Achieving optimal anonymity in transferable e-cash with a judge

Olivier Blazy[1] Sébastien Canard[2] *Georg Fuchsbauer*[3]
Aline Gouget[4] Hervé Sibert[5] Jacques Traoré[2]

Africacrypt 2011

[1] École Normale Supérieure
[2] Orange Labs
[3] University of Bristol
[4] Gemalto
[5] ST-Ericsson

Outline of this talk

1. Electronic cash
2. Commuting signatures
3. Our instantiation

Electronic Cash

should have the same properties as physical cash

- Unforgeability
- Anonymity
- (Transferability)

Electronic Cash

should have the same properties as physical cash

- Unforgeability $\Rightarrow$ Bank *signs* serial number of coin
- Anonymity $\Rightarrow$ Bank makes *blind* signature
- (Transferability)

Electronic Cash

should have the same properties as physical cash

- Unforgeability $\Rightarrow$ Bank *signs* serial number of coin
- Anonymity $\Rightarrow$ Bank makes *blind* signature
- (Transferability)

Digital signatures are a digital equivalent of hand-written signatures

User produces key pair

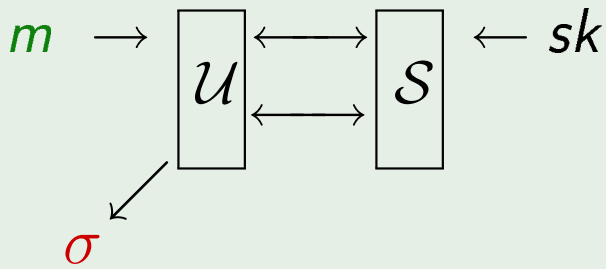
- uses signing key to produce signatures
- publishes verification key to verify signatures

Electronic Cash

should have the same properties as physical cash

- Unforgeability $\Rightarrow$ Bank *signs* serial number of coin
- Anonymity $\Rightarrow$ Bank makes *blind* signature
- (Transferability)

A **blind signature scheme** allows a *user* $\mathcal{U}$ to obtain a signature on a message hidden from the *signer* $\mathcal{S}$



Electronic Cash

should have the same properties as physical cash

- Unforgeability $\Rightarrow$ Bank *signs* serial number of coin
- Anonymity $\Rightarrow$ Bank makes *blind* signature
- (Transferability)

Double-spending

Unlike physical money, data can be copied

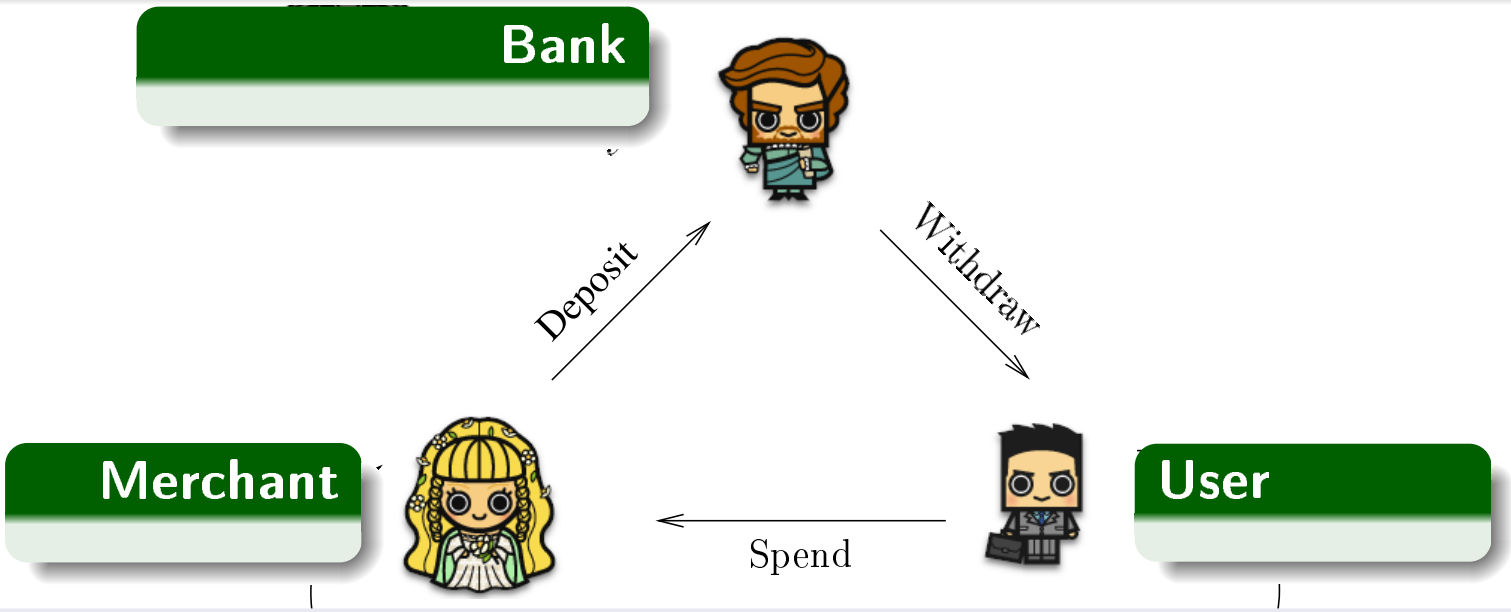
$\Rightarrow$ need mechanisms to trace double-spenders

Protocol

- Withdrawal: A user withdraws a coin c from the bank
- Spending: The user spends the coin with a merchant
- Deposit: The merchant deposits the coin at the bank

Protocol

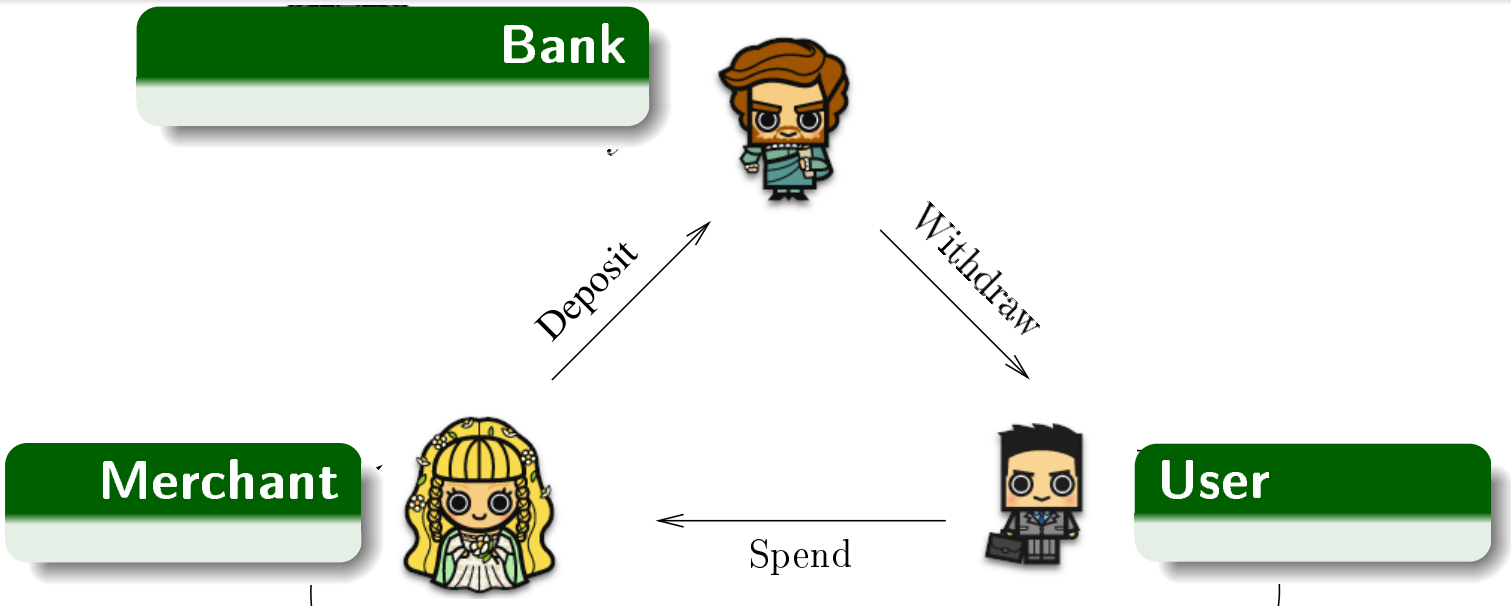
- Withdrawal: A user withdraws a coin c from the bank
- Spending: The user spends the coin with a merchant
- Deposit: The merchant deposits the coin at the bank



Classical protocol

Protocol

- Withdrawal: A user withdraws a coin c from the bank
- Spending: The user spends the coin with a merchant
- Deposit: The merchant deposits the coin at the bank



Transferable e-cash

A user can transfer a coin (offline) to other users before spending it

Anonymity in transferable e-cash

- Weak anonymity — Spender & receiver not identifiable
- Strong anonymity — User anonymous w.r.t. the bank

Anonymity in transferable e-cash

- Weak anonymity — Spender & receiver not identifiable
- Strong anonymity — User anonymous w.r.t. the bank

Stronger anonymity in transferable e-cash [Canard-Gouget'08]

[FA] Observe then receive:
- Adversary can impersonate the bank,
- cannot link a coin he receives to a previously (passively) observed transfer

[PA] Perfect anonymity:
- Adversary can impersonate the bank,
- the adversary cannot tell whether he has already owned a coin he receives

Anonymity in transferable e-cash

- Weak anonymity — Spender & receiver not identifiable
- Strong anonymity — User anonymous w.r.t. the bank

Stronger anonymity in transferable e-cash [Canard-Gouget'08]

[FA] Observe then receive:
- Adversary can impersonate the bank,
- cannot link a coin he receives to a previously (passively) observed transfer

[PA] Perfect anonymity:
- the adversary ca
a coin he receiv

Impossible to achieve

Bank must link coins to detect double-spending

Anonymity in transferable e-cash

- Weak anonymity — Spender & receiver not identifiable
- Strong anonymity — User anonymous w.r.t. the bank

Stronger anonymity in transferable e-cash [Canard-Gouget'08]

[FA] Observe then receive:
- Adversary can impersonate the bank,
- cannot link a coin he receives to a previously (passively) observed transfer

[PA_1] Spend then observe:
- Adversary can impersonate the bank,
- cannot link a (passively) observed coin to a coin he has already owned

[PA_2] Spend then receive:
- The bank is *trusted*;
- the adversary cannot tell whether he has already owned a coin he receives

Anonymity in transferable e-cash

- Weak anonymity — Spender & receiver not identifiable
- Strong anonymity — User anonymous w.r.t. the bank

Stronger anonymity in transferable e-cash [Canard-Gouget'08]

[FA] Observe then receive:
- Adversary can impersonate the bank,
- cannot link a coin he receives to a previously (passively) observed transfer

[PA_1] Spend then observe:
- Adversary can impersonate the bank,
- cannot link a (passively) observed coin to a coin he has already owned

[PA_2] Spend then receive:
- The bank is *trusted*;
- the adversary cannot tell whether he has already owned a coin he receives

Further security notions

Unforgeability

- No coalition of users can spend more coins than they withdrew (without being detected)

Further security notions

Unforgeability

- No coalition of users can spend more coins than they withdrew (without being detected)

Identification of double-spenders

- No coalition of users can spend a coin twice without revealing one of their identities

Further security notions

Unforgeability

- No coalition of users can spend more coins than they withdrew (without being detected)

Identification of double-spenders

- No coalition of users can spend a coin twice without revealing one of their identities

Exculpability

- The bank—even when colluding with malicious users—cannot wrongfully accuse an honest users of double-spending

Ideas for our instantiation

Need to hide coin from previous owner [PA_2]

- Instead of signature $\Rightarrow$ Proof of knowledge of signature
  $\Rightarrow$ Proof of knowledge of proof ...
  $\Rightarrow$ ...

Ideas for our instantiation

Need to hide coin from previous owner [PA_2]

- Instead of signature $\Rightarrow$ Proof of knowledge of signature
  $\Rightarrow$ Proof of knowledge of proof ...
  $\Rightarrow$ ...

Exponential growth of coin

Ideas for our instantiation

Need to hide coin from previous owner [PA_2]

- Instead of signature $\Rightarrow$ Proof of knowledge of signature
  $\Rightarrow$ Proof of knowledge of proof ...
  $\Rightarrow$ ... Exponential growth of coin
- Randomisable verifiably encrypted signatures

Ideas for our instantiation

Need to hide coin from previous owner [PA_2]

- Instead of signature $\Rightarrow$ Proof of knowledge of signature
  $\Rightarrow$ Proof of knowledge of proof . . .
  $\Rightarrow$ . . . Exponential growth of coin
- Randomisable verifiably encrypted signatures

Users have to be accountable for double-spending

- Dynamic serial number

  When receiving a coin, user picks new part of serial number

Ideas for our instantiation

Need to hide coin from previous owner [PA_2]

- Instead of signature $\Rightarrow$ Proof of knowledge of signature
  $\Rightarrow$ Proof of knowledge of proof . . .
  $\Rightarrow$ . . .
  Exponential growth of coin
- Randomisable verifiably encrypted signatures

Users have to be accountable for double-spending

- Dynamic serial number

  When receiving a coin, user picks new part of serial number
  $\Rightarrow$ Double-spender detection

Ideas for our instantiation

Need to hide coin from previous owner [PA_2]

- Instead of signature $\Rightarrow$ Proof of knowledge of signature
  $\Rightarrow$ Proof of knowledge of proof . . .
  $\Rightarrow$ . . .
  Exponential growth of coin
- Randomisable verifiably encrypted signatures

Users have to be accountable for double-spending

- Dynamic serial number
  When receiving a coin, user picks new part of serial number
  $\Rightarrow$ Double-spender detection
- Spender commits to transfer
  Sender signs part of the serial number

Ideas for our instantiation

Need to hide coin from previous owner [PA_2]

- Instead of signature $\Rightarrow$ Proof of knowledge of signature
  $\Rightarrow$ Proof of knowledge of proof . . .
  $\Rightarrow$ . . . Exponential growth of coin
- Randomisable verifiably encrypted signatures

Users have to be accountable for double-spending

- Dynamic serial number
  When receiving a coin, user picks new part of serial number
  $\Rightarrow$ Double-spender detection
- Spender commits to transfer
  Sender signs part of the serial number
  $\Rightarrow$ Exculpability

Ideas for our instantiation

Need to hide coin from previous owner [PA_2]

- Instead of signature $\Rightarrow$ Proof of knowledge of signature
  $\Rightarrow$ Proof of knowledge of proof . . .
  $\Rightarrow$ . . .

  Exponential growth of coin
- Randomisable verifiably encrypted signatures

Users have to be accountable for double-spending

- Dynamic serial number

  When receiving a coin, user picks new part of serial number

  $\Rightarrow$ Double-spender detection

  $\Rightarrow$ need to encrypt serial number
- Spender commits to transfer

  Sender signs part of the serial number

  $\Rightarrow$ Exculpability

Ideas for our instantiation

Need to hide coin from previous owner [PA_2]

- Instead of signature $\Rightarrow$ Proof of knowledge of signature
  $\Rightarrow$ Proof of knowledge of proof . . .
  $\Rightarrow$ . . .
  Exponential growth of coin
- Randomisable verifiably encrypted signatures

Users have to be accountable for double-spending

- Dynamic serial number
  When receiving a coin, user picks new part of serial number
  $\Rightarrow$ Double-spender detection
  $\Rightarrow$ need to encrypt serial number
- Spender commits to transfer
  Sender signs part of the serial number
  $\Rightarrow$ Exculpability
  $\Rightarrow$ need to sign encrypted value

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Verifiable encryption

$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} & \text{Verification: } vk, M, \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} & \text{Verification: } vk, \boxed{M}, \Sigma, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi & \text{Verification: } vk, \boxed{M}, \boxed{\Sigma}, \pi \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} & \text{Verification: } \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Randomisable Verifiable encryption

$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} \longrightarrow \boxed{M}, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Verifiable encryption

$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} & \text{Verification: } vk, M, \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} & \text{Verification: } vk, \boxed{M}, \Sigma, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi & \text{Verification: } vk, \boxed{M}, \boxed{\Sigma}, \pi \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} & \text{Verification: } \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Verifiable encryption

$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} & \text{Verification: } vk, M, \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} & \text{Verification: } vk, \boxed{M}, \Sigma, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi & \text{Verification: } vk, \boxed{M}, \boxed{\Sigma}, \pi \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} & \text{Verification: } \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

- Commuting signature and verifiable encryption

Proof adaptation:

$$\left.\begin{matrix} \widetilde{\pi} \\ \bar{\pi} \end{matrix}\right\} \longleftrightarrow \pi \longleftrightarrow \widehat{\pi}$$

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Verifiable encryption

$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} & \text{Verification: } vk, M, \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} & \text{Verification: } vk, \boxed{M}, \Sigma, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi & \text{Verification: } vk, \boxed{M}, \boxed{\Sigma}, \pi \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} & \text{Verification: } \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

- Commuting signature and verifiable encryption

Proof adaptation:

$$\left.\begin{matrix} \widetilde{\pi} \\ \bar{\pi} \end{matrix}\right\} \longleftrightarrow \pi \longleftrightarrow \widehat{\pi}$$

Sign M given $\boxed{M}$: $\boxed{M} \xrightarrow{sk} \Sigma$

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Verifiable encryption

$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} & \text{Verification: } vk, M, \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} & \text{Verification: } vk, \boxed{M}, \Sigma, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi & \text{Verification: } vk, \boxed{M}, \boxed{\Sigma}, \pi \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} & \text{Verification: } \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

- Commuting signature and verifiable encryption

Proof adaptation:

$$\left.\begin{matrix} \widetilde{\pi} \\ \bar{\pi} \end{matrix}\right\} \longleftrightarrow \pi \longleftrightarrow \widehat{\pi}$$

Sign M given $\boxed{M}$: $\boxed{M} \xrightarrow{sk} \Sigma$ (crossed out)

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Verifiable encryption

$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} & \text{Verification: } vk, M, \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} & \text{Verification: } vk, \boxed{M}, \Sigma, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi & \text{Verification: } vk, \boxed{M}, \boxed{\Sigma}, \pi \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} & \text{Verification: } \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

- Commuting signature and verifiable encryption

Proof adaptation: $\left.\begin{matrix}\widetilde{\pi}\\ \bar{\pi}\end{matrix}\right\} \longleftrightarrow \pi \longleftrightarrow \widehat{\pi}$

Sign M given $\boxed{M}$: $\boxed{M} \xrightarrow{sk} \boxed{\Sigma}, \pi$ Verification: $vk, \boxed{M}, \boxed{\Sigma}, \pi$

Commuting signatures and verifiable encryption I

- Signature $M \xrightarrow{sk} \Sigma$ Verification: vk, M, Σ
- Verifiable encryption

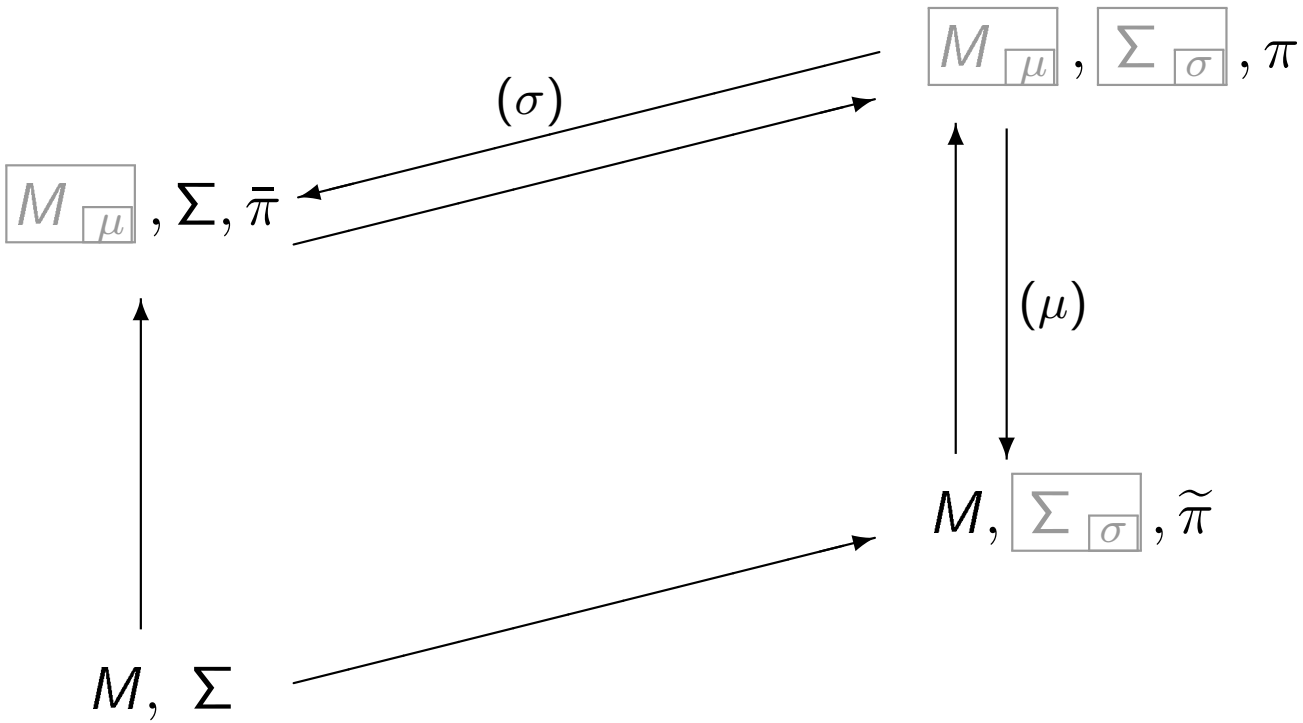
$$vk, M, \Sigma \longrightarrow \begin{cases} \longrightarrow \boxed{\Sigma}, \widetilde{\pi} & \text{Verification: } vk, M, \boxed{\Sigma}, \widetilde{\pi} \\ \longrightarrow \boxed{M}, \bar{\pi} & \text{Verification: } vk, \boxed{M}, \Sigma, \bar{\pi} \\ \longrightarrow \boxed{M}, \boxed{\Sigma}, \pi & \text{Verification: } \boxed{vk, M, \Sigma, \pi} \\ \longrightarrow \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} & \text{Verification: } \boxed{vk}, \boxed{M}, \boxed{\Sigma}, \widehat{\pi} \end{cases}$$

- Commuting signature and verifiable encryption

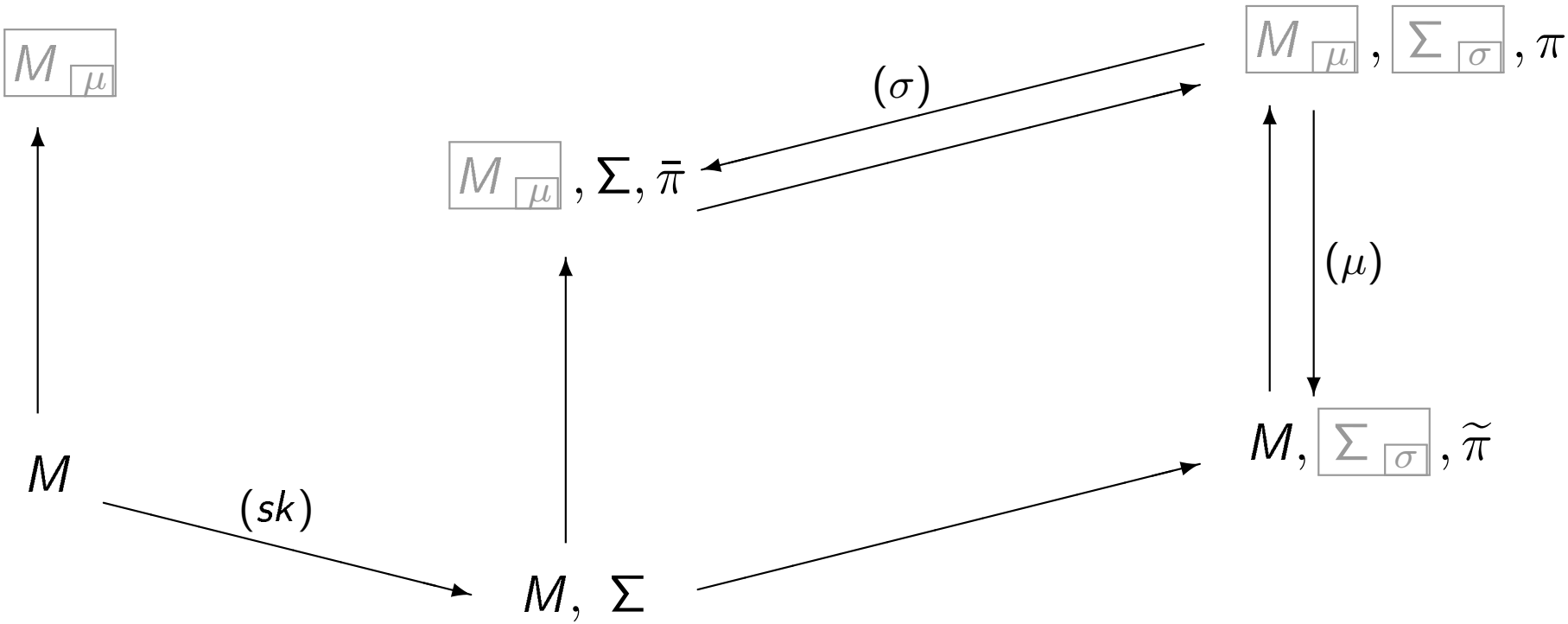
Sign plaintext then encrypt $\iff$ encrypt then sign plaintext

Sign M given $\boxed{M}$: $\boxed{M} \xrightarrow{sk} \boxed{\Sigma}, \pi$ Verification: $\boxed{vk, M, \Sigma, \pi}$

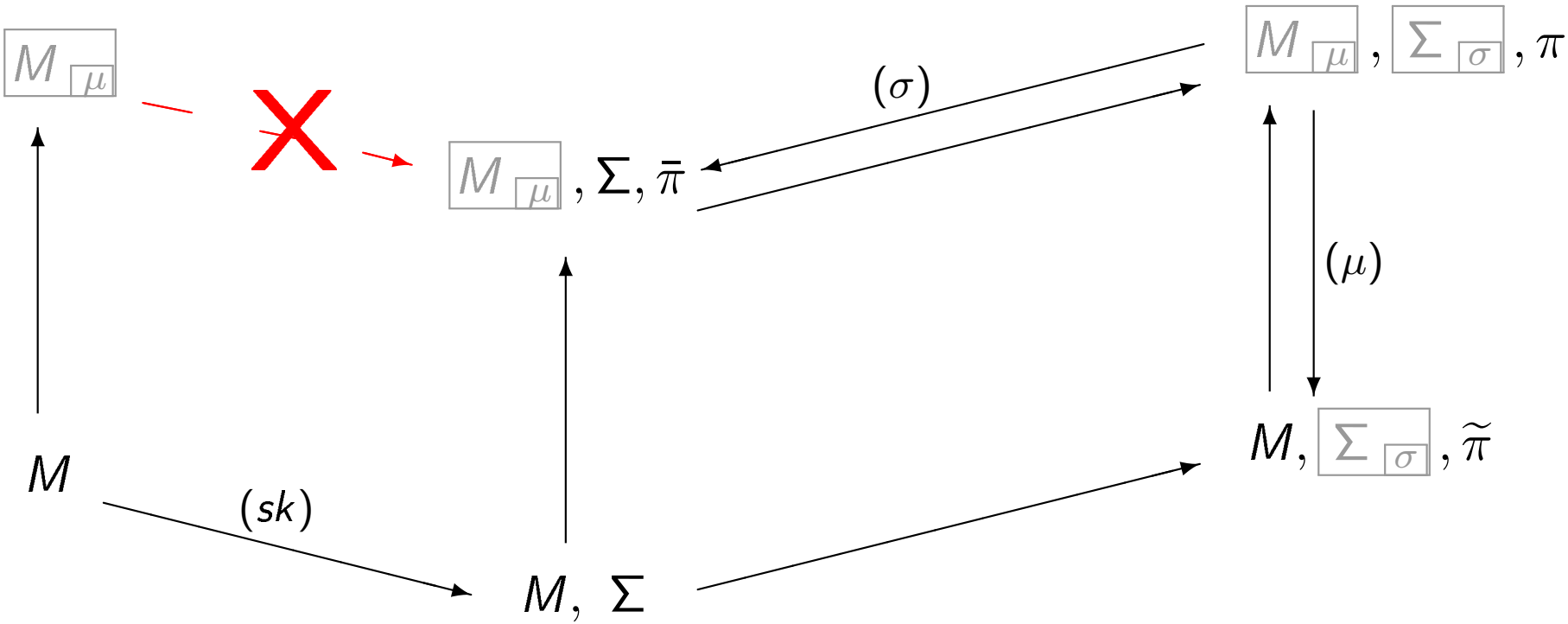
Commuting signatures and verifiable encryption II



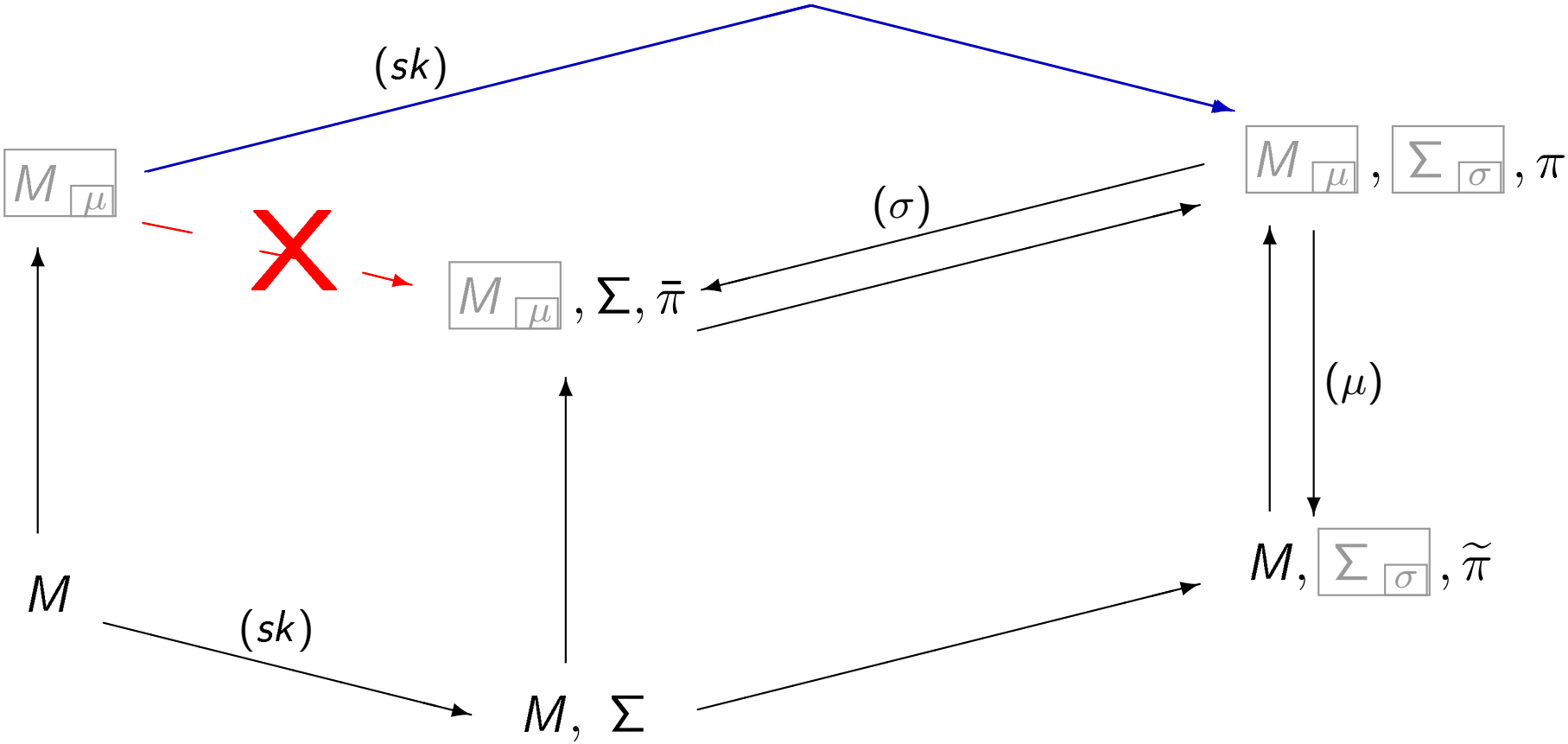
Commuting signatures and verifiable encryption II



Commuting signatures and verifiable encryption II



Commuting signatures and verifiable encryption II



Commuting signatures and verifiable encryption II

Instantiation

- given in [F'11]
- using pairing-friendly groups,
- Groth-Sahai proofs [Groth-Sahai'08]

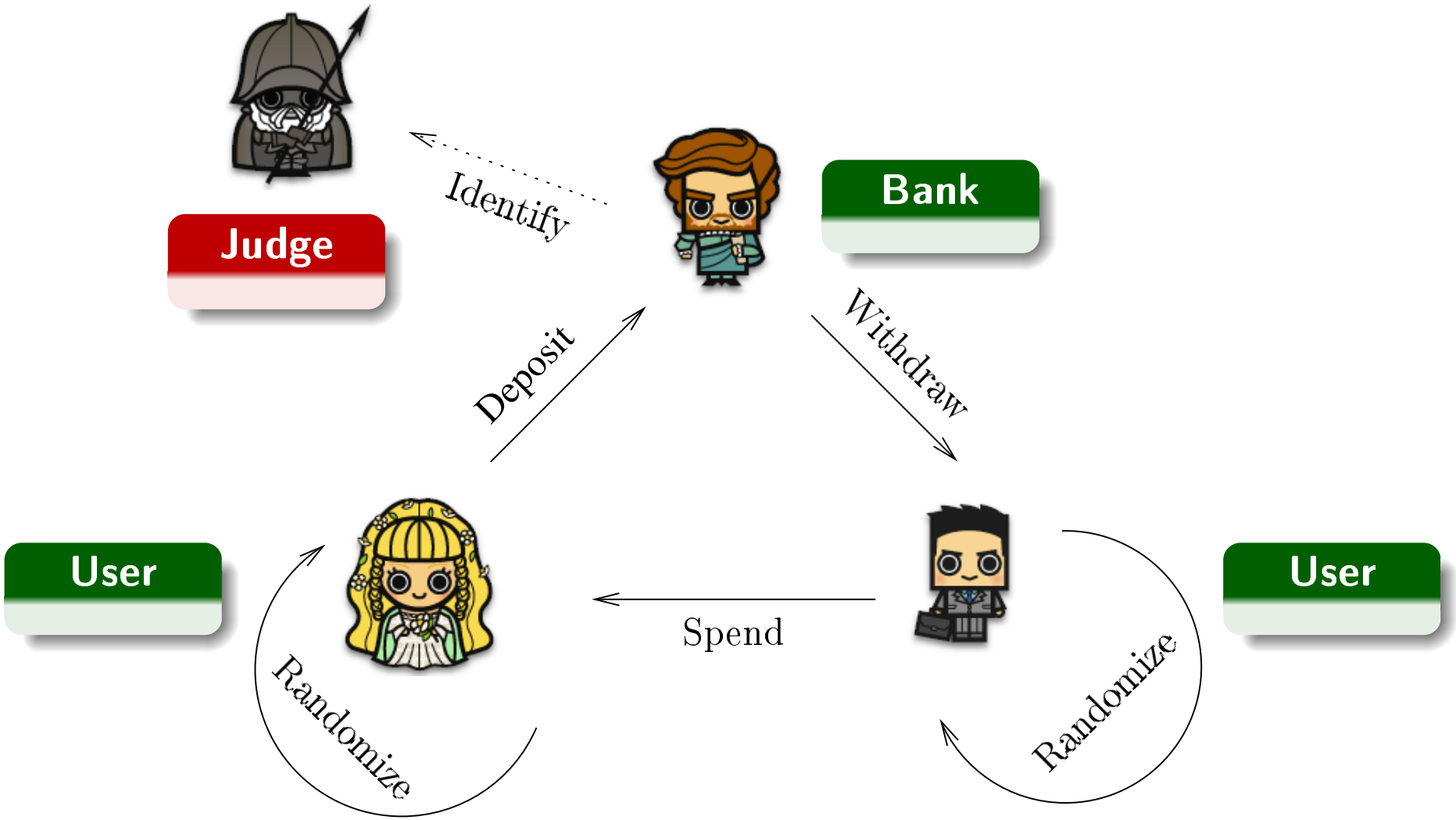
  which have been shown to be randomisable by [Belenkiy-Camenisch-Chase-Kohlweiss-Lysyanskaya-Shacham'09]
- and automorphic signatures [Abe-F-Groth-Haralambiev-Ohkubo'10]

1 Electronic cash

2 Commuting signatures

3 **Our instantiation**

Introducing the judge



Introducing the judge

Bank

$\boxed{n_1}$, $\boxed{pk_1}$

$\boxed{\Sigma_0}$, π_0

Σ_0 is signature on n_0, n_1, pk_1 under pk_B

User1

Judge

Introducing the judge

Bank

$\boxed{n_1}$, $\boxed{pk_1}$

$\boxed{\Sigma_0}$, π_0

Σ_0 is signature on n_0, n_1, pk_1 under pk_B

User1

$\boxed{n_2}$, $\boxed{pk_2}$

$\boxed{coin}$

$\boxed{\Sigma_1}$, π_1

Σ_1 is signature on n_1, n_2, pk_2 under pk_1

User2

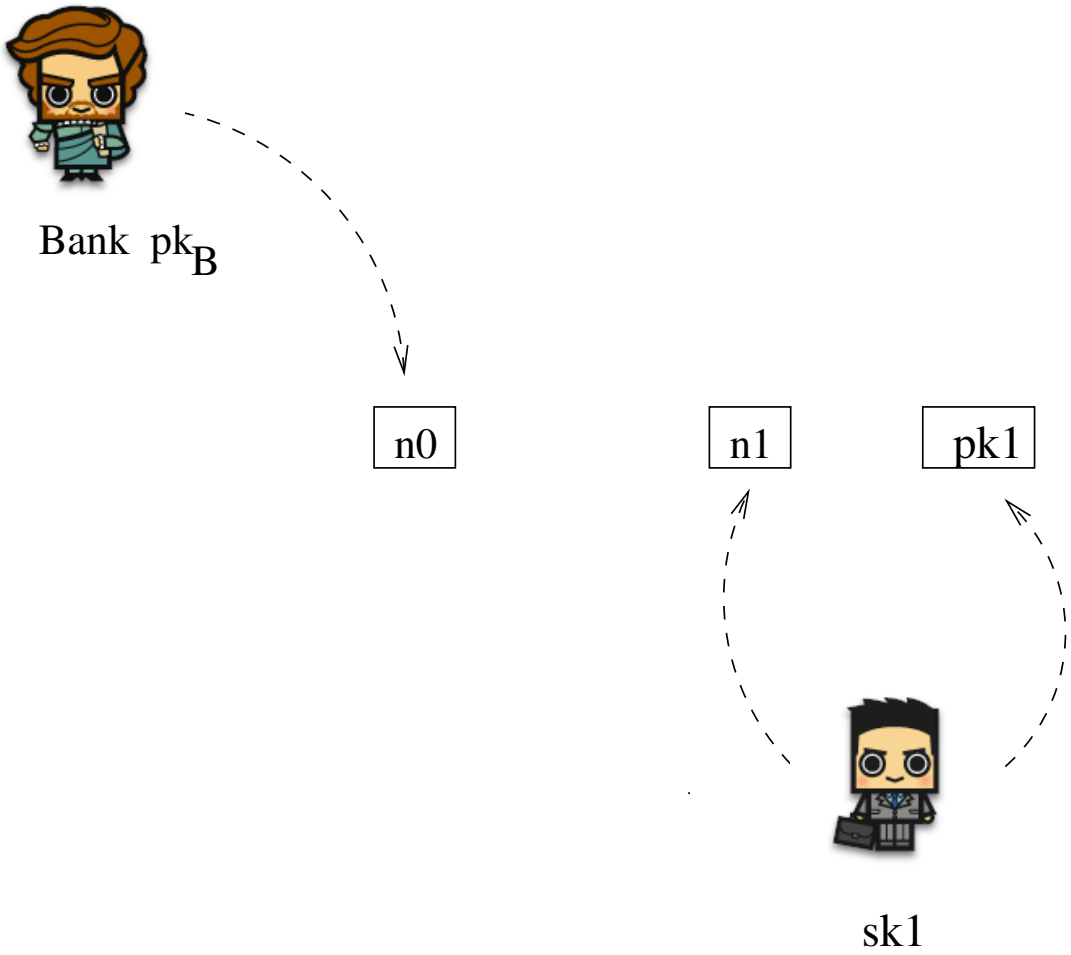
Judge

The form of a coin

Bank pk_B

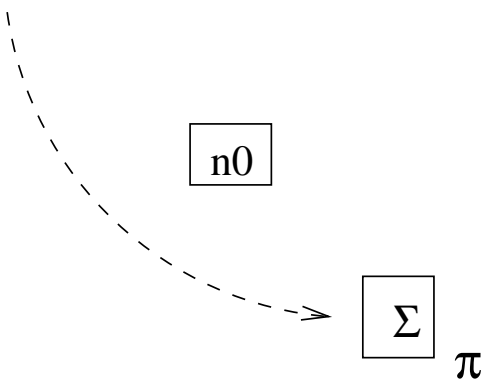
sk1

The form of a coin



The form of a coin

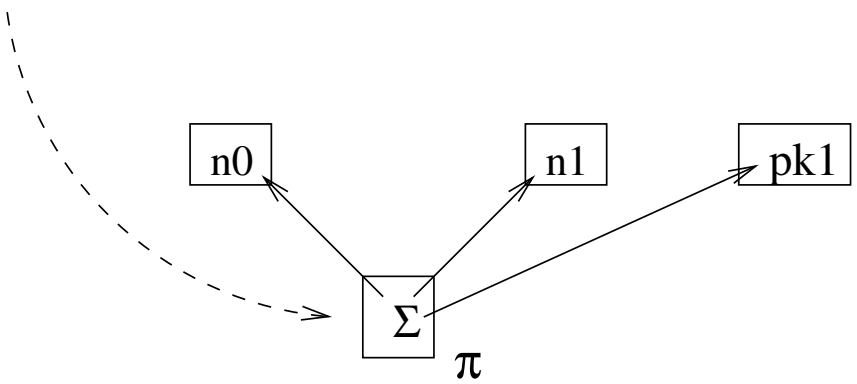
Bank pk_B



sk1

The form of a coin

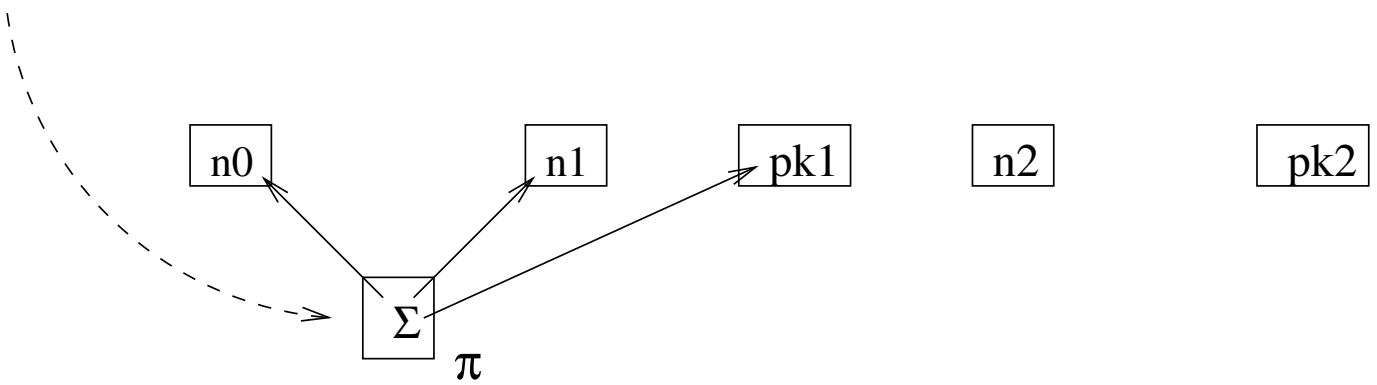
Bank pk_B



sk1

The form of a coin

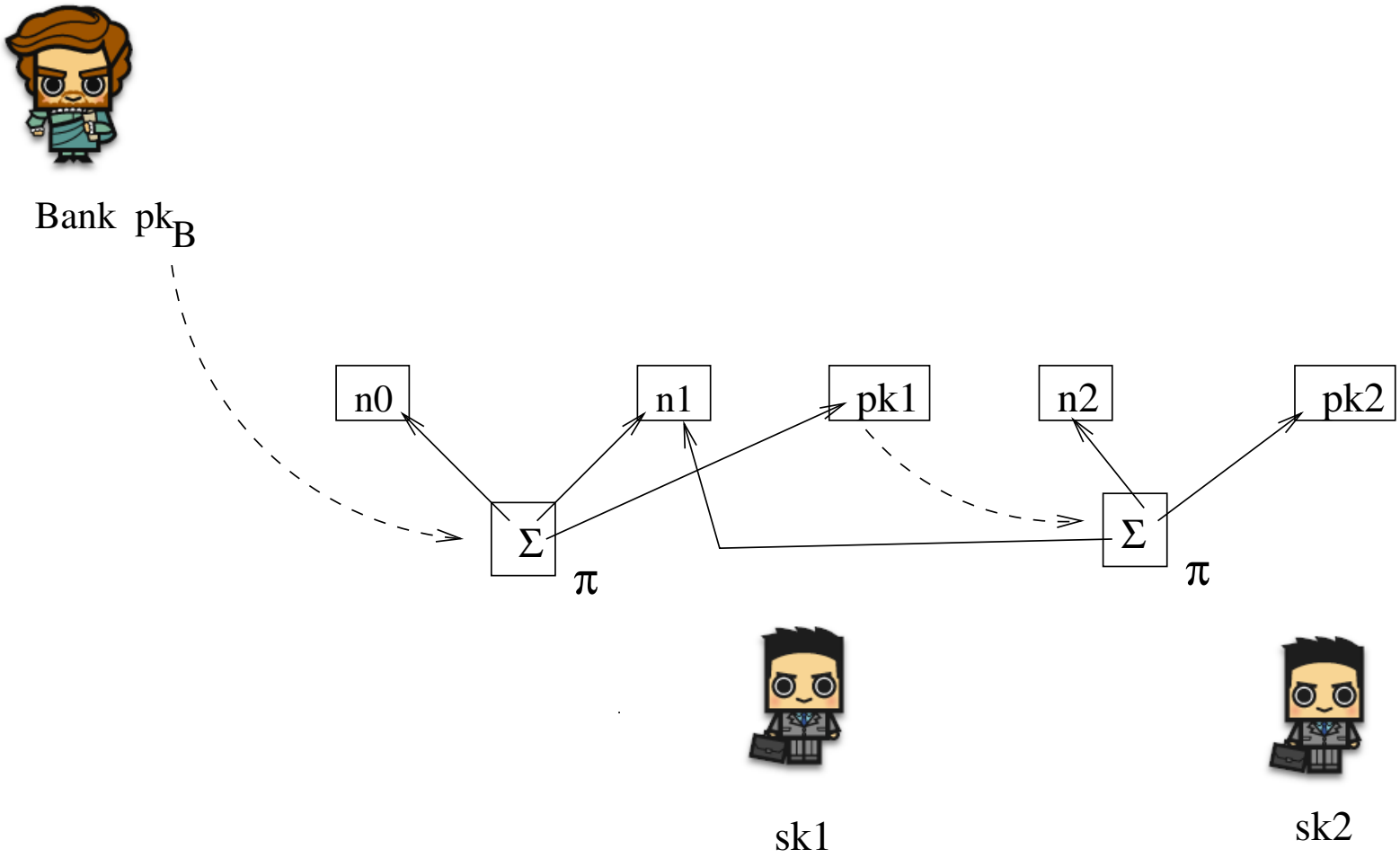
Bank pk_B



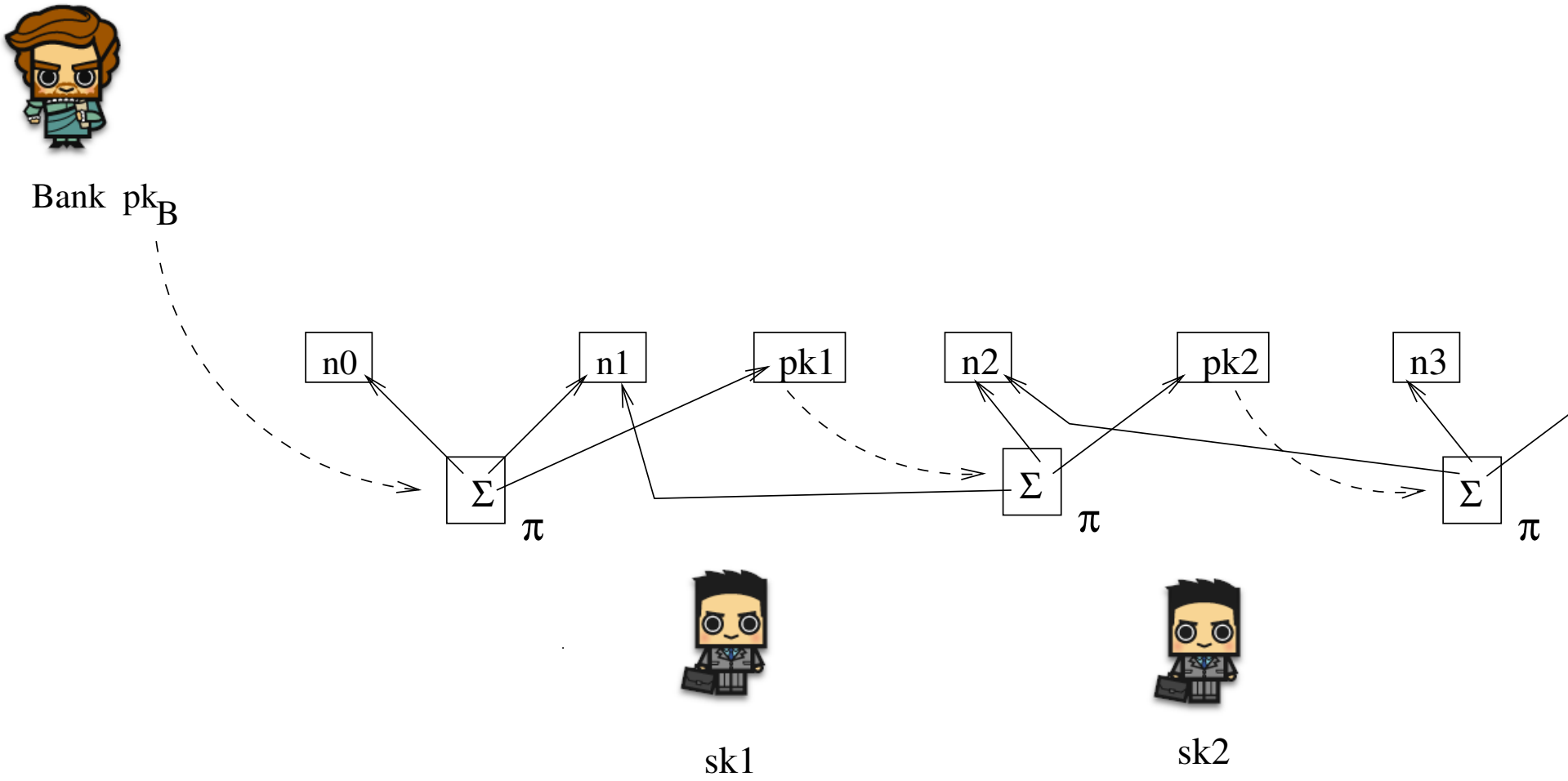
sk1

sk2

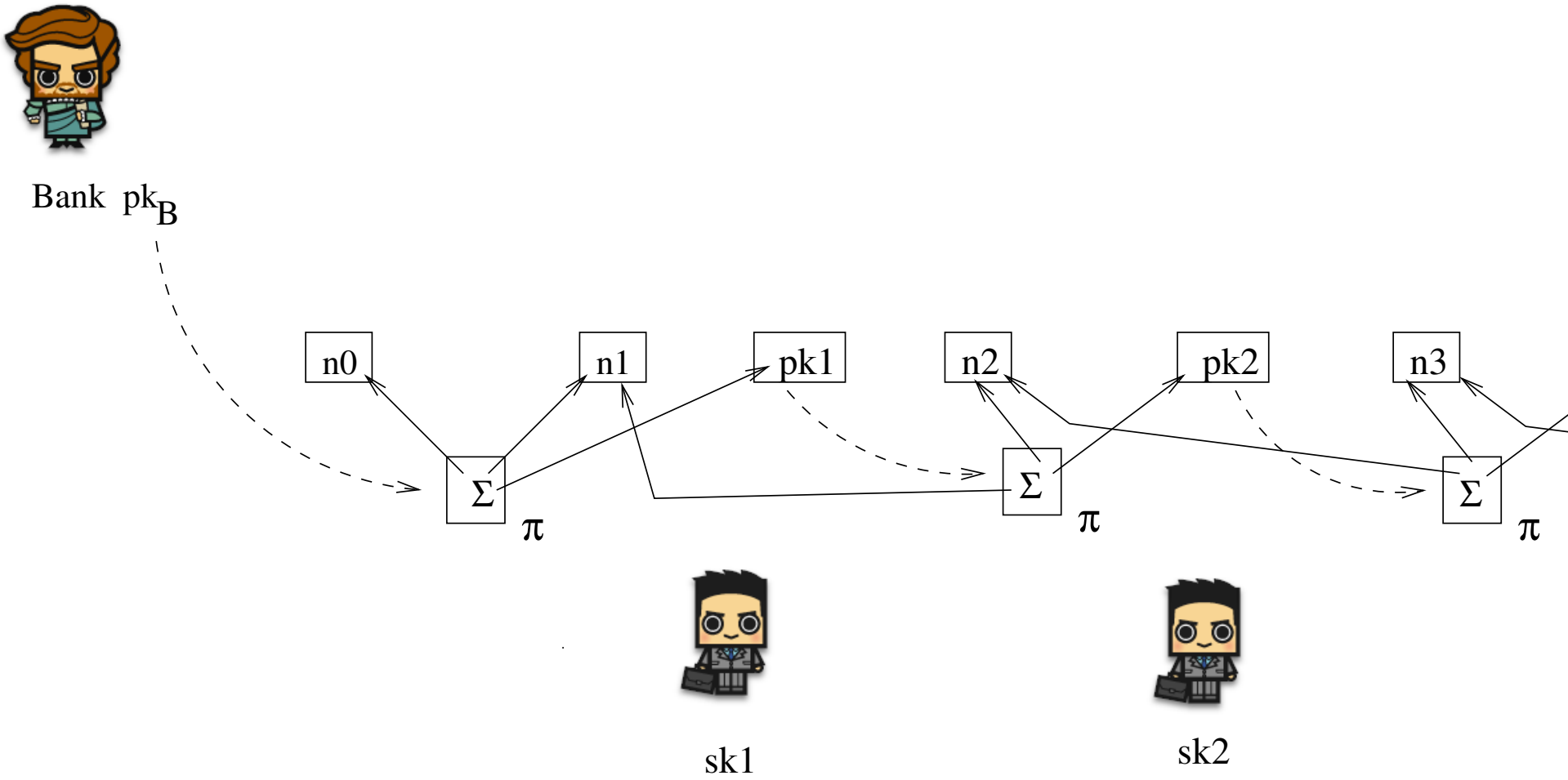
The form of a coin



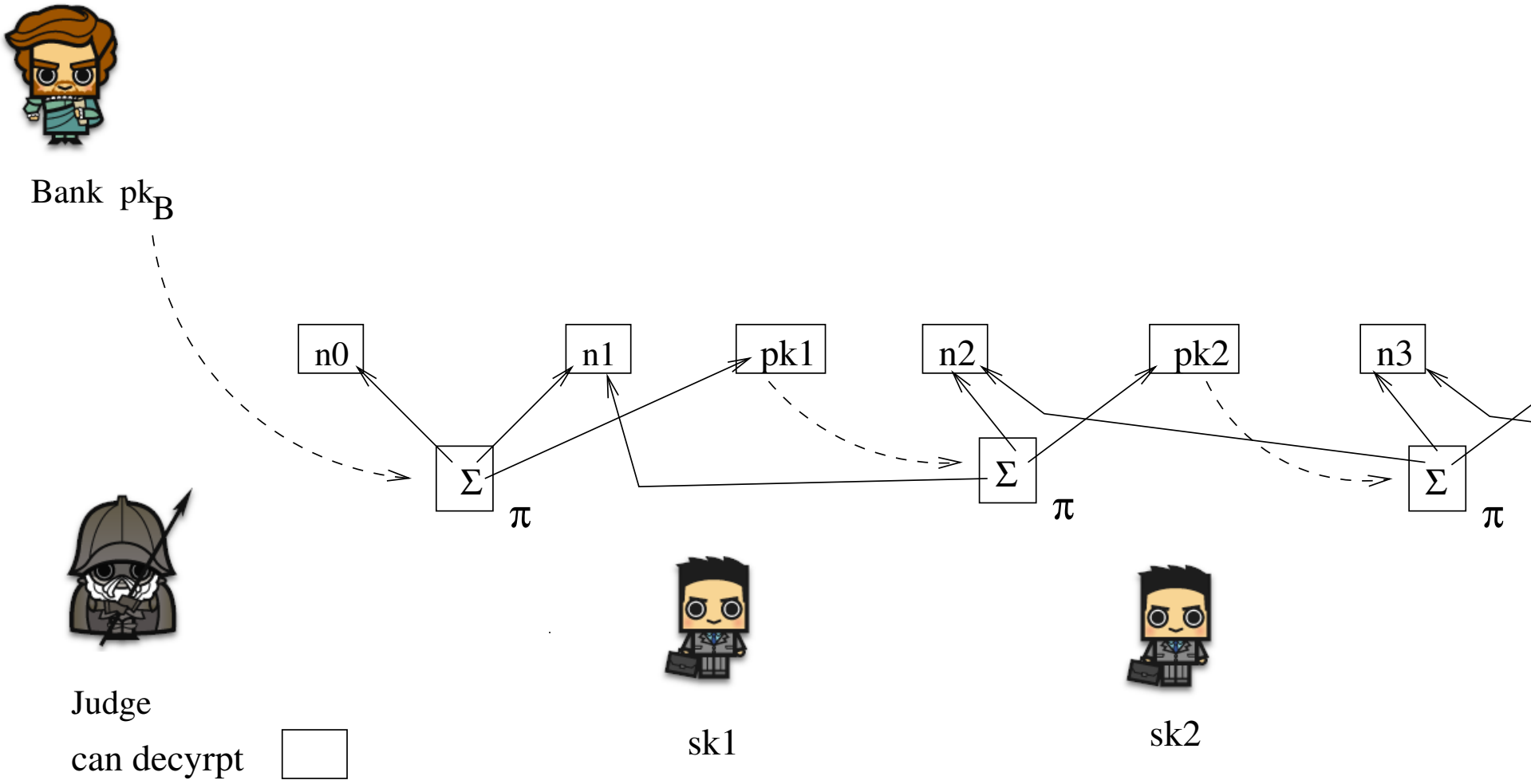
The form of a coin



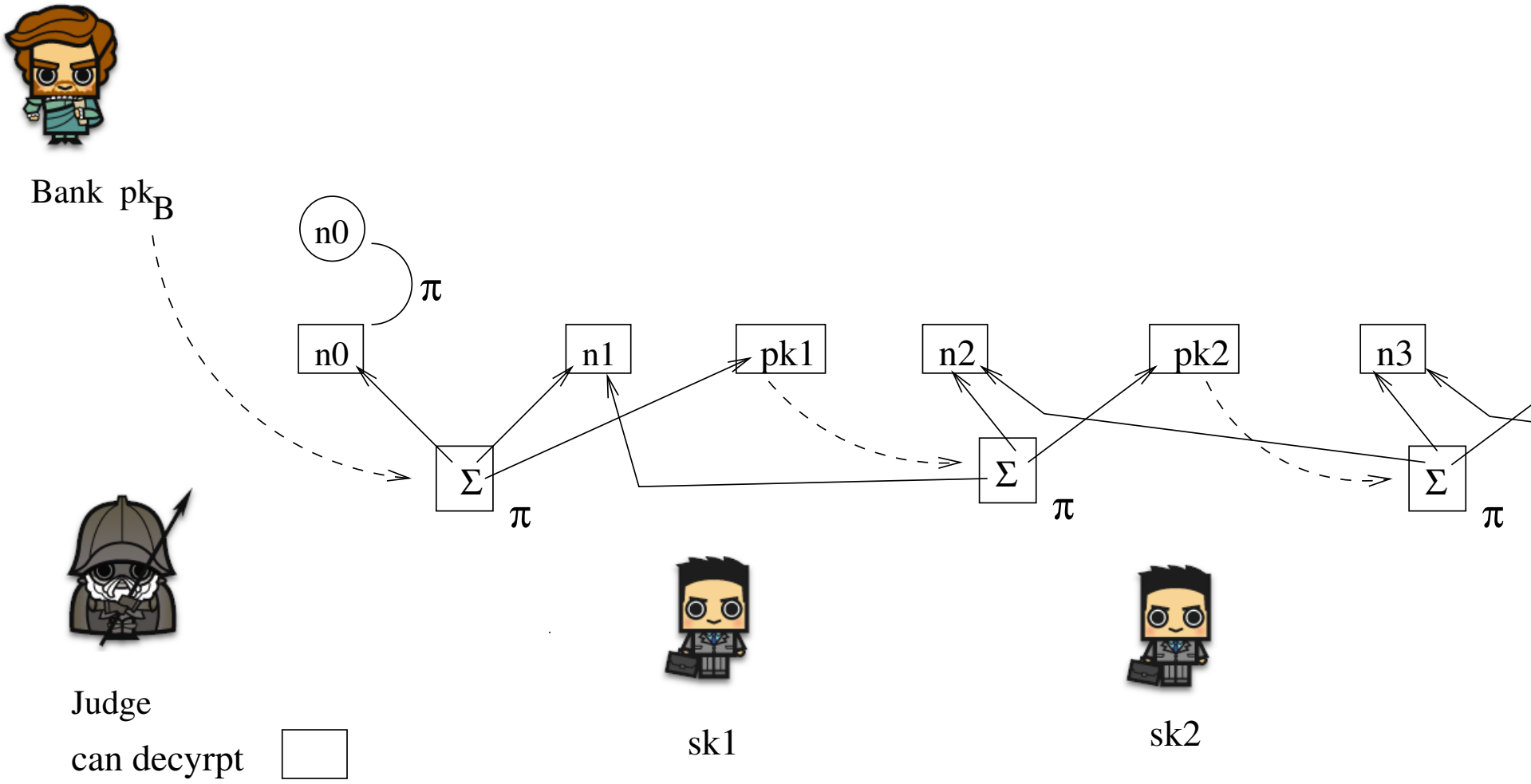
The form of a coin



The form of a coin

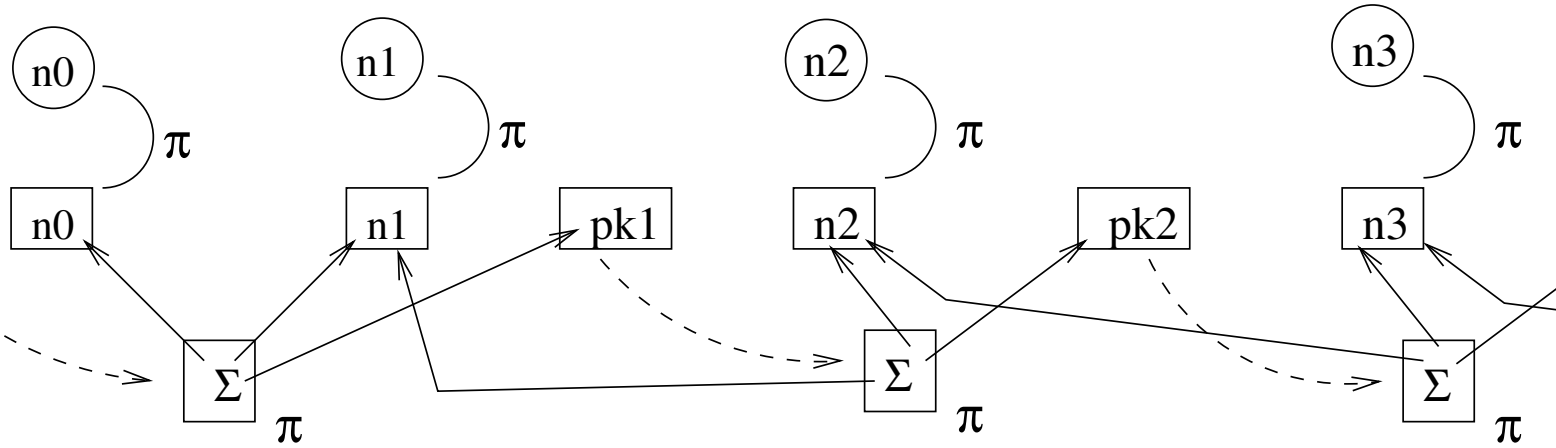


The form of a coin



The form of a coin

Bank pk_B



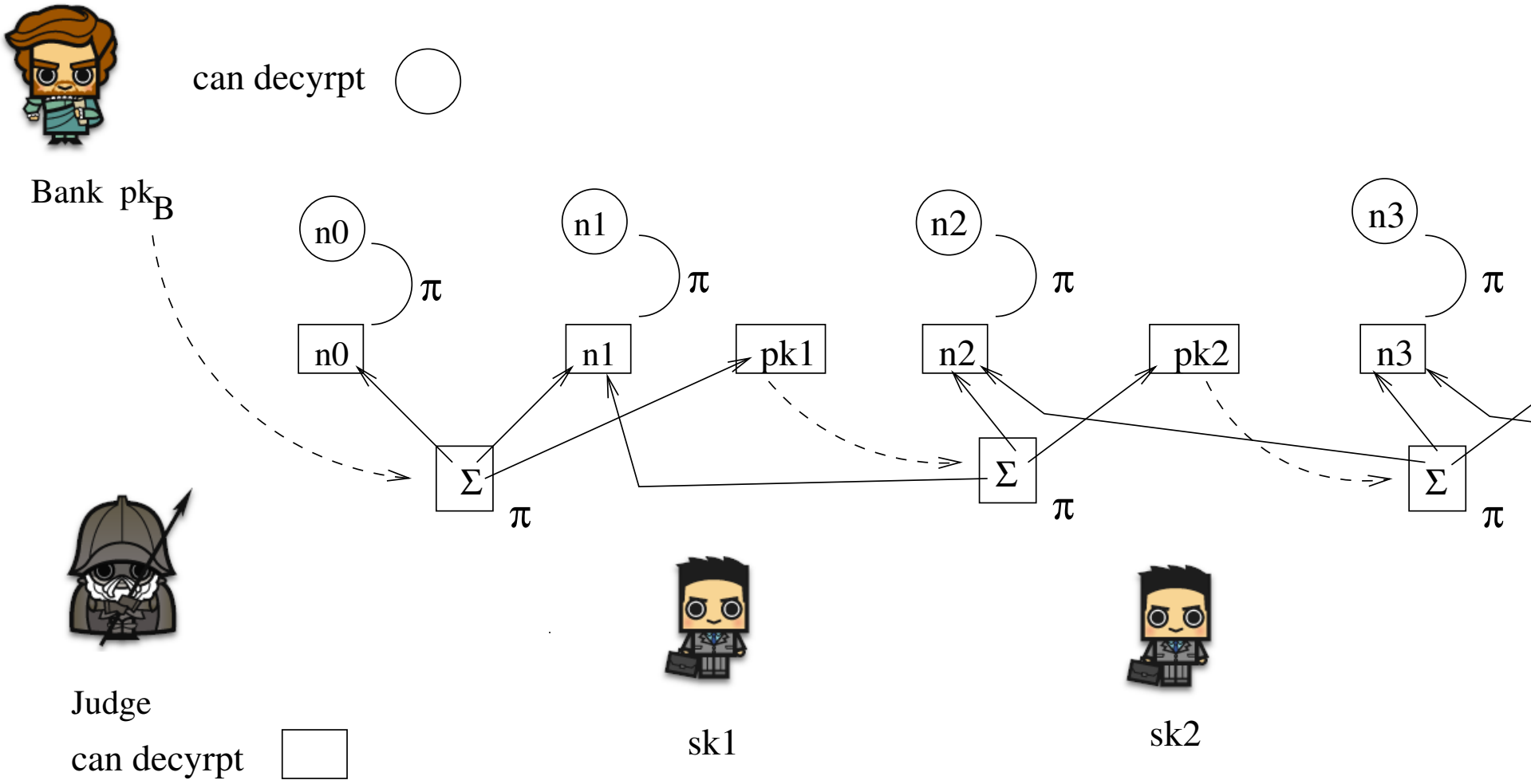
Judge

can decyrpt

sk1

sk2

The form of a coin



Conclusion

By introducing a trusted third party to trace users, we constructed the first *efficient* transferable e-cash scheme achieving all considered security notions

Thank you! ☺