Construction of Universal Designated-Verifier Signatures and Identity-Based Signatures from Standard Signatures

Siamak Shahandashti[1] Rei Safavi-Naini[2]

[1]SCSSE & CCISR, Uni Wollongong, Australia
www.uow.edu.au/∼sfs166

[2]Dept Comp Sci & iCIS, Uni Calgary, Canada
www.cpsc.ucalgary.ca/∼rei

PKC 2008

Outline

What's a Universal Designated-Verifier Signature?

a.k.a. UDVS

- Basically: a signature scheme with an extra functionality
- Goal: to protect user privacy when using credentials
- Idea: transform signature s.t. it only convinces a particular verifier

What's a Universal Designated-Verifier Signature?

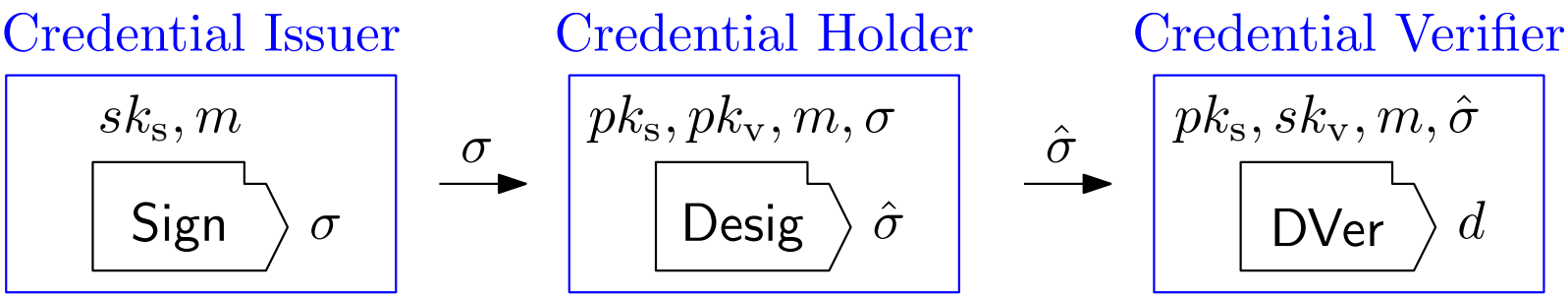
a.k.a. UDVS

- Basically: a signature scheme with an extra functionality
- Goal: to protect user privacy when using credentials
- Idea: transform signature s.t. it only convinces a particular verifier

What's a Universal Designated-Verifier Signature?

a.k.a. UDVS

- Basically: a signature scheme with an extra functionality
- Goal: to protect user privacy when using credentials
- Idea: transform signature s.t. it only convinces a particular verifier



How can we construct a UDVS?

- $\hat{\sigma}$ is a designated-verifier non-interactive proof of holding a valid signature on m.
- Jakobsson et al's intuition to verifier designation: *"Instead of proving X, Alice will prove the statement: Either X is true, or I am Bob."*
- In the Random Oracle Model, non-interactive proofs can be constructed using Fiat-Shamir heuristic from Σ protocols.
- So the only things we need are:
  - A Σ protocol for proof of knowledge of a signature on a message, and
  - A Σ protocol for proof of knowledge of the verifier's secret key.

How can we construct a UDVS?

- $\hat{\sigma}$ is a designated-verifier non-interactive proof of holding a valid signature on m.
- Jakobsson et al's intuition to verifier designation: *"Instead of proving X, Alice will prove the statement: Either X is true, or I am Bob."*
- In the Random Oracle Model, non-interactive proofs can be constructed using Fiat-Shamir heuristic from Σ protocols.
- So the only things we need are:
  - A Σ protocol for proof of knowledge of a signature on a message, and
  - A Σ protocol for proof of knowledge of the verifier's secret key.

How can we construct a UDVS?

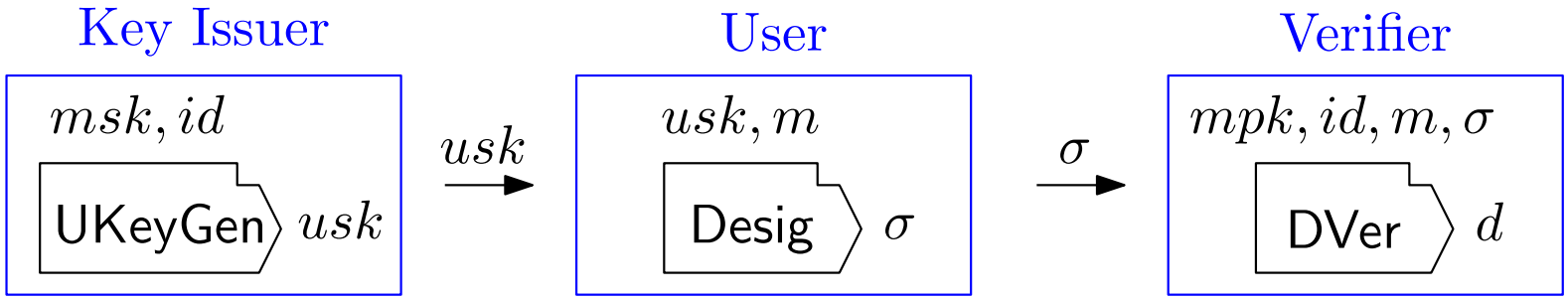
- $\hat{\sigma}$ is a designated-verifier non-interactive proof of holding a valid signature on m.
- Jakobsson et al's intuition to verifier designation: *"Instead of proving X, Alice will prove the statement: Either X is true, or I am Bob."*
- In the Random Oracle Model, non-interactive proofs can be constructed using Fiat-Shamir heuristic from Σ protocols.
- So the only things we need are:
  - A Σ protocol for proof of knowledge of a signature on a message, and
  - A Σ protocol for proof of knowledge of the verifier's secret key.

How can we construct a UDVS?

- $\hat{\sigma}$ is a designated-verifier non-interactive proof of holding a valid signature on m.
- Jakobsson et al's intuition to verifier designation: *"Instead of proving X, Alice will prove the statement: Either X is true, or I am Bob."*
- In the Random Oracle Model, non-interactive proofs can be constructed using Fiat-Shamir heuristic from Σ protocols.
- So the only things we need are:
  - A Σ protocol for proof of knowledge of a signature on a message, and
  - A Σ protocol for proof of knowledge of the verifier's secret key.

How can we construct an Identity-Based Signature?

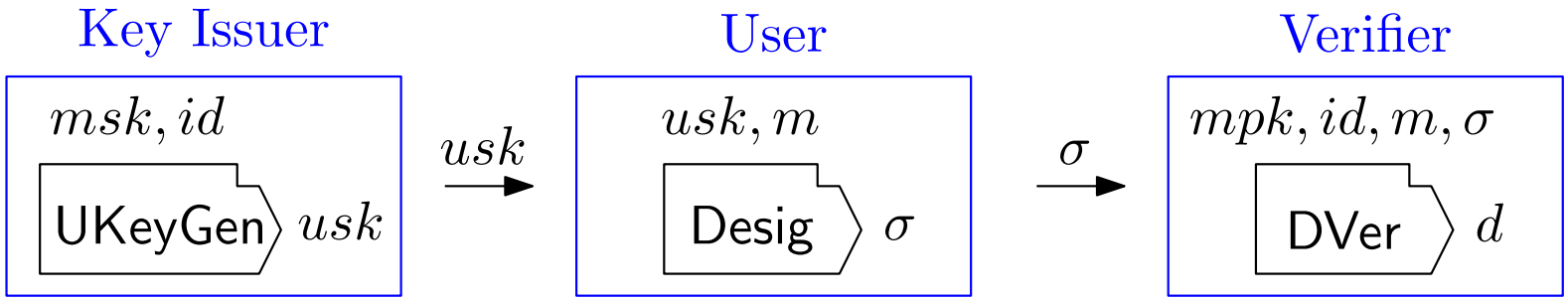
a.k.a. IBS



- σ is a signature on m that shows the signer has knowledge of usk
- In the Random Oracle Model, signatures can be constructed using Fiat-Shamir heuristic from Σ protocols.
- So again the only thing we need is:
  - A Σ protocol for proof of knowledge of a signature on a message.

How can we construct an Identity-Based Signature?

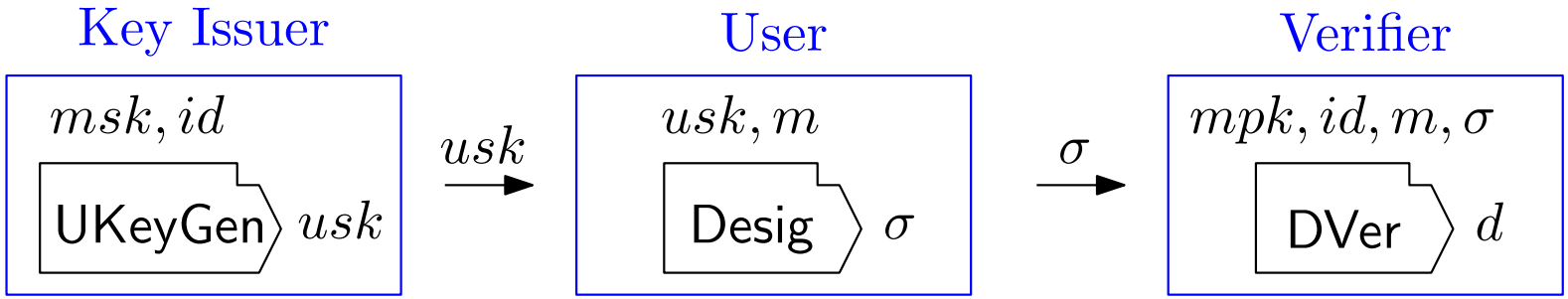
a.k.a. IBS



- σ is a signature on m that shows the signer has knowledge of usk
- In the Random Oracle Model, signatures can be constructed using Fiat-Shamir heuristic from Σ protocols.
- So again the only thing we need is:
  - A Σ protocol for proof of knowledge of a signature on a message.

How can we construct an Identity-Based Signature?

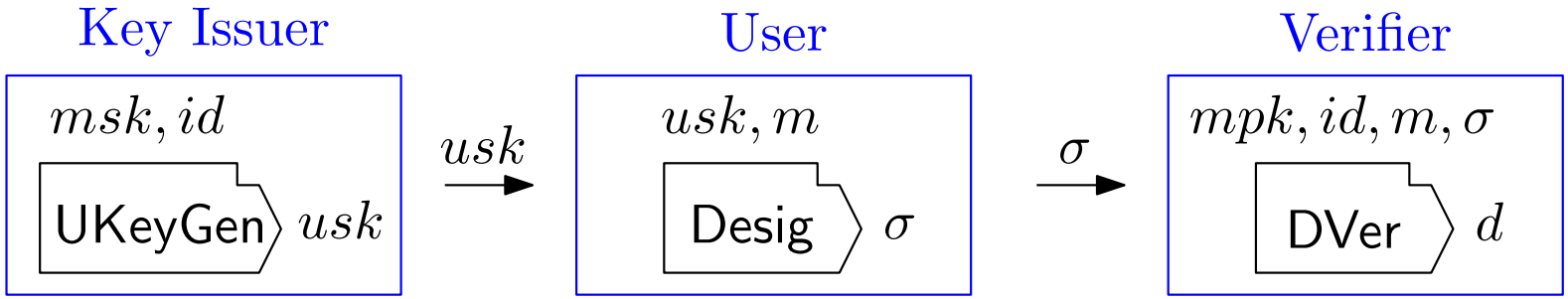
a.k.a. IBS



- σ is a signature on m that shows the signer has knowledge of usk
- In the Random Oracle Model, signatures can be constructed using Fiat-Shamir heuristic from Σ protocols.
- So again the only thing we need is:
  - A Σ protocol for proof of knowledge of a signature on a message.

How can we construct an Identity-Based Signature?

a.k.a. IBS



- σ is a signature on m that shows the signer has knowledge of usk
- In the Random Oracle Model, signatures can be constructed using Fiat-Shamir heuristic from Σ protocols.
- So again the only thing we need is:
  - A Σ protocol for proof of knowledge of a signature on a message.

So, What's the problem Then?

Although any NP relation has a Σ protocol, these generic protocols are normally not efficient!

Is there any more efficient way to do it?

Yes, There Is a Way!

We don't actually need strict honest-verifier zero-knowledge!

Example

Schnorr signature:

$$pk = (p, q, g, h = g^x), \quad \sigma = (c, z) : \quad c = H(g^z \cdot h^{-c}, m)$$

To prove knowledge of a signature

- give out $aux = g^z \cdot h^{-c}$
- prove knowledge of $z : \quad g^z = aux \cdot h^{H(aux,m)}$

Yes, There Is a Way!

We don't actually need strict honest-verifier zero-knowledge!

Example

Schnorr signature:

$$pk = (p, q, g, h = g^x), \quad \sigma = (c, z) : \quad c = H\left(g^z \cdot h^{-c}, m\right)$$

To prove knowledge of a signature

- give out $aux = g^z \cdot h^{-c}$
- prove knowledge of $z : \quad g^z = aux \cdot h^{H(aux,m)}$

Yes, There Is a Way!

We don't actually need strict honest-verifier zero-knowledge!

Example

Schnorr signature:

$$pk = (p, q, g, h = g^x), \quad \sigma = (c, z) : \quad c = H\left(g^z \cdot h^{-c}, m\right)$$

To prove knowledge of a signature

- give out $aux = g^z \cdot h^{-c}$
- prove knowledge of $z : \quad g^z = aux \cdot h^{H(aux,m)}$

Defining Class $\mathbb{C}$ of Signatures

There exist `Convert` and `Retrieve` s.t.

$\tilde{\sigma} \leftarrow \texttt{Convert}(pk, m, \sigma) \quad \Rightarrow \quad \sigma \leftarrow \texttt{Retrieve}(pk, m, \tilde{\sigma})$

and if $\tilde{\sigma} = (aux, pre)$ then there exists:

- An `AuxSim` that $\texttt{AuxSim}(pk, m)$ simulates aux, and
- A Σ protocol for proof of knowledge of a pre for known pk, m, and aux.

Defining Class $\mathbb{C}$ of Signatures

There exist `Convert` and `Retrieve` s.t.

$\tilde{\sigma} \leftarrow \texttt{Convert}(pk, m, \sigma) \quad \Rightarrow \quad \sigma \leftarrow \texttt{Retrieve}(pk, m, \tilde{\sigma})$

and if $\tilde{\sigma} = (aux, pre)$ then there exists:

- An `AuxSim` that $\texttt{AuxSim}(pk, m)$ simulates aux, and
- A Σ protocol for proof of knowledge of a pre for known pk, m, and aux.

Defining Class $\mathbb{C}$ of Signatures

There exist `Convert` and `Retrieve` s.t.

$\tilde{\sigma} \leftarrow \texttt{Convert}(pk, m, \sigma) \quad \Rightarrow \quad \sigma \leftarrow \texttt{Retrieve}(pk, m, \tilde{\sigma})$

and if $\tilde{\sigma} = (aux, pre)$ then there exists:

- An `AuxSim` that $\texttt{AuxSim}(pk, m)$ simulates aux, and
- A Σ protocol for proof of knowledge of a pre for known pk, m, and aux.

Defining Class $\mathbb{C}$ of Signatures

There exist `Convert` and `Retrieve` s.t.

$\tilde{\sigma} \leftarrow \texttt{Convert}(pk, m, \sigma) \quad \Rightarrow \quad \sigma \leftarrow \texttt{Retrieve}(pk, m, \tilde{\sigma})$

and if $\tilde{\sigma} = (aux, pre)$ then there exists:

- An `AuxSim` that $\texttt{AuxSim}(pk, m)$ simulates aux, and
- A Σ protocol for proof of knowledge of a pre for known pk, m, and aux.

Which Signatures Does Class $\mathbb{C}$ Cover?

RSA-FDH, Schnorr, Modified ElGamal, Boneh-Lynn-Shacham, Boneh-Boyen, Cramer-Shoup, Camenisch-Lysyanskaya-02, Camenisch-Lysyanskaya-04, Goldwasser-Micali-Rivest, Gennaro-Halevi-Rabin, and Cramer-Shoup.

But not PSS of Bellare and Rogaway!

Which Signatures Does Class $\mathbb{C}$ Cover?

RSA-FDH, Schnorr, Modified ElGamal, Boneh-Lynn-Shacham, Boneh-Boyen, Cramer-Shoup, Camenisch-Lysyanskaya-02, Camenisch-Lysyanskaya-04, Goldwasser-Micali-Rivest, Gennaro-Halevi-Rabin, and Cramer-Shoup.

But not PSS of Bellare and Rogaway!

How to Construct a UDVS from a Signature?

Use signature to sign

To designate:

$(aux, pre) \leftarrow \texttt{Convert}\,(pk_{\mathrm{S}}, m, \sigma)$

$\delta \leftarrow \texttt{SoK}\,\{(pre \vee sk_{\mathrm{V}}) : \texttt{Valid}\,(pk_{\mathrm{S}}, m, (aux, pre))\,, \texttt{Pair}\,(pk_{\mathrm{V}}, sk_{\mathrm{V}})\}$

$\hat{\sigma} \leftarrow (aux, \delta)$

Verification is straightforward.

How to Construct a UDVS from a Signature?

Use signature to sign

To designate:

$(aux, pre) \leftarrow \texttt{Convert}\,(pk_{\mathrm{s}}, m, \sigma)$

$\delta \leftarrow \texttt{SoK}\,\{(pre \vee sk_{\mathrm{v}}) : \texttt{Valid}\,(pk_{\mathrm{s}}, m, (aux, pre))\,, \texttt{Pair}\,(pk_{\mathrm{v}}, sk_{\mathrm{v}})\}$

$\hat{\sigma} \leftarrow (aux, \delta)$

Verification is straightforward.

How to Construct a UDVS from a Signature?

Use signature to sign

To designate:

$(aux, pre) \leftarrow \texttt{Convert}\,(pk_{\mathrm{s}}, m, \sigma)$

$\delta \leftarrow \texttt{SoK}\,\{(pre \vee sk_{\mathrm{v}}) : \texttt{Valid}\,(pk_{\mathrm{s}}, m, (aux, pre))\,, \texttt{Pair}\,(pk_{\mathrm{v}}, sk_{\mathrm{v}})\}$

$\hat{\sigma} \leftarrow (aux, \delta)$

Verification is straightforward.

Security of Our UDVS Construction

Let SS be any signature in $\mathbb{C}$ and P_{SS} be its underlying problem. Also, let KT be any key type in $\mathbb{K}$ and P_{KT} be its underlying problem. Then our UDVS construction:

- is *DV-unforgeable* if P_{SS} and P_{KT} are both hard.
- achieves *non-transferability privacy*.
- is *non-delegatable* if the challenge space of the proof protocol is big enough.

Security of Our UDVS Construction

Let SS be any signature in $\mathbb{C}$ and P_{SS} be its underlying problem. Also, let KT be any key type in $\mathbb{K}$ and P_{KT} be its underlying problem. Then our UDVS construction:

- is *DV-unforgeable* if P_{SS} and P_{KT} are both hard.
- achieves *non-transferability privacy*.
- is *non-delegatable* if the challenge space of the proof protocol is big enough.

Security of Our UDVS Construction

Let SS be any signature in $\mathbb{C}$ and P_{SS} be its underlying problem. Also, let KT be any key type in $\mathbb{K}$ and P_{KT} be its underlying problem. Then our UDVS construction:

- is *DV-unforgeable* if P_{SS} and P_{KT} are both hard.
- achieves *non-transferability privacy*.
- is *non-delegatable* if the challenge space of the proof protocol is big enough.

Security of Our UDVS Construction

Let SS be any signature in $\mathbb{C}$ and P_{SS} be its underlying problem. Also, let KT be any key type in $\mathbb{K}$ and P_{KT} be its underlying problem. Then our UDVS construction:

- is *DV-unforgeable* if P_{SS} and P_{KT} are both hard.
- achieves *non-transferability privacy*.
- is *non-delegatable* if the challenge space of the proof protocol is big enough.

How Good is Our Construction?

Comparison between Steinfeld et al's and our constructions

Scheme	Hard probl.	Desig cost		$\hat{\sigma}$ size	ND
		off-line	on-line		
DVSBM	BDH	**none**	1 pair.	**1.0 kb**	✗
BLS+DL	**CDH**	2 pair.	**1 mult.**	5.3 kb	✓
SchUDVS$_1$	SDH	**1 exp.**	1 exp.	2.0 kb	✗
SchUDVS$_2$	**DL**	2 exp.	1 exp.	**1.5 kb**	**?**
Schnorr+DL	**DL**	4 exp.	**1 mult.**	5.3 kb	✓
RSAUDVS	**RSA**	**1 exp.**	2 exp.	11.6 kb	**?**
RSA-FDH+DL	RSA & DL	2 exp.	**1 mult.**	**4.3 kb**	✓

ND: non-delegatability

Further Constructions

- *universal multi-designated-verifier signatures:* through non-interactive proof of knowledge of one out of $n + 1$ values: a (converted) signature and the secret keys of the n verifiers.
- designate more than one signature *at once:* e.g. to show at least k out of n certificates to a verifier, construct a non-interactive proof of knowledge of $k + 1$ out of $n + 1$ values: n (converted) signatures and the secret key of the verifier.
- a combination of the above two

Further Constructions

- *universal multi-designated-verifier signatures:* through non-interactive proof of knowledge of one out of $n + 1$ values: a (converted) signature and the secret keys of the n verifiers.
- designate more than one signature *at once:* e.g. to show at least k out of n certificates to a verifier, construct a non-interactive proof of knowledge of $k + 1$ out of $n + 1$ values: n (converted) signatures and the secret key of the verifier.
- a combination of the above two

Further Constructions

- *universal multi-designated-verifier signatures:* through non-interactive proof of knowledge of one out of $n + 1$ values: a (converted) signature and the secret keys of the n verifiers.
- designate more than one signature *at once:* e.g. to show at least k out of n certificates to a verifier, construct a non-interactive proof of knowledge of $k + 1$ out of $n + 1$ values: n (converted) signatures and the secret key of the verifier.
- a combination of the above two

How to Construct an IBS?

Use signature to issue user secret keys (signatures) on identities (messages)
$usk \leftarrow \mathsf{SS.Sign}(msk, id)$

To sign:
$(aux, pre) \leftarrow \mathtt{Convert}(mpk, id, usk)$
$\delta \leftarrow \mathtt{SoK}\{pre : \mathtt{Valid}(mpk, id, (aux, pre))\}(m)$
$\sigma \leftarrow (aux, \delta)$

Verification is straightforward.

How to Construct an IBS?

Use signature to issue user secret keys (signatures) on identities (messages)

$usk \leftarrow \mathsf{SS.Sign}\,(msk, id)$

To sign:

$(aux, pre) \leftarrow \mathtt{Convert}\,(mpk, id, usk)$

$\delta \leftarrow \mathsf{SoK}\,\{pre : \mathtt{Valid}\,(mpk, id, (aux, pre))\}\,(m)$

$\sigma \leftarrow (aux, \delta)$

Verification is straightforward.

How to Construct an IBS?

Use signature to issue user secret keys (signatures) on identities (messages)
$usk \leftarrow \mathsf{SS.Sign}\,(msk, id)$

To sign:
$(aux, pre) \leftarrow \mathtt{Convert}\,(mpk, id, usk)$
$\delta \leftarrow \mathsf{SoK}\,\{pre : \mathtt{Valid}\,(mpk, id, (aux, pre))\}\,(m)$
$\sigma \leftarrow (aux, \delta)$

Verification is straightforward.

Security and Further Construction

Let SS be a standard signature in $\mathbb{C}$ and P_{SS} be its underlying problem. Our IBS construction is ID-EUF-CMA-secure if P_{SS} is hard.

Further constructions:

- *hierarchical identity-based signatures*
- *identity-based universal designated verifier signatures*
- *identity-based ring signatures*

Security and Further Construction

Let SS be a standard signature in $\mathbb{C}$ and P_{SS} be its underlying problem. Our IBS construction is ID-EUF-CMA-secure if P_{SS} is hard.

Further constructions:

- *hierarchical identity-based signatures*
- *identity-based universal designated verifier signatures*
- *identity-based ring signatures*

Summary

Our constructions:

- are almost generic, yet comparable in size and cost.
- are provably non-delegatable and also offer signer-verifier setting independence.
- can be extended to generic UMDVS, HIBS, IBUDVS, and IBRS.

However:

- our security proofs are in the Random Oracle Model.
- our security reductions are not tight.

Summary

Our constructions:

- are almost generic, yet comparable in size and cost.
- are provably non-delegatable and also offer signer-verifier setting independence.
- can be extended to generic UMDVS, HIBS, IBUDVS, and IBRS.

However:

- our security proofs are in the Random Oracle Model.
- our security reductions are not tight.

Summary

Our constructions:

- are almost generic, yet comparable in size and cost.
- are provably non-delegatable and also offer signer-verifier setting independence.
- can be extended to generic UMDVS, HIBS, IBUDVS, and IBRS.

However:

- our security proofs are in the Random Oracle Model.
- our security reductions are not tight.

Summary

Our constructions:

- are almost generic, yet comparable in size and cost.
- are provably non-delegatable and also offer signer-verifier setting independence.
- can be extended to generic UMDVS, HIBS, IBUDVS, and IBRS.

However:

- our security proofs are in the Random Oracle Model.
- our security reductions are not tight.

Summary

Our constructions:

- are almost generic, yet comparable in size and cost.
- are provably non-delegatable and also offer signer-verifier setting independence.
- can be extended to generic UMDVS, HIBS, IBUDVS, and IBRS.

However:

- our security proofs are in the Random Oracle Model.
- our security reductions are not tight.

Acknowledgment and Further Reading

Thanks to:

- *i*CORE Information Security Lab of Uni of Calgary
- Shaoquan Jiang and anonymous reviewers of PKC ′08

Full paper:

Shahandashti and Safavi-Naini.
Construction of Universal Designated-Verifier Signatures and Identity-Based Signatures from Standard Signatures.
Cryptology ePrint Archive, Report 2007/462 (2007).
http://eprint.iacr.org/2007/462

Acknowledgment and Further Reading

Thanks to:

- *i*CORE Information Security Lab of Uni of Calgary
- Shaoquan Jiang and anonymous reviewers of PKC ′08

Full paper:

Shahandashti and Safavi-Naini.
Construction of Universal Designated-Verifier Signatures and Identity-Based Signatures from Standard Signatures.
Cryptology ePrint Archive, Report 2007/462 (2007).
`http://eprint.iacr.org/2007/462`