S C I E N C E ▪ P A S S I O N ▪ T E C H N O L O G Y

Revisiting Cryptographic Accumulators, Additional Properties and Relations to other Primitives

David Derler, Christian Hanser, and Daniel Slamanig,
IAIK, Graz University of Technology

April 21, 2015

Outline

1. Introduction
2. A Unified Formal Model
3. Accumulators from Zero-Knowledge Sets
4. Black-Box Construction of Commitments

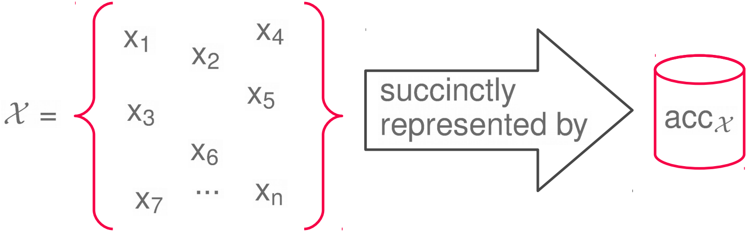
Outline

1. Introduction
2. A Unified Formal Model
3. Accumulators from Zero-Knowledge Sets
4. Black-Box Construction of Commitments

Static Accumulators

Finite set

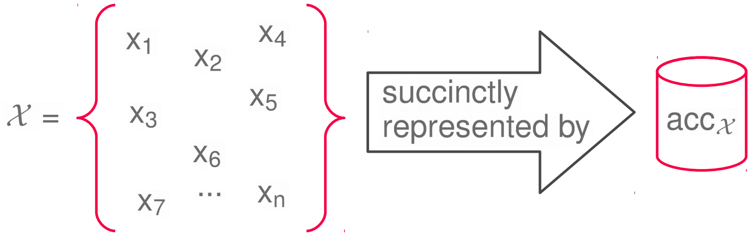
Accumulator



Static Accumulators

Finite set

Accumulator



Witnesses wit_x certifying membership of x in $\mathrm{acc}_{\mathcal{X}}$

- Efficiently computable $\forall\, x \in \mathcal{X}$
- Intractable to compute $\forall\, x \notin \mathcal{X}$

A Simple Example - RSA Accumulator

- RSA modulus N
- Accumulator for $\mathcal{X} = \{x_1, \ldots, x_n\}$
  - $\mathrm{acc}_{\mathcal{X}} \leftarrow g^{x_1 \cdot \ \cdots \ \cdot x_{i-1} \cdot x_i \cdot x_{i+1} \cdot \ \cdots \ \cdot x_n} \mod N$

A Simple Example - RSA Accumulator

- RSA modulus N
- Accumulator for $\mathcal{X} = \{x_1, \ldots, x_n\}$
  - $\mathsf{acc}_{\mathcal{X}} \leftarrow g^{x_1 \cdot \ \cdots \ \cdot x_{i-1} \cdot x_i \cdot x_{i+1} \cdot \ \cdots \ \cdot x_n} \mod N$
- Witness for x_i:
  - $\mathsf{wit}_{x_i} \leftarrow g^{x_1 \cdot \ \cdots \ \cdot x_{i-1} \cdot x_{i+1} \cdot \ \cdots \ \cdot x_n} \mod N$

A Simple Example - RSA Accumulator

- RSA modulus N
- Accumulator for $\mathcal{X} = \{x_1, \ldots, x_n\}$
  - $\mathsf{acc}_{\mathcal{X}} \leftarrow g^{x_1 \cdot \ \cdots \ \cdot x_{i-1} \cdot x_i \cdot x_{i+1} \cdot \ \cdots \ \cdot x_n} \mod N$
- Witness for x_i:
  - $\mathsf{wit}_{x_i} \leftarrow g^{x_1 \cdot \ \cdots \ \cdot x_{i-1} \cdot x_{i+1} \cdot \ \cdots \ \cdot x_n} \mod N$
- Verify witness:
  - Check whether $(\mathsf{wit}_{x_i})^{x_i} \equiv \mathsf{acc}_{\mathcal{X}} \mod N$.

A Simple Example - RSA Accumulator

- RSA modulus N
- Accumulator for $\mathcal{X} = \{x_1, \ldots, x_n\}$
  - $\mathsf{acc}_{\mathcal{X}} \leftarrow g^{x_1 \cdot \ldots \cdot x_{i-1} \cdot x_i \cdot x_{i+1} \cdot \ldots \cdot x_n} \mod N$
- Witness for x_i:
  - $\mathsf{wit}_{x_i} \leftarrow g^{x_1 \cdot \ldots \cdot x_{i-1} \cdot x_{i+1} \cdot \ldots \cdot x_n} \mod N$
- Verify witness:
  - Check whether $(\mathsf{wit}_{x_i})^{x_i} \equiv \mathsf{acc}_{\mathcal{X}} \mod N$.
- Witness for $y \notin \mathcal{X}$
  - Would imply breaking strong RSA
  - ...unless factorization of N is known.

Dynamic and Universal Features

Dynamically add/delete elements

- ...to/from accumulator $\mathsf{acc}_{\mathcal{X}}$
- Update witnesses accordingly
- All updates independent of $|\mathcal{X}|$

Dynamic and Universal Features

Dynamically add/delete elements

- ...to/from accumulator acc$_{\mathcal{X}}$
- Update witnesses accordingly
- All updates independent of $|\mathcal{X}|$

Universal features

- Demonstrate non-membership
- Non-membership witness wit$_x$
  - Efficiently computable $\forall\ x \notin$ acc$_{\mathcal{X}}$
  - Intractable to compute $\forall\ x \in$ acc$_{\mathcal{X}}$

Motivation

Accumulators widely used in various applications

- e.g., credential revocation, malleable signatures, ...
- Previous models tailored to specific constructions
  - Different features
  - Private/public updatability

Motivation

Accumulators widely used in various applications

- e.g., credential revocation, malleable signatures, ...
- Previous models tailored to specific constructions
  - Different features
  - Private/public updatability

Thus, accumulators not usable as black-boxes

- Limited exchangeability when used in other constructions
- Relations to other primitives hard to study

Contribution

- Unified formal model for
  - **Static/dynamic/universal** accumulators
  - Introduces **randomized** and **bounded** accumulators
  - Introduces **indistinguishability**
  - Includes **undeniability**

Contribution

- Unified formal model for
  - **Static/dynamic/universal** accumulators
  - Introduces **randomized** and **bounded** accumulators
  - Introduces **indistinguishability**
  - Includes **undeniability**
- First constructions fulfilling new notions
  - First **indistinguishable, dynamic** acc
  - First **undeniable, indistinguishable, universal** acc

Contribution

- Unified formal model for
  - **Static/dynamic/universal** accumulators
  - Introduces **randomized** and **bounded** accumulators
  - Introduces **indistinguishability**
  - Includes **undeniability**
- First constructions fulfilling new notions
  - First **indistinguishable, dynamic** acc
  - First **undeniable, indistinguishable, universal** acc
- **Black-box relations** to commitments and ZK-sets

Contribution

- Unified formal model for
  - **Static/dynamic/universal** accumulators
  - Introduces **randomized** and **bounded** accumulators
  - Introduces **indistinguishability**
  - Includes **undeniability**
- First constructions fulfilling new notions
  - First **indistinguishable, dynamic** acc
  - First **undeniable, indistinguishable, universal** acc
- **Black-box relations** to commitments and ZK-sets
- **Exhaustive classification** of existing schemes (see Paper)

Outline

1. Introduction
2. A Unified Formal Model
3. Accumulators from Zero-Knowledge Sets
4. Black-Box Construction of Commitments

Algorithms

Static Accumulators - Algorithms

- *Gen*
- *Eval*
- *WitCreate*
- *Verify*

Algorithms

Static Accumulators - Algorithms

- *Gen*
- *Eval*
- *WitCreate*
- *Verify*

We call accumulators

- t-bounded, if an upper bound for the set size exists
- randomized, if *Eval* is probabilistic
  - $Eval_r$ to make used randomness explicit

Algorithms

Static Accumulators - Algorithms

- *Gen*
- *Eval*
- *WitCreate*
- *Verify*

We call accumulators

- t-bounded, if an upper bound for the set size exists
- randomized, if *Eval* is probabilistic
  - $Eval_r$ to make used randomness explicit

Dynamic Accumulators additionally provide

- *Add*
- *Delete*
- *WitUpdate*

Algorithms - Universal Accumulators

Static or dynamic accumulator, but in addition

- *WitCreate* and *Verify* take additional parameter *type*

Algorithms - Universal Accumulators

Static or dynamic accumulator, but in addition

- *WitCreate* and *Verify* take additional parameter *type*
  - Membership ($type = 0$) vs. non-membership mode ($type = 1$)

Algorithms - Universal Accumulators

Static or dynamic accumulator, but in addition

- *WitCreate* and *Verify* take additional parameter *type*
  - Membership ($type = 0$) vs. non-membership mode ($type = 1$)
- For dynamic accumulator schemes
  - The same additionally applies to *WitUpdate*

Security

- Correctness
- Collision freeness
- Undeniability
- Indistinguishability

Security - Collision Freeness

Experiment $\mathbf{Exp}^{cf}_{\kappa}(\cdot)$:

$(\mathsf{sk}_{\mathsf{acc}}, \mathsf{pk}_{\mathsf{acc}}) \leftarrow Gen(1^{\kappa}, t)$

$\mathcal{A}$

$\xrightarrow{\mathsf{pk}_{\mathsf{acc}}}$

$\mathcal{O}^{Eval}, \boxed{\mathcal{O}^{Add}, \mathcal{O}^{Delete}}$

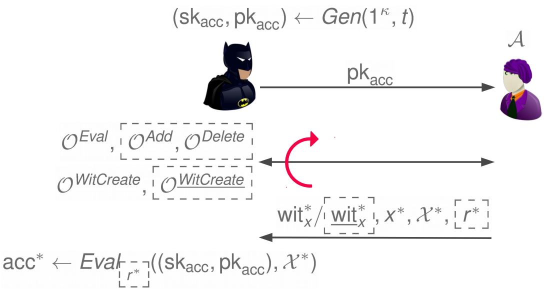
$\mathcal{O}^{WitCreate}, \boxed{\mathcal{O}^{\underline{WitCreate}}}$

$\mathsf{wit}^*_x / \boxed{\underline{\mathsf{wit}}^*_x}, x^*, \mathcal{X}^*, \boxed{r^*}$

$\mathsf{acc}^* \leftarrow Eval_{\boxed{r^*}}((\mathsf{sk}_{\mathsf{acc}}, \mathsf{pk}_{\mathsf{acc}}), \mathcal{X}^*)$

Security - Collision Freeness

Experiment $\mathbf{Exp}^{cf}_{\kappa}(\cdot)$:

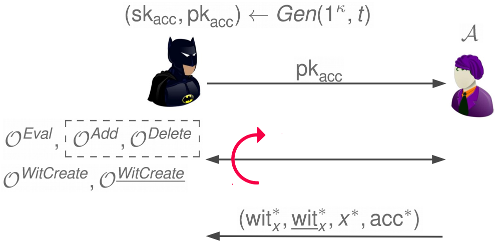


- $\mathcal{A}$ wins if
  - wit^*_x is membership witness for non-member, or
  - $\underline{\mathrm{wit}}^*_x$ is non-membership witness for member

Security - Undeniability

Defined for universal accumulators

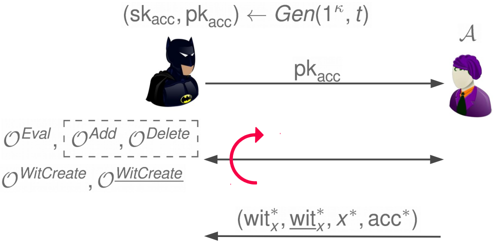
Experiment $\mathbf{Exp}^{ud}_{\kappa}(\cdot)$:



Security - Undeniability

Defined for universal accumulators

Experiment $\mathbf{Exp}_{\kappa}^{ud}(\cdot)$:



- $\mathcal{A}$ wins if verification succeeds for both wit_x^* and $\underline{\mathsf{wit}}_x^*$

Undeniability $\overset{\nLeftarrow}{\Rightarrow}$ Collision Freeness

We show that

- Efficient $\mathcal{A}^{cf}$ can be turned into efficient $\mathcal{A}^{ud}$

Undeniability $\overset{\not\Leftarrow}{\Rightarrow}$ Collision Freeness

We show that

- Efficient $\mathcal{A}^{cf}$ can be turned into efficient $\mathcal{A}^{ud}$

Other direction does not hold [BLL02]

Security - Indistinguishability I

So far, no meaningful formalization

- Existing formalization allows to prove indistinguishability
- for trivially distinguishable accumulators

Security - Indistinguishability I

So far, no meaningful formalization

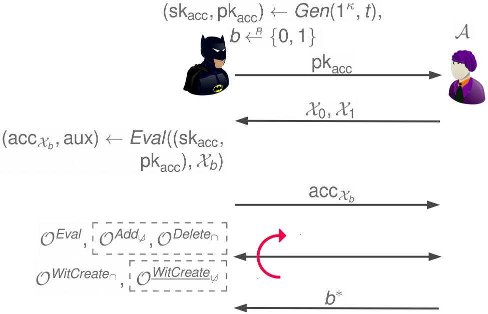
- Existing formalization allows to prove indistinguishability
- for trivially distinguishable accumulators

We provide formalization

- not suffering from shortcomings above

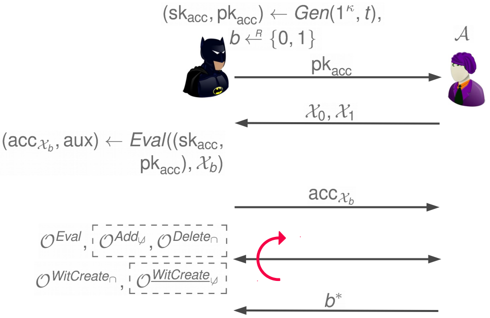
Security - Indistinguishability II

Experiment $\mathbf{Exp}_{\kappa}^{ind}(\cdot)$:



Security - Indistinguishability II

Experiment $\mathbf{Exp}^{ind}_{\kappa}(\cdot)$:



- $\mathcal{A}$ wins if guess correct

David Derler, IAIK, Graz University of Technology

Security - Indistinguishability III

Ad-hoc solution in literature

- Insert a (secret) random value z into acc.

Security - Indistinguishability III

Ad-hoc solution in literature

- Insert a (secret) random value z into acc.

However, weakens collision freeness

- Witness for z efficiently computable by definition

David Derler, IAIK, Graz University of Technology

Security - Indistinguishability III

Ad-hoc solution in literature

- Insert a (secret) random value z into acc.

However, weakens collision freeness

- Witness for z efficiently computable by definition

Thus, we distinguish

- Indistinguishability
- Collision freeness weakening (cfw)-indistinguishability

Security - Indistinguishability III

Ad-hoc solution in literature

- Insert a (secret) random value z into acc.

However, weakens collision freeness

- Witness for z efficiently computable by definition

Thus, we distinguish

- Indistinguishability
- Collision freeness weakening (cfw)-indistinguishability

We modify [Ngu05] to provide indistinguishability

- First **indistinguishable *t*-bounded dynamic** accumulator

Outline

1. Introduction
2. A Unified Formal Model
3. Accumulators from Zero-Knowledge Sets
4. Black-Box Construction of Commitments

Zero-Knowledge Sets

Commit to a set $\mathcal{X}$

- Prove predicates of the form
  - $x \in \mathcal{X}$
  - $x \notin \mathcal{X}$
  - While not revealing anything else about $\mathcal{X}$

Zero-Knowledge Sets

Commit to a set $\mathcal{X}$

- Prove predicates of the form
  - $x \in \mathcal{X}$
  - $x \notin \mathcal{X}$
  - While not revealing anything else about $\mathcal{X}$

Observation

- Similar to undeniable indistinguishable accumulators

Zero-Knowledge Sets

Commit to a set $\mathcal{X}$

- Prove predicates of the form
  - $x \in \mathcal{X}$
  - $x \notin \mathcal{X}$
  - While not revealing anything else about $\mathcal{X}$

Observation

- Similar to undeniable indistinguishable accumulators
- Algorithms compatible
- Security notions similar

Accumulators from Zero-Knowledge Sets

Security notions

- Perfect completeness $\equiv$ correctness
- Soundness $\equiv$ undeniability

Accumulators from Zero-Knowledge Sets

Security notions

- Perfect completeness $\equiv$ correctness
- Soundness $\equiv$ undeniability
- Zero-knowledge
  - Simulation-based notion
  - $\exists$ simulator $\mathcal{S}$, negl. ϵ, s.t. $\forall$ PPT distinguishers: Pr [distinguish sim/real] $\leq \epsilon(\kappa)$

Accumulators from Zero-Knowledge Sets

Security notions

- Perfect completeness $\equiv$ correctness
- Soundness $\equiv$ undeniability
- Zero-knowledge
    - Simulation-based notion
    - $\exists$ simulator $\mathcal{S}$, negl. ϵ, s.t. $\forall$ PPT distinguishers: Pr [distinguish sim/real] $\leq \epsilon(\kappa)$
    - We show that "zero-knowledge $\implies$ indistinguishability"
    - Other direction unclear, sim-based notion seems stronger

Accumulators from Zero-Knowledge Sets

Security notions

- Perfect completeness $\equiv$ correctness
- Soundness $\equiv$ undeniability
- Zero-knowledge
  - Simulation-based notion
  - $\exists$ simulator $\mathcal{S}$, negl. ϵ, s.t. $\forall$ PPT distinguishers: Pr [distinguish sim/real] $\leq \epsilon(\kappa)$
  - We show that “zero-knowledge $\implies$ indistinguishability”
  - Other direction unclear, sim-based notion seems stronger

First undeniable, unbounded, indistinguishable acc

- Nearly ZK sets $\rightarrow$ *t*-bounded

Outline

Commitments

- Compute commitment $\mathcal{C}$ to message m
- Later: provide opening $\mathcal{O}$ demonstrating that
  - $\mathcal{C}$ is commitment to m

Commitments

- Compute commitment $\mathcal{C}$ to message m
- Later: provide opening $\mathcal{O}$ demonstrating that
  - $\mathcal{C}$ is commitment to m
- Security (informal):
  - Correctness: straight forward

Commitments

- Compute commitment $\mathcal{C}$ to message m
- Later: provide opening $\mathcal{O}$ demonstrating that
  - $\mathcal{C}$ is commitment to m
- Security (informal):
  - Correctness: straight forward
  - Binding: Intractable to find $\mathcal{C}$, $\mathcal{O}$, $\mathcal{O}'$ such that $\mathcal{C}$ opens to two different messages $m \neq m'$

Commitments

- Compute commitment $\mathcal{C}$ to message m
- Later: provide opening $\mathcal{O}$ demonstrating that
  - $\mathcal{C}$ is commitment to m
- Security (informal):
  - Correctness: straight forward
  - Binding: Intractable to find $\mathcal{C}$, $\mathcal{O}$, $\mathcal{O}'$ such that $\mathcal{C}$ opens to two different messages $m \neq m'$
  - Hiding: For $\mathcal{C}$ to either m_0 or m_1. Intractable to decide whether $\mathcal{C}$ opens to m_0 or m_1

Commitments from Accumulators I

Use 1-bounded indistinguishable accumulators

- $\mathcal{C} \leftarrow \mathsf{acc}_{\{m\}}$
- $\mathcal{O} \leftarrow (m, r, \mathsf{wit}_m, aux)$ such that
  - $\mathsf{acc}_{\{m\}} = \mathit{Eval}_r((\emptyset, \mathsf{pk}_{\mathsf{acc}}), \{m\})$
  - $\mathit{Verify}(\mathsf{pk}_{\mathsf{acc}}, \mathsf{acc}_{\{m\}}, \mathsf{wit}_m, m) = \mathtt{true}$

Commitments from Accumulators I

Use 1-bounded indistinguishable accumulators

- $\mathcal{C} \leftarrow \mathsf{acc}_{\{m\}}$
- $\mathcal{O} \leftarrow (m, r, \mathsf{wit}_m, aux)$ such that
  - $\mathsf{acc}_{\{m\}} = \mathit{Eval}_r((\emptyset, \mathsf{pk}_{\mathsf{acc}}), \{m\})$
  - $\mathit{Verify}(\mathsf{pk}_{\mathsf{acc}}, \mathsf{acc}_{\{m\}}, \mathsf{wit}_m, m) = \mathtt{true}$
- Collision-freeness $\Rightarrow$ Binding

Commitments from Accumulators I

Use 1-bounded indistinguishable accumulators

- $\mathcal{C} \leftarrow \mathsf{acc}_{\{m\}}$
- $\mathcal{O} \leftarrow (m, r, \mathsf{wit}_m, \mathit{aux})$ such that
  - $\mathsf{acc}_{\{m\}} = \mathit{Eval}_r((\emptyset, \mathsf{pk}_{\mathsf{acc}}), \{m\})$
  - $\mathit{Verify}(\mathsf{pk}_{\mathsf{acc}}, \mathsf{acc}_{\{m\}}, \mathsf{wit}_m, m) = \mathtt{true}$
- Collision-freeness $\Rightarrow$ Binding
- Indistinguishability $\Rightarrow$ Hiding

Commitments from Accumulators I

Use 1-bounded indistinguishable accumulators

- $\mathcal{C} \leftarrow \mathsf{acc}_{\{m\}}$
- $\mathcal{O} \leftarrow (m, r, \mathsf{wit}_m, aux)$ such that
  - $\mathsf{acc}_{\{m\}} = \mathit{Eval}_r((\emptyset, \mathsf{pk}_{\mathsf{acc}}), \{m\})$
  - $\mathit{Verify}(\mathsf{pk}_{\mathsf{acc}}, \mathsf{acc}_{\{m\}}, \mathsf{wit}_m, m) = \mathtt{true}$
- Collision-freeness $\Rightarrow$ Binding
- Indistinguishability $\Rightarrow$ Hiding

Observe: cfw-indistinguishability not useful

Commitments from Accumulators II

Straight forward extension to set-commitments

- Use t-bounded accumulators
- Opening w.r.t. entire set

Commitments from Accumulators II

Straight forward extension to set-commitments

- Use t-bounded accumulators
- Opening w.r.t. entire set

Trapdoor commitments

- Use sk_{acc} as trapdoor

Conclusion

Unified model for accumulators

- Covering all features existing to date

Conclusion

Unified model for accumulators

- Covering all features existing to date

Introduce indistinguishability notion

- Provide first indistinguishable dynamic scheme

Conclusion

Unified model for accumulators

- Covering all features existing to date

Introduce indistinguishability notion

- Provide first indistinguishable dynamic scheme

Show relations to other primitives

- Commitments
- Zero-knowledge sets
  - Yields first undeniable, unbounded, indistinguishable, universal accumulator
- Inspiration for new constructions

Thank you.

david.derler@iaik.tugraz.at

Extended version: `http://eprint.iacr.org/2015/087`

References I

[BLL02] Ahto Buldas, Peeter Laud, and Helger Lipmaa. Eliminating counterevidence with applications to accountable certificate management. *Journal of Computer Security*, 10(3):273–296, 2002.

[Ngu05] Lan Nguyen. Accumulators from bilinear pairings and applications. In *Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings*, pages 275–292, 2005.