Multi Designated Verifier Signatures

Anca Nitulescu
Joint work with
Ivan Damgård, Helene Haagh, Rebekah Mercer, Claudio Orlandi, Sophia Yakoubov
Characters

- a signer $S$ (Snow White)
- a set of multiple verifiers $\mathcal{D}$ (Dwarfs)
- adversary should not learn the source of the messages
Scenario

Sender

Receivers

Adversary
Properties

Forge

Verify - Reject
Properties

- Corrupted verifiers
  Simulate

- Cannot link the signature to the signer
Digital Signatures

Sign

Verify - Accept

Learns the sender
Designated-Verifier

- Sign
- Verify - Accept
- Cannot Verify
Non-transferability

Corrupted verifier
Cannot prove

Not convinced
Inconsistency

Disagreement: not all verifiers come to the same conclusion for same authenticated message
MDVS

Stronger security notions

- Unforgeability
- Consistency
- Source Hiding
- Privacy of Identities
Outline

1. Formal Security Notions
2. Construction from FE
3. Construction from PSDVS
Security Definitions
Correctness
Any honest signature should verify for every designated verifier
Consistency

S cannot create an inconsistent signature: Accepted by a verifier $V_1$ and rejected by $V_2$
Consistency

By correctness, a honest signature should be accepted by all designated verifiers.
Unforgeability

An adversary cannot forge a signature to a honest verifier

**Strong:** Even when it corrupts part of the verifiers in $D$
Source Hiding

Even corrupted colluding verifiers should not be able to prove the source of the message to outsiders.
Source Hiding

The verifiers are able to simulate signatures.
No outsider can know where a given signature came from.
Unforgeability

A simulated signature is not a forgery
It does not convince a honest designated verifier
Inconsistency

Corrupt verifiers can simulate an inconsistent signature:
Rejected by honest V & Accepted by corrupted V'
Privacy of Identity

PSI: An outsider cannot tell who is the signer
Even after seeing prior signatures from those signers
Privacy of Identity

PDI: An outsider cannot tell the set of designated verifiers (for same cardinality)
# MDVS comparison

<table>
<thead>
<tr>
<th>Schemes</th>
<th>PSI</th>
<th>Verification</th>
<th>Simulation</th>
<th>Signature Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>[JSI96]</td>
<td>✗</td>
<td>Local</td>
<td>All</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>[NSM05]</td>
<td>✓</td>
<td>All</td>
<td>All</td>
<td>$O(</td>
</tr>
<tr>
<td>[LSMP07]</td>
<td>✗</td>
<td>Local</td>
<td>All</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>[MW08]</td>
<td>✓</td>
<td>All</td>
<td>All</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>[Ver08]</td>
<td>✓</td>
<td>All</td>
<td>All</td>
<td>$O(</td>
</tr>
<tr>
<td>[Tia12]</td>
<td>✓</td>
<td>Local</td>
<td>One</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Our FE-MDVS</td>
<td>✓</td>
<td>Local</td>
<td>any subset</td>
<td>$O(</td>
</tr>
<tr>
<td>Our PS-MDVS</td>
<td>✗</td>
<td>Local</td>
<td>any subset</td>
<td>$O(</td>
</tr>
</tbody>
</table>
Verify: **Local** vs **All**

**Local:** a single designated verifier can check

**All:** the designated verifiers need to work together in order to verify
## MDVS comparison

<table>
<thead>
<tr>
<th>Schemes</th>
<th>PSI</th>
<th>Verification</th>
<th>Simulation</th>
<th>Signature Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>[JSI96]</td>
<td>X</td>
<td>Local</td>
<td>All</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>[NSM05]</td>
<td>✓</td>
<td>All</td>
<td>All</td>
<td>$O(</td>
</tr>
<tr>
<td>[LSMP07]</td>
<td>X</td>
<td>Local</td>
<td>All</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>[MW08]</td>
<td>✓</td>
<td>All</td>
<td>All</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>[Ver08]</td>
<td>✓</td>
<td>All</td>
<td>All</td>
<td>$O(</td>
</tr>
<tr>
<td>[Tia12]</td>
<td>✓</td>
<td>Local</td>
<td>One</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Our FE-MDVS</td>
<td>✓</td>
<td>Local</td>
<td>any subset</td>
<td>$O(</td>
</tr>
<tr>
<td>Our PS-MDVS</td>
<td>X</td>
<td>Local</td>
<td>any subset</td>
<td>$O(</td>
</tr>
</tbody>
</table>
Strong Unforgeability

Previous schemes: All verifiers take part in simulation $\Rightarrow$ Forgery
Our work: any subset $C$ simulates, remaining honest verifiers reject
## MDVS comparison

<table>
<thead>
<tr>
<th>Schemes</th>
<th>PSI</th>
<th>Verification</th>
<th>Simulation</th>
<th>Signature Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>[JSI96]</td>
<td>✗</td>
<td>Local</td>
<td>All</td>
<td>(O(1))</td>
</tr>
<tr>
<td>[NSM05]</td>
<td>✓</td>
<td>All</td>
<td>All</td>
<td>(O(</td>
</tr>
<tr>
<td>[LSMP07]</td>
<td>✗</td>
<td>Local</td>
<td>All</td>
<td>(O(1))</td>
</tr>
<tr>
<td>[MW08]</td>
<td>✓</td>
<td>All</td>
<td>All</td>
<td>(O(1))</td>
</tr>
<tr>
<td>[Ver08]</td>
<td>✓</td>
<td>All</td>
<td>All</td>
<td>(O(</td>
</tr>
<tr>
<td>[Tia12]</td>
<td>✓</td>
<td>Local</td>
<td>One</td>
<td>(O(1))</td>
</tr>
<tr>
<td>Our FE-MDVS</td>
<td>✓</td>
<td>Local</td>
<td>any subset</td>
<td>(O(</td>
</tr>
<tr>
<td>Our PS-MDVS</td>
<td>✗</td>
<td>Local</td>
<td>any subset</td>
<td>(O(</td>
</tr>
</tbody>
</table>
FE Construction
FE-MDVS: Building Blocks

Σ

vk

sk

FE

pk

dk_f
**FE-MDVS: Building Blocks**

\[ \sum \]

\[ \text{vk} \quad \text{sk} \]

\[ m \quad + \quad \text{sk} \quad = \quad \text{sk} \]

\[ m \quad + \quad \text{vk} \quad = \quad \text{FE} \]

\[ \text{pk} \quad \text{dk}_f \]
FE-MDVS: Building Blocks

\[ \Sigma \]

\( vk \)  

\( sk \)

\[ \text{FE} \]

\( pk \)  

\( dk_f \)

\[ + \]

\[ = \]

\[ = f( ) \]
**FE-MDVS: Building Blocks**

\[ \sum \]

\[ vk \quad sk \]

\[ m + sk = \]

\[ + vk = \]

\[ \checkmark / \times \]

\[ \sum + \]

\[ pk \]

\[ + dk_f = \]

\[ f( ) \]
Idea

Sign

S

skS

vkS
Idea

Sign $s$ + $sk_s = $
Idea

Sign

$sk_s$

+ pk =

35
Idea

Sign

\[ S \]

\[ sk_s \]

\[ vk_s \]

Verify

\[ dk_i \]

\[ V_i \]
Idea

Sign

Verify

\( s \), \( sk_s \), \( dv_k \), \( vt_s \), \( vt_i \)
Idea

Verify

\[ S \]

\[ dk_i, \quad vk, \quad vi, \quad vk_s, \quad vk_i \]
Idea

Verify \( f_i \) checks that:

\[ dk_i + vk_{S_i} + vk_{i} = Vi \]
Idea

$S$ checks that:

$\text{Verify}$

$\text{f}_i \text{ checks that:}$

$\text{Verify}$
Idea

Verify

$S$

$dk_i$

$f_i$ checks that:

$vk_S$

$or$

$vk_i$

$= \checkmark$

$= \checkmark$

$V_i$
Colluding Verifiers

\[ \text{dk}_j + \text{vk}_S = \text{vk}_j \]

or

\[ \text{dk}_i + \text{vk}_S = \text{vk}_i \]
Colluding Verifiers

S + \{ v_{k_S}, v_{k_j} \} = \checkmark

S + \{ v_{k_S}, v_{k_i} \} = \times
FE-MDVS

Sign

$sk_s$
Sign

sk_
FE-MDVS
FE-MDVS

Sign

$sk_s$
FE-MDVS

Sign

$\text{sk}_S$

+ $\text{pk}$ =

Treasure chest
There exists $S$ such that:

$S$ FE-MDVS checks that:

$dk_i$ or $f_i$ checks that:

$vk_S$ or $vk_i$ = ✓

Verify $V_i$
Source Hiding
Simulation

sk₁, sk₂, sk₃, sk₄
Source Hiding

Simulation
Source Hiding

Simulation
Source Hiding
Source Hiding
Source Hiding

Simulated signature looks like one from signer $S$
Verifies under secret keys $d_{k_i}$ of designated verifiers in $C$
Source Hiding

\( \text{vk}_s \)

\( \text{dk}_1 \) \( \text{dk}_2 \) \( \text{dk}_3 \) \( \text{dk}_4 \)
Source Hiding

vk_s

vk_1

dk_1

dk_2

dk_3

dk_4
Source Hiding

vk_s

vk_1  vk_2

dk_1  dk_2  dk_3  dk_4
Source Hiding

vk_s

vk_1, vk_2, vk_3

dk_1, dk_2, dk_3, dk_4
Source Hiding

vk_s

vk_1  vk_2  vk_3  vk_4
dk_1  dk_2  dk_3  dk_4
Source Hiding
Unforgeability

No honest designated verifier will believe the simulated signature is from $S$. 
Unforgeability
Unforgeability

A simulated signature is not a forgery. The honest designated verifiers will reject it.
**FE-MDVS for short**

**Sign:**

\[ \text{Sign: } \{ s_k_s \} + \text{pk} = \text{vk} \]

**Simulate:**

\[ \text{Simulate: } \{ s_k_i \} + \text{pk} = \text{vk} \]

**Verify:**

\[ \text{Verify: } d_{k_i} \text{ Checks} + \{ v_k_i \} = \text{vk} \]

or

\[ \text{or } = \checkmark \]
PS
DVS
Construction
**DVS**: Designated Verifier Signature

**Sign**

\[ m + \text{ssk} + \text{vpk} = \text{Sign} \]

**Verify**

\[ \text{Verify} + \text{vsk} = \checkmark / \times \]
MDVS from DVS

\[ \text{Sign}_{sk_i} \left[ \text{vpk}_j \right]_{j \in D} \]

DVS.Sign
MDVS from DVS

\[ \text{MDVS from DVS} \]

\[ \text{Sign} \left\{ \text{vpk}_j \right\}_{j \in \mathcal{D}} \]

\[ \text{DVS.Sign} \left\{ \begin{array}{c}
\text{vpk}_1 \\
\text{vpk}_2 \\
\vdots \\
\text{vpk}_7
\end{array} \right\}_{j \in \mathcal{D}} \]
Consistency?

Prove consistency

Sign $\text{ssk}_i$ \{ $\text{vpk}_j$ \} $j \in \mathcal{D}$

DVS.Verify $\text{vsk}_2$

$V_2$
Provably Simulatable DVS

spk  ssk  π  vpk  vsk
Provably Simulatable DVS

Sign & Prove

Validate

Simulate & Prove

Validate

\[ \text{Sign & Prove} \quad \text{Validate} \quad \text{Simulate & Prove} \quad \text{Validate} \]
Provably Simulatable DVS

Sign & Prove

Validate

Simulate & Prove

Validate

ssk + spk = \pi

Real

Public Simulation
Features

- Sign and Verify
- Simulate: Verifiers / Anyone
- Prove: $\pi_{\text{real}}, \pi_{\text{vsim}}, \pi_{\text{psim}}$
- Validate signatures: Real / Simulated

Algorithms

- Setup & KeyGen
- Sign & Verify
- PubSigSim / VerSigSim
- PubSigVal / VerSigVal / RealVal
**PSDVS: Sign & Simulate**

- **Sign**
  - ssk

- **VerSigSim**
  - vsk

- **PubSigSim**
  - spk
  - vpk
Verify

Sign

VerSigSim

PubSigSim

ssk

vsk

spk

vsk

vpk

spk
Validate

- RealVal
- VerSigVal
- PubSigVal
MDVS from PSDVS

\[ \text{Sign} \left\{ \text{ssk}_j \right\}_{j \in \mathcal{D}} \]

all Real
Sign & Simulate

Sign: $ssk_i \{ \text{vpk}_j \}_{j \in \mathcal{D}}$

Sim: $spk_i \{ \text{vsk}_j \}_{j \in \mathcal{C}}$

VerSigSim: all Real
Sign & Simulate

Sign $ssk_i \{ \text{vpk}_j \} \mid j \in D$

Sim $spk_i \{ \text{vsk}_j \} \mid j \in C$

VerSigSim $\text{all Real} \mid j \in C$
Sign & Simulate

Sign: $\{ ssk_j \}_{j \in \mathcal{D}}$

Sim: $\{ spk_i \}_{j \in \mathcal{C}}, \{ vsk_j \}_{j \in \mathcal{C}}$

VerSigSim: $\{ \pi \}_{j \in \mathcal{C}}$

all Real
Sign & Simulate

Sign \( \{ \text{ssk}_j \} \) \( j \in \mathcal{D} \)

Sim \( \{ \text{spk}_j \} \) \( j \in \mathcal{C} \)

all Real

PubSigSim \( j \in \mathcal{D} \setminus \mathcal{C} \)
Sign & Simulate

Sign $\{ssk_j, vpk_j\}_{j \in \mathcal{D}}$

Sim $\{spk_j, vsk_j\}_{j \in \mathcal{C}}$

all Real

PubSigSim $\{\}_{j \in \mathcal{D} \setminus \mathcal{C}}$
Sign & Simulate

Sign \( ssk_j \) \( j \in \mathcal{D} \)

Sim \( spk_j \) \( j \in \mathcal{C} \)

all Real

PubSigSim \( j \in \mathcal{D} \setminus \mathcal{C} \)
Source Hiding?

\[ \text{Sign} \left\{ \text{ssk}_i \right\}_{j \in \mathcal{D}} \]

\[ \text{Sim} \left\{ \text{spk}_i \right\}_{j \in \mathcal{C}} \]

all Real

consistency
Source Hiding?

Sign \[ \text{ssk}_i \{ \text{vpk}_j \} \mid j \in \mathcal{D} \]

Sim \[ \text{spk}_i \{ \text{vsk}_j \} \mid j \in \mathcal{C} \]

all Real

all Simulated
Source Hiding

Sign $ssk_i \{ vpk_j \}_{j \in \mathcal{D}}$

Sim $spk_i \{ vsk_j \}_{j \in \mathcal{C}}$

all Real or all Simulated
Unforgeability

\[ \text{Sign} \quad \{ \text{ssk}_i \} \quad \{ \text{vpk}_j \} \quad j \in D \]

\[ \text{Sim} \quad \{ \text{spk}_i \} \quad \{ \text{vsk}_j \} \quad j \in C \]
MDVS from PSDVS

Sign $\{\text{ssk}_i\}_{j \in \mathcal{D}}$

Sim $\{\text{spk}_i\}_{j \in \mathcal{C}}$

Sign & Verify

Prove real

VerSigSim / PubSigSim

Prove simulation
PSDV from standard primitives

Scheme 1 from generic tools:
- pseudo-random functions
- non-interactive key exchange (Diffie-Hellman)
- zk-SNARKs: non-interactive zero-knowledge proofs of knowledge.

Scheme 2 with better concrete efficiency:
- based on DDH & strong RSA
- Paillier encryption
- Secure in the random oracle model
- requires a constant number of exponentiations
New security notions

Study of properties:
- Consistency
- Source-Hiding
- Strong Unforgeability

MDVS from Provably Simulatetable DVS

MDVS from Functional Encryption
Thanks!
Any questions?
Credits

Special thanks to all those who made and released these resources for free:

- Presentation template by SlidesCarnival
- Illustrations by Disneyclips and Iconfinder