Policy-Based Signature Scheme from Lattices

Shantian Cheng,
Khoa Nguyen and Huaxiong Wang

NANYANG TECHNOLOGICAL UNIVERSITY

Outline

1. Overview
2. Policy-Based Signatures
   - Definition
   - Our construction
3. Delegatable Policy-based Signature

Overview

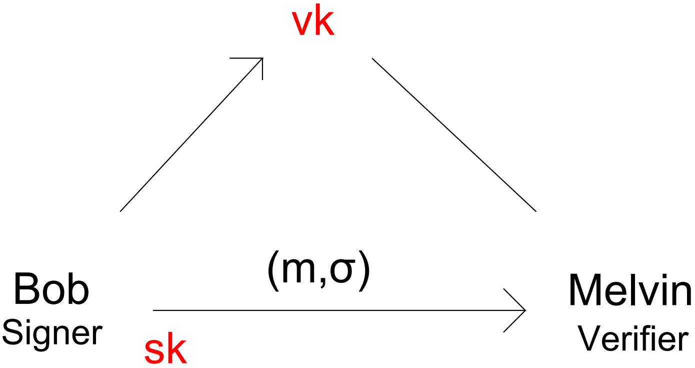
Background:

- Policy-based signature (PBS) is introduced by Bellare and Fuchsbauer ([BF14]).
- Signer is only allowed to sign messages satisfying certain policy.

Our innovation:

- We construct a PBS scheme based on lattices.

Digital Signature



NANYANG TECHNOLOGICAL UNIVERSITY

Practical Motivation

- A company allows the employees to sign contract anonymously on behalf of the company.
- Existing schemes:
  - **Group signature:** Anonymous signing, no control of what can be signed.
  - **Attribute-based signature (ABS):** The singer can sign a message with a predicate (public) that is satisfied by his attributes (private).

$$\text{CEO} \vee (\text{Senior Manager} \wedge \text{Leader of Project A})\,.$$

Practical motivation

A company issues a key to a staff Bob that

- he can only sign contracts with companies A and on behalf of company.

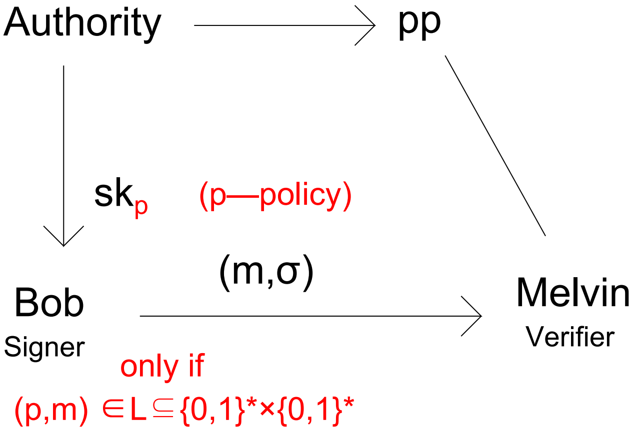
Advantage of PBS:

- There are some constraints on what can be signed.
- In ABS, to verify the signature, Melvin need to know whether

$$\text{Bob} \in \text{Manager} \vee \text{Leader of Project A}$$

- In PBS, the policies can be more complicated (for example, sign contract at fixed time) and the verifier Melvin only knows the public parameter.

Policy-Based Signature(PBS)



Intuitive Security

- **Unforgeability:** The signer can sign a message m only if he owns the key sk_p, where m satisfies policy p.
- **Privacy:** The signature does not reveal the policy.

Theoretical motivation

- Digital signature deserves more investigation.
- PBS unifies some existing signature schemes. For example, group signature and attribute-based signature can be derived from PBS.

Outline

1 Overview

2 Policy-Based Signatures
- Definition
- Our construction

3 Delegatable Policy-based Signature

Definition

Policy language: A policy checker is a **NP**-relation

$$\begin{aligned} \mathrm{PC} : \{0,1\}^* \times \{0,1\}^* &\rightarrow \{0,1\} \\ ((p,m),w) &\mapsto 0 \text{ or } 1. \end{aligned}$$

The policy language associated to PC is defined as

$$\mathcal{L}(\mathrm{PC}) := \{(p,m) : \exists w \text{ such that } \mathrm{PC}((p,m),w) = 1\}.$$

NANYANG TECHNOLOGICAL UNIVERSITY

PBS scheme

Algorithms:

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{pp}, \mathsf{msk})$
- $\mathsf{KeyGen}(\mathsf{msk}, p) \to \mathsf{sk}_p$
- $\mathsf{Sign}(\mathsf{sk}_p, m) \to \sigma$
- $\mathsf{Verify}(\mathsf{pp}, m, \sigma) \to 0$ or 1

Unforgeability

Game:

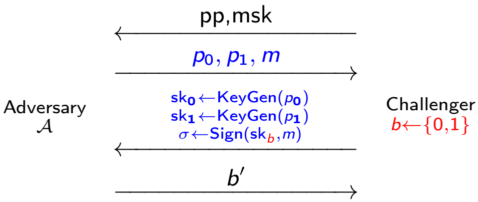
Adversary $\mathcal{A}$		Challenger
	$\xleftarrow{\quad \text{pp} \quad}$	
	Query key for p $\longrightarrow$	
	$\longleftarrow$ Return sk_p	
	Query signature on m $\longrightarrow$	
	$\longleftarrow$ Return σ	
	(m^*, σ^*) $\longrightarrow$	

$\mathcal{A}$ wins if

- $\text{Verify}(\text{pp}, m^*, \sigma^*) = 1$;
- $\mathcal{A}$ did not query signature of m^*;
- for all p ever queried for key: $(p, m^*) \notin \mathcal{L}(\text{PC})$.

Indistinguishability

Game:



$\mathcal{A}$ wins if

- $(p_0, m), \ (p_1, m) \in \mathcal{L}(\mathrm{PC})$;
- $b' = b$.

Stronger Security Notions

- **Simulatability.**
  - One scheme revealing policies may satisfy indistinguishability.
  - In simulatability, it requires there is an algorithm

$$\sigma \leftarrow \mathsf{Simulate}(m),$$

whose outputs are indistinguishable from real signatures.

- **Extractability.**
  - In unforgeability, to check whether $(p, m^*) \notin \mathcal{L}(\mathrm{PC})$ for all queried p may be inefficient.
  - In extractability, it requires there is an algorithm

$$(p^*, w^*) \leftarrow \mathsf{Extract}(m^*, \sigma^*).$$

It only need to verify p^* has not been queried.

Lattice based Policy Language.

We define the policy checker as PC : $(\{0,1\}^\ell \times \mathbb{Z}_q^n) \times \mathbb{Z}_q^n \to \{0,1\}$ satisfying

$$\mathsf{PC}((\mathbf{p},\mathbf{M}),\mathbf{w}) = 1 \iff \begin{cases} \mathbf{G}\cdot\mathbf{p} + \mathbf{w} = \mathbf{M} \mod q, \\ ||\mathbf{w}||_\infty \leq \beta \end{cases}$$

Construction

KeyGen($\mathbf{p}$): $\mathsf{sk}_\mathbf{p} = \mathbf{z}$ is one Bonsai signature ([CHKP10]) on $\mathbf{p}$.

Sign($\mathsf{sk}_\mathbf{p}, \mathbf{M}$): non-interactive zero-knowledge argument of knowledge to show the possession of $(\mathbf{p}, \mathbf{z}, \mathbf{w})$ such that

- $(\mathbf{p}, \mathbf{z})$ is a valid Bonsai massage-signature pair.
- $\mathsf{PC}((\mathbf{p}, \mathbf{M}), \mathbf{w}) = 1$.

We apply

- Stern's 3-move argument system ([LLNW14]). (Interactive)
- Fiat-Shamir heuristic. (Non-interactive)

Security

- **Simulatability.** Based on the zero-knowledge property of the underlying argument system.
- **Extractability.** We reduce the SIS-solver to the successful extractability adversary of our scheme.

Outline

Delegation.

- Policies may be set up hierarchically.
- Possessing sk_p, one can delegate $\mathsf{sk}_{p'}$ for subpolicy p'.

	Alice →	Bob →	Tom
Case 1:	sign contract with A,B,C	sign contract with B	
Case 2:	sign contract with X,Y	sign contract with X first, then Y	

Techniques

- Policy vectors: $(\mathbf{p}_1, \cdots, \mathbf{p}_r) \rightarrow (\mathbf{p}_1, \cdots, \mathbf{p}_r, \mathbf{p}_{r+1})$.
- Instead of short vector in PBS, signing key in DPBS is short basis of certain lattices.
- For signing phase, $\mathbf{z}_r \leftarrow \mathsf{SampleD}(\mathbf{A}_r, \mathbf{S}_r, \mathbf{u}, \sigma)$.
- For delegation phase,

$$\overline{\mathbf{S}}_{r+1} \leftarrow \mathsf{ExtBasis}(\mathbf{A}_r, \mathbf{A}_{r+1}, \mathbf{S}_r)$$

and

$$\mathbf{S}_{r+1} \leftarrow \mathsf{RandBasis}(\mathbf{A}_{r+1}, \overline{\mathbf{S}}_{r+1}, \sigma). \quad ([\mathrm{CHKP10}])$$

References

Mihir Bellare and Georg Fuchsbauer.
Policy-Based Signatures.
In *Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 520–537. Springer, 2014.

David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert.
Bonsai Trees, or How to Delegate a Lattice Basis.
In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer, 2010.

Adeline Langlois, San Ling, Khoa Nguyen, and Huaxiong Wang.
Lattice-Based Group Signature Scheme with Verifier-Local Revocation.
In *Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 345–361. Springer, 2014.

THANK YOU

THANK YOU!