocation}} V(x)
\]
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
Preliminary: Barak's Preamble

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

- P and V obtain trapdoor statement $G_{td}$ such that:
  - always false, for any $P^*$
  - can be true, for $S$

Note: Actually, we use a variant that is secure in leakage setting
Our Leakage-Resilient ZK Protocol

We consider Hamiltonicity ZK s.t.

- OurEquivCom_\chi is used to commit to graph
- statement to be proven is trapdoor statement G_{td}

\begin{align*}
\text{OurEquivCom}_\chi(\pi(G_{td})) & \quad \text{ch} \\
\text{open to } \pi(G_{td}) \text{ or cycle} &
\end{align*}

P(x, w) \quad V(x)
Correctness

$P$ can "simulate" Hamiltonicity ZK by equivocation

$P(x, w)$

Barak's Preamble

$G_{td}$

OurEquivCom$_x(0)$

$G_{td}$

$V(x)$

Use $\omega$ for equivocation

Copyright © 2016 NTT corp. All Rights Reserved.
Soundness

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

$P^*(x)$

Barak's Preamble

OurEquivCom$_{\chi}(\pi(G_{td}))$

$\text{ch}$

open to $\pi(G_{td})$ or cycle

false

binding, because $G$ is false
Warm-Up: Zero-Knowledge

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

Barak's Preamble

OurEquivCom$_{x}(\pi(G_{td}))$

$S(x)$

$V^*(x)$

$G_{td}, \nu_{td}$

true

open honestly

Copyright © 2016 NTT corp. All Rights Reserved.
Warm-Up: Zero-Knowledge

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

Simulation of $P$'s randomness?
Leakage-Resilient ZK

Consider hybrid experiment such that:

\[ \text{Barak's Preamble} \]

\[ \text{Phyb}(x, w) \]

\[ G_{td}, \nu_{td} \]

\[ G_{td} \]

\[ \text{OurEquivCom}_x(0) \]

\[ \text{ch} \]

\[ \text{cycle in} \]

\[ \text{open to } \pi(G_{td}) \text{ or } \pi(G_{td}) \]

\[ V^*(x) \]

Use \( \nu \) for equivocation

Copyright © 2016 NTT corp. All Rights Reserved.
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!