Policy-based signatures

Mihir Bellare
UCSD

Georg Fuchsbauer
IST Austria

PKC 2014, 28 March 2014

(Full version: eprint 2013/413)

Overview

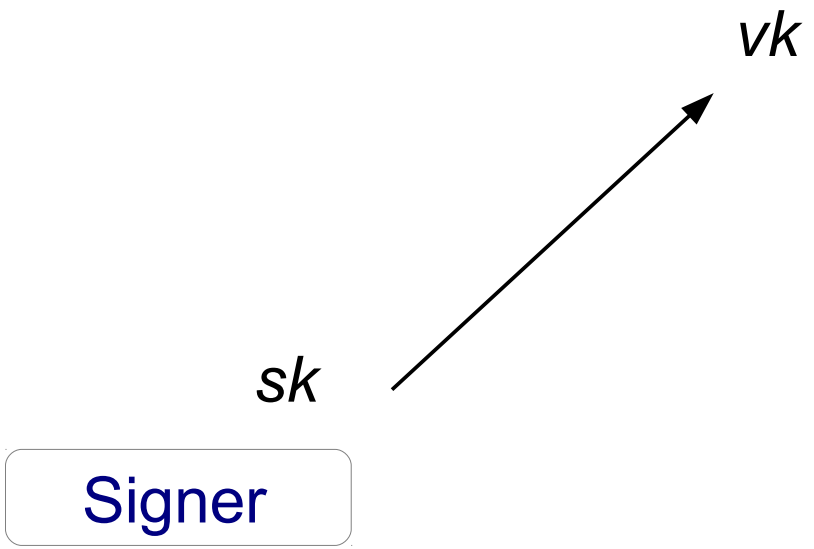
- New signature primitive
- Signer can only sign messages conforming to policy

Overview

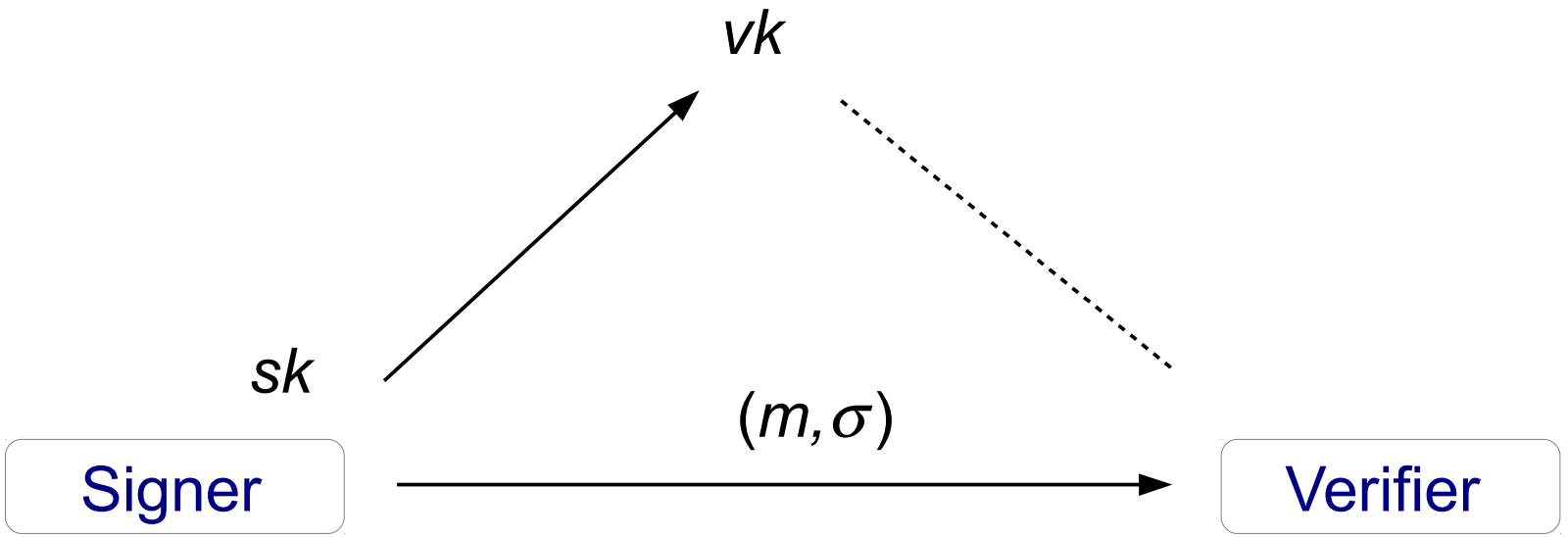
- New signature primitive
- Signer can only sign messages conforming to policy

- Practical applications: use for corporations
- Theoretical: unification of existing work

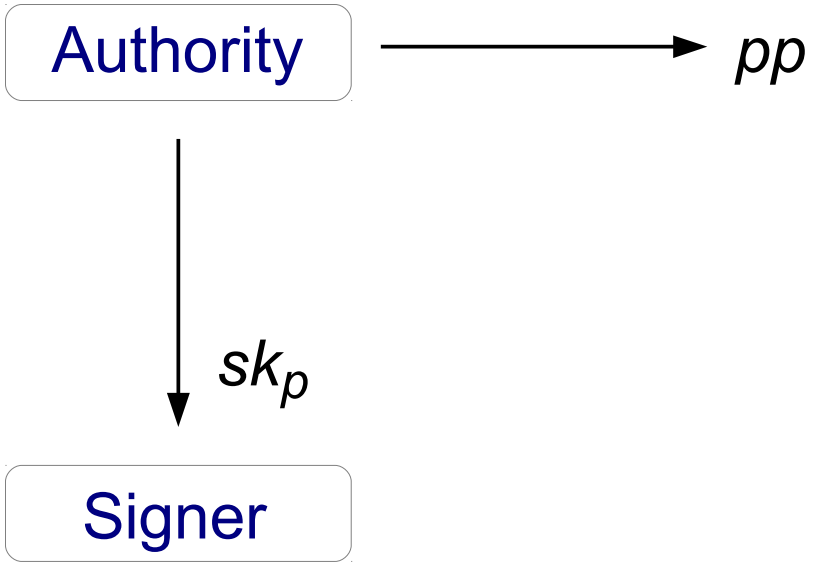
Signatures



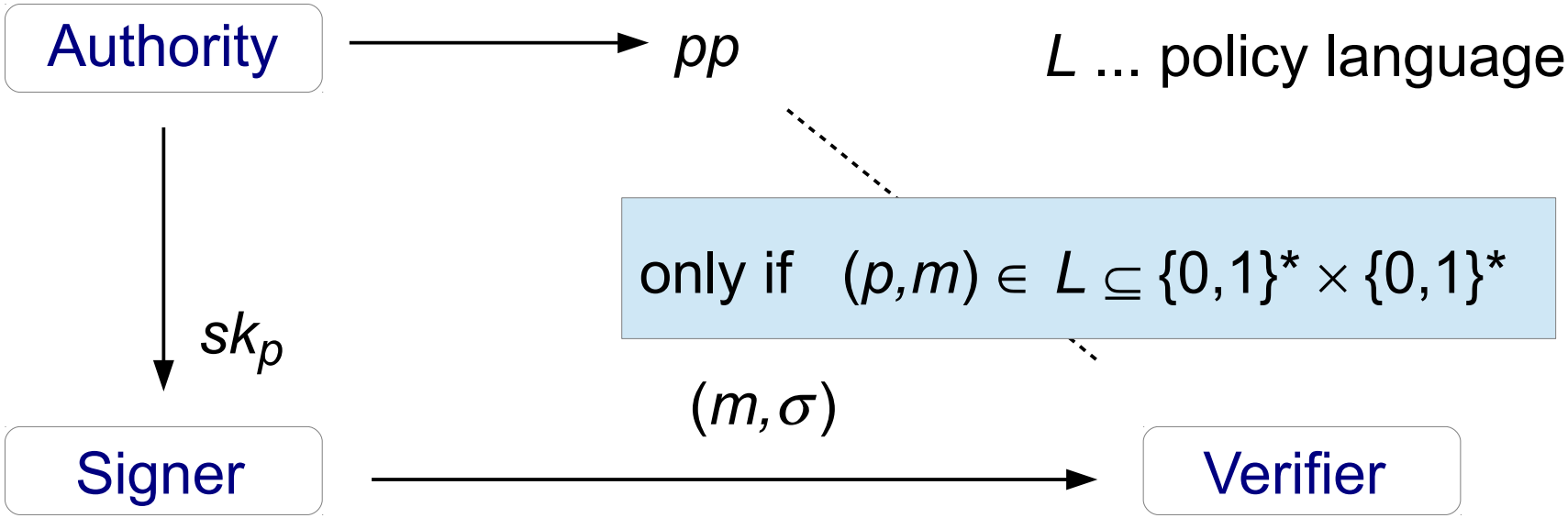
Signatures



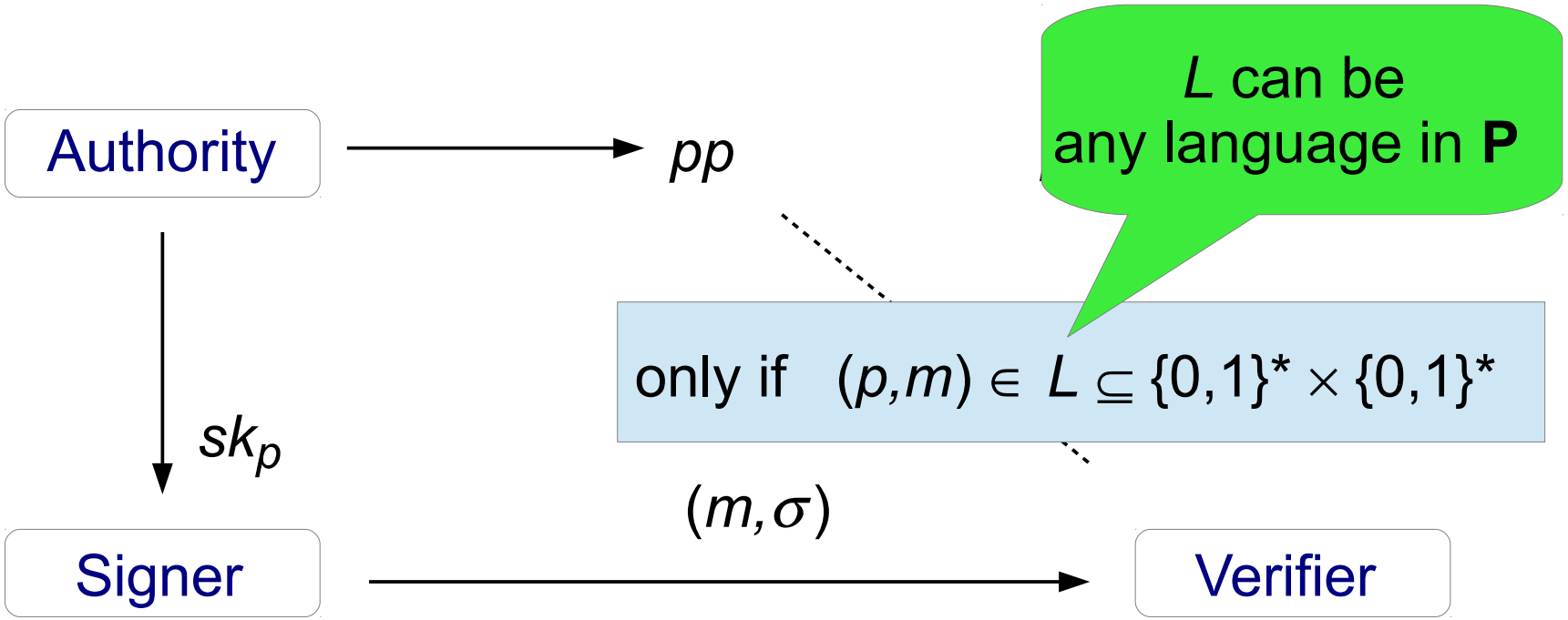
Policy-based signatures



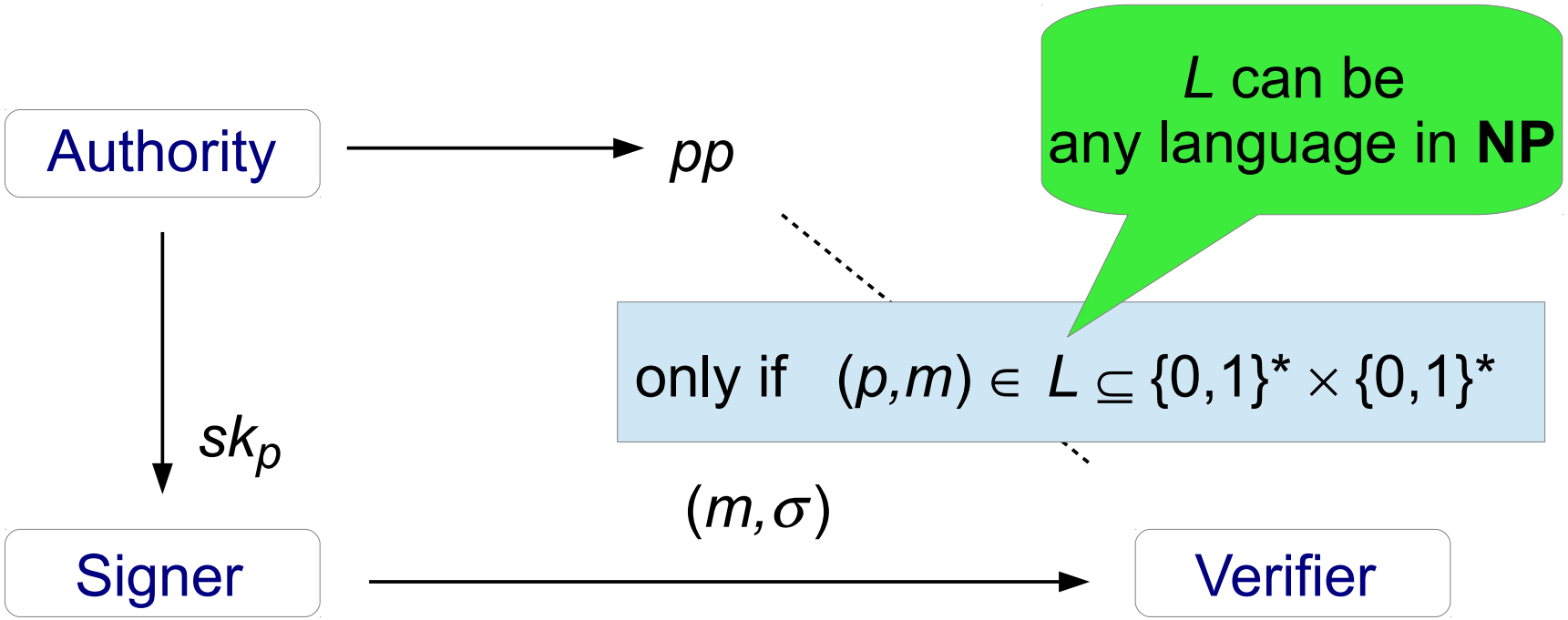
Policy-based signatures



Policy-based signatures



Policy-based signatures



Security

- Unforgeability:
  - You can only sign a message *m* if you have a key for a policy *p* satisfied by *m*

Security

- Unforgeability:
  - You can only sign a message *m* if you have a key for a policy *p* satisfied by *m*

- Privacy:
  - The signature hides the policy
  - Signatures under same key are unlinkable

Related work

- Functional signatures (Boyle, Goldwasser, Ivan [BGI13]):
  - Key sk_f allows signing messages in range of f
  - Interpret f as policy: $(f,m) \in L \;:\Leftrightarrow\; \exists\, w : f(w) = m$

Related work

- Functional signatures (Boyle, Goldwasser, Ivan [BGI13]):
  - Key sk_f allows signing messages in range of f
  - Interpret f as policy: $(f,m) \in L \;:\Leftrightarrow\; \exists\, w : f(w) = m$
  - Policy languages in **P**, succinctness condition

Related work

- Functional signatures (Boyle, Goldwasser, Ivan [BGI13]):
  - Key sk_f allows signing messages in range of f
  - Interpret f as policy: $(f,m) \in L \;:\Leftrightarrow\; \exists\, w : f(w) = m$
  - Policy languages in **P**, succinctness condition
- Delegatable functional signatures (Backes, Meiser, Schröder [BMS13]):
  - Signatures verified w.r.t. signer's public key

Related work

- Constrained / delegatable / functional PRFs [BW13, BGI13, KPTZ13]:
  - Keys enable evaluation of PRF on parts of domain

Related work

- Constrained / delegatable / functional PRFs [BW13, BGI13, KPTZ13]:
  - Keys enable evaluation of PRF on parts of domain
- Attribute-based signatures [MPR11]:
  - Keys issued for set of attributes $\{a_1, a_2, \ldots, a_n\}$
  - Signing w.r.t. predicate φ, possible iff $\varphi(a_1, a_2, \ldots, a_n) = 1$

Motivation for PBS

Practical motivation

- Company with public key *vk*
- Employees get signing keys enabling signing
  anonymously on behalf of company

Practical motivation

- Company with public key *vk*
- Employees get signing keys enabling signing
  anonymously on behalf of company

- Group signatures [Cv91]:
  - Anonymous signing, no control of what can be signed

Practical motivation

- Company with public key *vk*
- Employees get signing keys enabling signing anonymously on behalf of company

- Group signatures [Cv91]:
  - Anonymous signing, no control of what can be signed
- Attribute-based signatures [MPR11]:
  - Verification w.r.t. policies

Practical motivation

- Company with public key *vk*
- Employees get signing keys enabling signing
  anonymously on behalf of company

- Group signatures [Cv91]:
  - Anonymous signing, no control of what can be signed

- Attribute-based signatures [MPR11]:
  - Verification w.r.t. policies $\text{CEO} \vee (\text{board member} \wedge \text{manager})$

Can we do better?

⇒ Public policies...

- Does verifier need to know?

Can we do better?

⇒ Public policies...

- Does verifier need to know?

⇒ Verification w.r.t. policies...

- Verifier must judge if message OK under policy

Can we do better?

⇒ Public policies...

- Does verifier need to know?

⇒ Verification w.r.t. policies...

- Verifier must judge if message OK under policy, e.g.

$$\text{CEO} \vee \text{Intern}$$

Can we do better?

⇒ Public policies...

- Does verifier need to

⇒ Verification w.r.t. policies

- Verifier must judge if message OK under policy, e.g.

CEO ∨ Intern

Policy-based signatures:

- No public policies
- Verification w.r.t. *vk* only

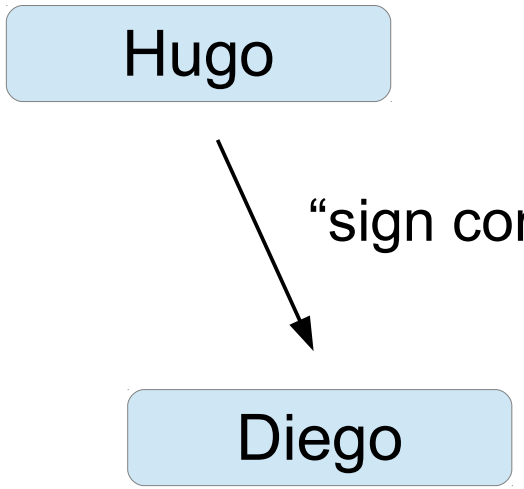
Can we do better?

Policy-based signatures:

- No public policies
- Verification w.r.t. *vk* only

Can we do better?

Example:

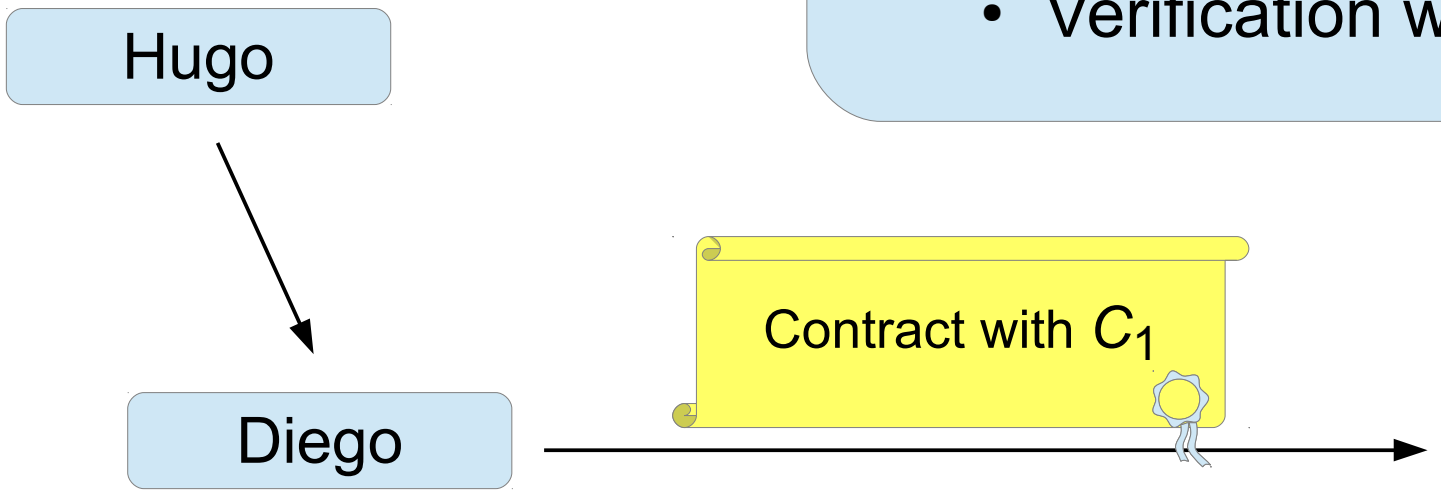


Policy-based signatures:

- No public policies
- Verification w.r.t. *vk* only

Can we do better?

Example:



Policy-based signatures:

- No public policies
- Verification w.r.t. *vk* only

Theoretical motivation

- Signature analog to functional encryption [BSW11]
  - FE: Simply encrypt message, let keys handle access
  - PBS: Simply verify signature; keys handle authorization

Theoretical motivation

- Signature analog to functional encryption [BSW11]
  - FE: Simply encrypt message, let keys handle access
  - PBS: Simply verify signature; keys handle authorization
- Unification of existing notions for signatures with privacy:

  (Anonymous) proxy signatures [MUO96, FP08]

  Ring signatures, mesh signatures [RST01, Boy07]

  Attribute-based signatures [MPR11]

  Anonymous credentials [CL01, BCKL08]

  Group signatures [Cv91]

Definition of PBS

Definition

- Policy languages:

  We allow any language in **NP**, defined by policy checker

$$(p,m) \in L(\mathrm{PC}) \;:\Leftrightarrow\; \exists\, w : \mathrm{PC}((p,m),w) = 1$$

Definition

- Policy languages:

  We allow any language in **NP**, defined by policy checker

$$(p,m) \in L(\mathrm{PC}) \;:\Leftrightarrow\; \exists\, w : \mathrm{PC}((p,m),w) = 1$$

m conforms to policy *p*

Definition

- Policy languages:

  We allow any language in **NP**, defined by policy checker

  $$(p,m) \in L(\mathrm{PC}) \;:\Leftrightarrow\; \exists\, w : \mathrm{PC}((p,m),w) = 1$$

- Algorithms:

  $\mathrm{Setup}(1^\lambda) \rightarrow (pp, msk)$

  $\mathrm{KeyGen}(msk, p) \rightarrow sk_p$

  $\mathrm{Sign}(sk_p, m, w) \rightarrow \sigma$

  $\mathrm{Verify}(pp, m, \sigma) \rightarrow b$

Security

- Indistinguishability

  An adversary, given *msk*, outputs sk_0 , sk_1

  and cannot tell with which key a signature was created

Security

- Indistinguishability

  An adversary, given *msk*, outputs sk_0 , sk_1

  and cannot tell with which key a signature was created

- Unforgeability

  An adversary, after querying: - keys for policies p_1, ... , p_n
  - signatures on messages

  should not be able to create signature on new m^*

  with (p_1, m^*), ... , $(p_n, m^*) \notin L$

Security

- Indistinguishability

  An adversary, given *msk*, outputs sk_0 , sk_1

  and cannot tell with which key a signature was created

- Unforgeability

  An adversary, after querying: - keys for p

  - signatures on messag

  should not be able to create signature on new m^*

  with $(p_1, m^*), \ldots , (p_n, m^*) \notin L$

not efficiently decidable

Sim/ext security

- Simulatability $\Rightarrow$ indistinguishability

Sim/ext security

- Simulatability $\not\Leftarrow$ $\Rightarrow$ indistinguishability

Sim/ext security

- Simulatability $\not\Leftarrow$ $\Rightarrow$ indistinguishability

- Extractability $\Rightarrow$ unforgeability

  is efficiently decidable

Constructions of PBS

Construction I

- Generic construction (à la [BMW03])

  based on - signatures

  - IND-CPA encryption

  - NIZK proofs for any policy language in **NP**

Construction II

- Concrete construction

  based on - structure-preserving signatures [AFG+10]

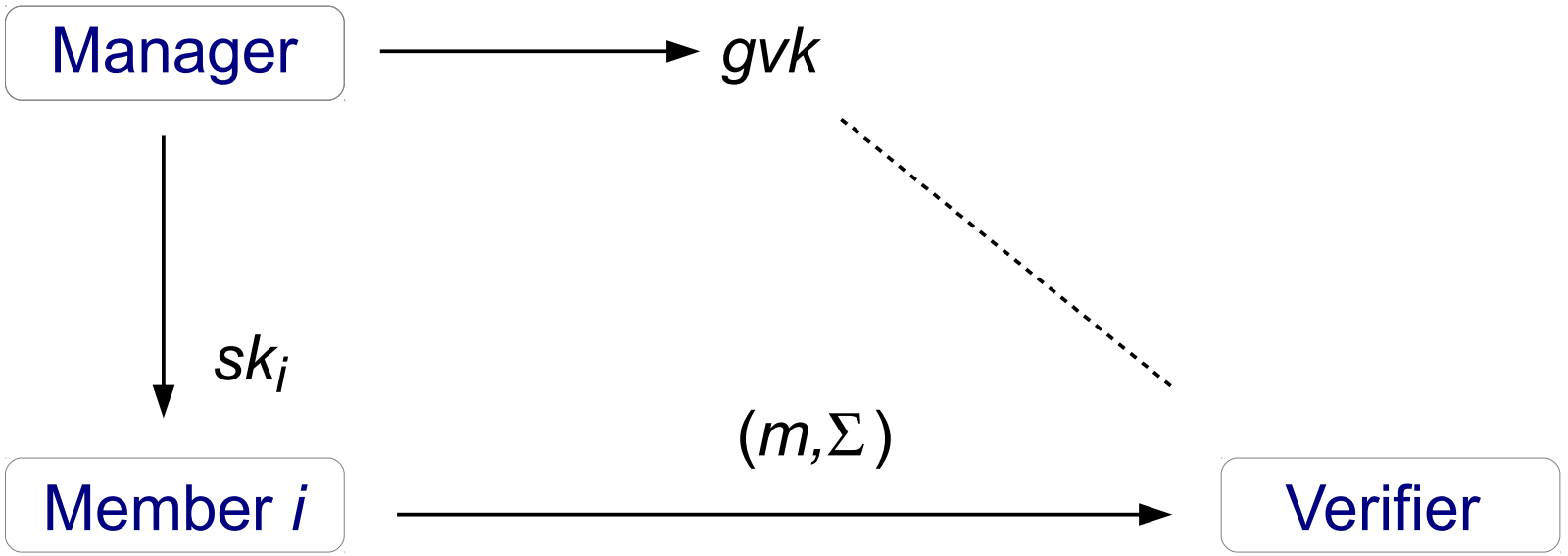
  - Groth-Sahai proofs [GS08]

  for policy languages over **pairing groups**

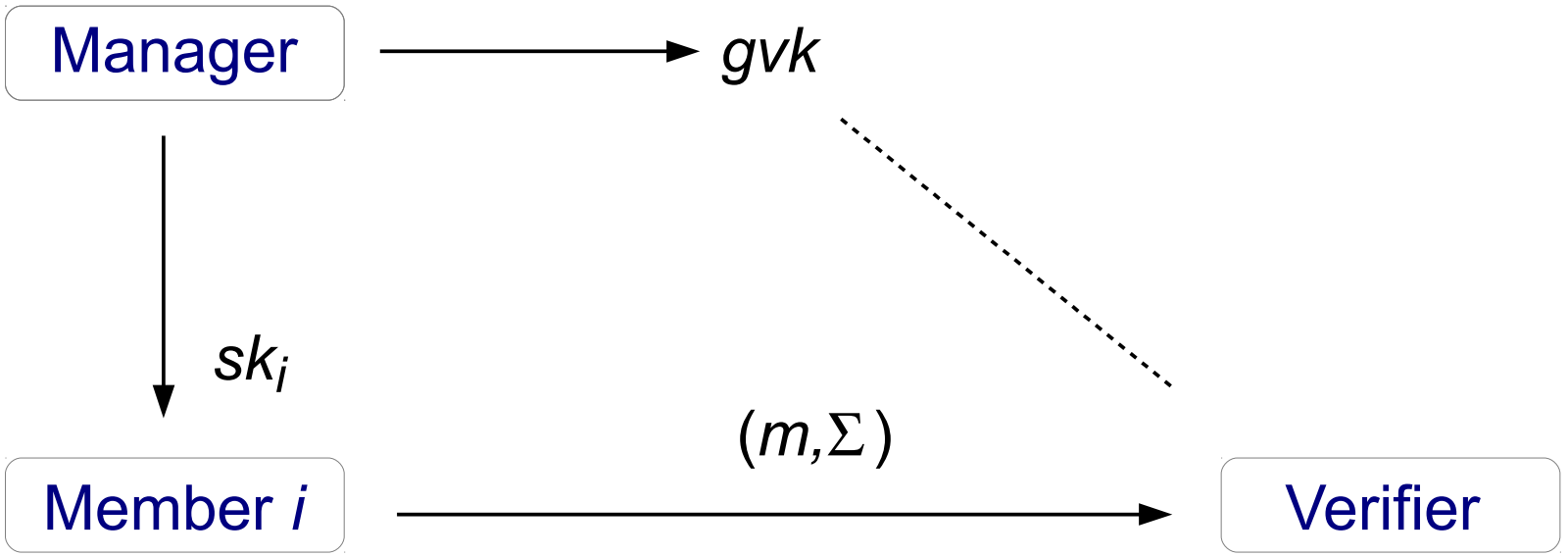
  (policies define pairing-product equations)

Primitives from PBS

CCA-secure group signatures

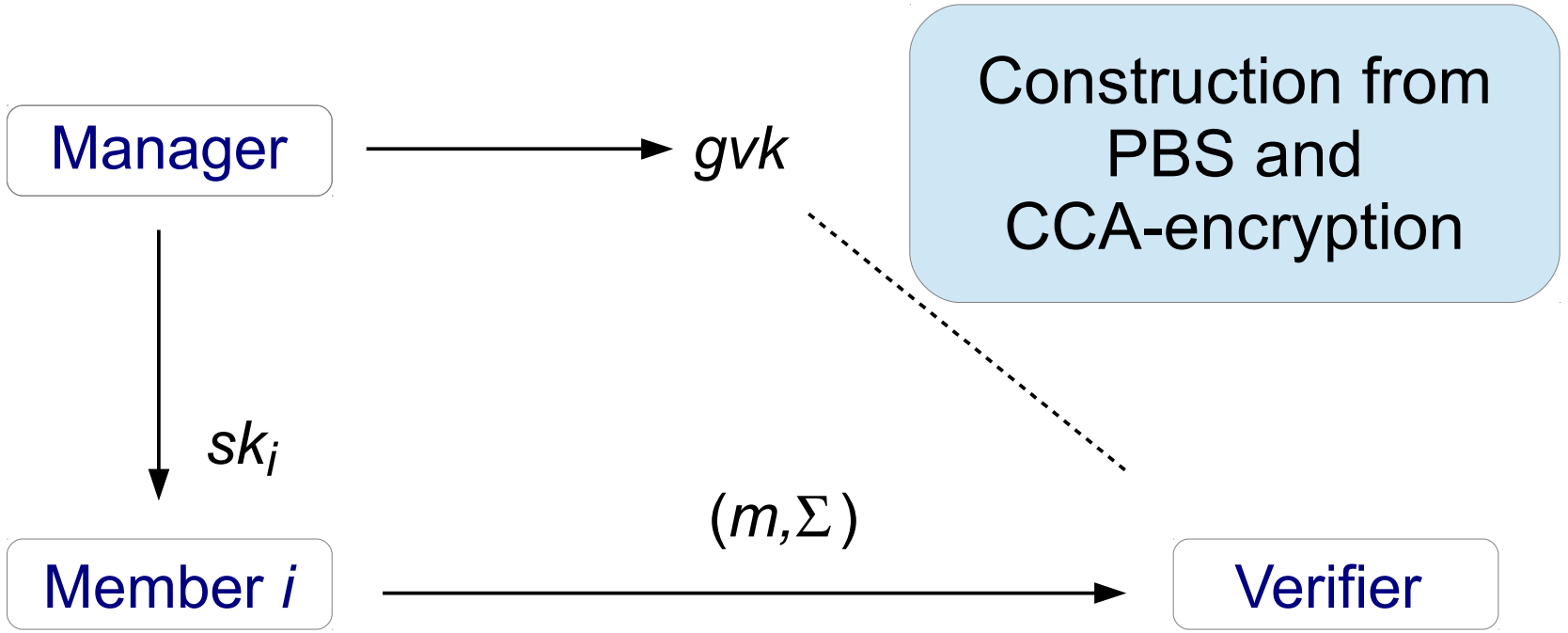


CCA-secure group signatures



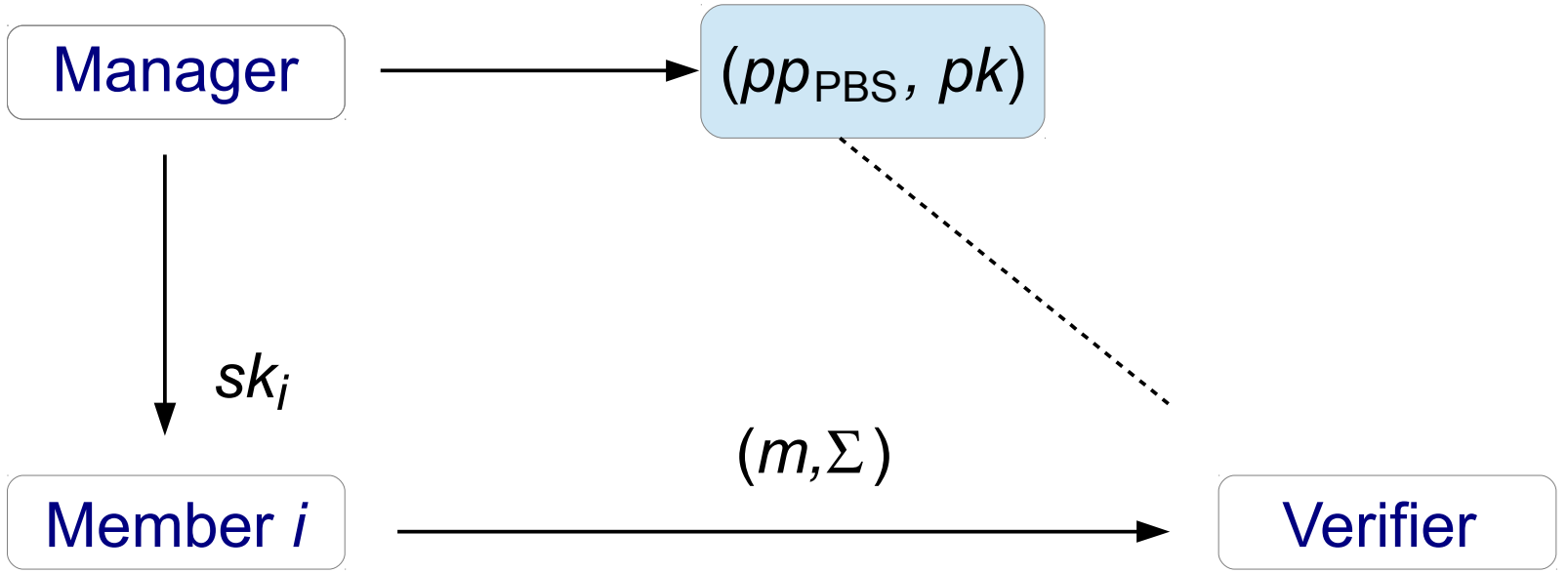
Σ can be opened: Σ —Manager→ *i*

CCA-secure group signatures



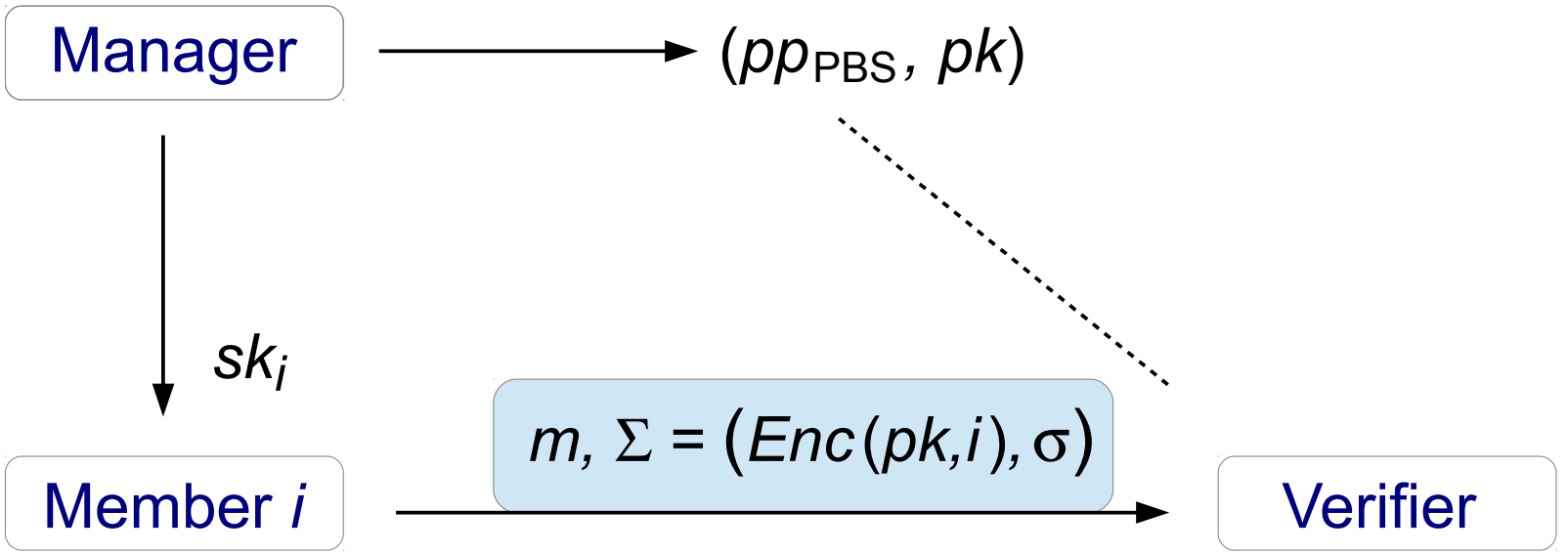
Σ can be opened: Σ —Manager→ *i*

CCA-secure group signatures



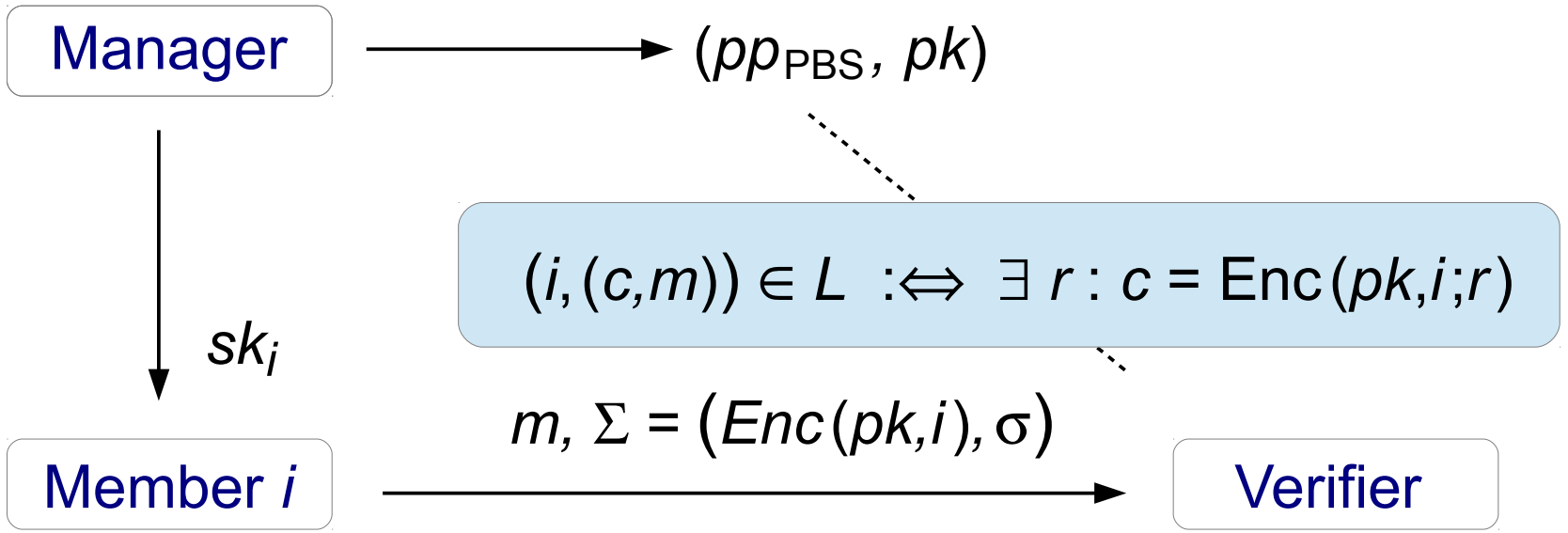
Σ can be opened: Σ —Manager→ i

CCA-secure group signatures



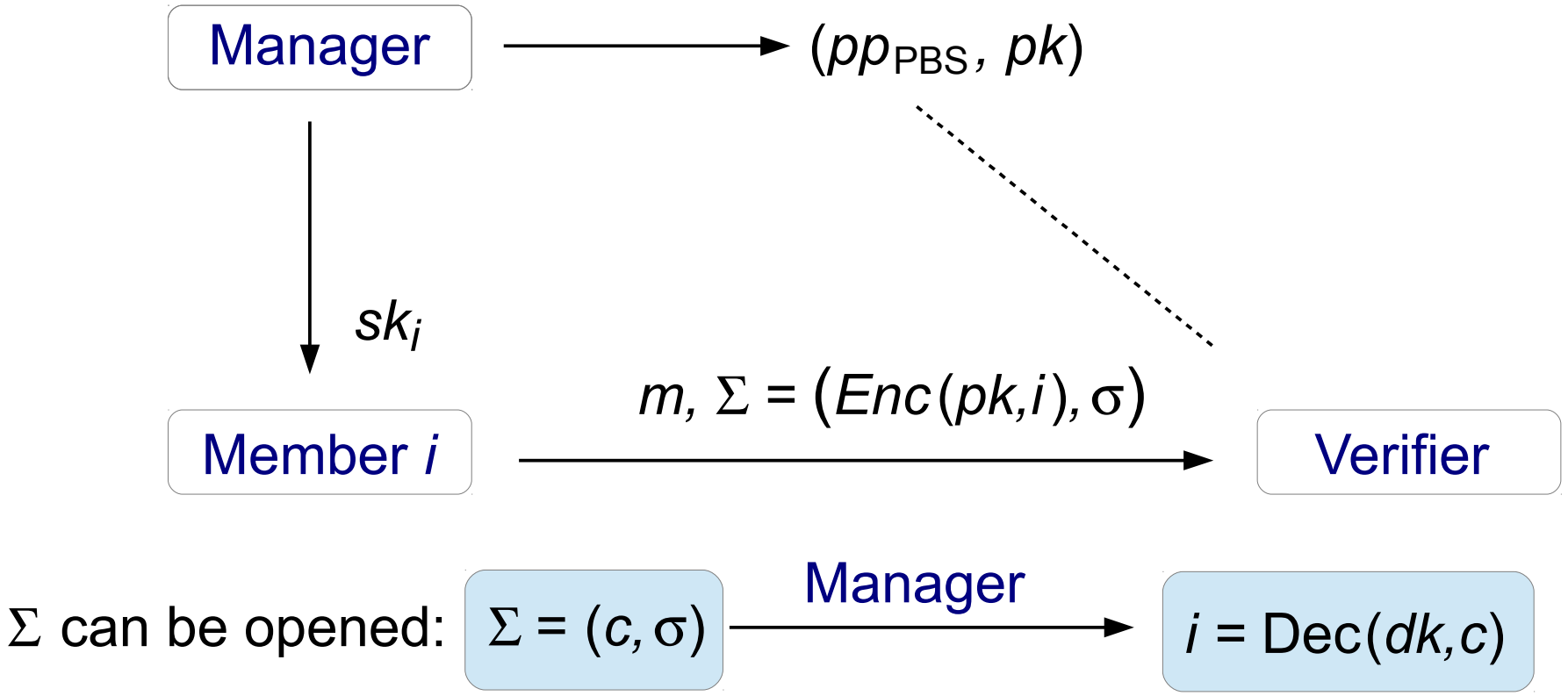
Σ can be opened: Σ —Manager→ i

CCA-secure group signatures



Σ can be opened: Σ —Manager→ i

CCA-secure group signatures



Other primitives from PBS

- Attribute-based signatures [MPR11]

Other primitives from PBS

- Attribute-based signatures [MPR11]
- Simulation-sound extractable NIZK proofs [Gro06]

Other primitives from PBS

- Attribute-based signatures [MPR11]
- Simulation-sound extractable NIZK proofs [Gro06]
- CPA-secure public-key encryption

Other primitives from PBS

- Attribute-based signatures [MPR11]
- Simulation-sound extractable NIZK proofs [Gro06]
- CPA-secure public-key encryption
- combining the above [Sah99]: CCA-secure encryption

  thus PBS $\Rightarrow$ group signatures

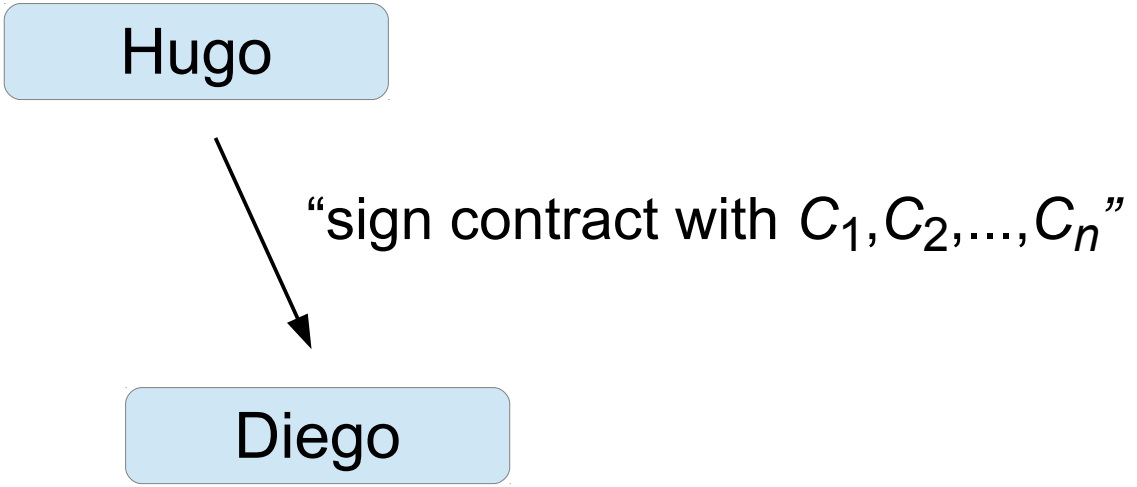
Delegatable PBS

Re-delegation

- Delegatable PBS
  - holding sk_p, derive $sk_{p'}$ for subpolicy p'
- Reflects hierarchies in organizations

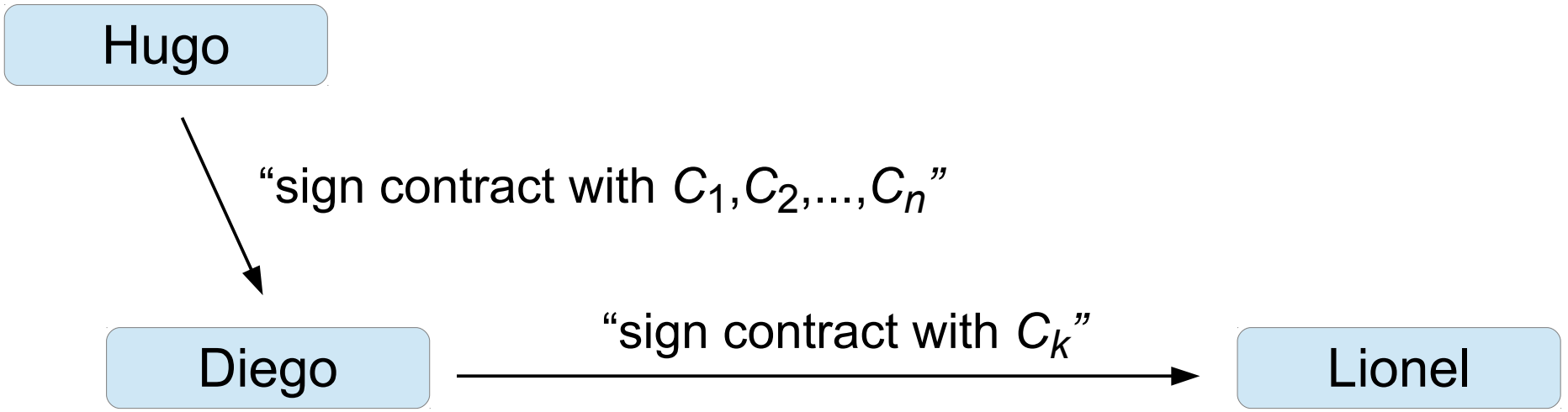
Re-delegation

- Delegatable PBS
  - holding sk_p, derive $sk_{p'}$ for subpolicy p'



Re-delegation

- Delegatable PBS
  - holding sk_p, derive $sk_{p'}$ for subpolicy p'



Conclusion

- New primitive, practically motivated
- Umbrella notion for previous primitives

Conclusion

- New primitive, practically motivated
- Umbrella notion for previous primitives
    - Definition
    - Constructions
    - Applications

Conclusion

- New primitive, practically motivated
- Umbrella notion for previous primitives

Open problems / future work

- Practical schemes for specific policy languages

Thank you