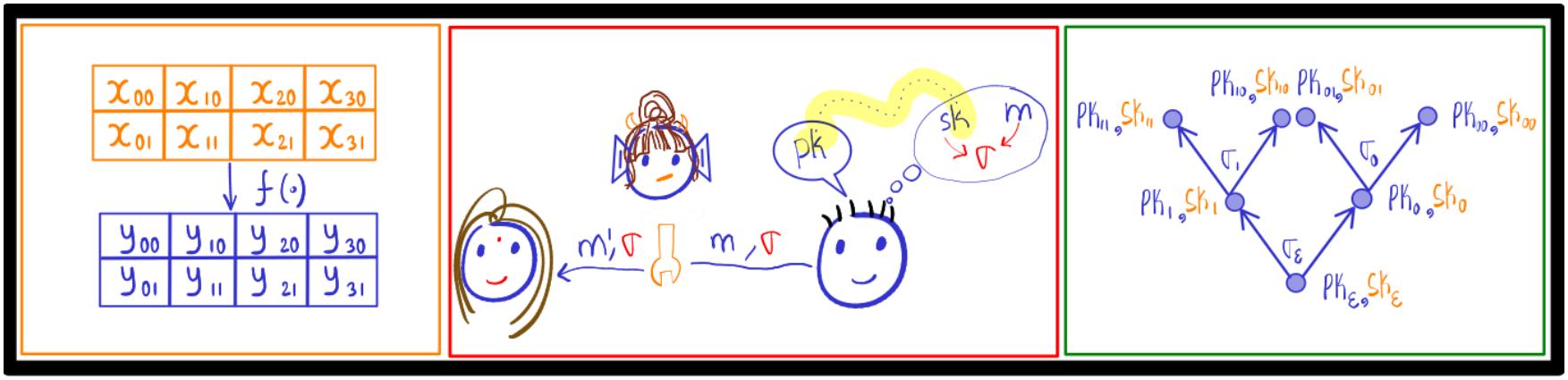


CS783: Theoretical Foundations of Cryptography

Lecture 11 (06/Sep/24)

Instructor: Chethan Kamath

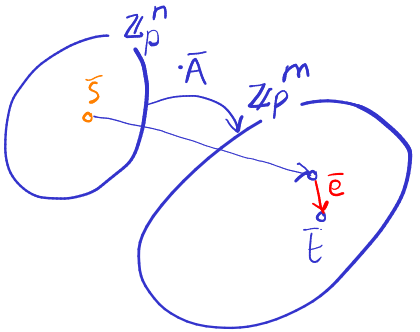
Recall from Last Lecture

- Discussed post-quantum cryptography
  -  Saw why assumptions like DLog and Factoring do not hold

Recall from Last Lecture

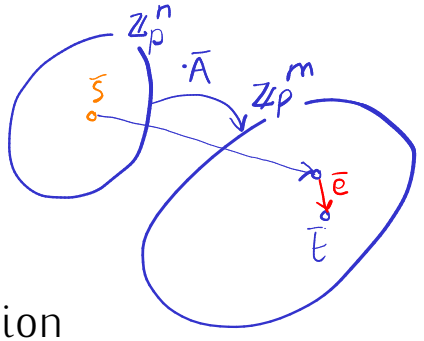
- Discussed post-quantum cryptography
  - ⚠ Saw why assumptions like DLog and Factoring do not hold

- New computational hardness assumption: LWE
  - Its decision and search variants are equivalent

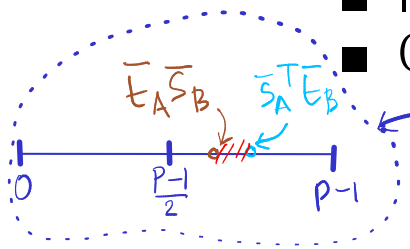


Recall from Last Lecture

- Discussed post-quantum cryptography
  - ⚠ Saw why assumptions like DLog and Factoring do not hold

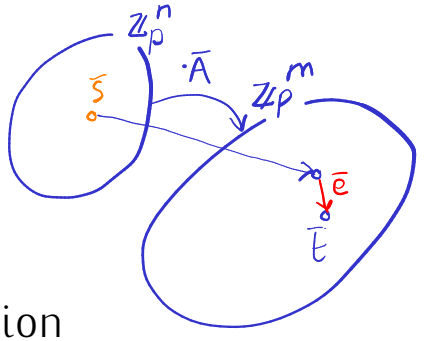


- New computational hardness assumption: LWE
  - Its decision and search variants are equivalent
  - Constructed key-exchange protocol from DLWE
    - "Noisy/approximate" Diffie-Hellman-like construction

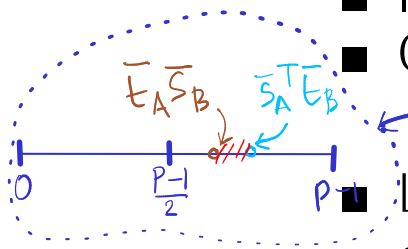


Recall from Last Lecture

- Discussed post-quantum cryptography
  - ⚠ Saw why assumptions like DLog and Factoring do not hold



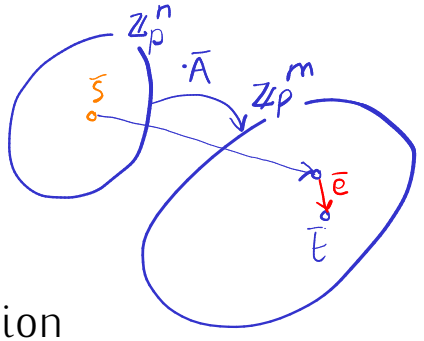
- New computational hardness assumption: LWE
  - Its decision and search variants are equivalent
  - Constructed key-exchange protocol from DLWE
    - "Noisy/approximate" Diffie-Hellman-like construction



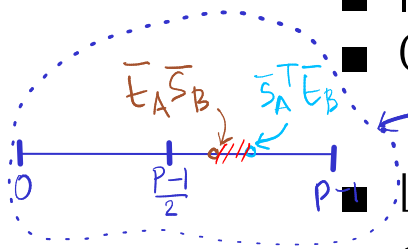
  - LWE has sufficient "structure" to support more advanced cryptographic primitives:
    1. Fully-homomorphic encryption (FHE): coming up, Lecture 19(?)
    2. Identity-based encryption
    3. Incrementally-verifiable computation...

Recall from Last Lecture

- Discussed post-quantum cryptography
  - ⚠ Saw why assumptions like DLog and Factoring do not hold



- New computational hardness assumption: LWE
  - Its decision and search variants are equivalent
  - Constructed key-exchange protocol from DLWE
    - "Noisy/approximate" Diffie-Hellman-like construction



  - LWE has sufficient "structure" to support more advanced cryptographic primitives:
    1. Fully-homomorphic encryption (FHE): coming up, Lecture 19(?)
    2. Identity-based encryption
    3. Incrementally-verifiable computation...

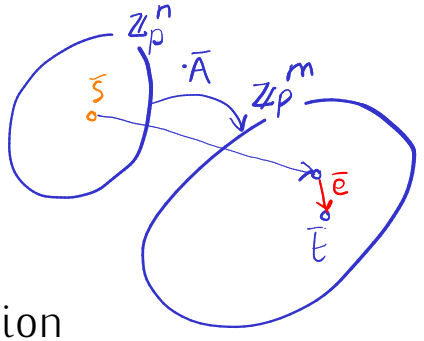
⚠ Paper 2024/555

Quantum Algorithms for Lattice Problems

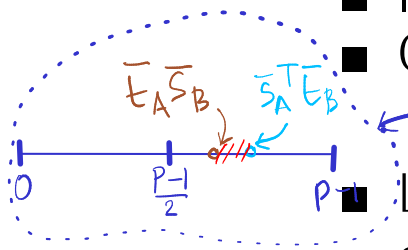
Yilei Chen, Tsinghua University, Shanghai Artificial Intelligence Laboratory, Shanghai Qi Zhi Institute

Recall from Last Lecture

- Discussed post-quantum cryptography
  - ⚠ Saw why assumptions like DLog and Factoring do not hold



- New computational hardness assumption: LWE
  - Its decision and search variants are equivalent
  - Constructed key-exchange protocol from DLWE
    - "Noisy/approximate" Diffie-Hellman-like construction
  - LWE has sufficient "structure" to support more advanced cryptographic primitives:
    1. Fully-homomorphic encryption (FHE): coming up, Lecture 19(?)
    2. Identity-based encryption
    3. Incrementally-verifiable computation...



⚠ Paper 2024/555 BUG!

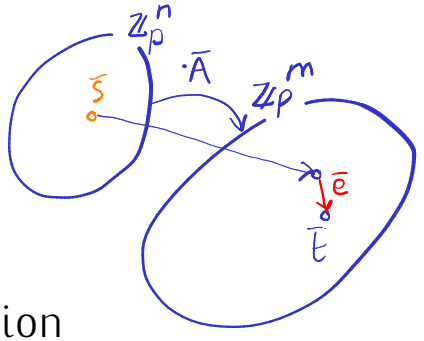
Quantum Algorithms for Lattice Problems

Yilei Chen, Tsinghua University, Shanghai Artificial Intelligence Laboratory, Shanghai Qi Zhi Institute

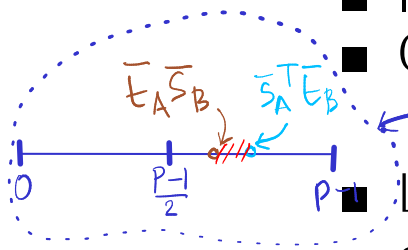
Recall from Last Lecture

Discussed post-quantum cryptography

⚠ Saw why assumptions like DLog and Factoring do not hold



- New computational hardness assumption: LWE
  - Its decision and search variants are equivalent
  - Constructed key-exchange protocol from DLWE
    - "Noisy/approximate" Diffie-Hellman-like construction

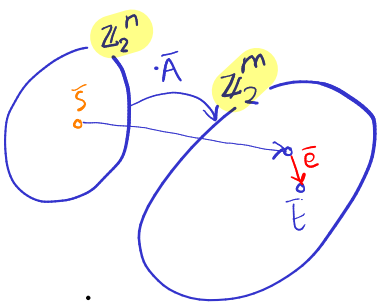


  - LWE has sufficient "structure" to support more advanced cryptographic primitives:
    1. Fully-homomorphic encryption (FHE): coming up, Lecture 19(?)
    2. Identity-based encryption
    3. Incrementally-verifiable computation...

⚠ Paper 2024/555 Bug!

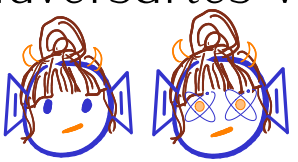
Quantum Algorithms for Lattice Problems

Yilei Chen, Tsinghua University, Shanghai Artificial Intelligence Laboratory, Shanghai Qi Zhi Institute



- Related computational problem: learning *parity* with noise

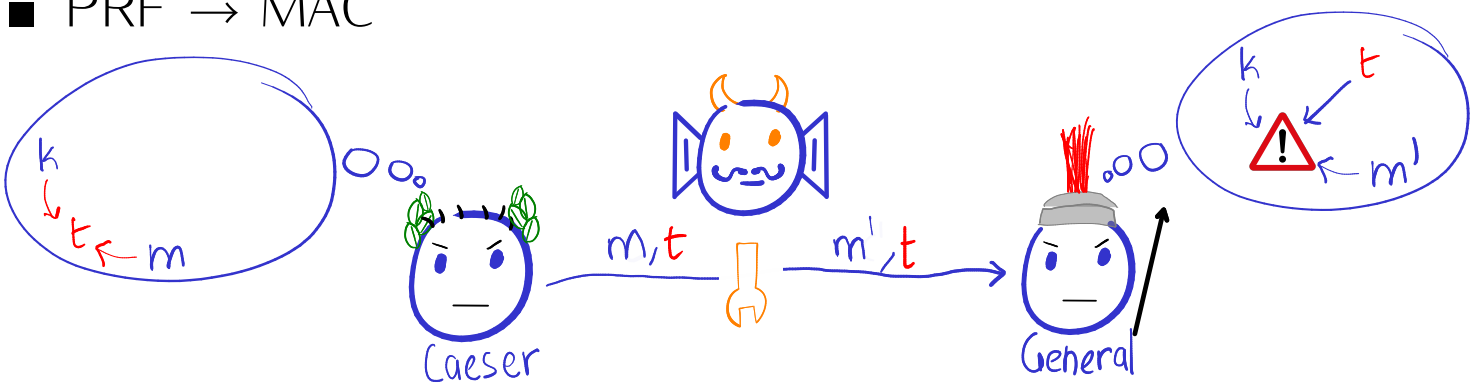
Plan for Today's Lecture...

- So far in the public-key setting: adversaries who are passive
  - Eavesdroppers of various forms 

Plan for Today's Lecture...

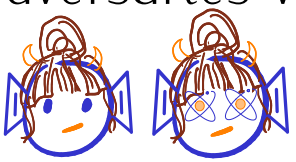
- So far in the public-key setting: adversaries who are passive
  - Eavesdroppers of various forms

- Lecture 7: integrity and authentication in secret-key setting
  - Message authentication code (MAC)
  - PRF → MAC

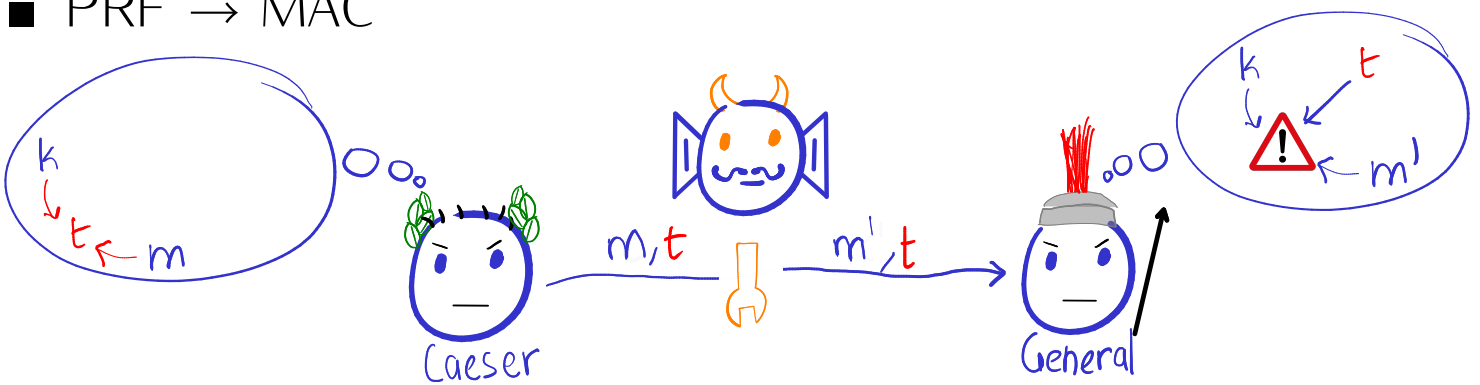


Plan for Today's Lecture...

- So far in the public-key setting: adversaries who are passive
  - Eavesdroppers of various forms



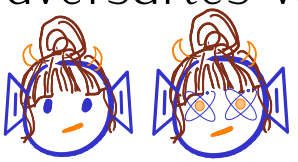
- Lecture 7: integrity and authentication in secret-key setting
  - Message authentication code (MAC)
  - PRF → MAC



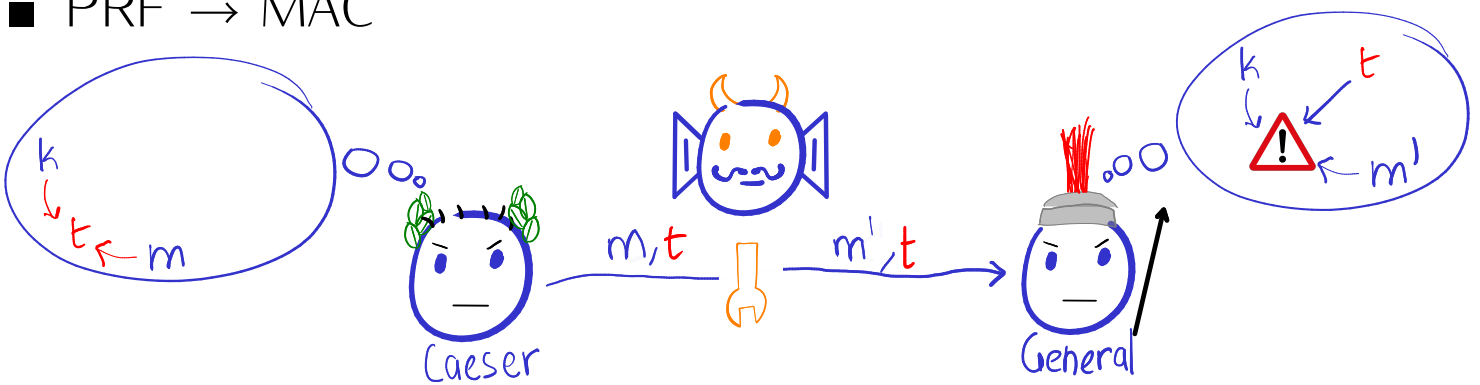
- Task 5: integrity and authentication in the *public-key* setting
  - Digital signatures (DS): public-key analogue of MAC

Plan for Today's Lecture...

- So far in the public-key setting: adversaries who are passive
  - Eavesdroppers of various forms



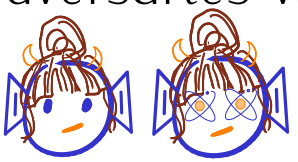
- Lecture 7: integrity and authentication in secret-key setting
  - Message authentication code (MAC)
  - PRF → MAC



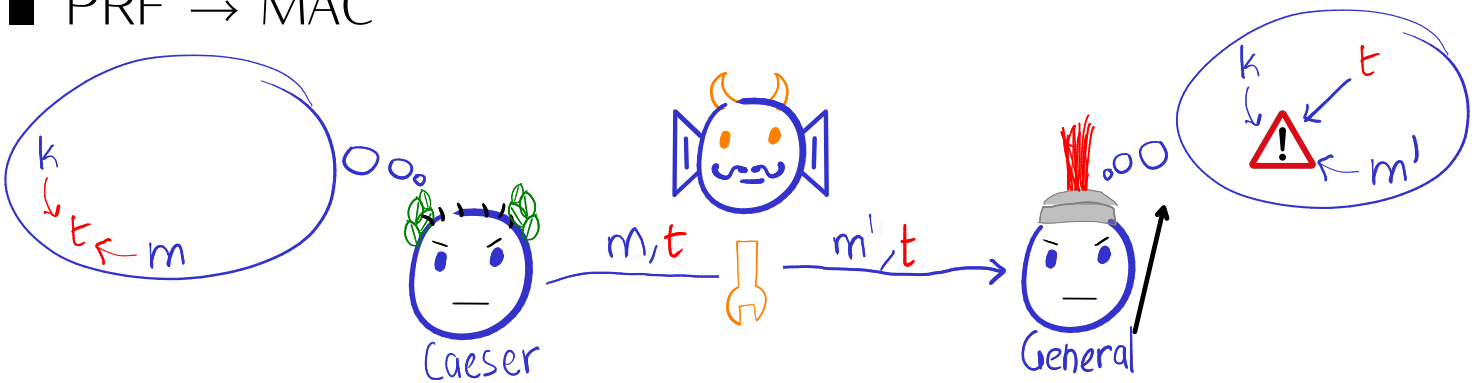
- Task 5: integrity and authentication in the *public-key* setting
  - Digital signatures (DS): public-key analogue of MAC

Plan for Today's Lecture...

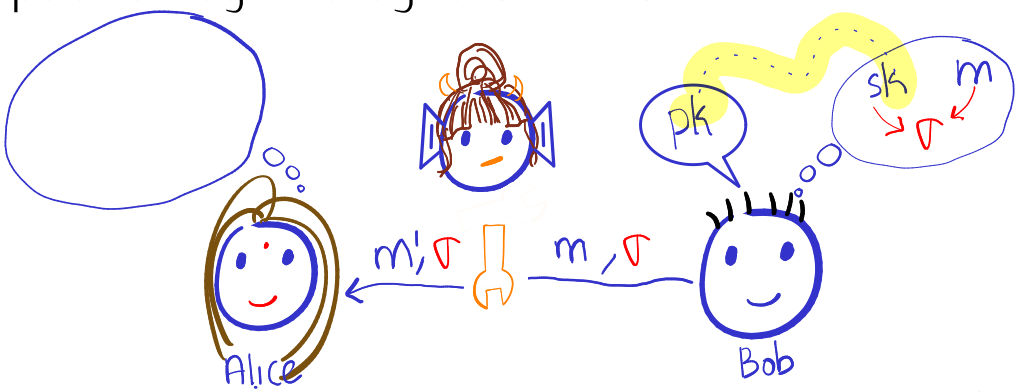
- So far in the public-key setting: adversaries who are passive
  - Eavesdroppers of various forms



- Lecture 7: integrity and authentication in secret-key setting
  - Message authentication code (MAC)
  - PRF → MAC

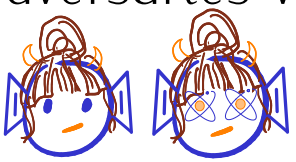


- Task 5: integrity and authentication in the *public-key* setting
  - Digital signatures (DS): public-key analogue of MAC

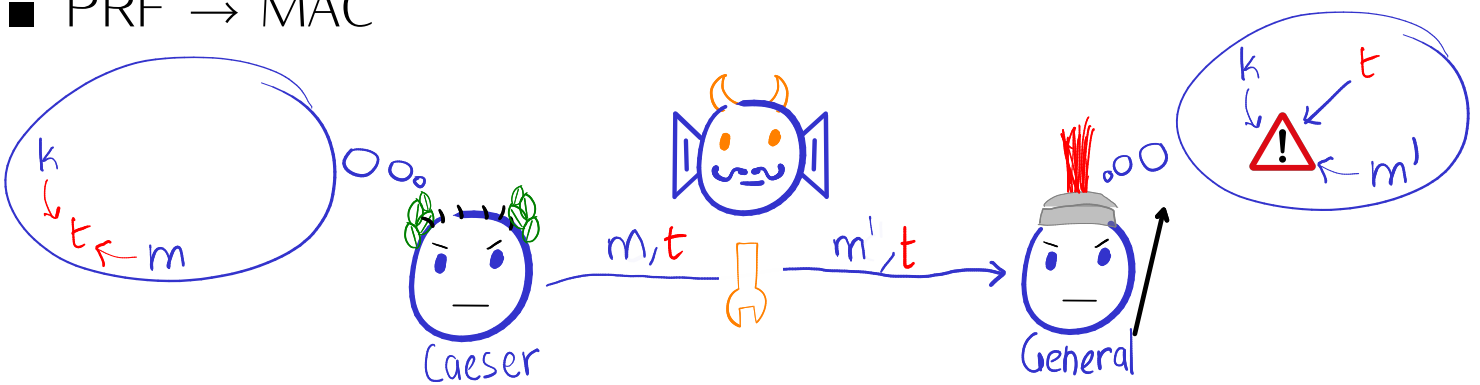


Plan for Today's Lecture...

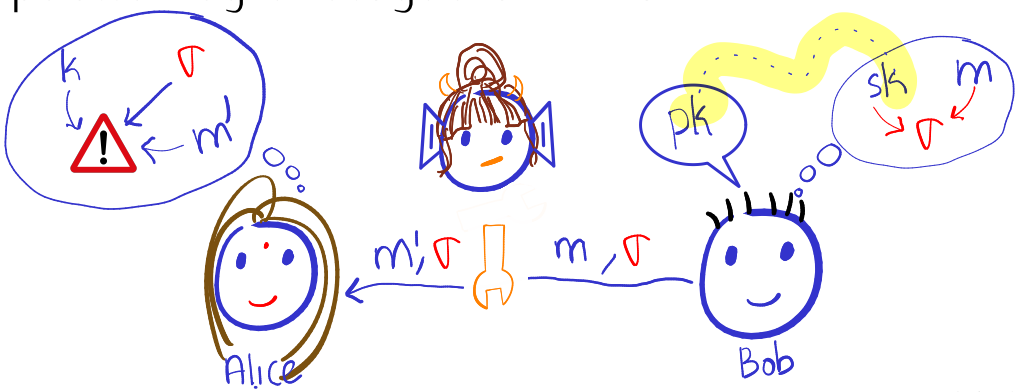
- So far in the public-key setting: adversaries who are passive
  - Eavesdroppers of various forms



- Lecture 7: integrity and authentication in secret-key setting
  - Message authentication code (MAC)
  - PRF $\rightarrow$ MAC

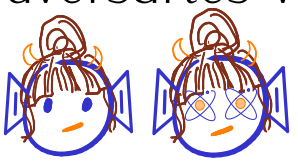


- Task 5: integrity and authentication in the *public-key* setting
  - Digital signatures (DS): public-key analogue of MAC

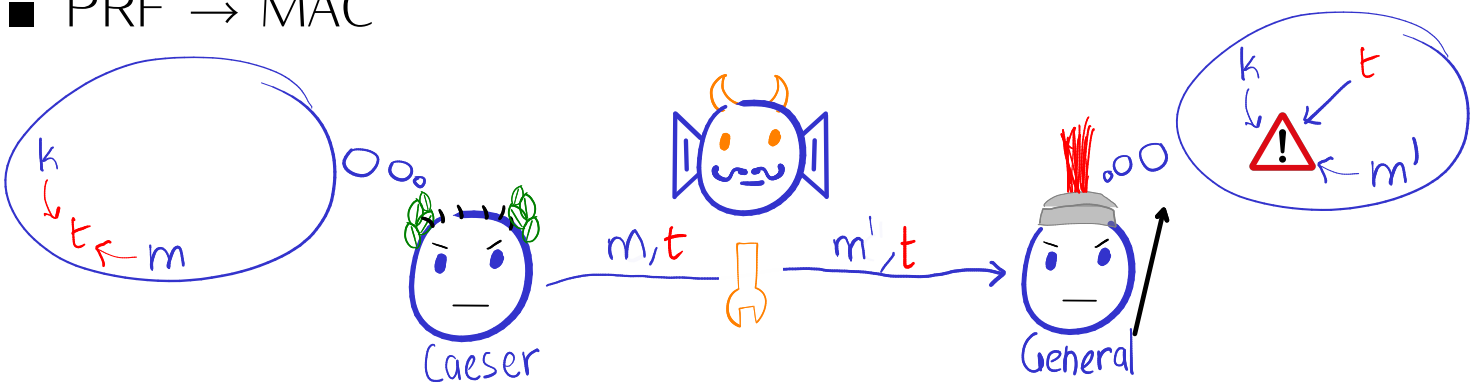


Plan for Today's Lecture...

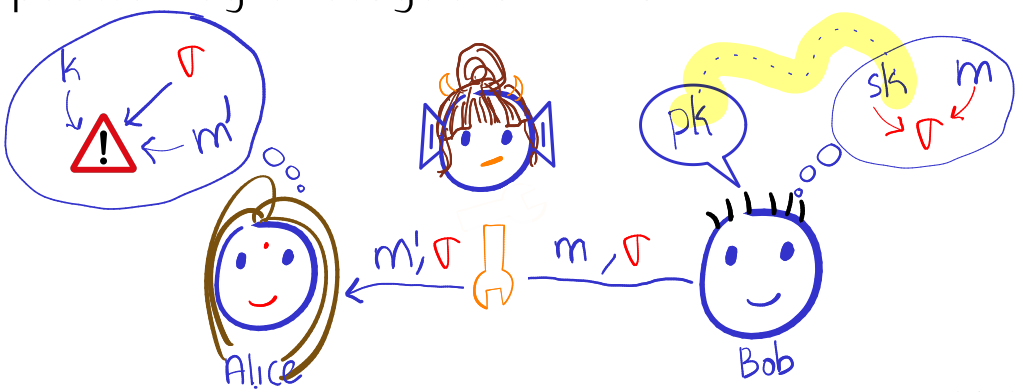
- So far in the public-key setting: adversaries who are passive
  - Eavesdroppers of various forms



- Lecture 7: integrity and authentication in secret-key setting
  - Message authentication code (MAC)
  - PRF $\rightarrow$ MAC



- Task 5: integrity and authentication in the *public-key* setting
  - Digital signatures (DS): public-key analogue of MAC
  - OWF $\rightarrow$ one-time DS
  - One-time DS $\xrightarrow{*}$ DS



Plan for Today's Lecture...

General *template*:

1. Identify the task
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities?
   - Security Goal: What does it mean to be secure?
3. Construct a scheme Π
4. Formally prove that Π in secure in model M

Plan for Today's Lecture...

General *template*:

Integrity/authentication in the public-key setting

1. Identify the task
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities?
   - Security Goal: What does it mean to be secure?
3. Construct a scheme Π
4. Formally prove that Π in secure in model M

Plan for Today's Lecture...

General *template*:

1. Identify the task
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities?
   - Security Goal: What does it mean to be secure?
3. Construct a scheme Π
4. Formally prove that Π in secure in model M

Integrity/authentication in the public-key setting

chosen-message attacker

Existentially unforgeable

Plan for Today's Lecture...

General *template*:

1. Identify the task — Integrity/authentication in the public-key setting
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities? — chosen-message attacker
   - Security Goal: What does it mean to be secure? — Existentially unforgeable
3. Construct a scheme Π — Tree-based signature
4. Formally prove that Π in secure in model M

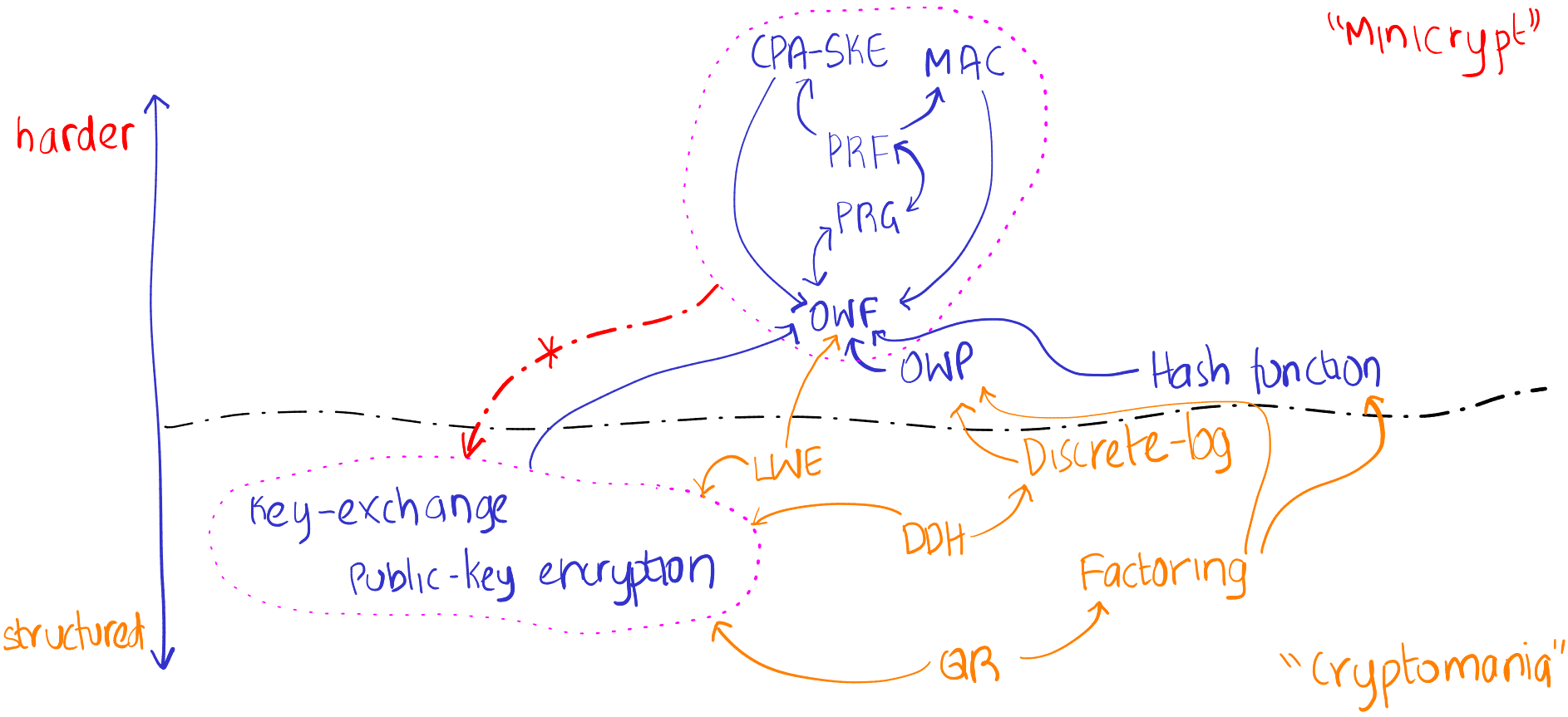
Plan for Today's Lecture...

General *template*:

1. Identify the task — Integrity/authentication in the public-key setting
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities? — chosen-message attacker
   - Security Goal: What does it mean to be secure? — Existentially unforgeable
3. Construct a scheme Π — Tree-based signature
4. Formally prove that Π in secure in model M — *Assuming OWF

Plan for Today's Lecture...

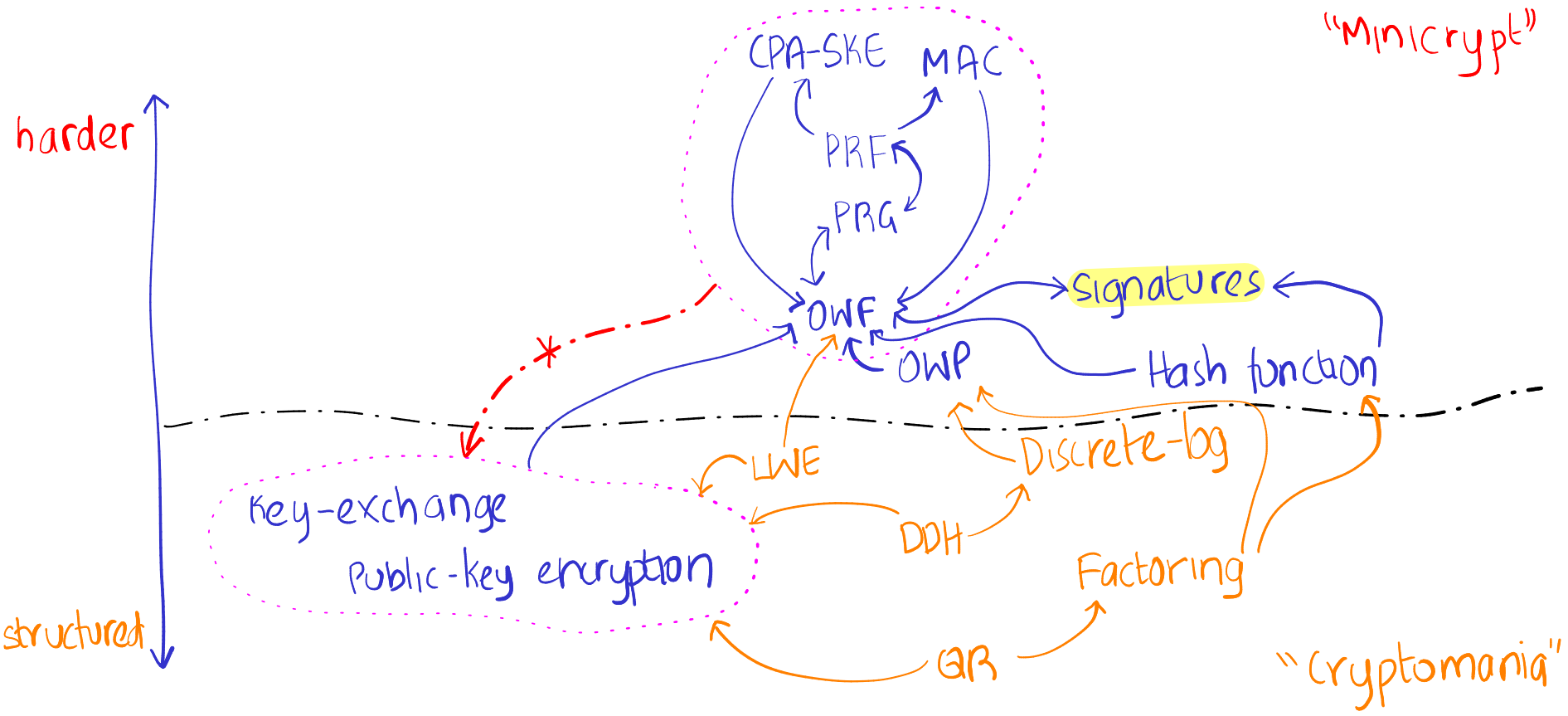
- Minicrypt to Cryptomania



- Task 5: integrity and authentication in the public-key setting

Plan for Today's Lecture...

- Minicrypt to Cryptomania



- Task 5: integrity and authentication in the public-key setting

Plan for Today's Lecture...

1. Digital Signature (DS)

2. One-Time Digital Signatures ← OWF

3. Many-Time (Stateful) Digital Signatures

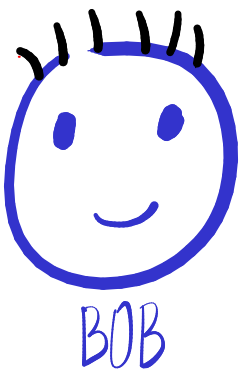
Plan for Today's Lecture

1. Digital Signature (DS)

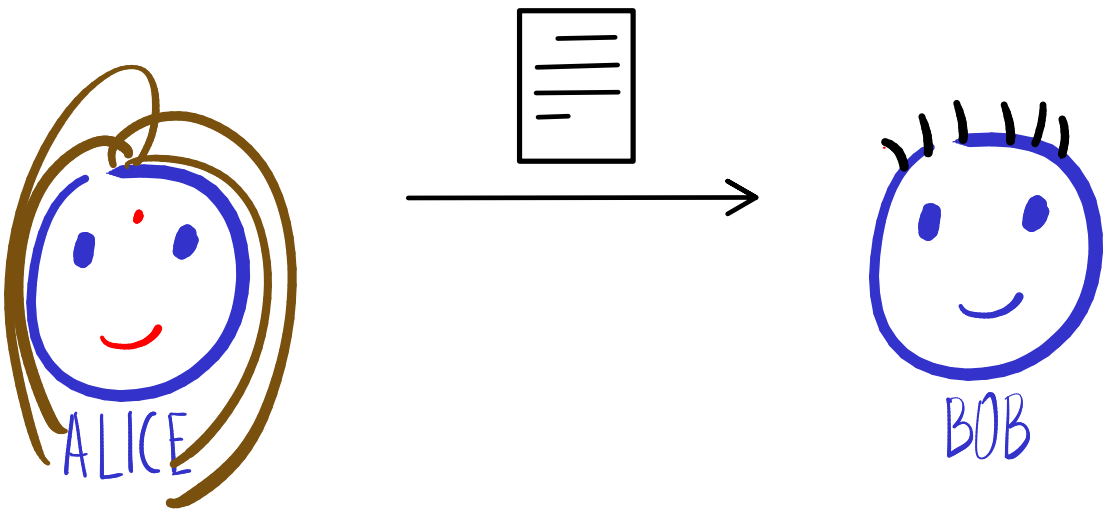
2. One-Time Digital Signatures ← OWF

3. Many-Time (Stateful) Digital Signatures

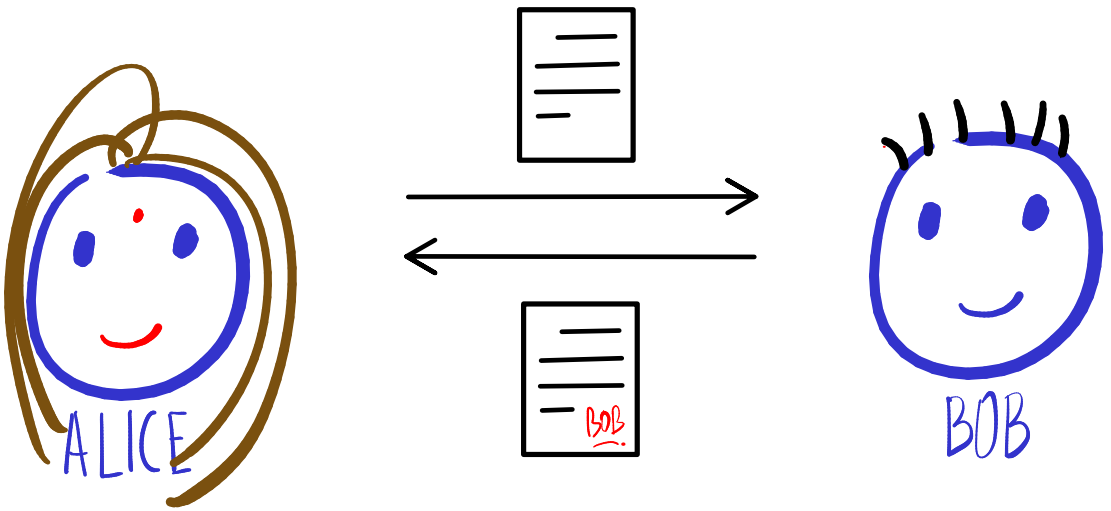
Digital (Analogues of Physical) Signatures...



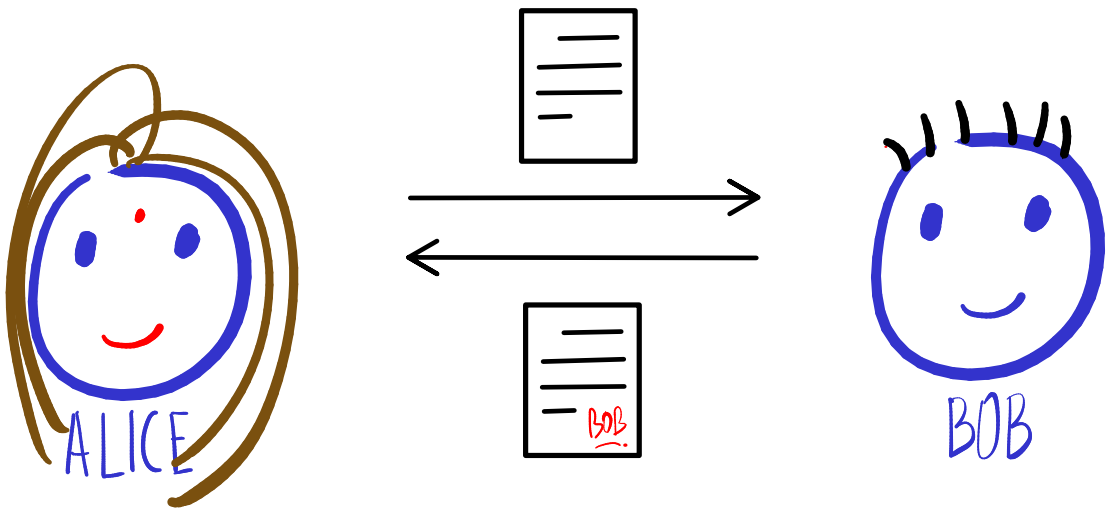
Digital (Analogues of Physical) Signatures...



Digital (Analogues of Physical) Signatures...



Digital (Analogues of Physical) Signatures...

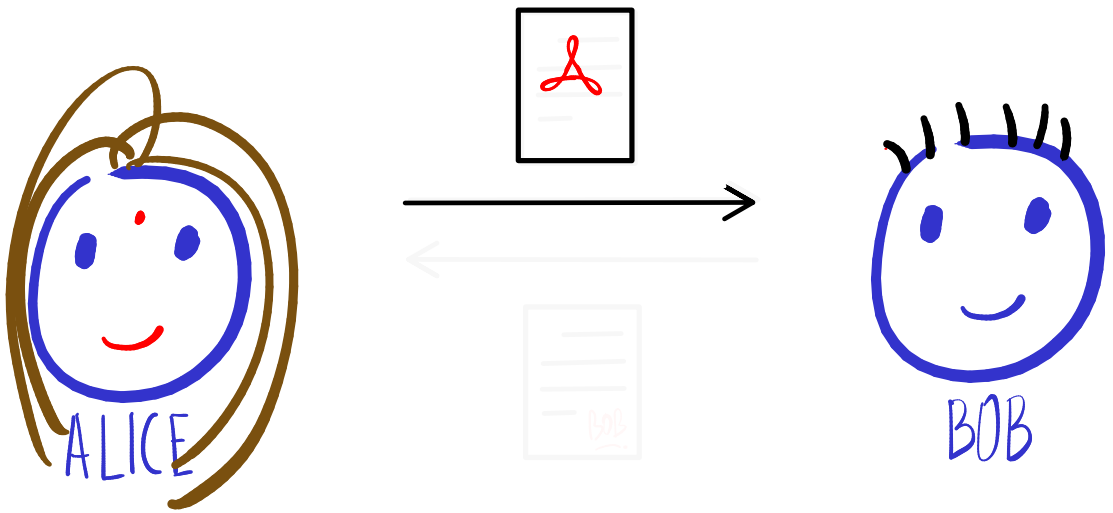


- Requirement: no one should be able to forge Bob's signature

Digital (Analogues of Physical) Signatures...

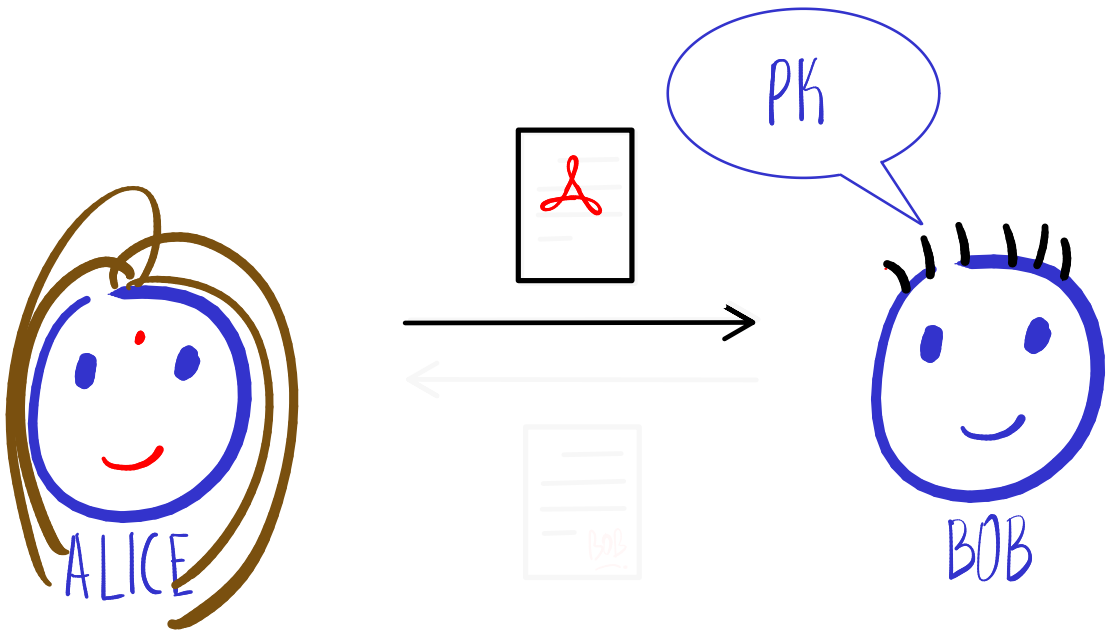
- Requirement: no one should be able to forge Bob's signature

Digital (Analogues of Physical) Signatures...



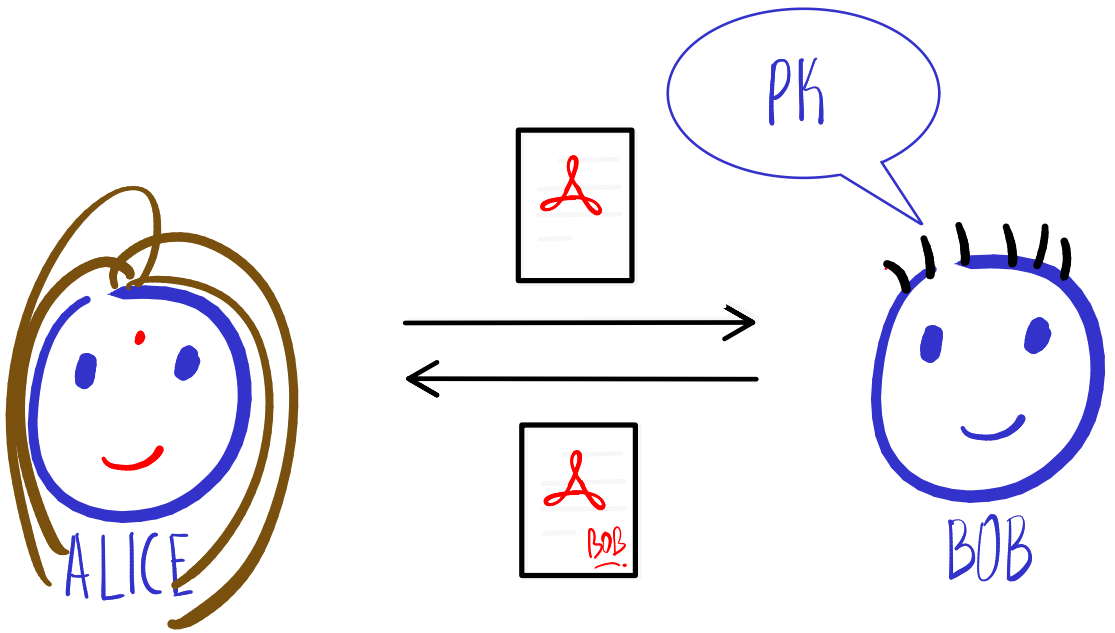
- Requirement: no one should be able to forge Bob's signature

Digital (Analogues of Physical) Signatures...



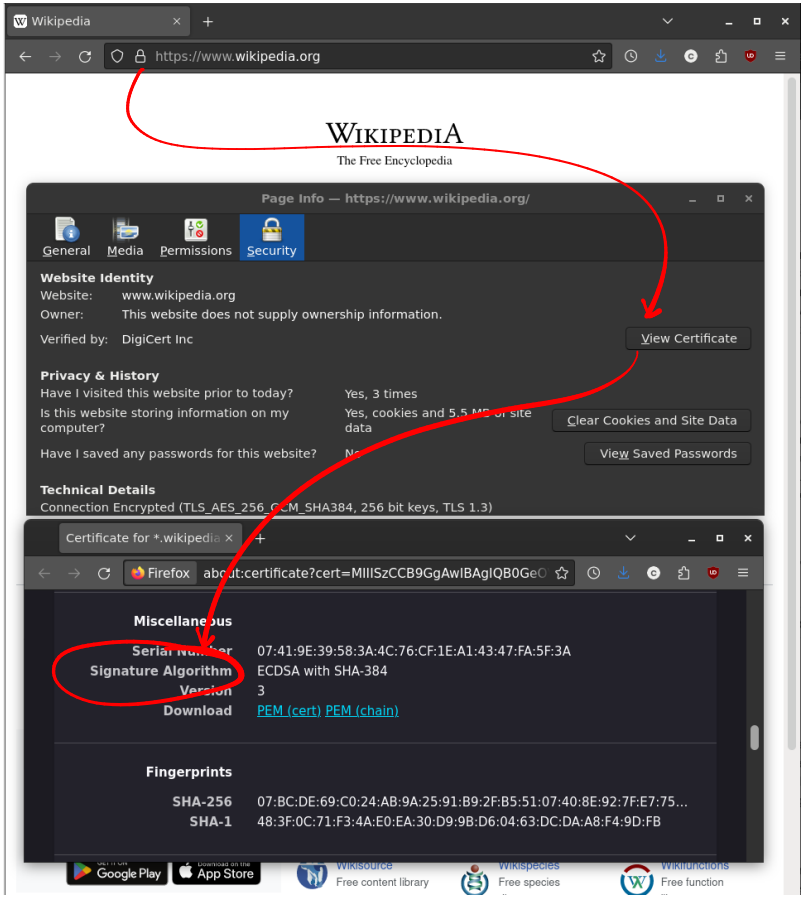
- Requirement: no one should be able to forge Bob's signature

Digital (Analogues of Physical) Signatures...

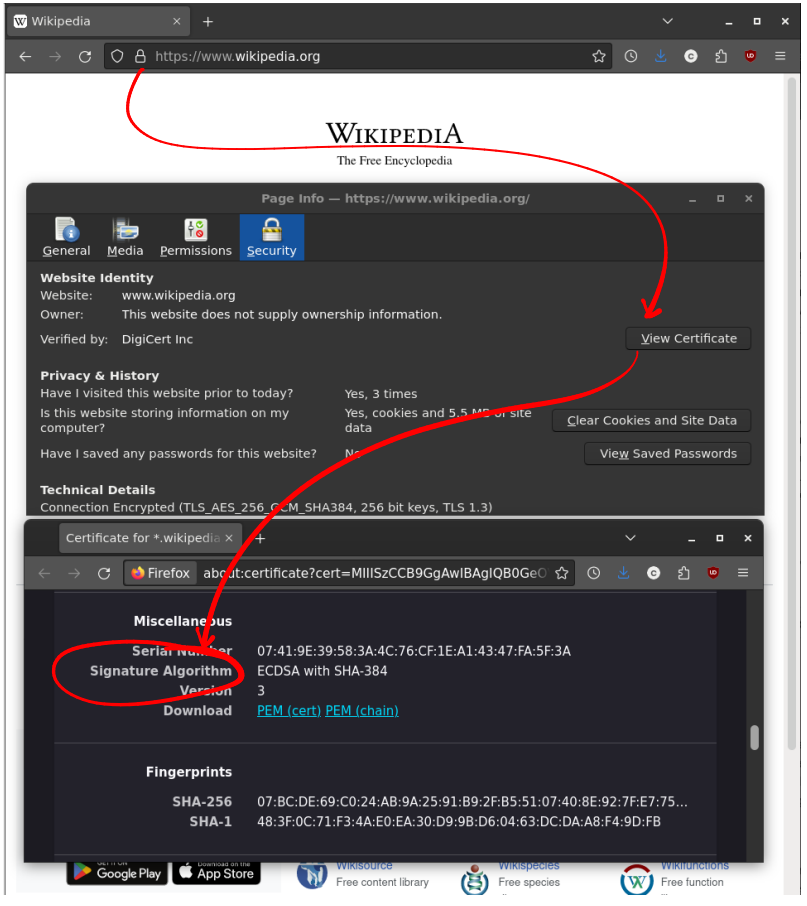


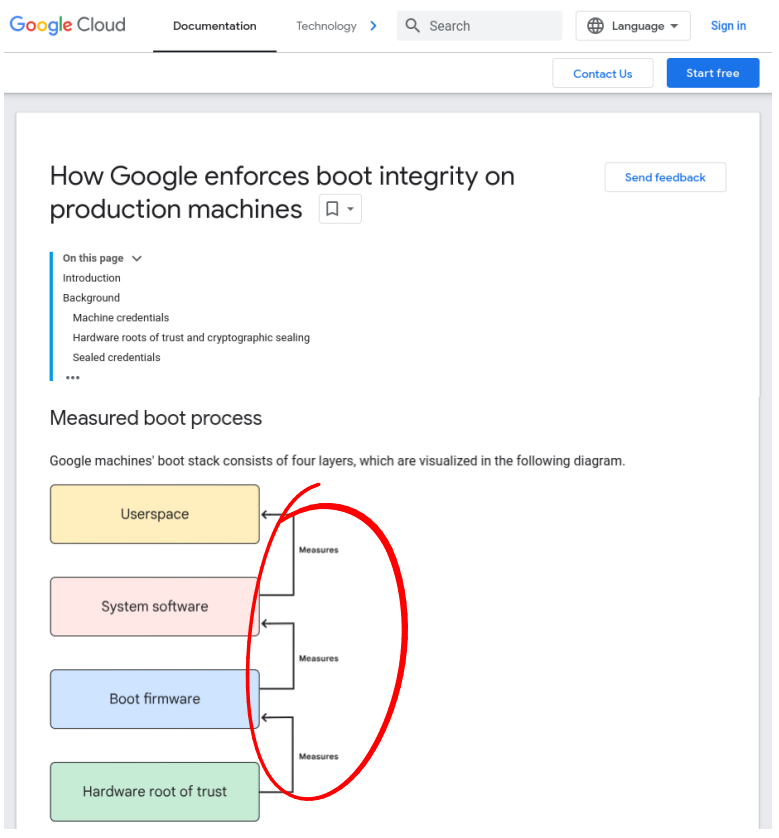
- Requirement: no one should be able to forge Bob's signature

Digital (Analogues of Physical) Signatures...



Digital (Analogues of Physical) Signatures...





Digital Signatures: Syntax

- *Public-key* analogue of message authentication codes (MAC)

Defintion 1 (Digital signature (DS))

A DS Σ *is a triple of efficient algorithms* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *with the following syntax:*

Digital Signatures: Syntax

- *Public-key* analogue of message authentication codes (MAC)

Defintion 1 (Digital signature (DS))

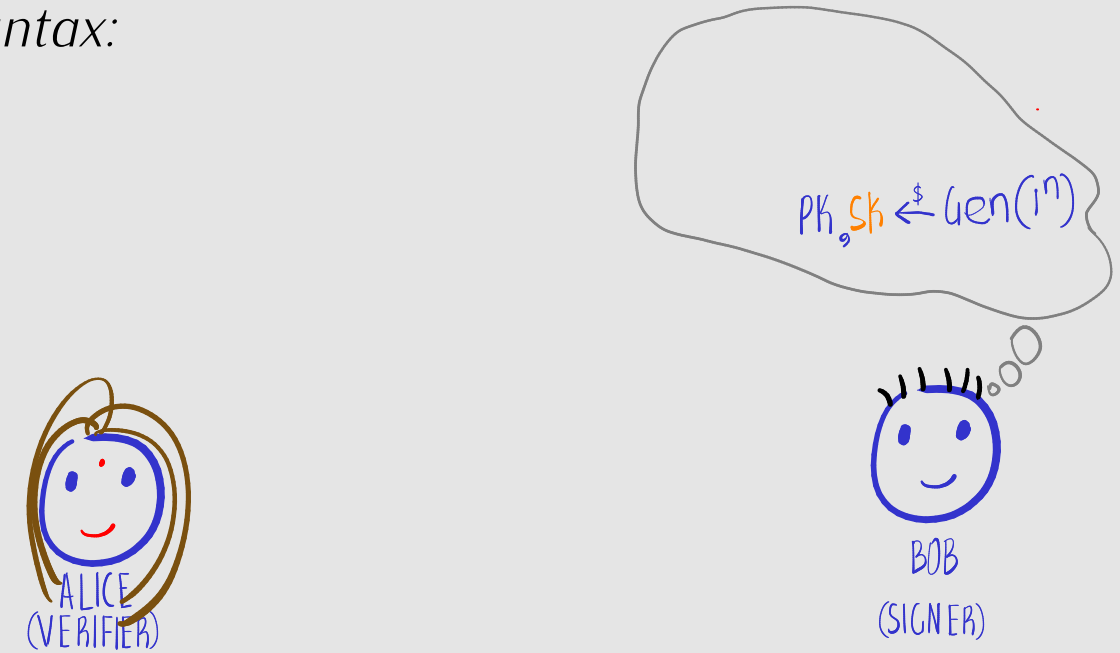
A DS Σ *is a triple of efficient algorithms* (Gen, Sign, Ver) *with the following syntax:*

Digital Signatures: Syntax

- *Public-key* analogue of message authentication codes (MAC)

Defintion 1 (Digital signature (DS))

A DS Σ *is a triple of efficient algorithms* (Gen, Sign, Ver) *with the following syntax:*

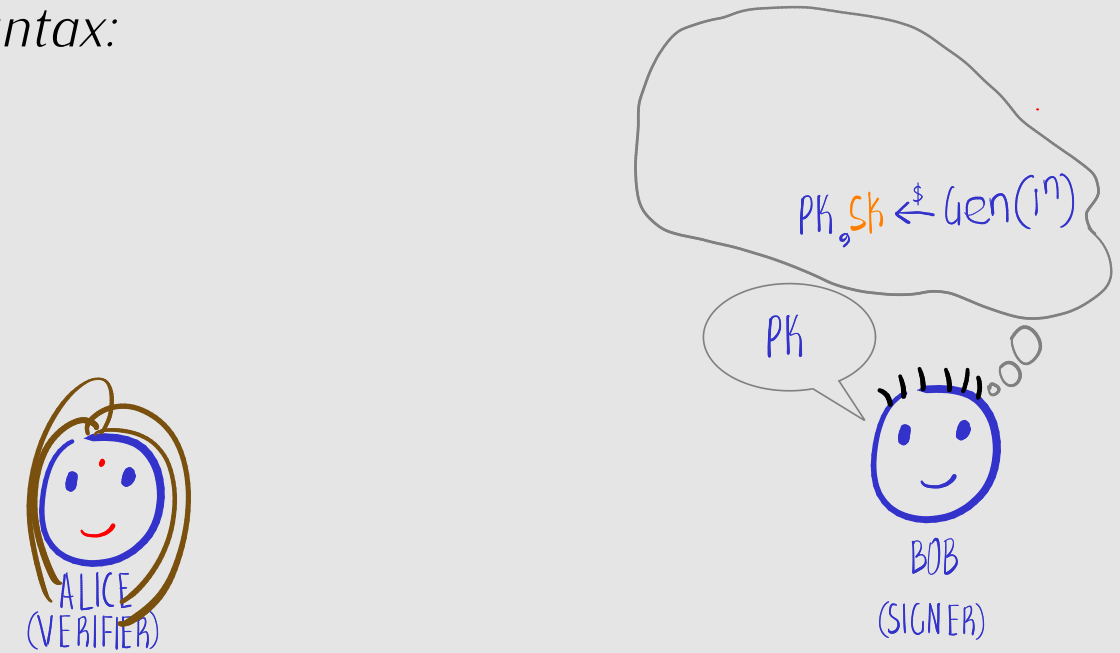


Digital Signatures: Syntax

- *Public-key* analogue of message authentication codes (MAC)

Defintion 1 (Digital signature (DS))

A DS Σ *is a triple of efficient algorithms* (Gen, Sign, Ver) *with the following syntax:*

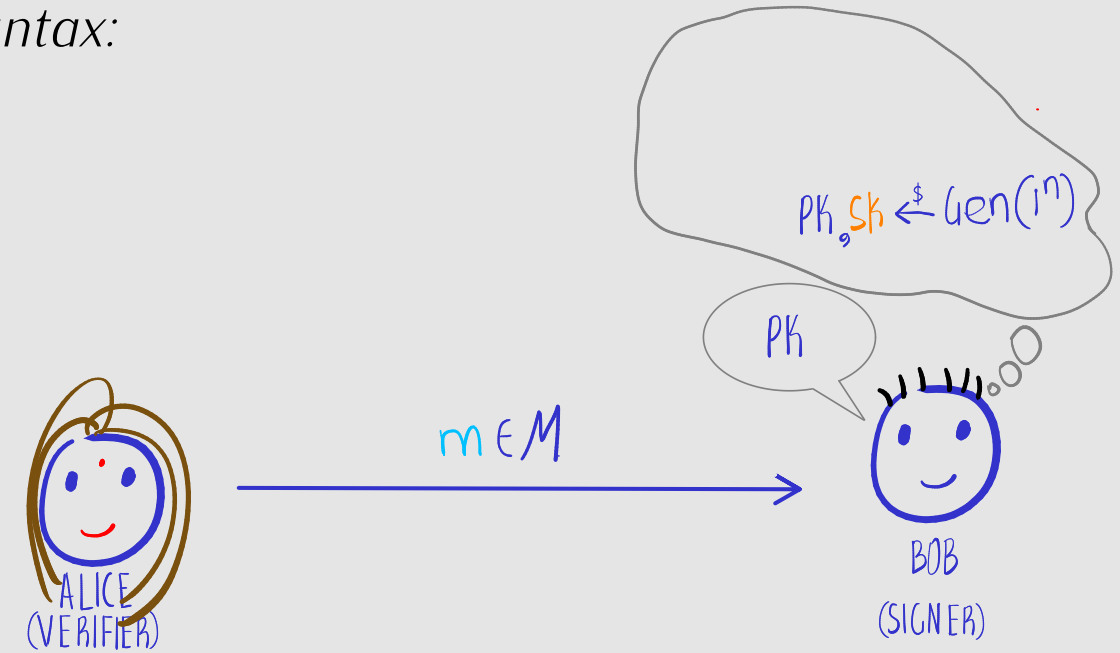


Digital Signatures: Syntax

- *Public-key* analogue of message authentication codes (MAC)

Defintion 1 (Digital signature (DS))

A DS Σ *is a triple of efficient algorithms* (Gen, Sign, Ver) *with the following syntax:*

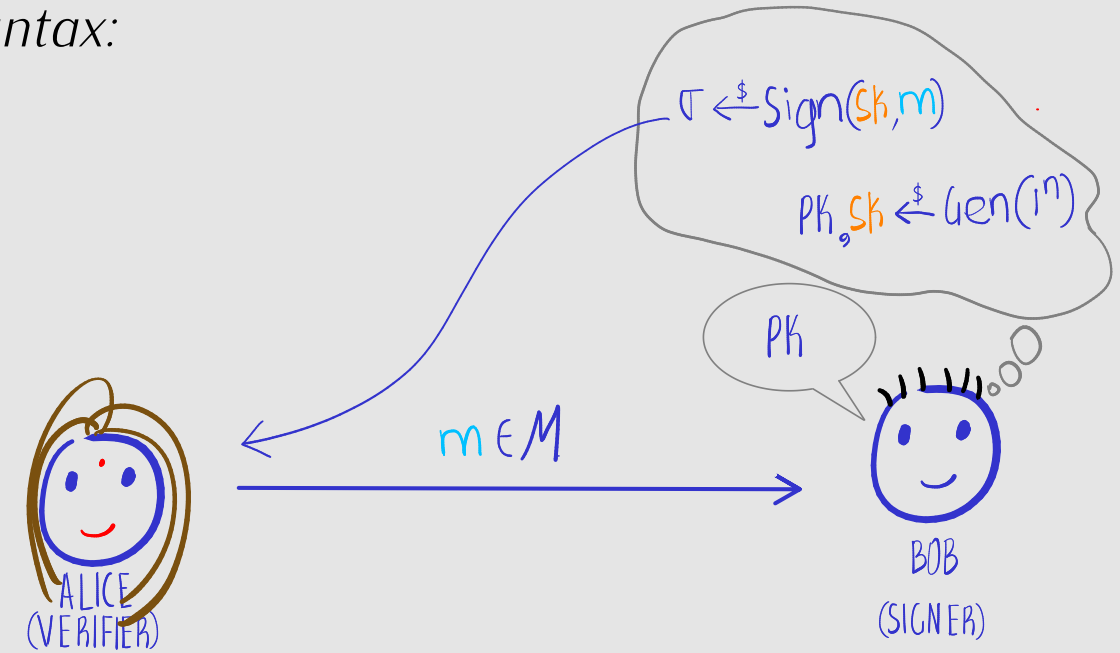


Digital Signatures: Syntax

- *Public-key* analogue of message authentication codes (MAC)

Defintion 1 (Digital signature (DS))

A DS Σ *is a triple of efficient algorithms* (Gen, Sign, Ver) *with the following syntax:*

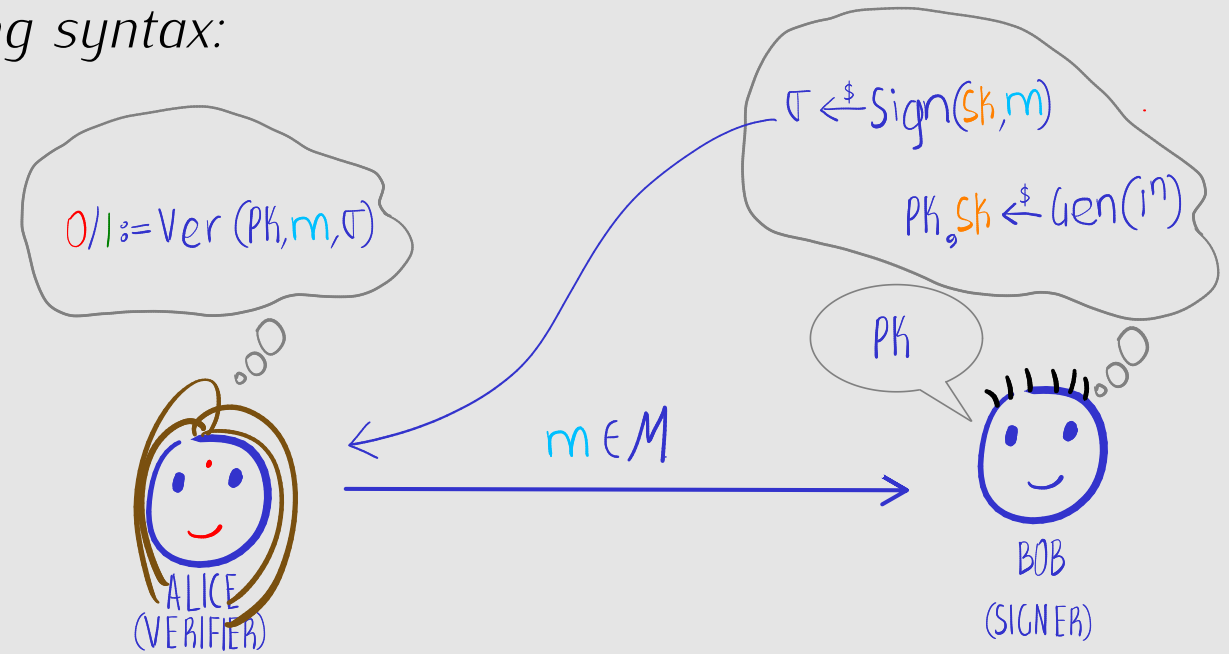


Digital Signatures: Syntax

- *Public-key* analogue of message authentication codes (MAC)

Defintion 1 (Digital signature (DS))

A DS Σ *is a triple of efficient algorithms* (Gen, Sign, Ver) *with the following syntax:*

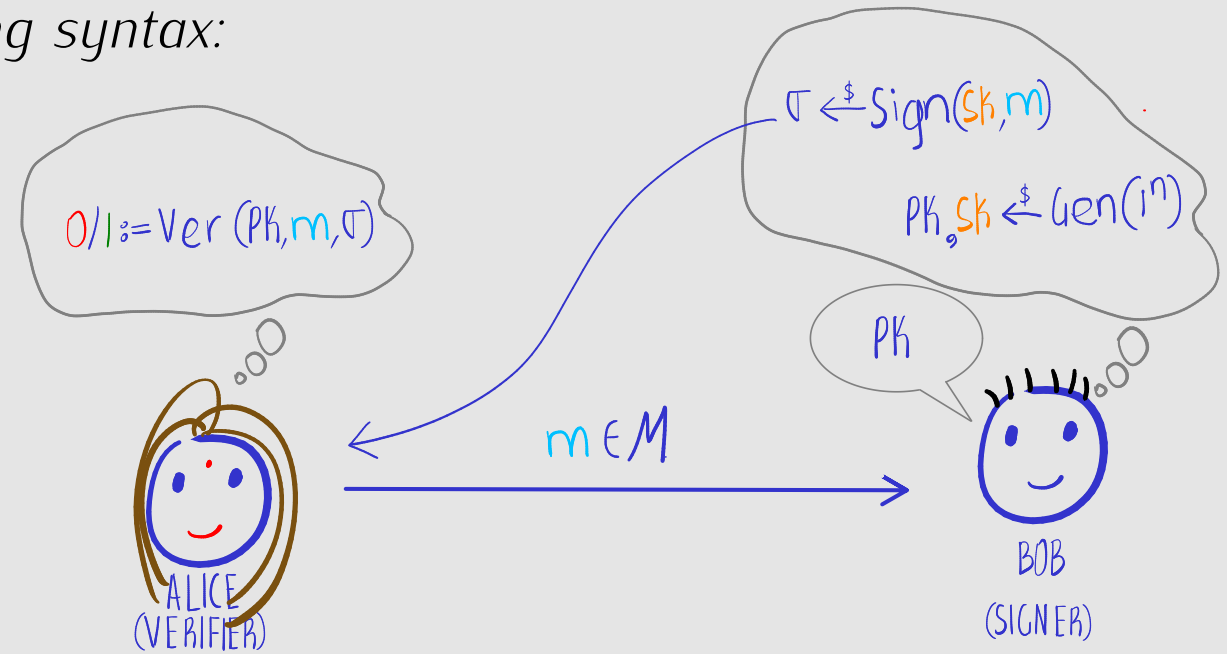


Digital Signatures: Syntax

- *Public-key* analogue of message authentication codes (MAC)

Defintion 1 (Digital signature (DS))

A DS Σ *is a triple of efficient algorithms* (Gen, Sign, Ver) *with the following syntax:*



- *Correctness of honest signing: for every* $n \in \mathbb{N}$*, message* $m \in \mathcal{M}_n$*,*

$$\Pr_{(\mathsf{pk},\mathsf{sk})\leftarrow\mathbf{Gen}(1^n),\sigma\leftarrow\mathbf{Sign}(\mathsf{sk},m)}[\mathbf{Ver}(\mathsf{pk},\sigma,m)=1]=1$$

How to Define Security?...

- Intuitively, what are the security requirements?

How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a valid *new* signature from previously-seen signatures...

How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a valid *new* signature from previously-seen signatures...
    - ... on messages of its choice

How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a valid *new* signature from previously-seen signatures...
    - ... on messages of its choice
  - Forged new signature can be on *any* message of Tam's choice

How to Define Security?...

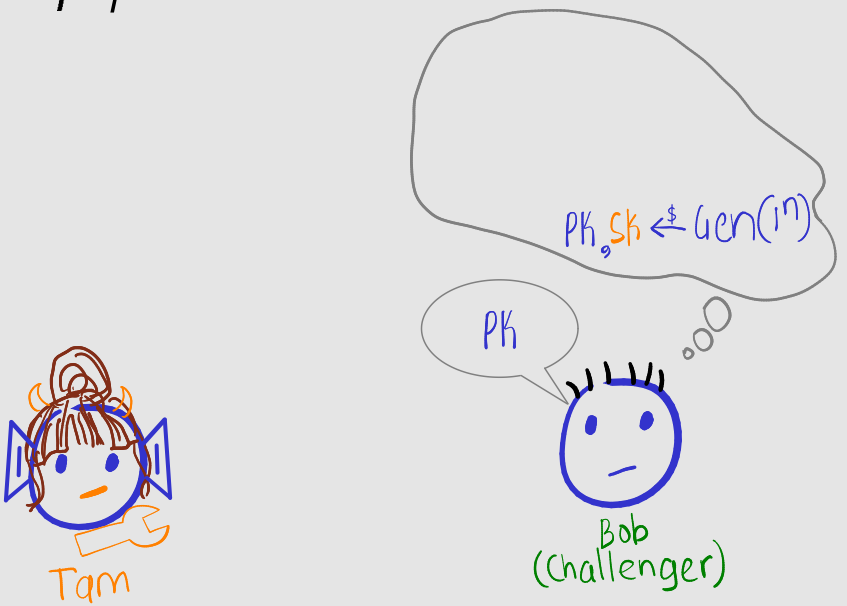
- Intuitively, what are the security requirements?
  - Tam must not be able to forge a valid *new* signature from previously-seen signatures...
    - ... on messages of its choice
  - Forged new signature can be on *any* message of Tam's choice
- Existential Unforgeability Under Chosen-Message Attack

How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a valid *new* signature from previously-seen signatures...
    - ... on messages of its choice
  - Forged new signature can be on *any* message of Tam's choice
- Existential Unforgeability Under Chosen-Message Attack

Defintion 2 (EU-CMA)

A DS $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *is* q*-EU-CMA secure if no PPT adversary Tam that makes at most* q *queries can break* Σ *as follows with non-negligible probability.*



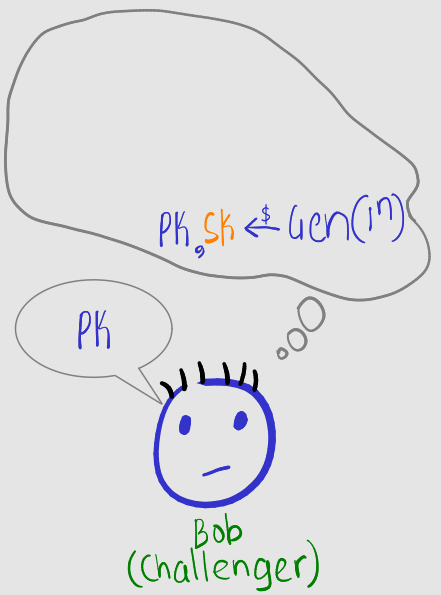
How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a valid *new* signature from previously-seen signatures...
    - ... on messages of its choice
  - Forged new signature can be on *any* message of Tam's choice
- Existential Unforgeability Under Chosen-Message Attack

Defintion 2 (EU-CMA)

A DS $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *is* q*-EU-CMA secure if no PPT adversary* Tam *that makes at most* q *queries can break* Σ *as follows with non-negligible probability.*

- Tam given PK



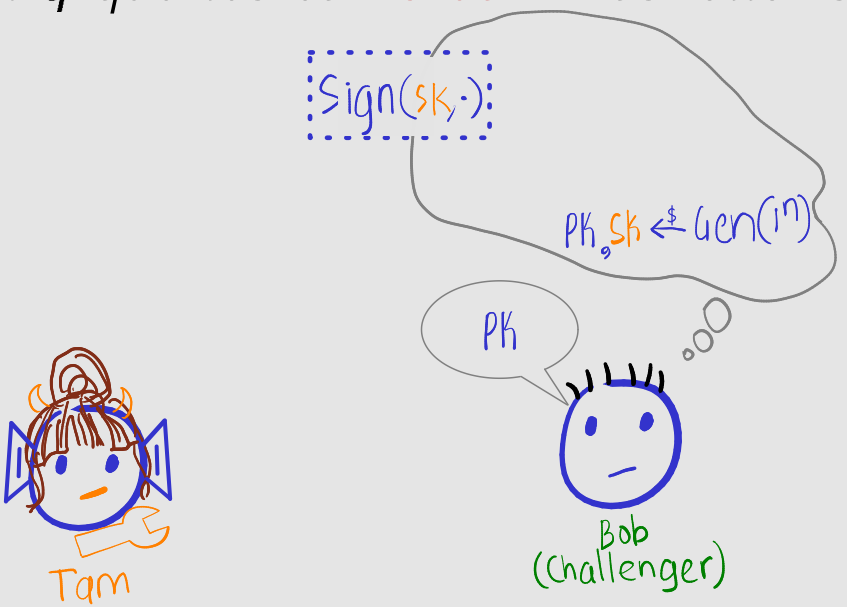
How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a valid *new* signature from previously-seen signatures...
    - ... on messages of its choice
  - Forged new signature can be on *any* message of Tam's choice
- Existential Unforgeability Under Chosen-Message Attack

Defintion 2 (EU-CMA)

A DS $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *is* q*-EU-CMA secure if no PPT adversary* Tam *that makes at most* q *queries can break* Σ *as follows with non-negligible probability.*

- Tam given PK



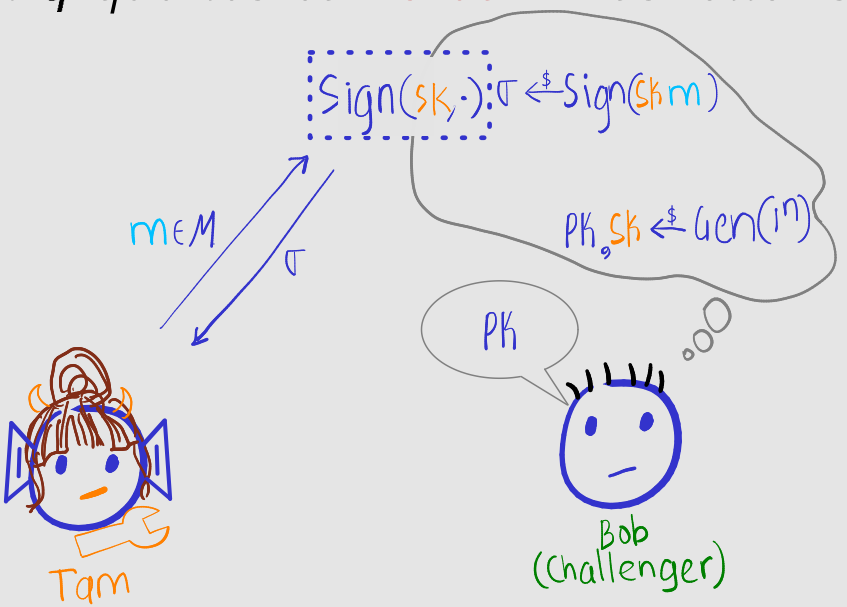
How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a valid *new* signature from previously-seen signatures...
    - ... on messages of its choice
  - Forged new signature can be on *any* message of Tam's choice
- Existential Unforgeability Under Chosen-Message Attack

Defintion 2 (EU-CMA)

A DS $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *is* q*-EU-CMA secure if no PPT adversary Tam that makes at most* q *queries can break* Σ *as follows with non-negligible probability.*

- Tam given PK



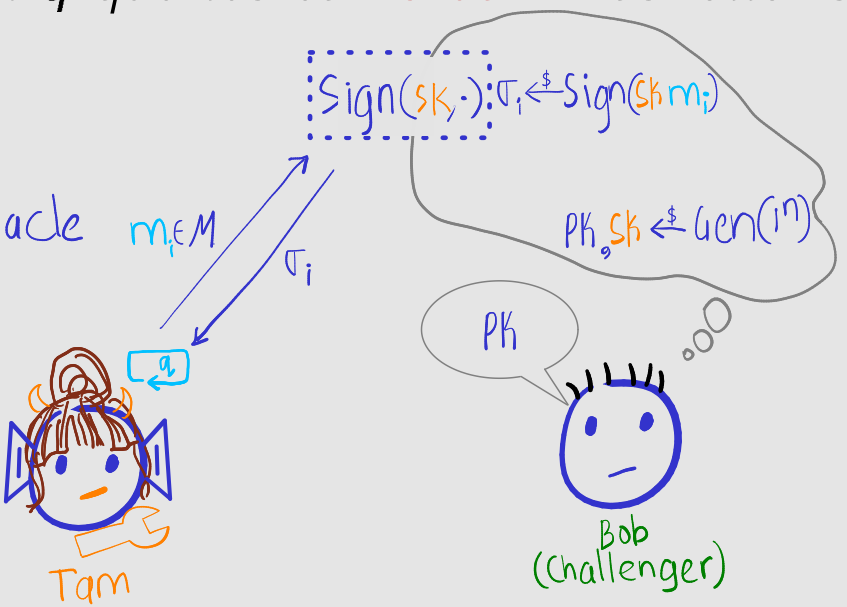
How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a valid *new* signature from previously-seen signatures...
    - ... on messages of its choice
  - Forged new signature can be on *any* message of Tam's choice
- Existential Unforgeability Under Chosen-Message Attack

Defintion 2 (EU-CMA)

A DS $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *is* q*-EU-CMA secure if no PPT adversary Tam that makes at most* q *queries can break* Σ *as follows with non-negligible probability.*

- Tam given PK
- Tam makes q queries to $\mathsf{Sign}(sk, \cdot)$ oracle



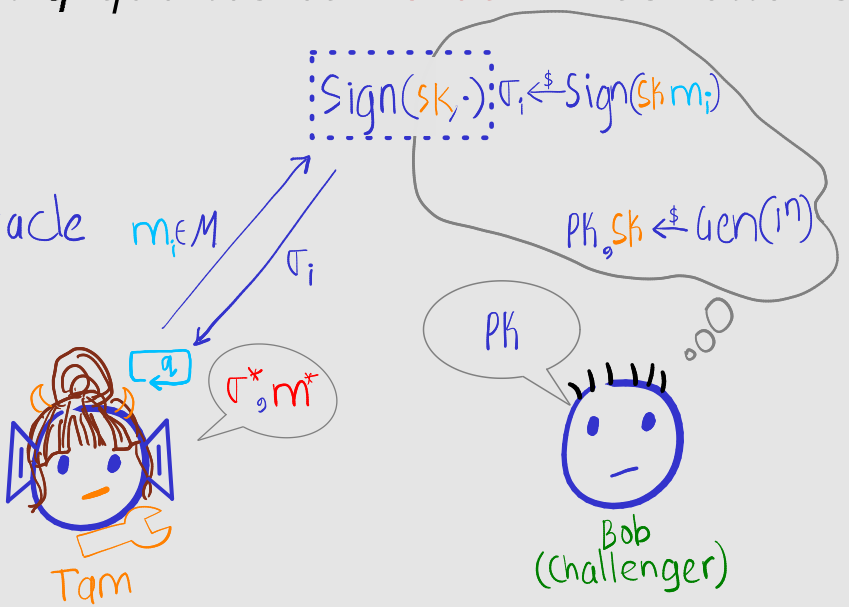
How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a valid *new* signature from previously-seen signatures...
    - ... on messages of its choice
  - Forged new signature can be on *any* message of Tam's choice
- Existential Unforgeability Under Chosen-Message Attack

Defintion 2 (EU-CMA)

A DS $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *is* q*-EU-CMA secure if no PPT adversary* Tam *that makes at most* q *queries can break* Σ *as follows with non-negligible probability.*

- Tam given PK
- Tam makes q queries to $\mathsf{Sign}(sk, \cdot)$ oracle
- In the end Tam outputs (σ^*, m^*) and breaks Σ if:
  - $\mathsf{Ver}(PK, m^*, \sigma^*) = 1$
  - $\forall i \in [q]: m^* \neq m_i$



EU-CMA Secure or Not?

$\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ EU-CMA-secure $\Rightarrow$ Σ' EU-CMA-secure?

1. Truncate-then-sign: define Σ' as
   - $\mathsf{Sign}'(\mathsf{sk}, m := m_1 \cdots m_{\ell-1} m_\ell) \leftarrow \mathsf{Sign}(\mathsf{sk}, m_1 \cdots m_{\ell-1})$
   - $\mathsf{Ver}'(\mathsf{pk}, \sigma, m) := \mathsf{Ver}(\mathsf{pk}, \sigma, m_1 \cdots m_{\ell-1})$

EU-CMA Secure or Not?

$\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ EU-CMA-secure $\Rightarrow$ Σ' EU-CMA-secure?

1. Truncate-then-sign: define Σ' as
   - $\mathsf{Sign}'(\mathsf{sk}, m := m_1 \cdots m_{\ell-1} m_\ell) \leftarrow \mathsf{Sign}(\mathsf{sk}, m_1 \cdots m_{\ell-1})$
   - $\mathsf{Ver}'(\mathsf{pk}, \sigma, m) := \mathsf{Ver}(\mathsf{pk}, \sigma, m_1 \cdots m_{\ell-1})$
2. Sign-then-truncate: define Σ' as
   - $\mathsf{Sign}'(\mathsf{sk}, m) := \sigma_1 \cdots \sigma_{s-1}$, where $\sigma_1 \cdots \sigma_{s-1} \sigma_s \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$
   - $\mathsf{Ver}'(\mathsf{pk}, \sigma', m)$: accept if
     - $\mathsf{Ver}(\mathsf{pk}, \sigma'0, m) = 1$ *or* $\mathsf{Ver}(\mathsf{pk}, \sigma'1, m) = 1$

EU-CMA Secure or Not?

$\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ EU-CMA-secure $\Rightarrow$ Σ' EU-CMA-secure?

1. Truncate-then-sign: define Σ' as
   - $\mathsf{Sign}'(\mathsf{sk}, m := m_1 \cdots m_{\ell-1} m_\ell) \leftarrow \mathsf{Sign}(\mathsf{sk}, m_1 \cdots m_{\ell-1})$
   - $\mathsf{Ver}'(\mathsf{pk}, \sigma, m) := \mathsf{Ver}(\mathsf{pk}, \sigma, m_1 \cdots m_{\ell-1})$
2. Sign-then-truncate: define Σ' as
   - $\mathsf{Sign}'(\mathsf{sk}, m) := \sigma_1 \cdots \sigma_{s-1}$, where $\sigma_1 \cdots \sigma_{s-1} \sigma_s \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$
   - $\mathsf{Ver}'(\mathsf{pk}, \sigma', m)$: accept if
     - $\mathsf{Ver}(\mathsf{pk}, \sigma'0, m) = 1$ *or* $\mathsf{Ver}(\mathsf{pk}, \sigma'1, m) = 1$
3. Sign-then-append: define Σ' as
   - $\mathsf{Sign}'(\mathsf{sk}, m) := \sigma 0$, where $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$
   - $\mathsf{Ver}'(\mathsf{pk}, \sigma b, m) := \mathsf{Ver}(\mathsf{pk}, \sigma, m)$

EU-CMA Secure or Not?

$\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ EU-CMA-secure $\Rightarrow$ Σ' EU-CMA-secure?

1. Truncate-then-sign: define Σ' as
   - $\mathsf{Sign}'(\mathsf{sk}, m := m_1 \cdots m_{\ell-1} m_\ell) \leftarrow \mathsf{Sign}(\mathsf{sk}, m_1 \cdots m_{\ell-1})$
   - $\mathsf{Ver}'(\mathsf{pk}, \sigma, m) := \mathsf{Ver}(\mathsf{pk}, \sigma, m_1 \cdots m_{\ell-1})$
2. Sign-then-truncate: define Σ' as
   - $\mathsf{Sign}'(\mathsf{sk}, m) := \sigma_1 \cdots \sigma_{s-1}$, where $\sigma_1 \cdots \sigma_{s-1} \sigma_s \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$
   - $\mathsf{Ver}'(\mathsf{pk}, \sigma', m)$: accept if
     - $\mathsf{Ver}(\mathsf{pk}, \sigma'0, m) = 1$ *or* $\mathsf{Ver}(\mathsf{pk}, \sigma'1, m) = 1$
3. Sign-then-append: define Σ' as
   - $\mathsf{Sign}'(\mathsf{sk}, m) := \sigma 0$, where $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$
   - $\mathsf{Ver}'(\mathsf{pk}, \sigma b, m) := \mathsf{Ver}(\mathsf{pk}, \sigma, m)$

Exercise 1

Prove by reduction that the Σ's in 1 and 3 are EU-CMA-secure.

Plan for Today's Lecture

1 Digital Signature (DS)

2 One-Time Digital Signatures ← OWF

3 Many-Time (Stateful) Digital Signatures

One-Time DS ($q = 1$): Lamport's Signature

$\{f_n : \{0,1\}^n \to \{0,1\}^n\}_n$

Construction 1 (OWF $f \to$ one-time DS Σ for $\mathcal{M} := \{0,1\}^\ell$)

One-Time DS ($q = 1$): Lamport's Signature

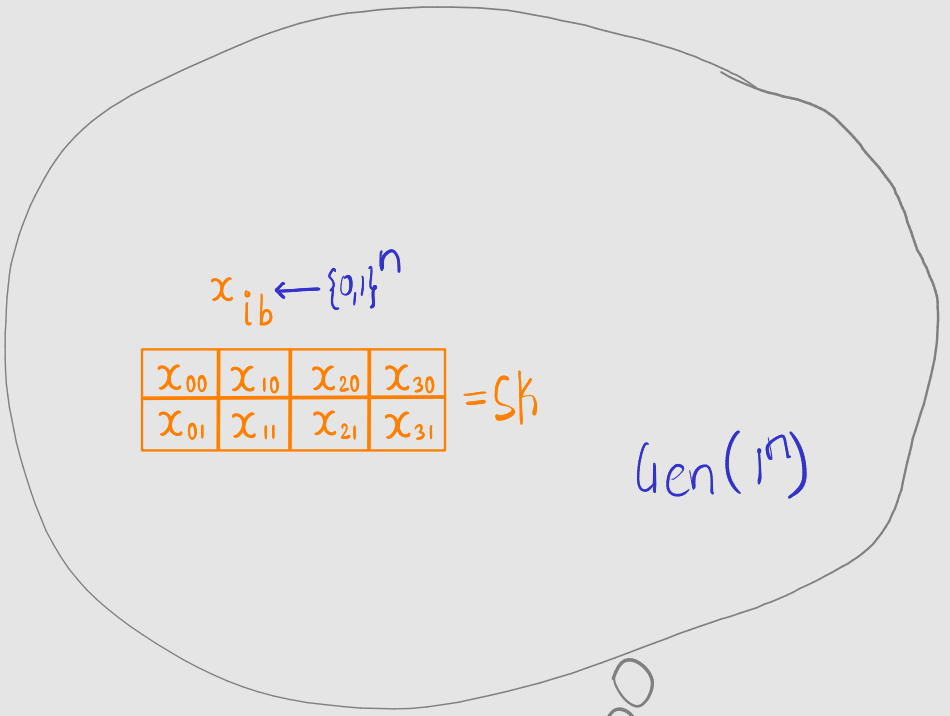
$\{f_n : \{0,1\}^n \to \{0,1\}^n\}_n$

Construction 1 (OWF $f \to$ one-time DS Σ for $\mathcal{M} := \{0,1\}^\ell$)

One-Time DS ($q = 1$): Lamport's Signature

$\{f_n : \{0,1\}^n \to \{0,1\}^n\}_n$

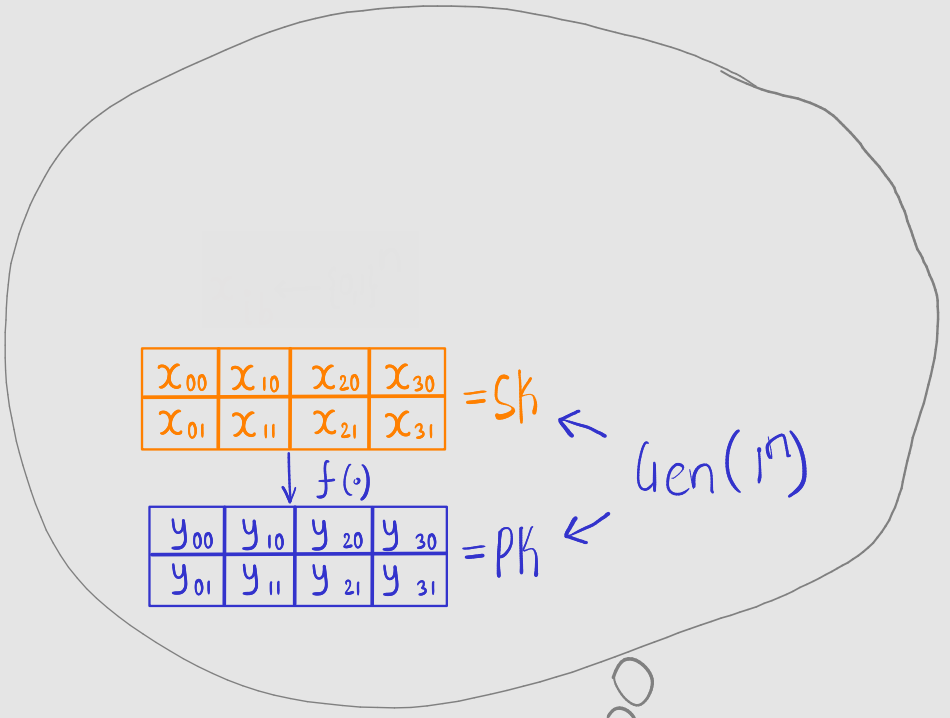
Construction 1 (OWF $f \to$ one-time DS Σ for $\mathcal{M} := \{0,1\}^{\ell}$) $\ell = 4$



One-Time DS ($q = 1$): Lamport's Signature

$\{f_n : \{0,1\}^n \to \{0,1\}^n\}_n$

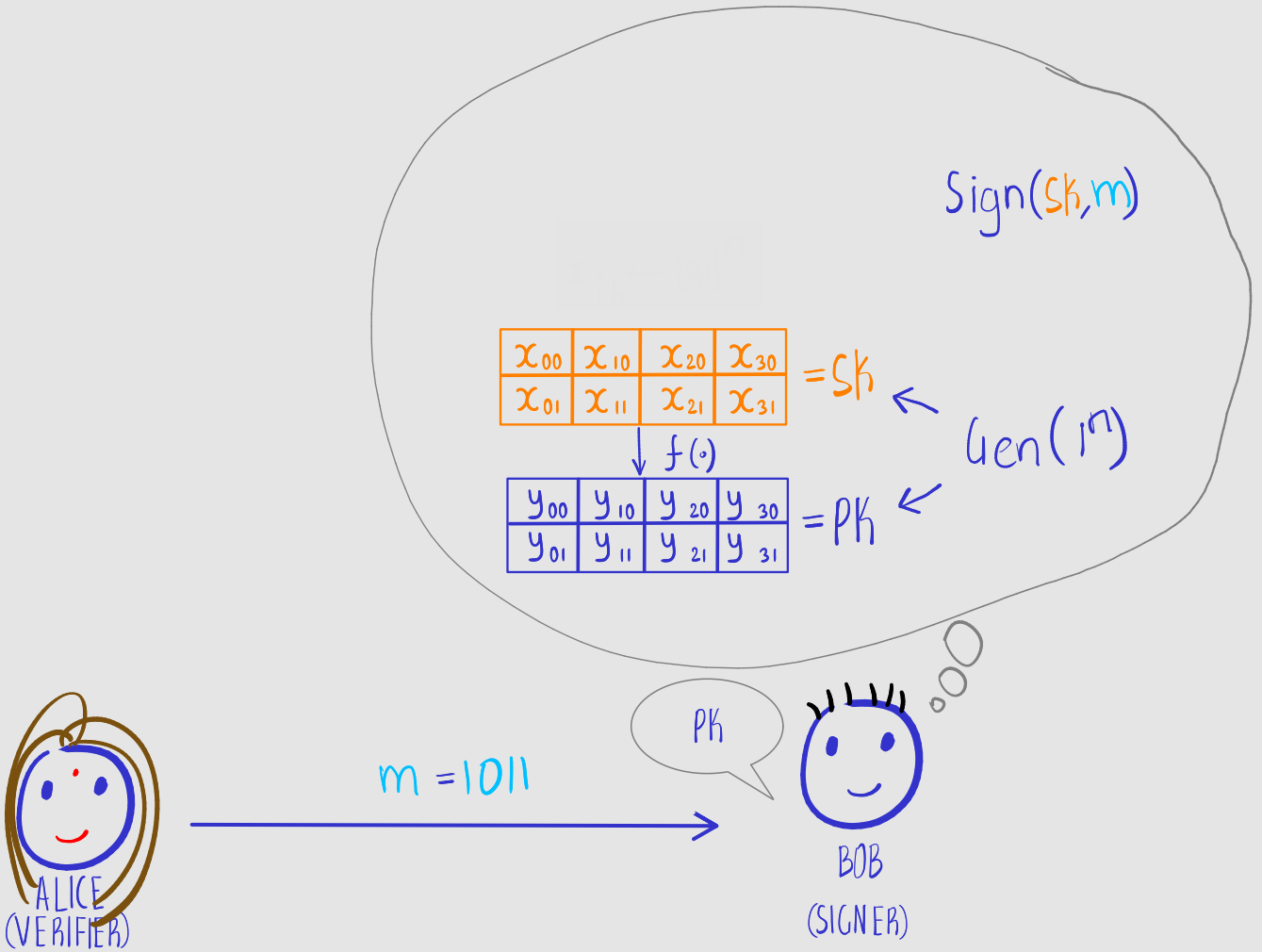
Construction 1 (OWF $f \to$ one-time DS Σ for $\mathcal{M} := \{0,1\}^{\ell}$) $\ell = 4$



One-Time DS ($q = 1$): Lamport's Signature

$\{f_n : \{0,1\}^n \to \{0,1\}^n\}_n$

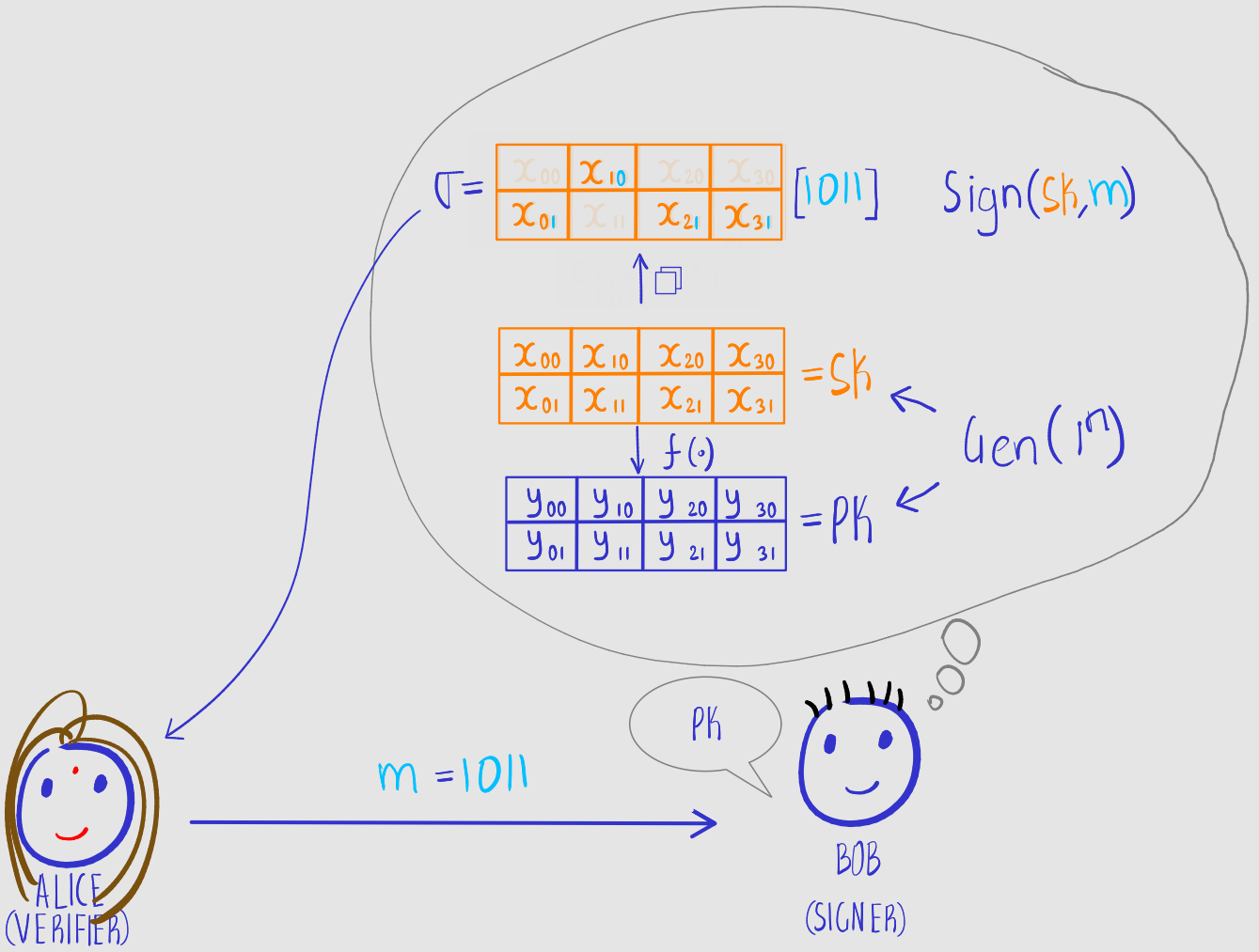
Construction 1 (OWF $f \to$ one-time DS Σ for $\mathcal{M} := \{0,1\}^{\ell}$) $\ell = 4$



One-Time DS ($q = 1$): Lamport's Signature

$\{f_n : \{0,1\}^n \to \{0,1\}^n\}_n$

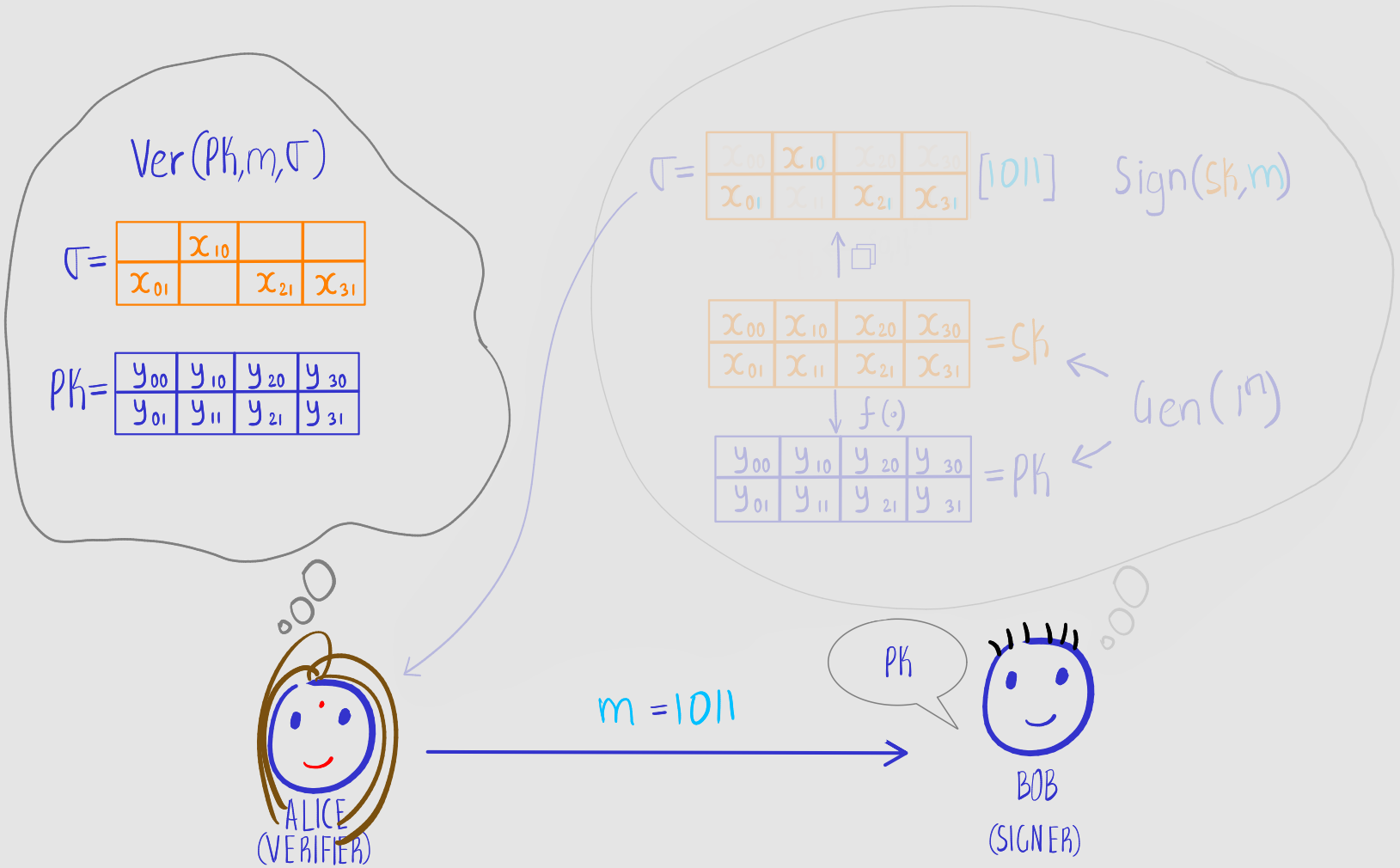
Construction 1 (OWF $f \to$ one-time DS Σ for $\mathcal{M} := \{0,1\}^{\ell}$) $\ell = 4$



One-Time DS ($q = 1$): Lamport's Signature

$\{f_n : \{0,1\}^n \to \{0,1\}^n\}_n$

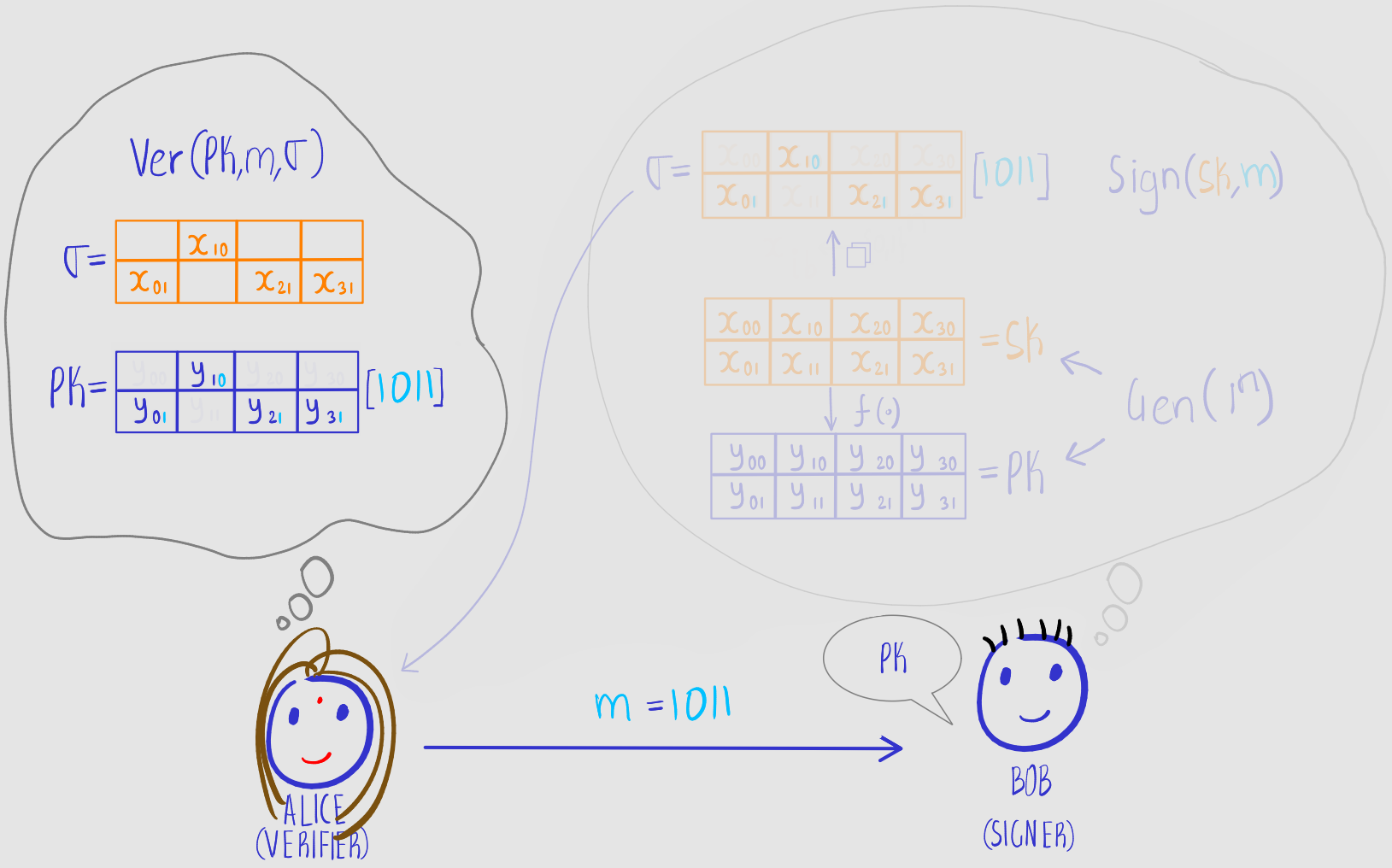
Construction 1 (OWF $f \to$ one-time DS Σ for $\mathcal{M} := \{0,1\}^{\ell}$) $\ell = 4$



One-Time DS ($q = 1$): Lamport's Signature

$\{f_n : \{0,1\}^n \to \{0,1\}^n\}_n$

Construction 1 (OWF $f \to$ one-time DS Σ for $\mathcal{M} := \{0,1\}^{\ell}$) $\ell = 4$

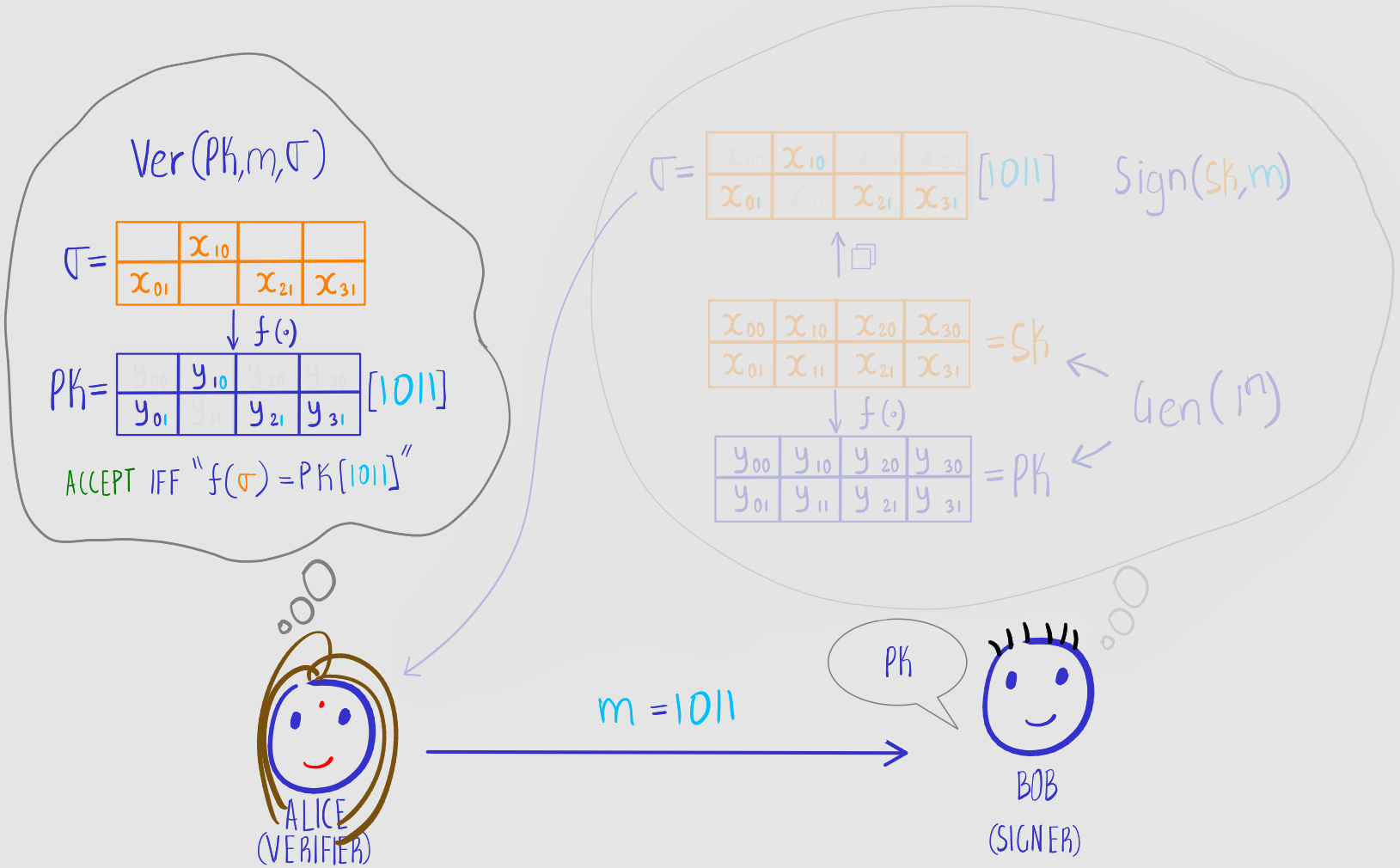


One-Time DS ($q = 1$): Lamport's Signature

$\{f_n : \{0,1\}^n \to \{0,1\}^n\}_n$

Construction 1 (OWF $f \to$ one-time DS Σ for $\mathcal{M} := \{0,1\}^{\ell}$)

$\ell = 4$



Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction.  Idea: "plug and pray".

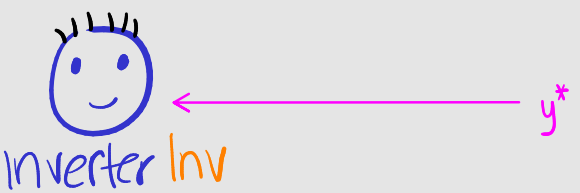
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction.  Idea: "plug and pray".



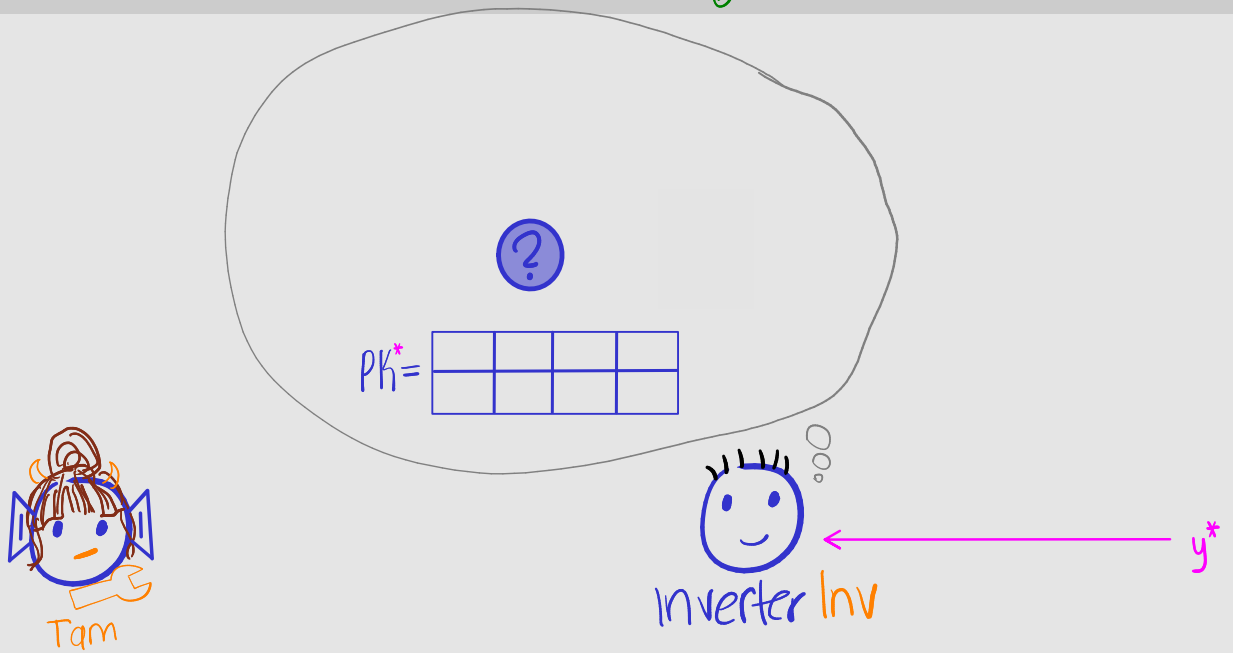
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



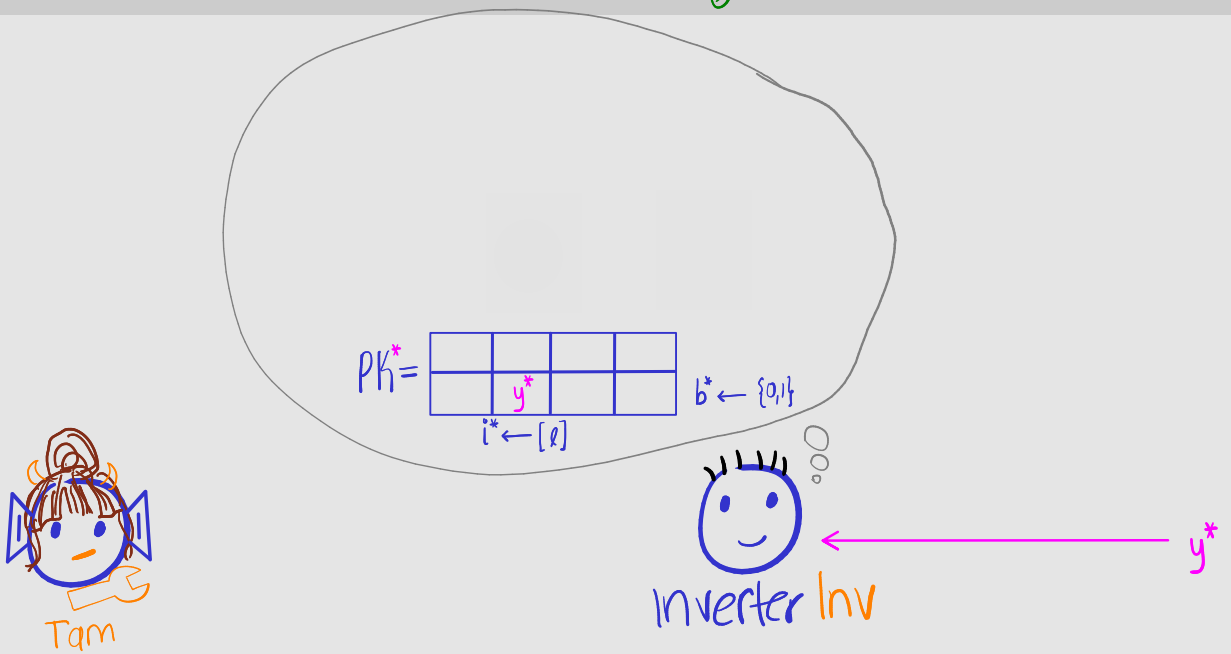
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



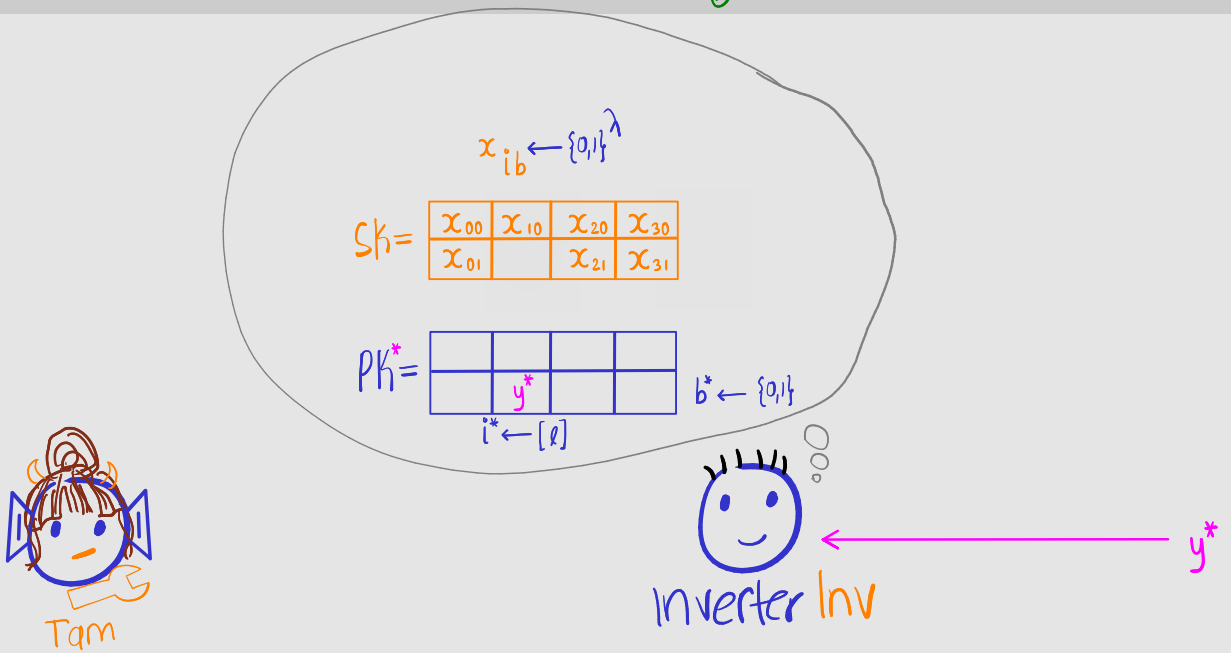
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



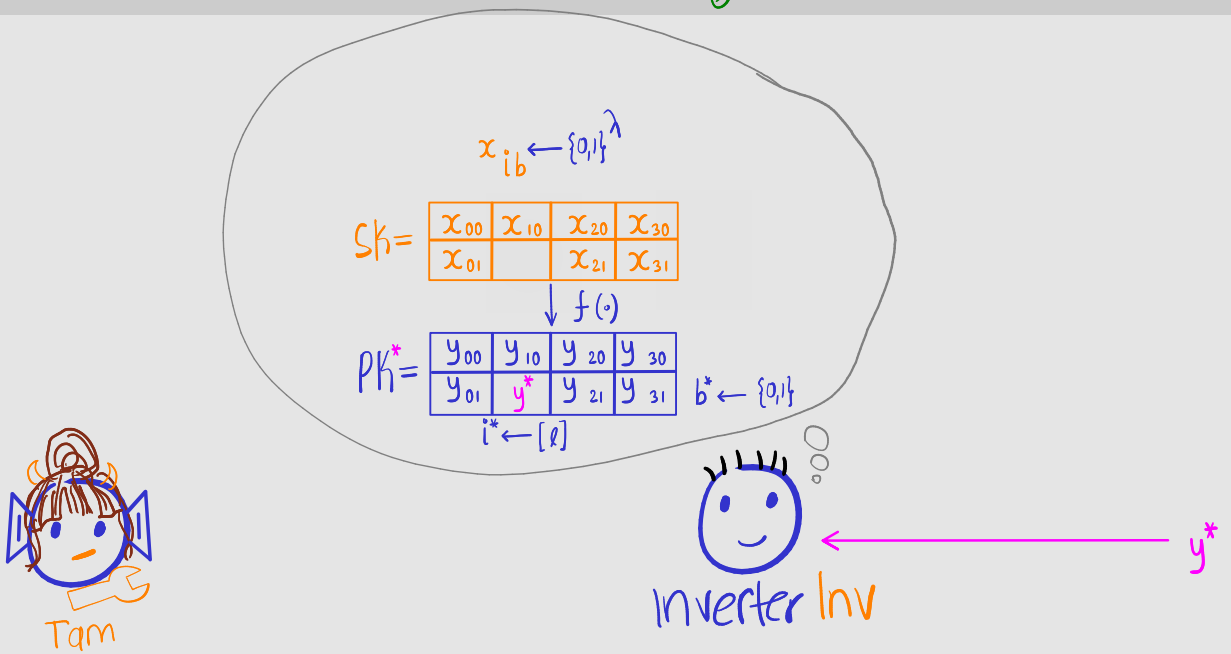
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



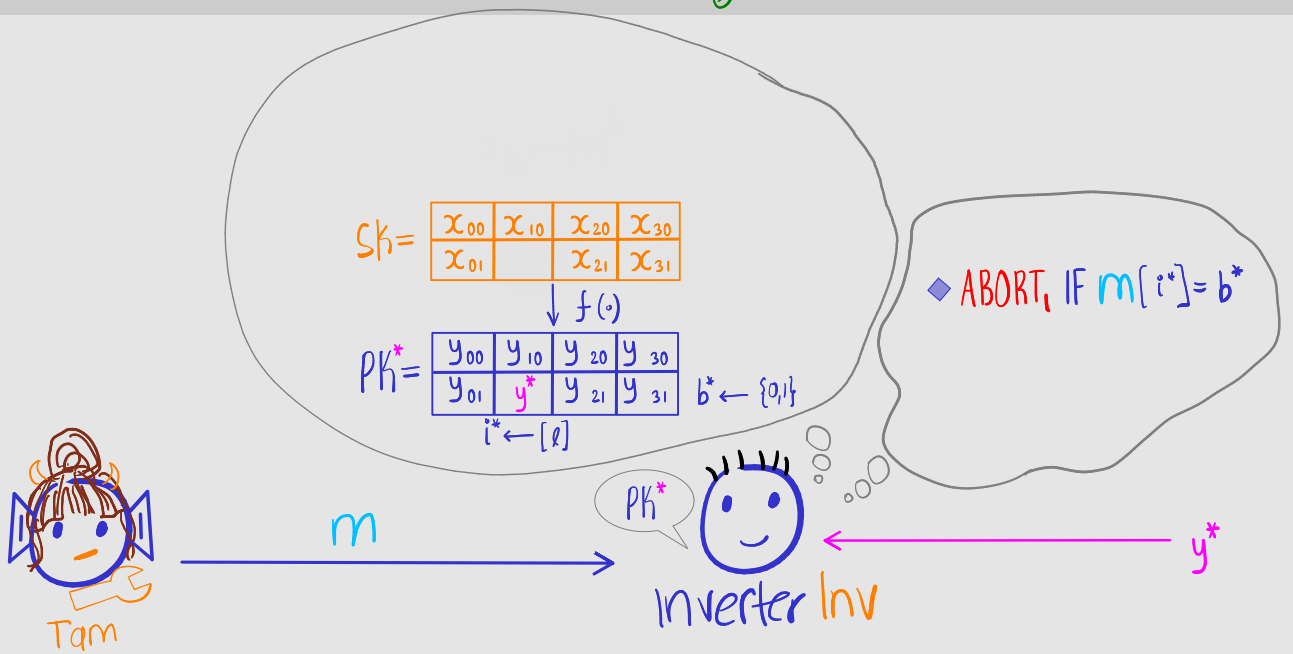
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



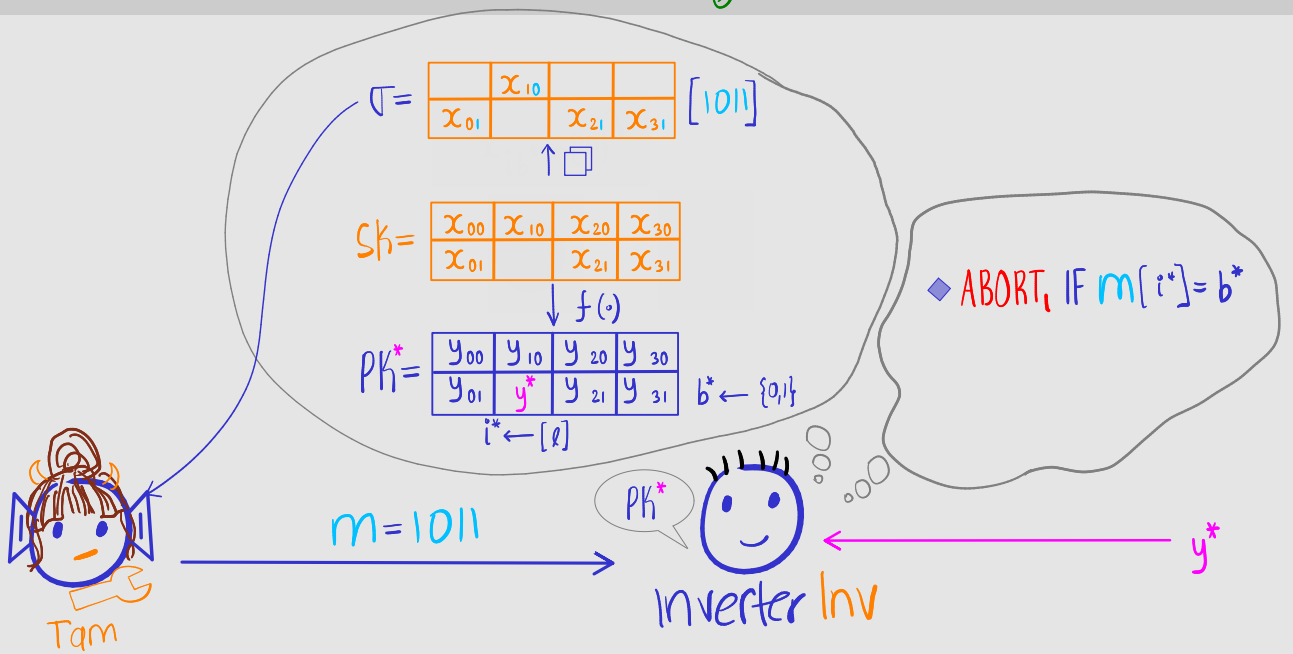
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



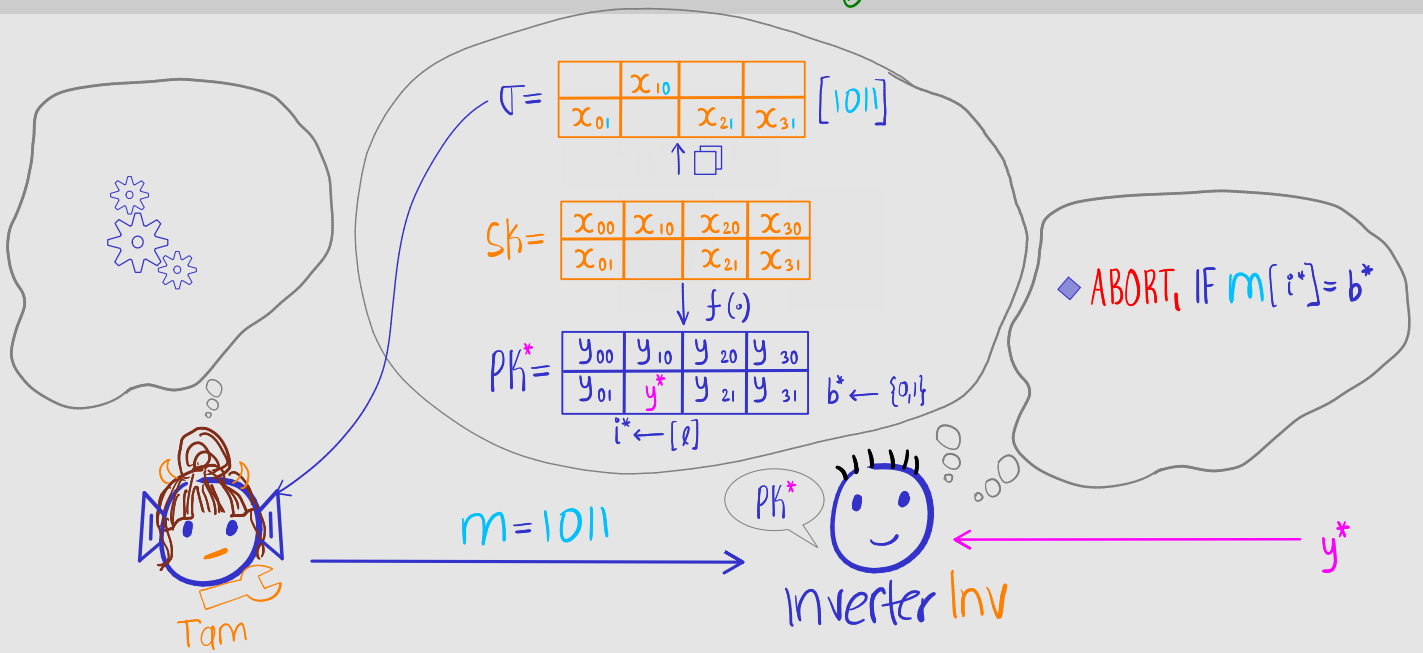
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



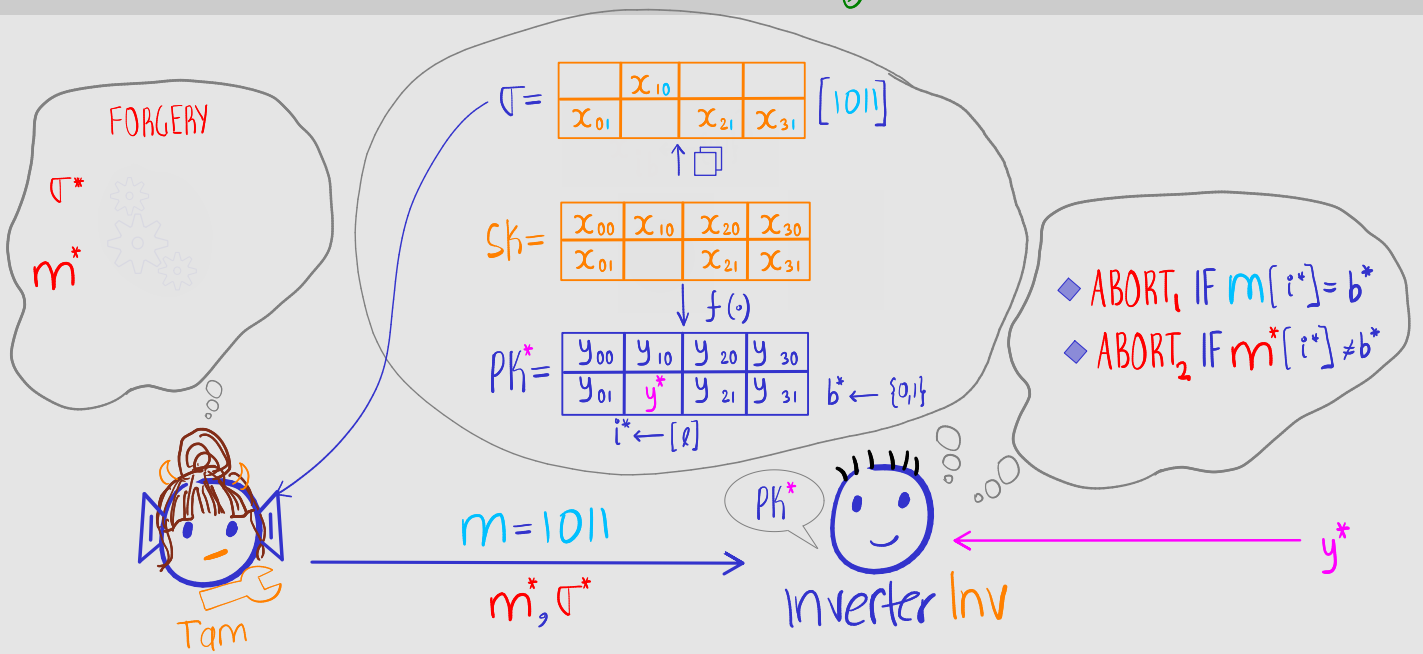
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".

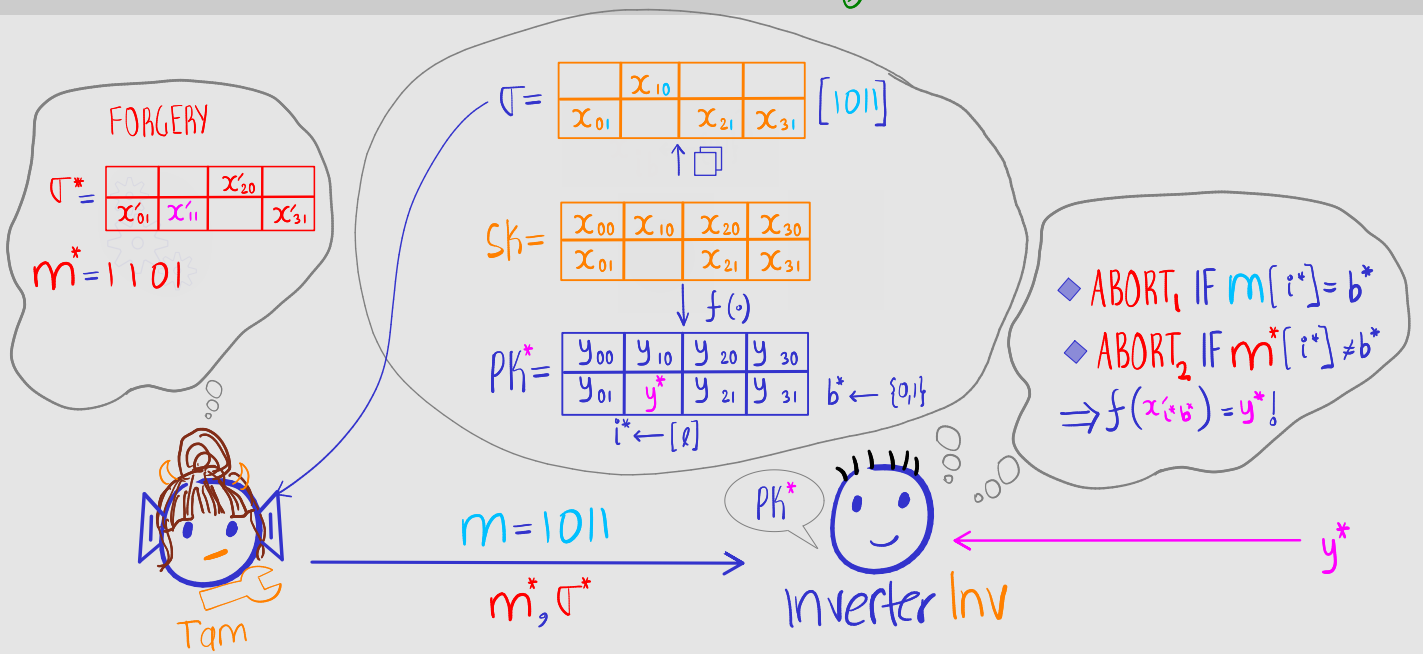


Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



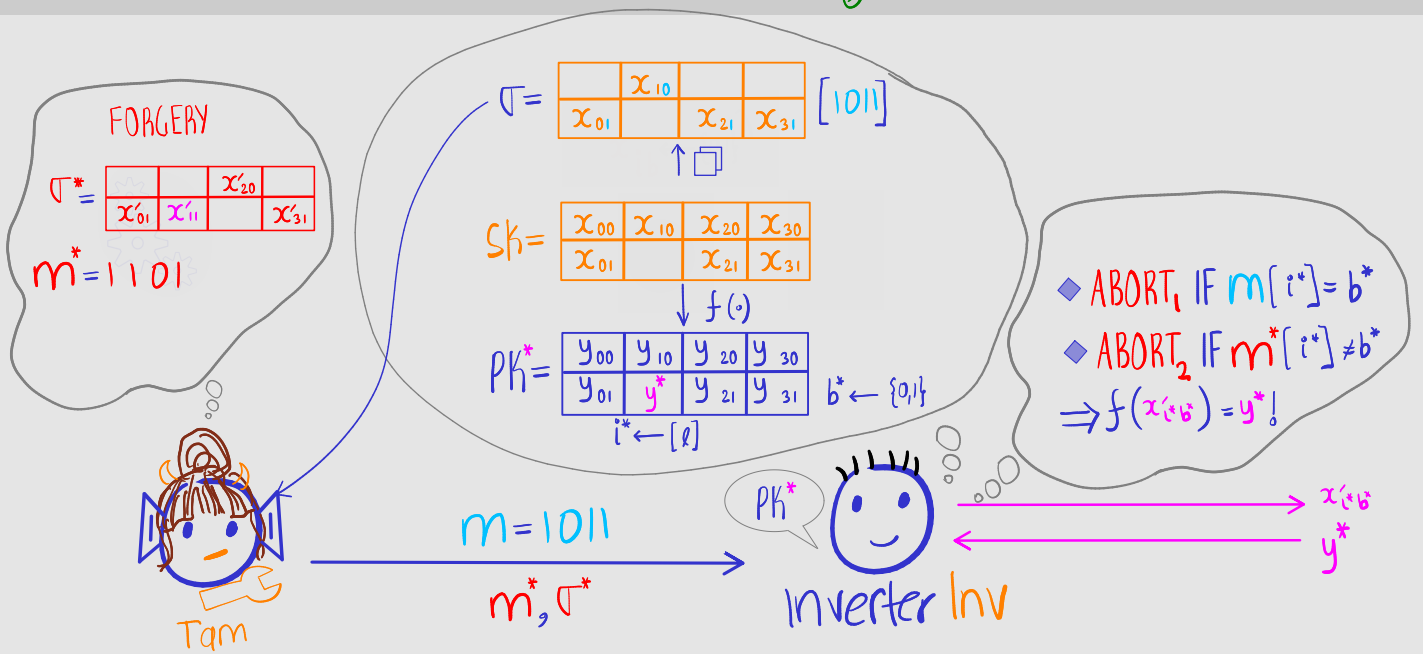
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



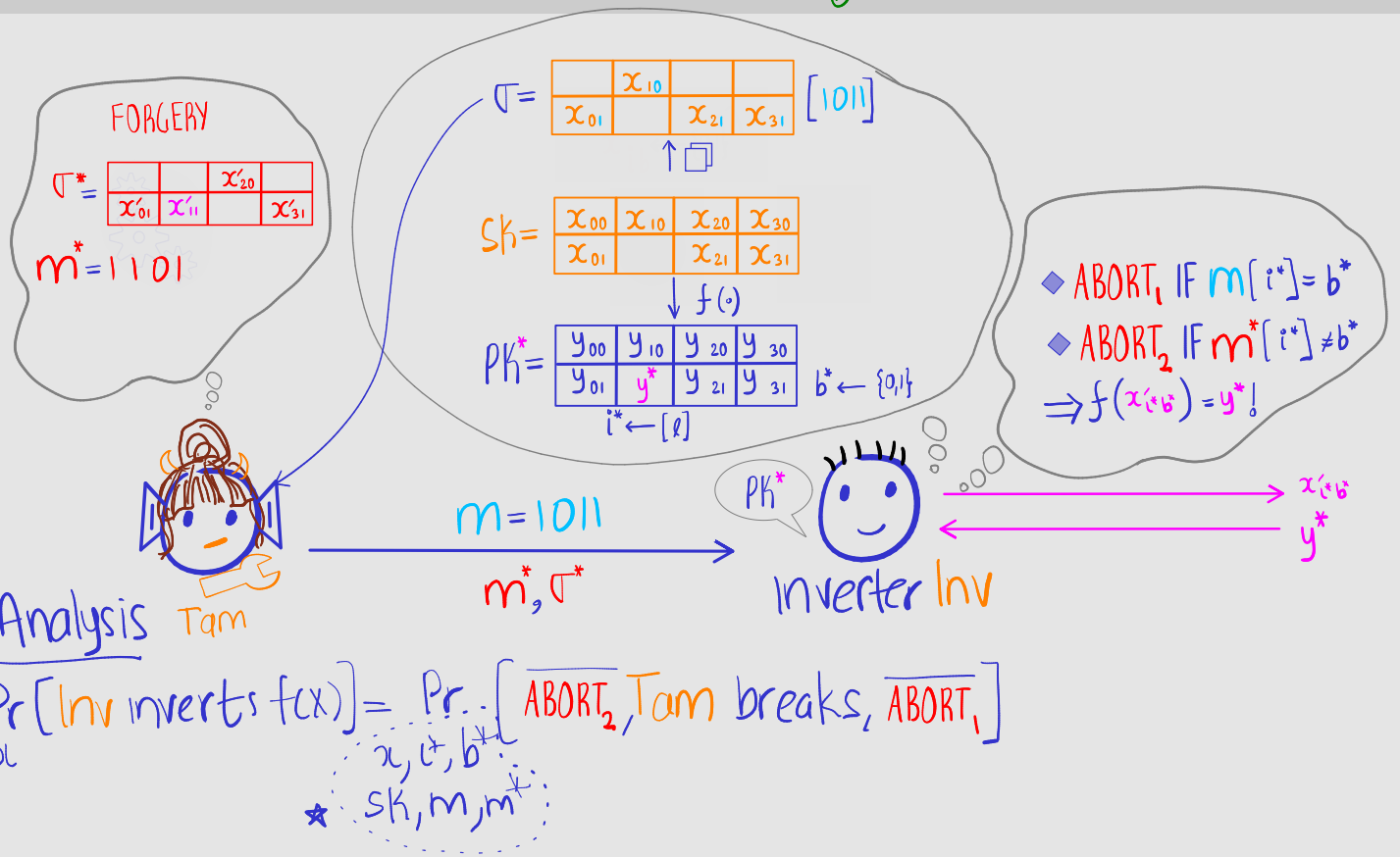
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



Analysis

$\Pr_{x}[\text{Inv inverts } f(x)] = \Pr[\overline{\text{ABORT}_2}, \text{Tam breaks}, \overline{\text{ABORT}_1}]$

x, i^*, b^*

sk, m, m^*

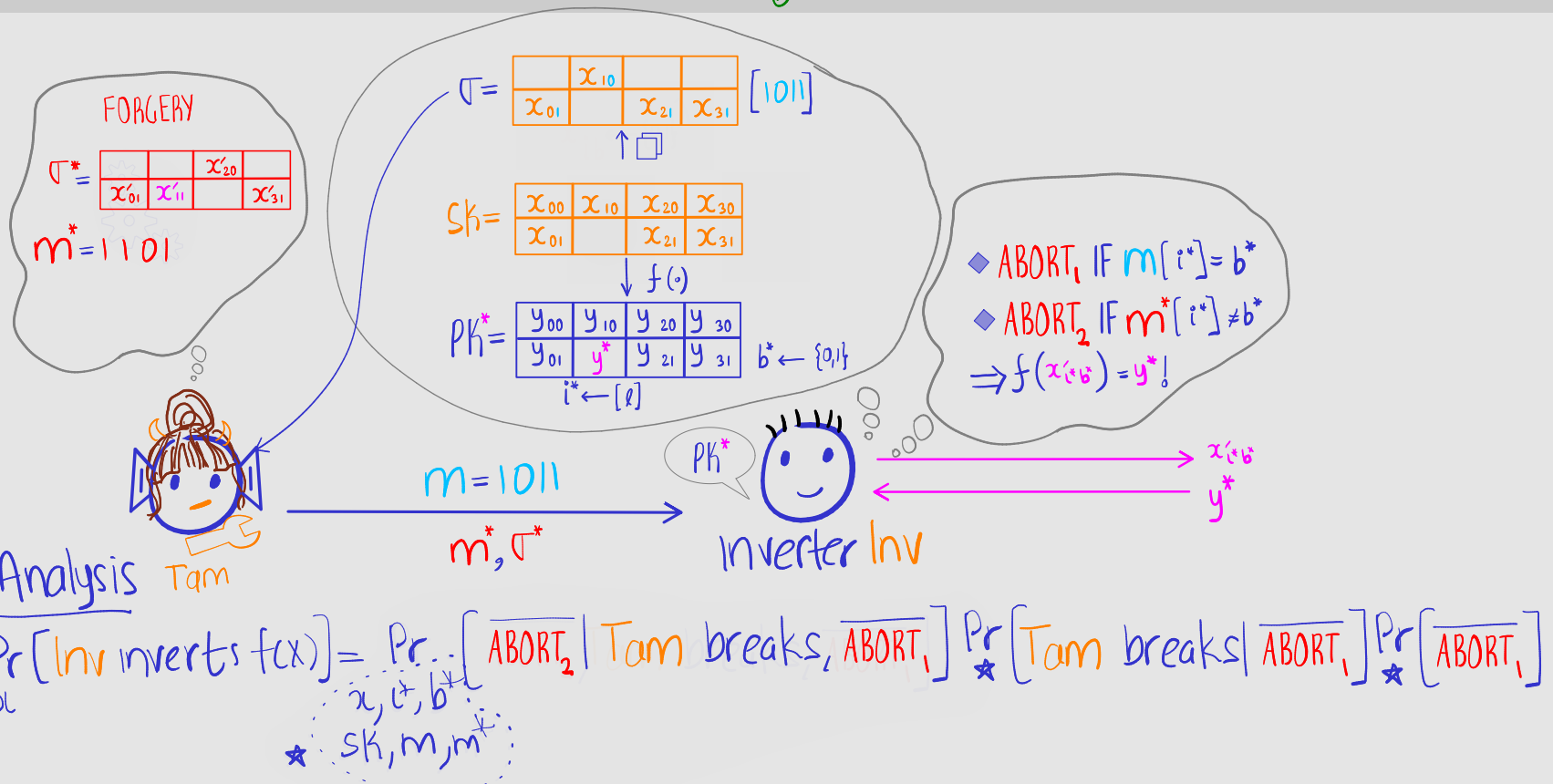
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



$$\Pr_{x}[\text{Inv inverts } f(x)] = \Pr_{\star}[\overline{\text{ABORT}_2} \mid \text{Tam breaks}, \overline{\text{ABORT}_1}] \Pr_{\star}[\text{Tam breaks} \mid \overline{\text{ABORT}_1}] \Pr_{\star}[\overline{\text{ABORT}_1}]$$

$\star$: x, i^*, b^*, sk, m, m^*

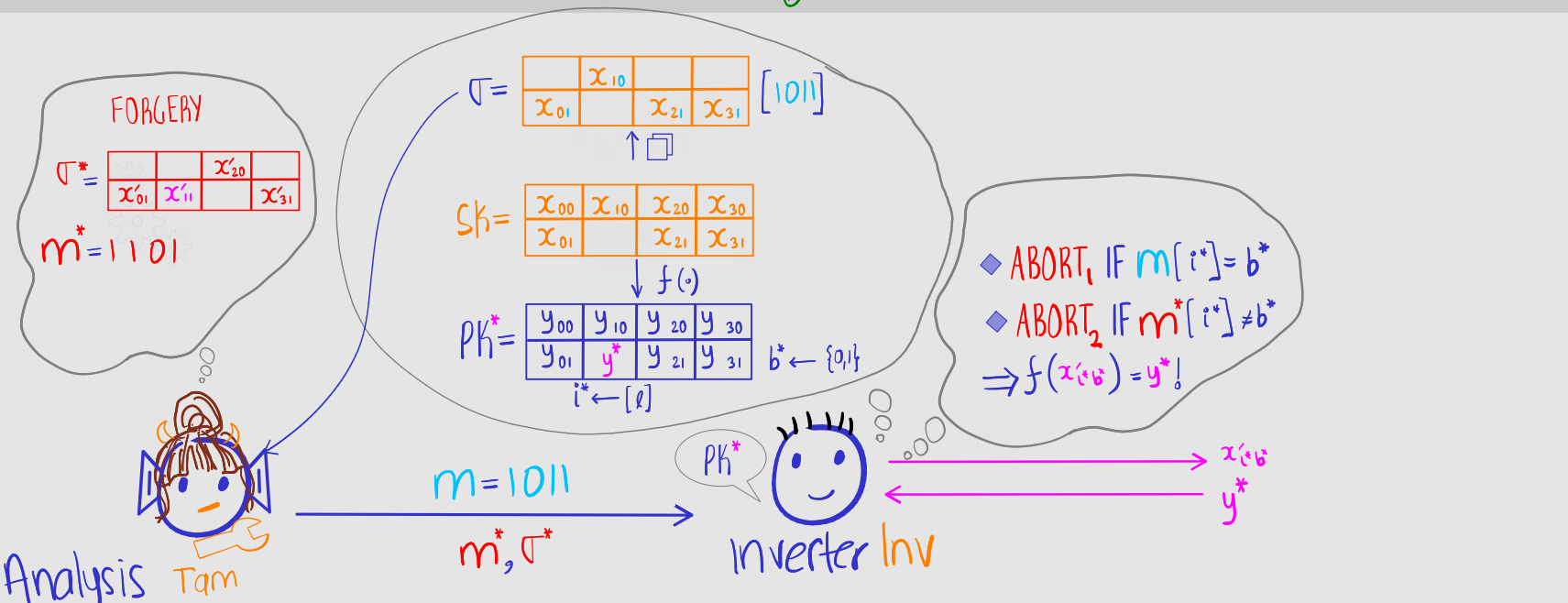
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



$$\Pr_{x}[\text{Inv inverts } f(x)] = \Pr_{\star}[\overline{\text{ABORT}_2} \mid \text{Tam breaks}, \overline{\text{ABORT}_1}] \Pr_{\star}[\text{Tam breaks} \mid \overline{\text{ABORT}_1}] \Pr_{\star}[\overline{\text{ABORT}_1}]$$

$\star$: x, i^*, b^*, sk, m, m^*

independence of b^*: $\frac{1}{2}$

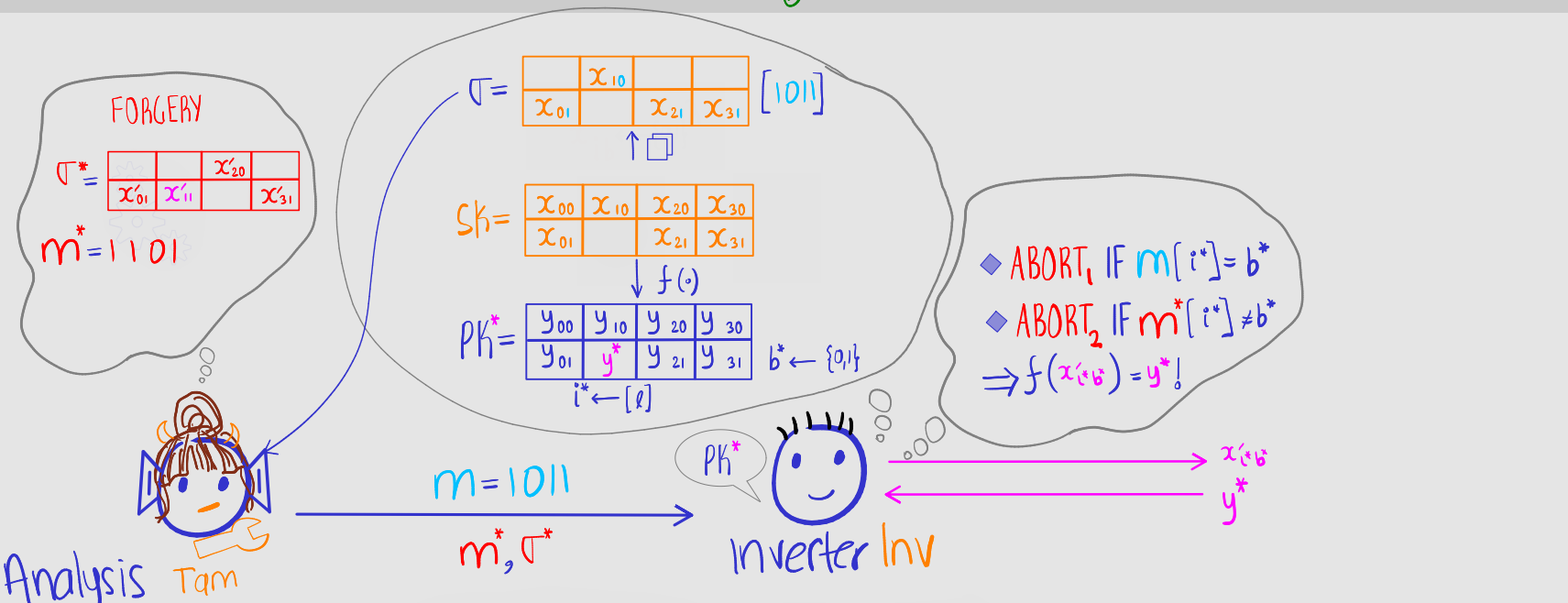
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



Analysis

$$\Pr_{x}[\text{Inv inverts } f(x)] = \Pr_{\star}[\overline{\text{ABORT}_2} \mid \text{Tam breaks}, \overline{\text{ABORT}_1}] \Pr_{\star}[\text{Tam breaks} \mid \overline{\text{ABORT}_1}] \Pr_{\star}[\overline{\text{ABORT}_1}]$$

$\star$: x, i^*, b^*, sk, m, m^*

$\Pr[\text{Tam breaks} \mid \overline{\text{ABORT}_1}] \geq 1/p(n)$ independence of b^*

$\Pr[\overline{\text{ABORT}_1}] = 1/2$

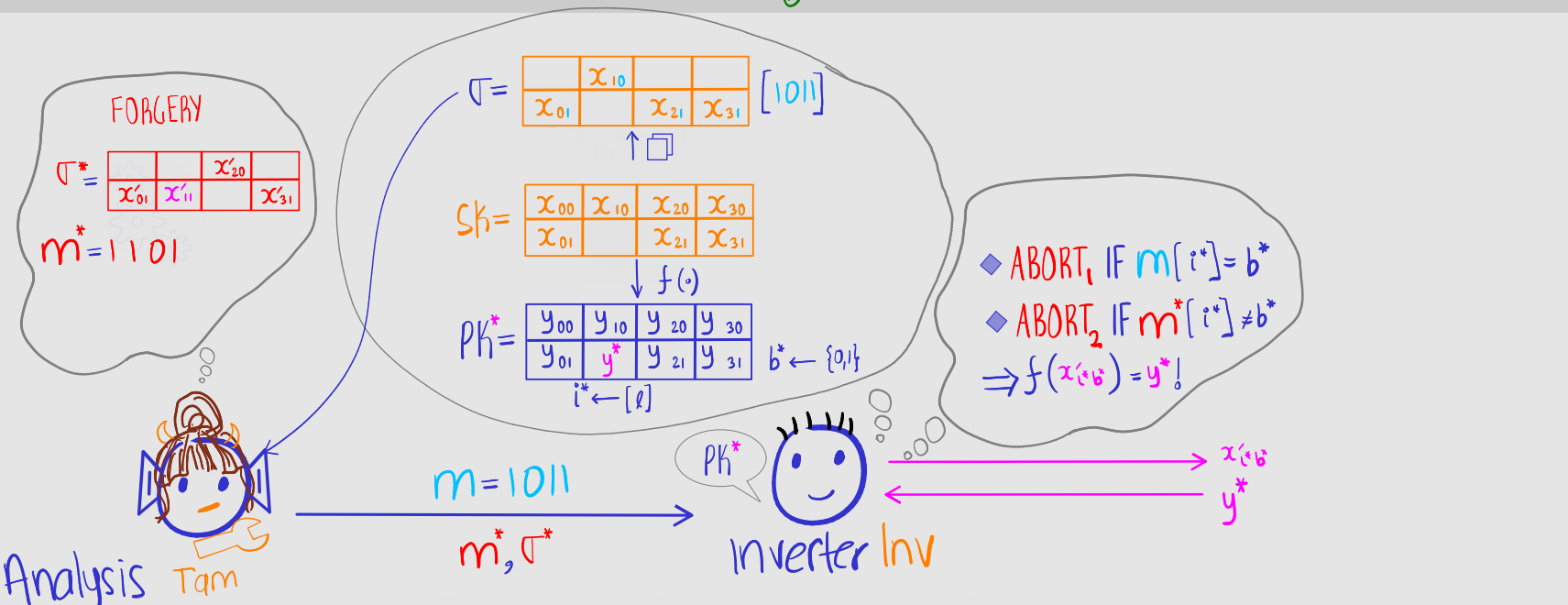
□

Lamport's Signature is One-Time Secure...

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



$$\Pr_{x}[\text{Inv inverts } f(x)] = \Pr_{\star}[\overline{\text{ABORT}_2} \mid \text{Tam breaks}, \overline{\text{ABORT}_1}] \Pr_{\star}[\text{Tam breaks} \mid \overline{\text{ABORT}_1}] \Pr_{\star}[\overline{\text{ABORT}_1}]$$

$\star$: x, i^*, b^*, sk, m, m^*

$\Pr[\overline{\text{ABORT}_2} \mid \text{Tam breaks}, \overline{\text{ABORT}_1}] \leq 1/\ell$ (independence of i^*)

$\Pr[\text{Tam breaks} \mid \overline{\text{ABORT}_1}] \geq 1/p(n)$ (independence of b^*)

$\Pr[\overline{\text{ABORT}_1}] = 1/2$

$\geq 1/p(n) \cdot 2\ell$ ↯

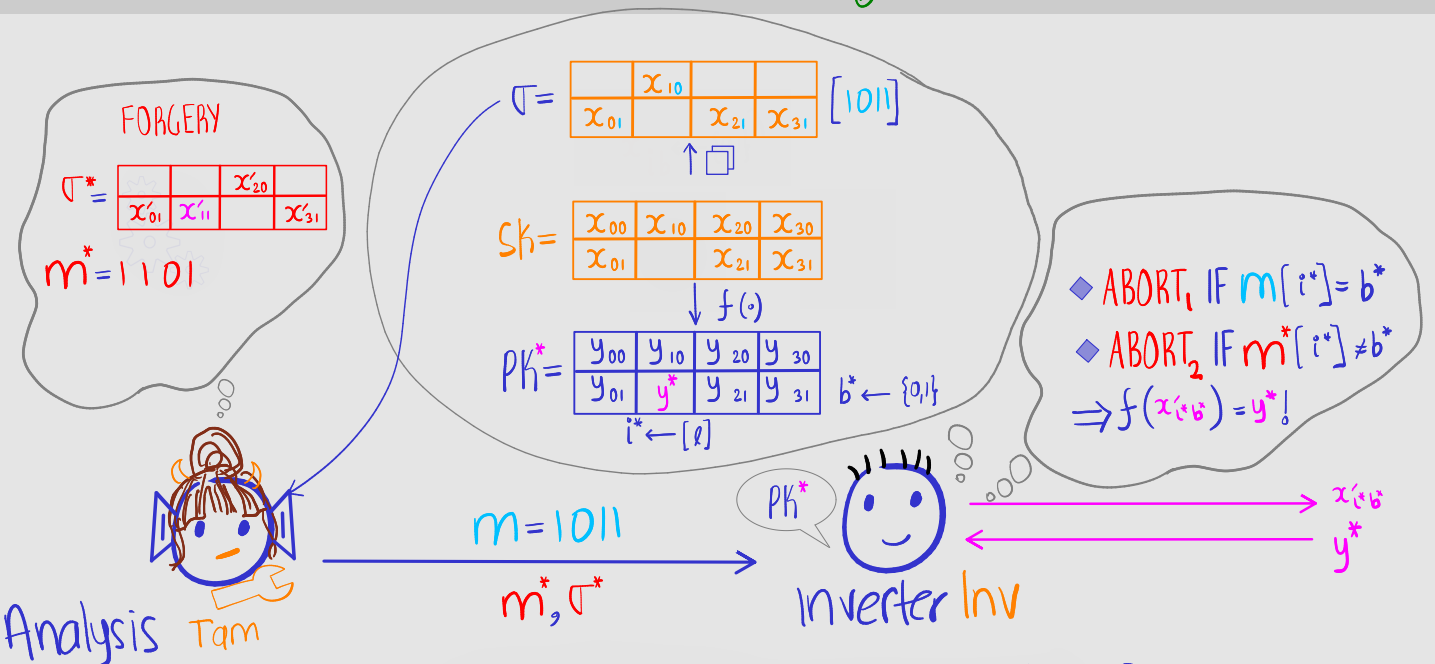
□

Lamport's Signature is One-Time Secure

Theorem 1

If f is a OWF then Lamport's scheme is a one-time DS.

Proof sketch: proof by reduction. Idea: "plug and pray".



Analysis

$\Pr_x[\text{Inv inverts } f(x)] = \Pr_{\star}[\overline{\text{ABORT}_2} \mid \text{Tam breaks}, \overline{\text{ABORT}_1}] \Pr_{\star}[\text{Tam breaks} \mid \overline{\text{ABORT}_1}] \Pr_{\star}[\overline{\text{ABORT}_1}]$

$\star$: x, i^*, b^*, sk, m, m^*

$\overline{\text{ABORT}_2}$: independence of i^* → $1/\ell$

Tam breaks: $\geq 1/p(n)$

$\overline{\text{ABORT}_1}$: independence of b^* → $1/2$

$\geq 1/p(n) \cdot 2\ell$ ↯

□

Lamport's Signature is One-Time Secure...

Exercise 2

- *Can a forger break EU-CMA given two signatures?*
- *Are the signatures unique? If not, can it be made unique?*

Lamport's Signature is One-Time Secure...

Exercise 2

- *Can a forger break EU-CMA given two signatures?*
- *Are the signatures unique? If not, can it be made unique?*
- *Can we avoid the* $1/2\ell$ *loss in inverting advantage?*

Theorem 2

If f *is a OWF then Lamport's scheme is a one-time DS*

Lamport's Signature is One-Time Secure...

Exercise 2

- *Can a forger break EU-CMA given two signatures?*
- *Are the signatures unique? If not, can it be made unique?*
- *Can we avoid the* $1/2\ell$ *loss in inverting advantage?*

Theorem 2

If f *is a OWF then Lamport's scheme is a one-time DS for fixed-length messages.*

Exercise 3 (Domain Extension)

Given a compressing function $\mathsf{H} : \{0,1\}^{2\ell} \to \{0,1\}^{\ell}$*, construct a one-time DS for arbitrary-length messages. What are the properties you need from* H *to ensure that the one-time DS is secure?*

Plan for Today's Lecture

1 Digital Signature (DS)

2 One-Time Digital Signatures ⟵ OWF

3 Many-Time (Stateful) Digital Signatures

(Many-Time) DS with Stateful Signer

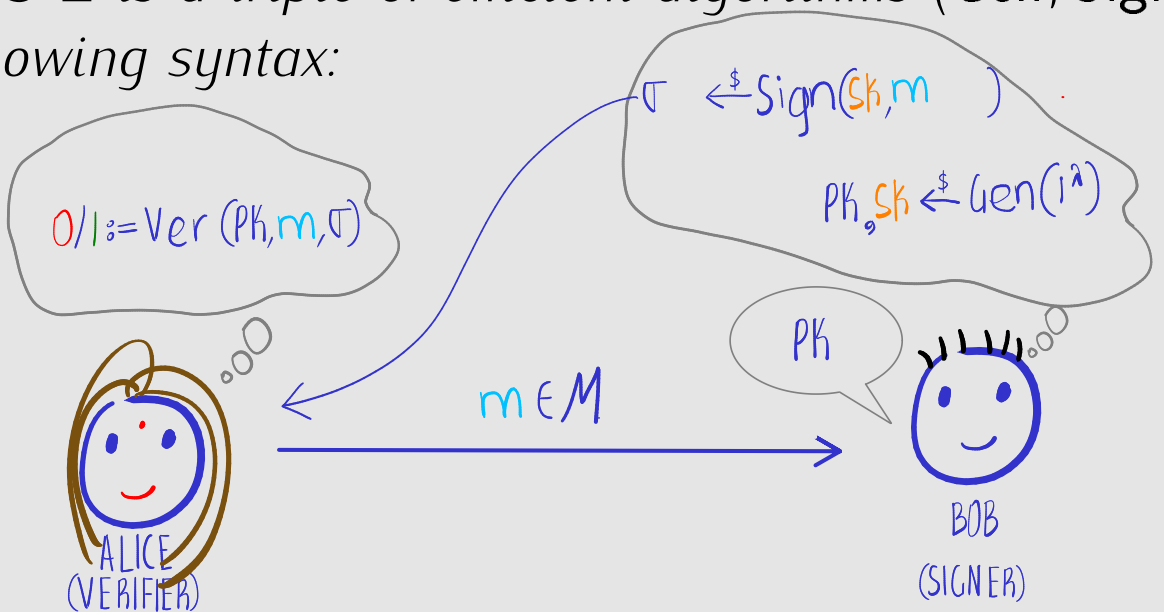
- Syntax: same as before except that **Sign** is *stateful*

(Many-Time) DS with Stateful Signer

- Syntax: same as before except that **Sign** is *stateful*

Defintion 3 (Stateful DS)

A stateful DS Σ is a triple of efficient algorithms (**Gen**, **Sign**, **Ver**) *with the following syntax:*

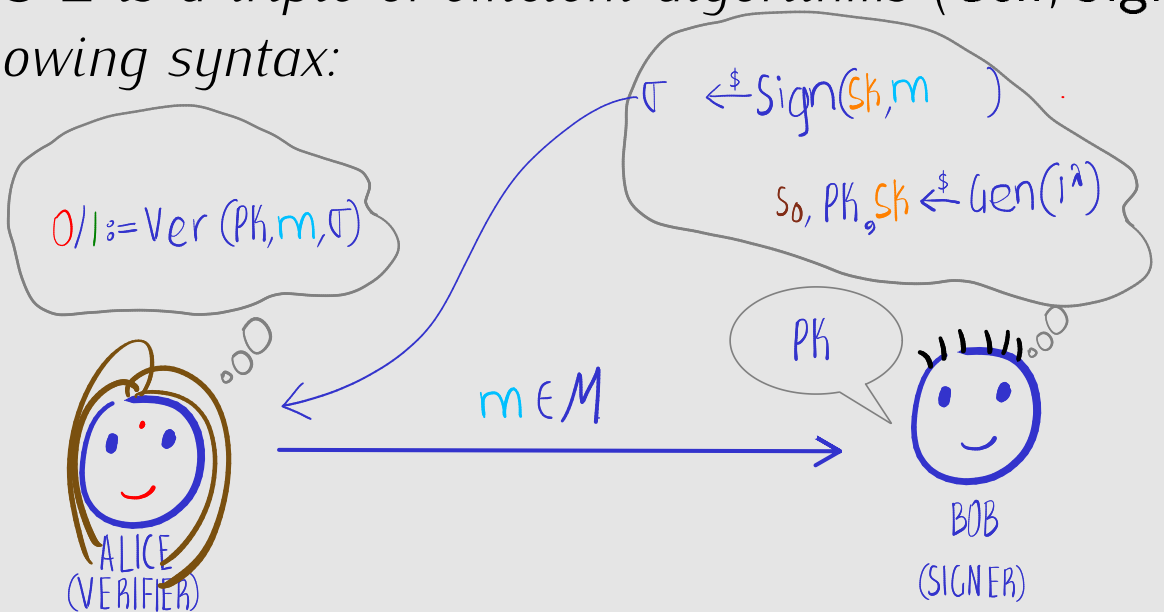


(Many-Time) DS with Stateful Signer

- Syntax: same as before except that **Sign** is *stateful*

Defintion 3 (Stateful DS)

A stateful DS Σ is a triple of efficient algorithms (**Gen**, **Sign**, **Ver**) *with the following syntax:*

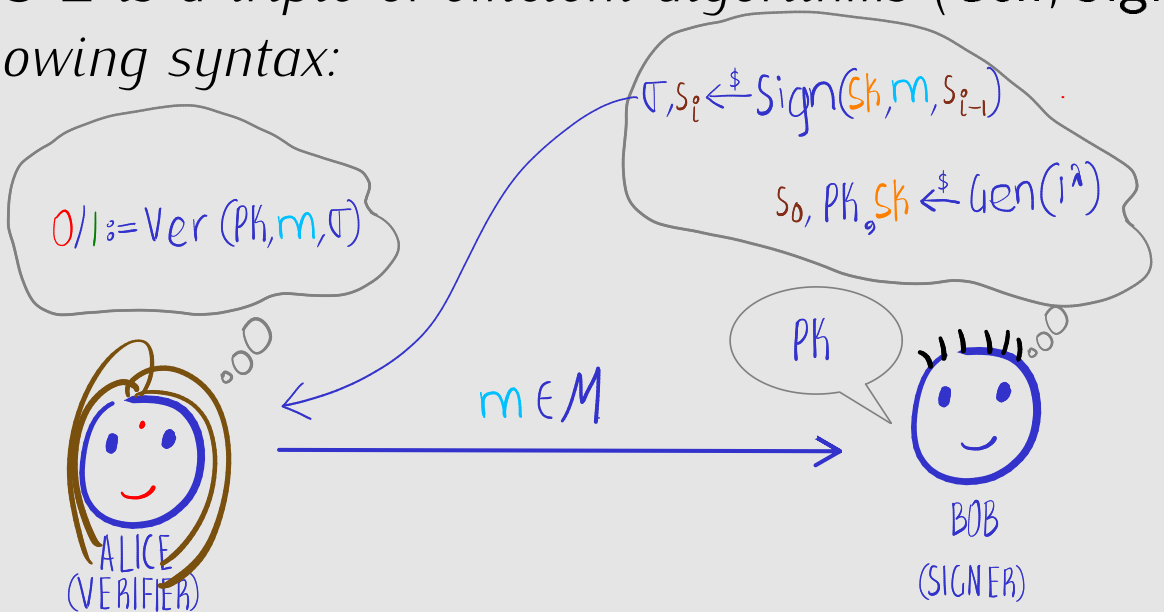


(Many-Time) DS with Stateful Signer

- Syntax: same as before except that **Sign** is *stateful*

Defintion 3 (Stateful DS)

A stateful DS Σ is a triple of efficient algorithms (**Gen**, **Sign**, **Ver**) *with the following syntax:*

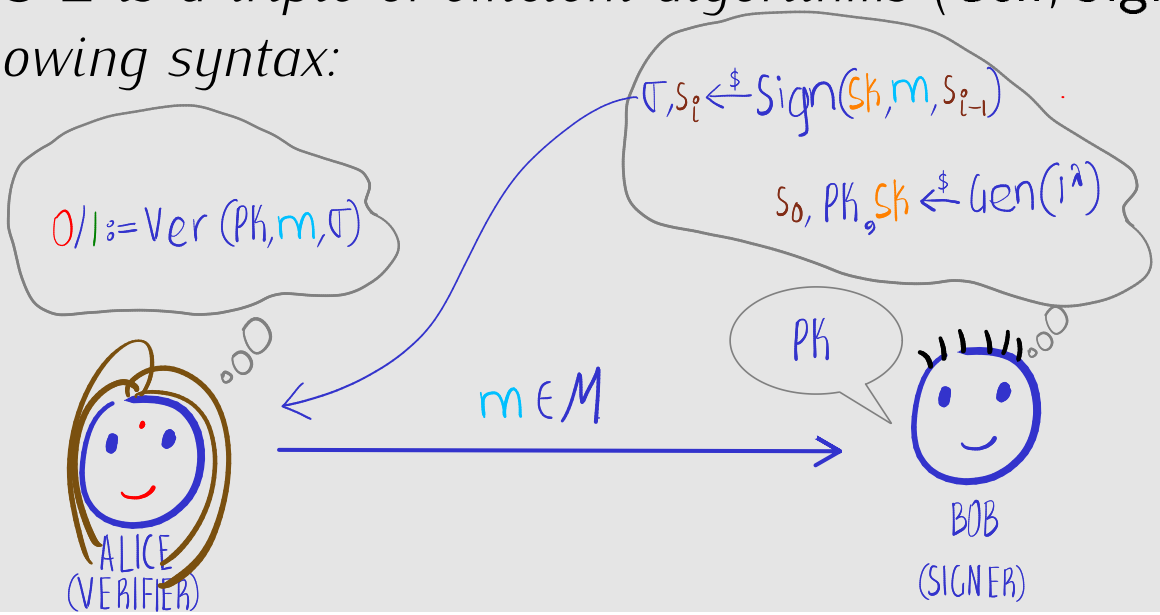


(Many-Time) DS with Stateful Signer

- Syntax: same as before except that **Sign** is *stateful*

Defintion 3 (Stateful DS)

A stateful DS Σ is a triple of efficient algorithms (**Gen**, **Sign**, **Ver**) *with the following syntax:*



Exercise 4

1. *Write down the requirement for correctness of honest signing*
2. *What is different in the security definition EU-CMA?*

(Many-Time) Stateful Digital Signature...

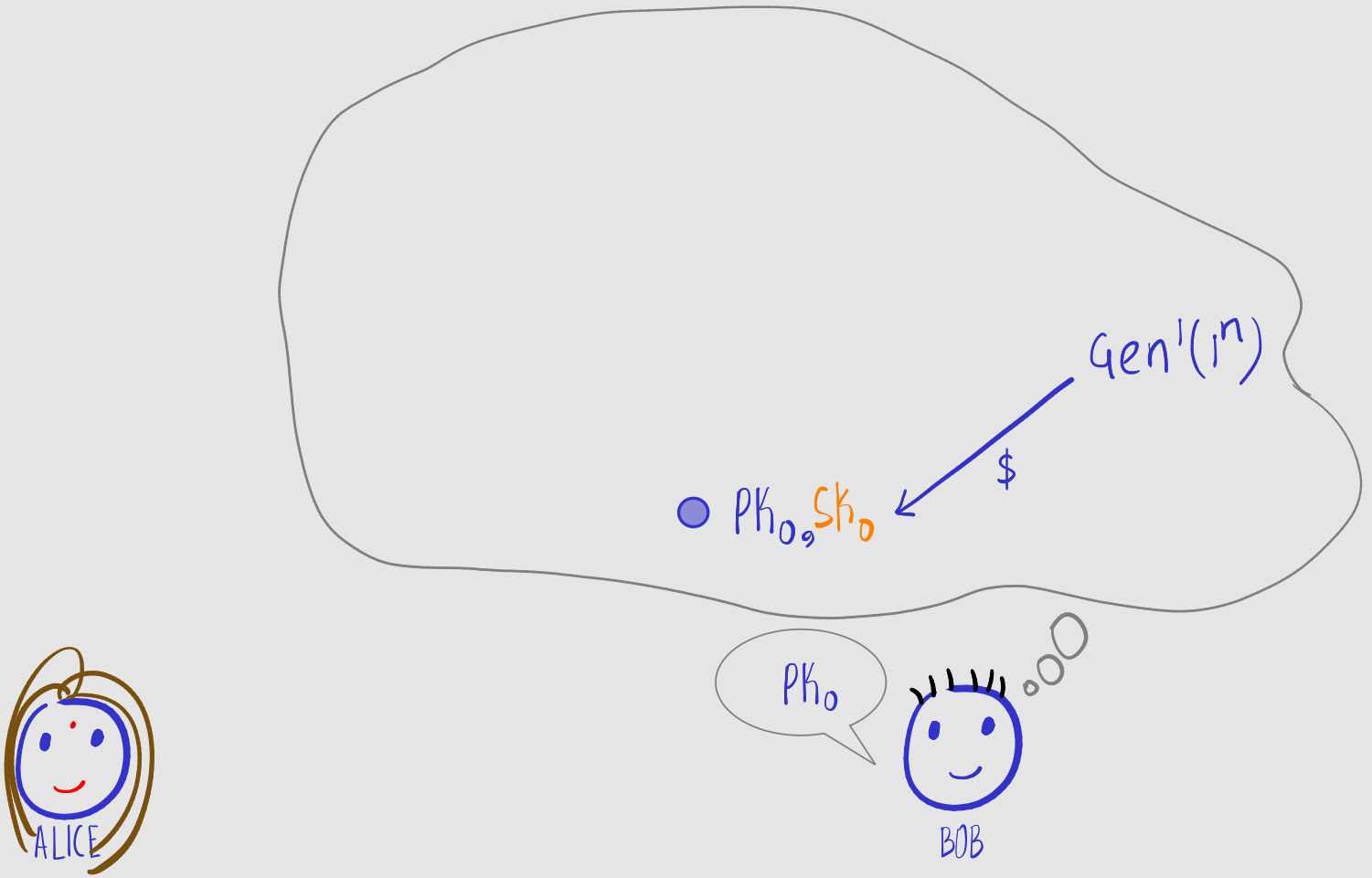
Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s.

(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s.  Idea: "chain signatures")

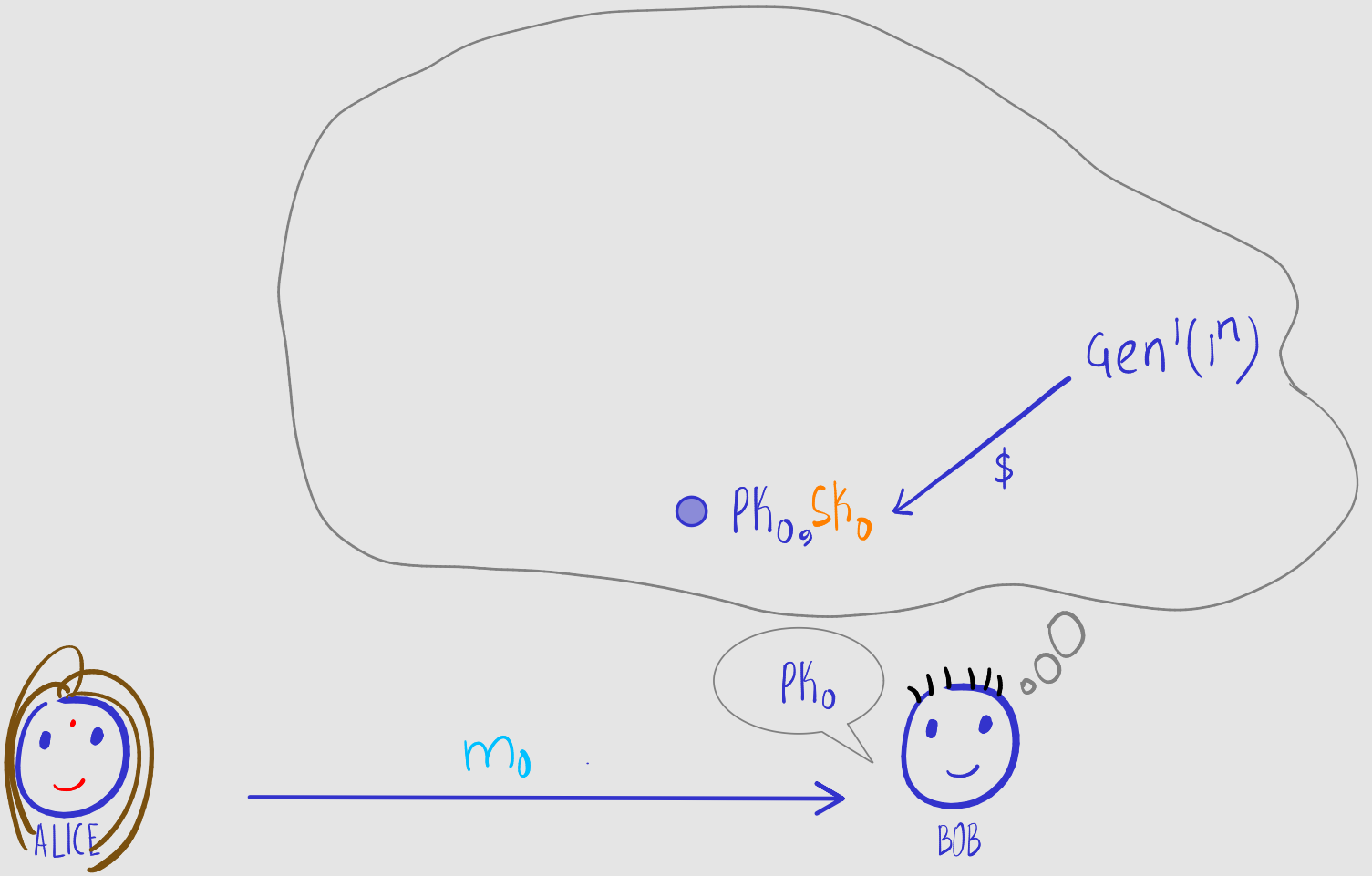
(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s. Idea: "chain signatures")



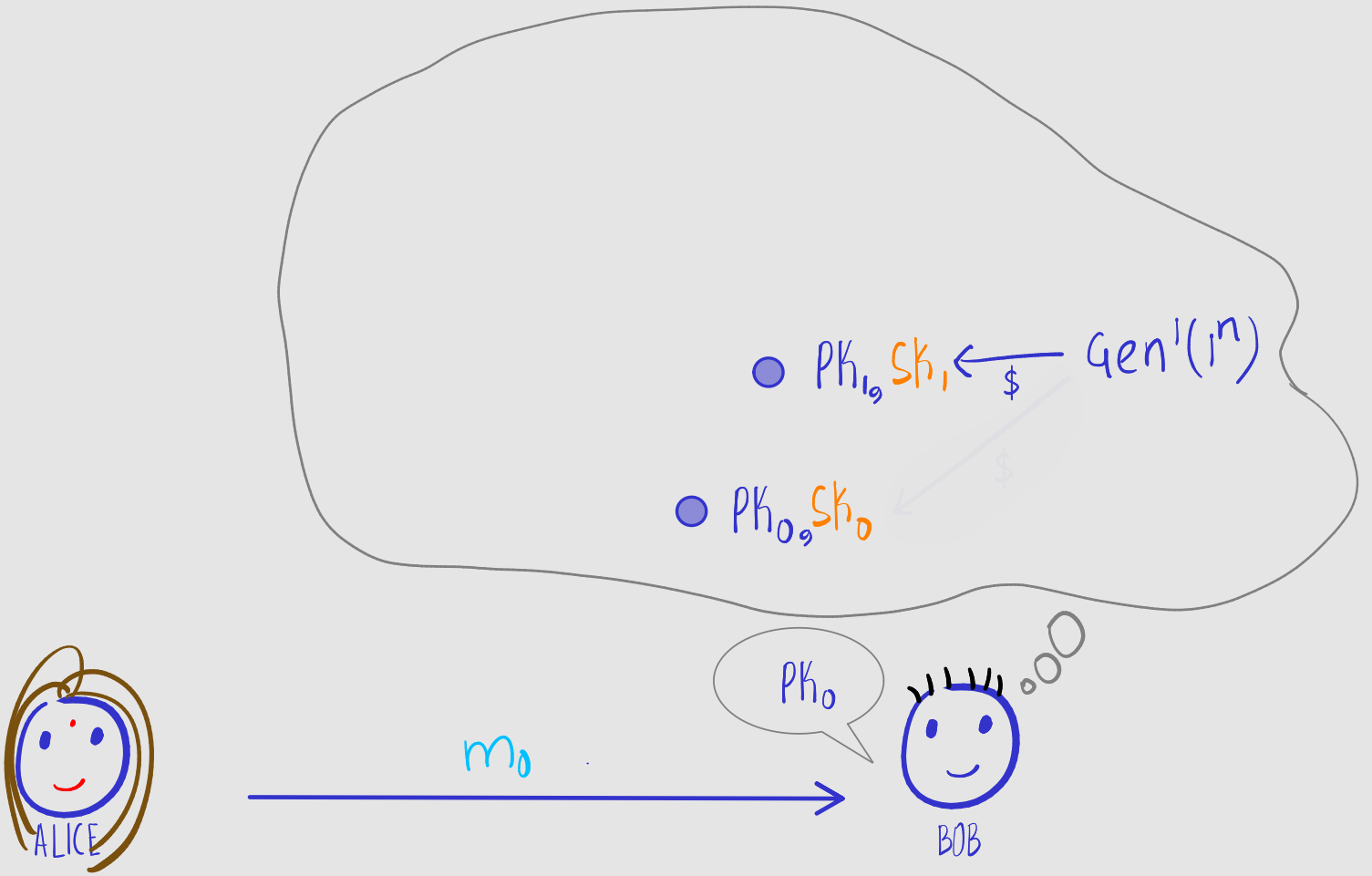
(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s. Idea: "chain signatures")



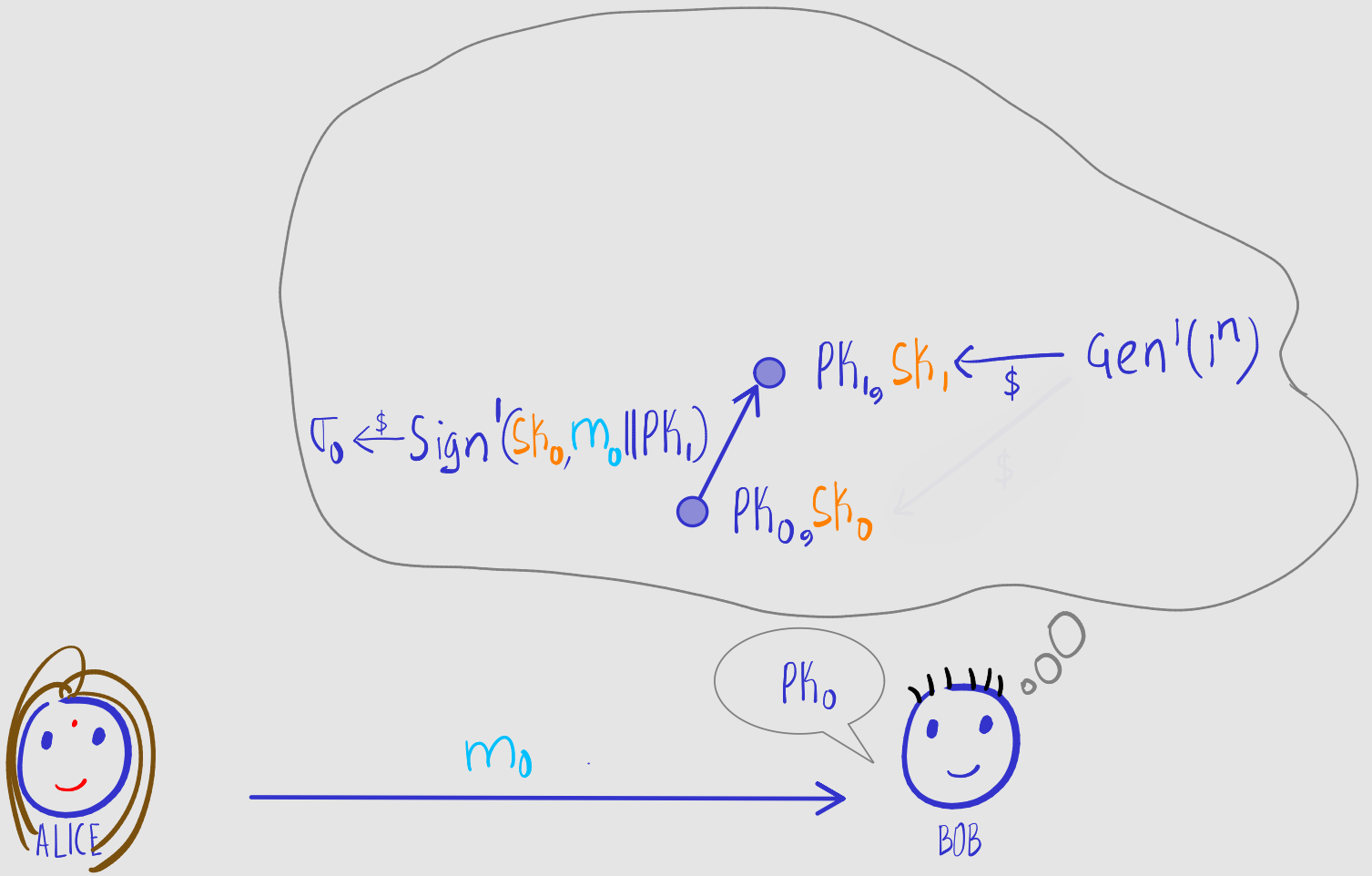
(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s. Idea: "chain signatures")



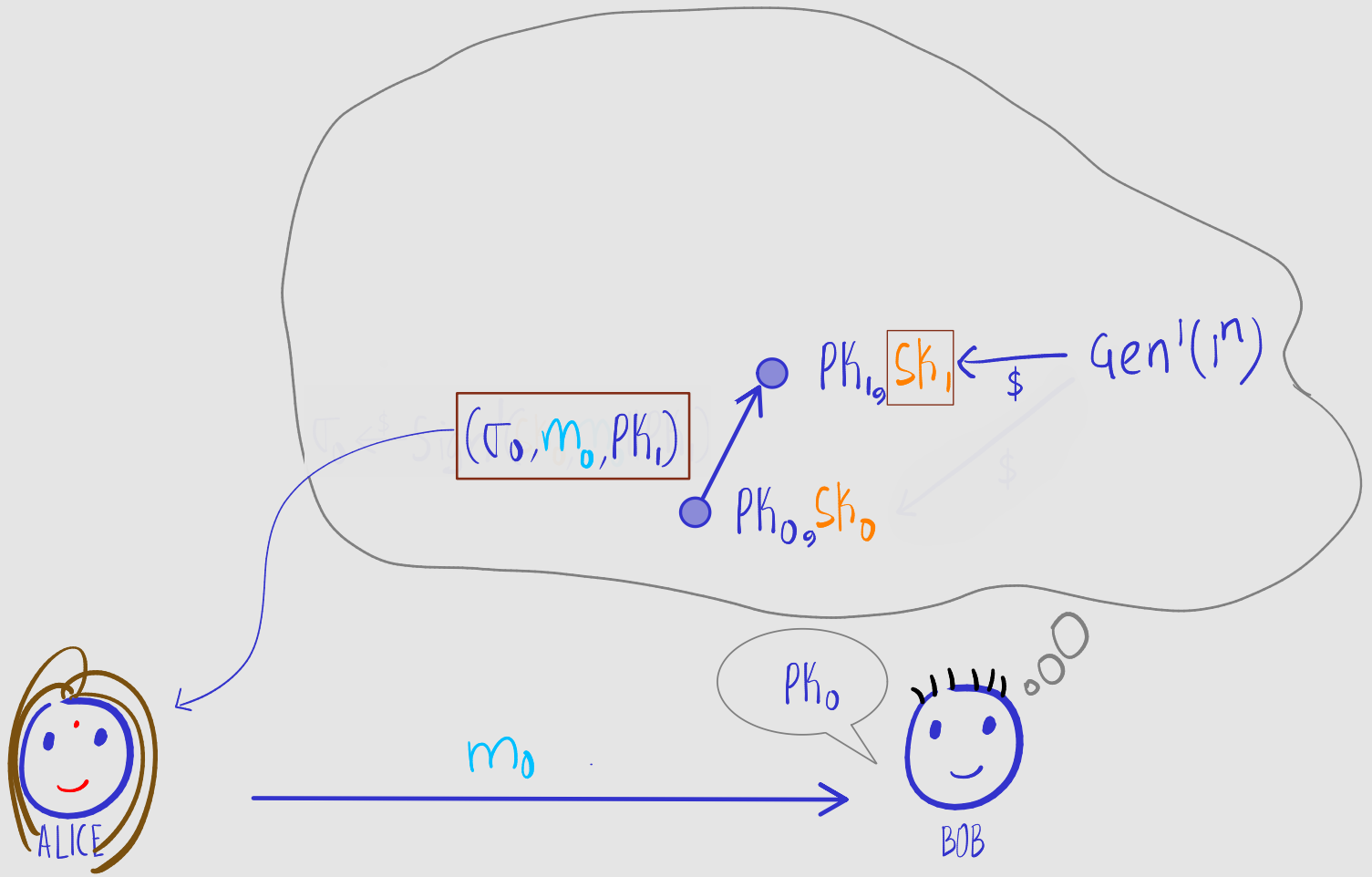
(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s. Idea: "chain signatures")



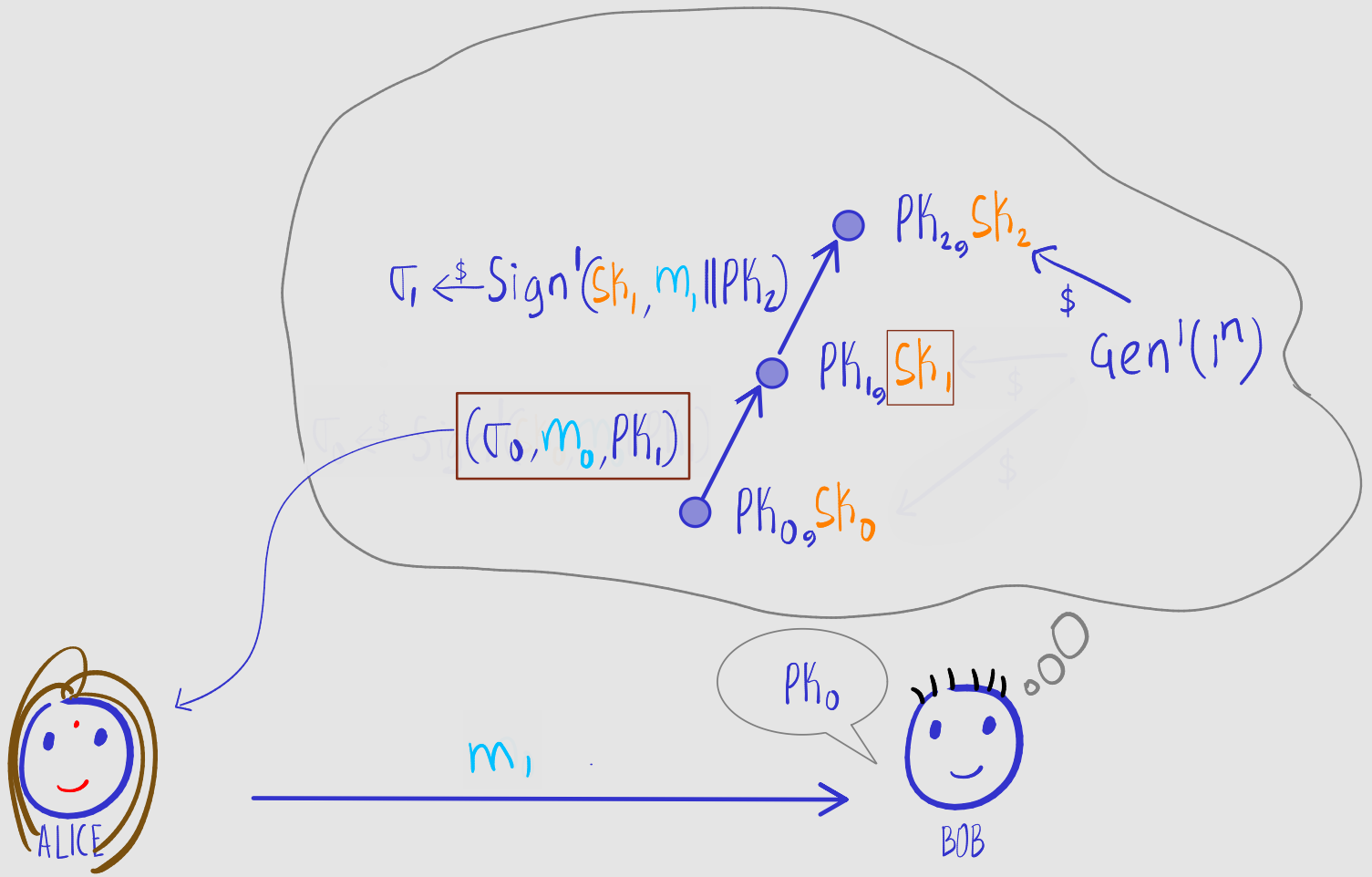
(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s. Idea: "chain signatures")



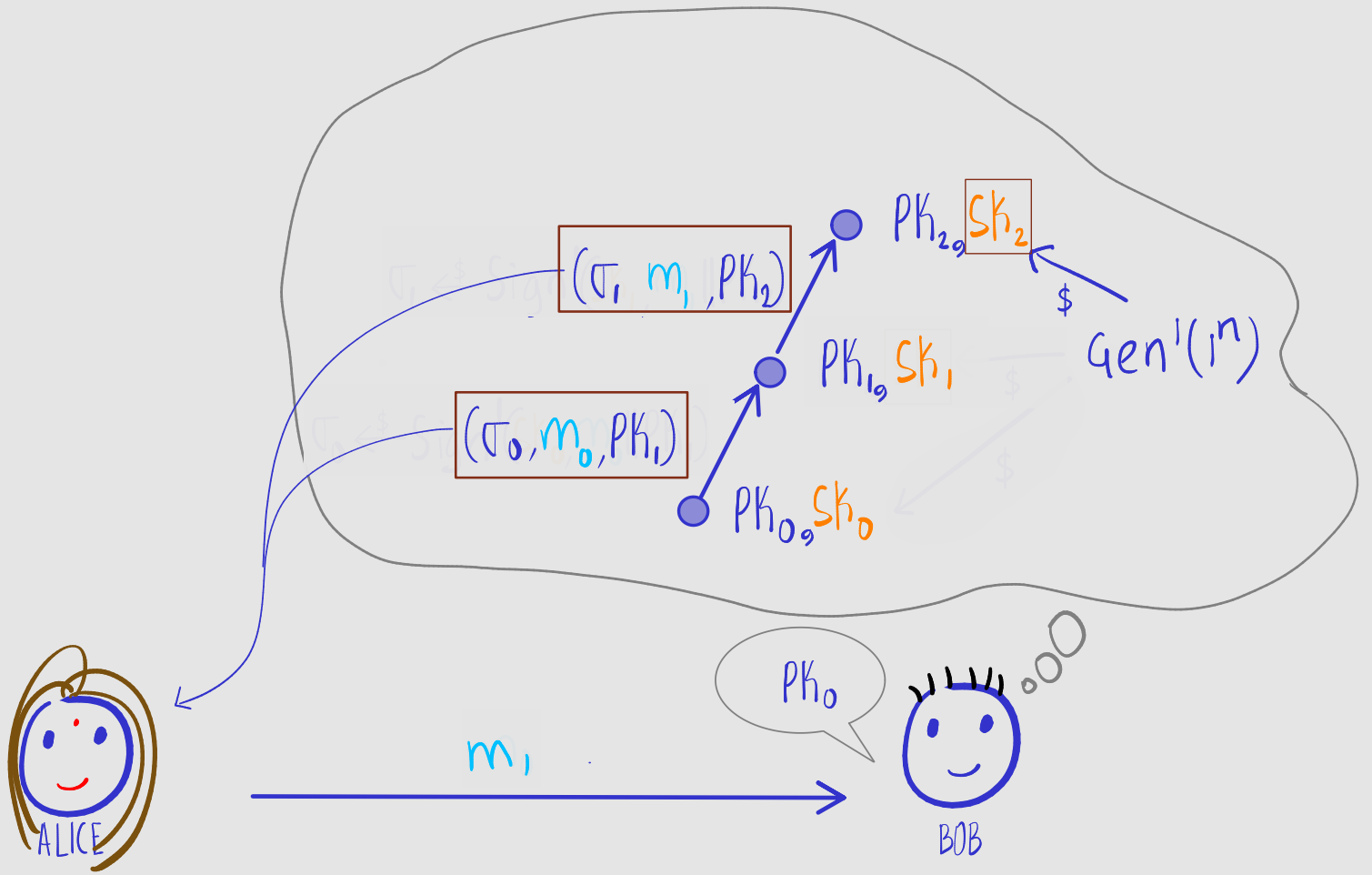
(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s. Idea: "chain signatures")



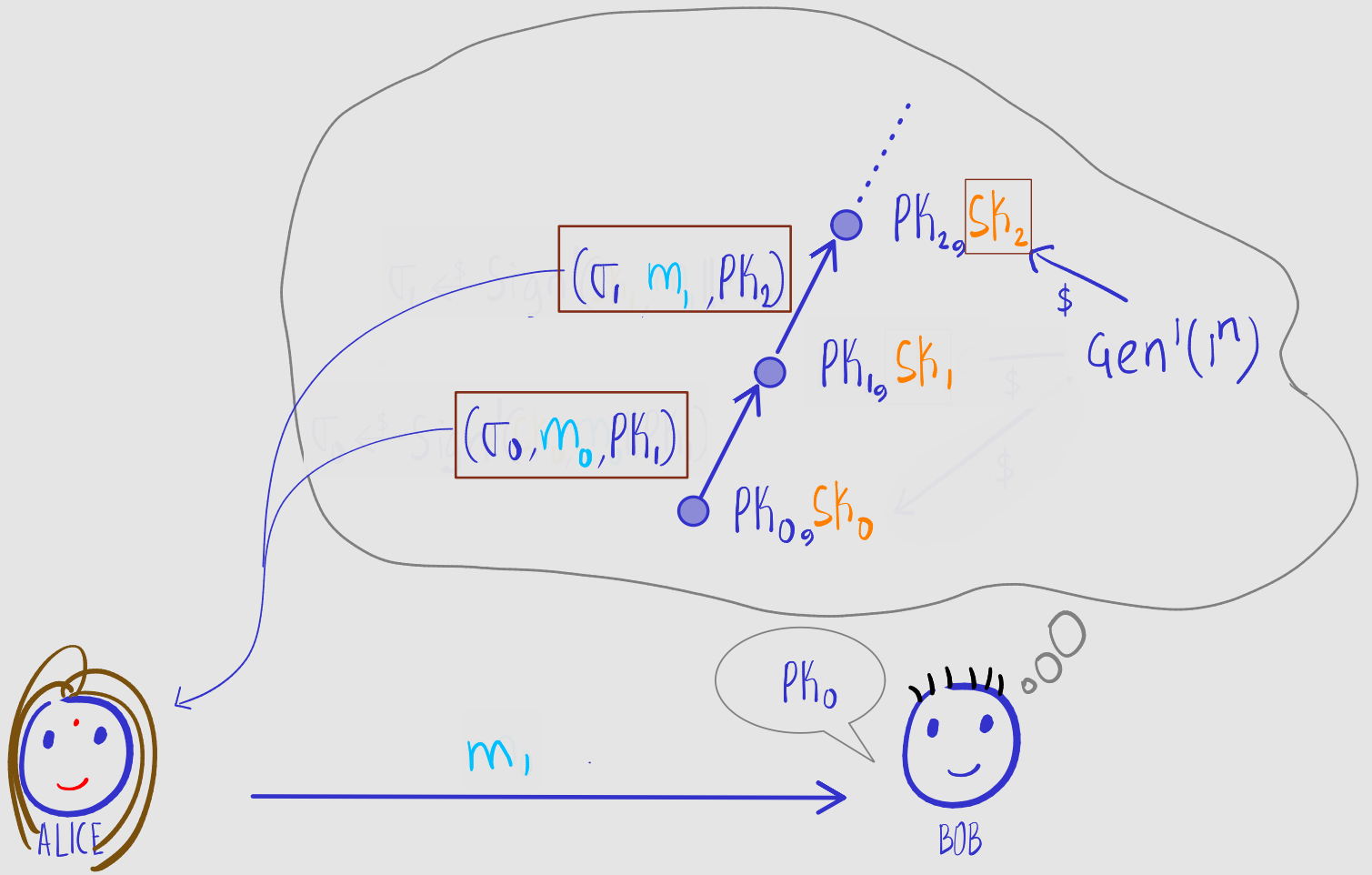
(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s. Idea: "chain signatures")



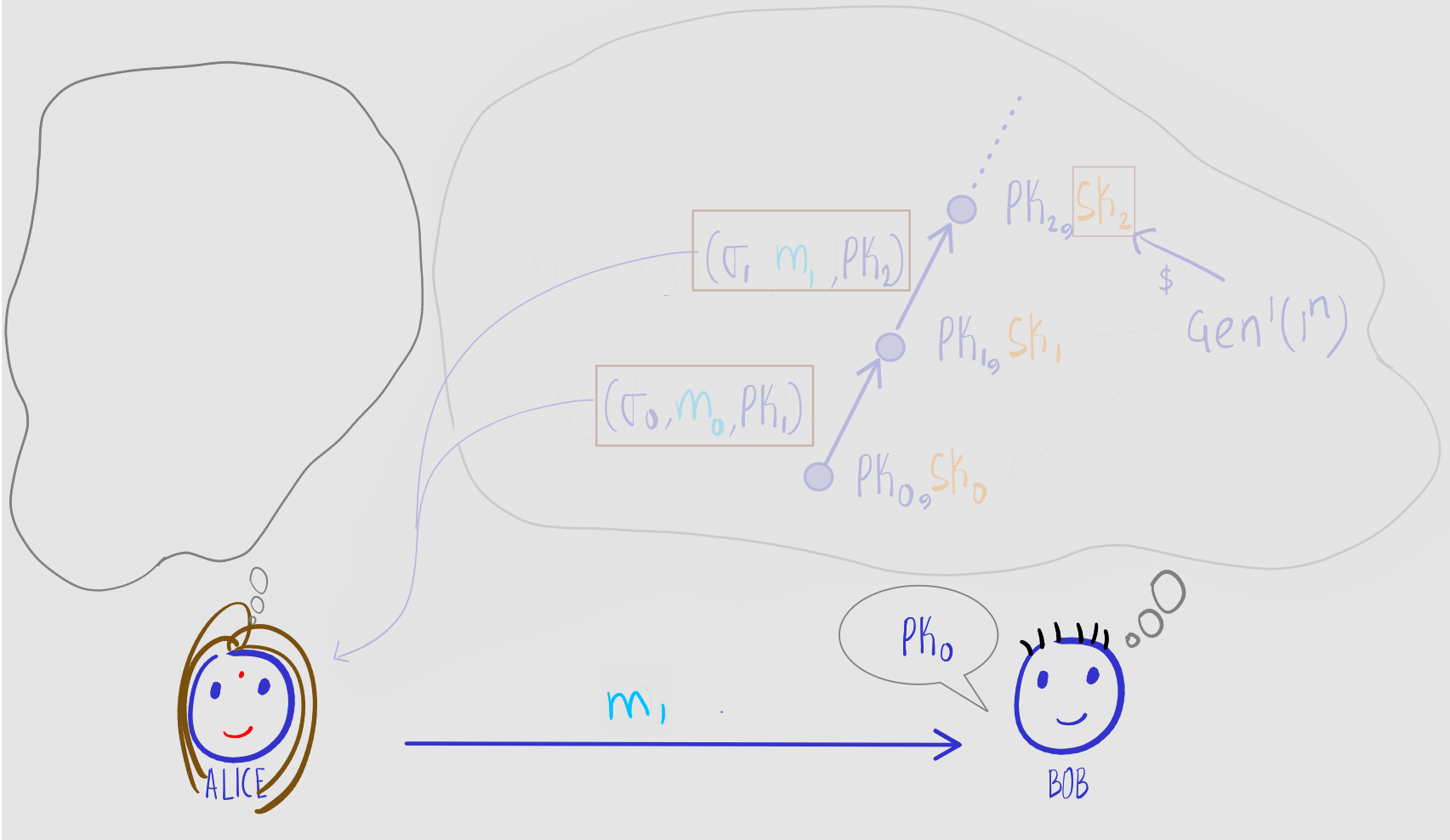
(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s. Idea: "chain signatures")



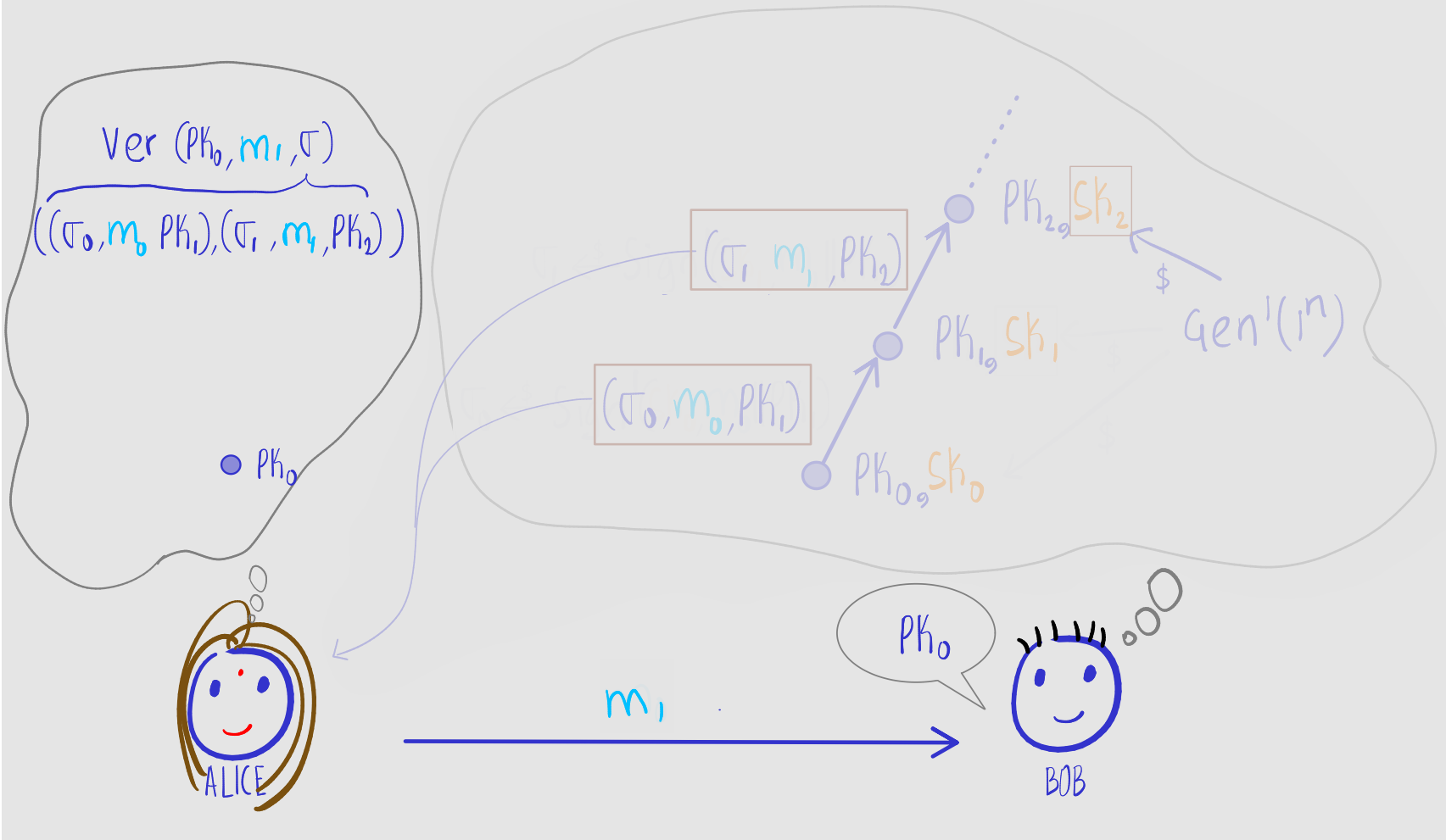
(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s. Idea: "chain signatures")



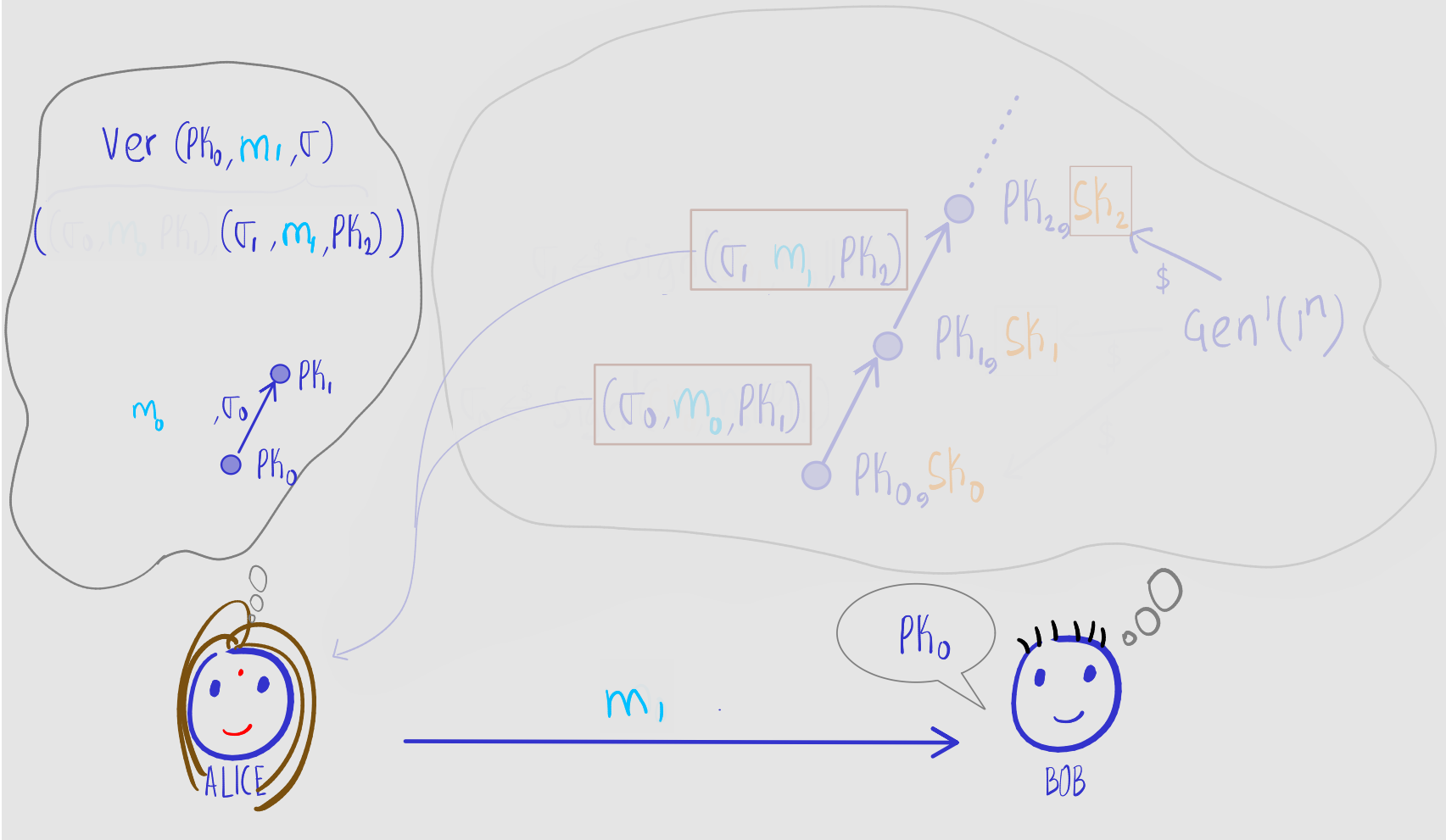
(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s. Idea: "chain signatures")



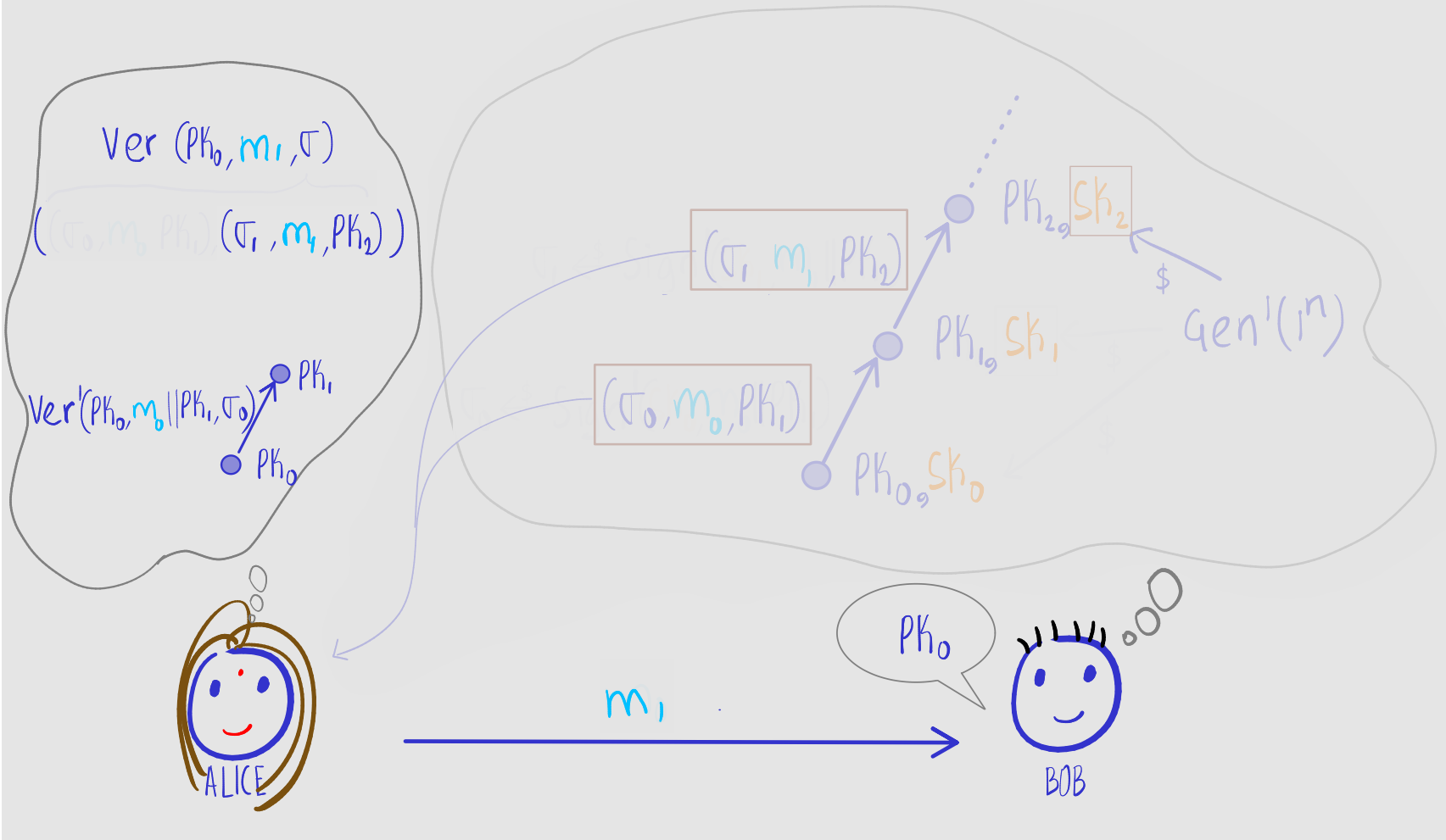
(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s. Idea: "chain signatures")



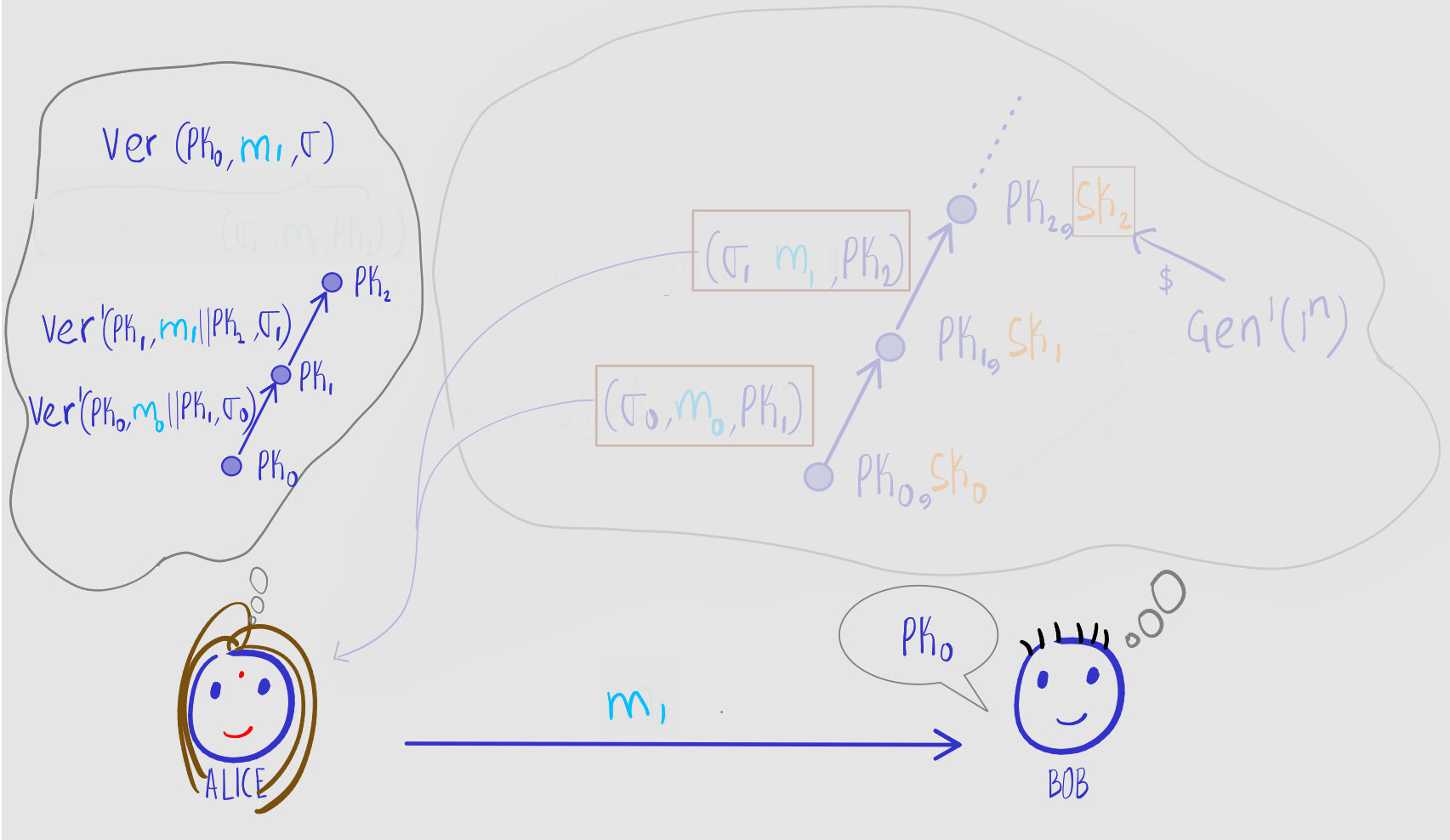
(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s. Idea: "chain signatures")



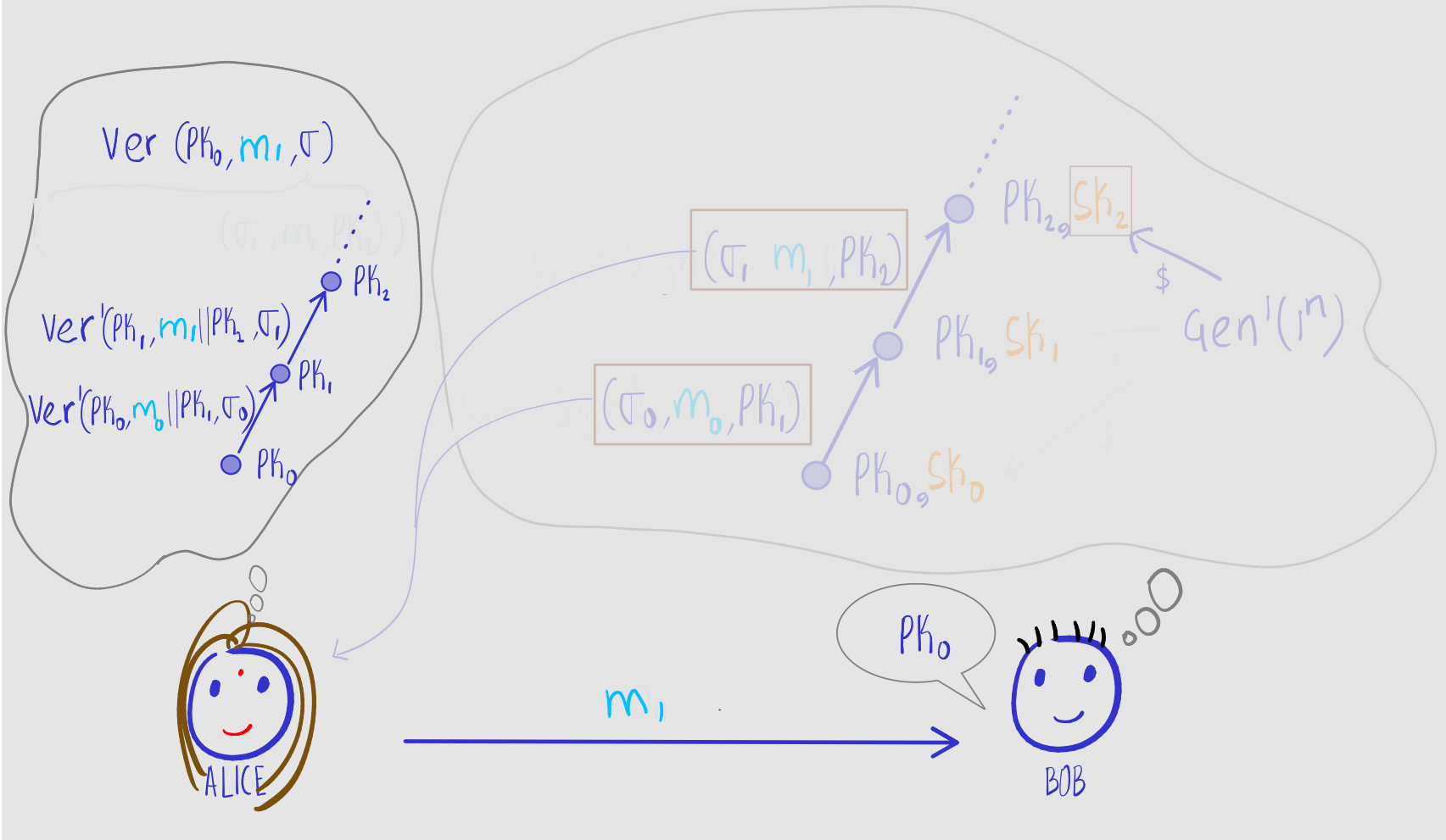
(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s. Idea: "chain signatures")



(Many-Time) Stateful Digital Signature...

Construction 2 (One-time DS $\Sigma^1 = (\mathsf{Gen}^1, \mathsf{Sign}^1, \mathsf{Ver}^1) \Rightarrow$ stateful DS Σ^s. Idea: "chain signatures")



(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

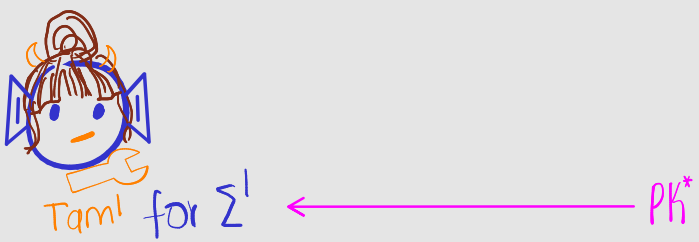
Proof sketch: plug and pray, again.

(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

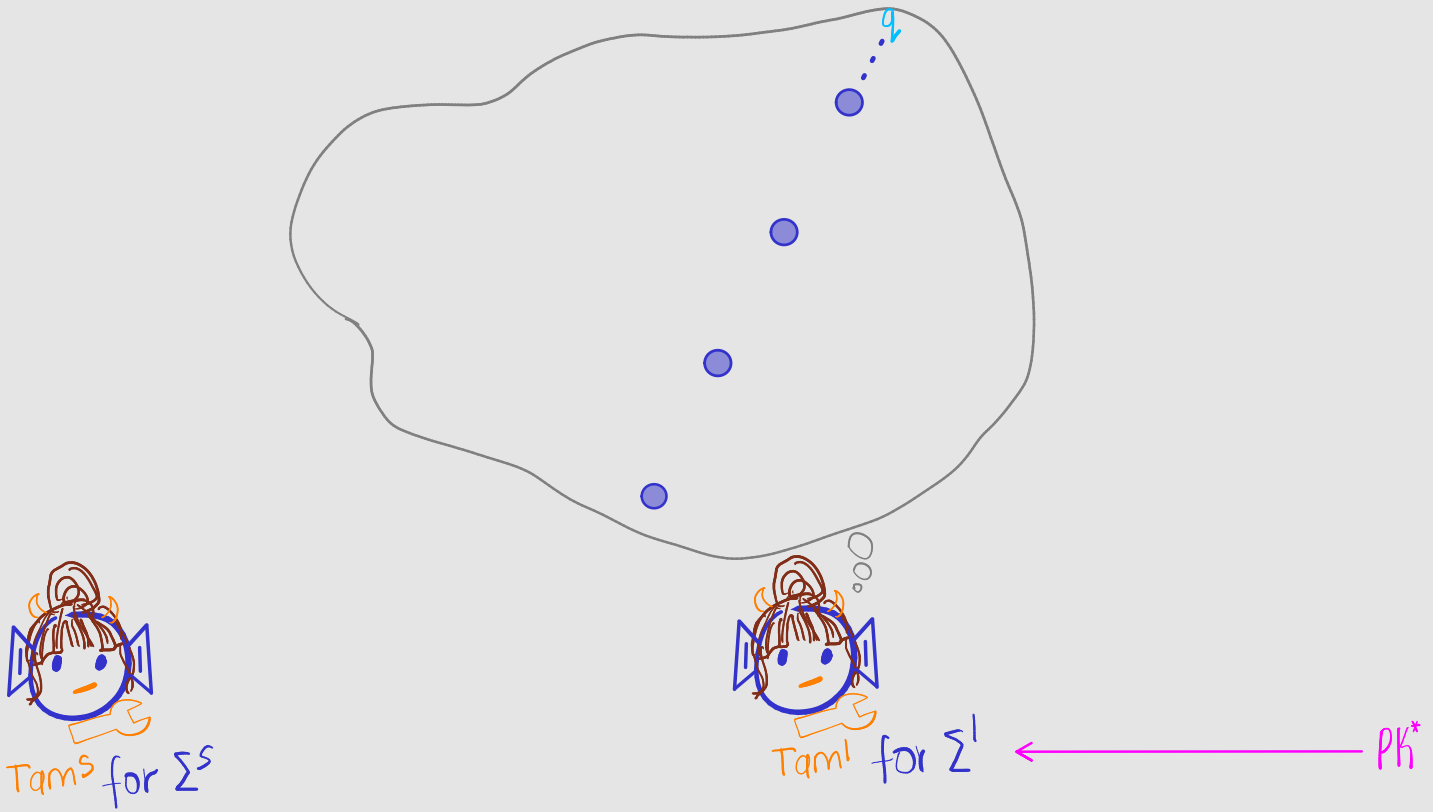


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

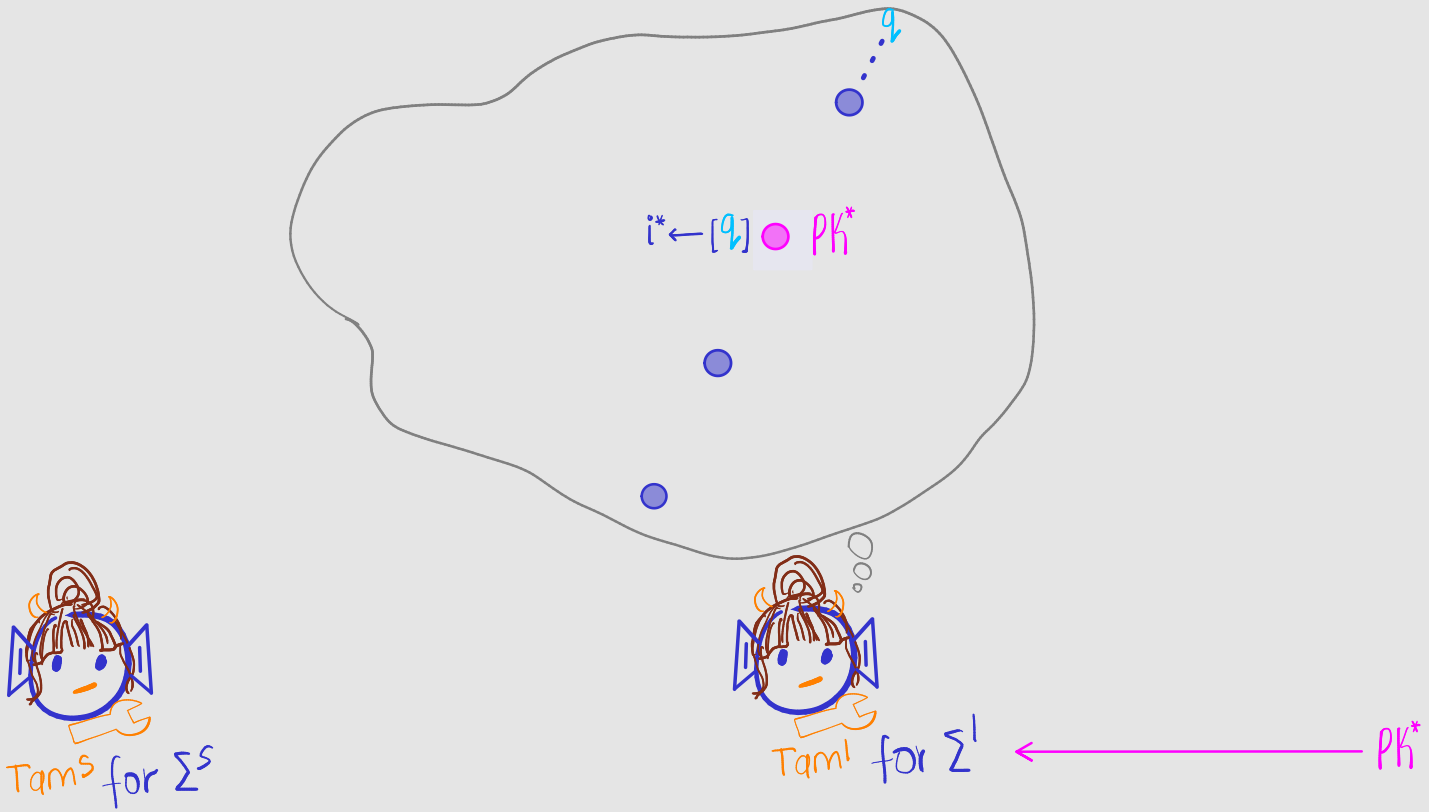


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

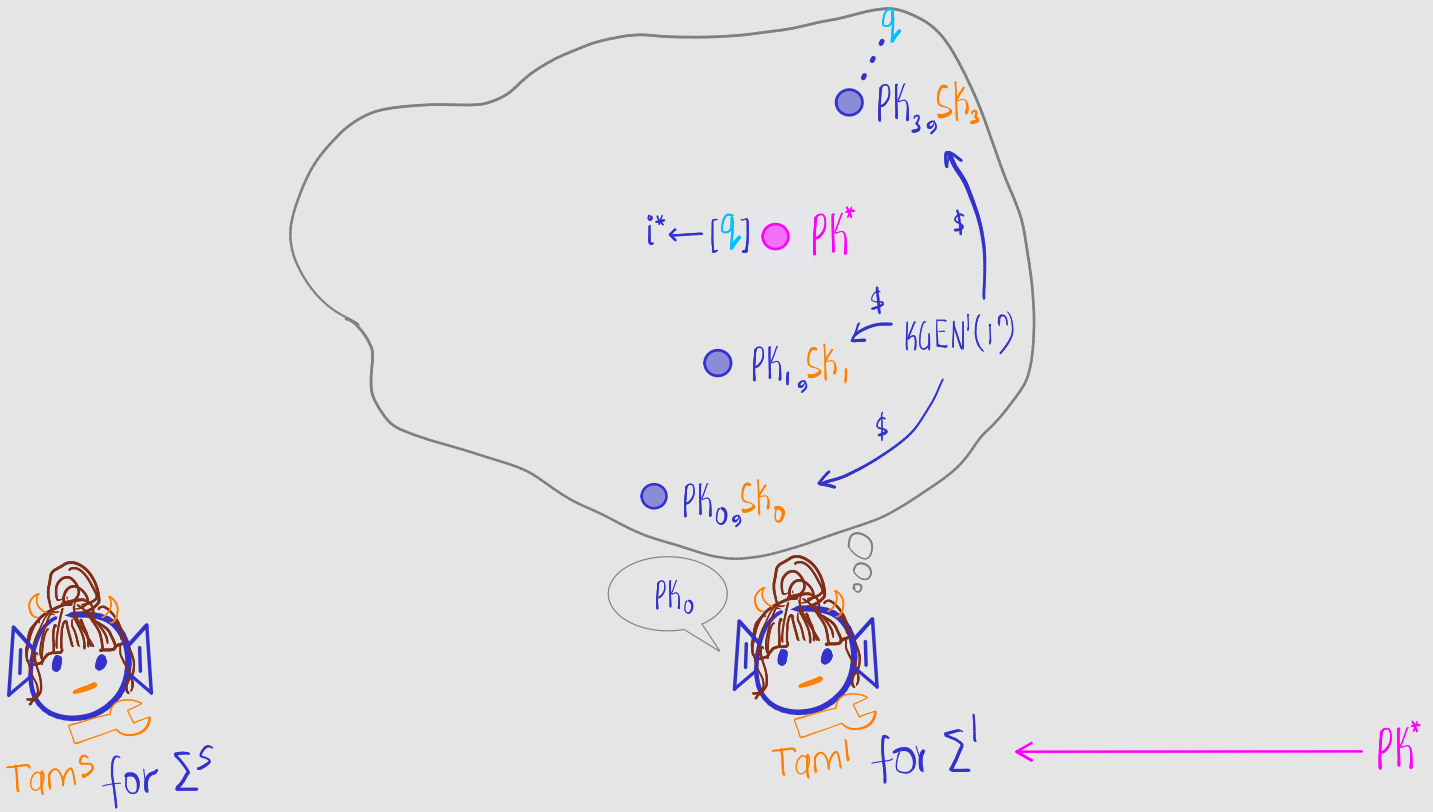


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

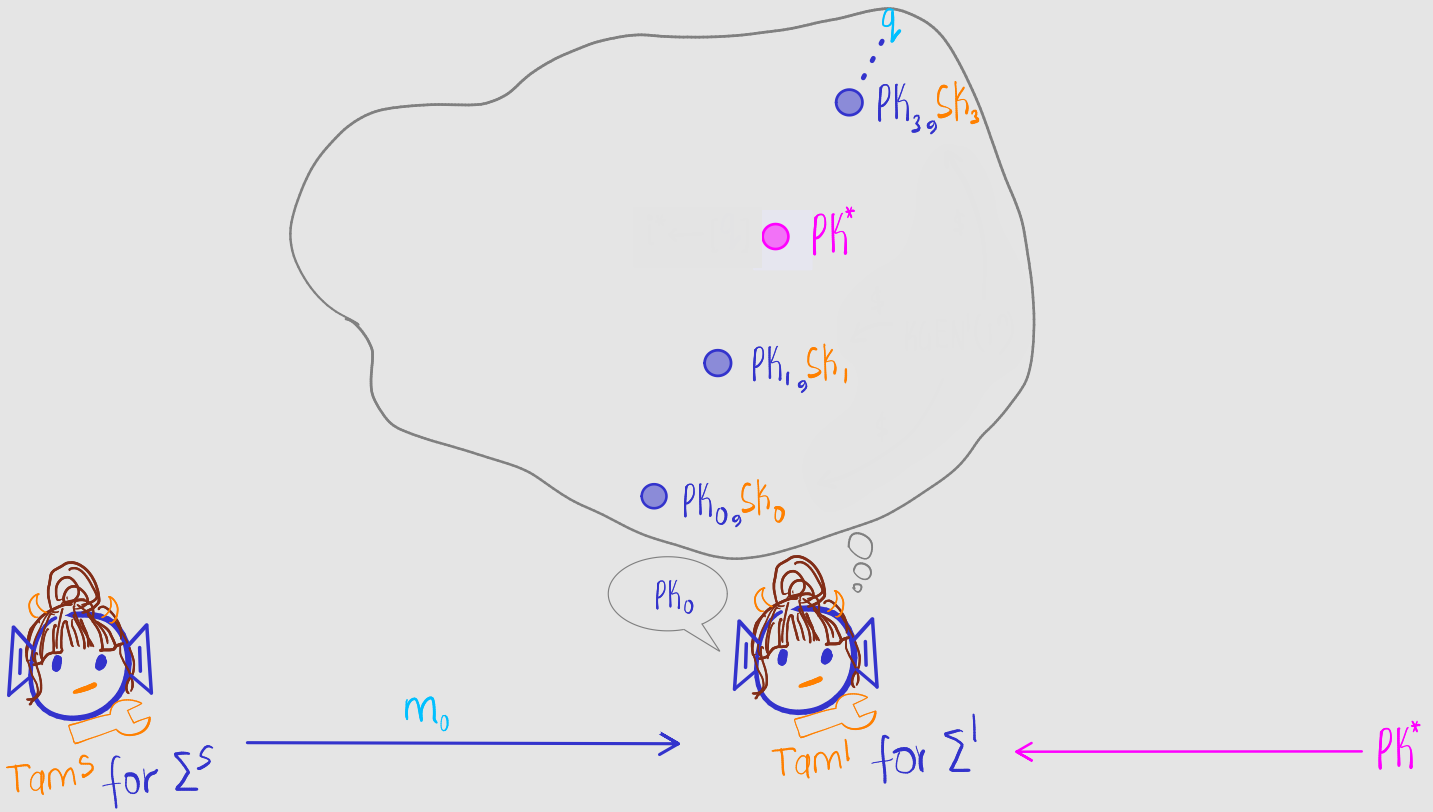


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

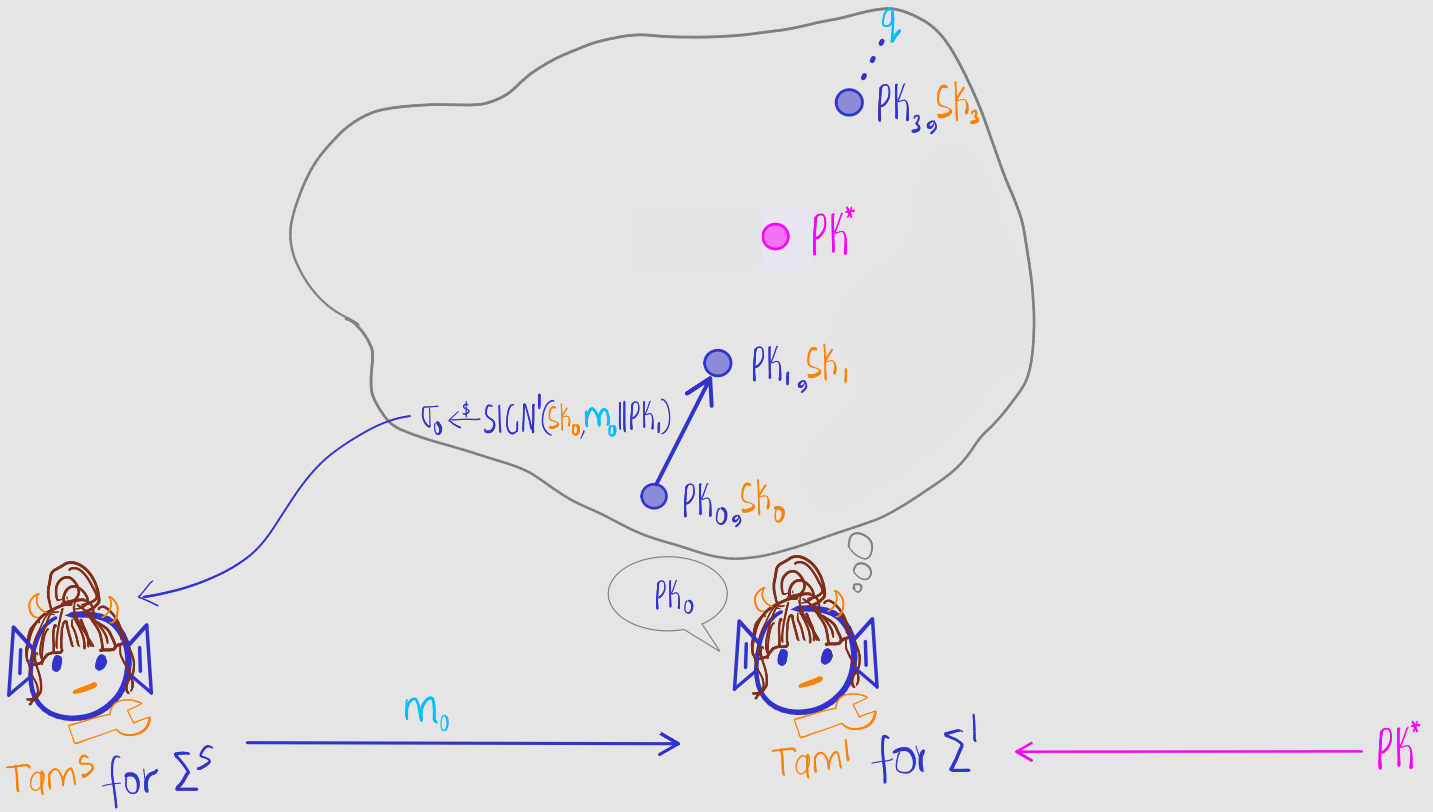


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

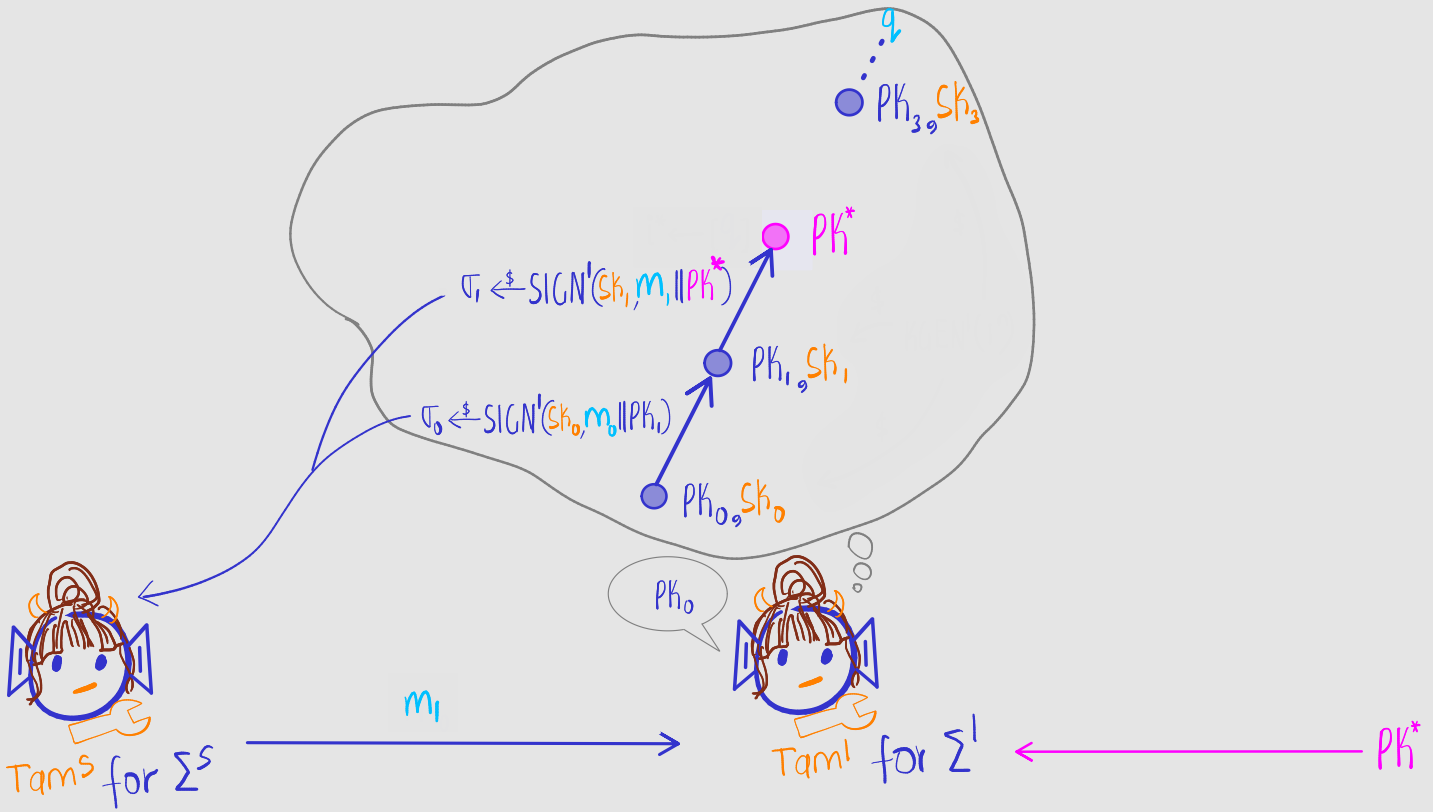


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

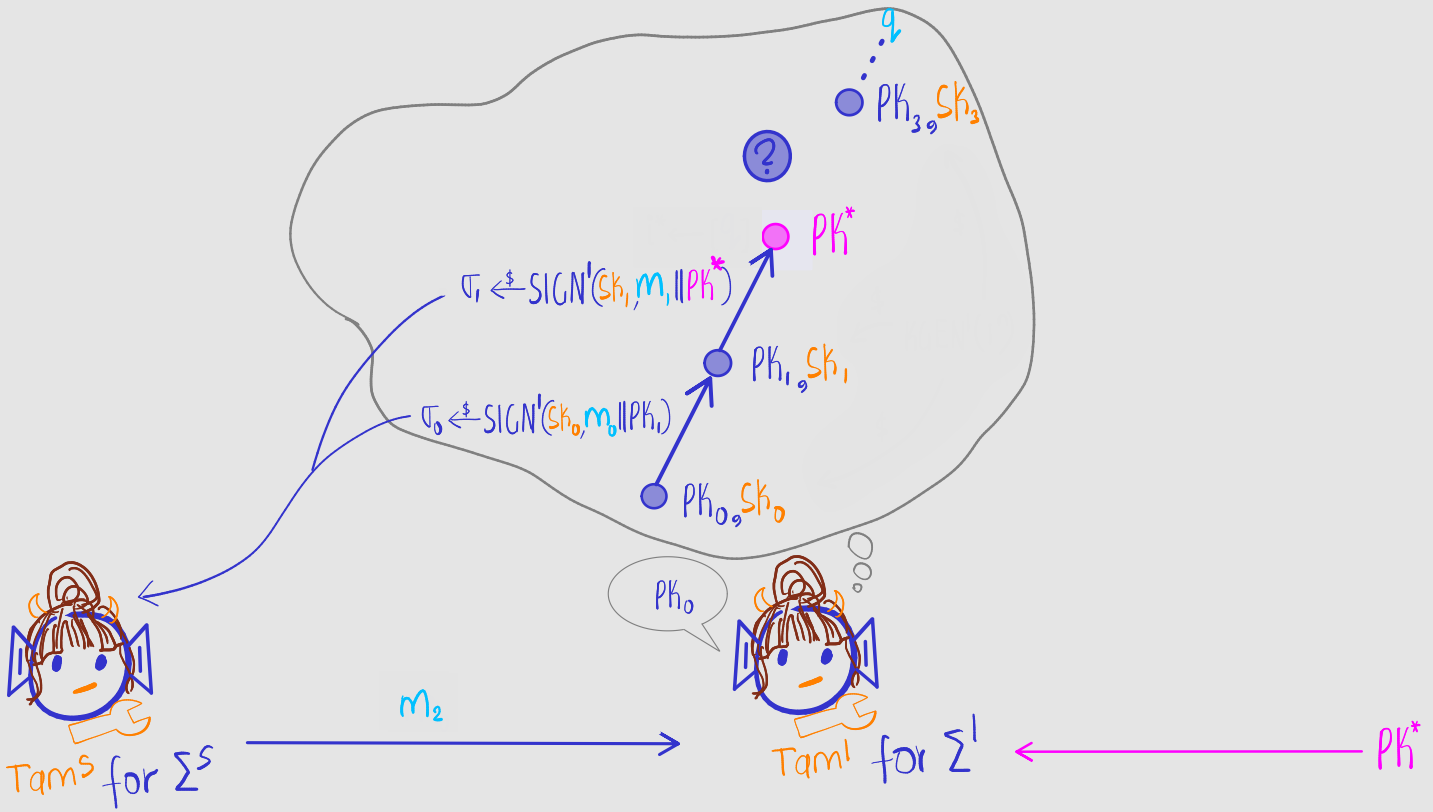


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

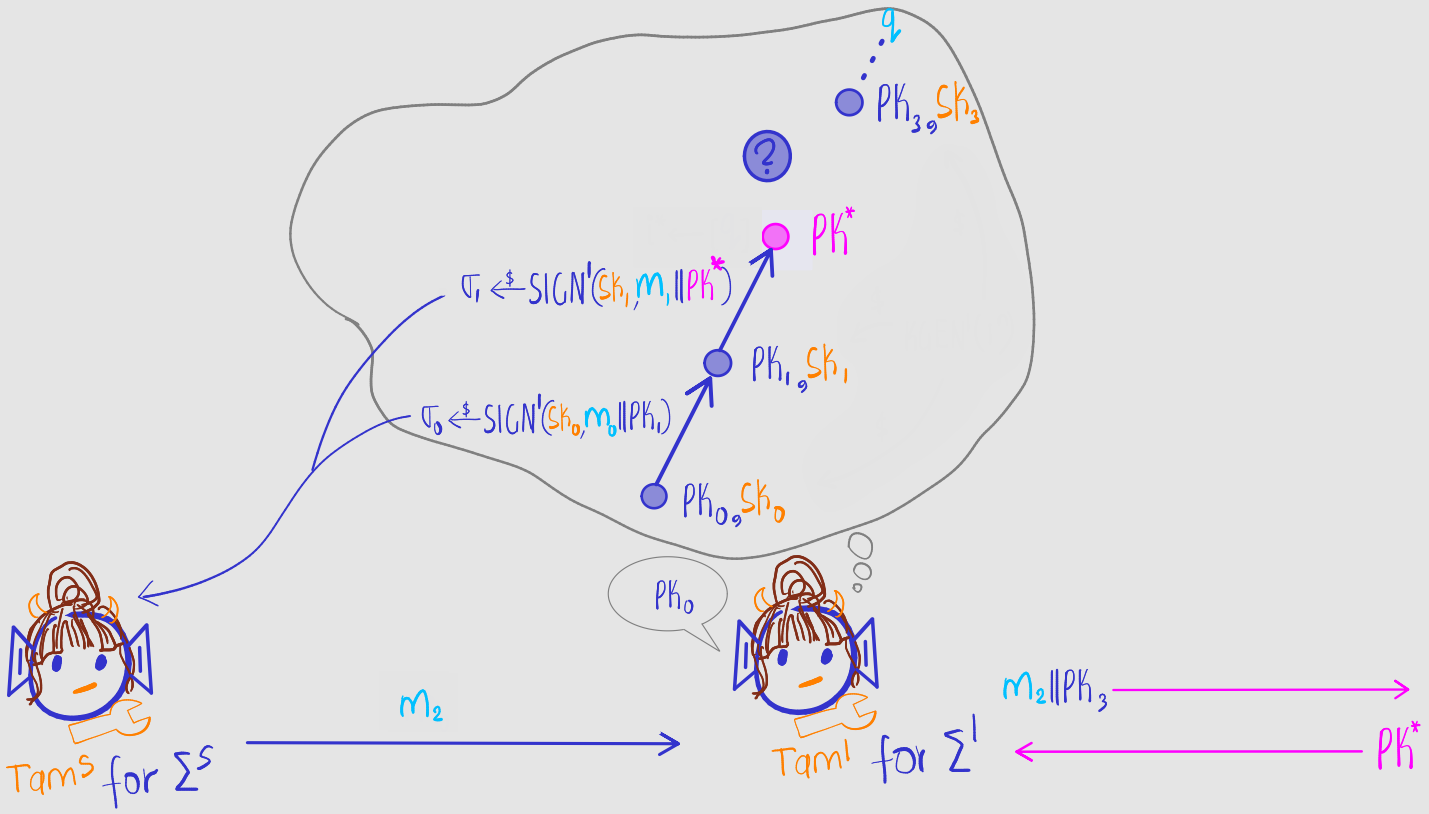


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

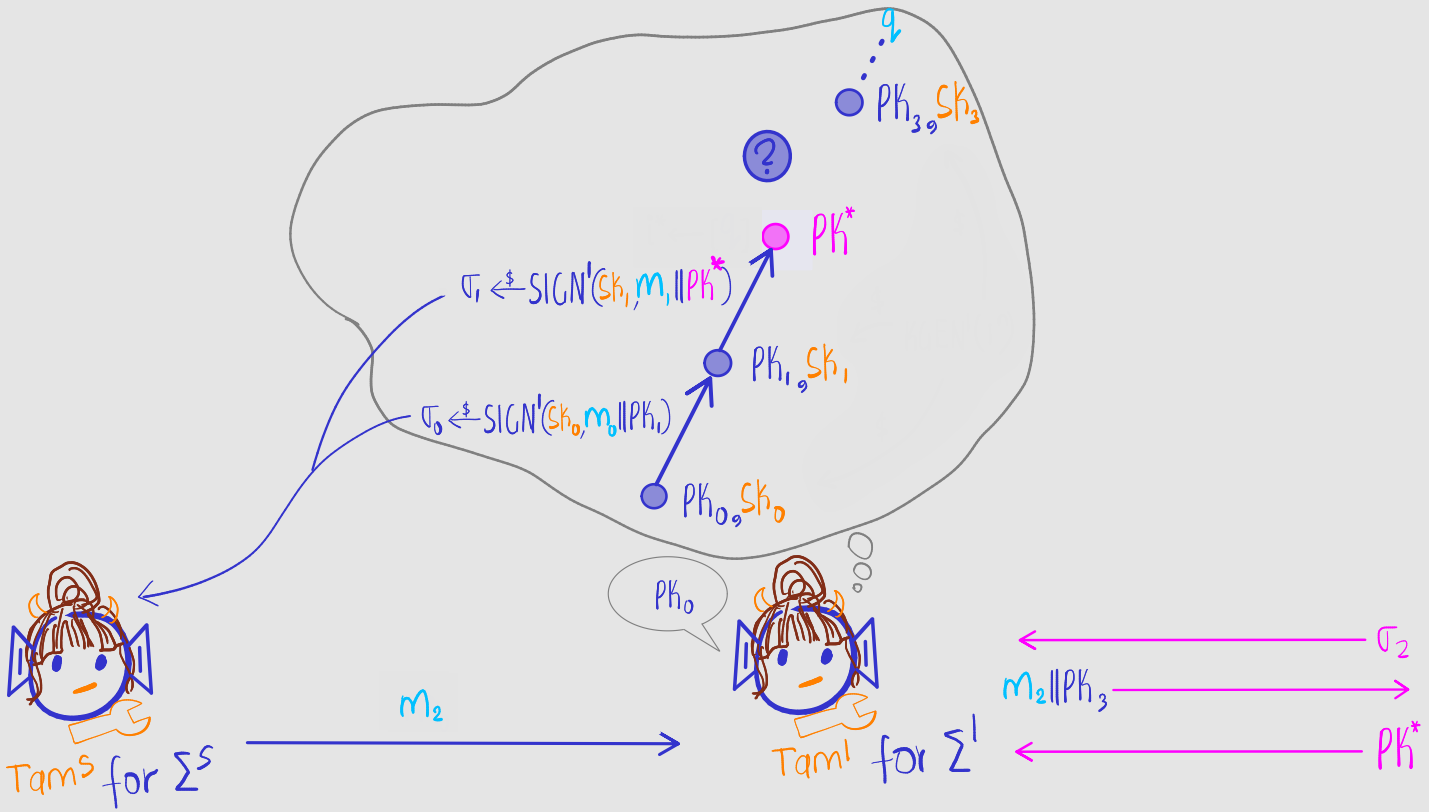


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

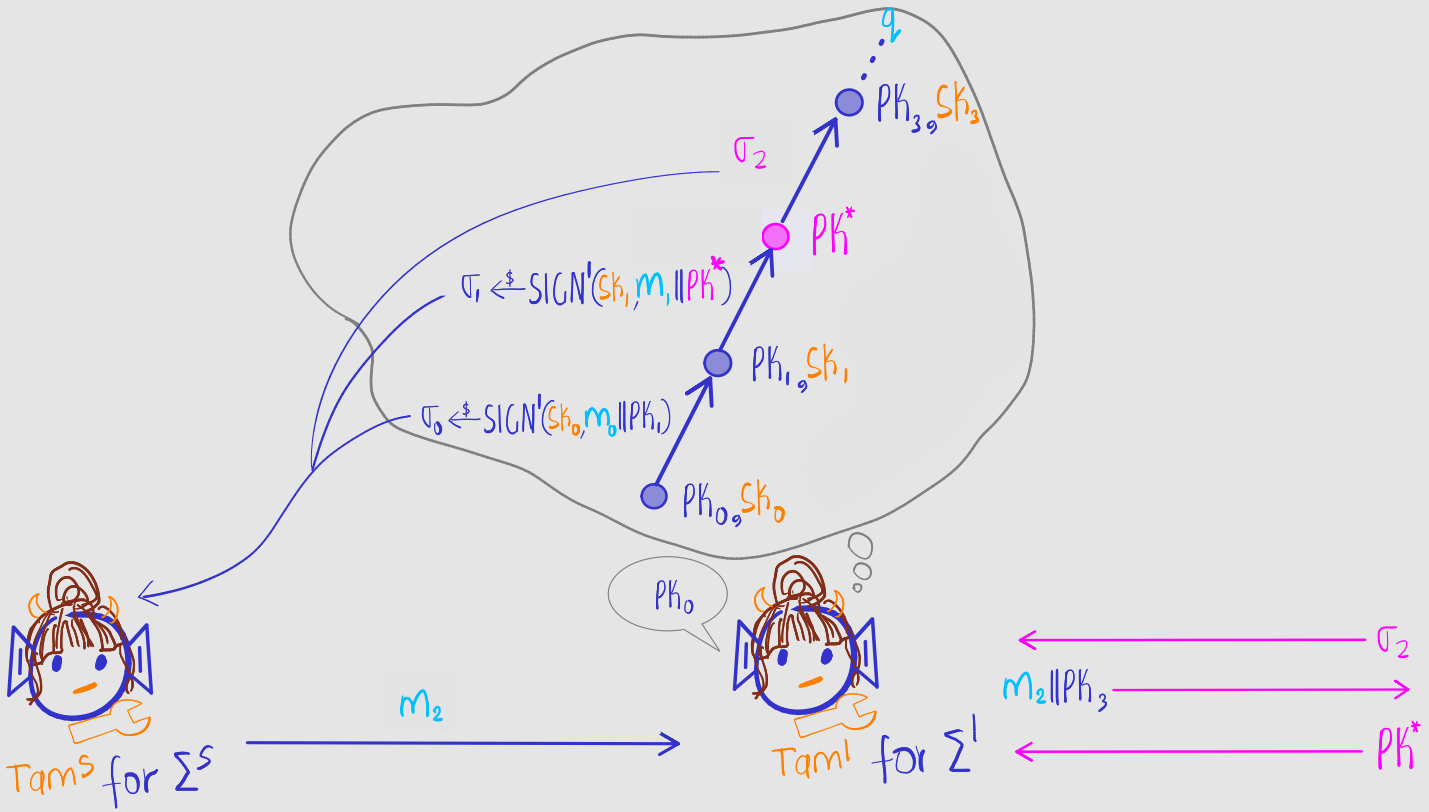


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

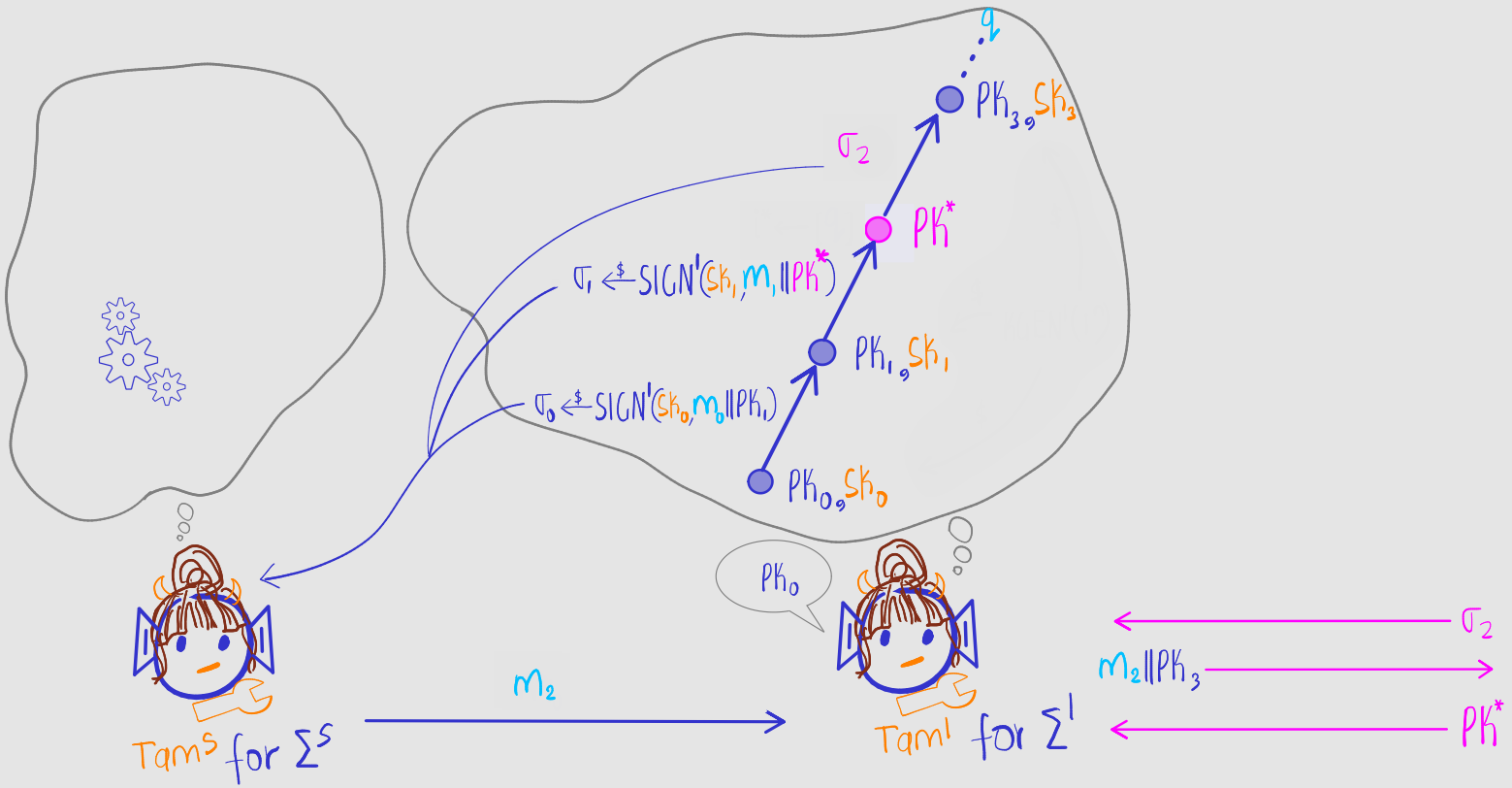


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

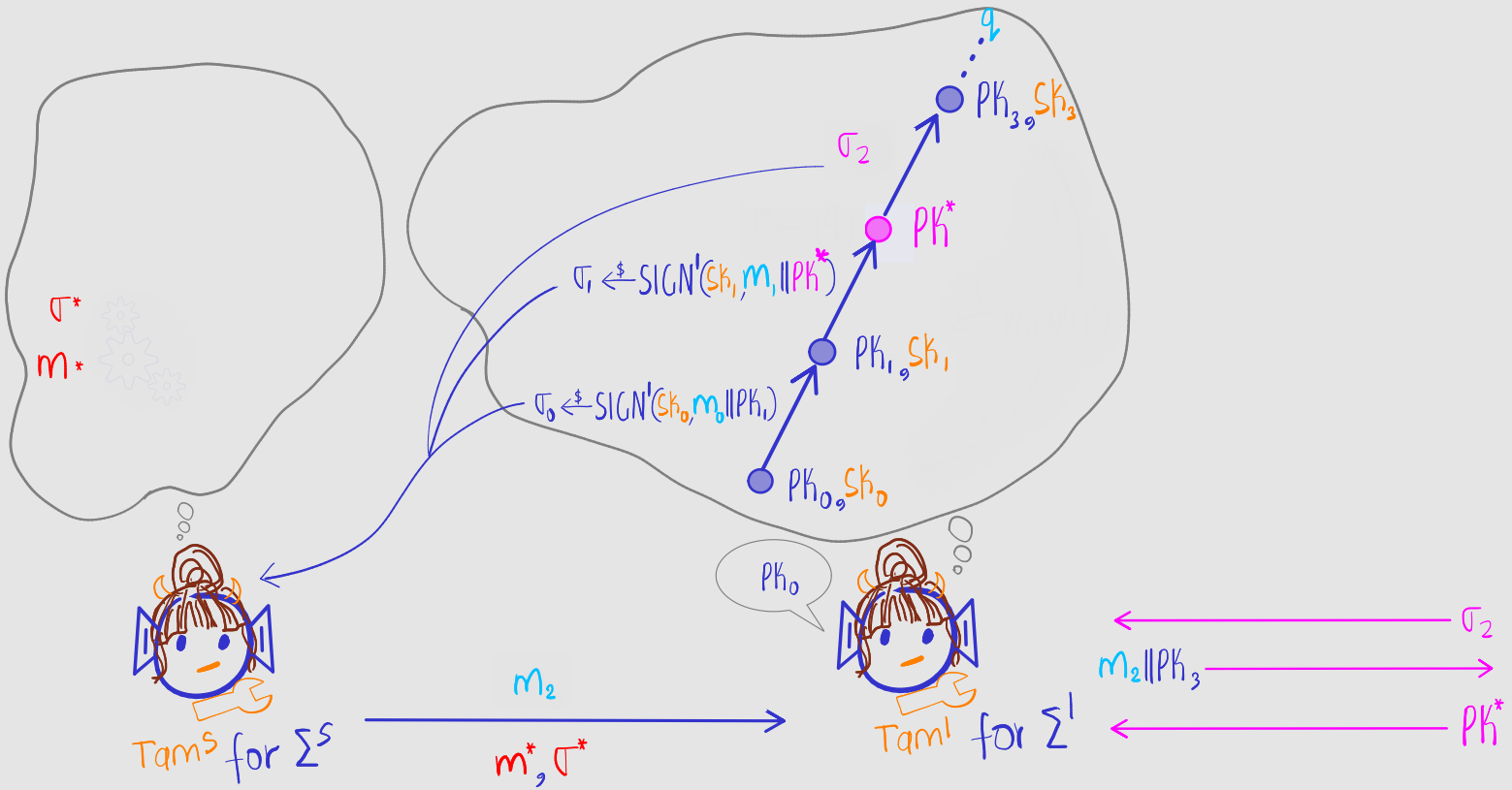


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

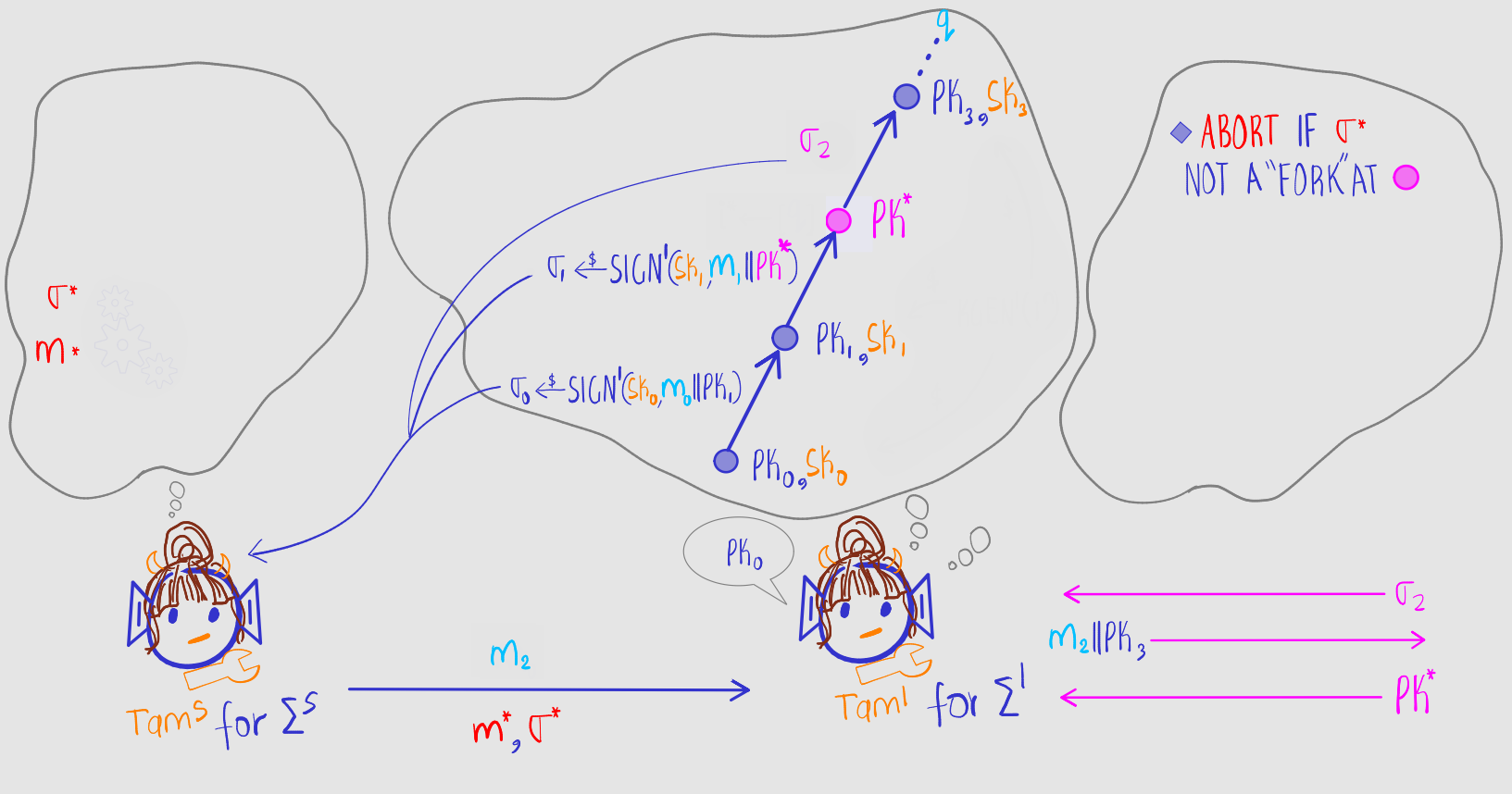


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

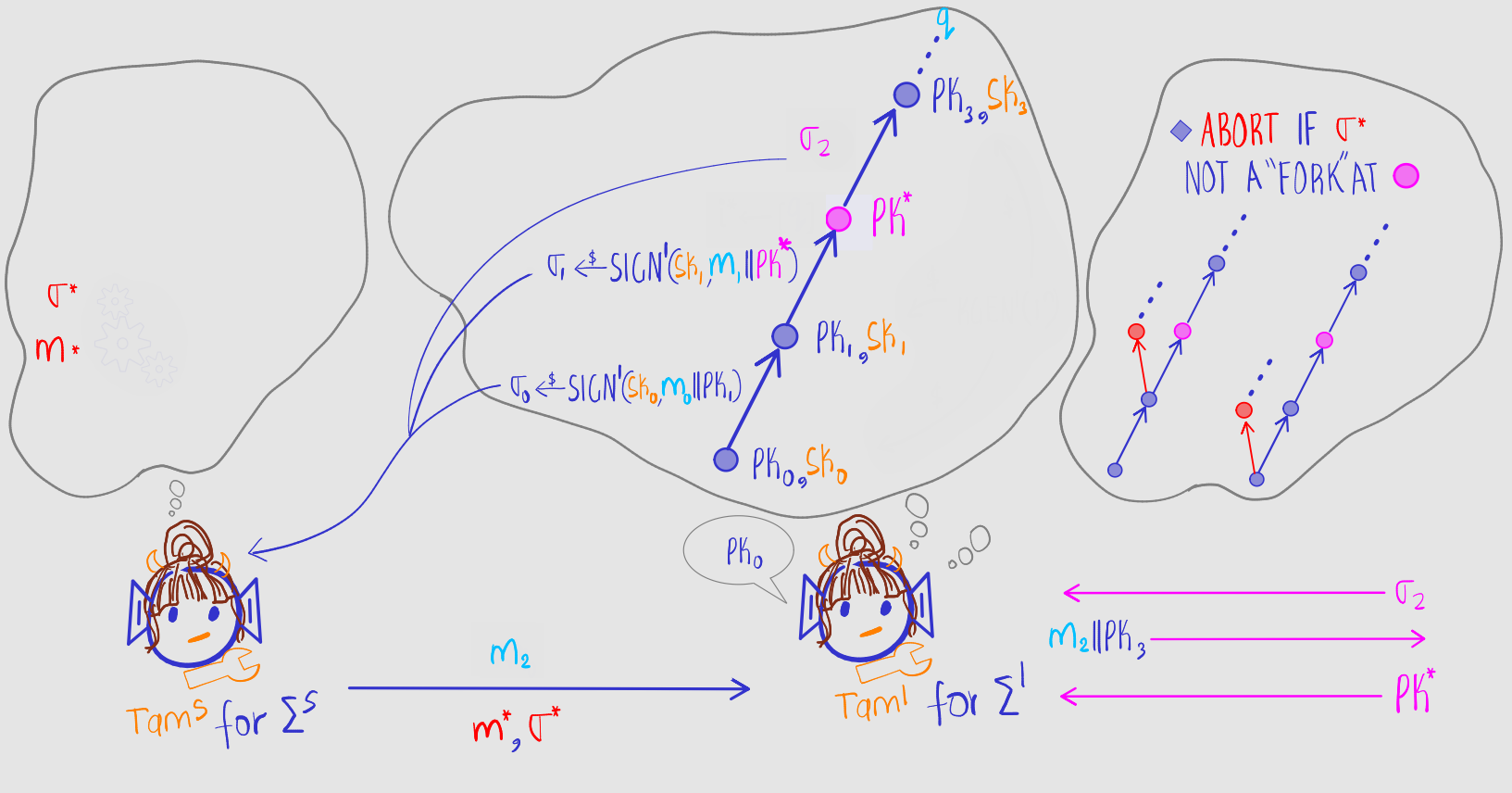


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

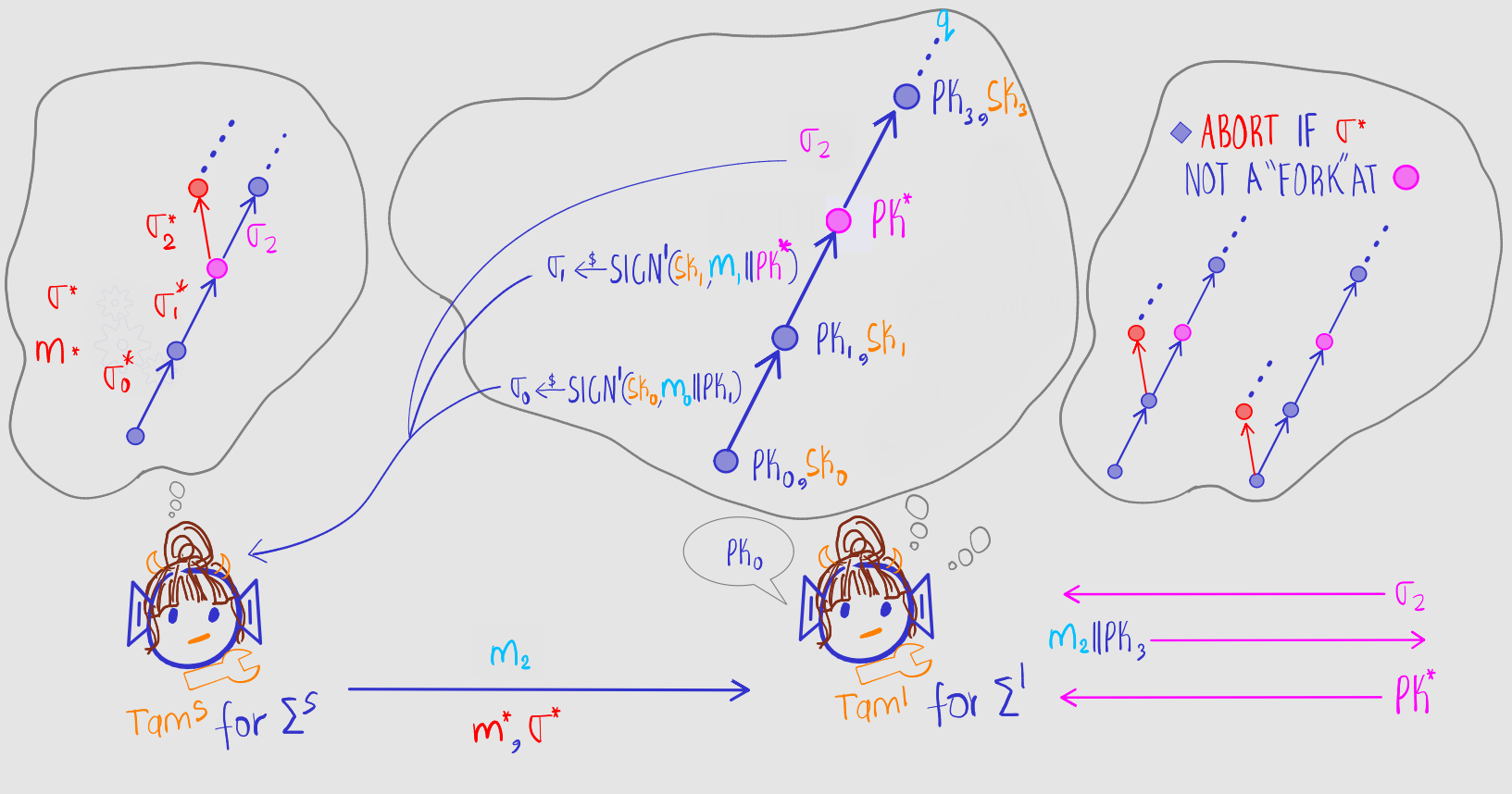


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

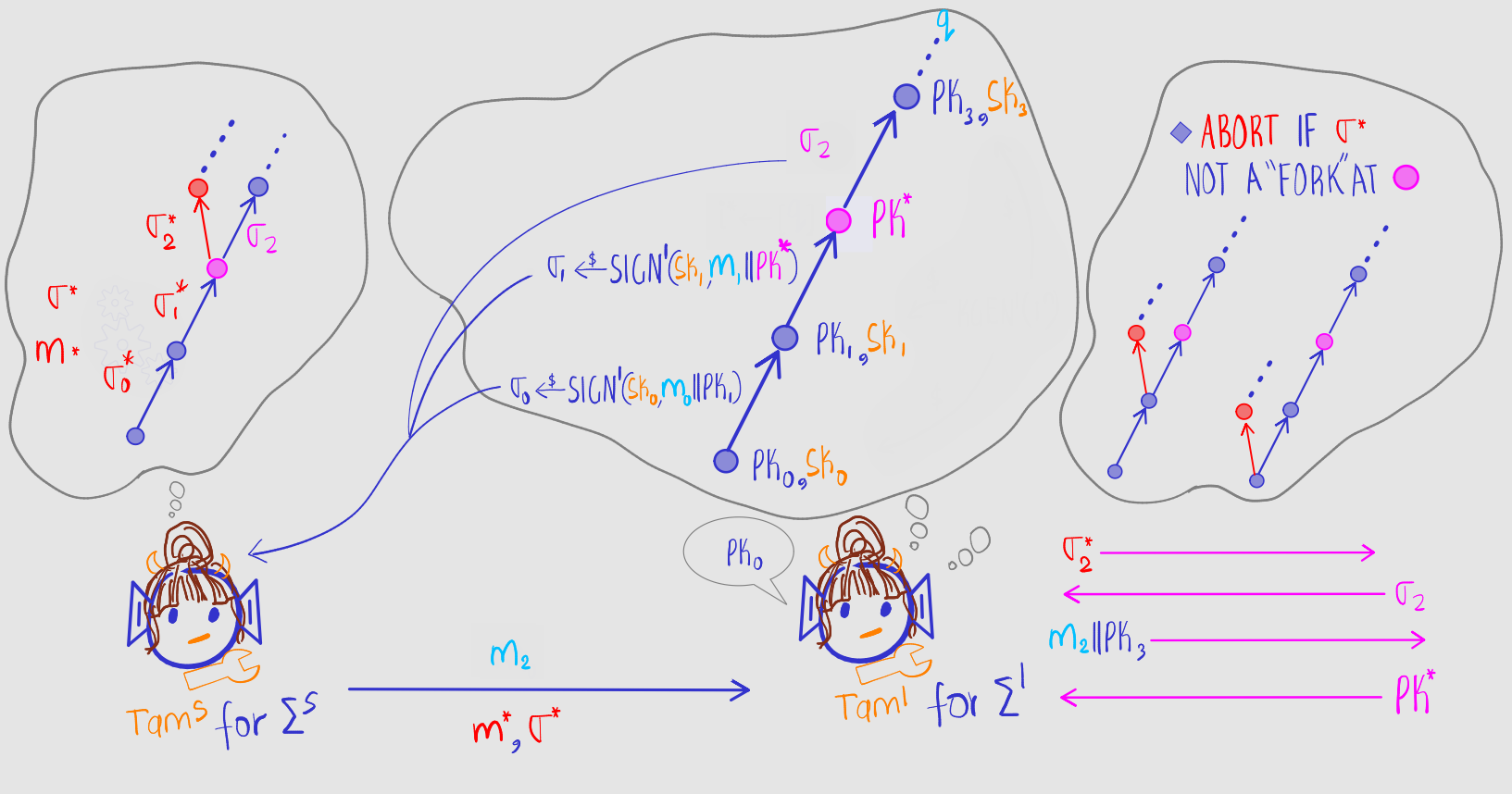


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.

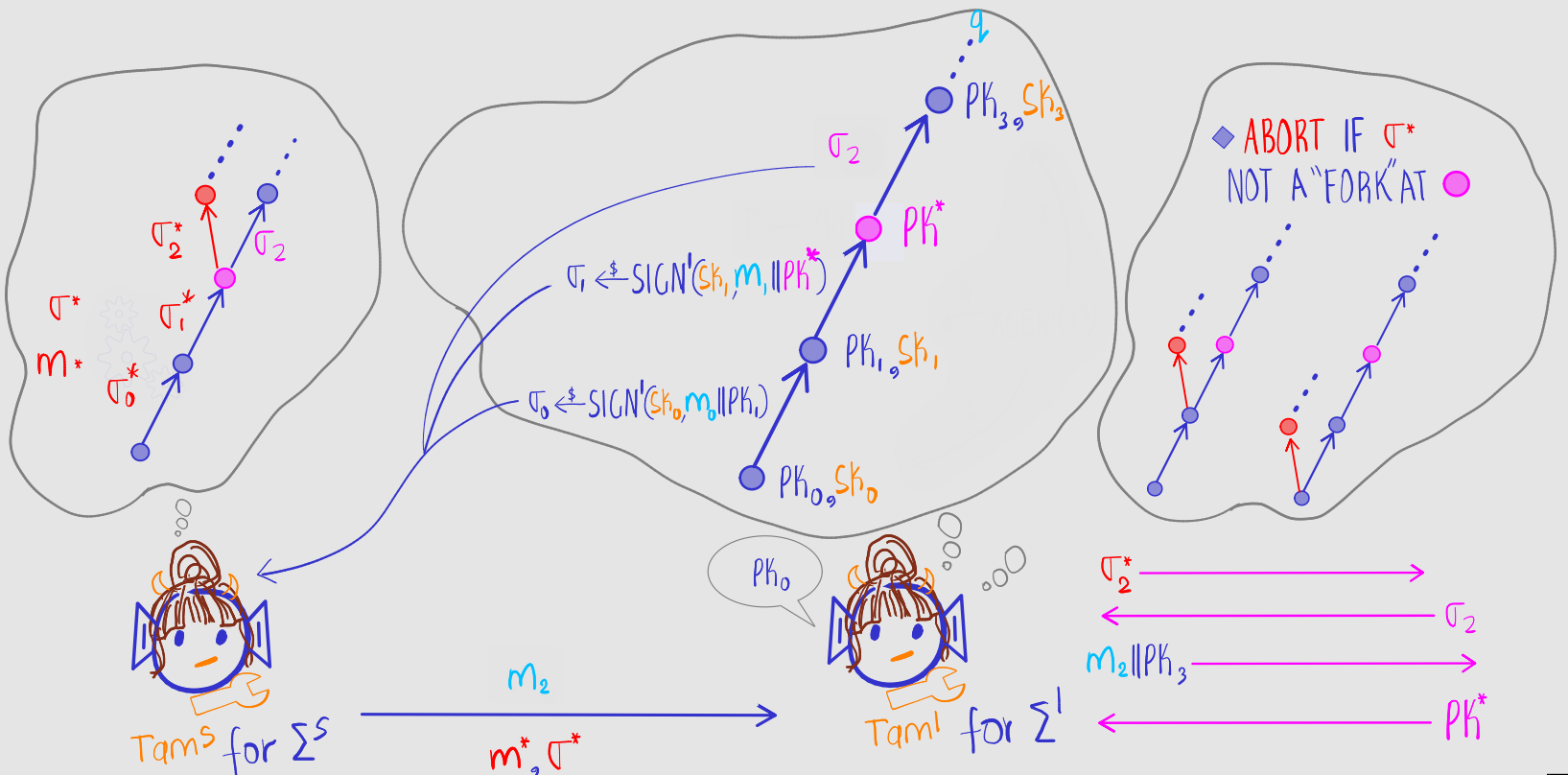


(Many-Time) Stateful Digital Signature...

Theorem 3

If Σ^1 is an one-time DS supporting arbitrary-length messages then Σ^s is a stateful DS.

Proof sketch: plug and pray, again.



□

(Many-Time) Stateful Digital Signature...

The size of signatures in Σ^s grows linearly with the number of signatures issued by the signer. How to fix this?

(Many-Time) Stateful Digital Signature...

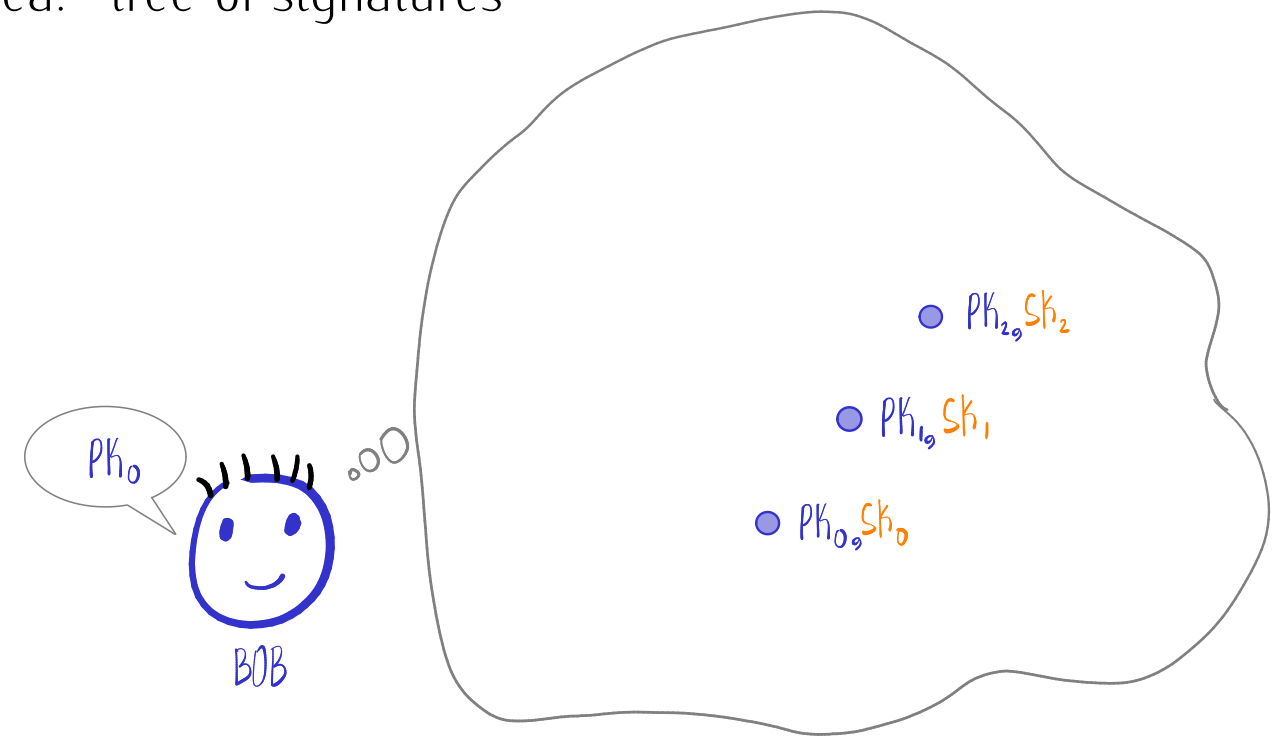
The size of signatures in Σ^s grows linearly with the number of signatures issued by the signer. How to fix this?

Idea: "tree of signatures"

(Many-Time) Stateful Digital Signature...

The size of signatures in Σ^s grows linearly with the number of signatures issued by the signer. How to fix this?

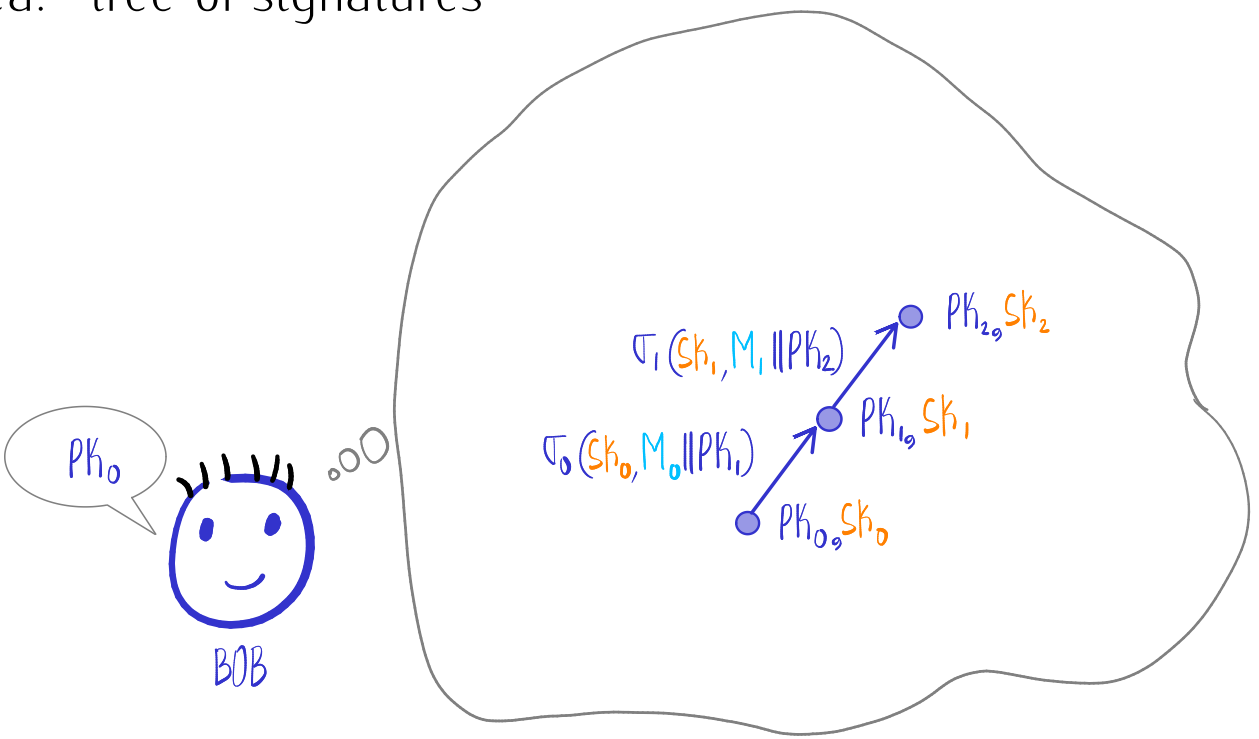
Idea: "tree of signatures"



(Many-Time) Stateful Digital Signature...

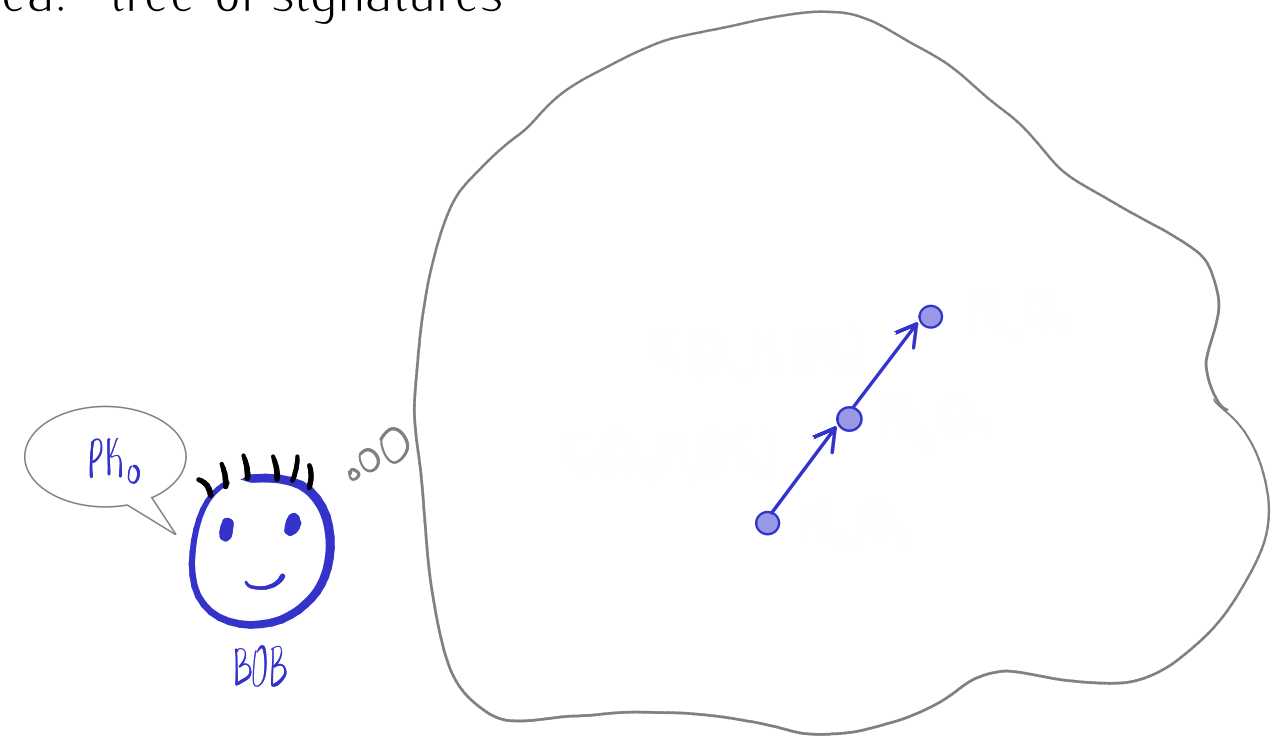
The size of signatures in Σ^s grows linearly with the number of signatures issued by the signer. How to fix this?

Idea: "tree of signatures"



(Many-Time) Stateful Digital Signature...

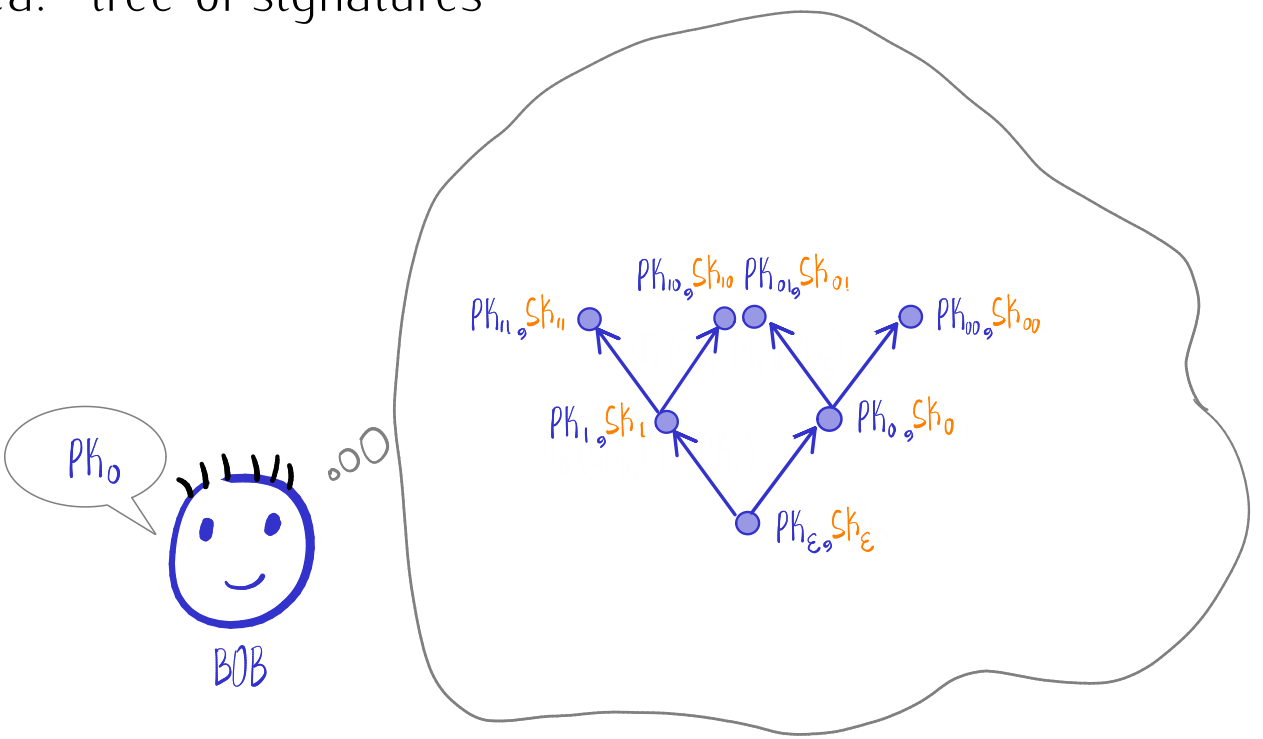
- The size of signatures in Σ^s grows linearly with the number of signatures issued by the signer. How to fix this?
  - Idea: "tree of signatures"



(Many-Time) Stateful Digital Signature...

The size of signatures in Σ^s grows linearly with the number of signatures issued by the signer. How to fix this?

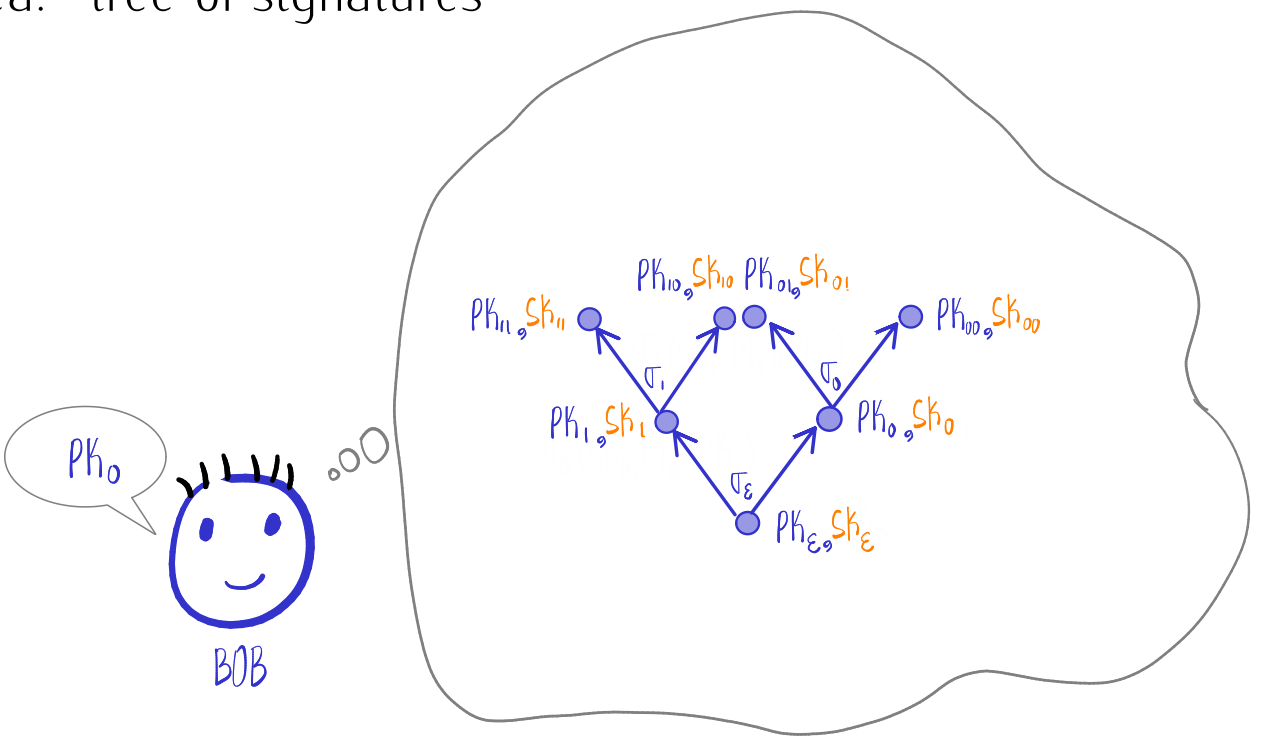
Idea: "tree of signatures"



(Many-Time) Stateful Digital Signature...

The size of signatures in Σ^s grows linearly with the number of signatures issued by the signer. How to fix this?

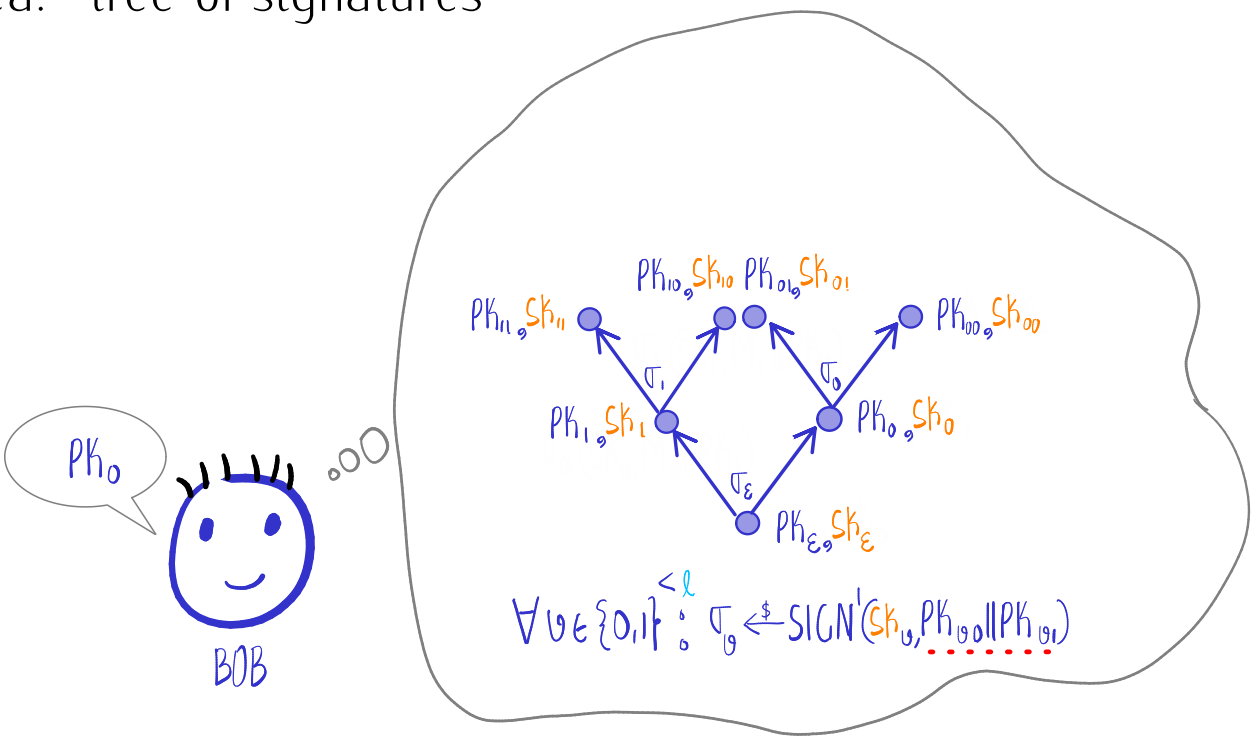
Idea: "tree of signatures"



(Many-Time) Stateful Digital Signature...

The size of signatures in Σ^s grows linearly with the number of signatures issued by the signer. How to fix this?

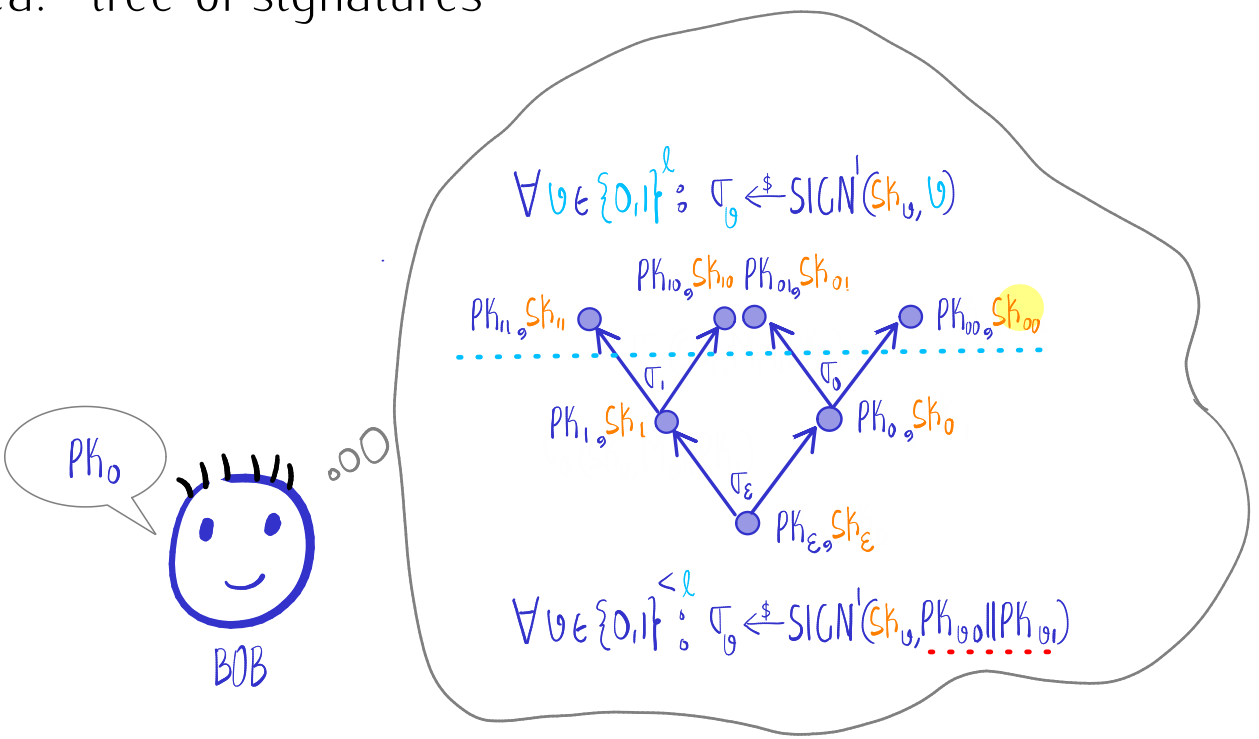
Idea: "tree of signatures"



(Many-Time) Stateful Digital Signature...

The size of signatures in Σ^s grows linearly with the number of signatures issued by the signer. How to fix this?

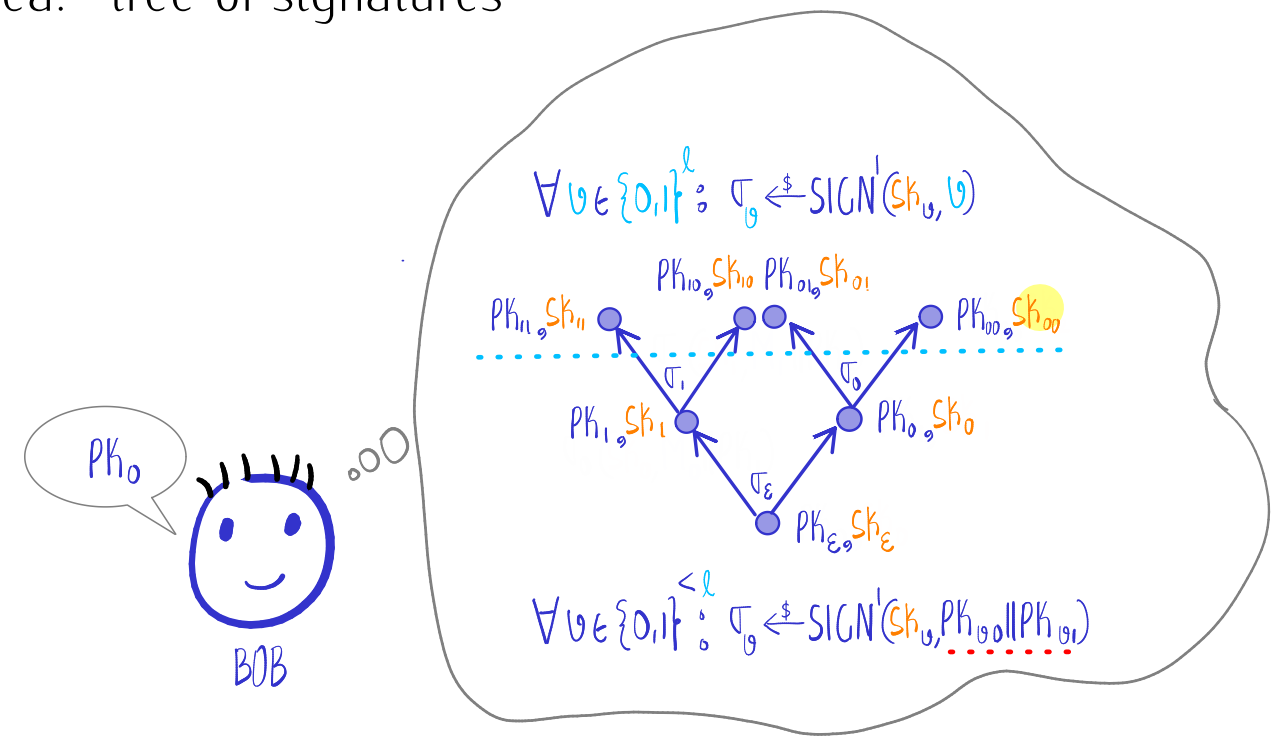
Idea: "tree of signatures"



(Many-Time) Stateful Digital Signature...

The size of signatures in Σ^s grows linearly with the number of signatures issued by the signer. How to fix this?

Idea: "tree of signatures"



Exercise 5 (Compact stateful DS)

Prove that the construction Σ^c is secure. (Hint: plug and pray.)

(Many-Time) Digital Signature...

- Compact stateful DS Σ^c + pseudo-random function $F_K : \{0,1\}^{\ell+1} \to \{0,1\}^n \Rightarrow$ DS Σ

(Many-Time) Digital Signature...

- Compact stateful DS Σ^c + pseudo-random function $F_K : \{0,1\}^{\ell+1} \to \{0,1\}^n \Rightarrow$ DS Σ
  -  Idea: Use to *derandomise* underlying signature and key gen.

(Many-Time) Digital Signature...

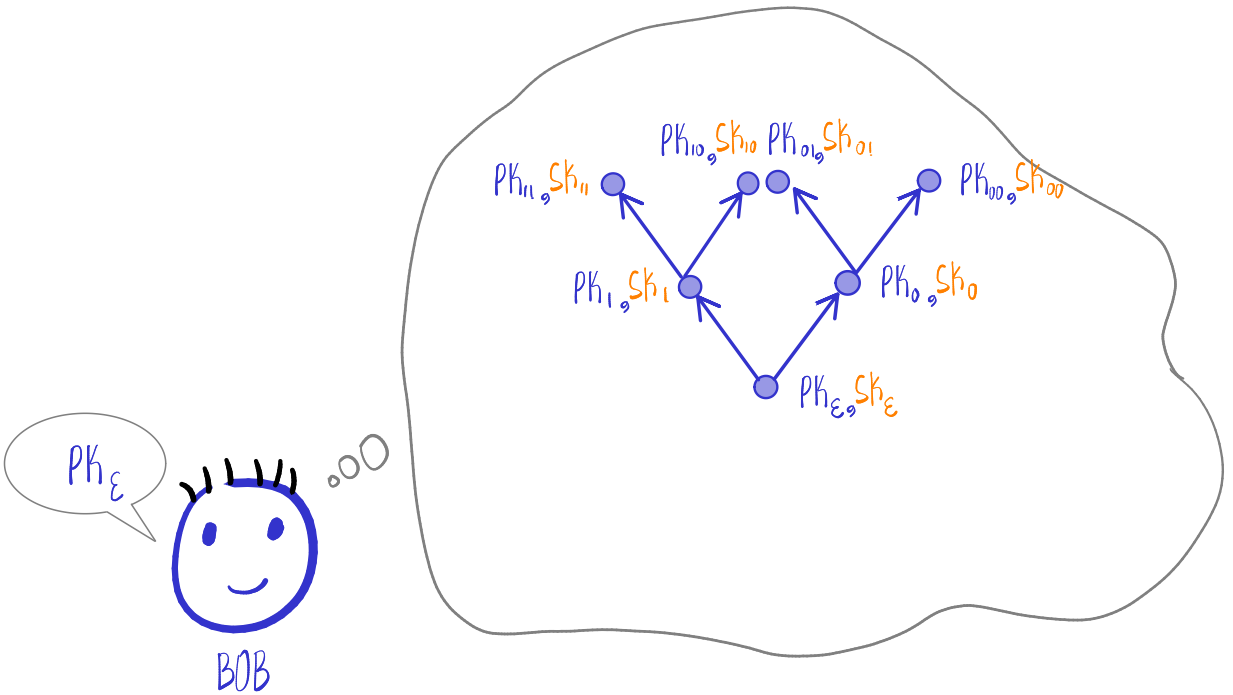
- Compact stateful DS Σ^c + pseudo-random function $F_K : \{0,1\}^{\ell+1} \to \{0,1\}^n \Rightarrow$ DS Σ
  -  Idea: Use to *derandomise* underlying signature and key gen.

(Many-Time) Digital Signature...

- Compact stateful DS Σ^c + pseudo-random function $F_K : \{0,1\}^{\ell+1} \to \{0,1\}^n \Rightarrow$ DS Σ
  -  Idea: Use to *derandomise* underlying signature and key gen.

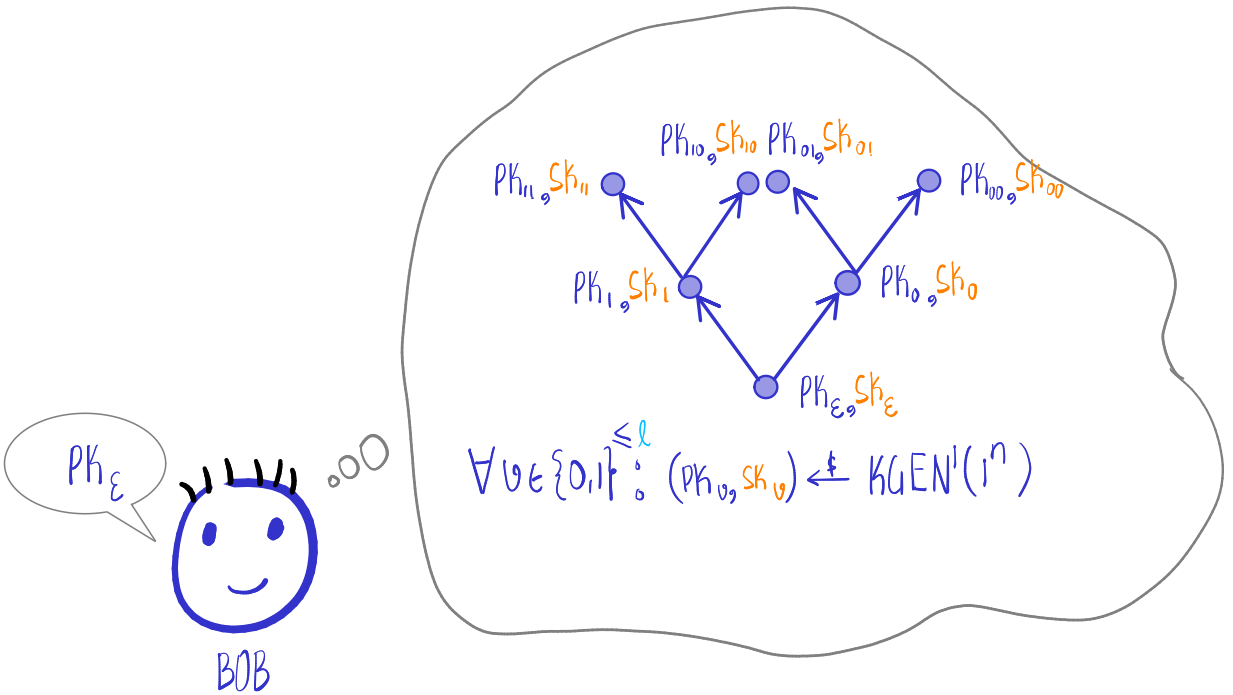
(Many-Time) Digital Signature...

- Compact stateful DS Σ^c + pseudo-random function $F_K : \{0,1\}^{\ell+1} \to \{0,1\}^n \Rightarrow$ DS Σ
  - Idea: Use to *derandomise* underlying signature and key gen.



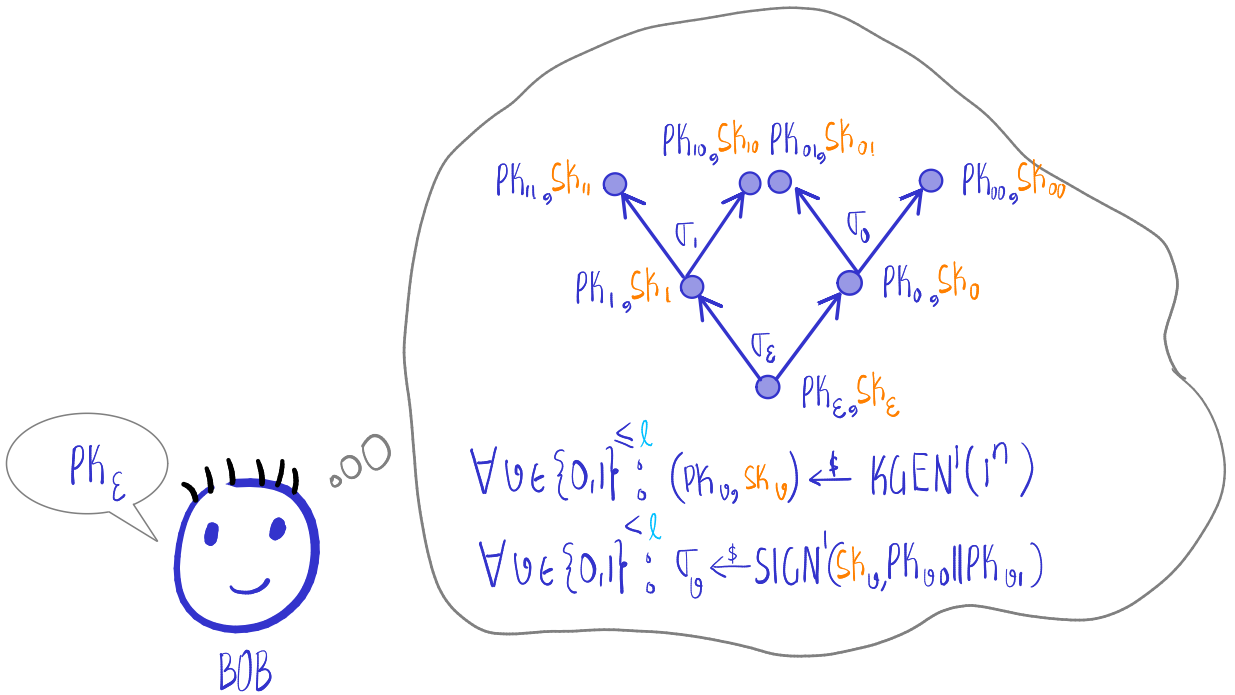
(Many-Time) Digital Signature...

- Compact stateful DS Σ^c + pseudo-random function
  $F_K : \{0,1\}^{\ell+1} \to \{0,1\}^n \Rightarrow$ DS Σ
  - Idea: Use to *derandomise* underlying signature and key gen.



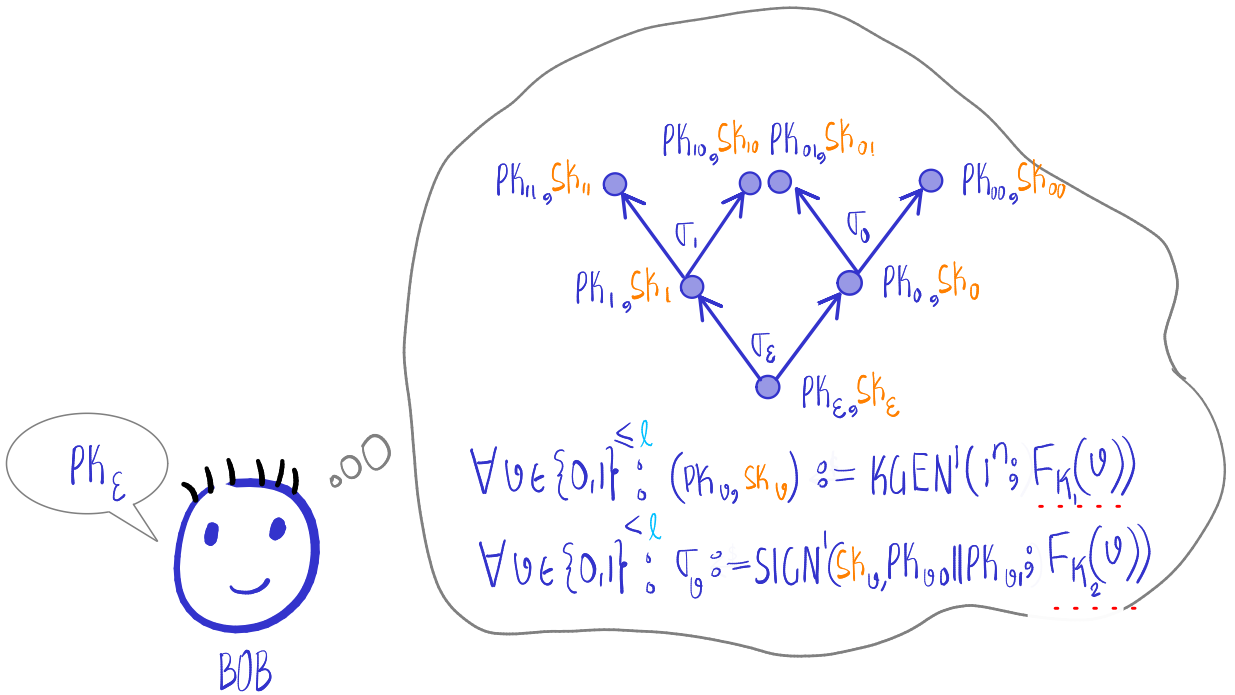
(Many-Time) Digital Signature...

- Compact stateful DS Σ^c + pseudo-random function
  $F_K : \{0,1\}^{\ell+1} \to \{0,1\}^n \Rightarrow$ DS Σ
  - Idea: Use to *derandomise* underlying signature and key gen.



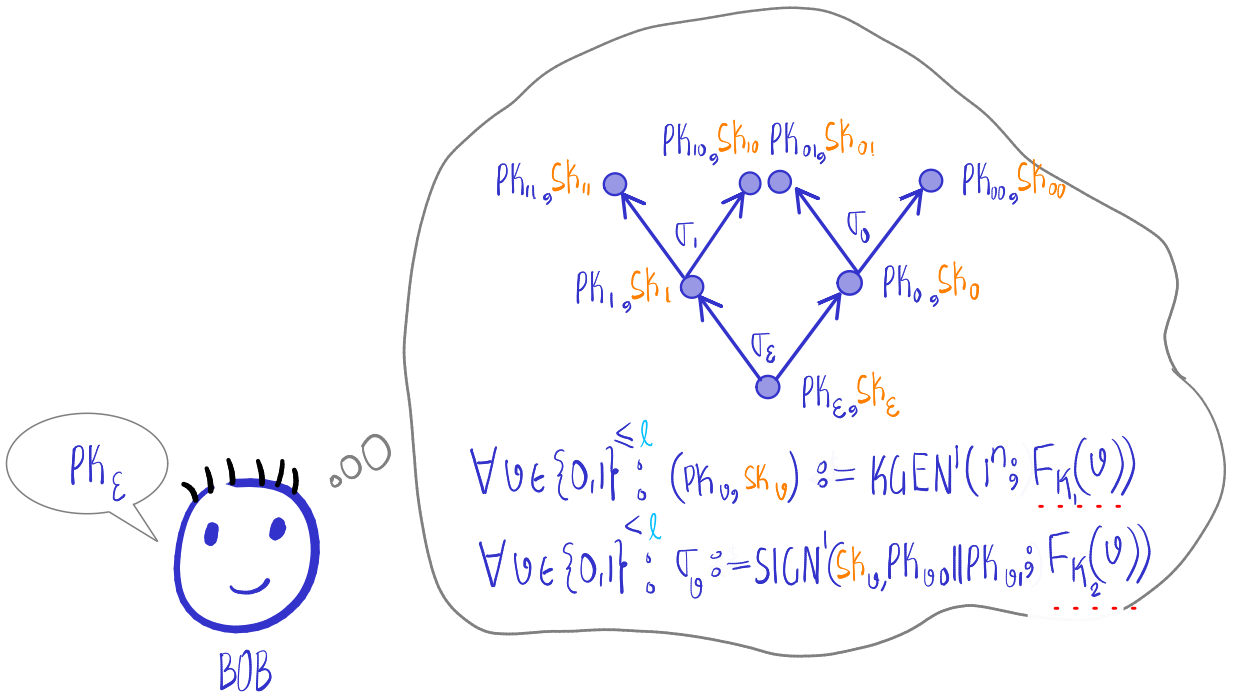
(Many-Time) Digital Signature...

- Compact stateful DS Σ^c + pseudo-random function
  $F_K : \{0,1\}^{\ell+1} \to \{0,1\}^n \Rightarrow$ DS Σ
  - Idea: Use to *derandomise* underlying signature and key gen.



(Many-Time) Digital Signature...

- Compact stateful DS Σ^c + pseudo-random function
  $F_K : \{0,1\}^{\ell+1} \to \{0,1\}^n \Rightarrow$ DS Σ
  - Idea: Use to *derandomise* underlying signature and key gen.



Exercise 6 (EU-CMA-secure DS)

Prove that Σ is secure.

To Recap Today's Lecture

- Introduced digital signatures: public-key analogue of MAC
- Theoretical constructions of DS
  - Lamport's one-time DS
  - Tree-based many-time (stateful) DS from one-time DS

To Recap Today's Lecture

- Introduced digital signatures: public-key analogue of MAC
- Theoretical constructions of DS
  - Lamport's one-time DS
  - Tree-based many-time (stateful) DS from one-time DS
- Lectures 13 and 15(?): efficient DS in "random-oracle model"
  - From trapdoor OWF via hash-then-invert
  - Via Fiat-Shamir transform (e.g., Schnorr)

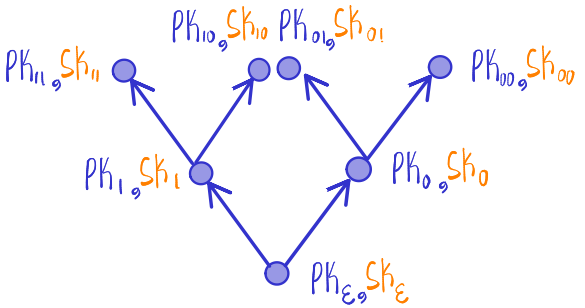
To Recap Today's Lecture

- Introduced digital signatures: public-key analogue of MAC
- Theoretical constructions of DS
  - Lamport's one-time DS
  - Tree-based many-time (stateful) DS from one-time DS
- Lectures 13 and 15(?): efficient DS in "random-oracle model"
  - From trapdoor OWF via hash-then-invert
  - Via Fiat-Shamir transform (e.g., Schnorr)

- Takeaways:

To Recap Today's Lecture

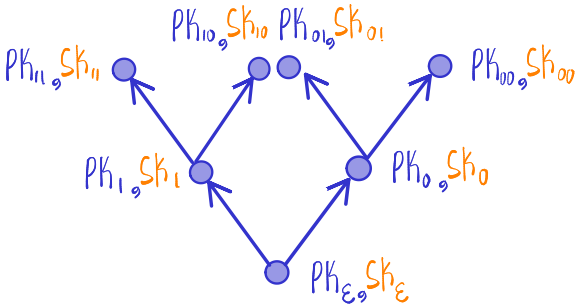
- Introduced digital signatures: public-key analogue of MAC
- Theoretical constructions of DS
  - Lamport's one-time DS
  - Tree-based many-time (stateful) DS from one-time DS
- Lectures 13 and 15(?): efficient DS in "random-oracle model"
  - From trapdoor OWF via hash-then-invert
  - Via Fiat-Shamir transform (e.g., Schnorr)



- Takeaways:
  - Constructive:
    - Bottom up constructive approach
    - Tree-based "bootstrapping" construction

To Recap Today's Lecture

- Introduced digital signatures: public-key analogue of MAC
- Theoretical constructions of DS
  - Lamport's one-time DS
  - Tree-based many-time (stateful) DS from one-time DS
- Lectures 13 and 15(?): efficient DS in "random-oracle model"
  - From trapdoor OWF via hash-then-invert
  - Via Fiat-Shamir transform (e.g., Schnorr)



- Takeaways:
  - Constructive:
    - Bottom up constructive approach
    - Tree-based "bootstrapping" construction
  - Proof techniques: "Plug and pray"

$PK^* =$

y_{00}	y_{10}	y_{20}	y_{30}
y_{01}	y^*	y_{21}	y_{31}

$i^* \leftarrow [\ell]$ $\quad b^* \leftarrow \{0,1\}$

Next Lecture

Exercise 7 (Domain Extension)

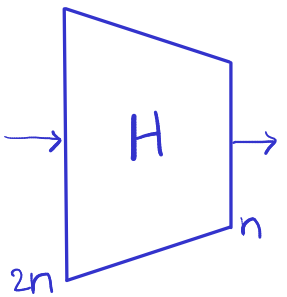
Given a compressing function $\mathsf{H} : \{0,1\}^{2\ell} \to \{0,1\}^{\ell}$*, construct a one-time DS for arbitrary-length messages. What are the properties you need from* H *to ensure that the one-time DS is secure?*

Next Lecture

Exercise 7 (Domain Extension)

Given a compressing function $\mathsf{H} : \{0,1\}^{2\ell} \to \{0,1\}^{\ell}$*, construct a one-time DS for arbitrary-length messages. What are the properties you need from* H *to ensure that the one-time DS is secure?*

- New cryptographic primitive: *hash function*
- Properties of hash function
  - Preimage resistance
  - Collision resistance

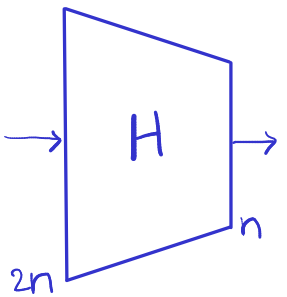


Next Lecture

Exercise 7 (Domain Extension)

Given a compressing function $\mathsf{H} : \{0,1\}^{2\ell} \to \{0,1\}^{\ell}$*, construct a one-time DS for arbitrary-length messages. What are the properties you need from* H *to ensure that the one-time DS is secure?*

- New cryptographic primitive: *hash function*
- Properties of hash function
  - Preimage resistance
  - Collision resistance



- Domain extension for hash function
  - Merkle-Damgård construction
  - Merkle trees

References

1. Digital signature and its security models were formally studied in [GMR88]
2. Lamport's OTS is from [Lam79]
3. The stateful many-time signature is from [KL14], and is in spirit with Merkle's signatures [Mer90]


Mihir Bellare and Phillip Rogaway.
Random oracles are practical: A paradigm for designing efficient protocols.
In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.


Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest.
A digital signature scheme secure against adaptive chosen-message attacks.
SIAM J. Comput., 17(2):281–308, 1988.


Jonathan Katz and Yehuda Lindell.
Introduction to Modern Cryptography (3rd ed.).
Chapman and Hall/CRC, 2014.


Leslie Lamport.
Constructing digital signatures from a one-way function.
Technical report, 1979.


Ralph C. Merkle.
A certified digital signature.
In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 218–238. Springer, Heidelberg, August 1990.