Evolving Secret Sharing made Short

Danilo Francati
Royal Holloway, University of London

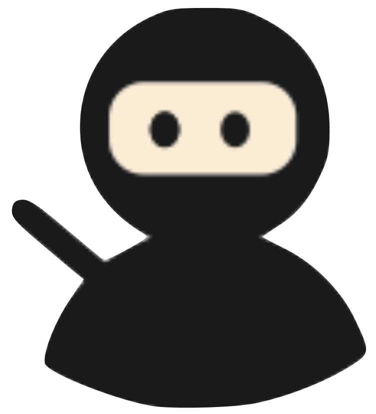
Daniele Venturi
Sapienza University of Rome

Secret Sharing

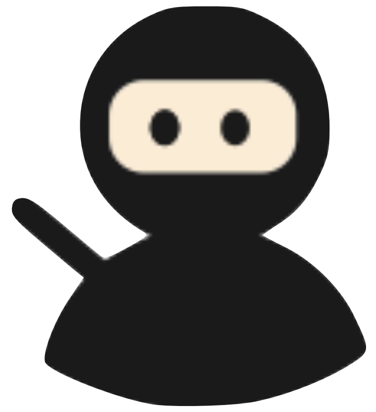
Dealer

secret s

P_1



P_2



P_3

Secret Sharing

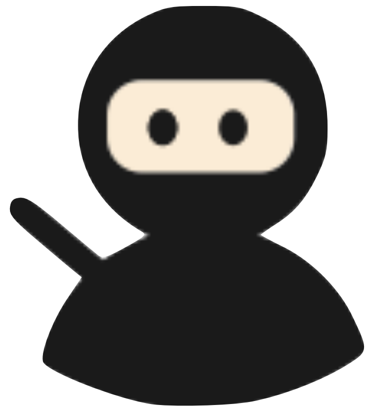
Dealer

secret s

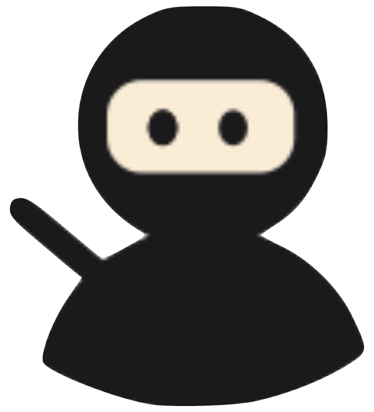
$\downarrow$

$\mathsf{Share}(s) = \sigma_1, \sigma_2, \sigma_3$

P_1

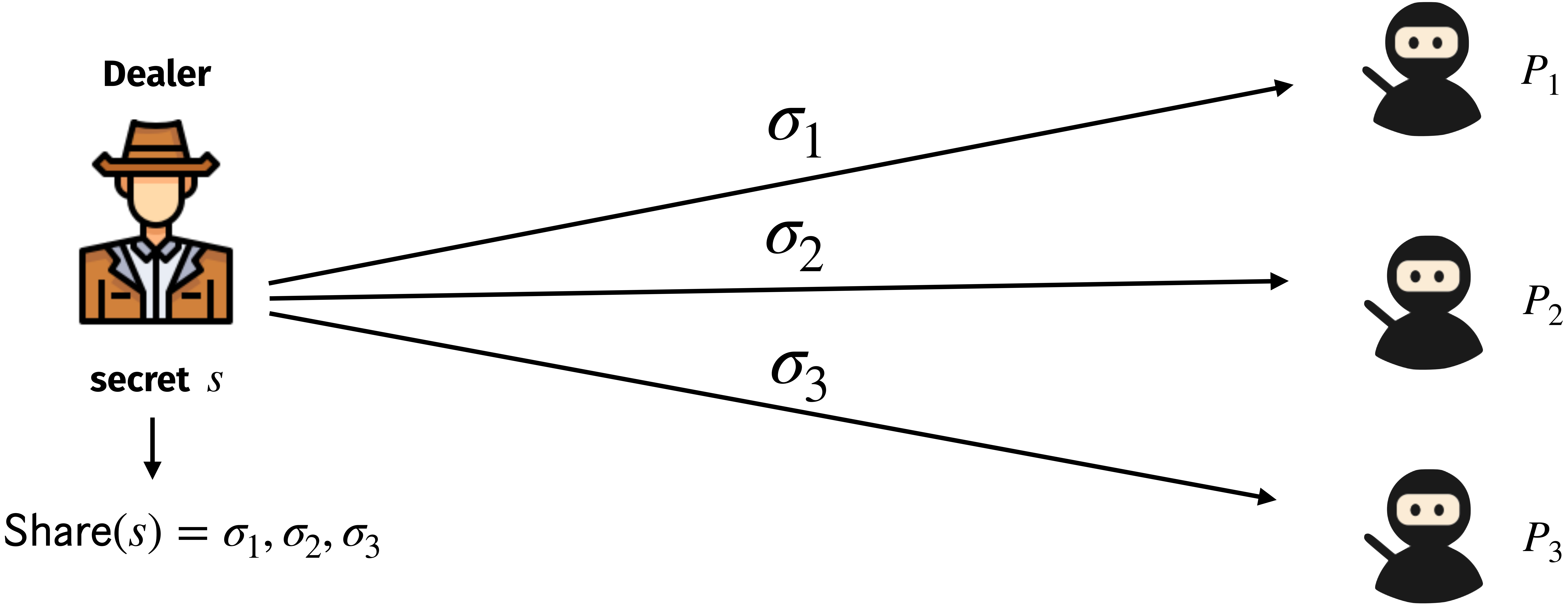


P_2



P_3

Secret Sharing



Secret Sharing

(Correctness)

$\mathsf{Reconstruct}(\sigma_1, \sigma_2) = s$

σ_1 P_1

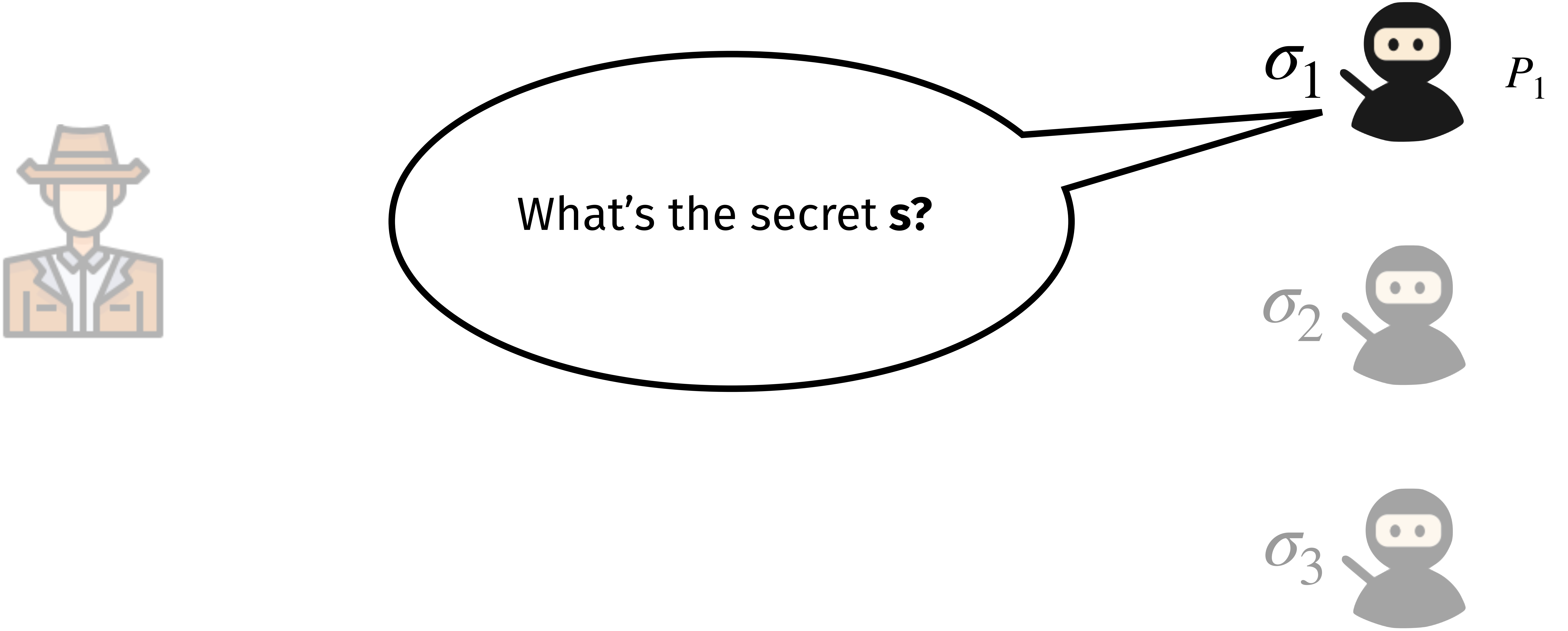
σ_2 P_2

σ_3 P_3

Correctness of t-out-of-n SS

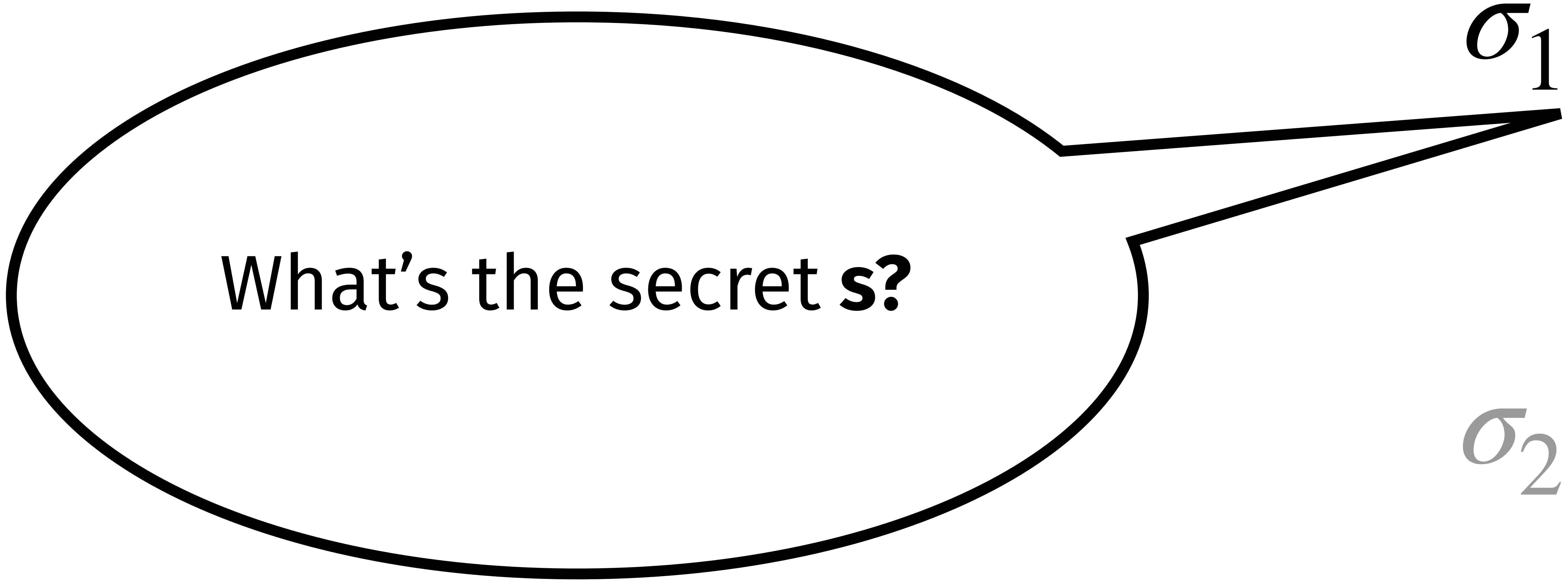
Any **t** parties can reconstruct the secret using their shares $\{\sigma_i\}$

Secret Sharing
(Security)



Secret Sharing

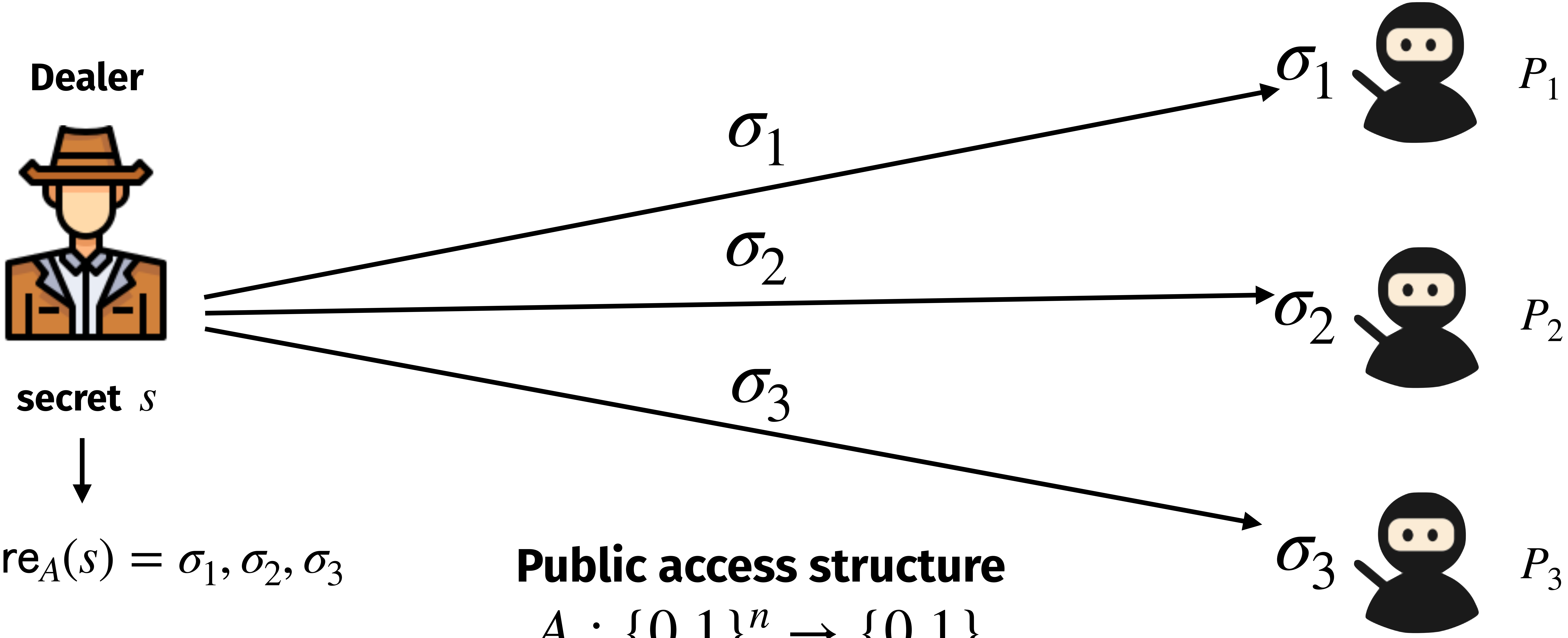
(Security)



Security of **t**-out-of-**n** SS
Any **t-1** parties infer no information about the secret **s**

Secret Sharing

(other access structures)



$\mathsf{Share}_A(s) = \sigma_1, \sigma_2, \sigma_3$

Public access structure

$A : \{0,1\}^n \to \{0,1\}$

Secret Sharing

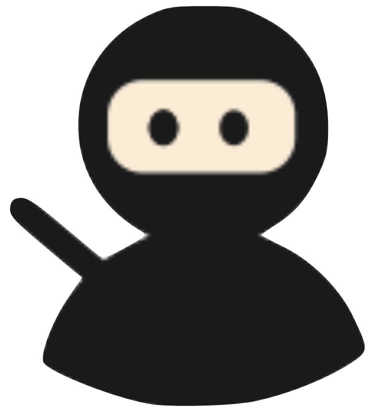
(other access structures)

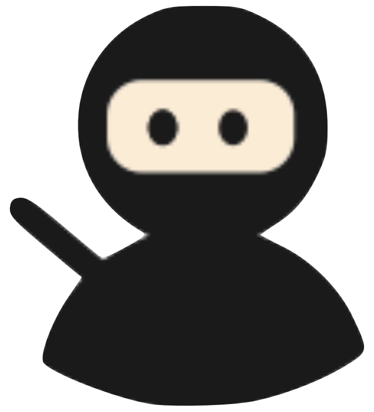
Dealer

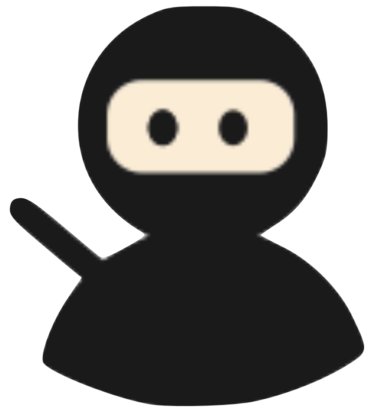
secret s

Correctness of SS w.r.t. A

Any **t** parties (encoded by string $x \in \{0,1\}^n$) s.t. $A(x) = 1$ reconstructs secret **s**

σ_1  P_1

σ_2  P_2

σ_3  P_3

Public access structure

$$A : \{0,1\}^n \to \{0,1\}$$

Secret Sharing
(other access structures)

Dealer

secret s

Correctness of SS w.r.t. A

Any **t** parties (encoded by string $x \in \{0,1\}^n$) s.t. $A(x) = 1$ reconstructs secret **s**

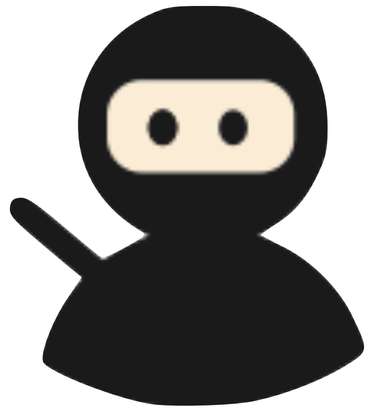
Security of SS w.r.t. A

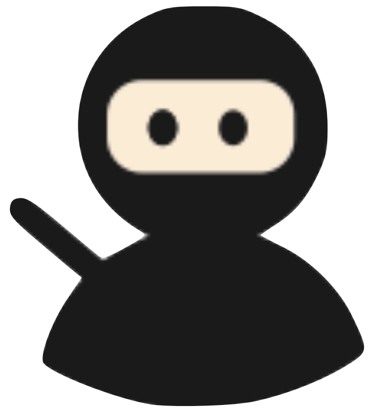
Any **t** parties (encoded by string $x \in \{0,1\}^n$) s.t. $A(x) = 0$ infer no information about the secret **s.**

Public access structure

$$A : \{0,1\}^n \rightarrow \{0,1\}$$

σ_1  P_1

σ_2  P_2

σ_3  P_3

Monotonicity

Monotone Access Structure A **(informal)**

If parties $\{P_1, P_2, P_3\}$ can reconstruct **(authorized set)** the secret **s**

$\Downarrow$

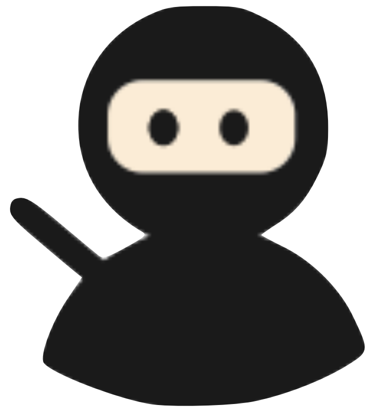
then parties $\{P_1, P_2, P_3, P_4\}$ **(superset of)** can reconstruct **(authorized set)** the secret **s**

Required to define **security** of **SS**

Evolving Secret Sharing

Dealer

secret s



P_1

Public access structure

$$A_1 : \{0,1\}^1 \rightarrow \{0,1\}$$

Evolving Secret Sharing

Dealer

secret s

$\downarrow$

$\mathsf{Share}_{A_1}(s) = \sigma_1$

P_1

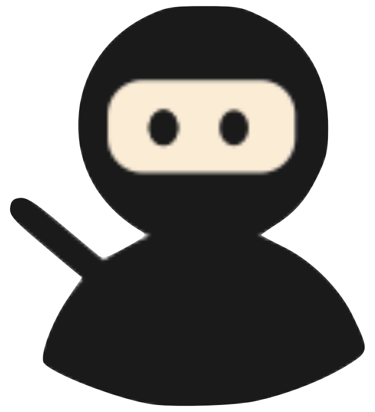
Public access structure

$A_1 : \{0,1\}^1 \rightarrow \{0,1\}$

Evolving Secret Sharing

Dealer

σ_1



P_1

secret s

$\downarrow$

$\mathsf{Share}_{A_1}(s) = \sigma_1$

Public access structure

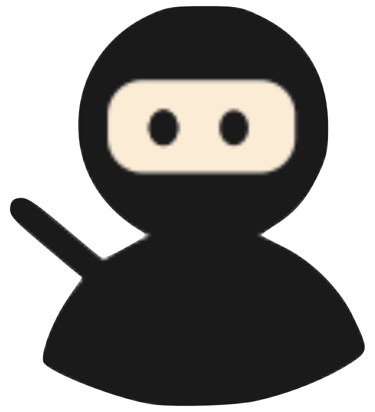
$A_1 : \{0,1\}^1 \to \{0,1\}$

Evolving Secret Sharing

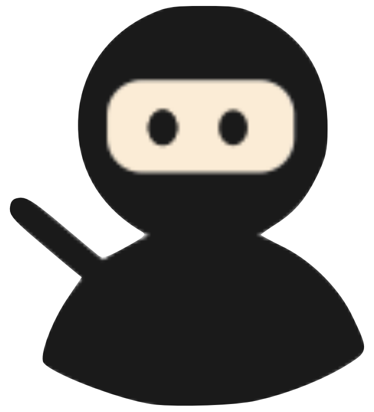
Dealer

secret s

σ_1



P_1



P_2

Public access structure

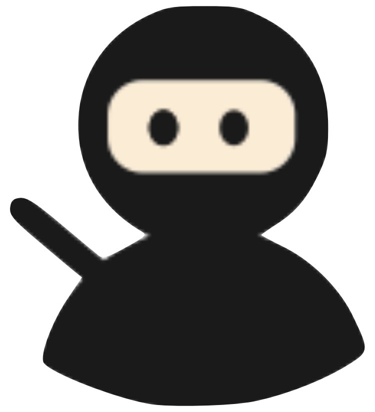
$$A_1 : \{0,1\}^1 \to \{0,1\}$$

Evolving Secret Sharing

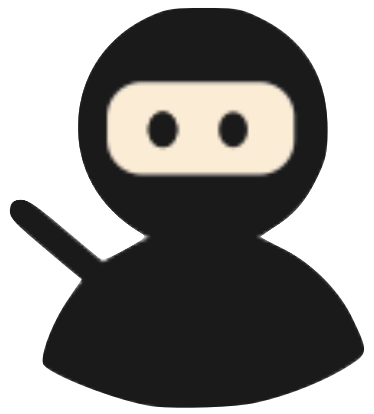
Dealer

secret s

σ_1



P_1



P_2

Public access structure

$$A_2 : \{0,1\}^2 \to \{0,1\}$$

~~**Public access structure**~~

$$\cancel{A_1 : \{0,1\}^1 \to \{0,1\}}$$

Evolving Secret Sharing

Dealer

secret s

$\downarrow$

$\mathsf{Share}_{A_2}(\{\sigma_i\}_{i\in[1]}, s) = \sigma_2$

Public access structure

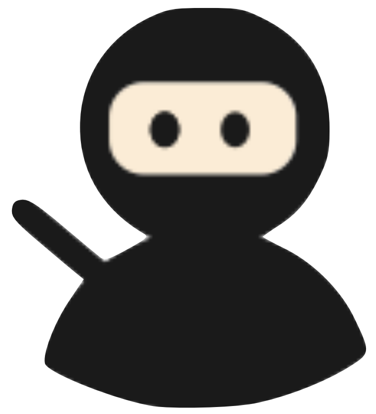
$A_2 : \{0,1\}^2 \rightarrow \{0,1\}$

~~**Public access structure**~~

$\cancel{A_1 : \{0,1\}^1 \rightarrow \{0,1\}}$

σ_1

P_1



P_2

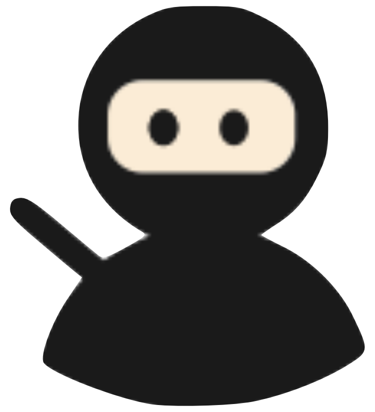
Evolving Secret Sharing

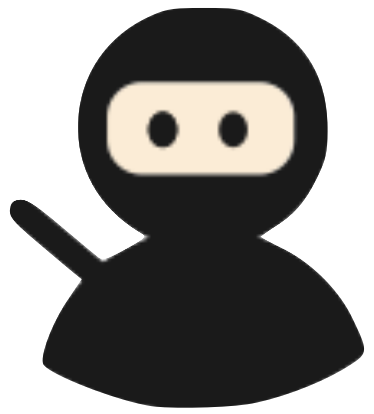
Dealer

secret s

$\downarrow$

$\mathsf{Share}_{A_2}(\{\sigma_i\}_{i\in[1]}, s) = \sigma_2$

σ_1  P_1

$\xrightarrow{\sigma_2}$  P_2

Public access structure

$$A_2 : \{0,1\}^2 \rightarrow \{0,1\}$$

~~**Public access structure**~~

$$\cancel{A_1 : \{0,1\}^1 \rightarrow \{0,1\}}$$

Evolving Secret Sharing

Dealer

secret s

$\downarrow$

$\text{Share}_{A_3}(\{\sigma_i\}_{i\in[2]}, s) = \sigma_3$

σ_3

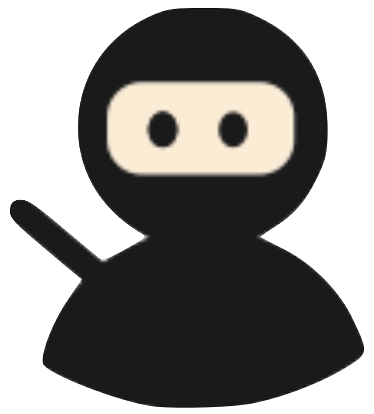
Public access structure

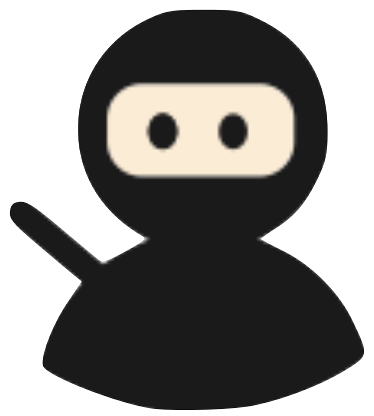
$A_3 : \{0,1\}^3 \to \{0,1\}$

~~**Public access structure**~~

$\cancel{A_2 : \{0,1\}^2 \to \{0,1\}}$

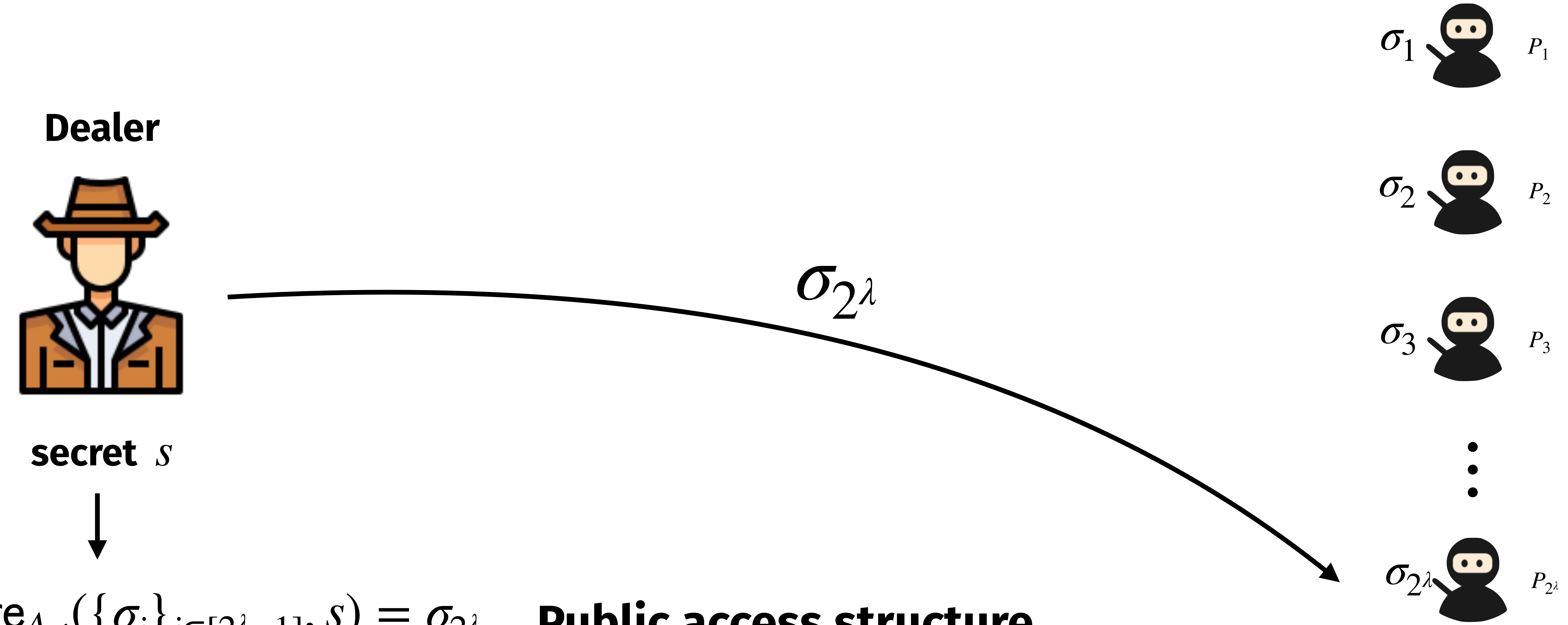
σ_1  P_1

σ_2  P_2

 P_3

Evolving Secret Sharing

(Information Theoretic Case)



$\mathsf{Share}_{A_{2^\lambda}}(\{\sigma_i\}_{i\in[2^\lambda-1]}, s) = \sigma_{2^\lambda}$

Public access structure

$A_{2^\lambda} : \{0,1\}^{2^\lambda} \to \{0,1\}$

Properties of Evolving Secret Sharing

Properties of Evolving Secret Sharing

Monotone Evolving Access Structure A **(informal)**

For every $i \in [n], A_i$ is a **monotone** access structure (as in the non-evolving case).

AND

For every $i \in [n-1]$, we have $A_i \subseteq A_{i+1}$

Properties of Evolving Secret Sharing

Monotone Evolving Access Structure A **(informal)**

For every $i \in [n], A_i$ is a **monotone** access structure (as in the non-evolving case).

AND

For every $i \in [n-1]$, we have $A_i \subseteq A_{i+1}$

Security/Correctness w.r.t. **Evolving** setting **(informal)**

Let $\{P_{i_1}, \ldots, P_{i_{t-1}}, P_n\}$ a set of **t** parties (encoded by the string $x \in \{0,1\}^n$).

Correctness: If $A_n(x) = 1 \Rightarrow$ Reconstruction is **possible**.

Security: if $A_n(x) = 0 \Rightarrow$ **No information revealed.**

Properties of Evolving Secret Sharing

Monotone Evolving Access Structure A **(informal)**

For every $i \in [n], A_i$ is a **monotone** access structure (as in the non-evolving case).

AND

For every $i \in [n-1]$, we have $A_i \subseteq A_{i+1}$

Security/Correctness w.r.t. **Evolving** setting **(informal)**

Let $\{P_{i_1}, \ldots, P_{i_{t-1}}, P_n\}$ a set of **t** parties (encoded by the string $x \in \{0,1\}^n$).

Correctness: If $A_n(x) = 1 \Rightarrow$ Reconstruction is **possible**.

Security: if $A_n(x) = 0 \Rightarrow$ **No information revealed.**

Share **size**

$|\sigma_n|$ can depend on the number of **current parties, e.g.,** $|\sigma_n| \in O(n) \cdot \text{poly}(\lambda)$

Properties of Evolving Secret Sharing

Monotone Evolving Access Structure A **(informal)**

For every $i \in [n], A_i$ is a **monotone** access structure (as in the non-evolving case).

AND

For every $i \in [n-1]$, we have $A_i \subseteq A_{i+1}$

Security/Correctness w.r.t. **Evolving** setting **(informal)**

Let $\{P_{i_1}, \ldots, P_{i_{t-1}}, P_n\}$ a set of **t** parties (encoded by the string $x \in \{0,1\}^n$).

Correctness: If $A_n(x) = 1 \Rightarrow$ Reconstruction is **possible**.

Security: if $A_n(x) = 0 \Rightarrow$ **No information revealed.**

Share **size**

$|\sigma_n|$ can depend on the number of **current parties, e.g.,** $|\sigma_n| \in O(n) \cdot \text{poly}(\lambda)$

Unknown evolution

The **evolution** of the access structure is **NOT known in advance** by the **dealer**.
(otherwise, evolving SS is trivial)

Properties of Evolving Secret Sharing

Monotone Evolving Access Structure A **(informal)**

For every $i \in [n], A_i$ is a **monotone** access structure (as in the non-evolving case).

AND

For every $i \in [n-1]$, we have $A_i \subseteq A_{i+1}$

Security/Correctness w.r.t. **Evolving** setting **(informal)**

Let $\{P_{i_1}, \ldots, P_{i_{t-1}}, P_n\}$ a set of **t** parties (encoded by the string $x \in \{0,1\}^n$).

Correctness: If $A_n(x) = 1 \Rightarrow$ Reconstruction is **possible**.

Security: if $A_n(x) = 0 \Rightarrow$ **No information revealed.**

Share **size**

$|\sigma_n|$ can depend on the number of **current parties, e.g.,** $|\sigma_n| \in O(n) \cdot \text{poly}(\lambda)$

Unknown evolution

The **evolution** of the access structure is **NOT known in advance** by the **dealer**.
(otherwise, evolving SS is trivial)

Our Work

Extend the notion of **Evolving** Secret Sharing to the **Computational Setting.**

Our Work

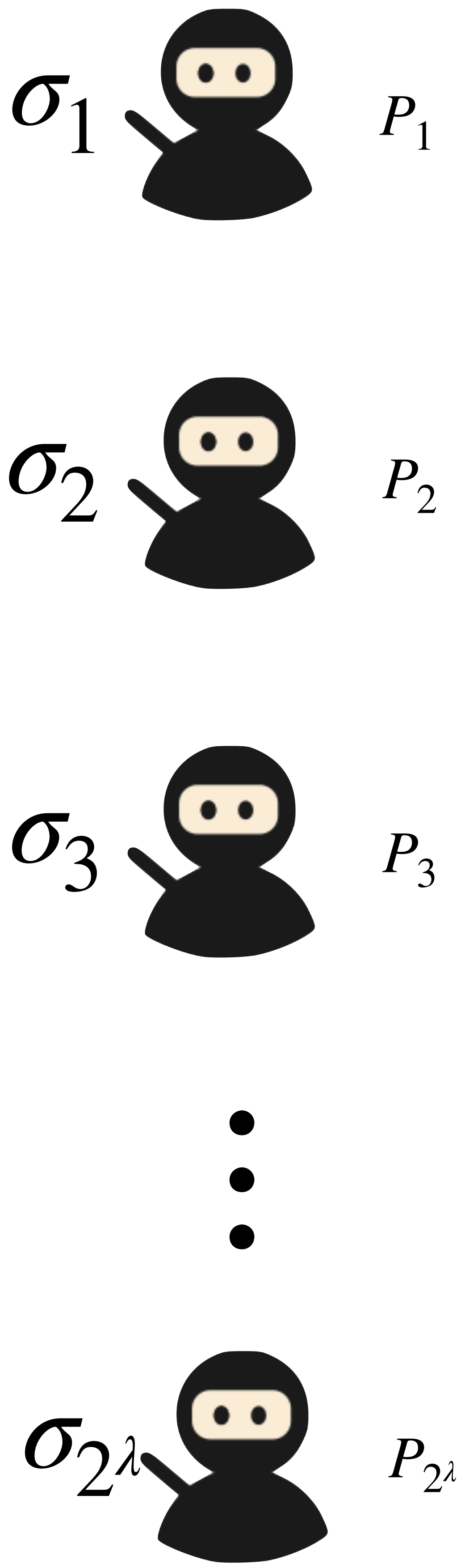
Extend the notion of **Evolving** Secret Sharing
to the **Computational Setting.**

Objectives	
1.	New **Evolving** Secret Sharing schemes for specific **Evolving** Access Structures.
2.	Reducing the share size of these **Evolving** Secret Sharing schemes (we can leverage **computationally** secure primitives 😁).

"Computational" Evolution

Dealer

σ_{2^λ}



secret s

$\downarrow$

$\text{Share}_{A_{2^\lambda}}(\{\sigma_i\}_{i\in[2^\lambda-1]}, s) = \sigma_{2^\lambda}$

Public access structure

$A_{2^\lambda} : \{0,1\}^{2^\lambda} \to \{0,1\}$

"Computational" Evolution

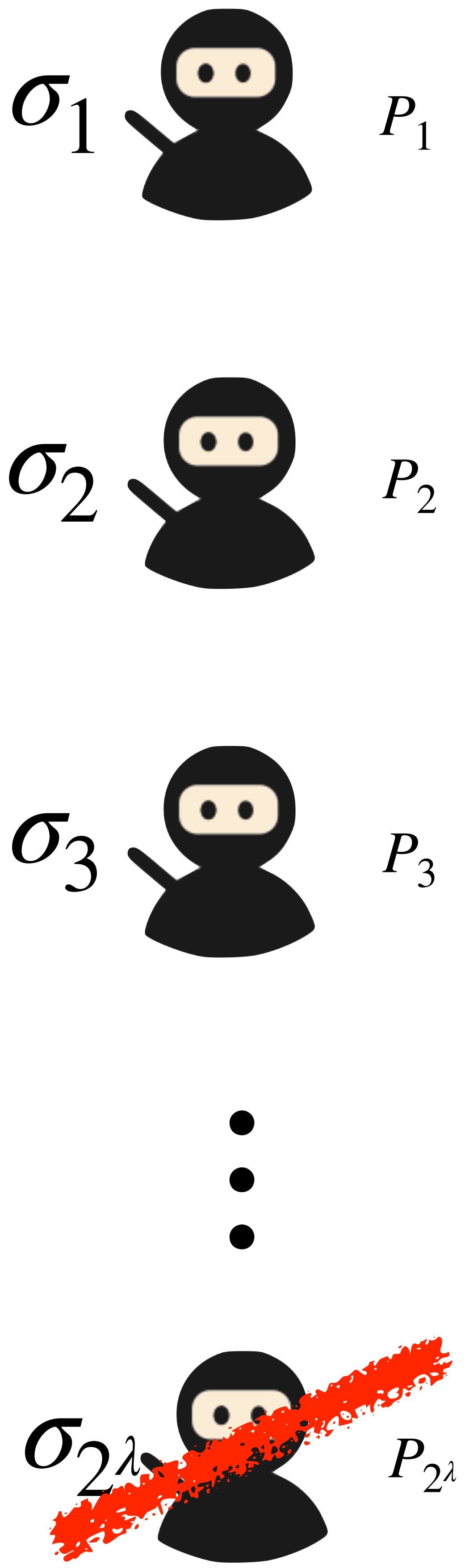
Dealer

secret s

$\downarrow$

$\cancel{\text{Share}_{A_{2^\lambda}}(\{\sigma_i\}_{i\in[2^\lambda-1]}, s) = \sigma_{2^\lambda}}$

$\cancel{\sigma_{2^\lambda}}$



~~Public access structure~~

$\cancel{A_{2^\lambda}: \{0,1\}^{2^\lambda}} \to \{0,1\}$

“Computational” Evolution

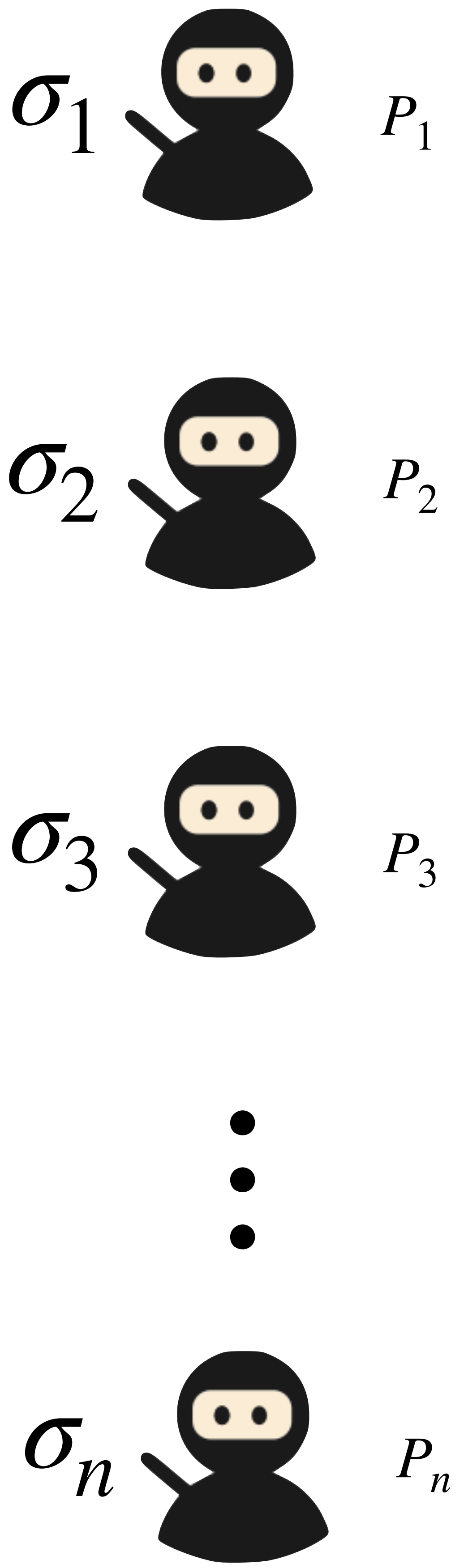
Dealer

secret s

$\downarrow$

$\mathsf{Share}_{A_n}(\{\sigma_i\}_{i\in[n-1]}, s) = \sigma_n$

σ_n



Public access structure

$$A_n : \{0,1\}^n \rightarrow \{0,1\}$$

"Computational" Evolution

Unknown polynomial bound

The dealer **knows** n will be **polynomial** but no guarantees on its upper bound.

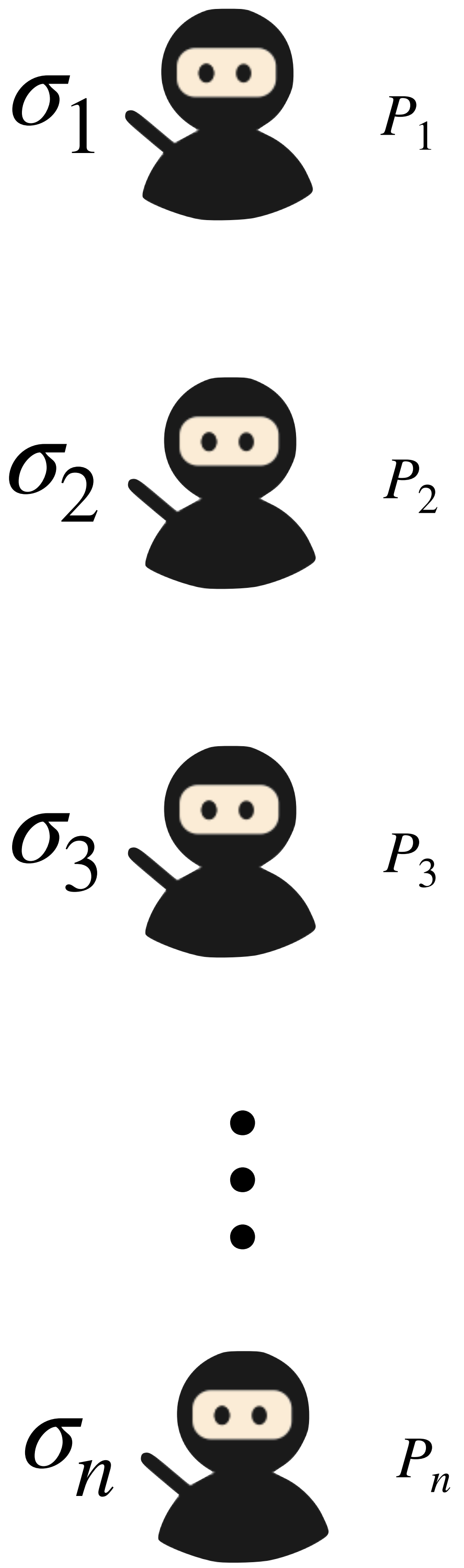
Dealer

secret s

$\downarrow$

$\mathsf{Share}_{A_n}(\{\sigma_i\}_{i\in[n-1]}, s) = \sigma_n$

σ_n



Public access structure

$A_n : \{0,1\}^n \to \{0,1\}$

“Computational” Representation

“Computational” Representation

Information Theoretic representation of the **Evolving** Access Structure A

An **evolving** access structure is usually represented as incrementally defined sets:

$$A_1 \subseteq A_2 \subseteq A_3 \subseteq A_4 \subseteq \ldots \subseteq A_n$$

without caring if these sets have an **efficient (polynomial) representation**.

“Computational” Representation

Information Theoretic representation of the **Evolving** Access Structure A

An **evolving** access structure is usually represented as incrementally defined sets:

$$A_1 \subseteq A_2 \subseteq A_3 \subseteq A_4 \subseteq \ldots \subseteq A_n$$

without caring if these sets have an **efficient (polynomial) representation**.

(having an efficient representation is important for the computational case)

<u>**Examples of efficient representable access structures**</u>

“Computational” Representation

Information Theoretic representation of the Evolving Access Structure A

An **evolving** access structure is usually represented as incrementally defined sets:

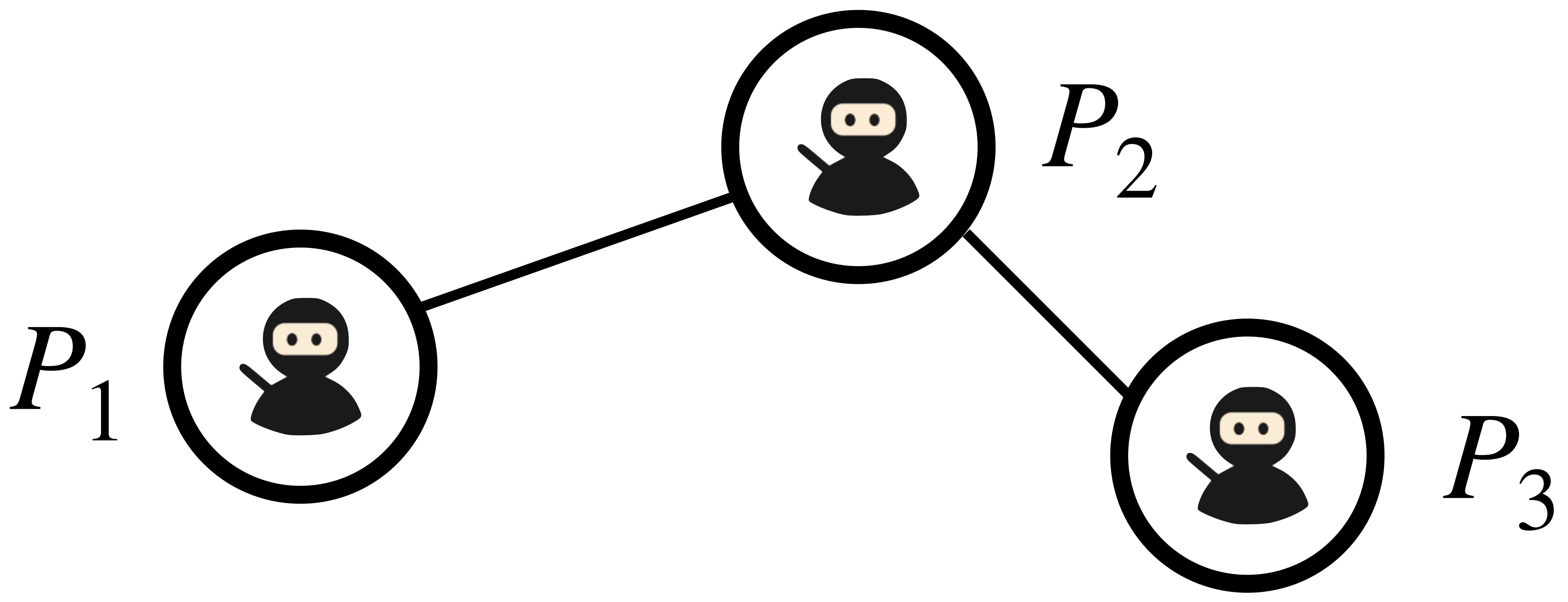
$$A_1 \subseteq A_2 \subseteq A_3 \subseteq A_4 \subseteq \ldots \subseteq A_n$$

without caring if these sets have an **efficient (polynomial) representation**.

(having an efficient representation is important for the computational case)

Examples of efficient representable access structures

Graphs



Parties $\{P_i, P_j\}$ are **authorized** (i.e., $A_n(P_i, P_j) = 1$)

if **edge** $P_i \rightarrow P_j$ **exists**

"Computational" Representation

Information Theoretic representation of the **Evolving** Access Structure $\mathcal{A}$

An **evolving** access structure is usually represented as incrementally defined sets:

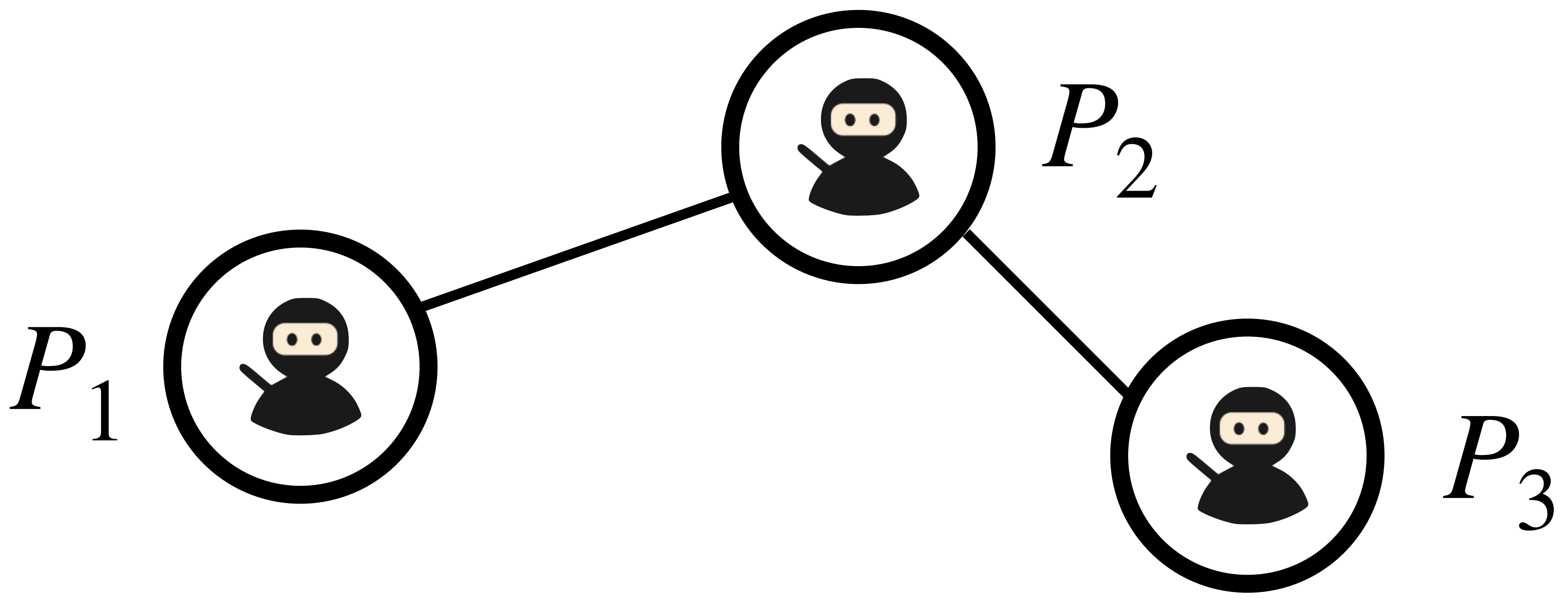
$$A_1 \subseteq A_2 \subseteq A_3 \subseteq A_4 \subseteq \ldots \subseteq A_n$$

without caring if these sets have an **efficient (polynomial) representation**.

(having an efficient representation is important for the computational case)

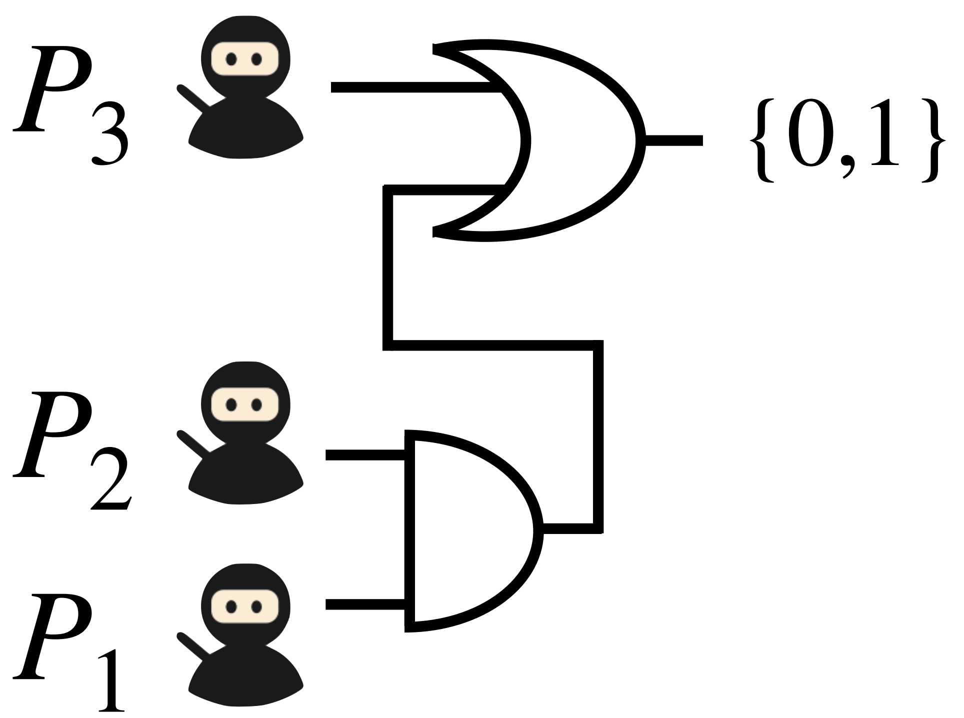
Examples of efficient representable access structures

Graphs



Parties $\{P_i, P_j\}$ are **authorized** (i.e., $A_n(P_i, P_j) = 1$) if **edge** $P_i \rightarrow P_j$ **exists**

Circuits

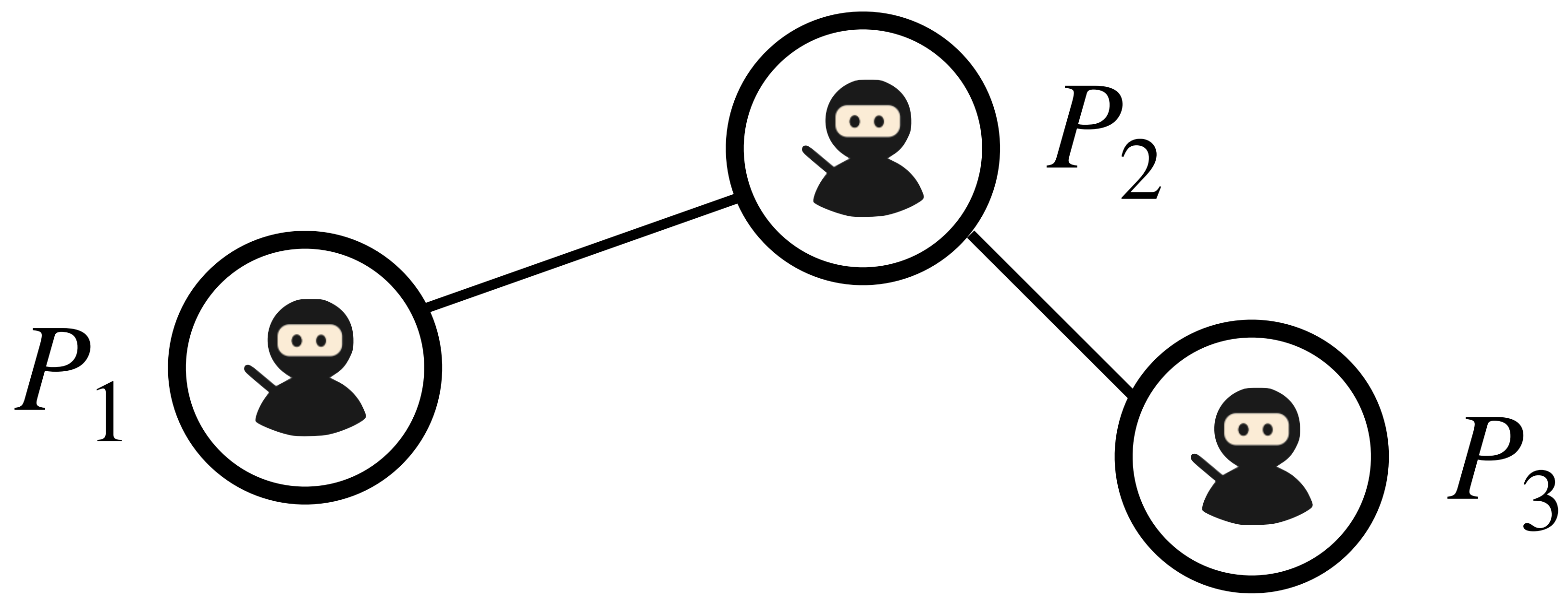


Parties $\{P_{i_j}\}_{j\in[t]}$ are **authorized** if the circuit evaluates to 1

How does the Representation evolves?

How does the Representation evolves?

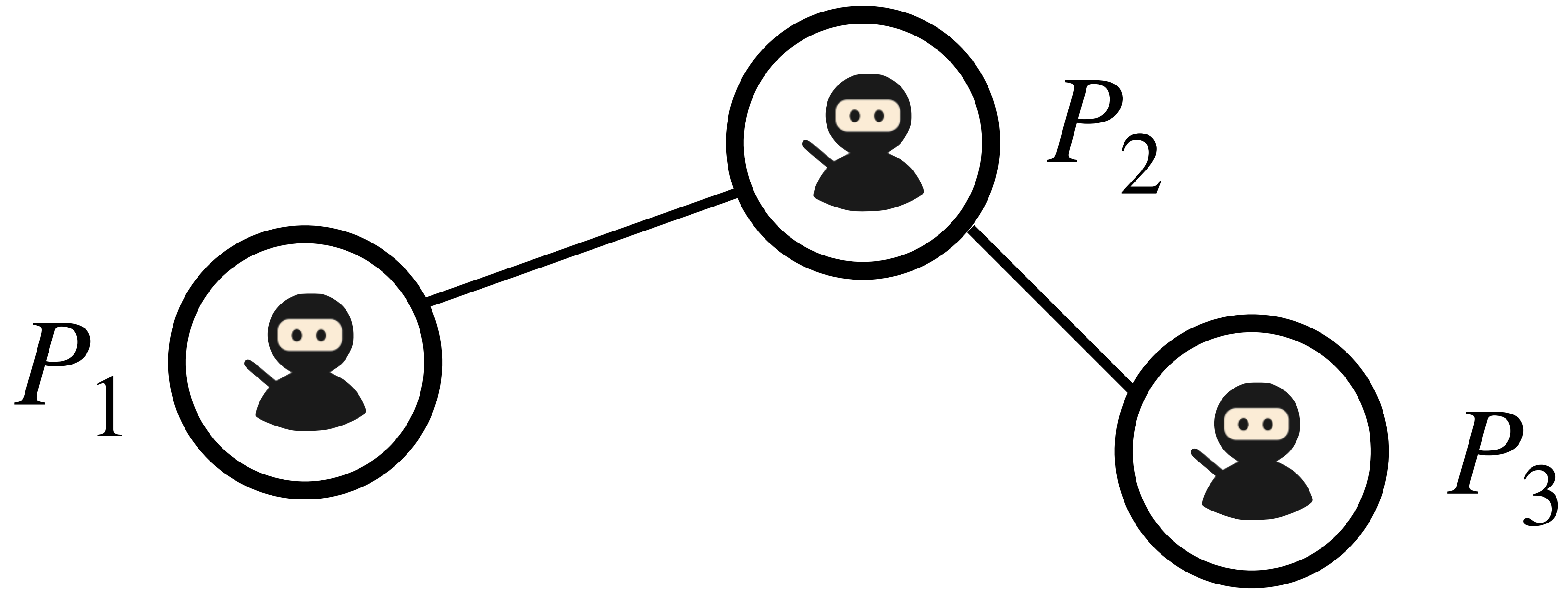
A_3



Graphs

How does the Representation evolves?

A_3



Graphs

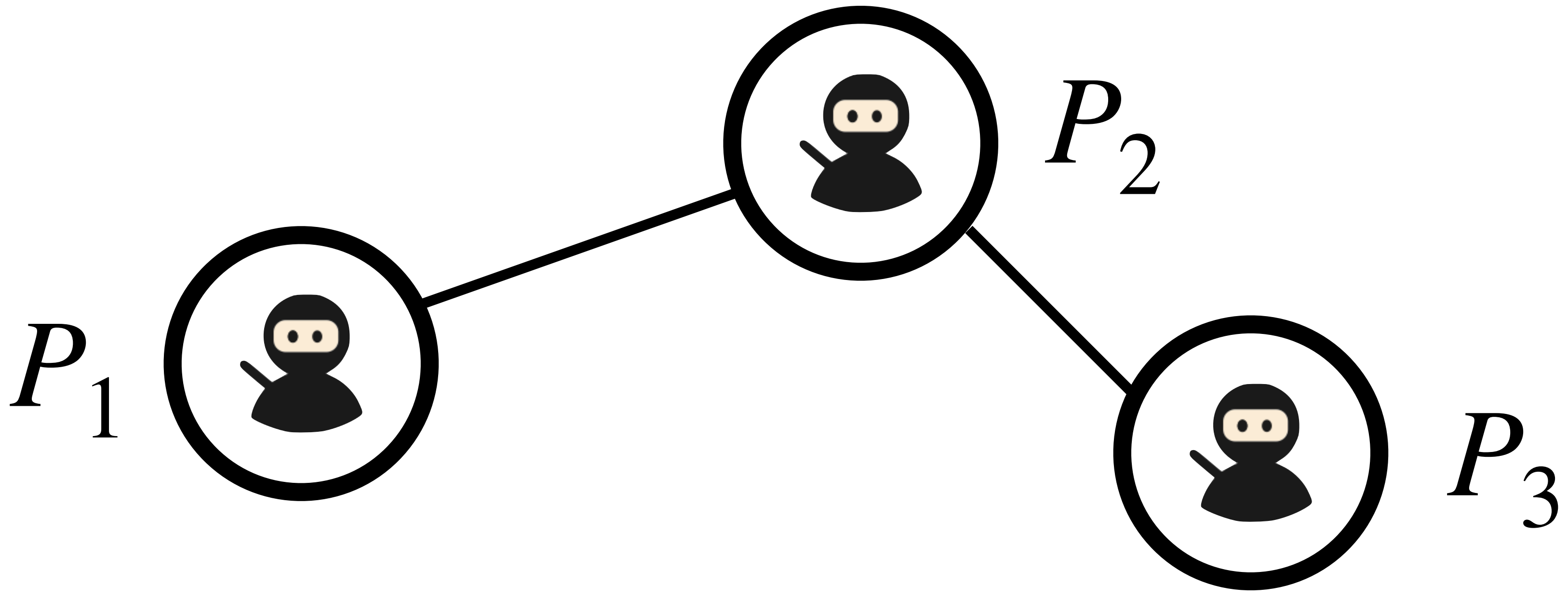
$\Longrightarrow$

A_4

- Can we add a new node?
- Can we add new edges? Edges between new or old nodes?
- Can we remove edges?

How does the Representation evolves?

A_3



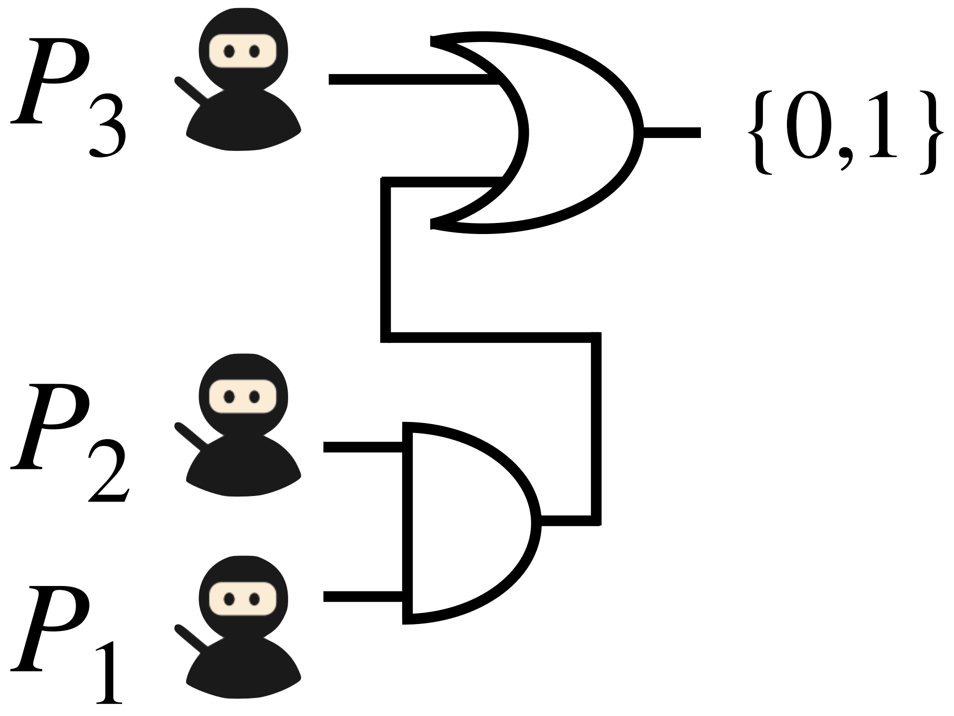
Graphs

Circuits

A_4

- Can we add a new node?
- Can we add new edges? Edges between new or old nodes?
- Can we remove edges?

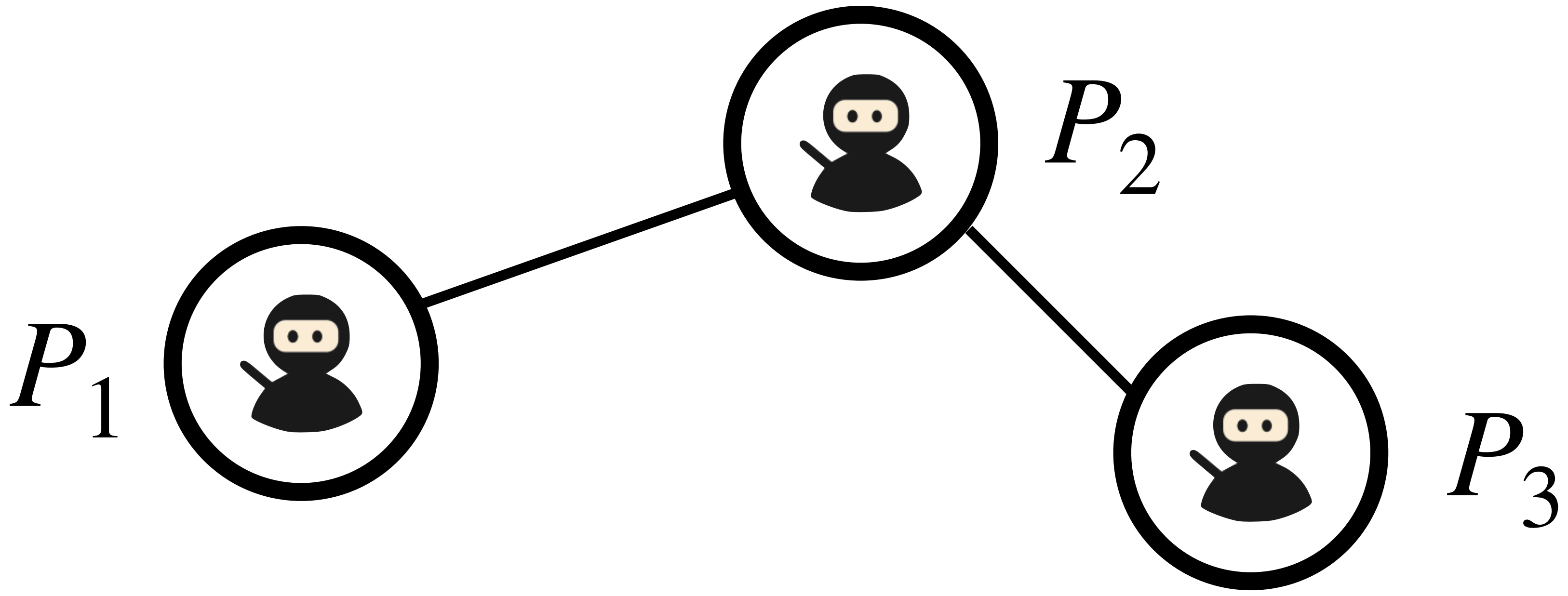
A_3



How does the Representation evolves?

Graphs

A_3

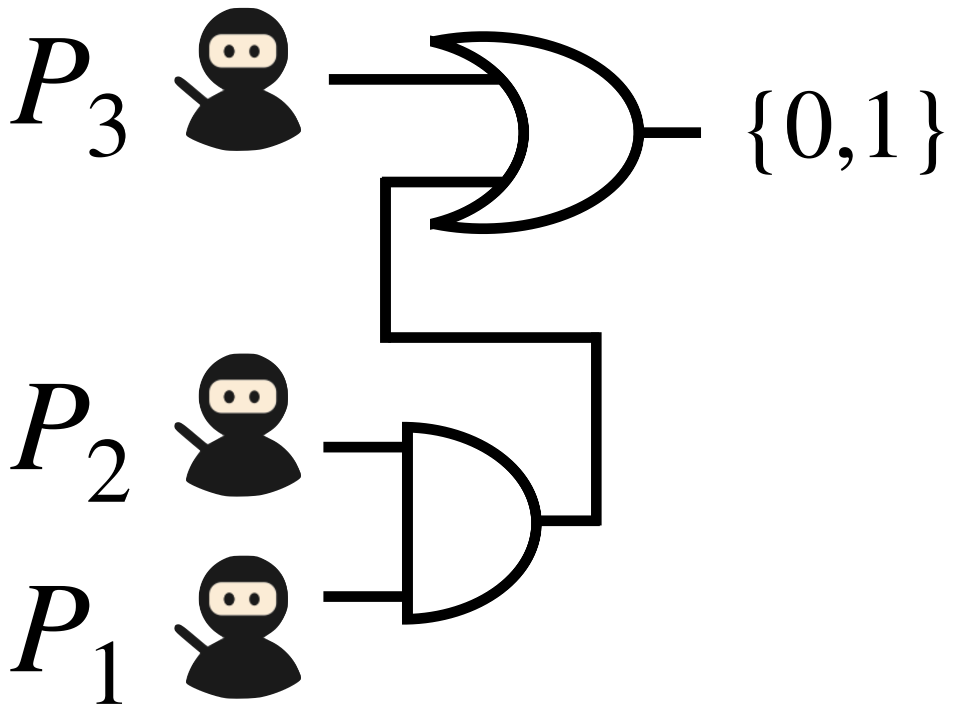


A_4

- Can we add a new node?
- Can we add new edges? Edges between new or old nodes?
- Can we remove edges?

Circuits

A_3



A_4

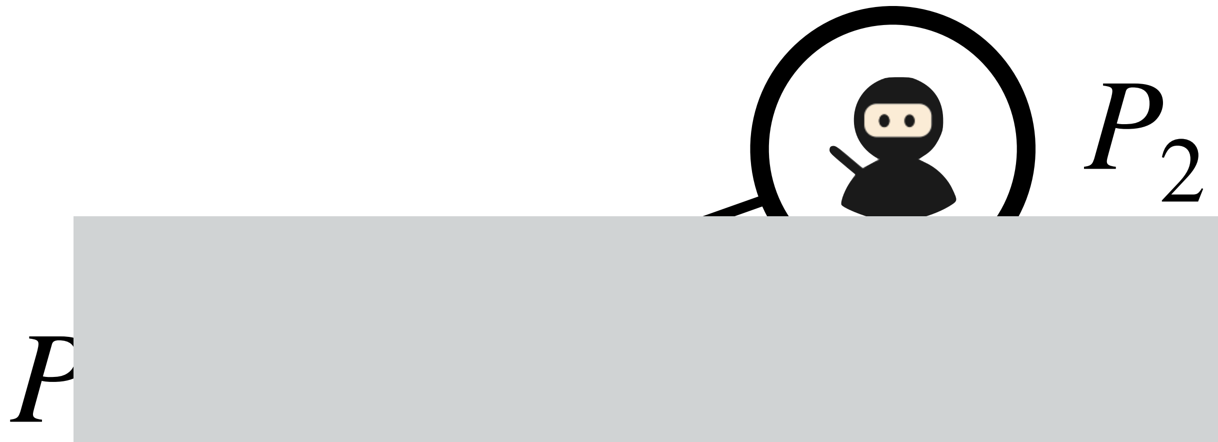
- Can we add new inputs wires?
- Can we add new AND gates? After which gate (e.g., OR, AND)?
- Can we remove old gates?

How does the Representation evolves?

A_3

Graphs

A_4



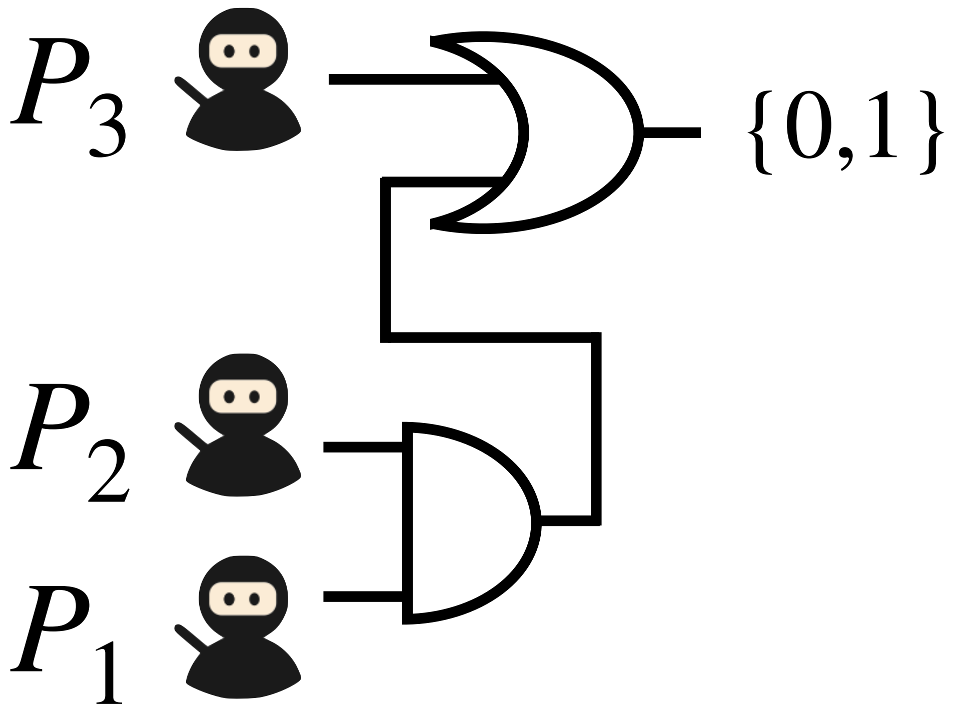
- Can we add a new node?
- Can we add new edges? Edges

Evolution of representation

Determined by two properties:

Monotonicity + *<u>Rigidity</u>*

(next slide)



- Can we add new inputs wires?
- Can we add new AND gates? After which gate (e.g., OR, AND)?
- Can we remove old gates?

Rigidity

Setting: No **CRS +** Shares of old parties remain **unchanged**

Rigidity

Setting: No **CRS +** Shares of old parties remain **unchanged**

Rigid Evolving Access Structure **(informal)**

If a set $U = \{P_{i_1}, \ldots, P_{i_t}\} \subseteq [n-1]$ is **unathorized** w.r.t. A_{n-1}

$\Downarrow$

the same set U is **unathorized** w.r.t. A_n

Rigidity

Setting: No **CRS +** Shares of old parties remain **unchanged**

Rigid Evolving Access Structure **(informal)**

If a set $U = \{P_{i_1}, \ldots, P_{i_t}\} \subseteq [n-1]$ is **unathorized** w.r.t. A_{n-1}

$$\Downarrow$$

the same set U is **unathorized** w.r.t. A_n

!! TAKEAWAY !!

After the arrival of n-th party (which defines the new access structure A_n),
the newly inserted **authorized sets MUST** contain P_n, i.e.,
for every $X \in A_n \backslash A_{n-1}$, we have $n \in X$.

Evolving Bipartite Graphs

(Projective PRGs)

Evolving Bipartite Graphs

(Projective PRGs)

$$\text{PRG} : \{0,1\}^n \to \{0,1\}^m$$

Evolving Bipartite Graphs

(Projective PRGs)

$$\mathsf{PRG} : \{0,1\}^n \to \{0,1\}^m$$

$$\mathsf{Setup}(1^\lambda, 1^m) \to \mathsf{msk}$$ **(the seed)**

Evolving Bipartite Graphs

(Projective PRGs)

$$\mathsf{PRG} : \{0,1\}^n \to \{0,1\}^m$$

$\mathsf{Setup}(1^\lambda, 1^m) \to \mathsf{msk}$ **(the seed)**

$\mathsf{KeyGen}(\mathsf{msk}, T \subseteq [m]) \to \alpha_T$ **(the projective seed)**

Evolving Bipartite Graphs

(Projective PRGs)

$$\mathsf{PRG} : \{0,1\}^n \to \{0,1\}^m$$

$$\mathsf{Setup}(1^\lambda, 1^m) \to \mathsf{msk}$$ **(the seed)**

$$\mathsf{KeyGen}(\mathsf{msk}, T \subseteq [m]) \to \alpha_T$$ **(the projective seed)**

Correctness
$\mathsf{Eval}(\mathsf{msk}) \to y \in \{0,1\}^m$
$\mathsf{Eval}(\alpha_T) \to y \in \{0,1\}^{\lvert T \rvert}$

Evolving Bipartite Graphs

(Projective PRGs)

$$\text{PRG} : \{0,1\}^n \to \{0,1\}^m$$

$\text{Setup}(1^\lambda, 1^m) \to \text{msk}$ **(the seed)**

$\text{KeyGen}(\text{msk}, T \subseteq [m]) \to \alpha_T$ **(the projective seed)**

Correctness
$\text{Eval}(\text{msk}) \to y \in \{0,1\}^m$
$\text{Eval}(\alpha_T) \to y \in \{0,1\}^{\lvert T \rvert}$

Identical to $\text{PRG}(\text{msk})$

Evolving Bipartite Graphs

(Projective PRGs)

$$\mathsf{PRG} : \{0,1\}^n \to \{0,1\}^m$$

$\mathsf{Setup}(1^\lambda, 1^m) \to \mathsf{msk}$ **(the seed)**

$\mathsf{KeyGen}(\mathsf{msk}, T \subseteq [m]) \to \alpha_T$ **(the projective seed)**

Correctness
$\mathsf{Eval}(\mathsf{msk}) \to y \in \{0,1\}^m$
$\mathsf{Eval}(\alpha_T) \to y \in \{0,1\}^{\lvert T \rvert}$

Identical to $\mathsf{PRG}(\mathsf{msk})\,|_T$**, i.e., the PRG output restricted on indexes** T

Evolving Bipartite Graphs
(Projective PRGs)

$$\text{PRG} : \{0,1\}^n \to \{0,1\}^m$$

$\text{Setup}(1^\lambda, 1^m) \to \text{msk}$ **(the seed)**

$\text{KeyGen}(\text{msk}, T \subseteq [m]) \to \alpha_T$ **(the projective seed)**

Security

Given α_T and $\text{Eval}(\text{msk}) \to y$, the y's bits associated to indexes $[m] \backslash T$ **(where $T = \{i_1, \ldots, i_k\}$)** are **pseudorandom.**

Evolving Bipartite Graphs
(Projective PRGs)

$$\mathsf{PRG} : \{0,1\}^n \to \{0,1\}^m$$

$\mathsf{Setup}(1^\lambda, 1^m) \to \mathsf{msk}$ **(the seed)**

$\mathsf{KeyGen}(\mathsf{msk}, T \subseteq [m]) \to \alpha_T$ **(the projective seed)**

Succinctness

α_T must be **succinct (i.e., sublinear)** compared to $|T|$.

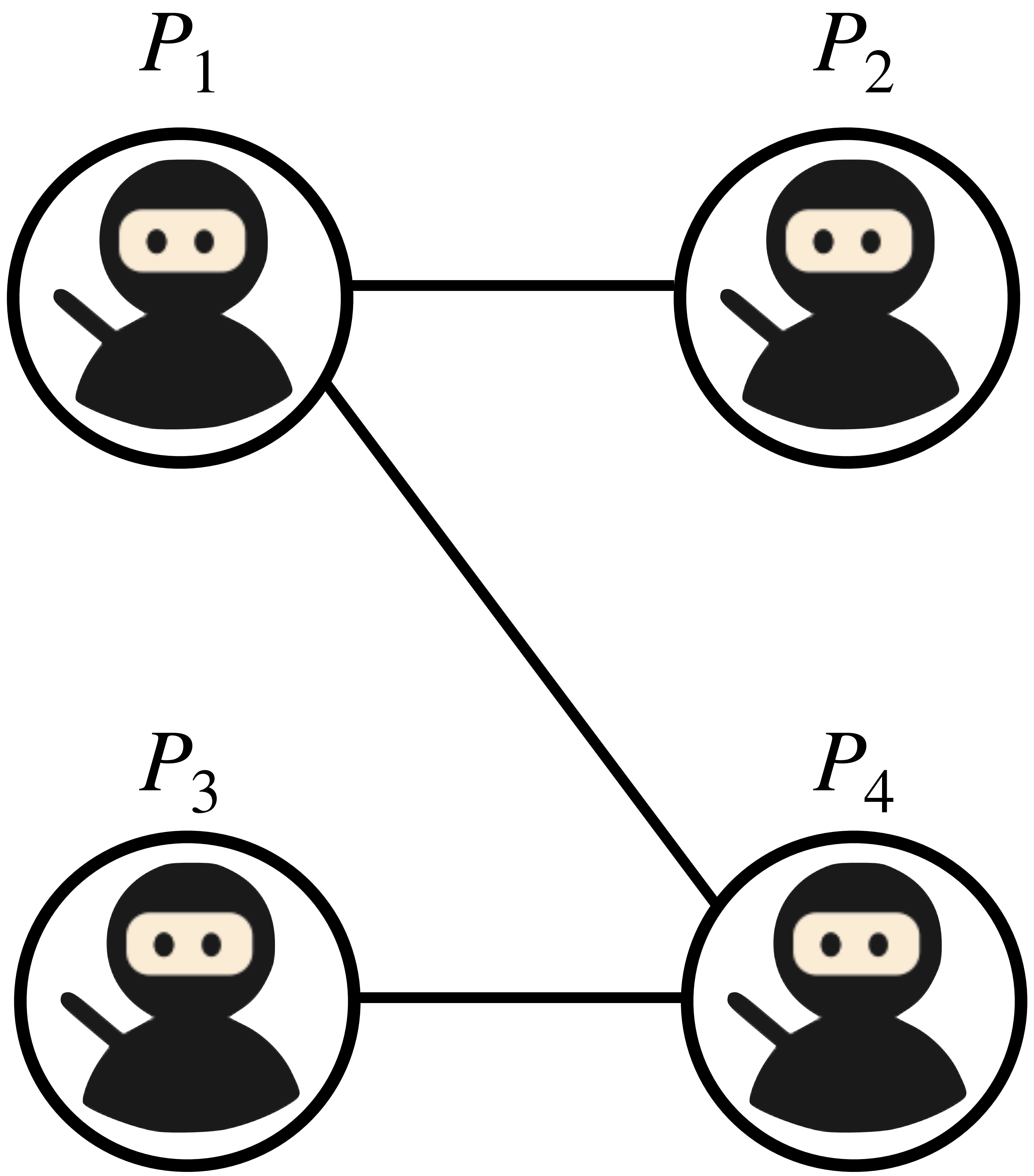
[A] Gave a construction based on **RSA** where $|\alpha_T| \in \mathsf{poly}(\lambda)$.

[A] Applebaum, Benny, Amos Beimel, Yuval Ishai, Eyal Kushilevitz, Tianren Liu, Vinod Vaikuntanathan.
"Succinct computational secret sharing."
STOC 23.

Evolving Bipartite Graphs
(Formalisation)

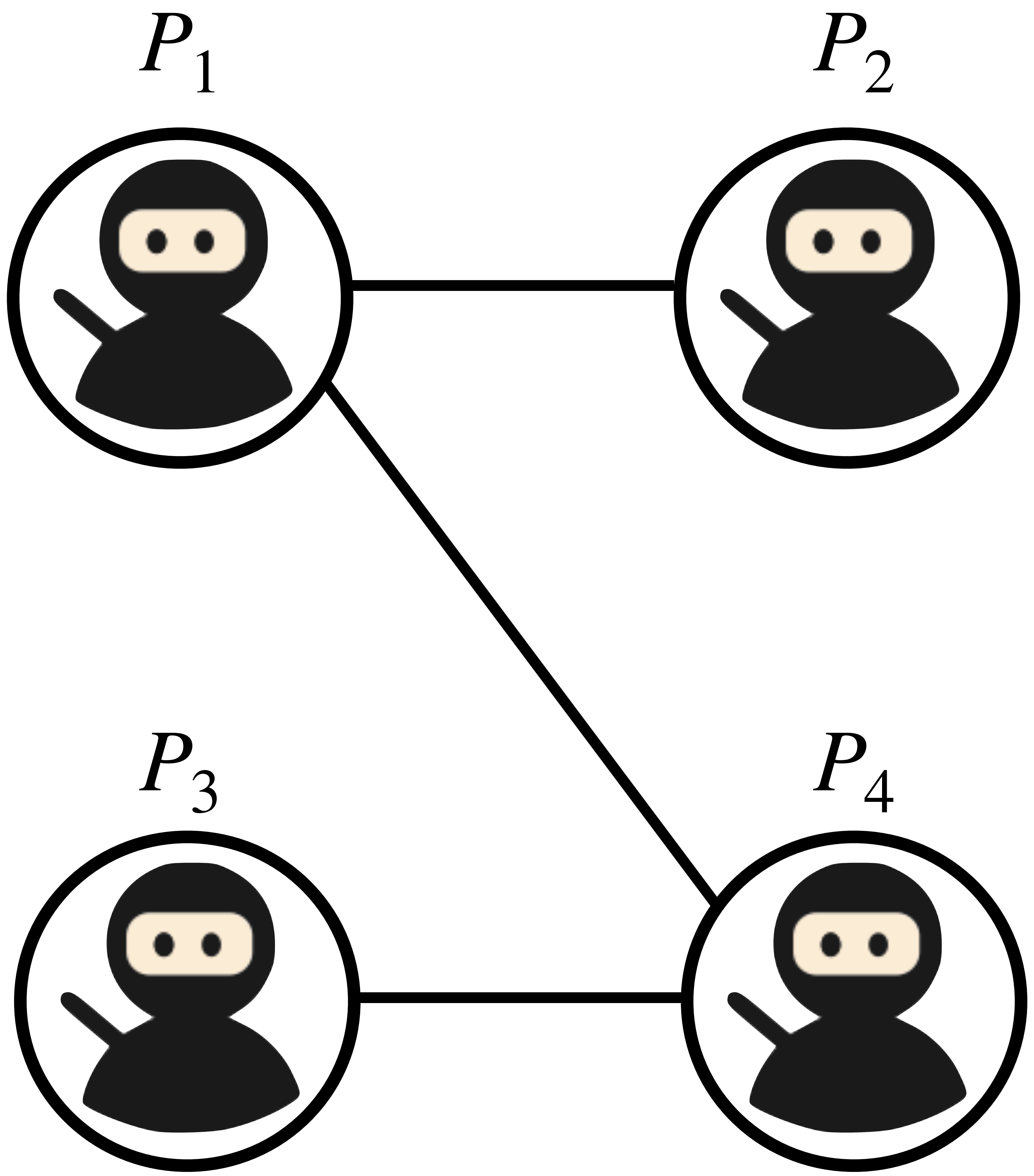
(Evolving) Access structure

$\{P_i, P_j\}$ are **authorized** if there is an **edge** between nodes i and j.



Evolving Bipartite Graphs
(Formalisation)

Monotonicity
By definition.



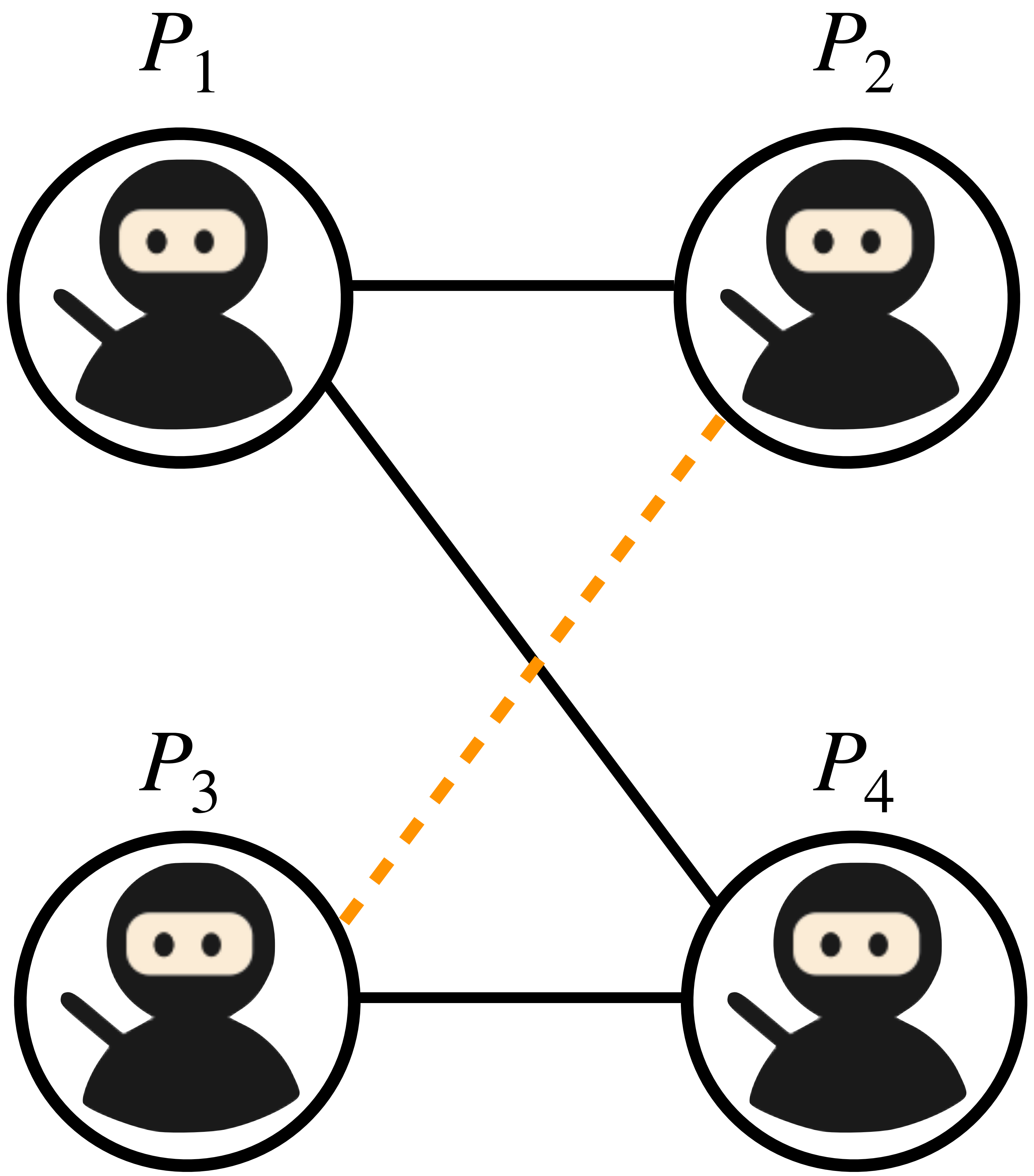
Evolving Bipartite Graphs
(Formalisation)

Monotonicity

By definition.

Rigidity

We can **add edges** only if one **end hits** the **newly introduced party, i.e.,** no new edges between **old** nodes.



Evolving Bipartite Graphs

(Construction from Projective PRGs)

Dealer

secret s

P_1

$\mathrm{PRG}^{\mathrm{left}}$

$\mathrm{PRG}^{\mathrm{right}}$

Evolving Bipartite Graphs

(Construction from Projective PRGs)

Dealer

secret s

- Compute $\mathsf{Eval}(\mathsf{msk}^{\text{left}}) \rightarrow y^{\text{left}}$
- Compute $\mathsf{KeyGen}(\mathsf{msk}^{\text{right}}, \{\perp\}) \rightarrow \alpha^{\text{right}}_{\{\perp\}}$
- Set $\sigma_1 = (y^{\text{left}}|_{\{1\}} \oplus s, \alpha^{\text{right}}_{\{\perp\}})$

P_1

$\mathsf{PRG}^{\text{left}}$

$\mathsf{PRG}^{\text{right}}$

Evolving Bipartite Graphs

(Construction from Projective PRGs)

Dealer

secret s

$\sigma_1 = (y^{\text{left}}|_{\{1\}} \oplus s, \alpha^{\text{right}}_{\{\perp\}})$

P_1

- Compute $\text{Eval}(\text{msk}^{\text{left}}) \rightarrow y^{\text{left}}$
- Compute $\text{KeyGen}(\text{msk}^{\text{right}}, \{\perp\}) \rightarrow \alpha^{\text{right}}_{\{\perp\}}$
- Set $\sigma_1 = (y^{\text{left}}|_{\{1\}} \oplus s, \alpha^{\text{right}}_{\{\perp\}})$

PRG^{left}

$\text{PRG}^{\text{right}}$

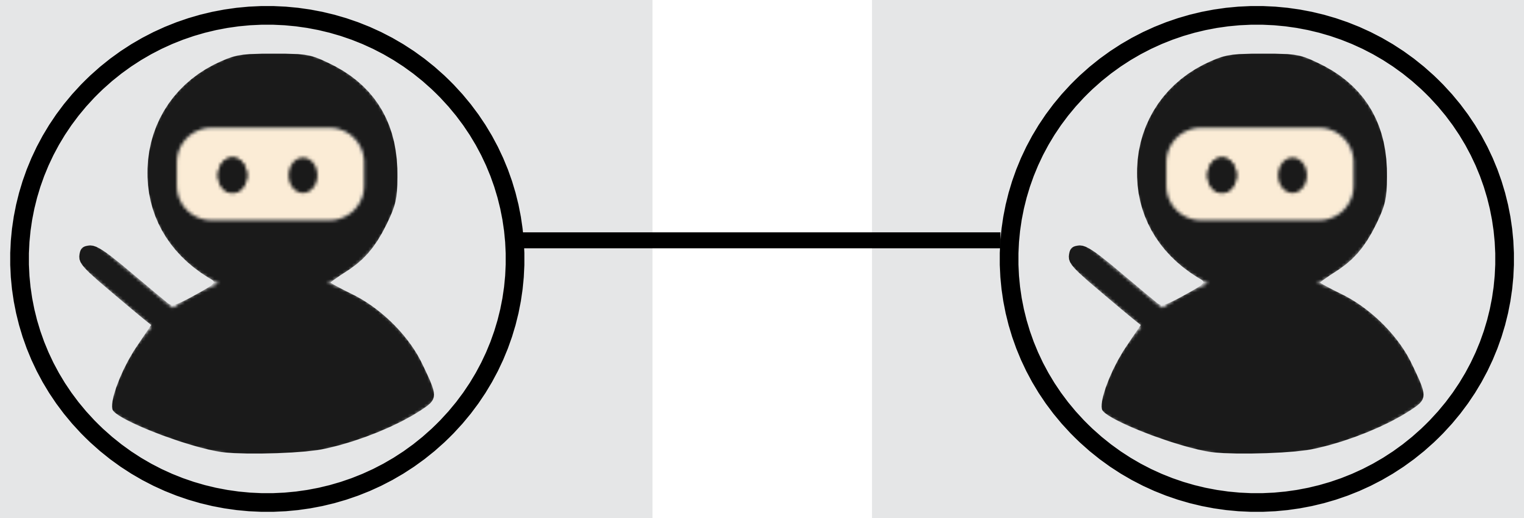
Evolving Bipartite Graphs

(Construction from Projective PRGs)

Dealer

secret s

P_1 P_2



$\sigma_2 = (y^{\text{right}}|_{\{2\}} \oplus s, \alpha^{\text{left}}_{\{1\}})$

σ_1

- Compute $\text{Eval}(\text{msk}^{\text{right}}) \rightarrow y^{\text{right}}$
- Compute $\text{KeyGen}(\text{msk}^{\text{left}}, \{1\}) \rightarrow \alpha^{\text{left}}_{\{1\}}$
- Set $\sigma_2 = (y^{\text{right}}|_{\{2\}} \oplus s, \alpha^{\text{left}}_{\{1\}})$

PRG^{left} $\text{PRG}^{\text{right}}$

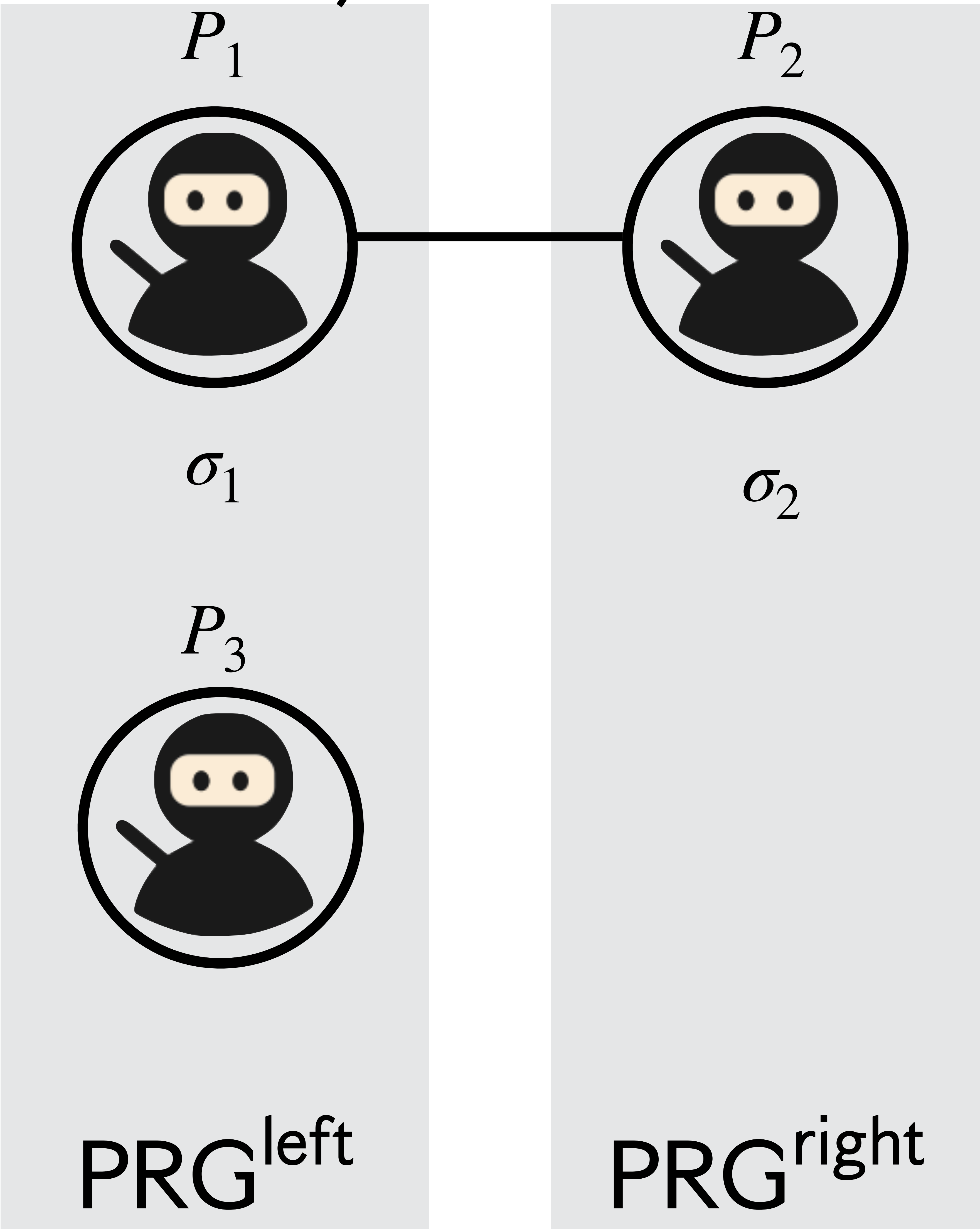
Evolving Bipartite Graphs

(Construction from Projective PRGs)

Dealer

secret s

$\sigma_3 = (y^{\text{left}}|_{\{3\}} \oplus s, \alpha^{\text{right}}_{\{\perp\}})$



- Compute $\text{Eval}(\text{msk}^{\text{left}}) \rightarrow y^{\text{left}}$
- Compute $\text{KeyGen}(\text{msk}^{\text{right}}, \{\perp\}) \rightarrow \alpha^{\text{right}}_{\{\perp\}}$
- Set $\sigma_3 = (y^{\text{left}}|_{\{3\}} \oplus s, \alpha^{\text{right}}_{\{\perp\}})$

Evolving Bipartite Graphs

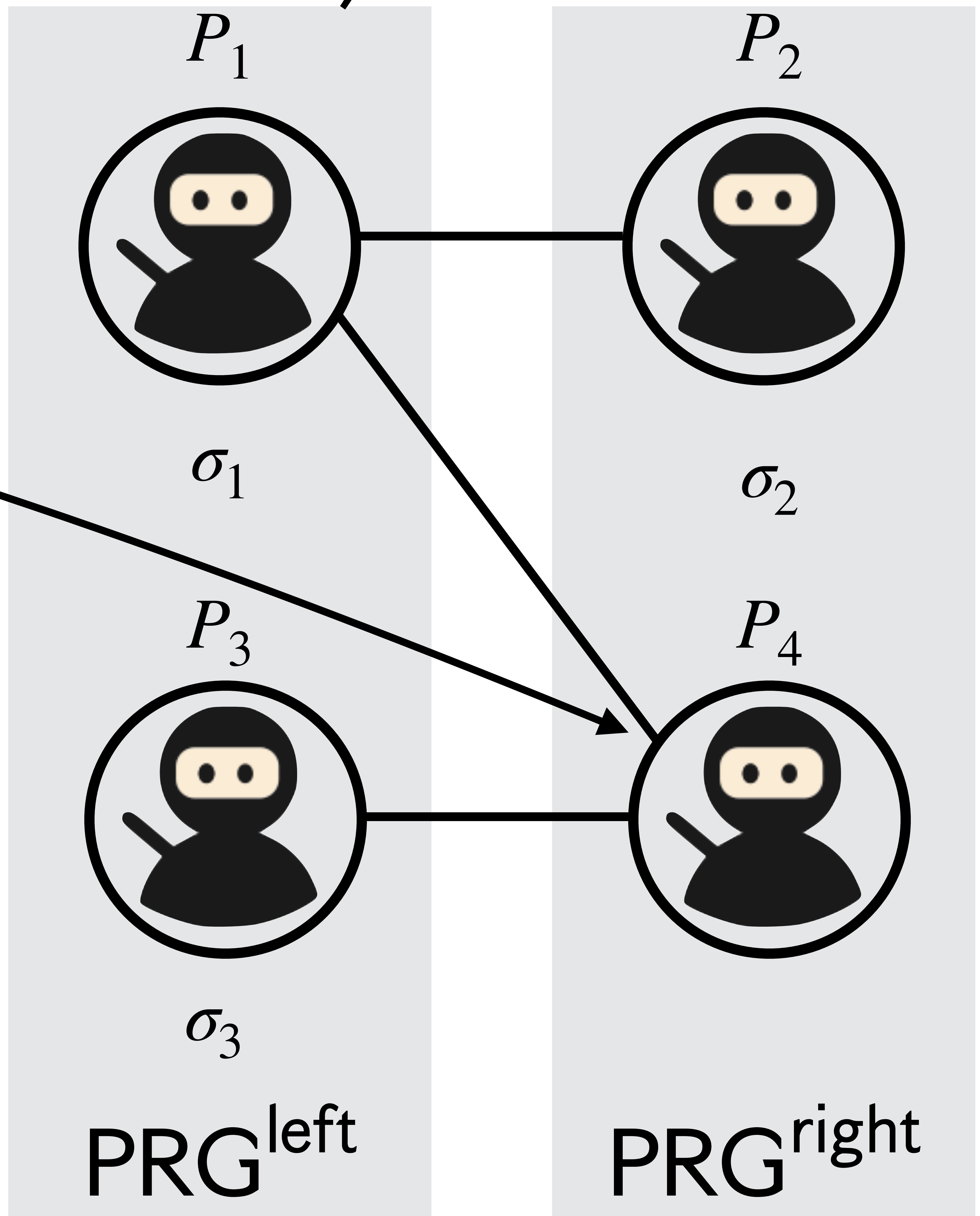
(Construction from Projective PRGs)

Dealer

secret s

$\sigma_4 = (y^{\text{right}}|_{\{4\}} \oplus s, \alpha^{\text{left}}_{\{1,3\}})$

- Compute $\text{Eval}(\text{msk}^{\text{right}}) \rightarrow y^{\text{right}}$
- Compute $\text{KeyGen}(\text{msk}^{\text{left}}, \{1,3\}) \rightarrow \alpha^{\text{left}}_{\{1,3\}}$
- Set $\sigma_4 = (y^{\text{right}}|_{\{4\}} \oplus s, \alpha^{\text{left}}_{\{1,3\}})$

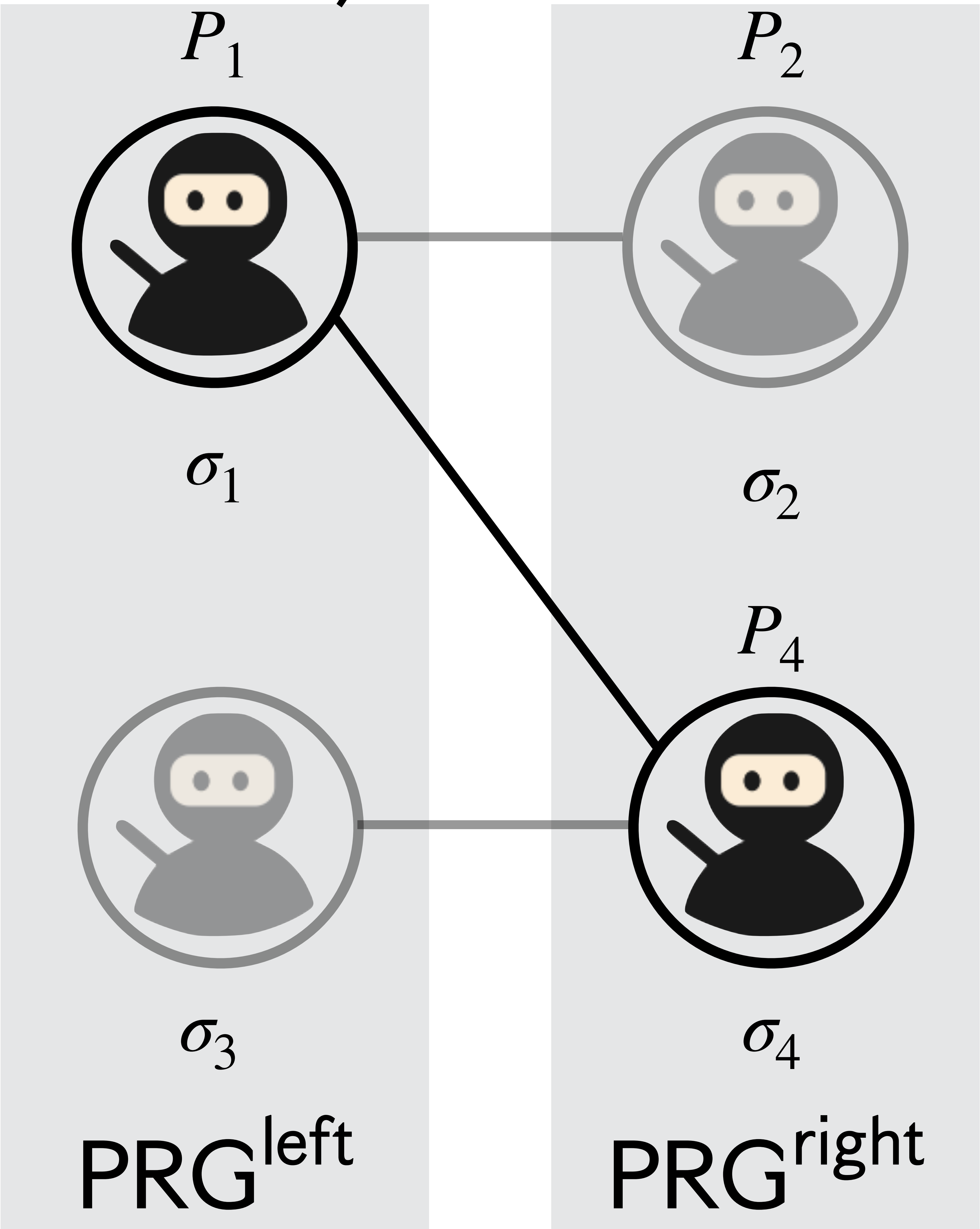


Evolving Bipartite Graphs

(Construction from Projective PRGs)

$\sigma_1 = (y^{\mathsf{left}}|_{\{1\}} \oplus s, \alpha^{\mathsf{right}}_{\{\perp\}})$

$\sigma_4 = (y^{\mathsf{right}}|_{\{4\}} \oplus s, \alpha^{\mathsf{left}}_{\{1,3\}})$



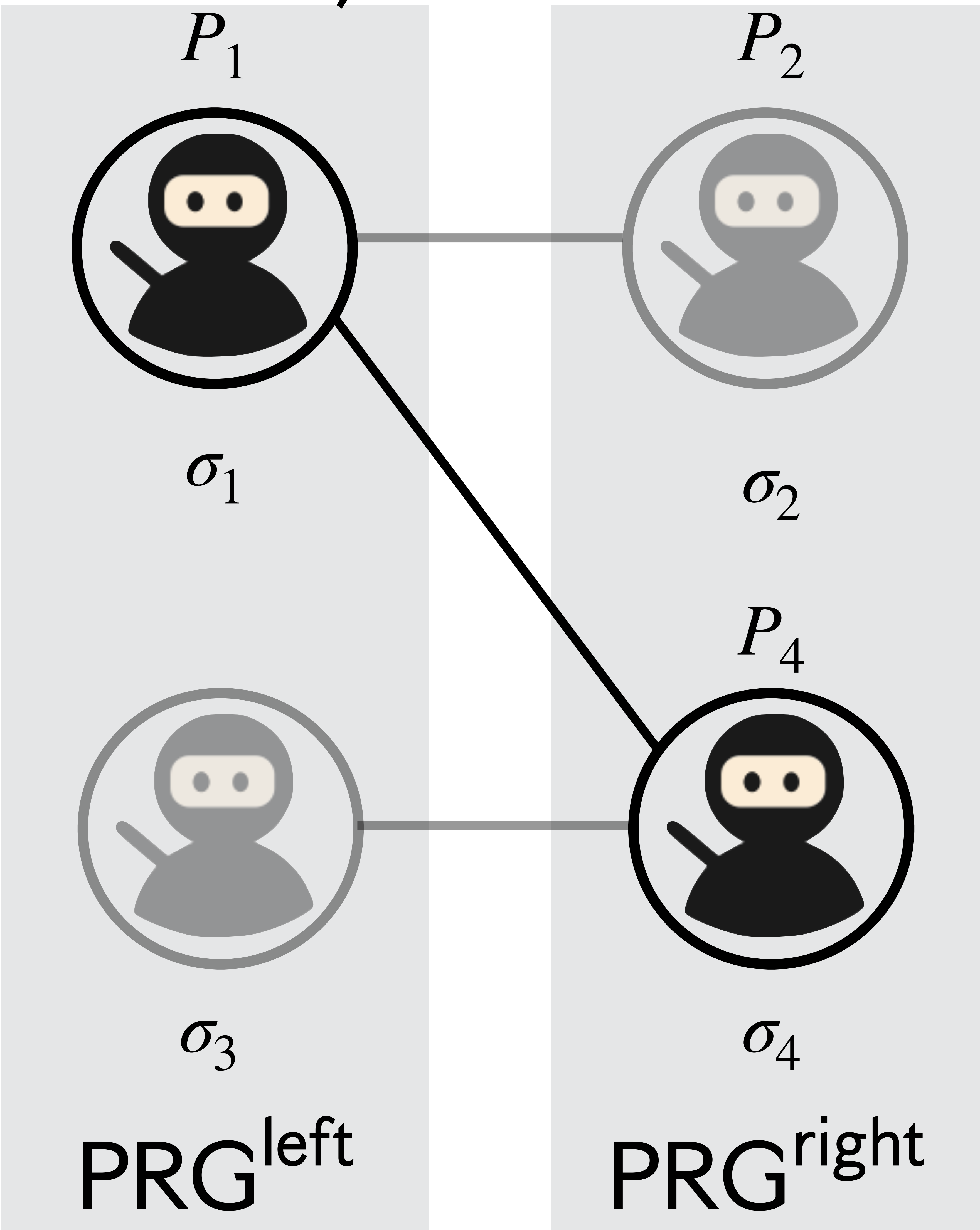
Evolving Bipartite Graphs

(Construction from Projective PRGs)

$\sigma_1 = (\boxed{y^{\text{left}}|_{\{1\}} \oplus s,} \alpha^{\text{right}}_{\{\perp\}})$ $\sigma_4 = (y^{\text{right}}|_{\{4\}} \oplus s, \alpha^{\text{left}}_{\{1,3\}})$

Take the **encryption of the secret** from the **share** of the **older** party:

$$y^{\text{left}}|_{\{1\}} \oplus s$$



Evolving Bipartite Graphs

(Construction from Projective PRGs)

$\sigma_1 = (\boxed{y^{\text{left}}|_{\{1\}} \oplus s}, \alpha^{\text{right}}_{\{\perp\}})$ $\quad\quad$ $\sigma_4 = (y^{\text{right}}|_{\{4\}} \oplus s, \boxed{\alpha^{\text{left}}_{\{1,3\}}})$

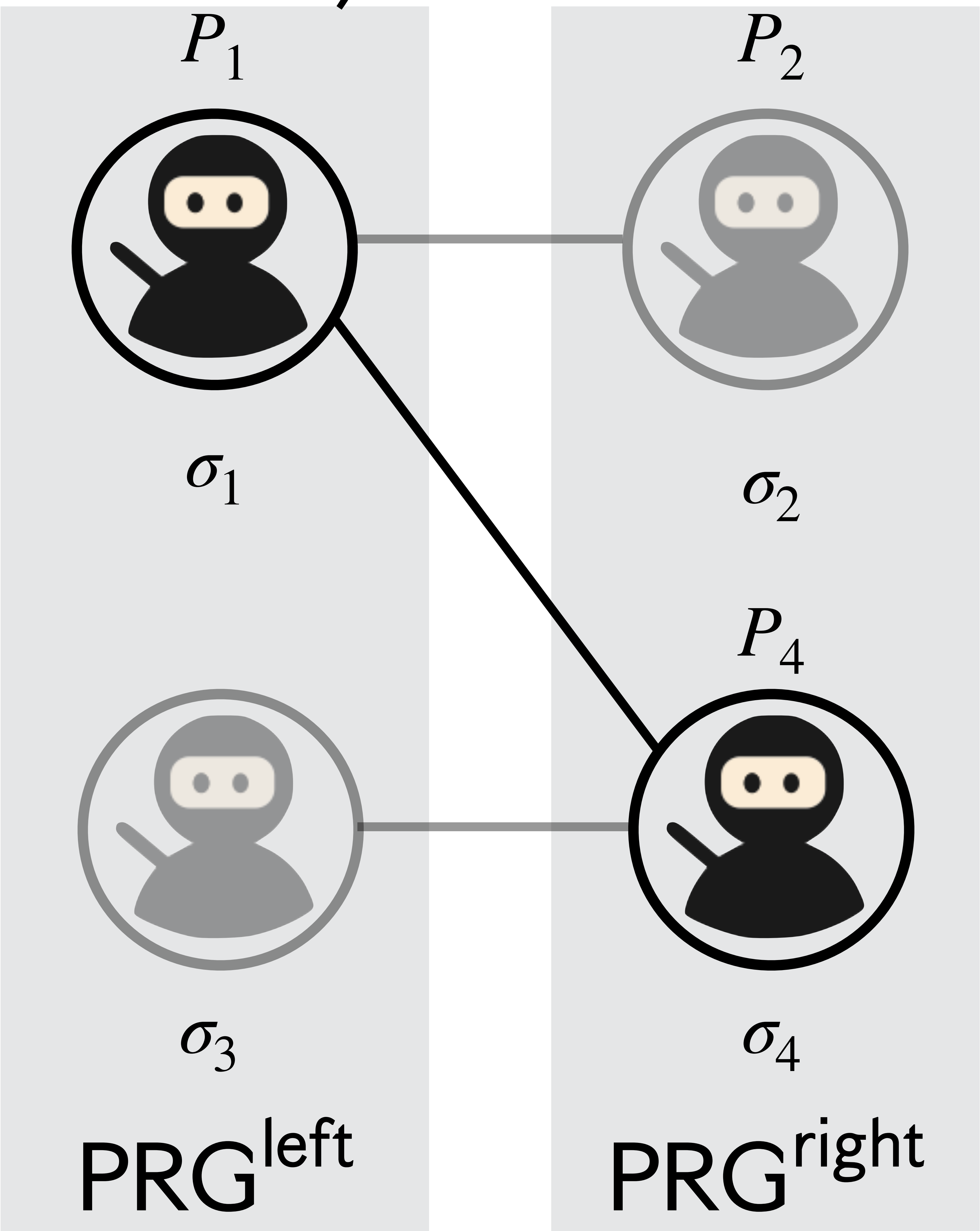
Take the **encryption of the secret** from the **share** of the **older** party:

$$y^{\text{left}}|_{\{1\}} \oplus s$$

$$\downarrow$$

Use the **projective key** of the **recent** party to re-compute the **pseudorandom value**:

$\text{Eval}(\alpha^{\text{left}}_{\{1,3\}}) \rightarrow y^{\text{left}}|_{\{1,3\}}$ and **restrict** the output to $y^{\text{left}}|_{\{1\}}$



Evolving Bipartite Graphs
(Construction from Projective PRGs)

$\sigma_1 = (\boxed{y^{\text{left}}|_{\{1\}} \oplus s}, \alpha^{\text{right}}_{\{\perp\}})$ $\quad$ $\sigma_4 = (y^{\text{right}}|_{\{4\}} \oplus s, \boxed{\alpha^{\text{left}}_{\{1,3\}}})$

Take the **encryption of the secret** from the **share** of the **older** party:

$y^{\text{left}}|_{\{1\}} \oplus s$

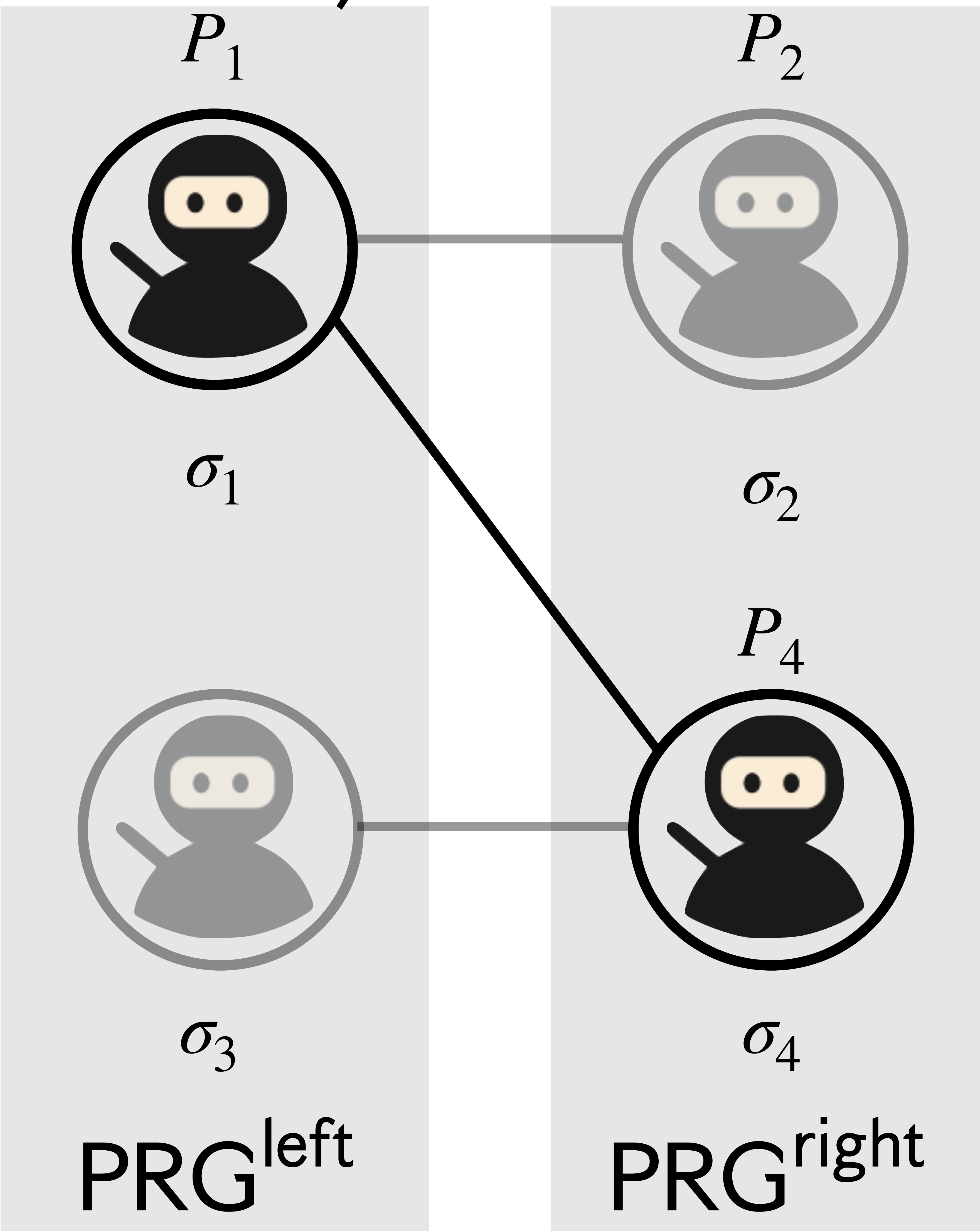
$\downarrow$

Use the **projective key** of the **recent** party to re-compute the **pseudorandom value**:

Eval$(\alpha^{\text{left}}_{\{1,3\}}) \rightarrow y^{\text{left}}|_{\{1,3\}}$ and **restrict** the output to $y^{\text{left}}|_{\{1\}}$

$\downarrow$

Get the **secret** s = $y^{\text{left}}|_{\{1\}} \oplus s \oplus y^{\text{left}}|_{\{1\}}$

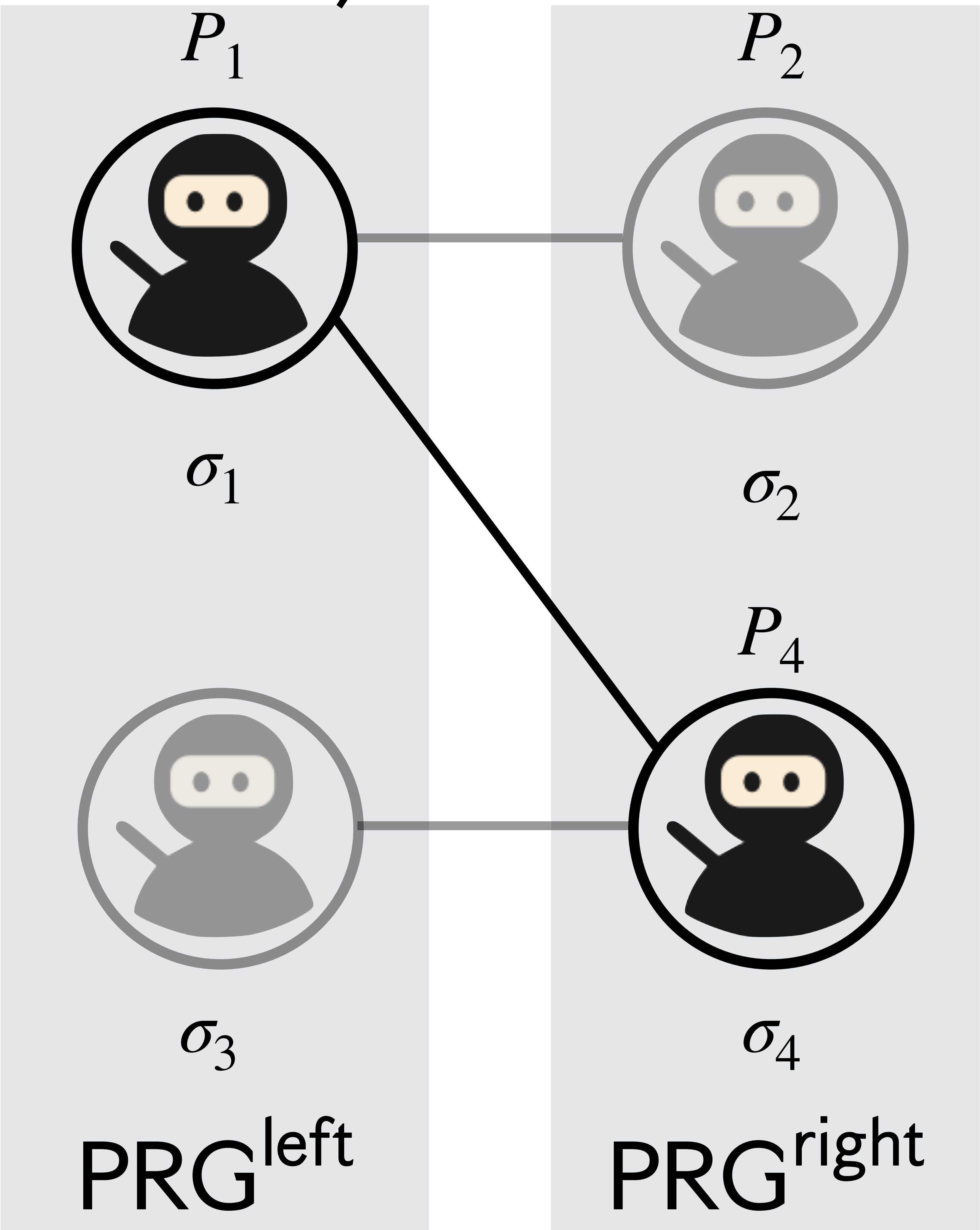


Evolving Bipartite Graphs

(Construction from Projective PRGs)

Share size

Assuming **RSA,** we have shares of size poly(λ) **(independent of the number of parties).**



Evolving Threshold
(Formalisation)

Evolving Threshold Access Structure

Let $t_1 \leq t_2 \leq \ldots \leq t_n$.

The threshold access structure A_i at time $i \in [n]$ is defined as follows:

$$A_i = A_{i-1} \cup \{\text{all sets } X \subseteq [n] \text{ of size at least } t_i\}$$

Evolving Threshold
(Formalisation)

Evolving Threshold Access Structure

Let $t_1 \leq t_2 \leq \ldots \leq t_n$.
The threshold access structure A_i at time $i \in [n]$ is defined as follows:
$A_i = A_{i-1} \cup \{$all sets $X \subseteq [n]$ of size at least $t_i\}$

EXAMPLE

Fix $t_1 = 2,\ t_2 = 2,\ t_3 = 2,\ t_4 = 4.$

$A_1 = \varnothing$
$A_2 = \{\{1,2\}\}$
$A_3 = \{\{1,2\}, \{2,3\}, \{1,3\}, \{1,2,3\}\}$
$A_4 = \{\{1,2\}, \{2,3\}, \{1,3\}, \{1,2,3\}, \{1,2,3,4\}\}$

Evolving Threshold

(Formalisation)

Evolving Threshold Access Structure

Let $t_1 \leq t_2 \leq \ldots \leq t_n$.

The threshold access structure A_i at time $i \in [n]$ is defined as follows:

$A_i = A_{i-1} \cup \{$all sets $X \subseteq [n]$ of size at least $t_i\}$

EXAMPLE

Fix $t_1 = 2,\ t_2 = 2,\ t_3 = 2,\ t_4 = 4.$

Monotonicity

Rigidity

$$A_1 = \varnothing$$
$$A_2 = \{\{1,2\}\}$$
$$A_3 = \{\{1,2\}, \{2,3\}, \{1,3\}, \{1,2,3\}\}$$
$$A_4 = \{\{1,2\}, \{2,3\}, \{1,3\}, \{1,2,3\}, \{1,2,3,4\}\}$$

Evolving Threshold

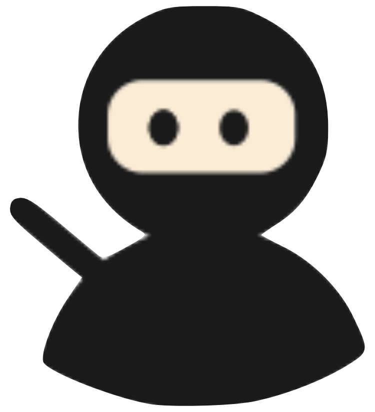
(Construction from OWF)

Public access structure

$t_1 \leq t_2 \leq \ldots$

Dealer

secret s



P_1

Public access structure

$t_1 \leq t_2 \leq \dots$

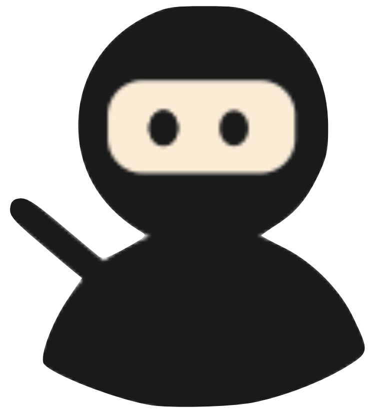
Evolving Threshold

(Construction from OWF)

Dealer

secret s

- Sample random f_1 of degree $t_1 - 1$ such that $f_1(0) = s$
- Sample random PRG seed k_1



P_1

Public access structure

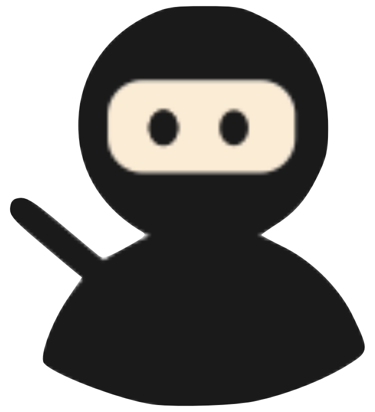
$t_1 \leq t_2 \leq \dots$

Evolving Threshold

(Construction from OWF)

Dealer

$\sigma_1 = (f_1(1), k_1)$



P_1

secret s

- Sample random f_1 of degree $t_1 - 1$ such that $f_1(0) = s$
- Sample random PRG seed k_1

Public access structure

$t_1 \leq t_2 \leq \ldots$

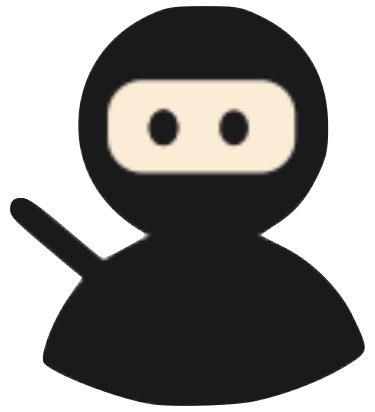
Evolving Threshold

(Construction from OWF)

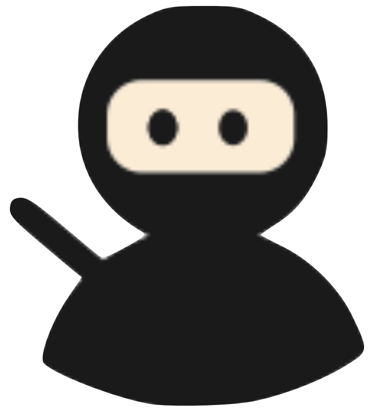
Dealer

secret s

$(f_1(1), k_1) = \sigma_1$



P_1



P_2

Public access structure

$t_1 \leq t_2 \leq \dots$

Evolving Threshold

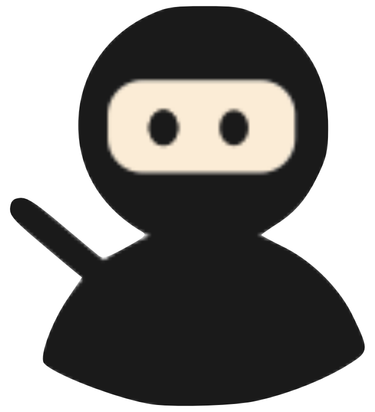
(Construction from OWF)

Dealer

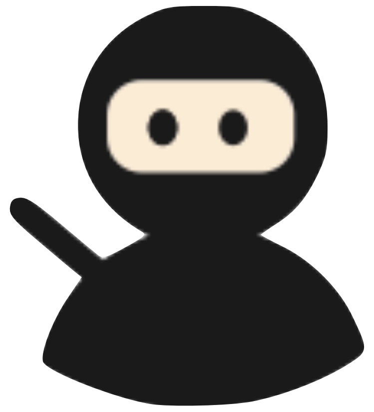
secret s

- Sample random f_2 of degree $t_2 - 1$ such that $f_2(0) = s$
- Sample random PRG seed k_2
- Let γ_1^2 the next unused block of $\mathsf{PRG}(k_1)$

$(f_1(1), k_1) = \sigma_1$



P_1



P_2

Public access structure

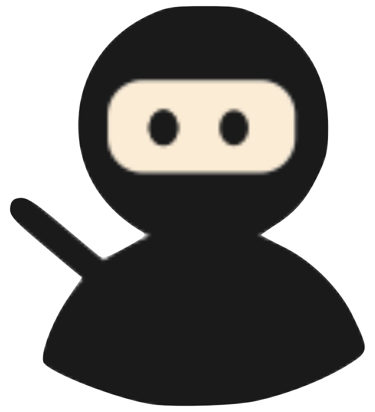
$t_1 \leq t_2 \leq \dots$

Evolving Threshold

(Construction from OWF)

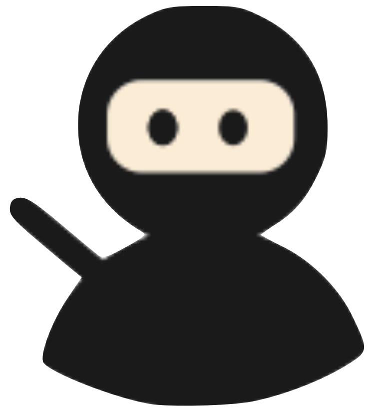
Dealer

$(f_1(1), k_1) = \sigma_1$



P_1

$\sigma_2 = (f_2(2), k_2, f_2(1) \oplus \gamma_1^2)$



P_2

secret s

- Sample random f_2 of degree $t_2 - 1$ such that $f_2(0) = s$
- Sample random PRG seed k_2
- Let γ_1^2 the next unused block of $\mathsf{PRG}(k_1)$

Public access structure
$t_1 \le t_2 \le \dots$

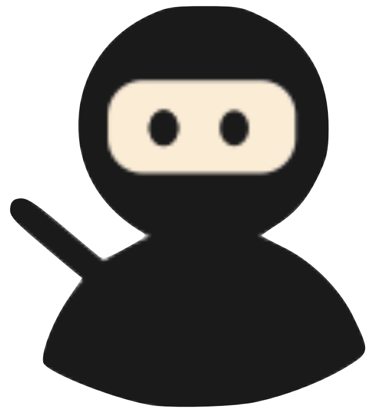
Evolving Threshold
(Construction from OWF)

Dealer

secret s

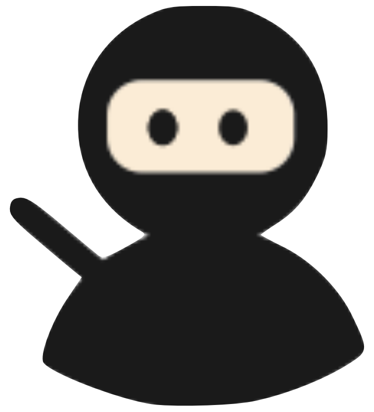
- Sample random f_3 of degree $t_3 - 1$ such that $f_3(0) = s$
- Sample random PRG seed k_3
- For $i \in [2]$, let γ_i^3 the next unused block of $\mathsf{PRG}(k_i)$

$(f_1(1), k_1) = \sigma_1$



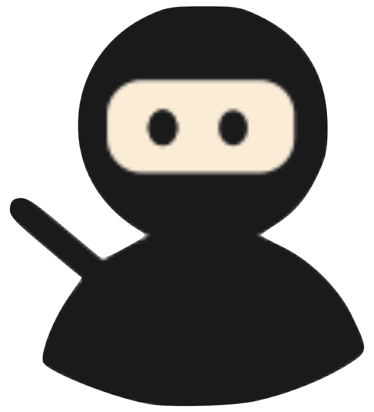
P_1

$(f_2(2), k_2, f_2(1) \oplus \gamma_1^1) = \sigma_2$



P_2

$\sigma_3 = (f_3(3), k_3, f_3(1) \oplus \gamma_1^3, f_3(2) \oplus \gamma_2^3)$



P_3

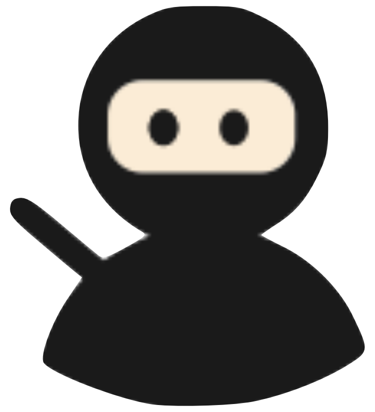
Public access structure

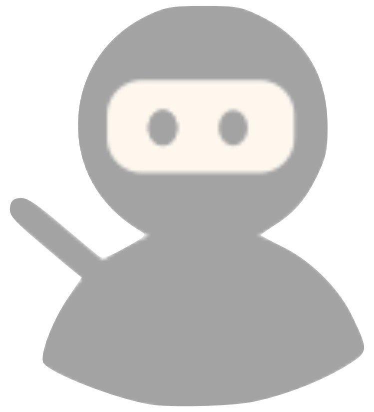
$t_1 \leq t_2 \leq \dots$

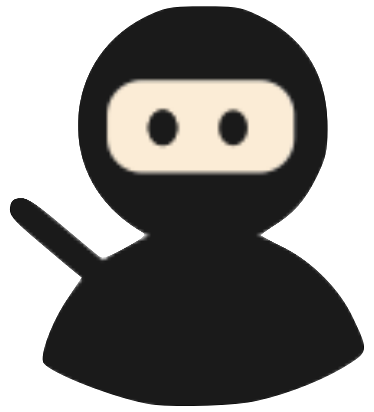
Evolving Threshold

(Construction from OWF)

Assume $t_3 = 2$

$(f_1(1), k_1) = \sigma_1$  P_1

$(f_2(2), k_2, f_2(1) \oplus \gamma_1^1) = \sigma_2$ 

$(f_3(3), k_3, f_3(1) \oplus \gamma_1^3, f_3(2) \oplus \gamma_2^3) = \sigma_3$  P_3

Public access structure

$t_1 \leq t_2 \leq \dots$

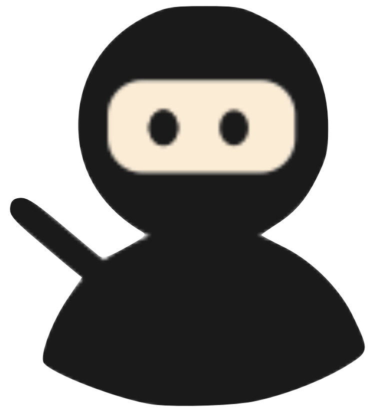
Evolving Threshold

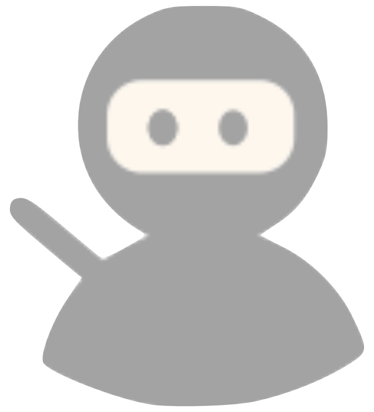
(Construction from OWF)

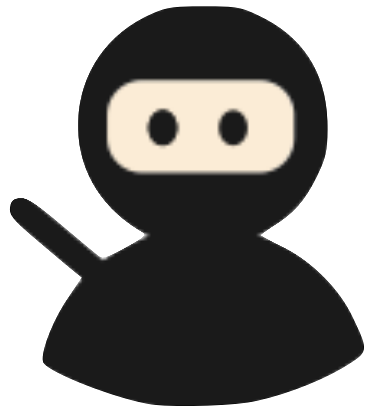
Assume $t_3 = 2$

$\downarrow$

f_3 **is of degree** $t_3 - 1 = 1$

$(f_1(1), k_1) = \sigma_1$  P_1

$(f_2(2), k_2, f_2(1) \oplus \gamma_1^1) = \sigma_2$ 

$(f_3(3), k_3, f_3(1) \oplus \gamma_1^3, f_3(2) \oplus \gamma_2^3) = \sigma_3$  P_3

Public access structure

$t_1 \leq t_2 \leq \dots$

Evolving Threshold

(Construction from OWF)

Assume $t_3 = 2$

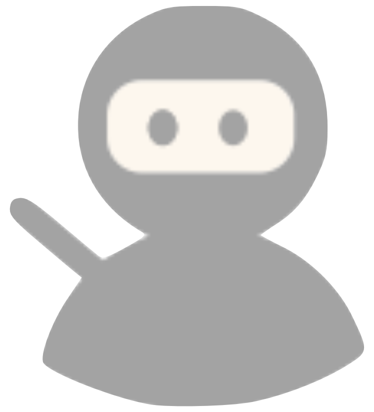
$\downarrow$

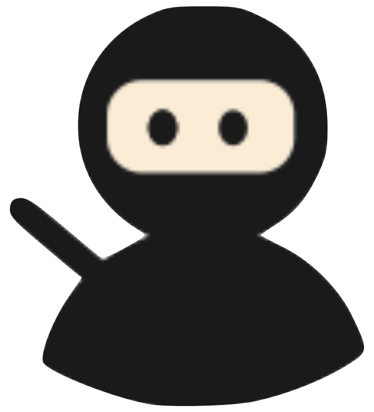
f_3 **is of degree** $t_3 - 1 = 1$

$\downarrow$

Get $f_3(3)$ **and** $f_3(1)$ **using the PRG seed** k_1

$(f_1(1), \boxed{k_1}) = \sigma_1$  P_1

$(f_2(2), k_2, f_2(1) \oplus \gamma_1^1) = \sigma_2$ 

$(\boxed{f_3(3)}, k_3, \boxed{f_3(1) \oplus \gamma_1^3}, f_3(2) \oplus \gamma_2^3) = \sigma_3$  P_3

Public access structure

$t_1 \leq t_2 \leq \dots$

Evolving Threshold

(Construction from OWF)

Assume $t_3 = 2$

$\downarrow$

f_3 **is of degree** $t_3 - 1 = 1$

$\downarrow$

Get $f_3(3)$ **and** $f_3(1)$ **using the PRG seed** k_1

$\downarrow$

Get $s = f_3(0)$ **using Lagrange interpolation.**

$(f_1(1), \boxed{k_1}) = \sigma_1$ P_1

$(f_2(2), k_2, f_2(1) \oplus \gamma_1^1) = \sigma_2$

$(\boxed{f_3(3)}, k_3, \boxed{f_3(1) \oplus \gamma_1^3}, f_3(2) \oplus \gamma_2^3) = \sigma_3$ P_3

Public access structure

$t_1 \leq t_2 \leq \ldots$

Evolving Threshold

(Construction from OWF)

Share size

Our scheme: $|\sigma_n| = \lambda \cdot (n+1)$

V.S.

IT setting: [A] $|\sigma_n| \in \lambda \cdot O(n^4 \cdot \log(n))$ **—— [B]** $|\sigma_n| \in \lambda \cdot O(n^4)$

[A] Komargodski, Ilan, and Anat Paskin-Cherniavsky.
"Evolving secret sharing: dynamic thresholds and robustness."
TCC 2017.

[B] Xing, Chaoping, and Chen Yuan.
"Evolving secret sharing schemes based on polynomial evaluations and algebraic geometry codes."
IEEE Transactions on Information Theory (2024).

P_1

f_3 **is**

Get f

G

Lagrange interpolation.

$(\boxed{f_3(3)}\; k_3; \boxed{f_3(1) \oplus \gamma_1}; f_3(2) \oplus \gamma_2) = \sigma_3$

P_3

Other Results

(See full version eprint.iacr.org/2023/1534)

Other Results

(See full version eprint.iacr.org/2023/1534)

Other Evolving Access Structures

Arbitrary Access Structures **(with polynomially many authorized sets)**

*The computational setting permits to circumvent Mazor's IT lower bound **[A]**.*

Monotone Circuits — CNF — DNF

[A] Mazor, Noam.
"A lower bound on the share size in evolving secret sharing."
ITC 2023.

Other Results

(See full version eprint.iacr.org/2023/1534)

Other Evolving Access Structures

Arbitrary Access Structures **(with polynomially many authorized sets)**

The computational setting permits to circumvent Mazor's IT lower bound ***[A]****.*

Monotone Circuits — CNF — DNF

[A] Mazor, Noam.
"A lower bound on the share size in evolving secret sharing."
ITC 2023.

Evolving information dispersal

We extend the notion of **Information Dispersal** to the **evolving setting.**

We generalise **Krawczyk's compiler [B]** to the **evolving setting** (for some access structures).

[B] Krawczyk, Hugo.
"Secret sharing made short."
CRYPTO 93.

Thank You!