Unlinkable Policy-Compliant Signatures for Compliant and Decentralized Anonymous Payments*

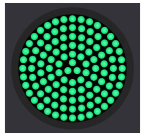
Christian Badertscher[1], Mahdi Sedaghat[2] and Hendrik Waldner[3]

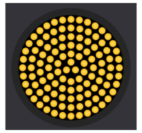
1) Input Output (Switzerland)
2) COSIC, KU Leuven, Belgium
3) University of Maryland, College Park & Max Planck Institute for Security and Privacy

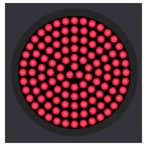
* Link to full-version: https://eprint.iacr.org/2023/1070.pdf.

Outline:

- **Problem Statement**
  - Lack of privacy in the 1st generation of cryptocurrencies.
  - Lack of accountability in the 2nd generation of cryptocurrencies.
- **Accountable Privacy and Existing Solutions**
  - Fine-grained Privacy Balancing
  - Prevention vs Detection
- **UL-PCS**
  - Generic Construction
  - Applications: Accountable Decentralized Anonymous Payment (DAP) systems
  - Benchmarks
- **Open questions and ongoing projects**


Easy: 23 Slides


Semi-hard: 5 slides

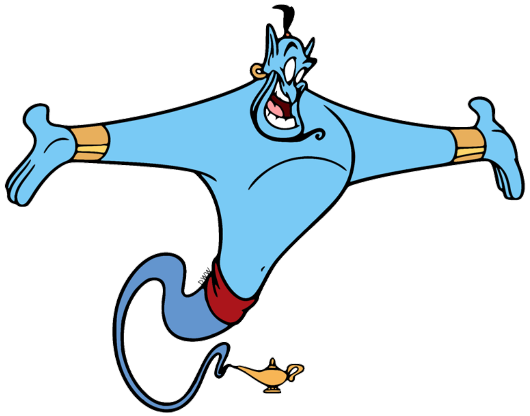

Needs background: 2 slides

Main Actors:

Jasmin

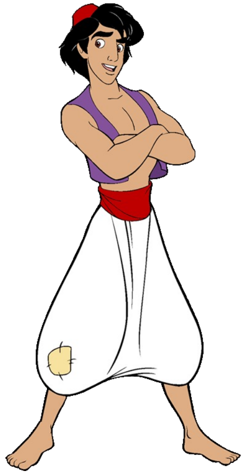
Sender
Prover

Genie



Trusted Central Authority

Aladdin

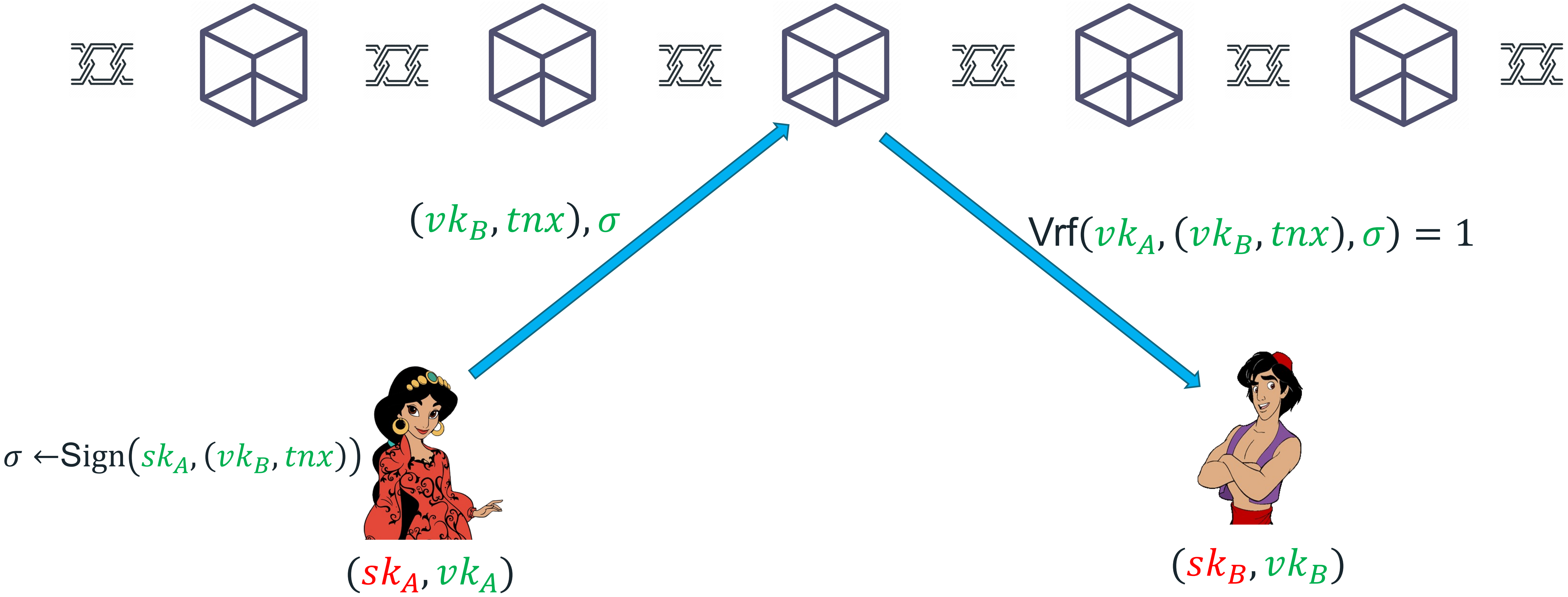


Receiver
Verifier

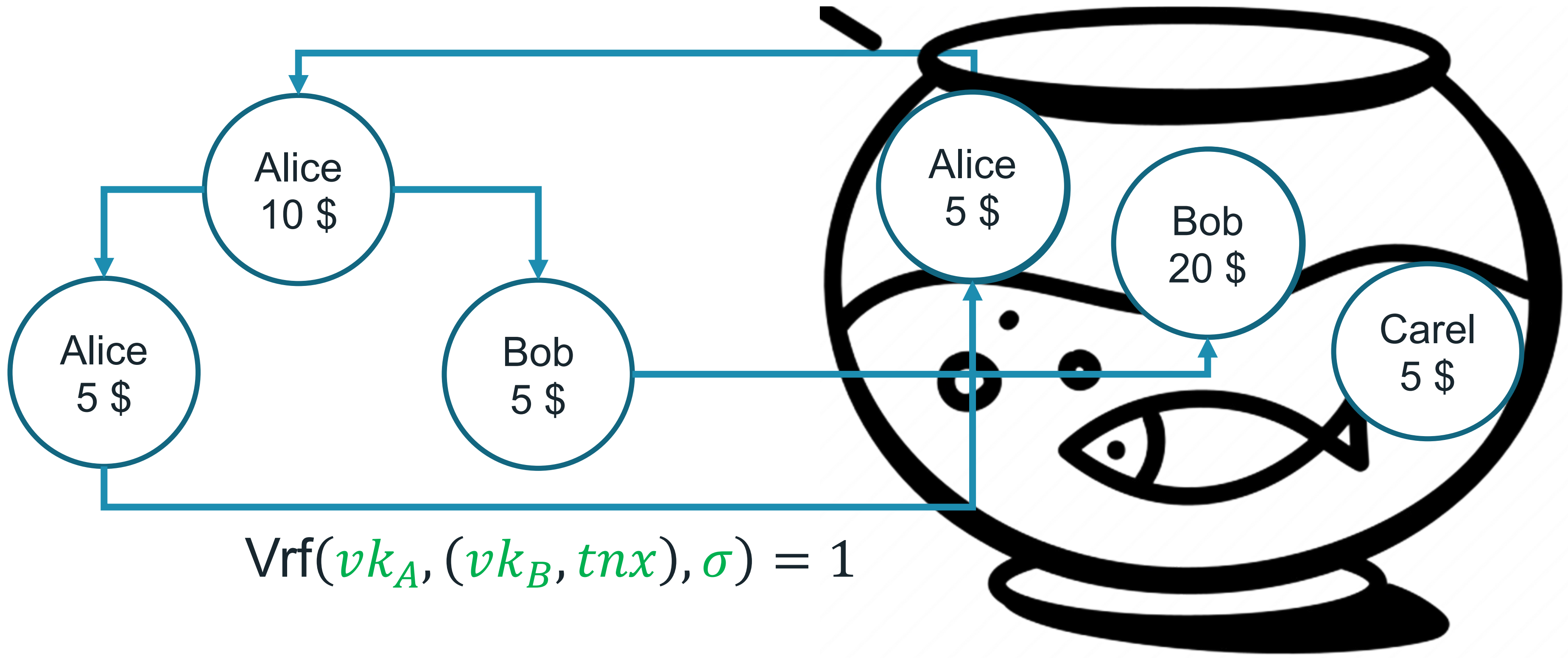
Jafar

Malicious Party

Motivation: UTxO-based cryptocurrencies



Motivation: UTxO-based cryptocurrencies



$\text{Vrf}(vk_A, (vk_B, tnx), \sigma) = 1$

Pseudonymity ≠ Anonymity

The PID of the **payee** and **payer** and the **value** in Bitcoin are publicly available!!

If CUHK pays employee in Bitcoin?! All salaries are visible

Distributed anonymous payments (DAP).

The identity and the values are hidden.

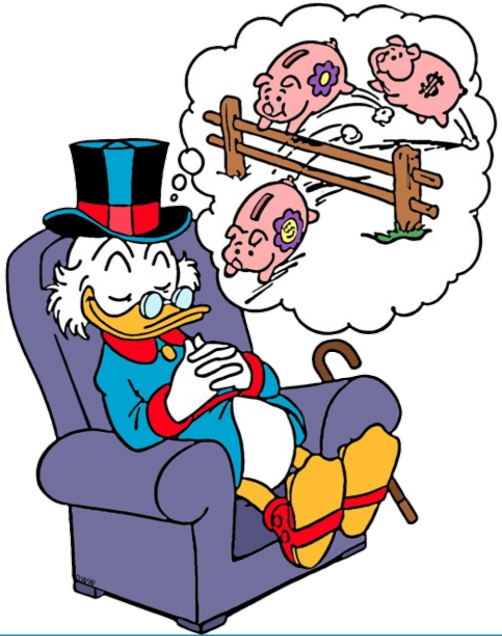
Such cryptocurrencies can be used in an illegal context

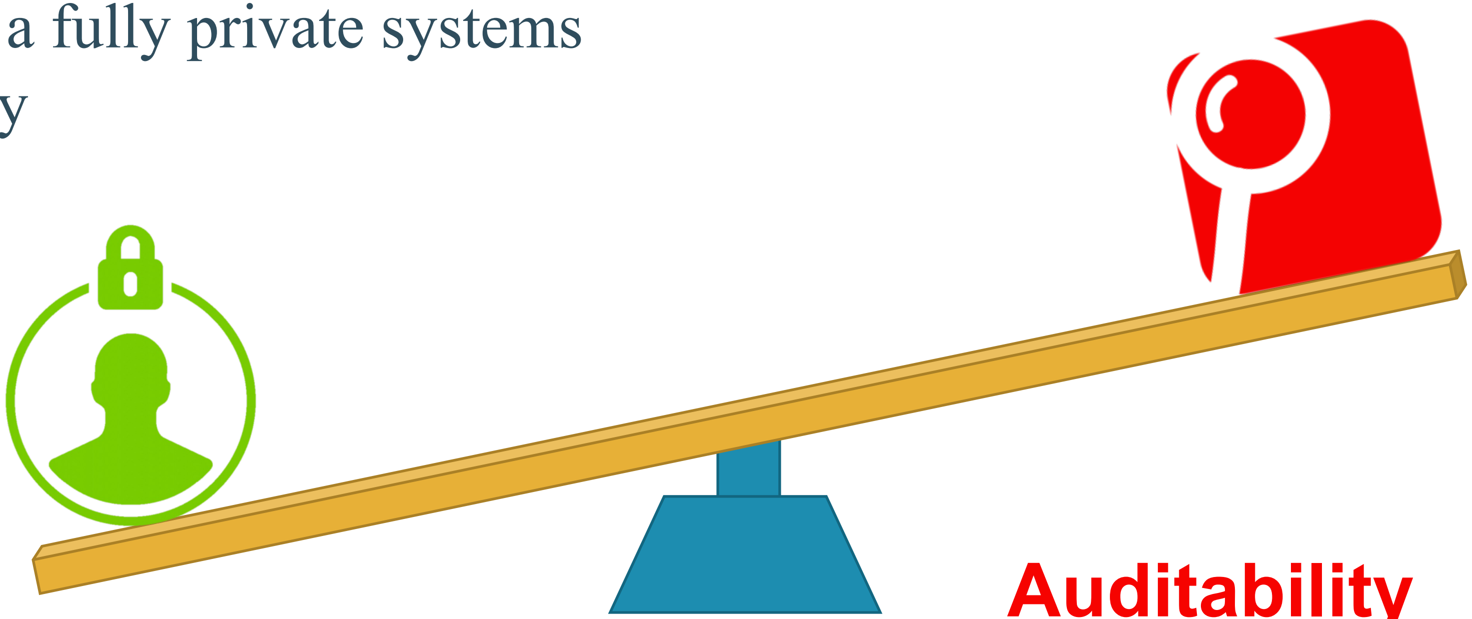
- Tax evasion
- Ransomware
- Drug trafficking
- Terrorist funding
- etc.

Privacy vs Accountability: In theory

Privacy

- Users willing a fully private systems
- No traceability
- Unlinkability





Auditability

- To prevent possible illicit activities
- To trace the suspicious actions

Privacy vs Accountability: In practice

CRYPTO BRIEFING

Business Markets Analysis Technology People Education Briefings SIMETRI Research

News

EU Moving to Ban Privacy Coins: Report

by Robert Knight Nov. 15, 2

The European Union may be stepping up its regulatory actions in the privacy sector.

An official website of the United States Government Accessibility Languages Contact

U.S. DEPARTMENT OF THE TREASURY

ABOUT TREASURY POLICY ISSUES DATA SERVICES NEWS SEARCH

HOME › NEWS › PRESS RELEASES

NEWS

- Press Releases
  - Statements & Remarks
  - Readouts
  - Testimonies
- Featured Stories
- Webcasts
- Press Contacts

PRESS RELEASES

U.S. Treasury Sanctions Notorious Virtual Currency Mixer Tornado Cash

385

August 8, 2022

WASHINGTON – Today, the U.S. Department of the Treasury's Office of Foreign Assets Control (OFAC) sanctioned virtual currency mixer Tornado Cash, which has been used to launder more than $7 billion worth of virtual currency since its creation in 2019. This includes over $455 million stolen by the Lazarus Group, a Democratic People's Republic of Korea (DPRK) state-sponsored hacking group that was sanctioned by the U.S. in 2019, in the largest known virtual currency heist to date. Tornado Cash was subsequently used to launder more than $96 million of malicious cyber actors' funds derived from the June 24, 2022 Harmony Bridge Heist, and at least $7.8 million from the August 2, 2022 Nomad Heist. Today's action is being taken pursuant to Executive Order (E.O.) 13694, as amended, and follows OFAC's May 6, 2022 designation of virtual currency mixer Blender.io (Blender).

LATEST NEWS

March 12, 2023

Joint Statement by the Department of the Treasury, Federal Reserve, and FDIC

READOUT: U.S. Candidate for President of the World Bank Ajay Banga Visit to the United Kingdom

March 10, 2023

The U.S. Economic Recovery in International Context

READOUT: Secretary of the Treasury Janet L. Yellen Convenes Financial Regulators

Testimony of Secretary of the Treasury Janet L. Yellen Before the Committee on Ways and Means, U.S. House of Representatives

The press release also claimed that the protocol had been used "to launder more than $7 billion worth of virtual currency since its creation in 2019."

Some Existing Solutions:

1. Public Key Encryption:
   - A central auditor can open the details of suspicious tnx.
   - If Jafar be the auditor then he can see all tnx details.

2. Threshold Encryption:
   - The majority of auditors can open the details of suspicious tnx.
   - If Jafar and his friends be the auditors then they can see all tnx details.

1- Prevention is better than cure!

2- How an auditor can be suspicious to a fully anonymous tnx?!

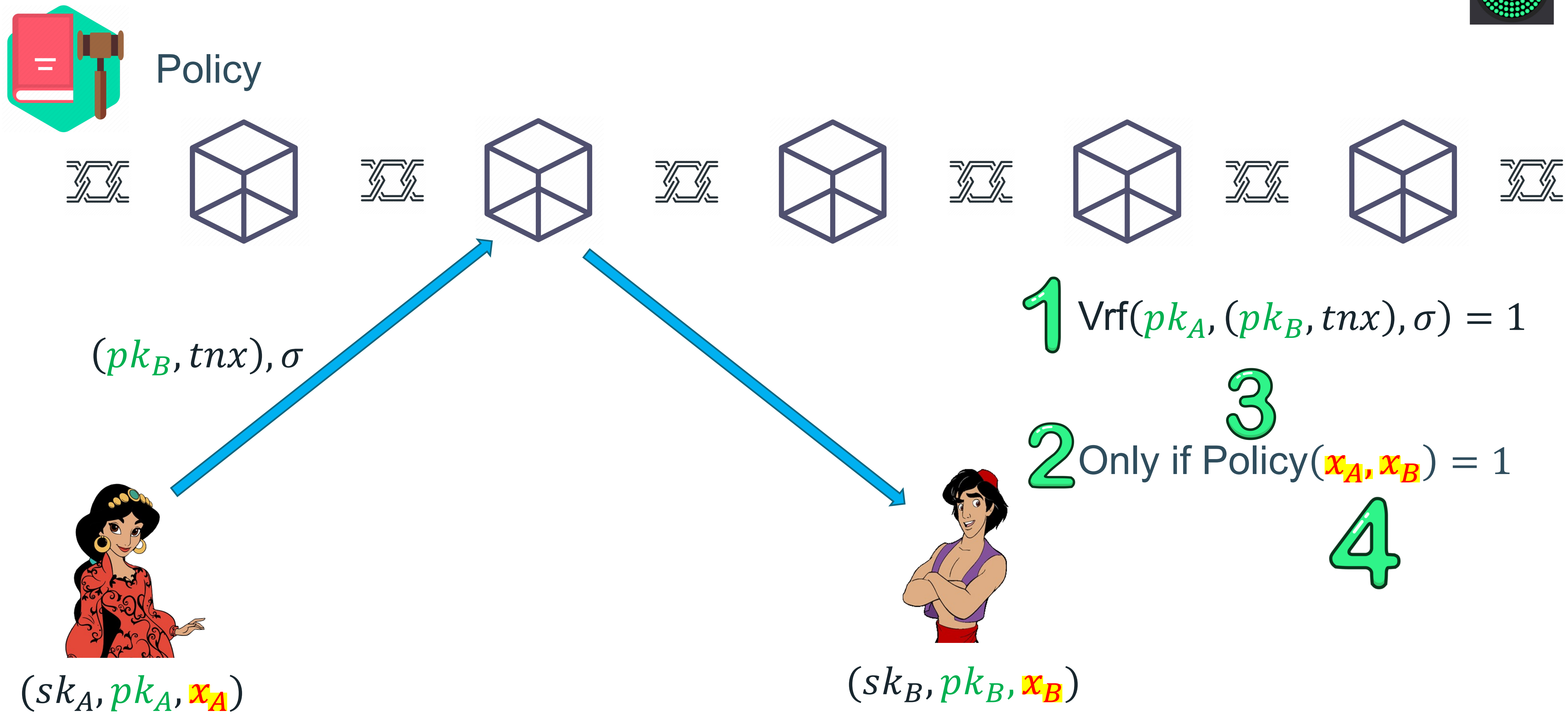
Prevention vs. Detection:

We are interested in:

Prevention rather than Detection

Joint policy

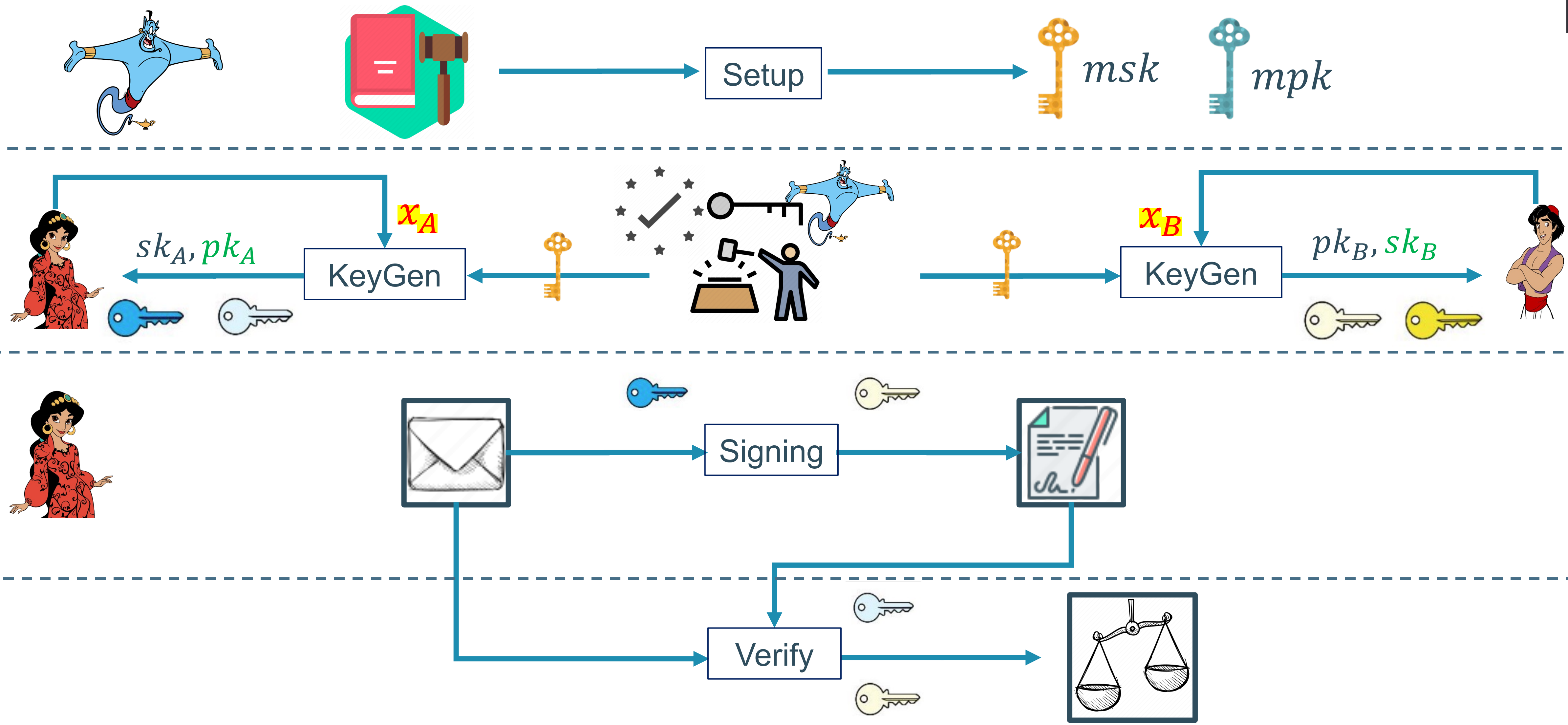
Possible solution for UTxO-based systems:



Some Possible Solutions:

	1 Unforgeability	2 P/A-based	3 Joint policy	4 S/R privacy
Digital Signatures	+	−	⊘	⊘
Attribute-based Signatures	+	+	−	⊘
Policy-based Signatures	+	+	+	−

Policy-Complaint Signatures [BMW21]:



PCS: Is this compatible with PP-Cryptocurrencies?

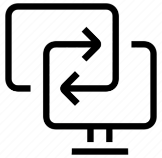
Correctness

Unforgeability

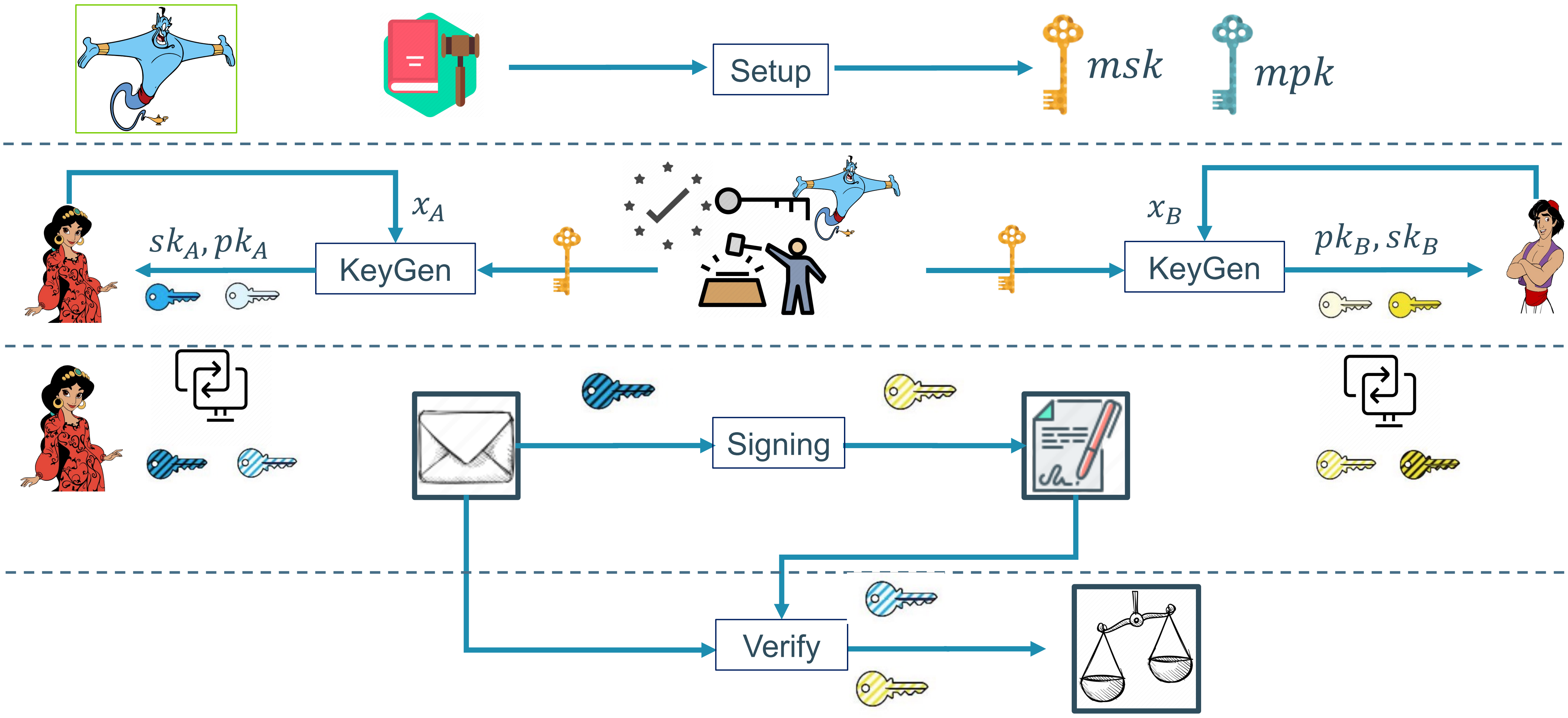
Attribute-Hiding

IF we want to remove the links then the users must be able to update their keys!

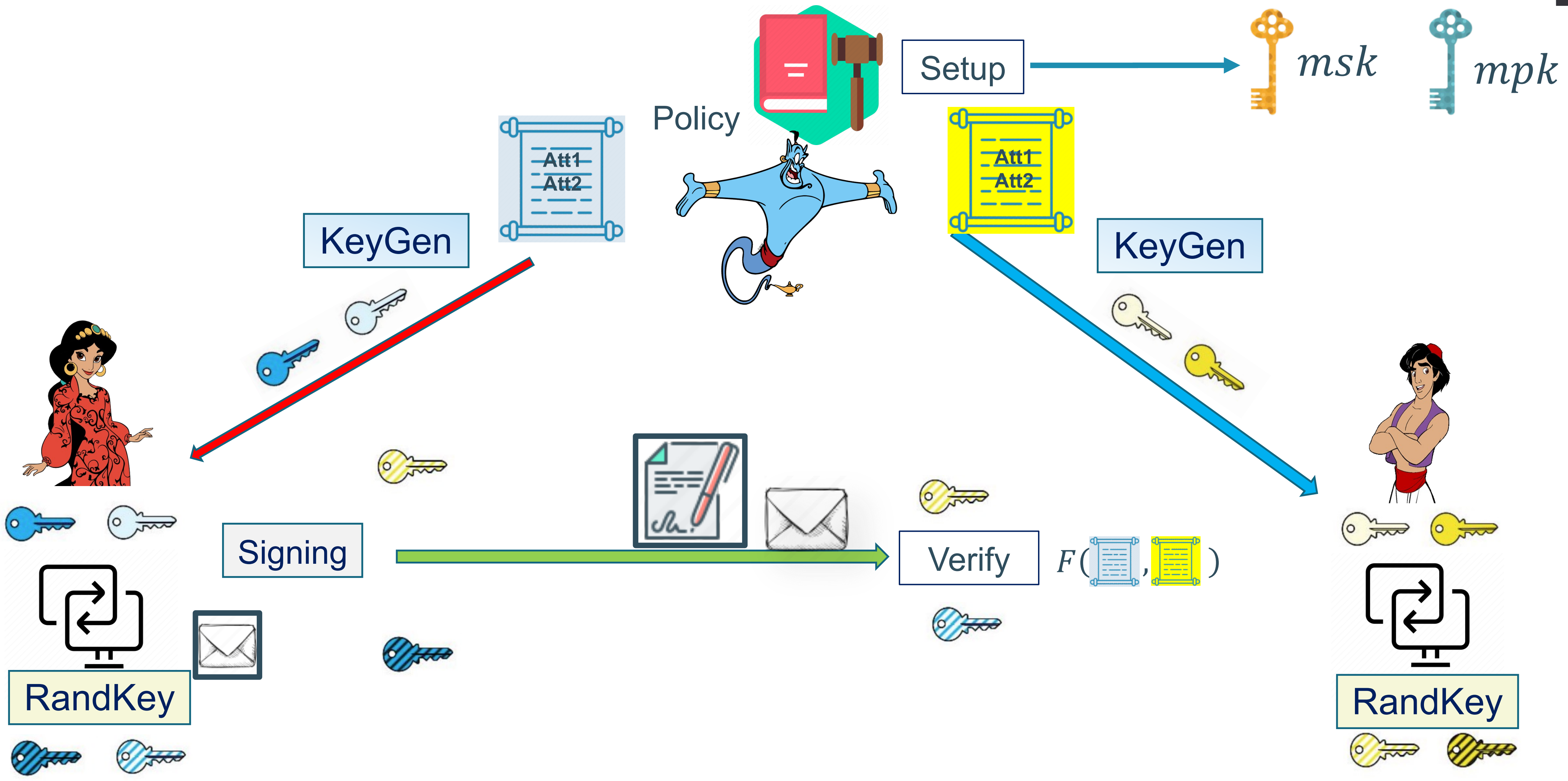
We need an extra algorithm called KeyRand(.)



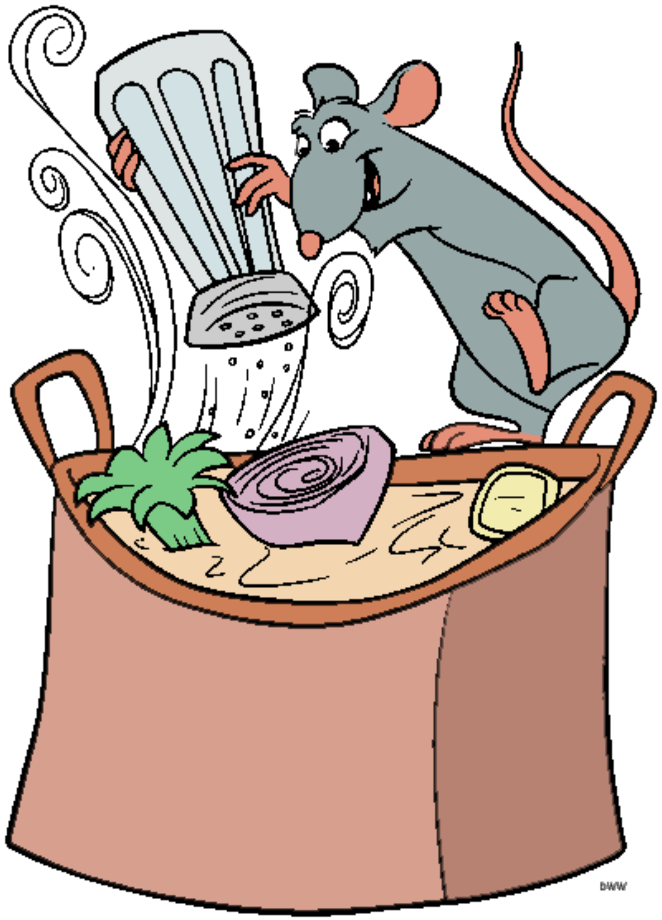
Unlinkable Policy-Complaint Signatures: Syntax



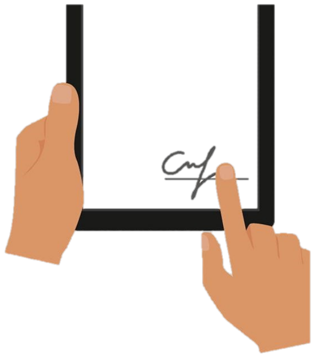
Unlinkable PCS: Architecture



Main Ingredients:



Digital Signatures

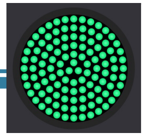


Predicate-Only
Predicate Encryptions

Pseudo-Random Functions

Zero-Knowledge proofs

Structure-Preserving Cryptography [AFG+10]:



- A general framework for efficient generic constructions of cryptographic primitives over bilinear groups*.


1 Groth-Sahai [GS08] proof system friendly

- ➢ Straight-line extraction.
- ➢ Standard Model.
- ➢ Applications: group signatures, blind signatures, etc.

2 Enabling Modular Design in complex systems

- ➢ Makes easy to combine building blocks.

Structure-Preserving Signatures [AFG+10]:

1

Source group elements of either $\mathbb{G}_1$ or $\mathbb{G}_2$

No Non-Linear operation like **Hash Functions**

2

Verify(, ,):

Done by:

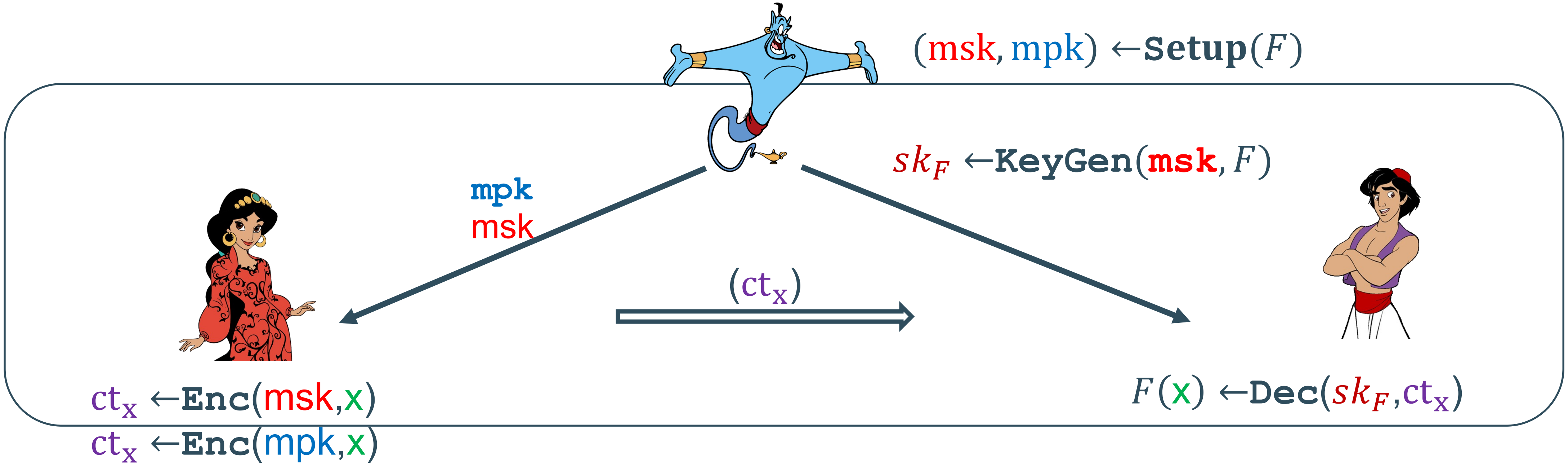
- membership tests

  $\in \mathbb{G}_1 \vee \mathbb{G}_2$

- pairing product equations

  $e(\quad,\quad)e(\quad, G_2) = 1_{\mathbb{G}_T}$

Private-Key Predicate-Encryptions [KSW07]:

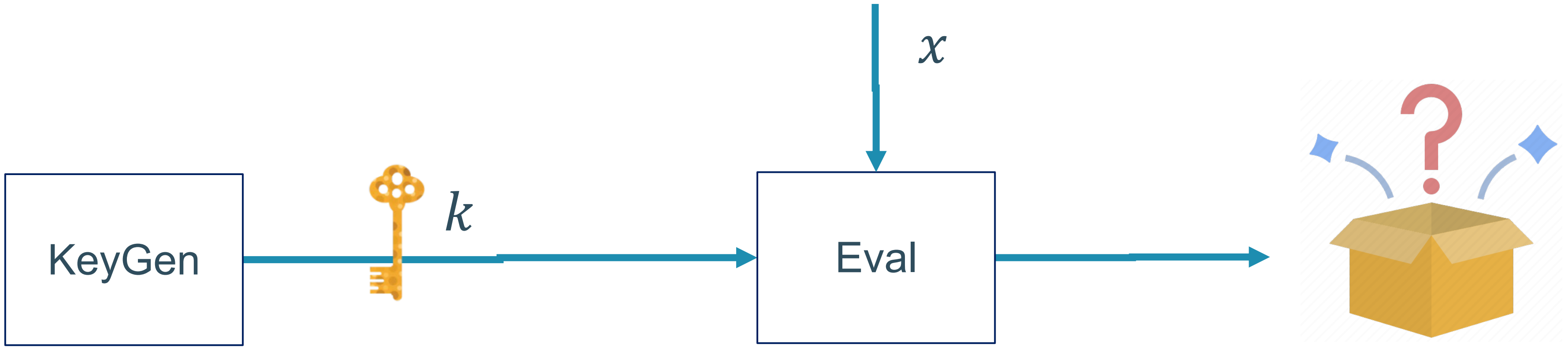


- **Correctness:** The decryption of a correctly generated ciphertext based on x returns $F(x)$.
- **Attribute-Hiding:** Ciphertext does not reveal any information about attribute sets x

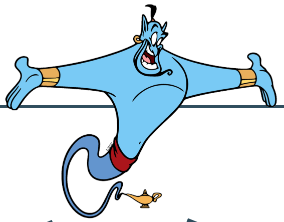
Inner-Product functionality: $F(x, y) = \sum x_i y_i$

Predicate-Only Predicate-Encryptions

Pseudo-Random Functions (PRF):



NIZKs [GMR89]



(CRS, Ext-TD, Sim-TD) $\leftarrow$ **Setup**(1^λ)

Witness w
$(x,w) \in R_L$

CRS

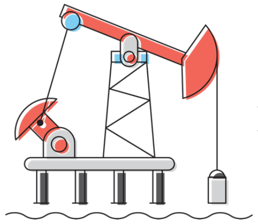
(stat, proof) $\Rightarrow$

CRS

proof $\leftarrow$ **Prove**(CRS, stat, witness)

{1, 0} $\leftarrow$ **Verify**(CRS, stat, proof)

- **Completeness:** honest P always will convince the honest V
- **Zero-Knowledge (ZK):** dishonest V only gets to know that the statement is true.
- **Knowledge Soundness:** dishonest P cannot convince honest V, unless she knows some secret "wit"



Ext(proof, Ext-TD) $\rightarrow$ witness: (stat, witness) $\in R_L$

Sim(stat, Sim-TD) $\rightarrow$ proof' $\approx_c$ proof

Our Generic Construction:

k ←PRF.KeyGen(1^λ)

$(sk_{sig}, vk_{sig}) \leftarrow \mathrm{DS.Setup}(1^\lambda)$

$sk_{f_x} \leftarrow \mathrm{PE.KeyGen}(msk_{PE}, f_x)$

$\sigma^1_{sig} \leftarrow \mathrm{DS.Sign}\left(sk^A_{sig}, (k, x)\right)$

$\sigma^2_{sig} \leftarrow \mathrm{DS.Sign}\left(sk^A_{sig}, (k, vk_{sig})\right)$

$\sigma^3_{sig} \leftarrow \mathrm{DS.Sign}\left(sk^A_{sig}, (k, sk_{f_x})\right)$

$usk = (k, sk_{f_x}, x, \sigma^1_{sig}, \sigma^2_{sig}, \sigma^3_{sig})$

$CRS_{Rand} \leftarrow \mathrm{NIZK}_{\mathcal{L}_1}.\mathrm{Setup}(1^\lambda)$

$CRS_{Sign} \leftarrow \mathrm{NIZK}_{\mathcal{L}_2}.\mathrm{Setup}(1^\lambda)$

$(mpk_{PE}, msk_{PE}) \leftarrow \mathrm{PE.Setup}(1^\lambda)$

$(sk^A_{sig}, vk^A_{sig}) \leftarrow \mathrm{DS.Setup}(1^\lambda)$

PRF setup

$(pk_0, sk_0) \leftarrow \mathrm{ReRand}(usk, -1)$

ctr←ctr+1

$ID_{ctr} \leftarrow PRF.\mathrm{Eval}(k, ctr)$

$(sk^{ctr}_{sig}, vk^{ctr}_{sig}) \leftarrow \mathrm{DS.Setup}(1^\lambda)$

$\sigma_{ctr} \leftarrow \mathrm{DS.Sign}\left(sk_{sig}, (vk^{ctr}_{sig}, ID_{ctr})\right)$

$ct_{ctr} \leftarrow \mathrm{PE.Enc}(mpk_{PE}, x)$

$\pi_{ctr} \leftarrow \mathrm{NIZK}_{\mathcal{L}_1}.\mathrm{Prove}(wit_r, ins_r)$

Return (pk_{ctr}, sk_{ctr})

If $\mathrm{NIZK}_{\mathcal{L}_1}.\mathrm{Verify}(ins_r, \pi_{ctr}) = 1$:

ID_S ←PRF.Eval(k,ctr)

If $ct_{ctr} \leftarrow \mathrm{PE.Dec}(sk_{f_x}, ct_R) = 1$:

$\pi_s \leftarrow \mathrm{NIZK}_{\mathcal{L}_2}.\mathrm{Prove}(wit_s, ins_s)$

If $\mathrm{NIZK}_{\mathcal{L}_1}.\mathrm{Verify}(ins_r, \pi_{ctr}) = 1$
and
$\mathrm{NIZK}_{\mathcal{L}_2}.\mathrm{Verify}(ins_s, \pi_s) = 1$

NIZK Relations:

Language $\mathcal{L}_1$

wit_r	Relations
ctr	$\mathsf{ctr} \in [0, T_{\mathsf{rand}})$
k	$\mathsf{ID}_{\mathsf{ctr}} = \mathsf{PRF.Eval}(\mathsf{k}, \mathsf{ctr})$
σ^1_{sig}	$\mathsf{SPS.Verify}\left(\mathsf{vk}^{A}_{\mathsf{sig}}, (\mathsf{k}, \mathsf{vk}_{\mathsf{sig}}), \sigma^1_{\mathsf{sig}}\right) = 1$
$\mathsf{vk}_{\mathsf{sig}}$	$\mathsf{SPS.Verify}\left(\mathsf{vk}^{A}_{\mathsf{sig}}, (\mathsf{k}, x), \sigma^2_{\mathsf{sig}}\right) = 1$
σ^2_{sig}	$\mathsf{ct}_{\mathsf{ctr}} = \mathsf{PE.Enc}(\mathsf{mpk}_{\mathsf{FE}}, x)$
x	$\mathsf{DS.Verify}\left(\mathsf{vk}_{\mathsf{sig}}, \left(\mathsf{vk}^{\mathsf{ctr}}_{\mathsf{sig}}, \mathsf{ID}_{\mathsf{ctr}}\right), \sigma_{\mathsf{ctr}}\right) = 1$
σ_{ctr}	

Language $\mathcal{L}_2$

wit_s	Relations
ctr	$\mathsf{ID}_S = \mathsf{PRF.Eval}(\mathsf{k}, \mathsf{ctr})$
k	$\mathsf{SPS.Verify}\left(\mathsf{vk}^{A}_{\mathsf{sig}}, (\mathsf{k}, \mathsf{sk}_{f_x}), \sigma^2_{\mathsf{sig}}\right) = 1$
σ^2_{sig}	$\mathsf{PE.Dec}(\mathsf{sk}_{f_x}, \mathsf{ct}_R) = 1$
sk_{f_x}	

Range-proofs | Sigma protocols | Groth-Sahai proofs

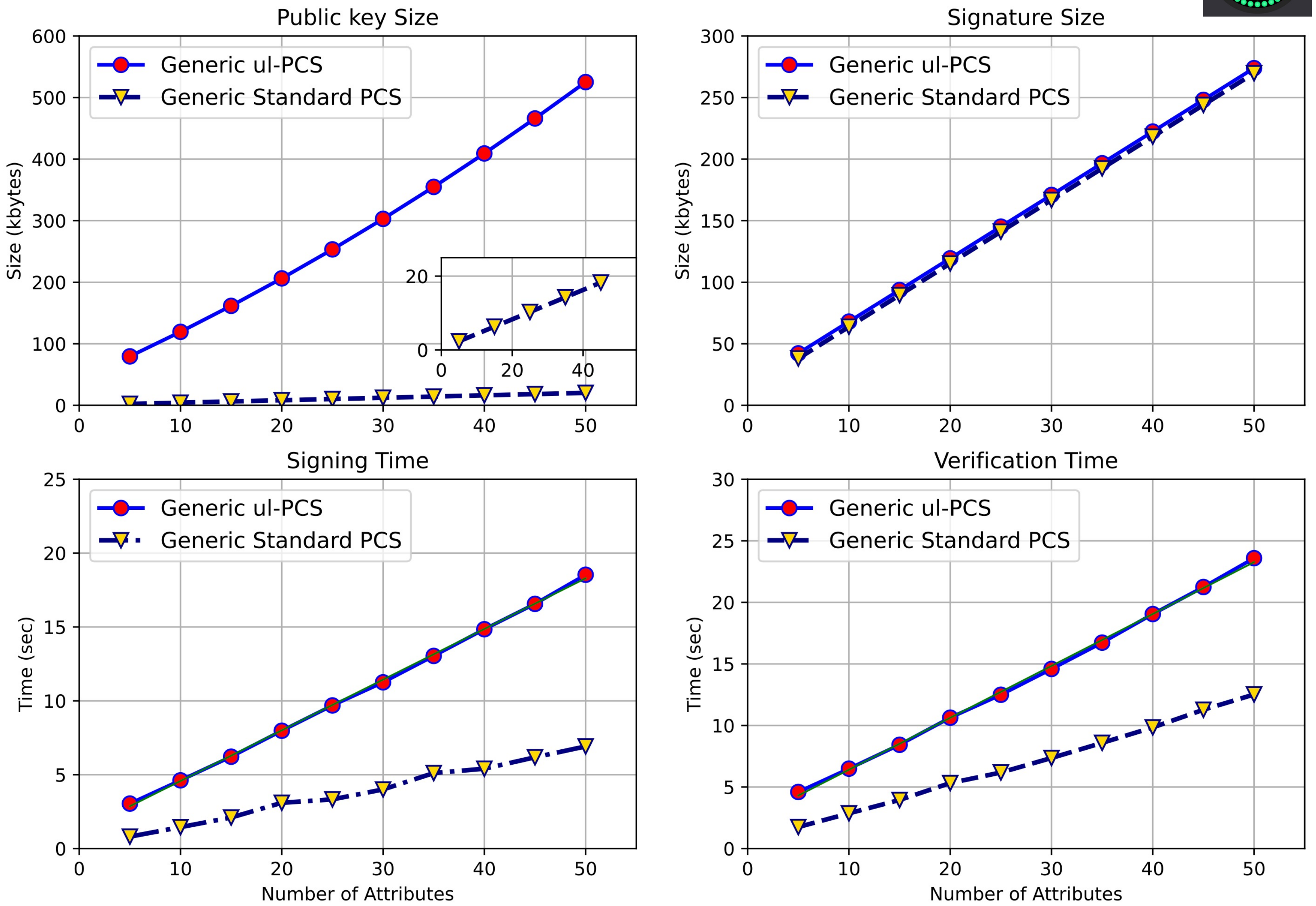
An Instantiation of Generic construction:

1. **Digital Signatures**
    - BLS signatures [BLS04] when message and signatures are public, else
        - Selectively Randomizable SPS and SPS-EQ in [FHS19]
        - Constant signature size (3 base group elements)
        - Groth-Sahai [GS08] proof system friendly
2. **Predicate-Only Predicate Encryptions**
    - Okamoto-Takashima [OT12]
    - Policy: Inner-products predicate functionalities
3. **Pseudo-Random functions**
    - Dodis-Yampolsky PRF [DY05]
4. **NIZK**
    - Sigma protocols [Sch89]: when the scalar is known
    - Groth-Sahai [GS08] proof systems: when all witnesses are group elements (batched version from ACM CCS'2017 [HHK+17])
    - Bulletproof range-proofs [BBP+18]

Privacy is expensive?!

Ubuntu 20.04.2 LTS
an Intel Core i7-9850H CPU @ 2.60 GHz
with 16 GB of memory

Charm-Crypto framework
Barreto-Naehrig asymmetric curve
BN254
with embedding degree 12



How the policies can be defined? IP vs. Role-based

$$F: \mathcal{S} \times \mathcal{R} \rightarrow \{0,1\}$$

$$F(x,y) = \sum x_i y_i$$

x_{P2}:
"Austrian"
"Prof."
"Age>?"

x_{P1}:
"HK"
"Prof."
"Age>?"

x_{P4}:
"Japanese"
"PostDoc"
"Age<?"

x_{P3}:
"Iranian"
"PhD"
"Age>?"

	P1	P2	P3	P4
P1	0	1	1	0
P2	1	0	0	1
P3	1	1	1	1
P4	0	1	0	1

Fine-grained policies

$$F: [n_R] \times [n_R] \rightarrow \{0,1\}$$
$$F(x,y) \rightarrow \{0,1\}$$

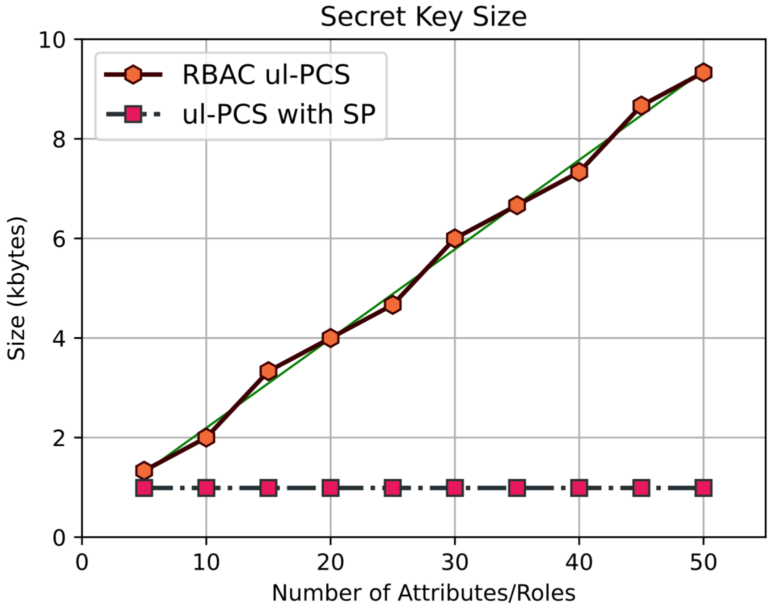
	P1	P2	P3	P4
P1	0	1	1	0
P2	1	0	0	1
P3	1	1	1	1
P4	0	1	0	1

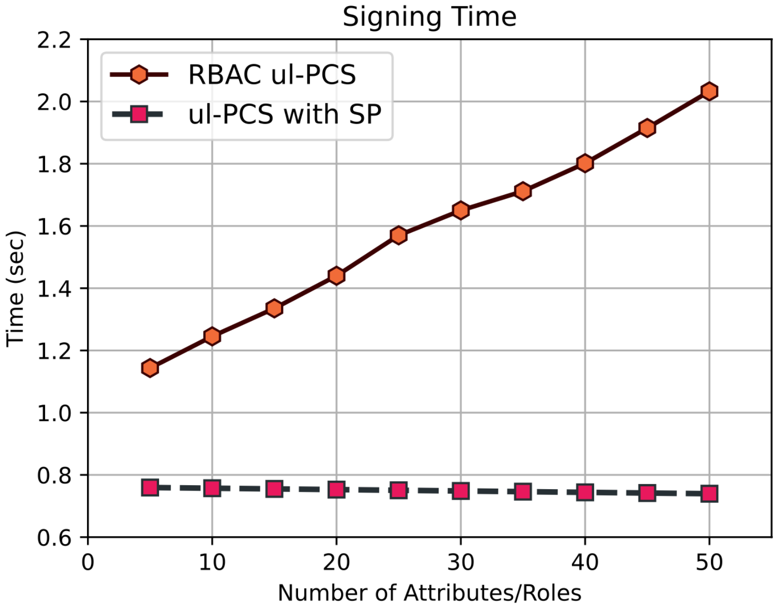
Role-based Access control

$$F: \mathcal{S} \wedge \mathcal{R} \rightarrow \{0,1\}$$
$$F(x,y) = S(x) \wedge R(y)$$

UL-PCS with separable policies

Benchmarks: Role-based and Separate Policies





Scheme	KeyGen time (ms)	RandKey time (ms)	Verify time (ms)	pk size (kbytes)	σ size (kbytes)
ul-PCS, role-based	750	550	1 630	28	16
ul-PCS, separable policies	490	480	1 020	28	14.5

There is space for further improvements:

Elliptic Curve	Library	$\mathbf{M}_1$ time	$\mathbf{E}_1$ time	$\mathbf{M}_2$ time	$\mathbf{E}_2$ time	$\mathbf{M}_T$ time	$\mathbf{E}_T$ time	$\mathbf{P}$ time
BN-254	Charm-Crypto	3.3 μs	0.9 ms	7.1 μs	1.6 ms	21.4 μs	4.8 ms	18.5 ms
BN-256	bplib	3.8 μs	0.3 ms	6 μs	1 ms	3 μs	2.3 ms	2.74 ms

Conclusion:

1. We talked the importance of accountable anonymous.
2. The existing challenges and possible solutions.
3. We overviewed the syntax of unlinkable PCS.
4. We discussed their applications and main building blocks.
5. We talked about two more efficient instantiation than the generic model.
6. We discussed the complexity of the proposed solutions.

Potential open questions and subsequent works:

- Minimize the needed trust to the central issuer.

- Design more efficient PO-PE → more efficient generic construction.

- Take a different approach with the same security properties.

References:

- [Schnorr89] C.P. Schnorr. “Efficient identification and signatures for smart cards.”, CRYPTO 1989
- [GMR89] Sh. Goldwasser, S. Micali, and Ch. Rackoff. “The knowledge complexity of interactive proof-systems.”, STOC 1985
- [GS08] J. Groth and A. Sahai. “Efficient non-interactive proof systems for bilinear groups.”, EUROCRYPT 2008
- [Shamir79] A. Shamir. “How to Share a Secret”. In Commun. ACM 22(11), pp. 612–613, 1979.
- [OT12] T. Okamoto, K. Takashima: “Adaptively attribute-hiding (hierarchical) inner product encryption.”, EUROCRYPT 2012
- [BMW21] C. Badertscher, C. Matt, H. Waldner: “Policy-Compliant Signatures.”, TCC 2021
- [BBP+18] B. Bunz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, G. Maxwell: “Bulletproofs: Short proofs for confidential transactions and more.”, IEEE Symposium on Security and Privacy 2018
- [DY05] Y. Dodis, A. Yampolskiy: “A verifiable random function with short proofs and keys.”, PKC 2005
- [EKKV22] F. Engelmann, T. Kerber, M. Kohlweiss, M. Volkhov: “ZSwap: zk-SNARK based Non-Interactive multi-asset swaps”, PETS 2022
- [FHS19] G. Fuchsbauer, C. Hanser, D. Slamanig: “Structure-preserving signatures on equivalence classes and constant-size anonymous credentials.” Journal of Cryptology 2019
- [BLS04] D. Boneh, B. Lynn, H. Shacham: “Short signatures from the Weil pairing.”, ASIACRYPT 2001
- [AFG+10] Abe et al. “Structure-preserving signatures and commitments to group elements.”, CRYPTO 2010.
- [KSW07] J. Katz, A. Sahai, B. Waters: “Predicate encryption supporting disjunctions, polynomial equations, and inner products.”, EUROCRYPT 2008
- [HHK+17] G. Herold, M. Hoffmann, M. Klooß, C. R`afols, A. Rupp: “New techniques for structural batch verification in bilinear groups with applications to groth-sahai proofs.”, ACM CCS 2017

Thank You!

The illustrations are credited to Disneyclips.