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* **$\mathfrak{D}$** *(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

1

Correctness

Any honest signature should verify for every designated verifier

2

Consistency

S cannot create an inconsistent signature:
Accepted by a verifier $\mathbf{V_1}$ and rejected by $\mathbf{V_2}$

2

Consistency

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

3

Unforgeability

An adversary cannot forge a signature to a honest verifier

Strong: Even when it corrupts part of the verifiers in $\mathcal{D}$

4

Source Hiding

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

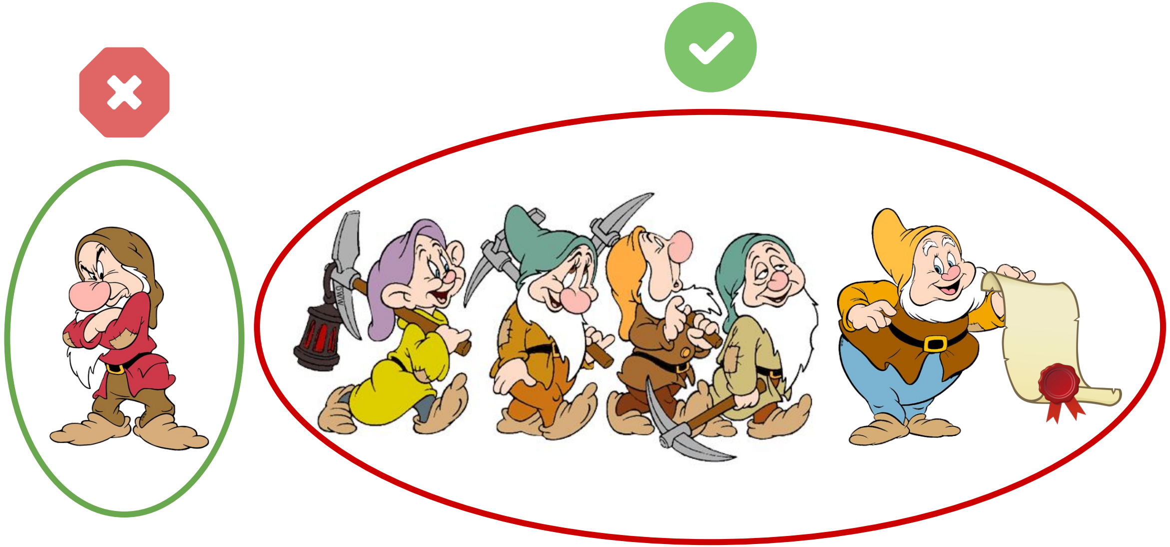
4

Source Hiding

The verifiers are able to simulate signatures

No outsider can know where a given signature came from.

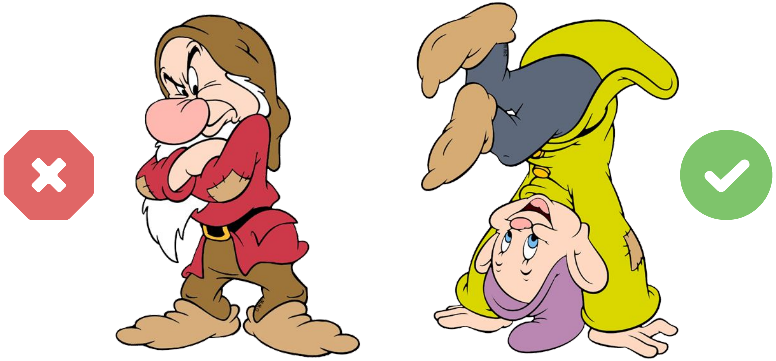
4



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’**

5

Privacy of Identity

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

?

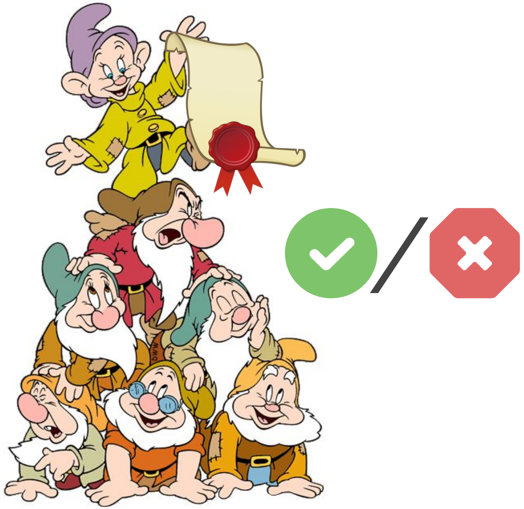
Privacy of Identity

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

MDVS comparison

Schemes	PSI	Verification	Simulation	Signature Size
[JSI96]	✗	Local	All	$\mathcal{O}(1)$
[NSM05]	✓	All	All	$\mathcal{O}(\lvert\mathcal{D}\rvert)$
[LSMP07]	✗	Local	All	$\mathcal{O}(1)$
[MW08]	✓	All	All	$\mathcal{O}(1)$
[Ver08]	✓	All	All	$\mathcal{O}(\lvert\mathcal{D}\rvert)$
[Tia12]	✓	Local	One	$\mathcal{O}(1)$
Our FE-MDVS	✓	Local	**any subset**	$\mathcal{O}(\lvert\mathcal{C}\rvert)$
Our PS-MDVS	✗	Local	**any subset**	$\mathcal{O}(\lvert\mathcal{D}\rvert)$

5



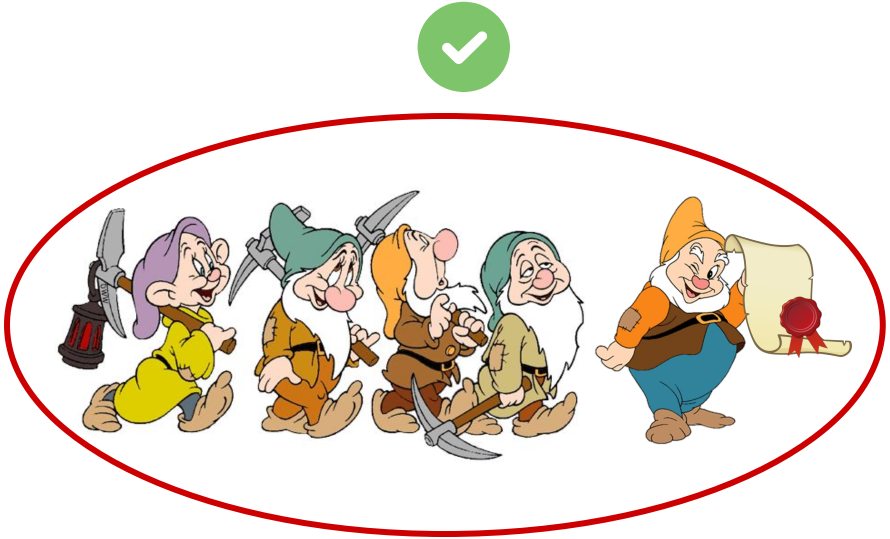
Verify: Local vs All

Local: a single designated verifier can check

All: the designated verifiers need to work together in order to verify

MDVS comparison

Schemes	PSI	Verification	Simulation	Signature Size
[JSI96]	✗	Local	All	$\mathcal{O}(1)$
[NSM05]	✓	All	All	$\mathcal{O}(\lvert\mathcal{D}\rvert)$
[LSMP07]	✗	Local	All	$\mathcal{O}(1)$
[MW08]	✓	All	All	$\mathcal{O}(1)$
[Ver08]	✓	All	All	$\mathcal{O}(\lvert\mathcal{D}\rvert)$
[Tia12]	✓	Local	One	$\mathcal{O}(1)$
Our FE-MDVS	✓	Local	**any subset**	$\mathcal{O}(\lvert\mathcal{C}\rvert)$
Our PS-MDVS	✗	Local	**any subset**	$\mathcal{O}(\lvert\mathcal{D}\rvert)$



Strong Unforgeability

Previous schemes: All verifiers take part in simulation => Forgery

Our work: any subset $\mathcal{C}$ simulates, remaining honest verifiers reject

MDVS comparison

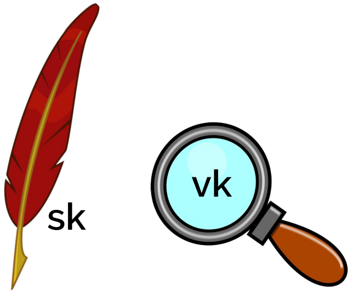
Schemes	PSI	Verification	Simulation	Signature Size
[JSI96]	✗	Local	All	$\mathcal{O}(1)$
[NSM05]	✓	All	All	$\mathcal{O}(\lvert\mathcal{D}\rvert)$
[LSMP07]	✗	Local	All	$\mathcal{O}(1)$
[MW08]	✓	All	All	$\mathcal{O}(1)$
[Ver08]	✓	All	All	$\mathcal{O}(\lvert\mathcal{D}\rvert)$
[Tia12]	✓	Local	One	$\mathcal{O}(1)$
Our FE-MDVS	✓	Local	**any subset**	$\mathcal{O}(\lvert\mathcal{C}\rvert)$
Our PS-MDVS	✗	Local	**any subset**	$\mathcal{O}(\lvert\mathcal{D}\rvert)$

optimal

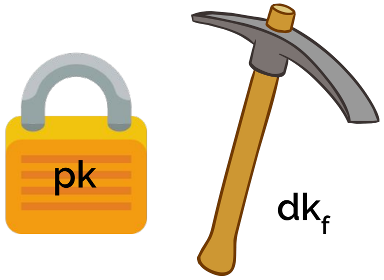
FE Construction

FE-MDVS: Building Blocks

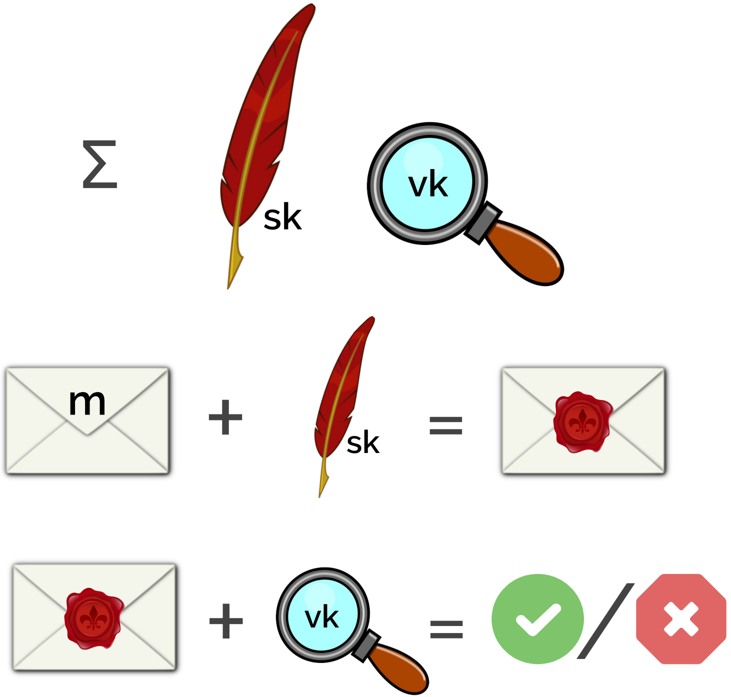
Σ



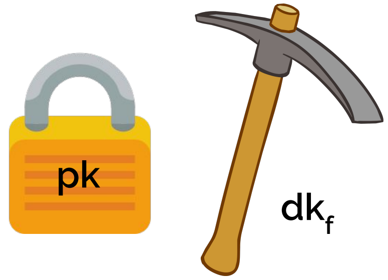
FE



FE-MDVS: Building Blocks

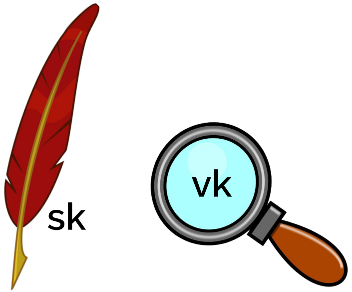


FE

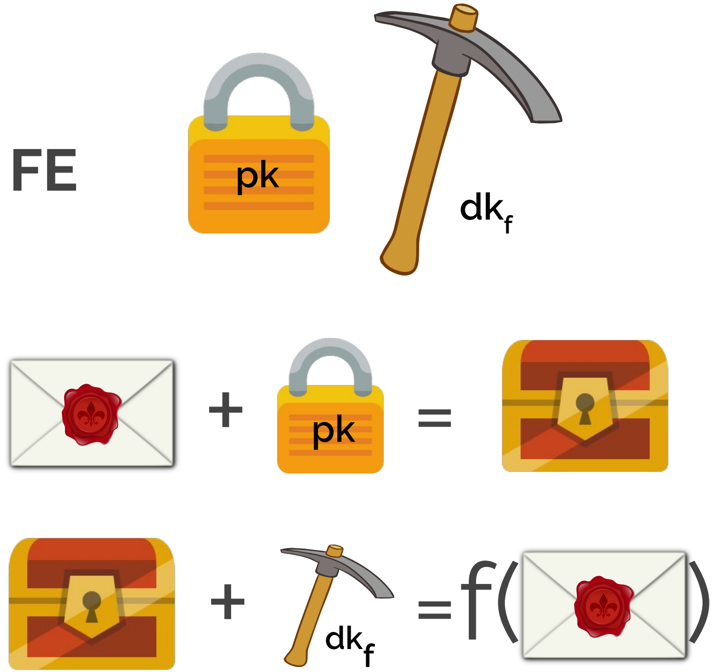


FE-MDVS: Building Blocks

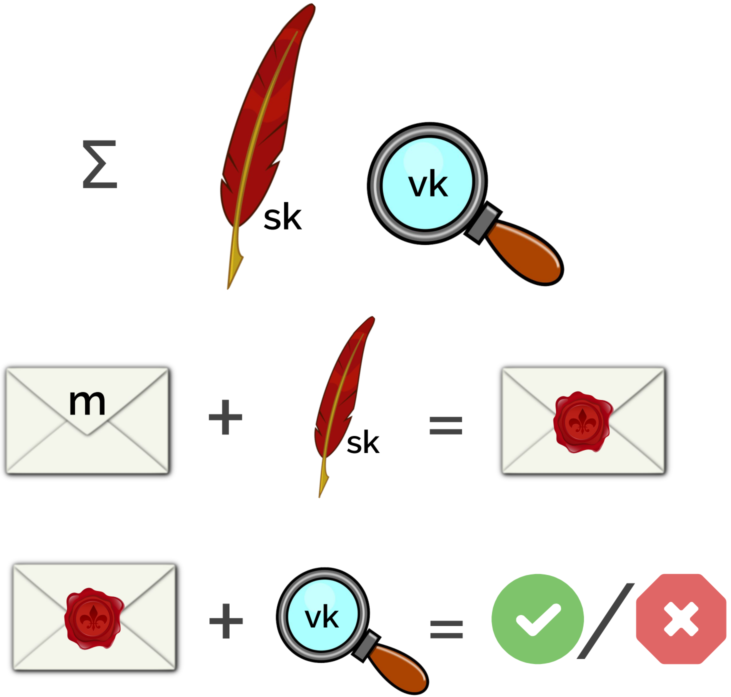
Σ

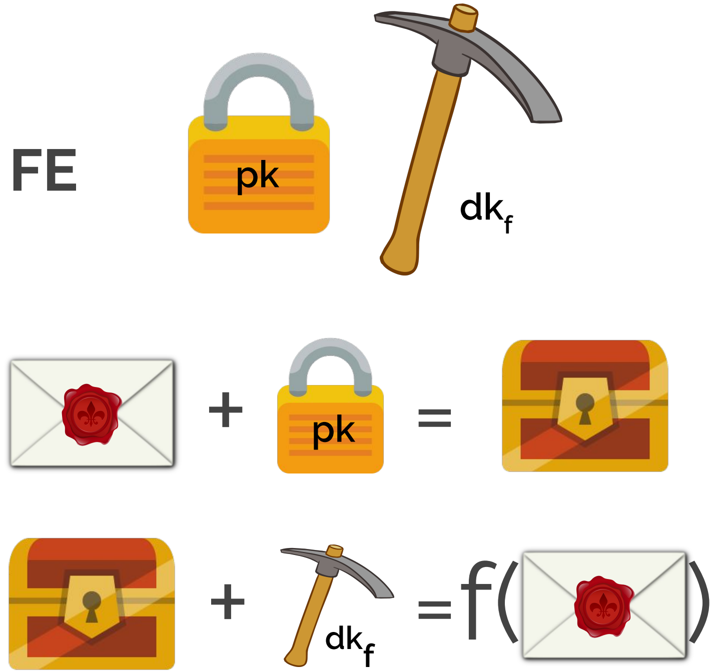


FE



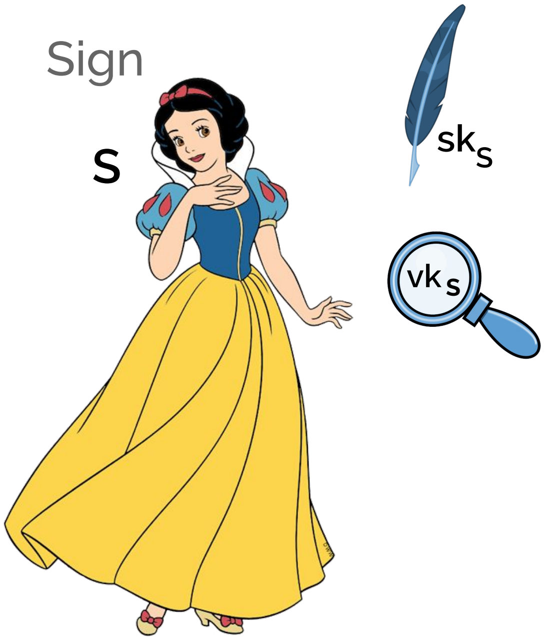
FE-MDVS: Building Blocks





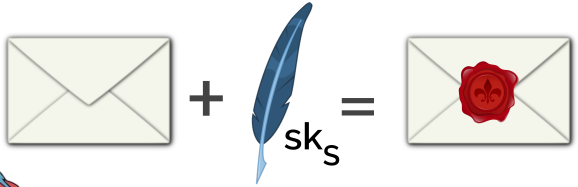
Idea

Sign



Idea

Sign



Idea

Sign

S

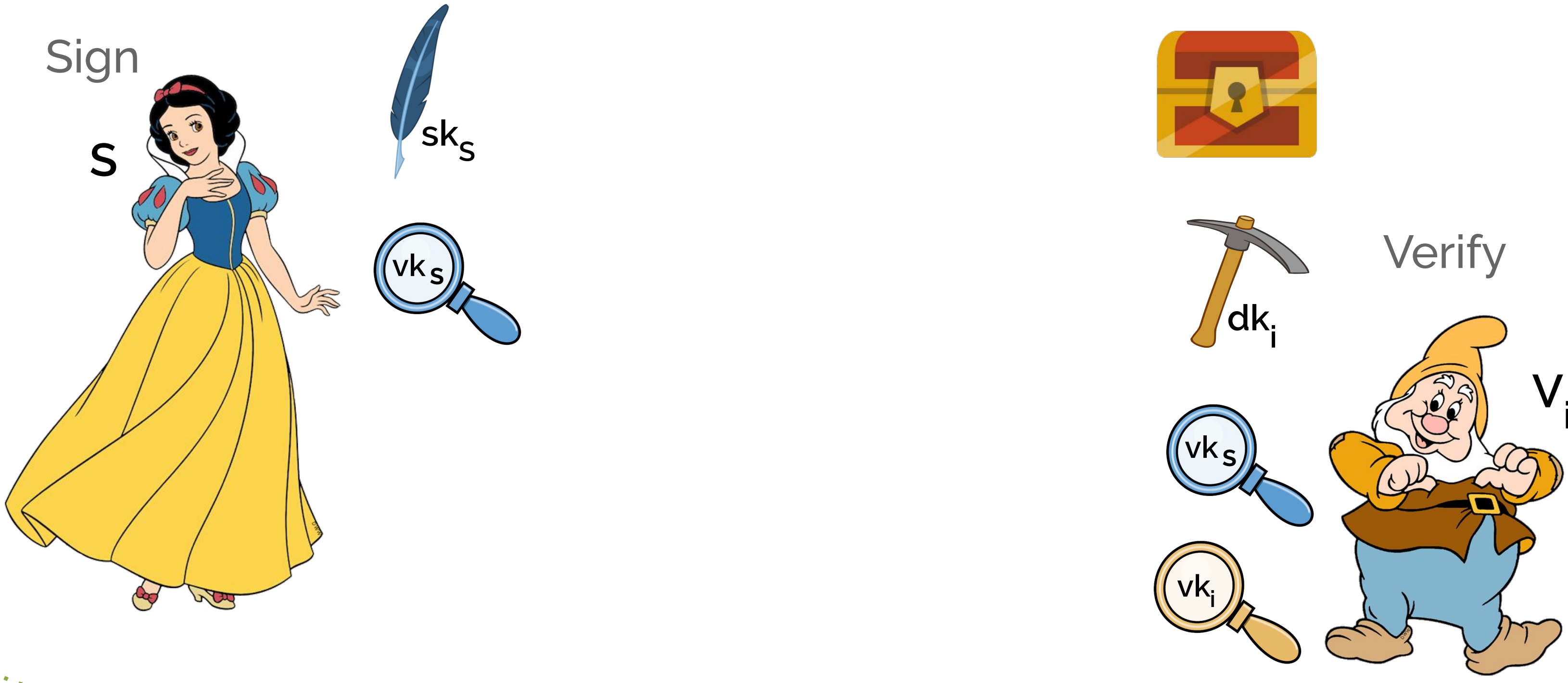
sk_S

+

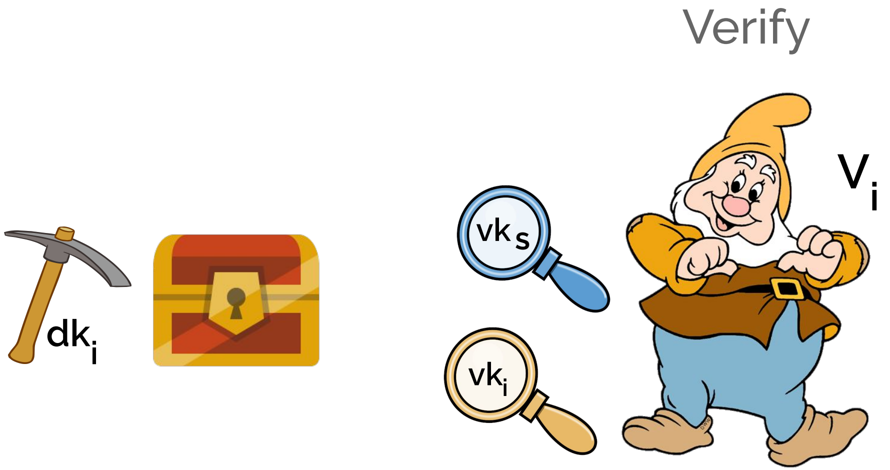
=

Idea

Idea



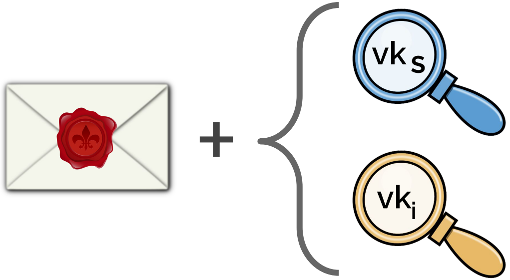
Idea



Idea

S

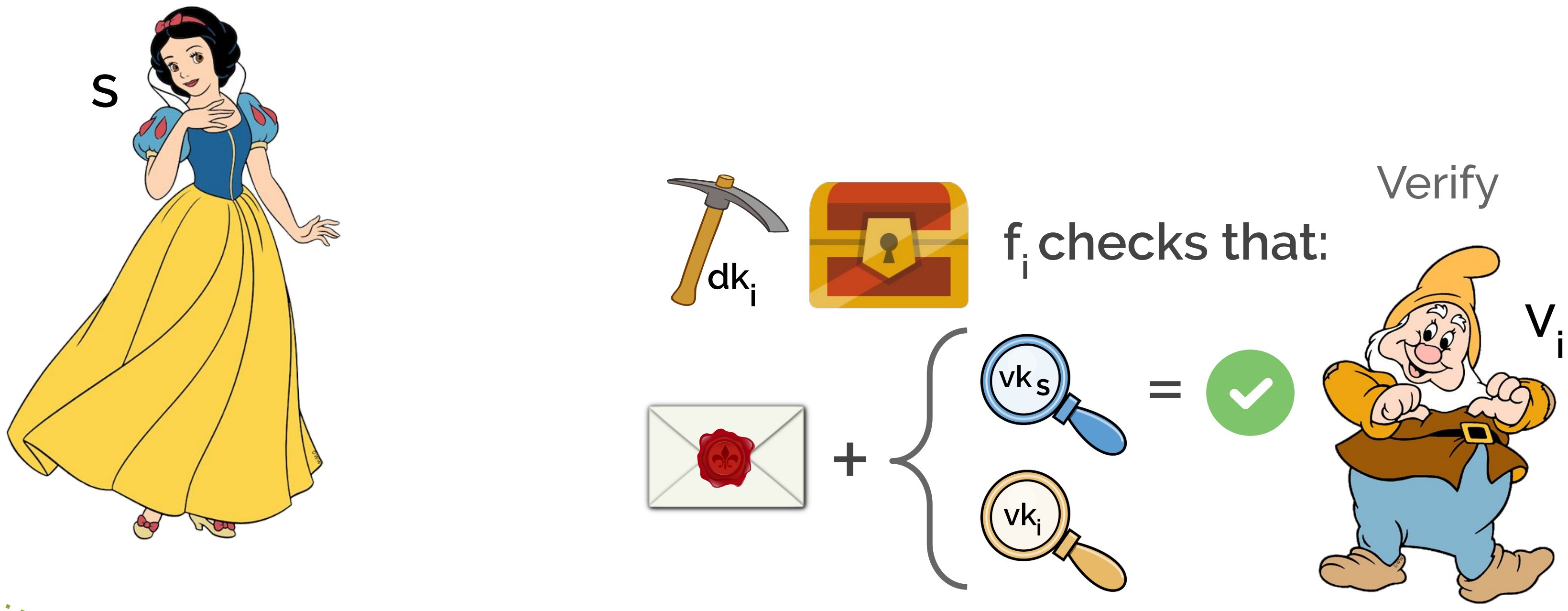
f_i checks that:



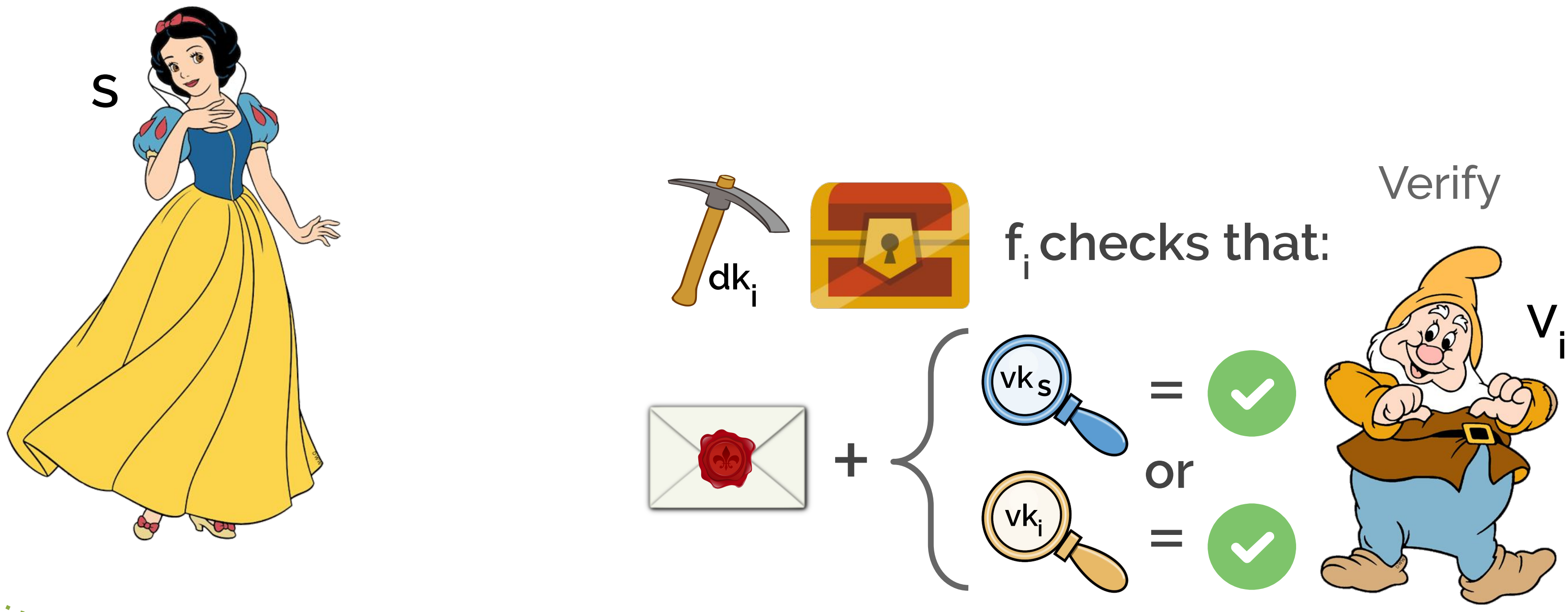
Verify

V_i

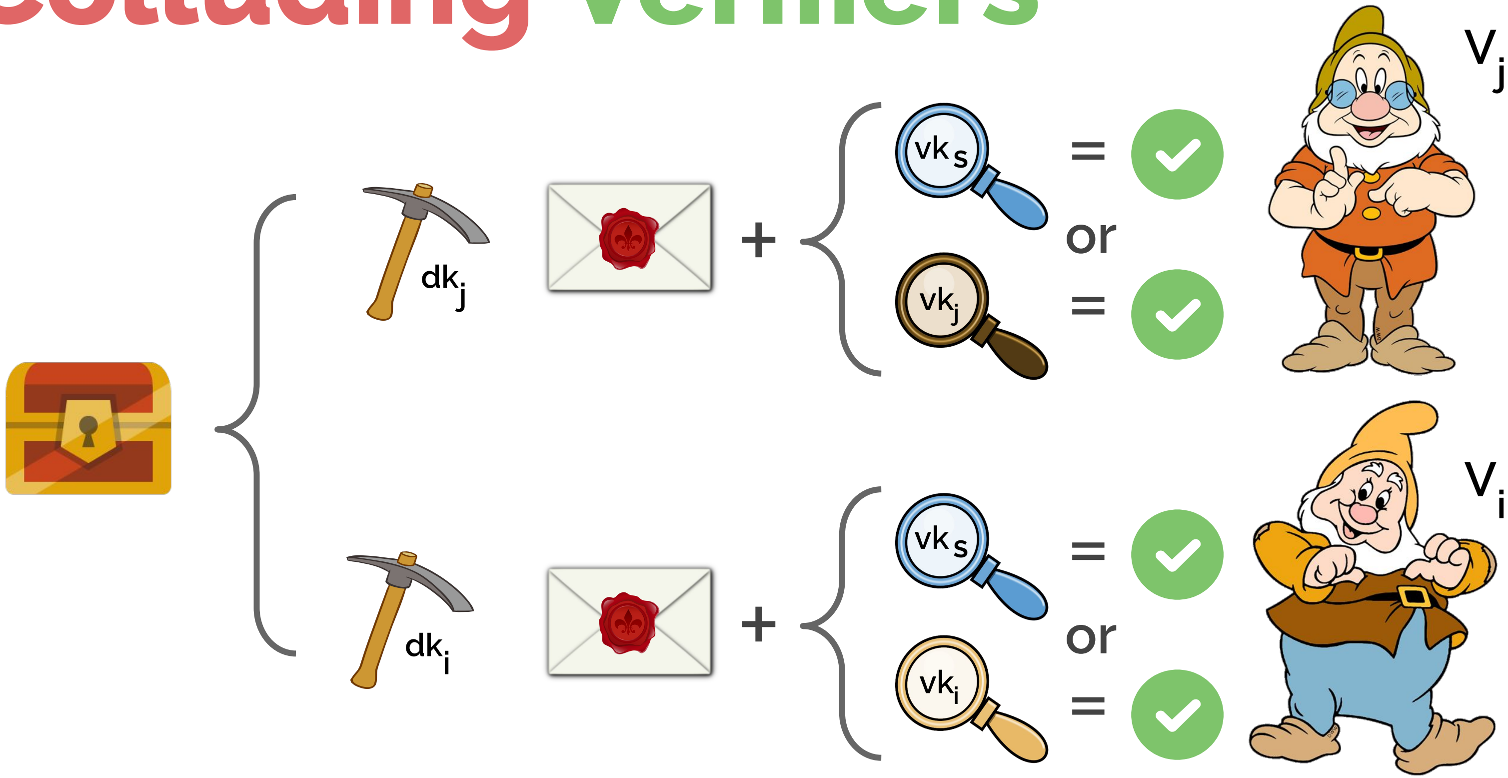
Idea



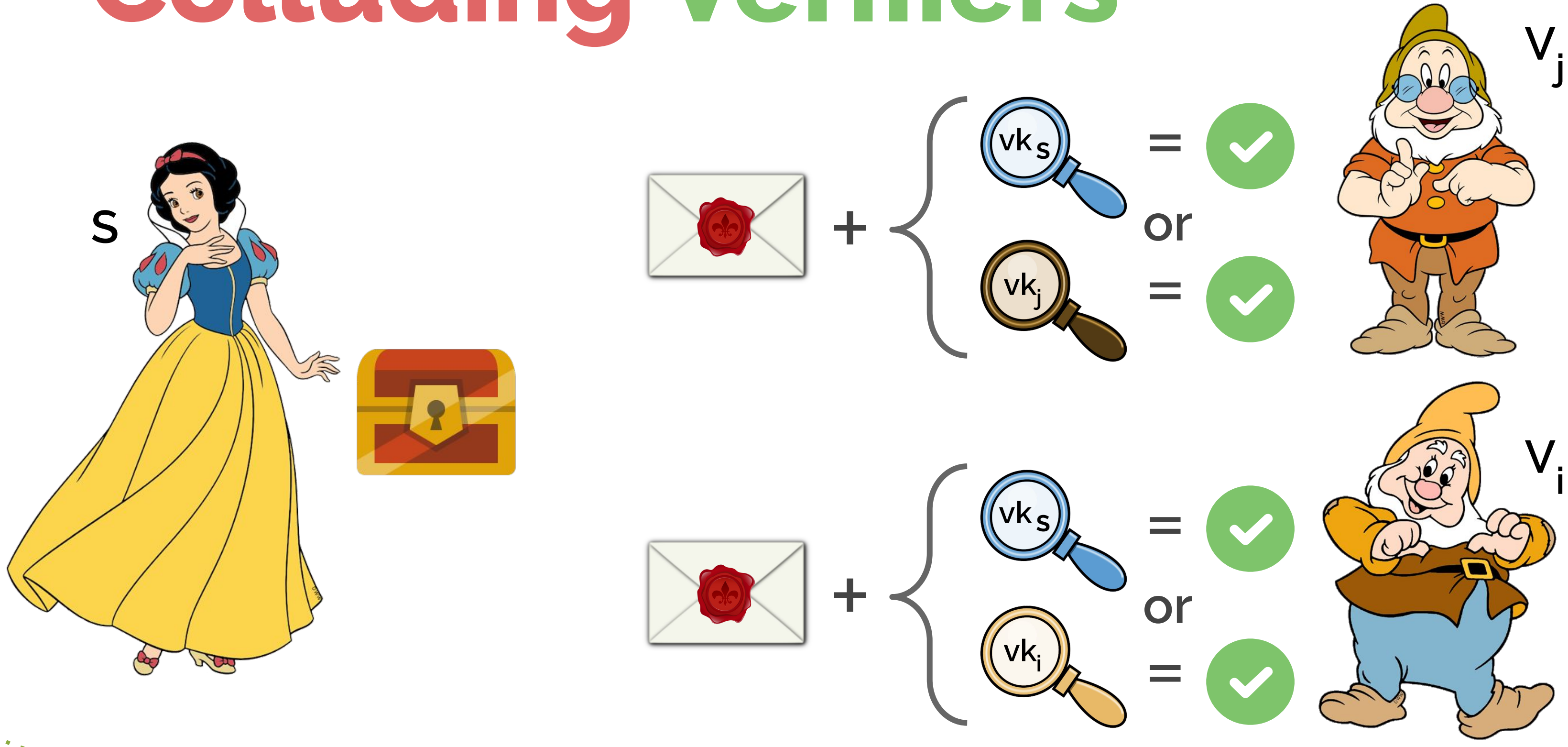
Idea



Colluding Verifiers



Colluding Verifiers

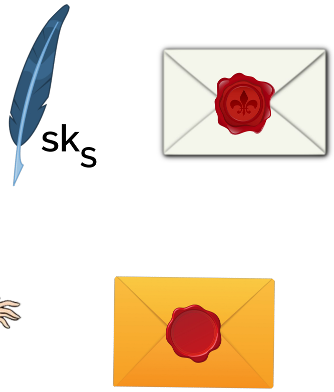


FE-MDVS

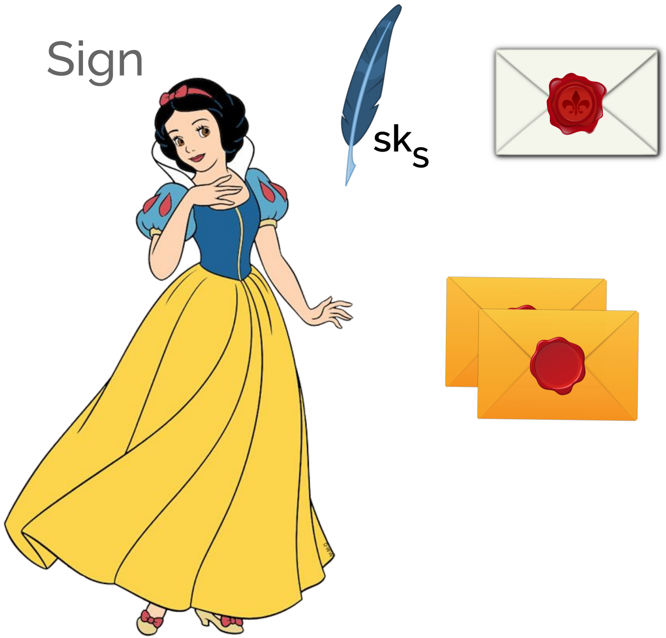
Sign

FE-MDVS

Sign

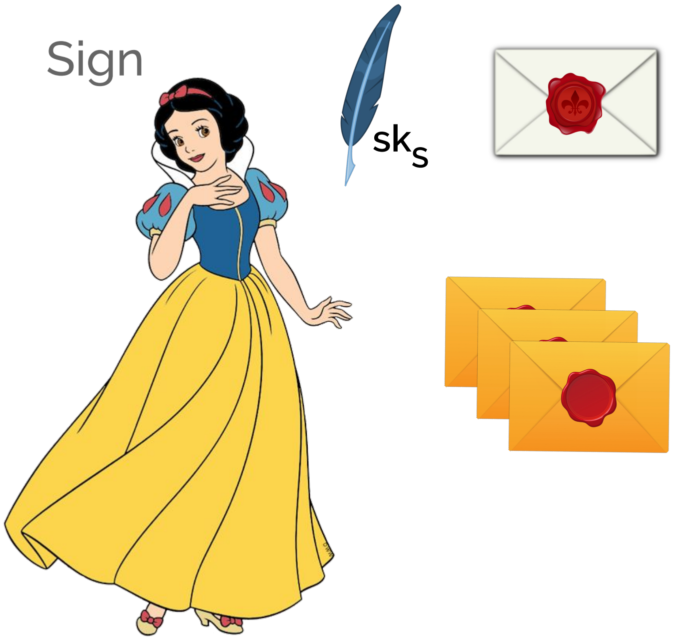


FE-MDVS

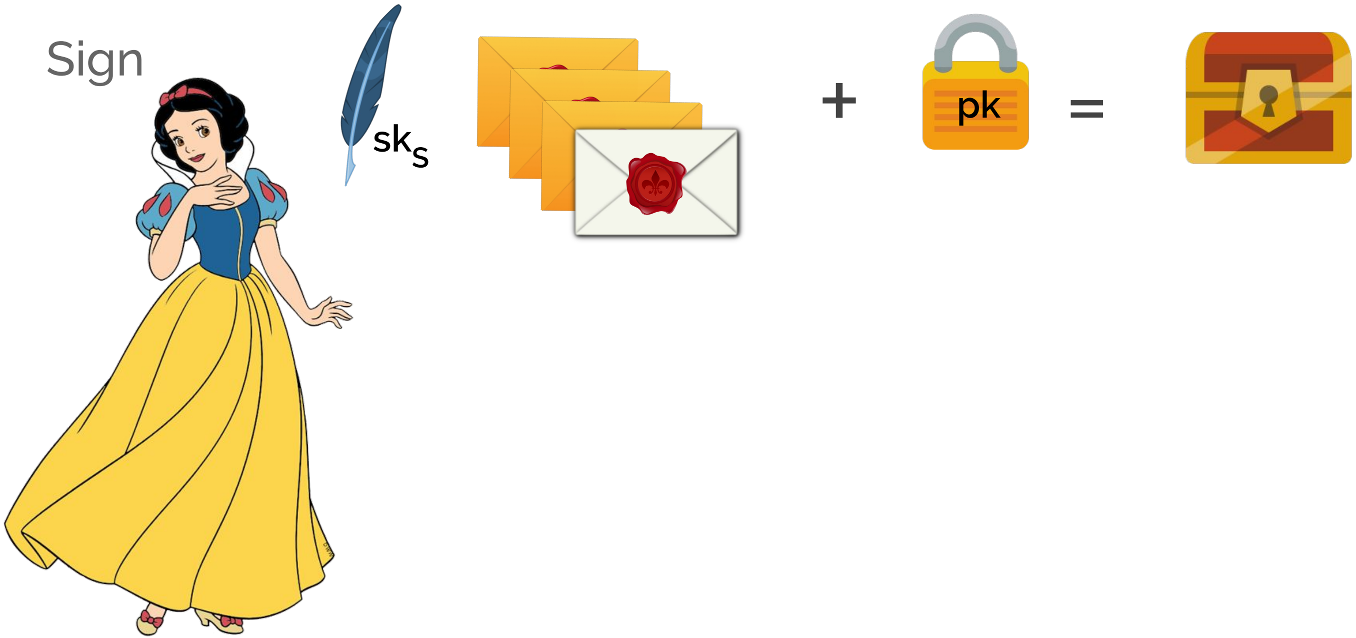


FE-MDVS

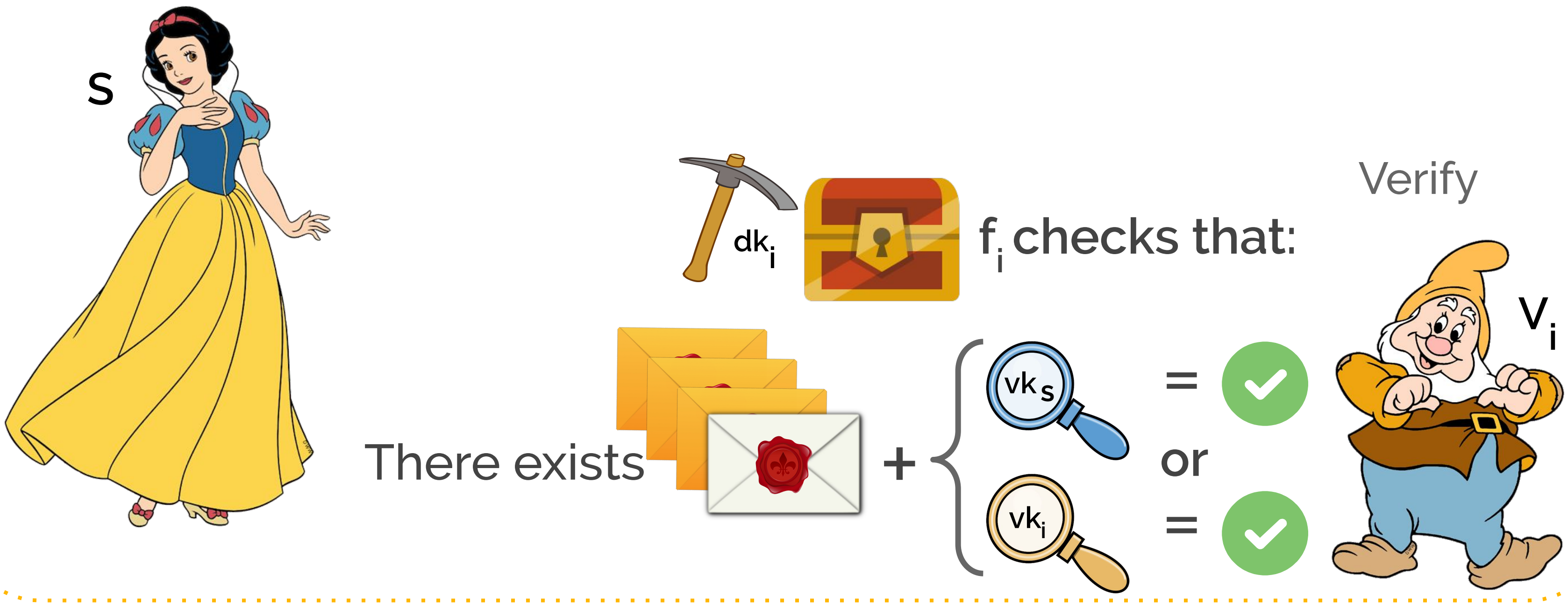
Sign



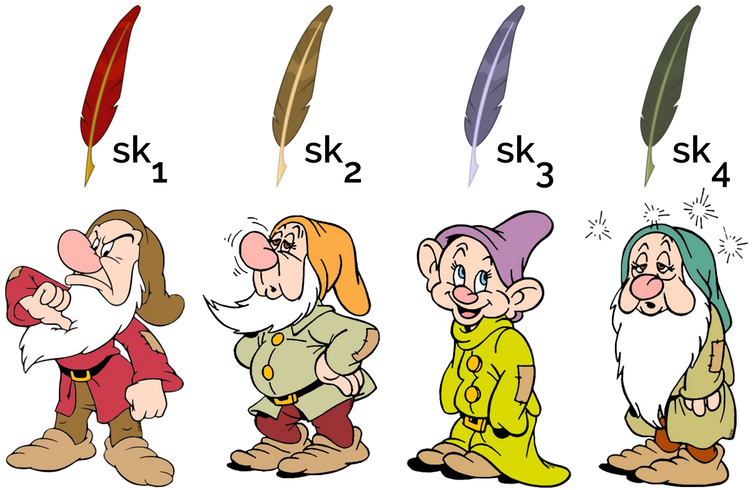
FE-MDVS



FE-MDVS

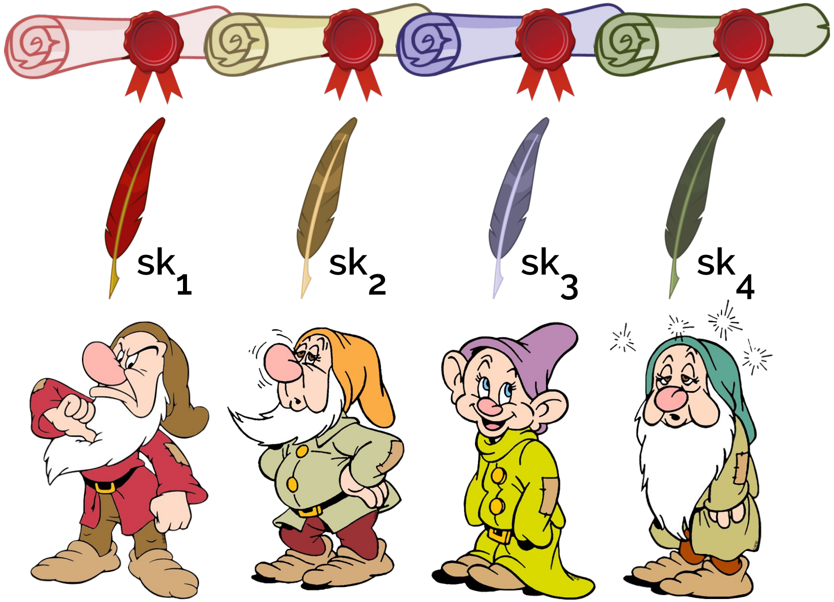


Source Hiding
Simulation

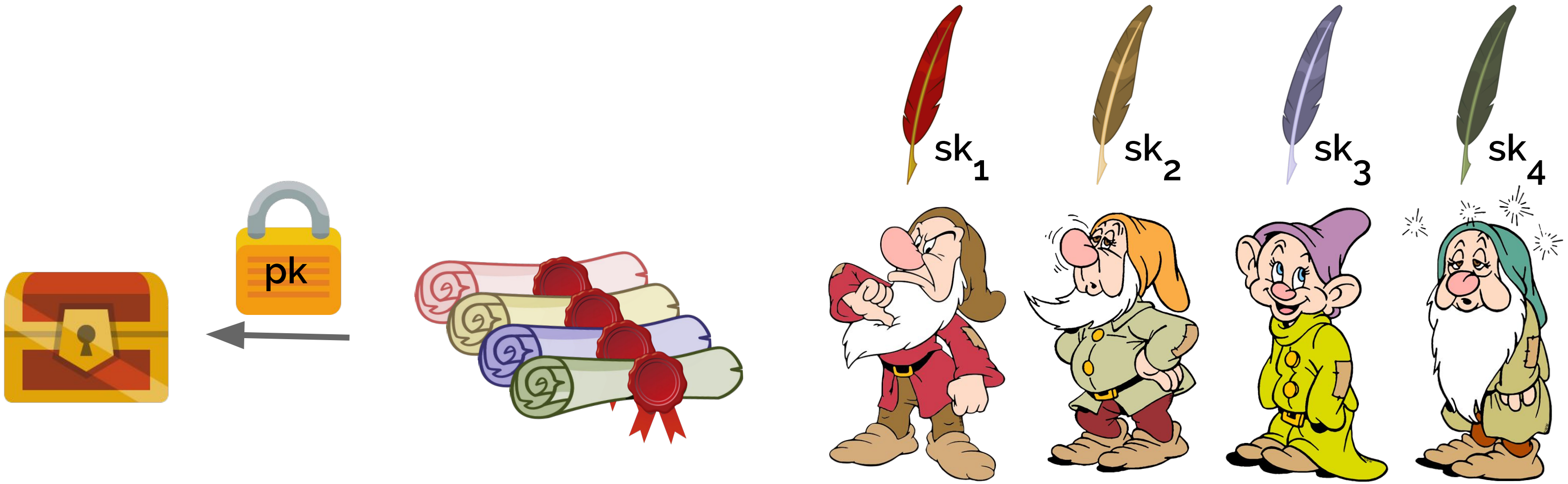


Source Hiding

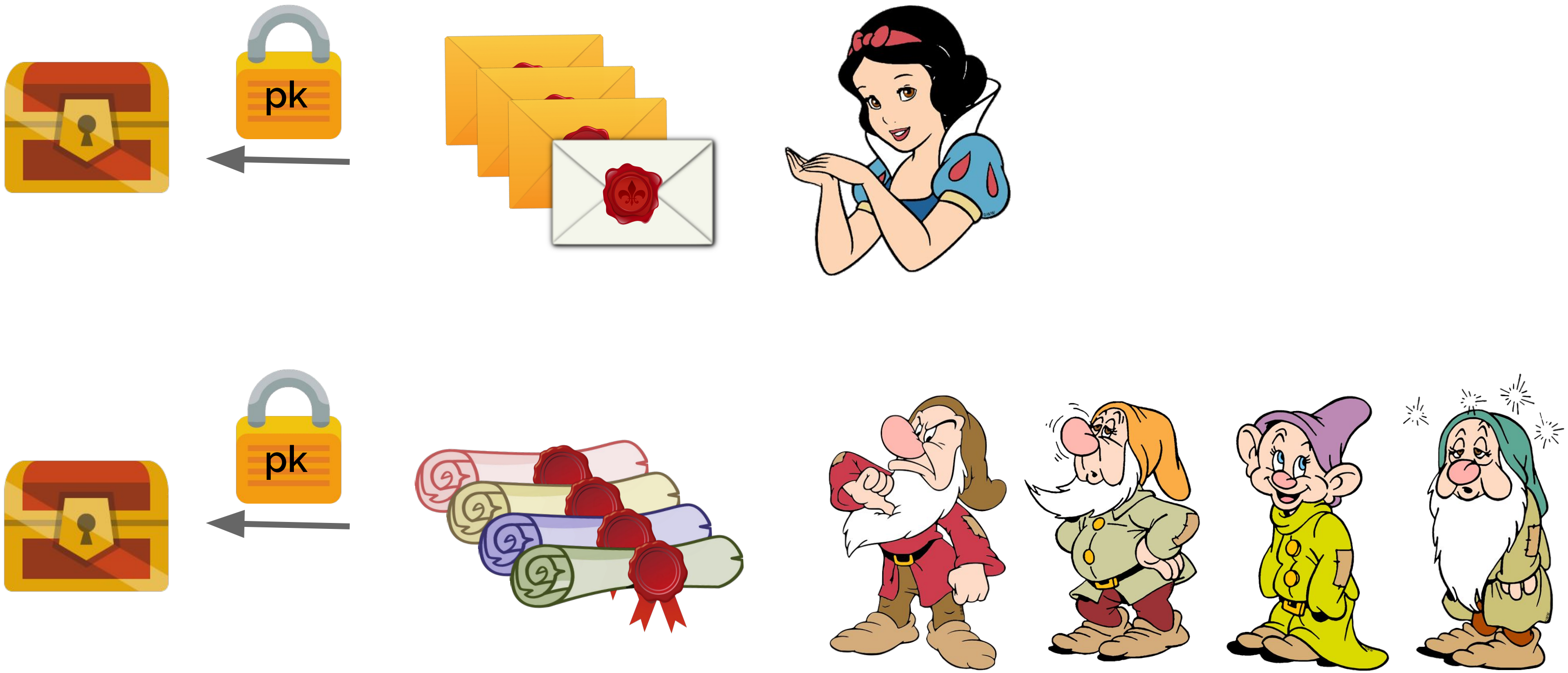
Simulation



Source Hiding
Simulation

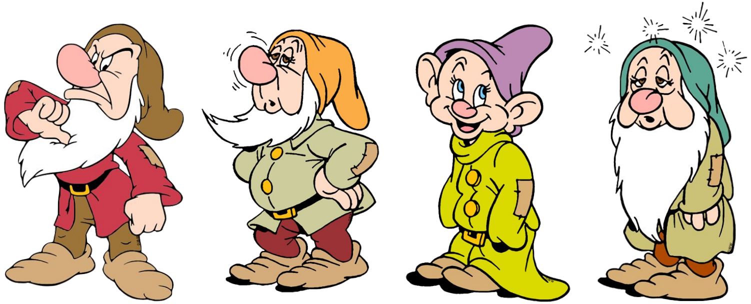


Source Hiding



Source Hiding

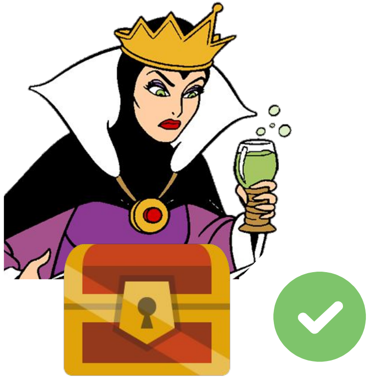
?

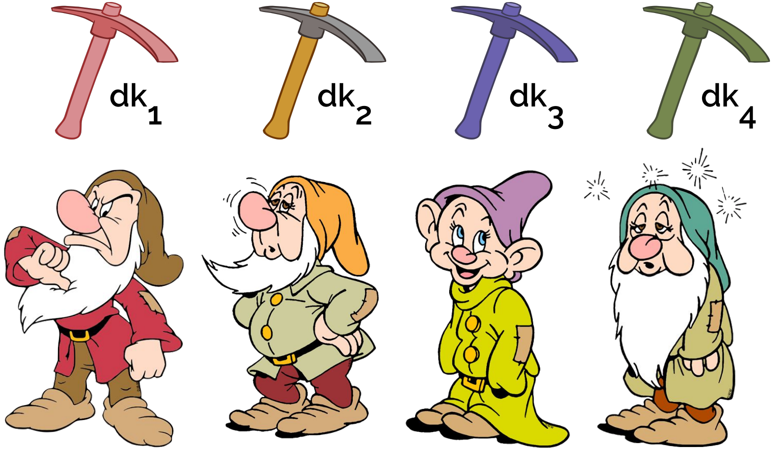


Source Hiding

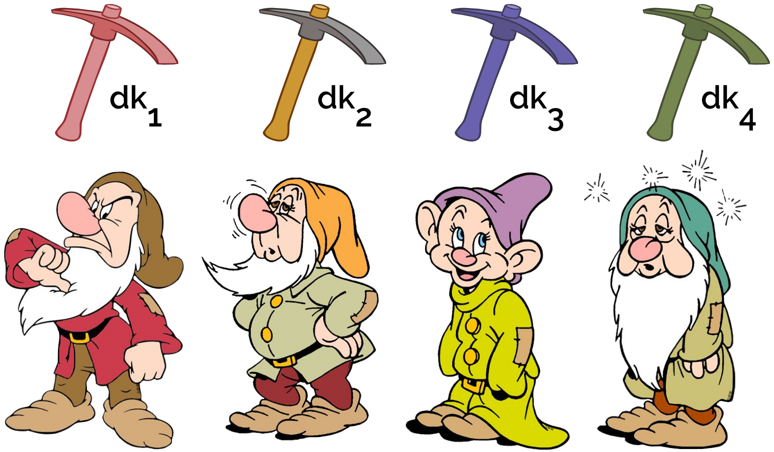
Simulated signature looks like one from signer **S**

Verifies under *secret keys* $\mathbf{dk}_i$ of designated verifiers in $\mathcal{C}$

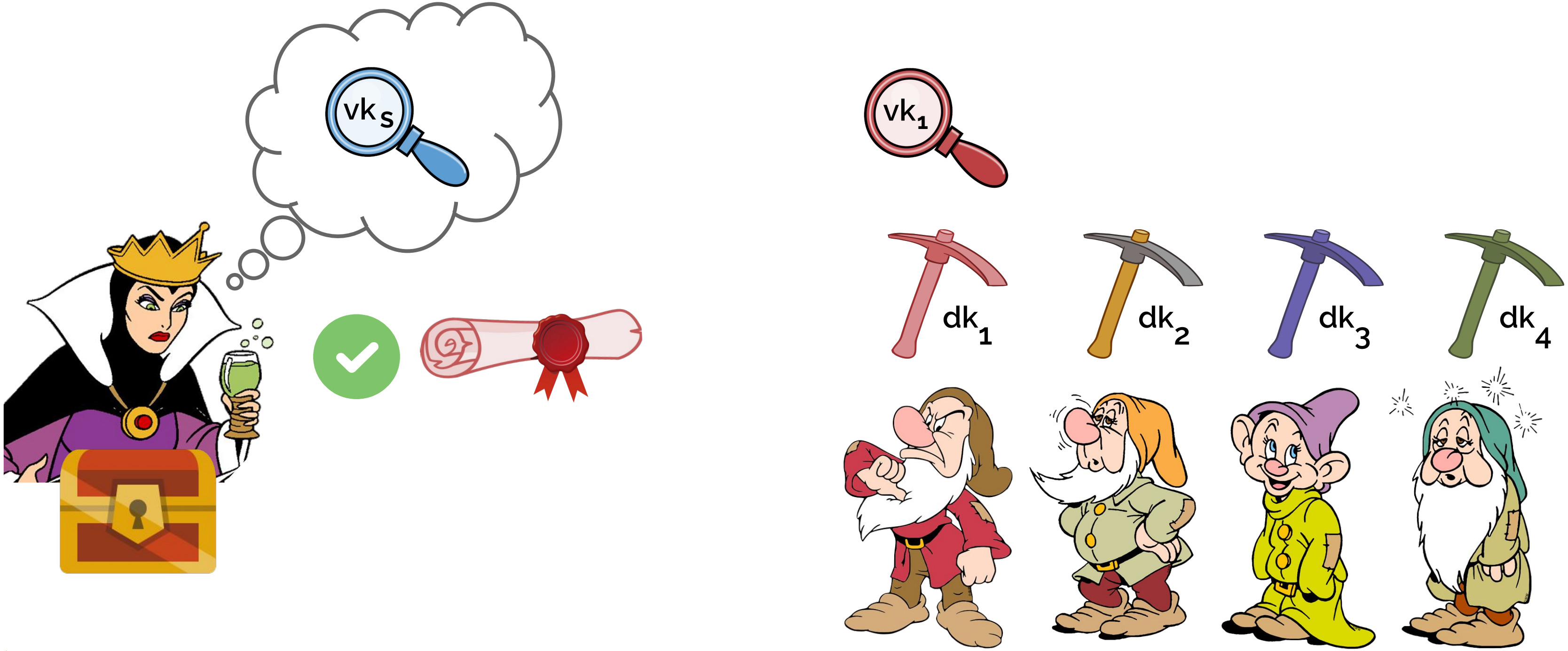




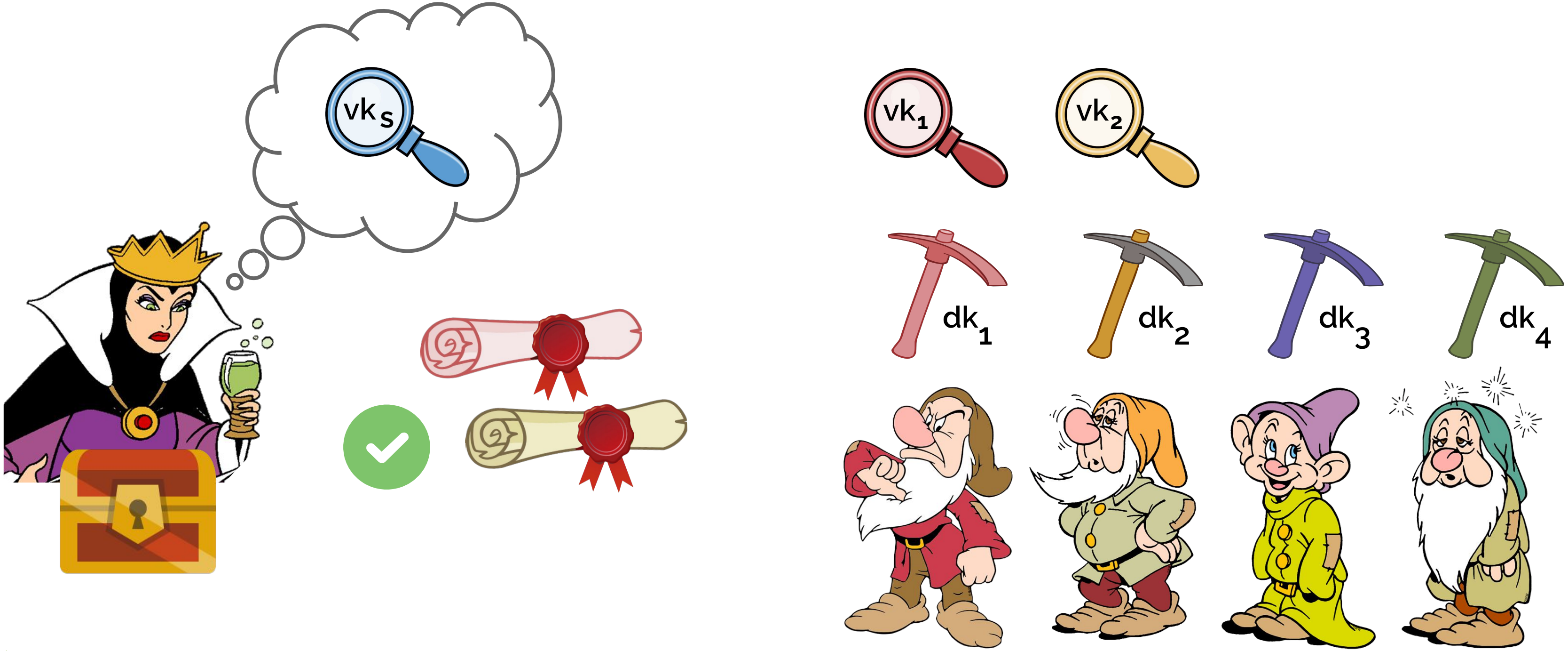
Source Hiding



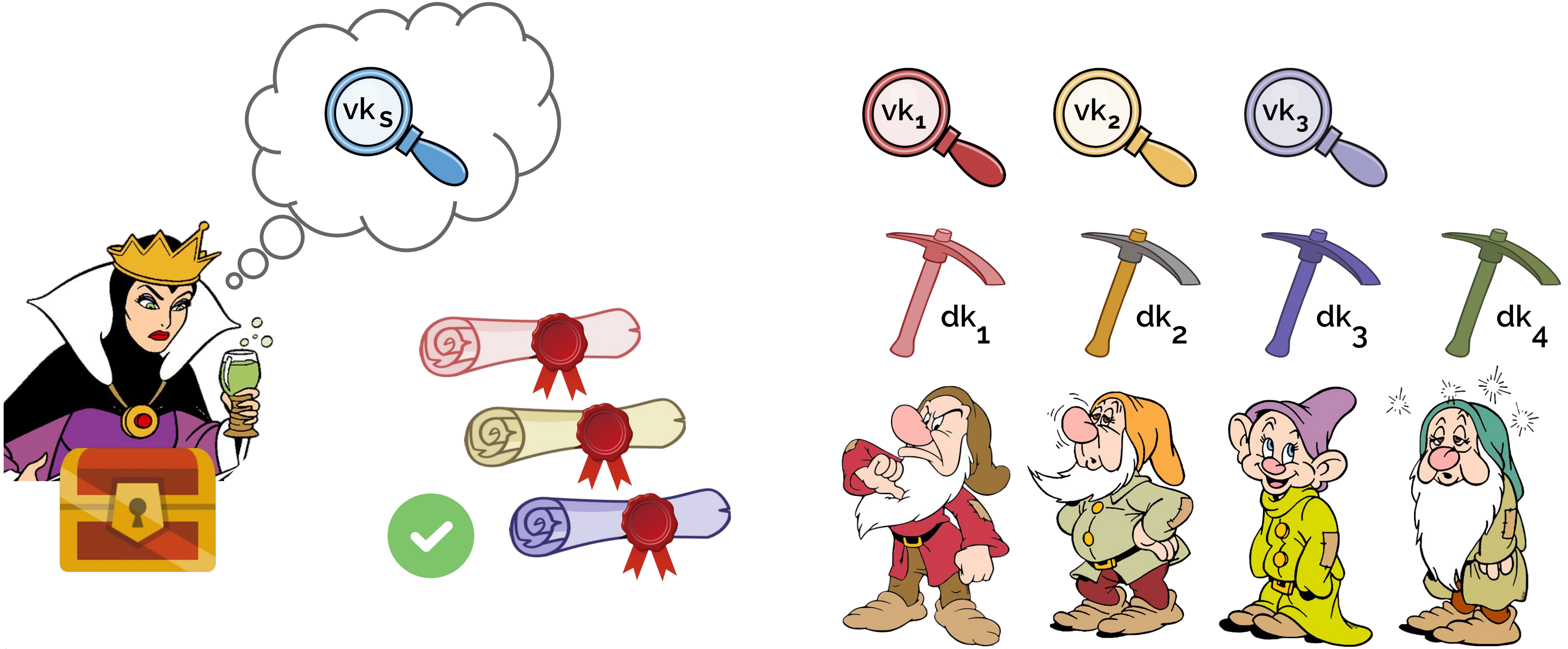
Source Hiding



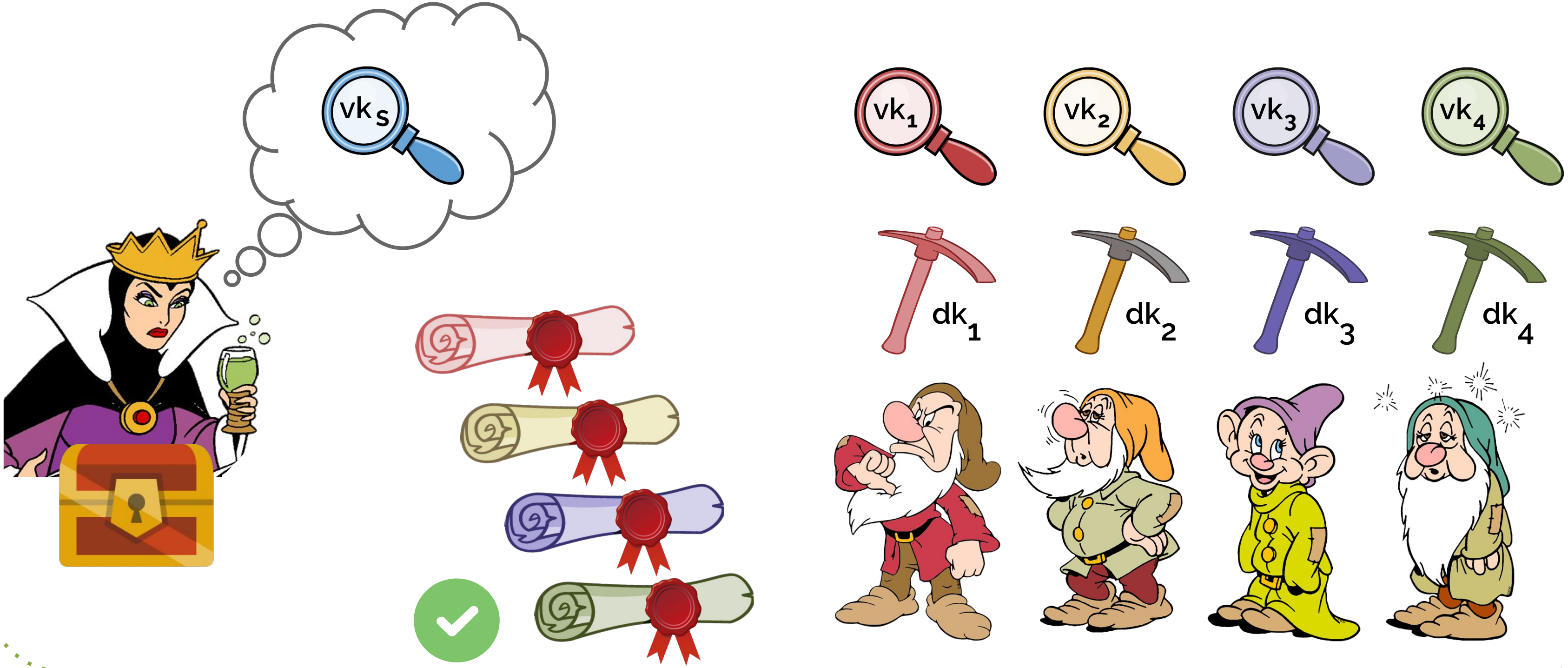
Source Hiding



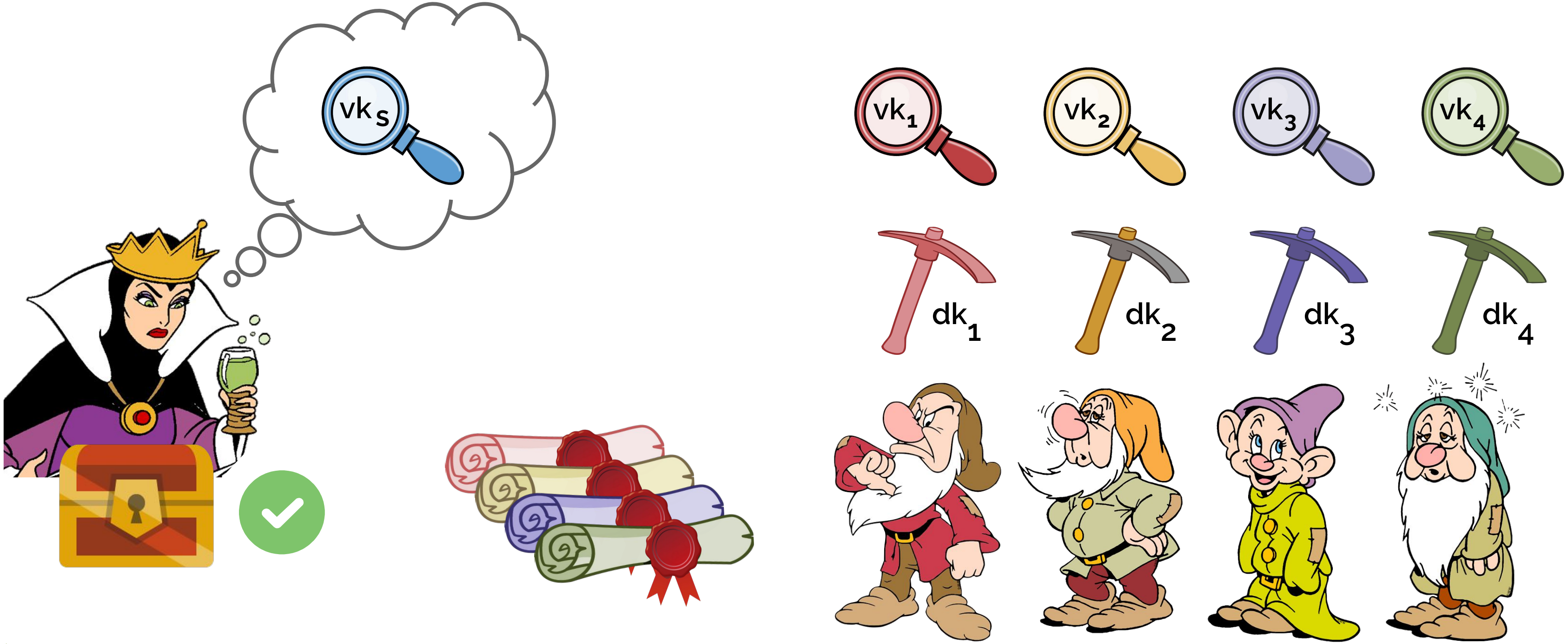
Source Hiding



Source Hiding

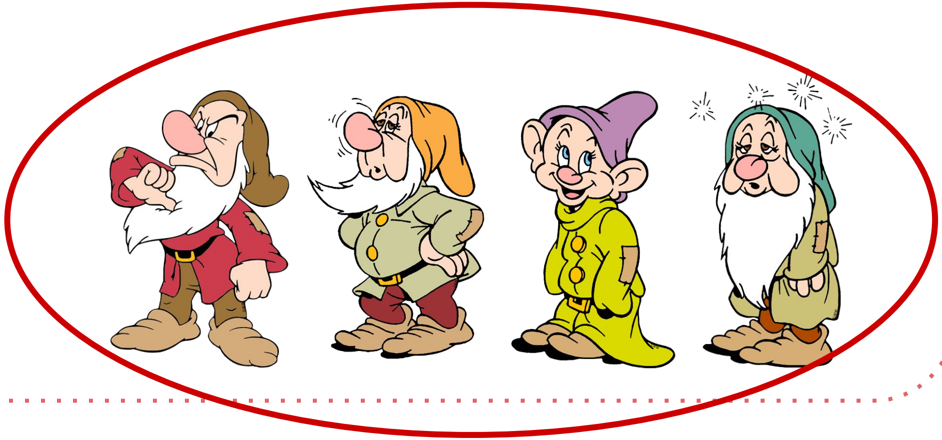


Source Hiding

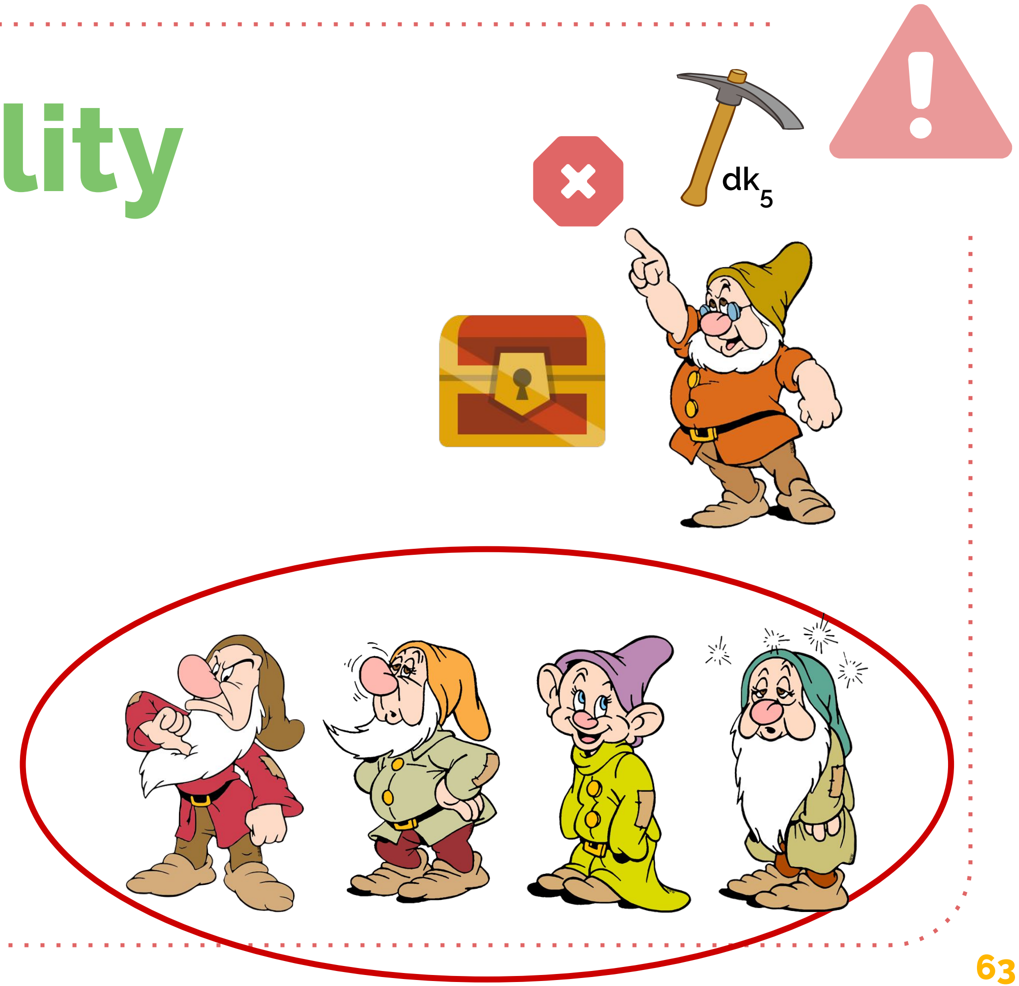


Unforgeability

No honest designated verifier will believe the simulated signature is from **S**



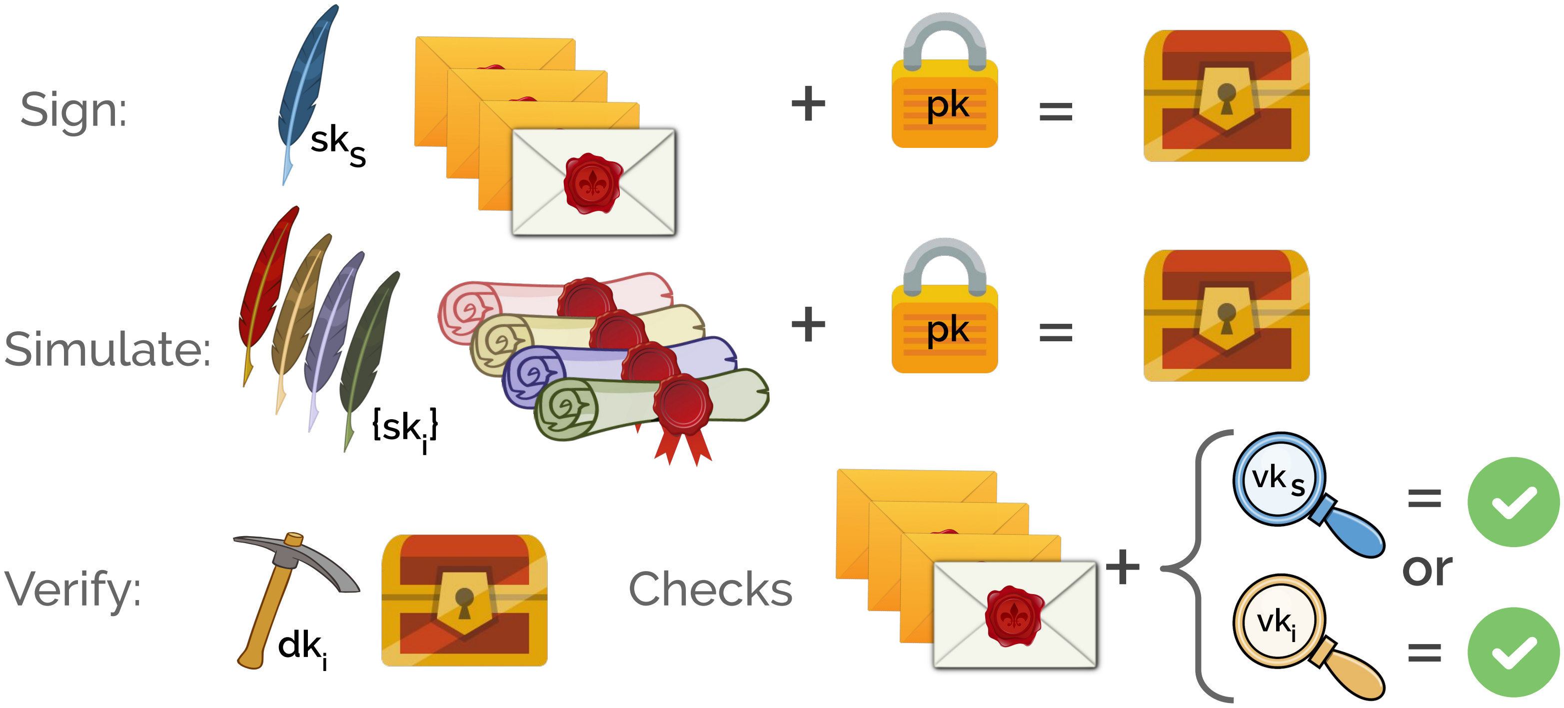
Unforgeability



Unforgeability

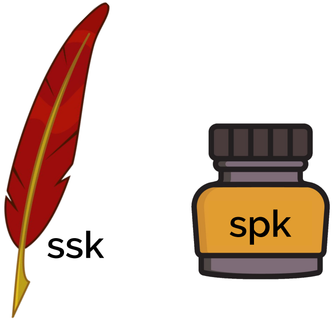
A simulated signature is not a forgery.
The honest designated verifiers will reject it.

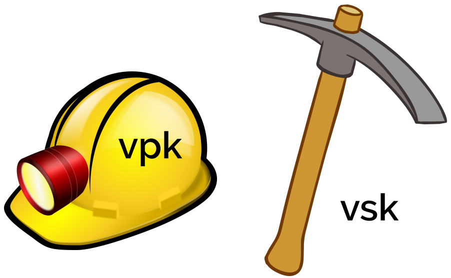
FE-MDVS for short



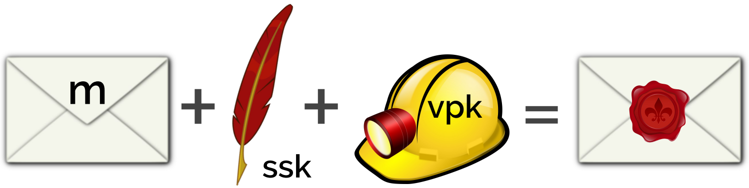
PSDVS Construction

DVS: Designated Verifier Signature





Sign



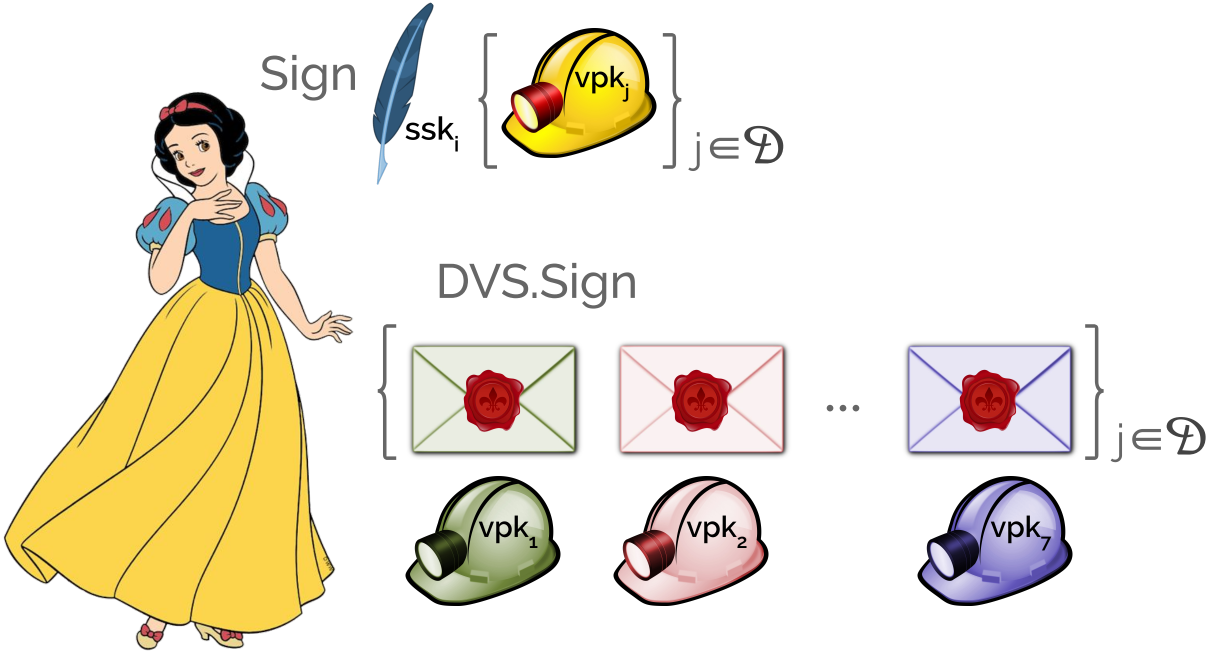
Verify

MDVS from DVS

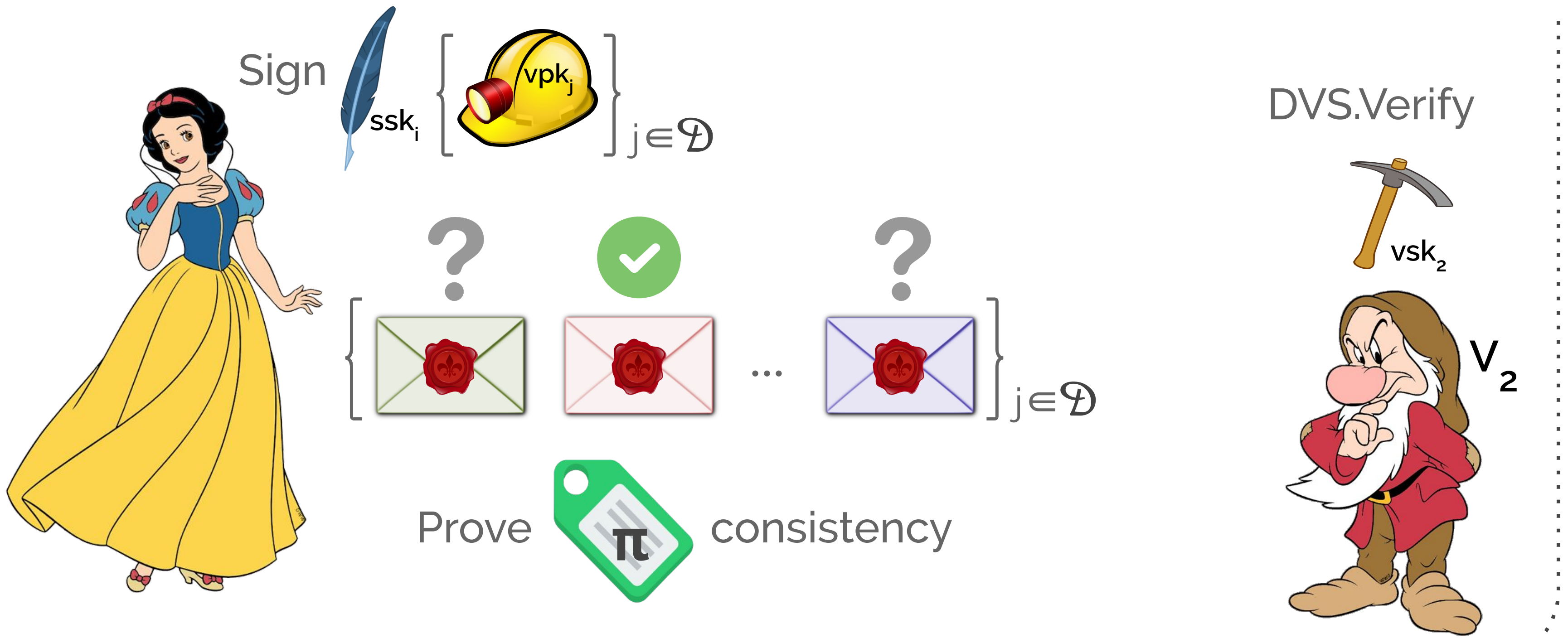
Sign $_{ssk_i}\left[vpk_j \right]_{j \in \mathcal{D}}$

DVS.Sign

MDVS from DVS

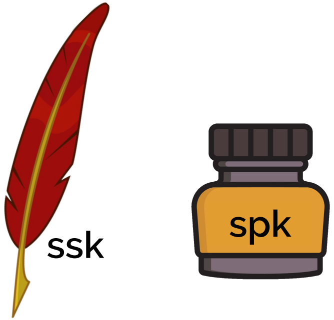


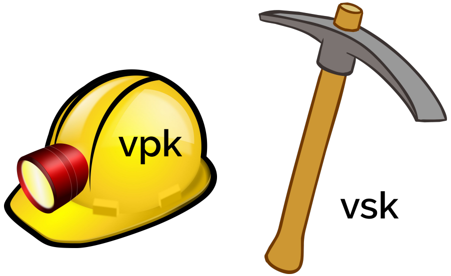
Consistency?



Provably Simulatable DVS

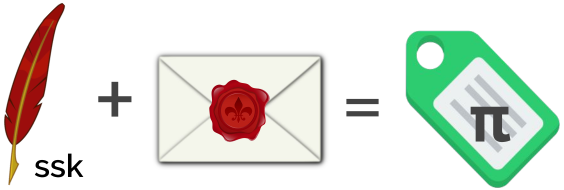
Provably Simulatable DVS



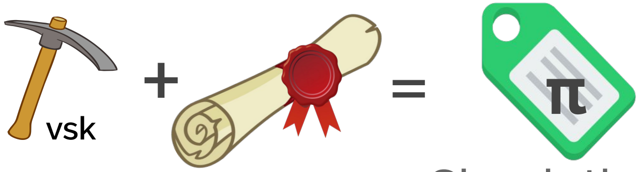


Sign & Prove Validate

Simulate & Prove Validate

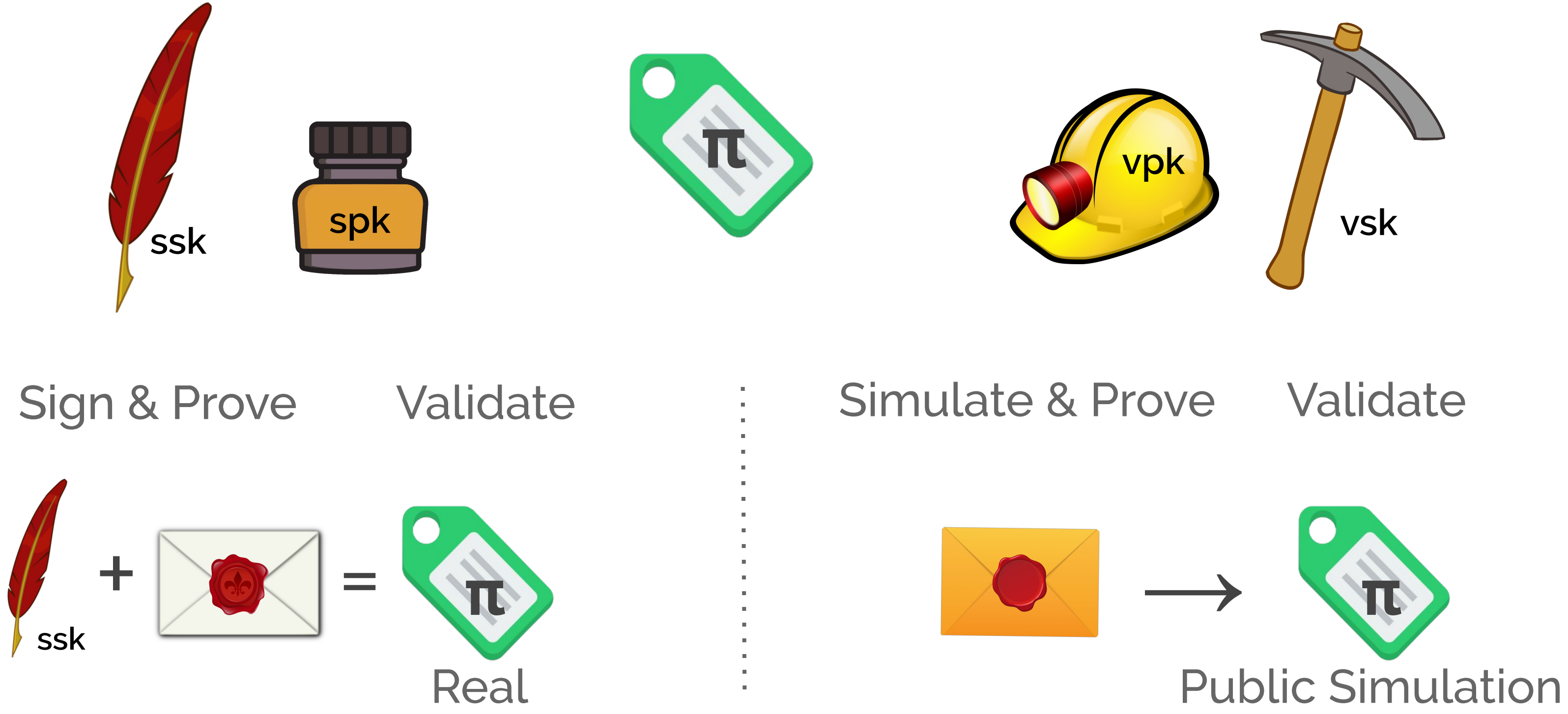


Real

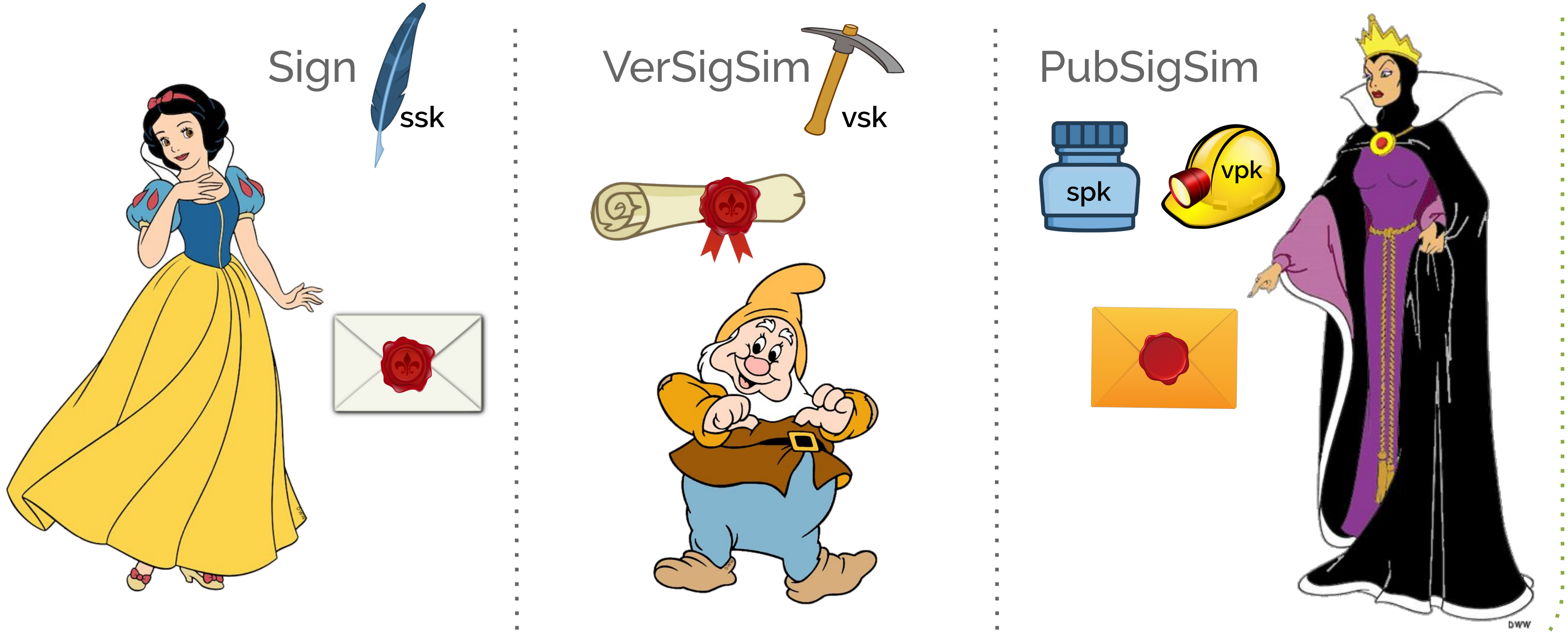


Simulation V

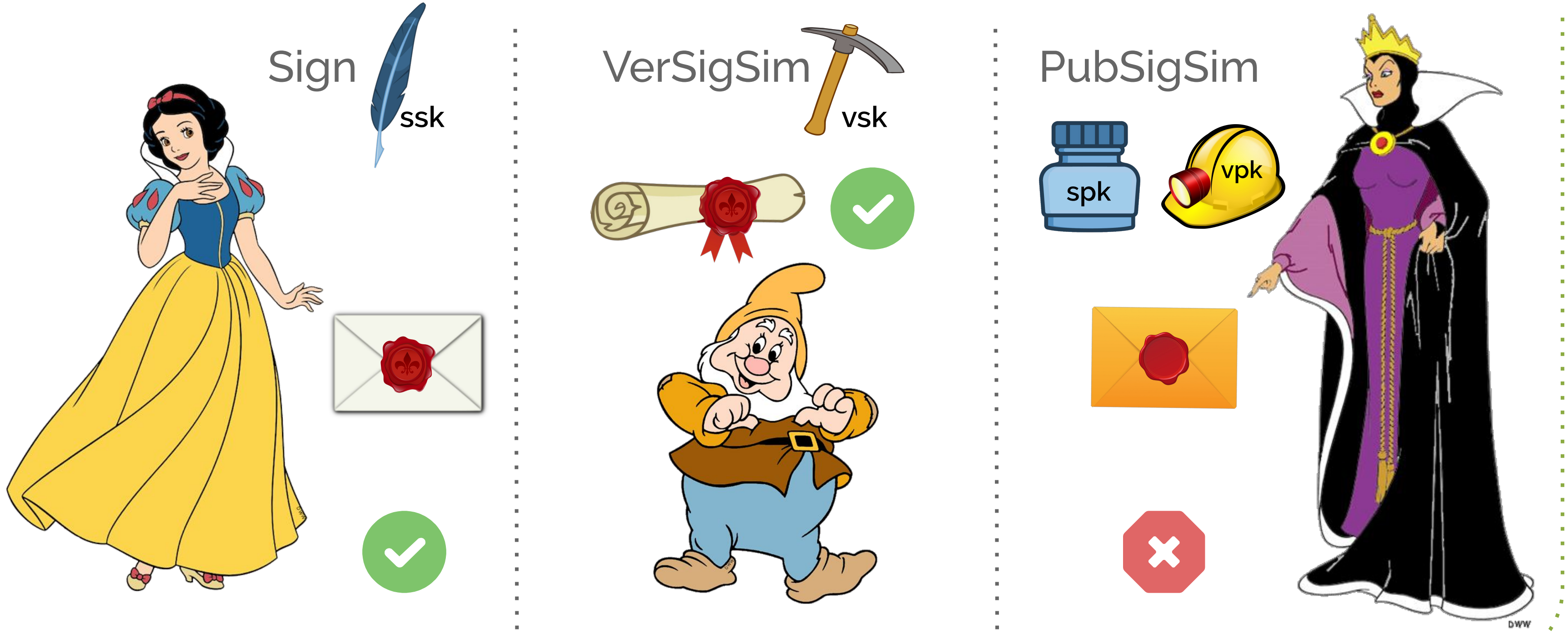
Provably Simulatable DVS



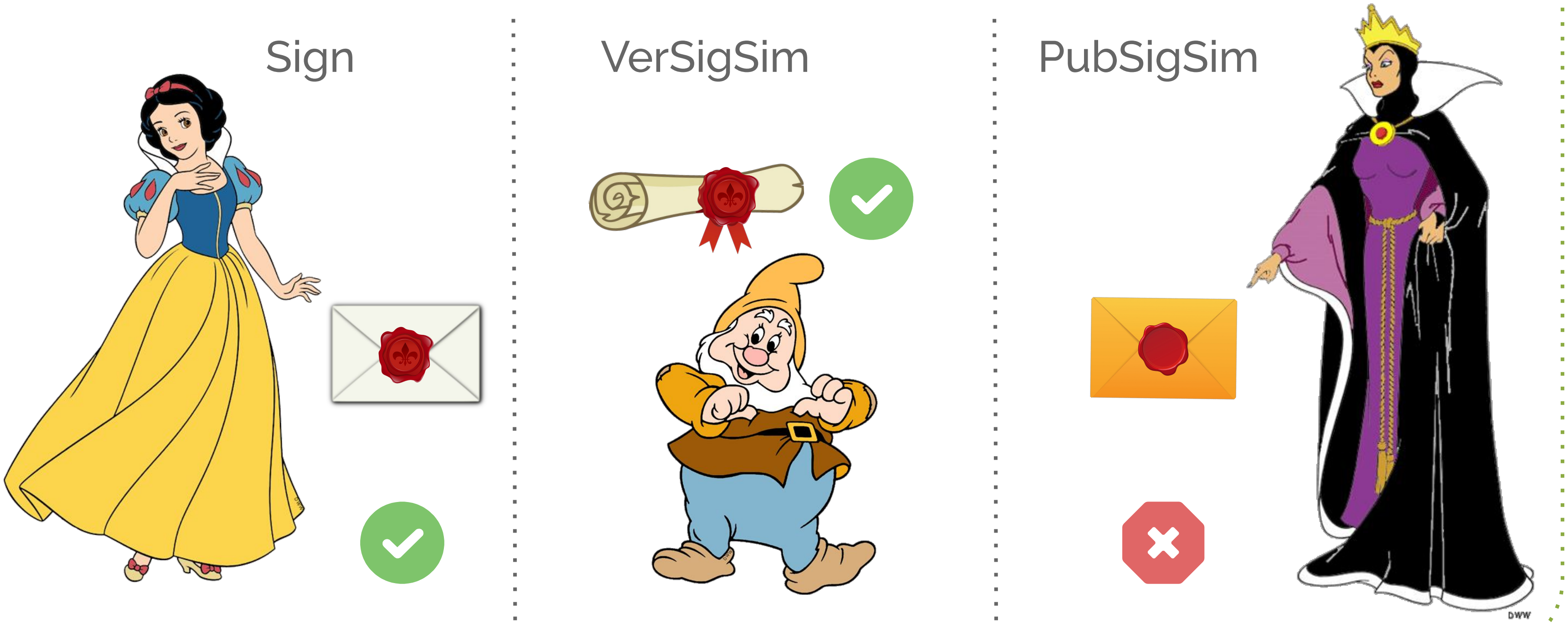
PSDVS: Sign & Simulate



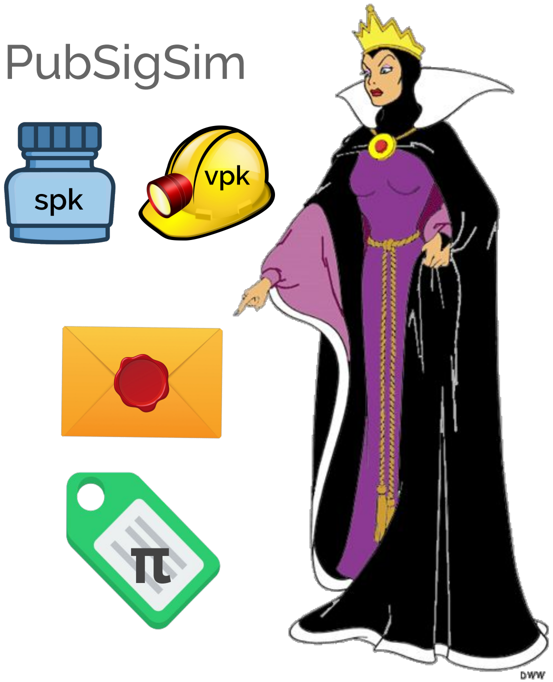
Verify



Verify

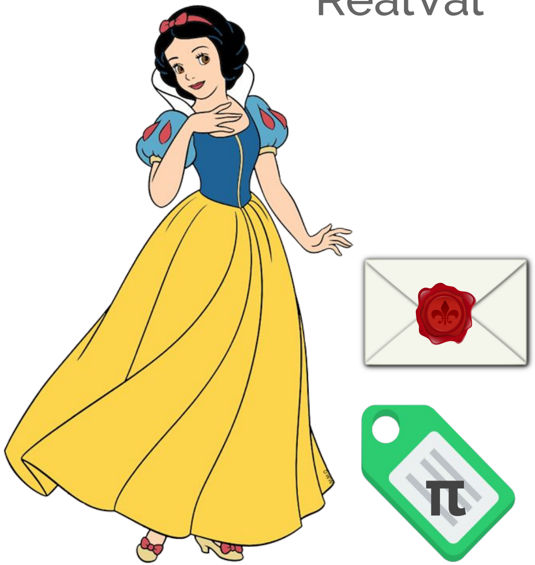


Provable



Validate

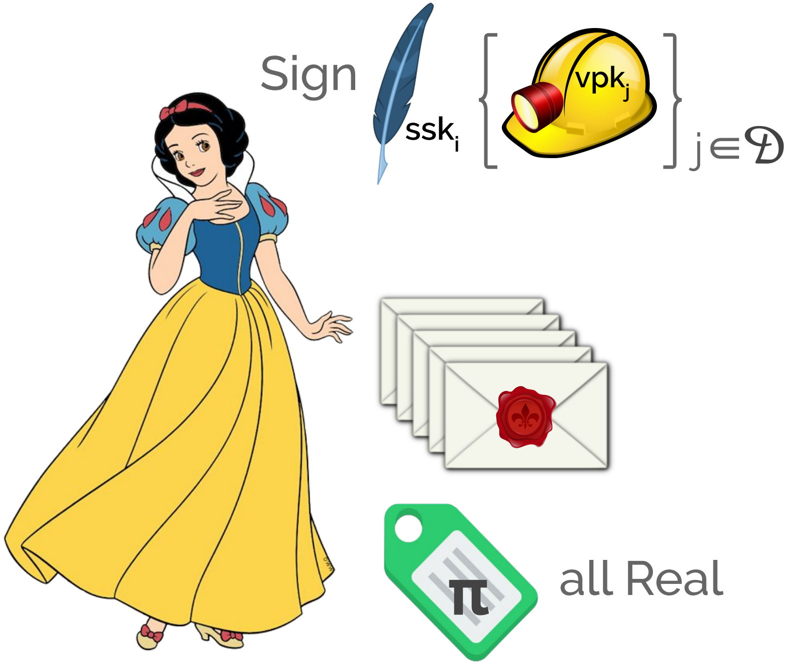
RealVal



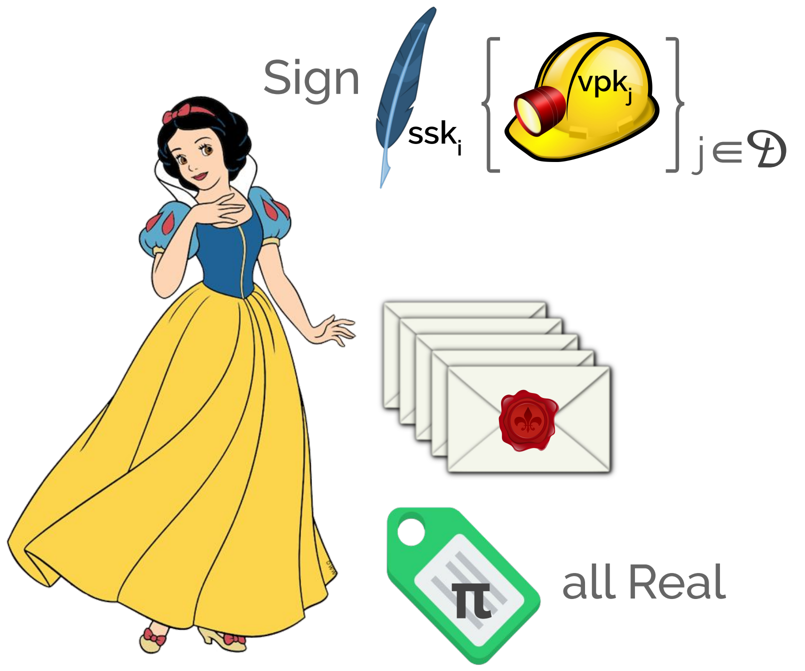
VerSigVal

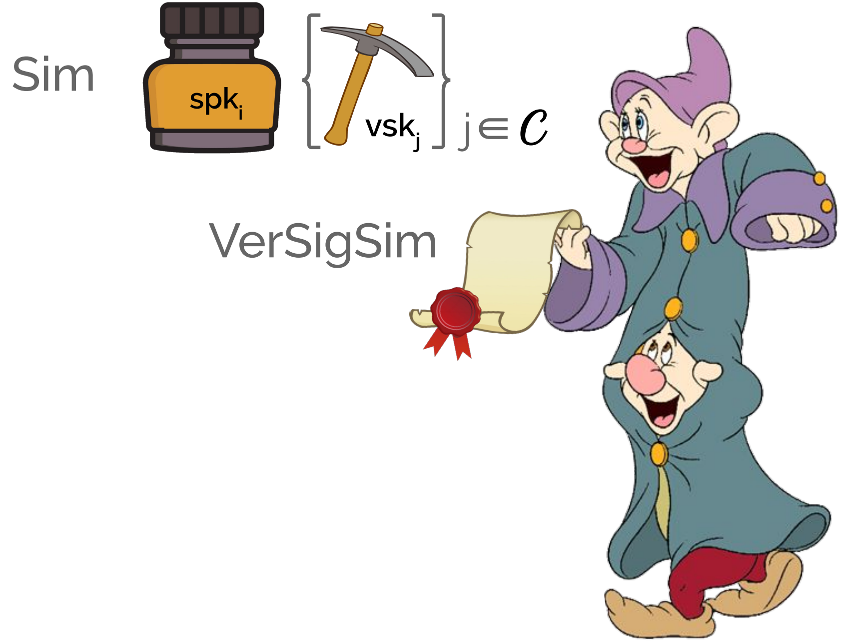
PubSigVal

MDVS from PSDVS



Sign & Simulate





Sign & Simulate

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

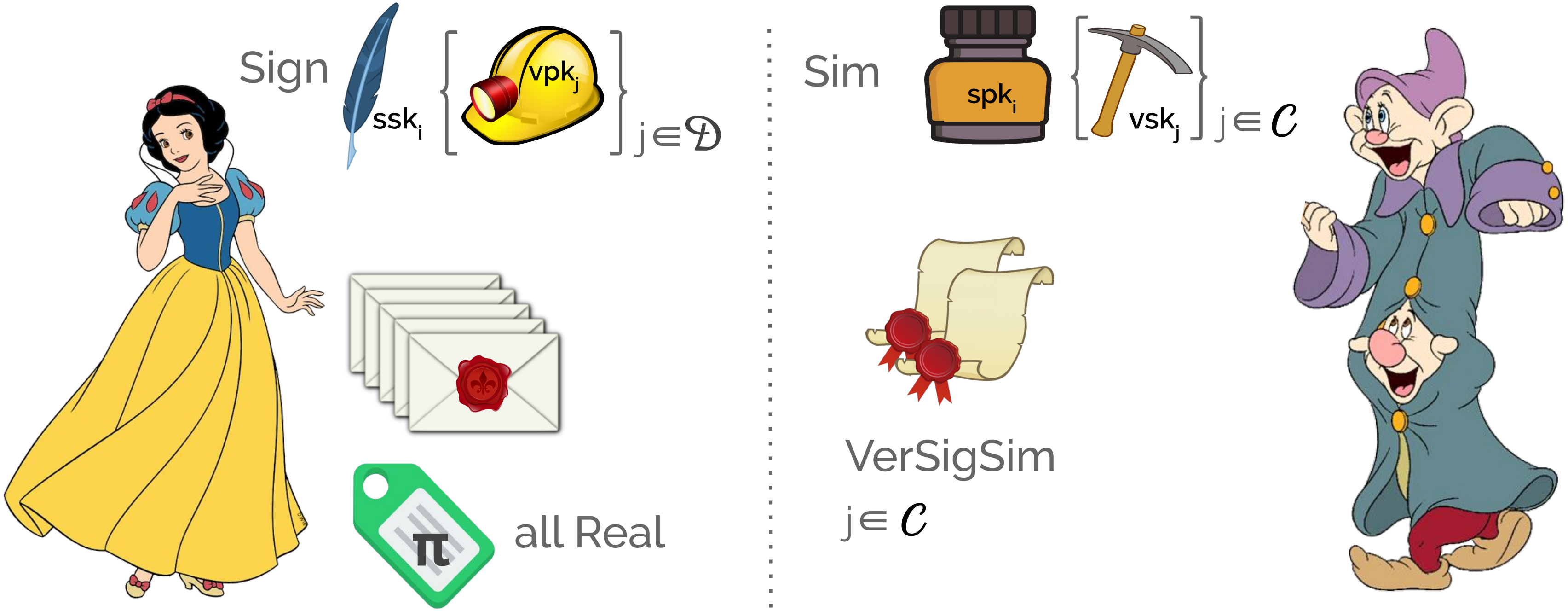
all Real

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

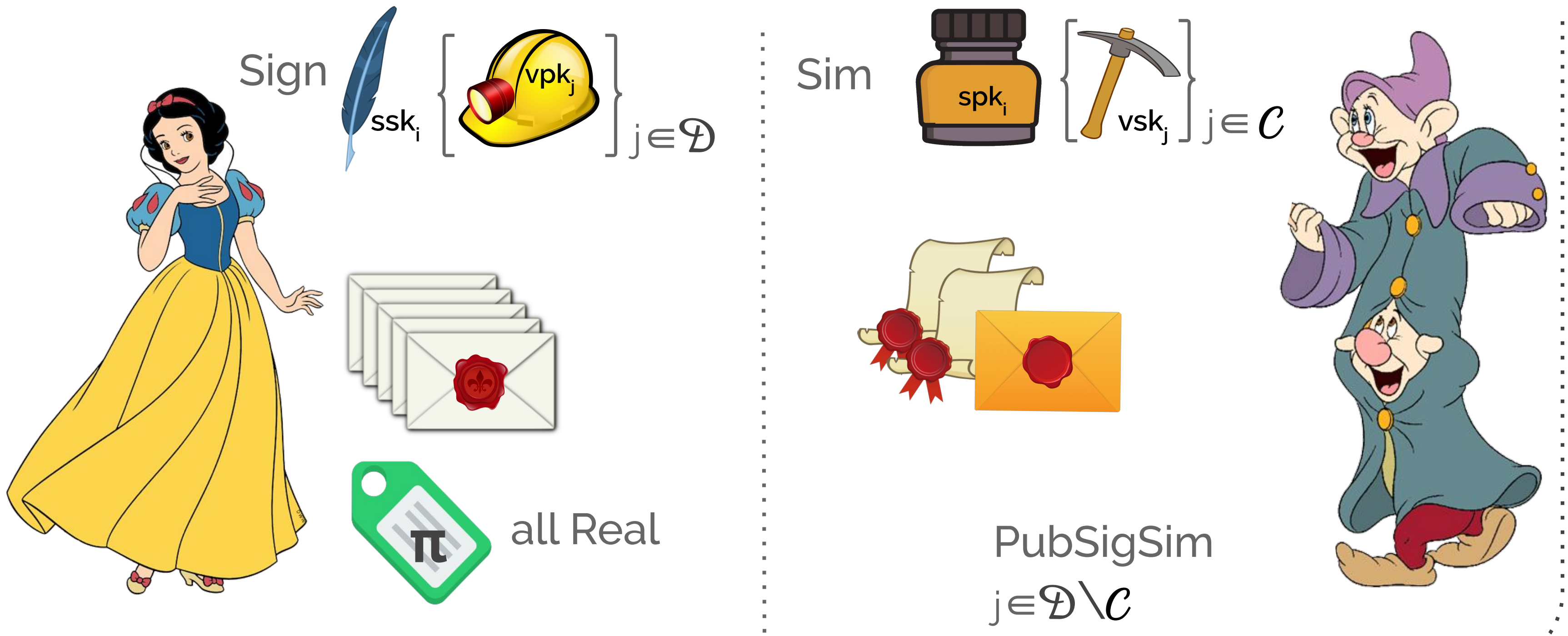
VerSigSim

$j \in \mathcal{C}$

Sign & Simulate



Sign & Simulate



Sign & Simulate

Sign  ssk_i $\left[\right.$  $\left. \mathrm{vpk}_j \right]_{j \in \mathcal{D}}$

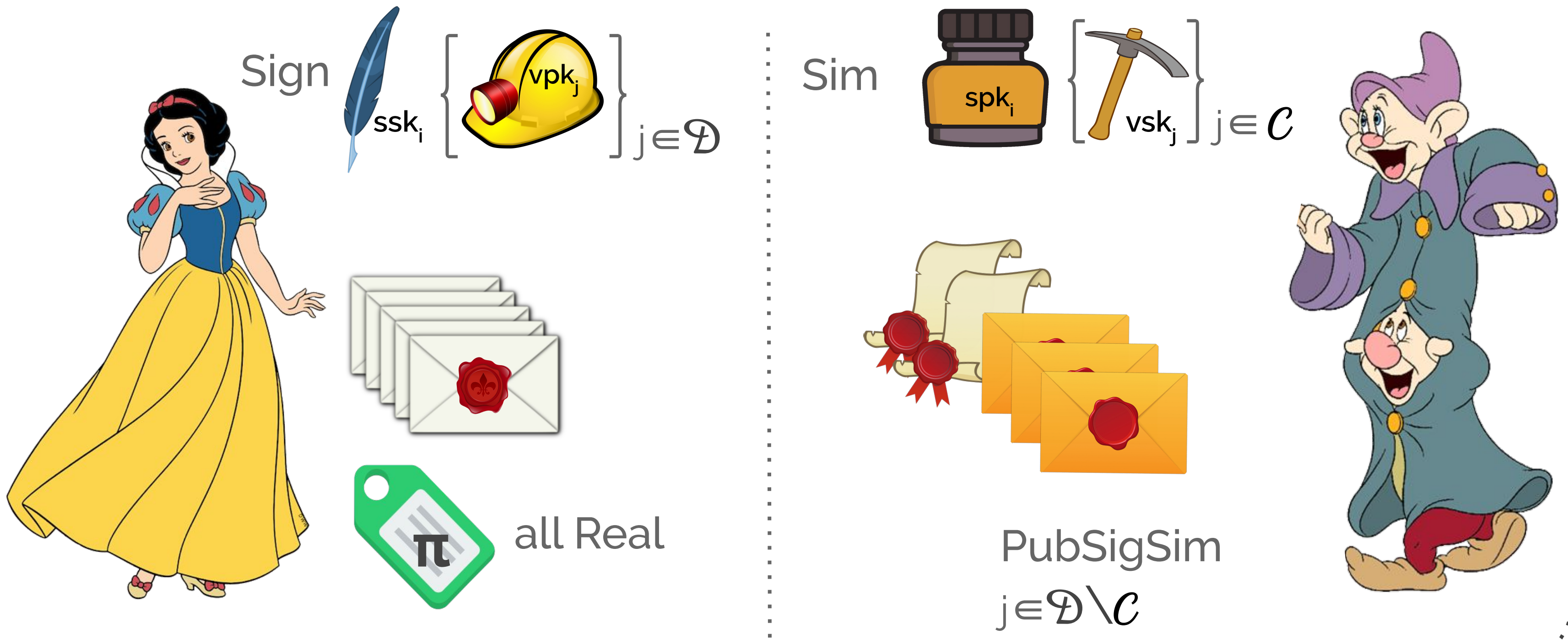
all Real

Sim  spk_i $\left[\right.$  $\left. \mathrm{vsk}_j \right]_{j \in \mathcal{C}}$

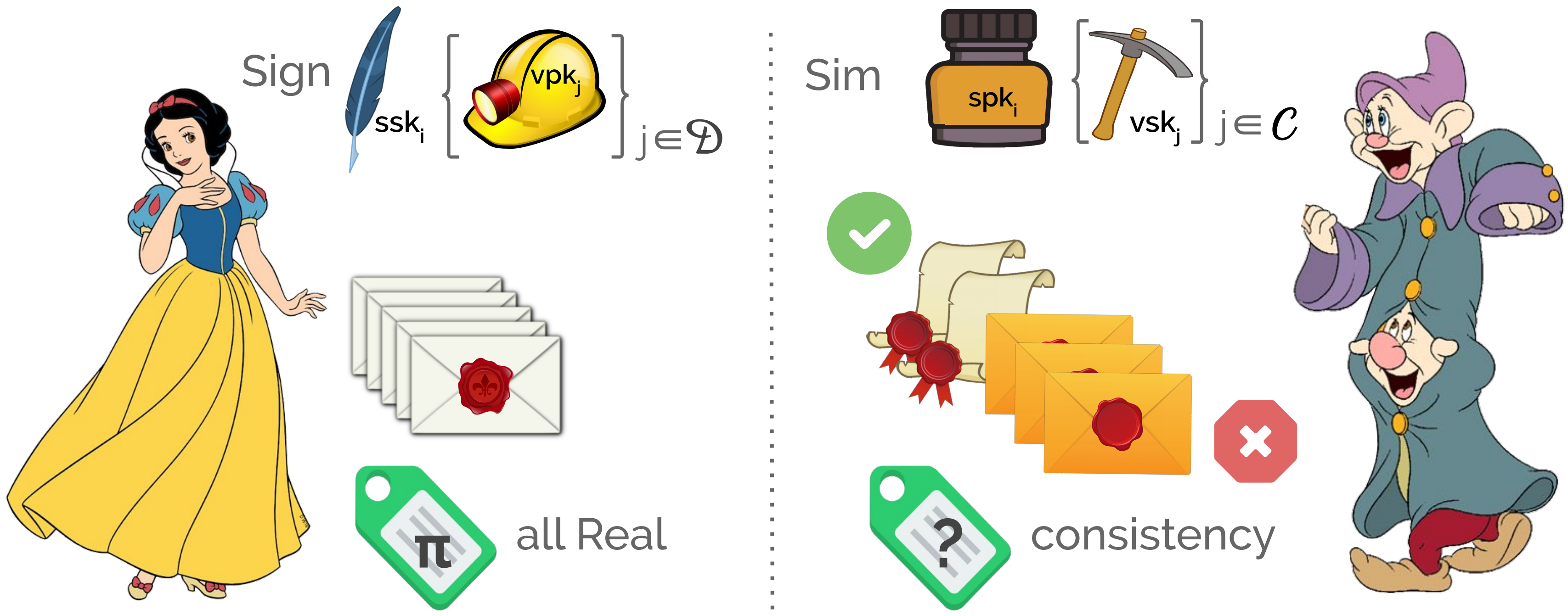
PubSigSim

$j \in \mathcal{D} \backslash \mathcal{C}$

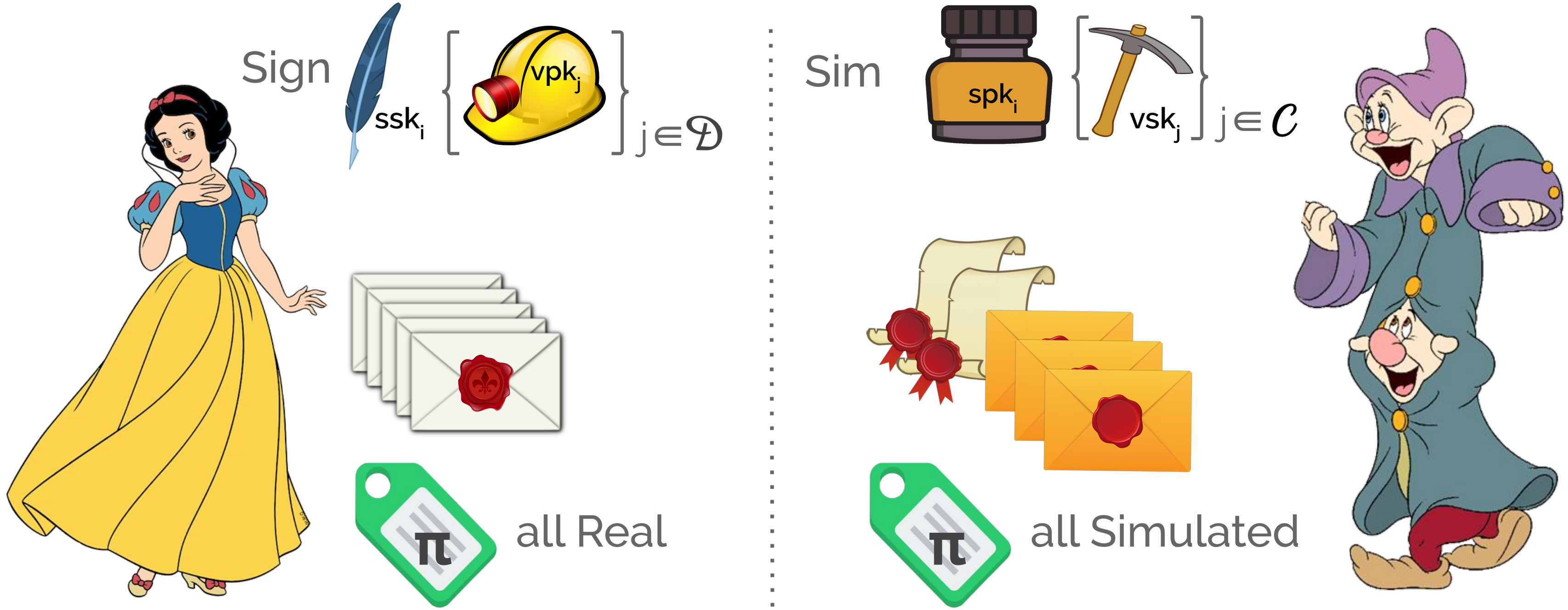
Sign & Simulate



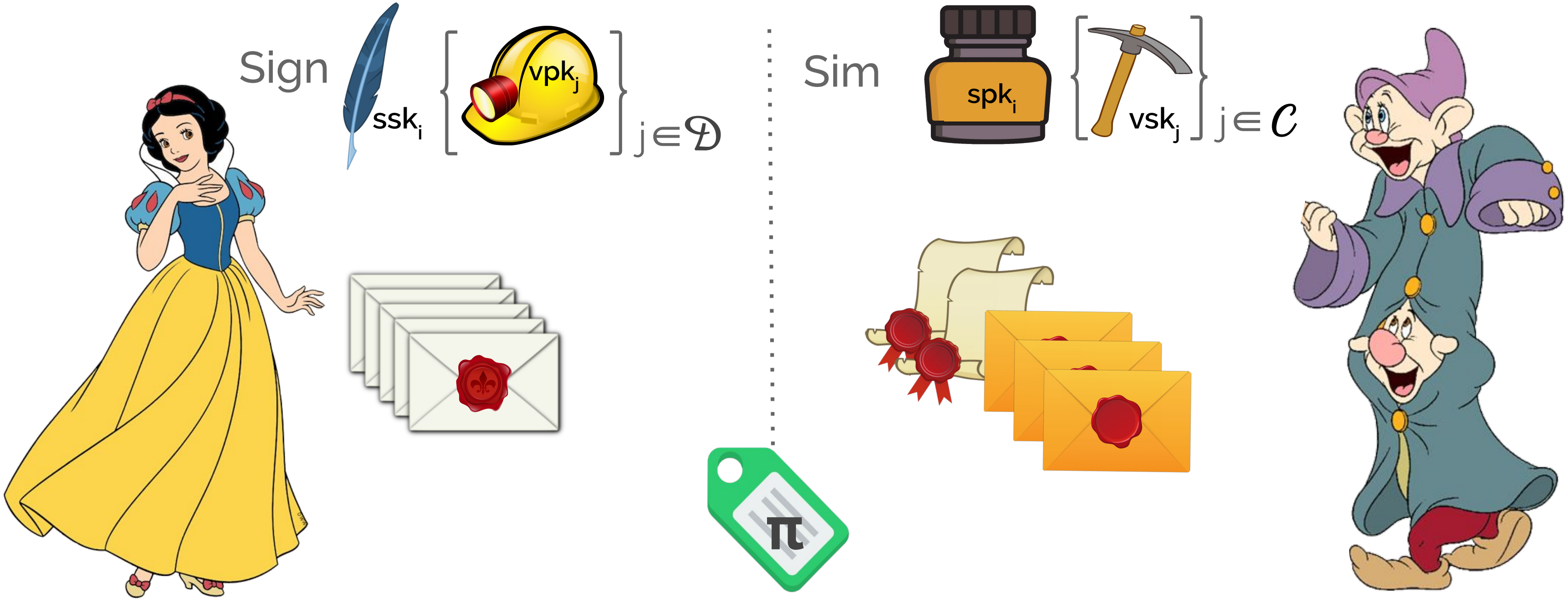
Source Hiding?



Source Hiding?

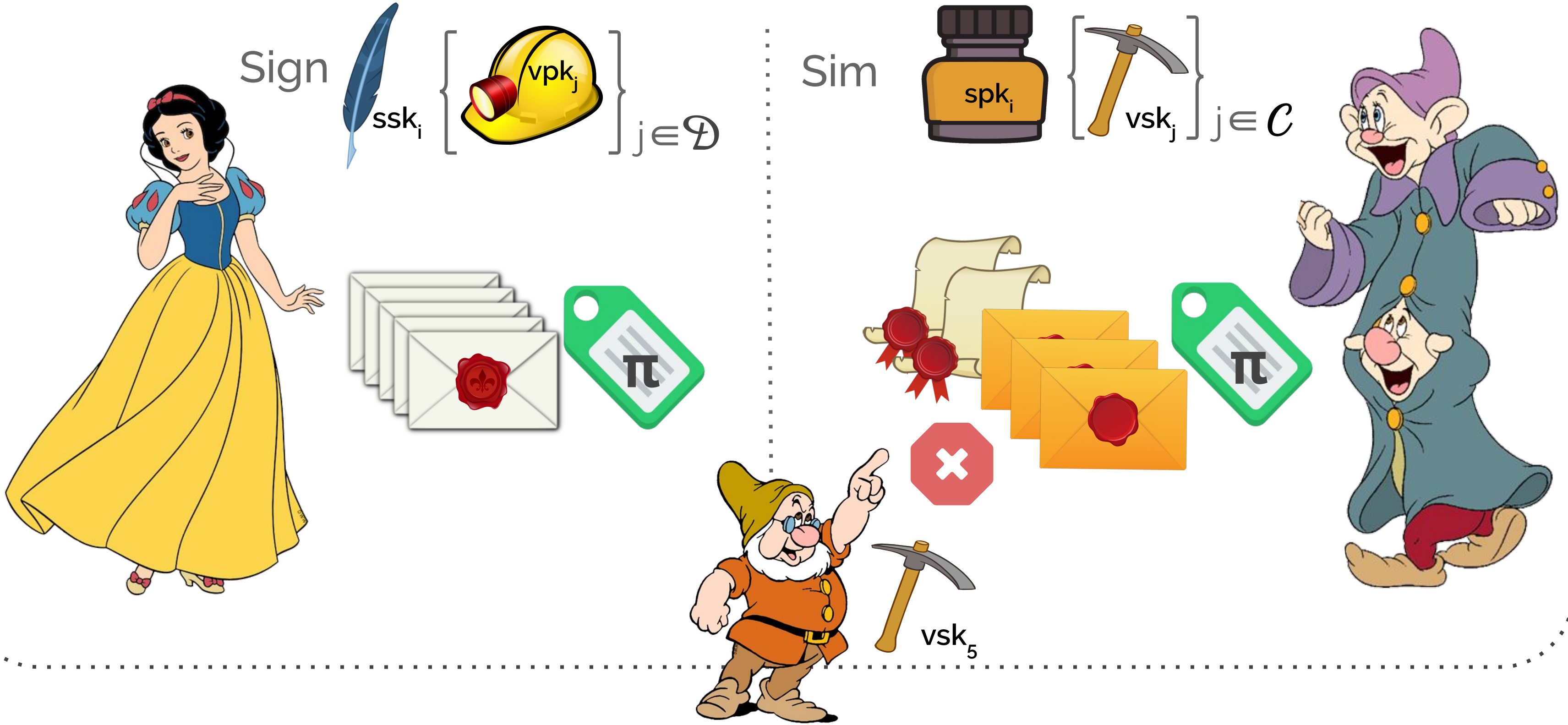


Source Hiding

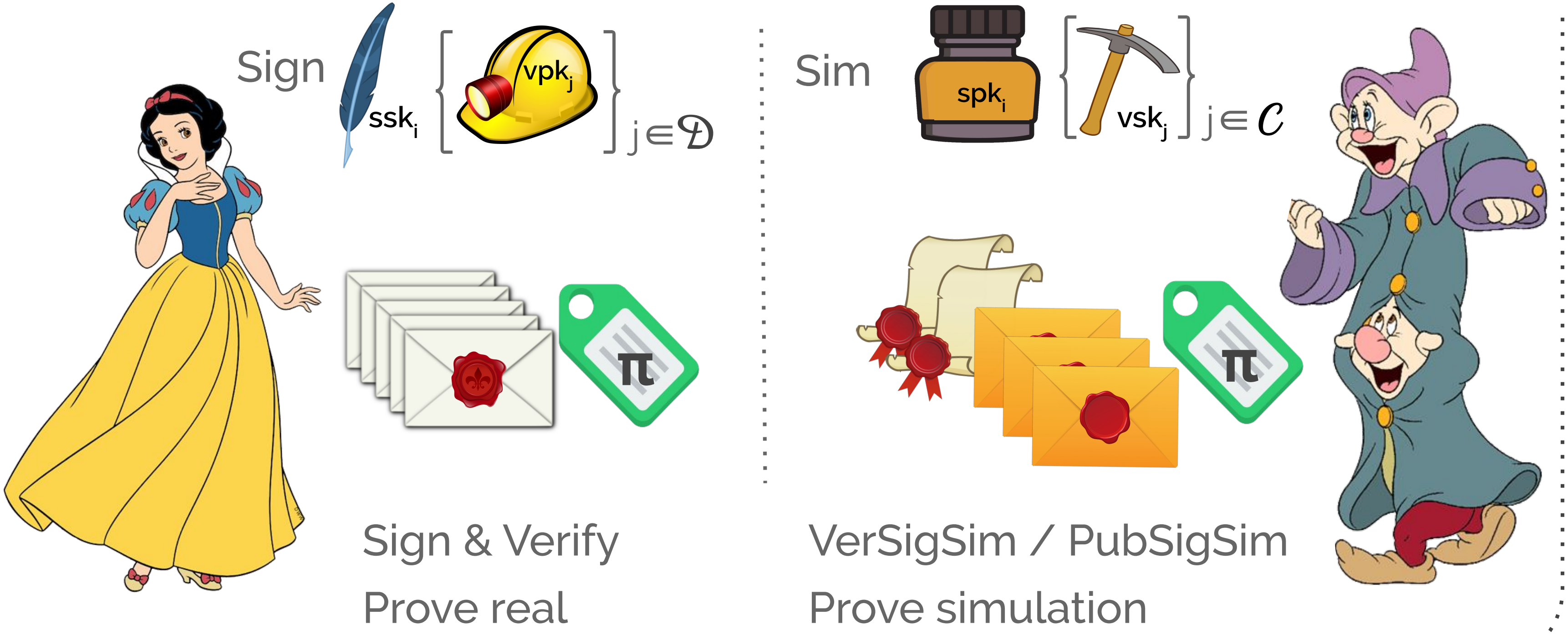


all Real or all Simulated

Unforgeability



MDVS from PSDVS



PSDVS 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-**H**iding
Strong **U**nforgeability

MDVS from
Functional
Encryption

MDVS from
Provably
Simulatable
DVS

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