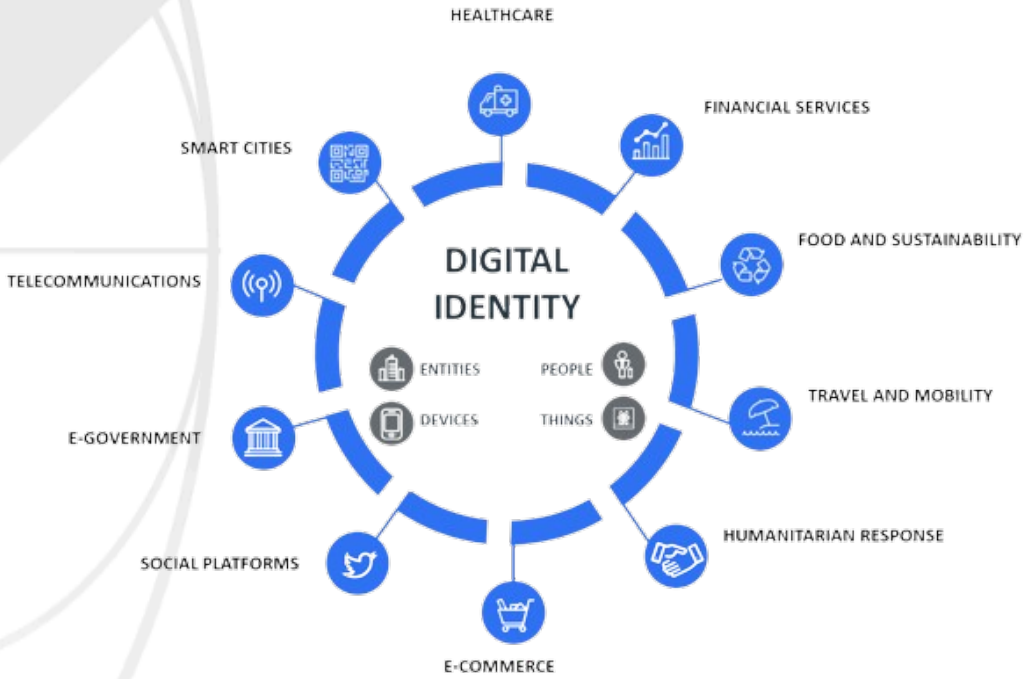
Malleable Cryptography: Advances and Applications to Privacy-enhancing Technologies

Octavio Pérez Kempner

DIENS, École normale supérieure, CNRS, PSL University, Paris, France
Thèse CIFRE effectuée au sein de Almerys (be-ys group)

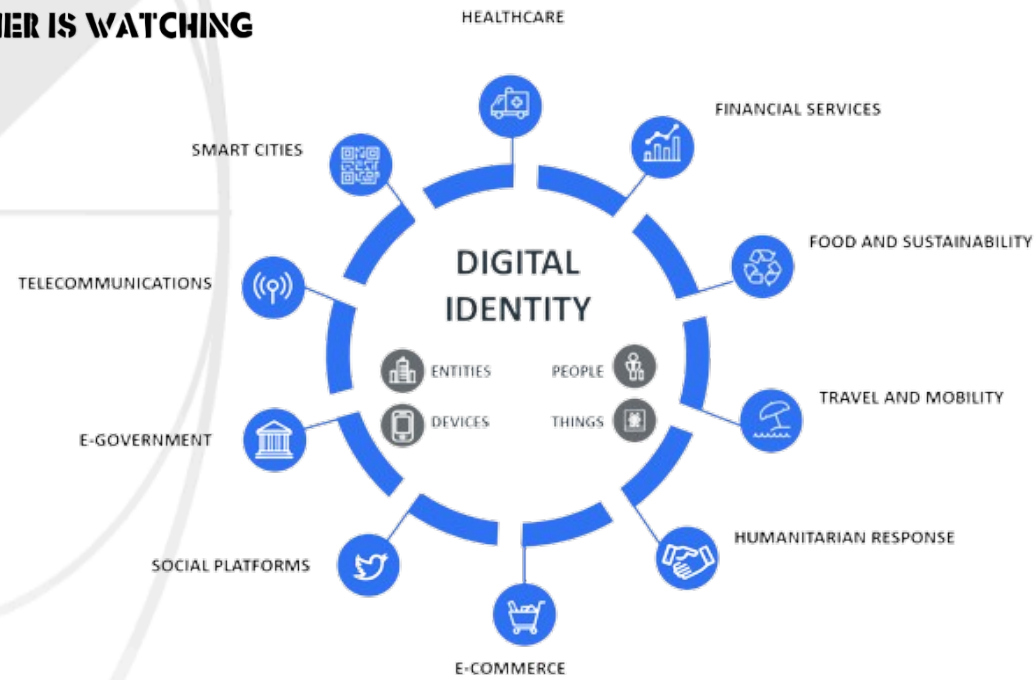
Soutenance de thèse de doctorat - 26 octobre 2022

Introduction



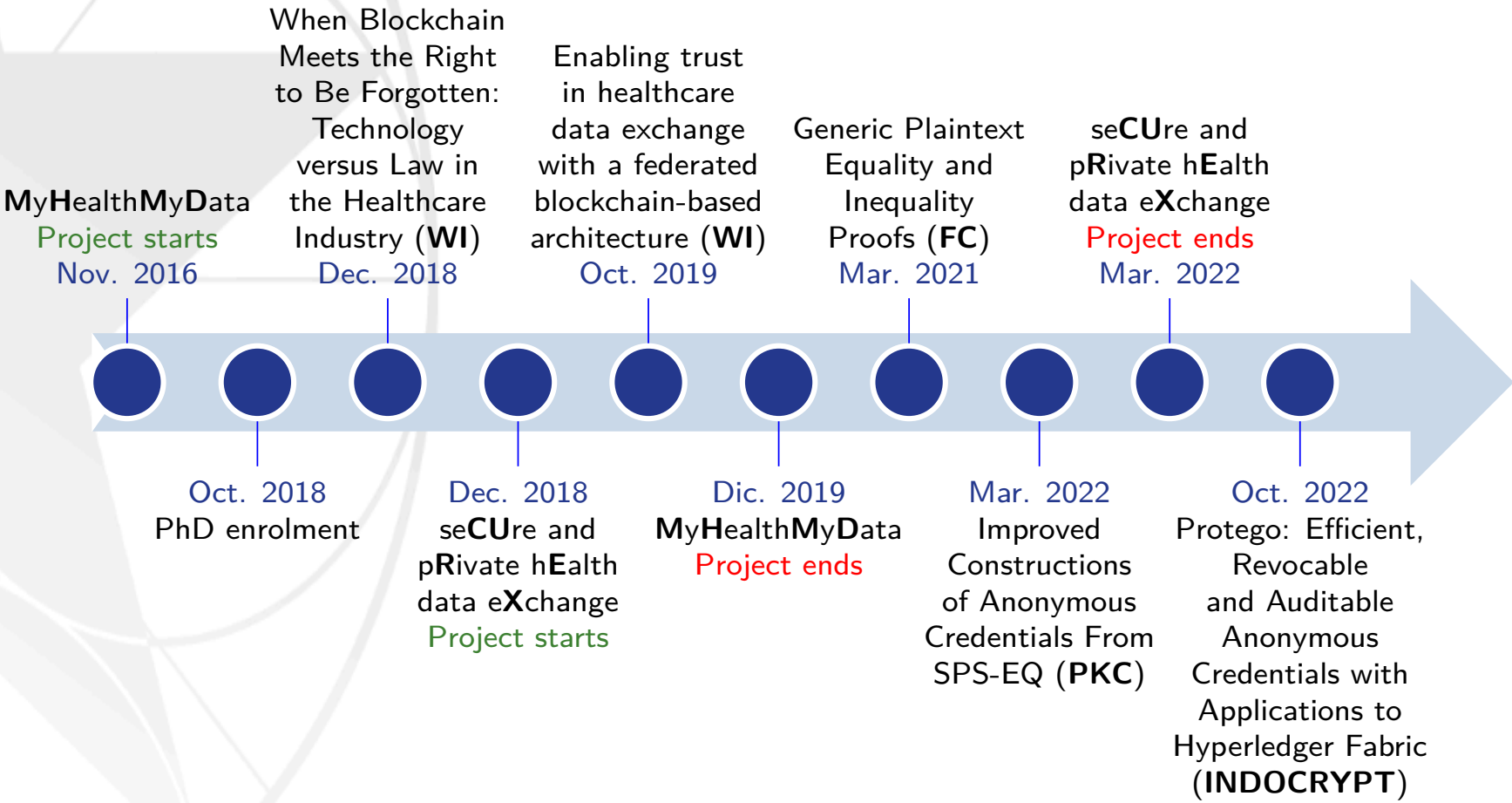
Source: World Economic Forum: Identity in a Digital World

Introduction



Source: World Economic Forum: Identity in a Digital World

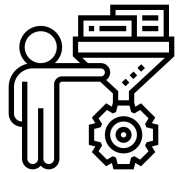
Chronology


MyHealthMyData
Project starts
Nov. 2016
Oct. 2018
PhD enrolment
When Blockchain Meets the Right to Be Forgotten: Technology versus Law in the Healthcare Industry (WI)
Dec. 2018
Dec. 2018
seCUre and pRivate hEalth data eXchange
Project starts
Enabling trust in healthcare data exchange with a federated blockchain-based architecture (WI)
Oct. 2019
Dic. 2019
MyHealthMyData
Project ends
Generic Plaintext Equality and Inequality Proofs (FC)
Mar. 2021
Mar. 2022
Improved Constructions of Anonymous Credentials From SPS-EQ (PKC)
seCUre and pRivate hEalth data eXchange
Project ends
Mar. 2022
Oct. 2022
Protego: Efficient, Revocable and Auditable Anonymous Credentials with Applications to Hyperledger Fabric (INDOCRYPT)

Privacy-enhancing Technologies

Measures to protect privacy

Eliminate or minimise use of personal data



Prevent unnecessary processing

Without loss of functionality

I'm 21 years old

Privacy-enhancing Technologies

Measures to protect privacy

Eliminate or minimise use of personal data

Prevent unnecessary processing

Without loss of functionality

~~I'm 21 years old~~

I'm not underage

Malleable Cryptography

What do they have in common?

Malleable Cryptography

Malleable Cryptography: Public-key Encryption

Ron: 🔑, ✉

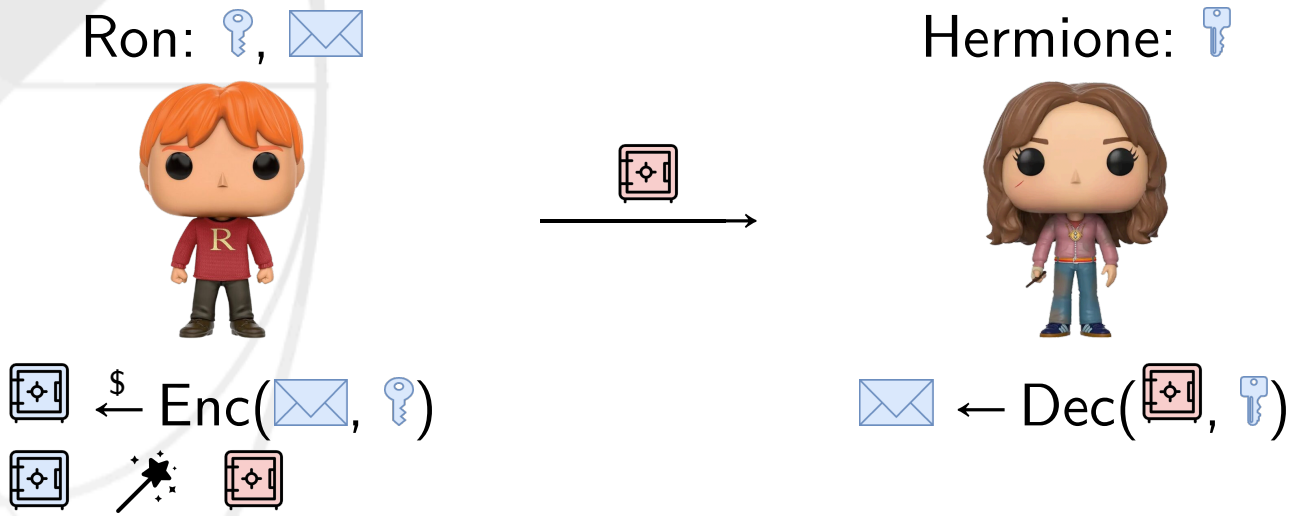
Hermione: 🔑

🔒 $\xleftarrow{\$}$ Enc(✉, 🔑)

✉ ← Dec(🔒, 🔑)

Malleable Cryptography: Public-key Encryption

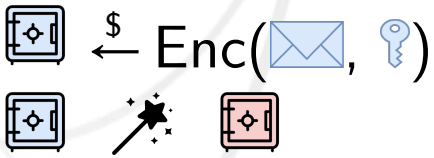
Malleability w.r.t. ciphertexts



Malleability w.r.t. messages

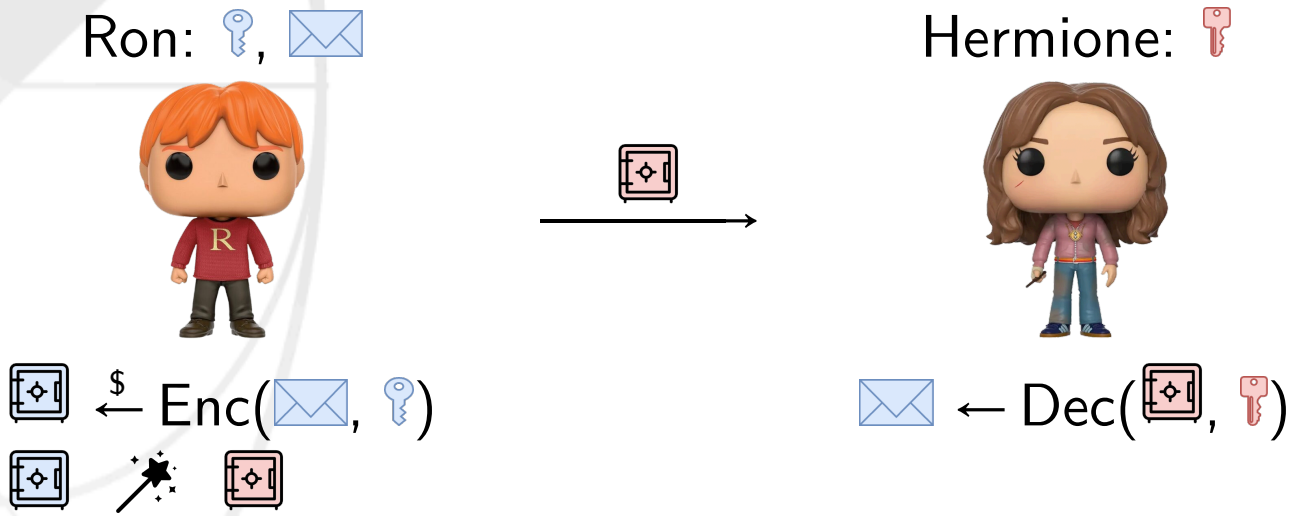
Ron: 

Hermione:



Malleable Cryptography: Public-key Encryption

Malleability w.r.t. keys



Malleable Cryptography: Digital Signatures

Ron: 🔑, ✉

$\sigma \xleftarrow{\$} \mathsf{Sign}(✉, 🔑)$

Hermione: 🔑, ✉

$\mathit{True} \leftarrow \mathsf{Verify}(\sigma, ✉, 🔑)$

Malleability w.r.t. signatures

Ron: , 

Hermione:  ,

 $\xleftarrow{\$}$ Sign( , )

True $\leftarrow$ Verify( ,  , )

Malleability w.r.t. messages

Ron:  , 

Hermione:  ,

 $\xleftarrow{\$}$ Sign( , )

True ← Verify( ,  , )

Malleability w.r.t. keys

Ron:  , 

Hermione:  ,

 $\xleftarrow{\$}$ Sign( , )

True $\leftarrow$ Verify( ,  , )

Malleable Cryptography: Zero-knowledge Proofs

Ron: , 

Hermione:

 $\xleftarrow{\$}$ Prove( , )

True ← Verify( ,)

Malleability w.r.t. proofs

Ron: ,

Hermione:

$\xleftarrow{\$}$ Prove(,)

True ← Verify(,)

Malleability w.r.t. statements

Ron: ,

Hermione:

$\xleftarrow{\$}$ Prove(,)

True $\leftarrow$ Verify(,)

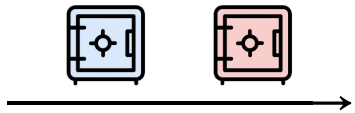
Contributions: Advances and Applications

- Formalized Generic Randomizable Encryption (**FC '21**)
- Generic Plaintext Equality and Inequality Proofs (**FC '21**)
- More efficient Structure-Preserving Signatures on Equivalence Classes (SPS-EQ) in the standard model (**PKC '22**)
- First mercurial signature in the standard model (**PKC '22**)
- Extended Attribute-based credentials (ABC) (**PKC '22, INDOCRYPT '22**)
- Permissioned Blockchains (Hyperledger Fabric) (**INDOCRYPT '22**)

Outline

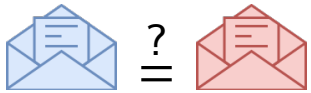
Generic Plaintext Equality and Inequality Proofs

Ron



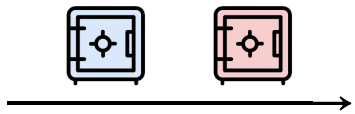
Hermione

Plaintext Equality:



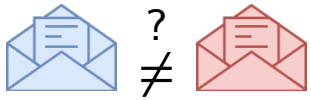
Generic Plaintext Equality and Inequality Proofs

Ron



Hermione

Plaintext Inequality:



Applications

Voting

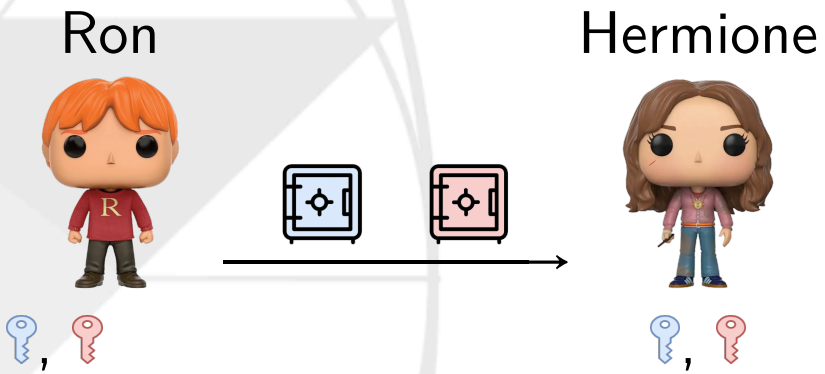
Reputation systems

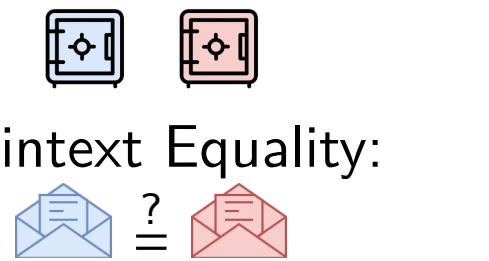
Cloud applications

Broadcast

Storage

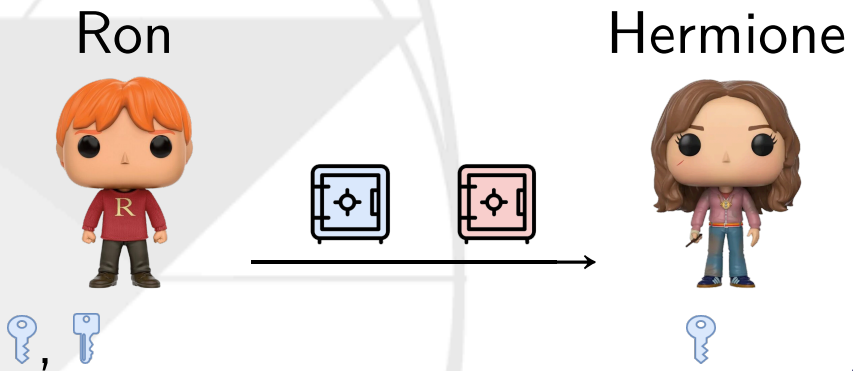
Generic Plaintext Equality and Inequality Proofs

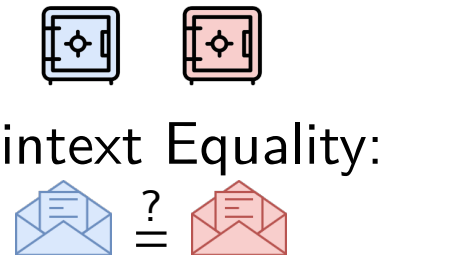




- = and Ron knows

Generic Plaintext Equality and Inequality Proofs

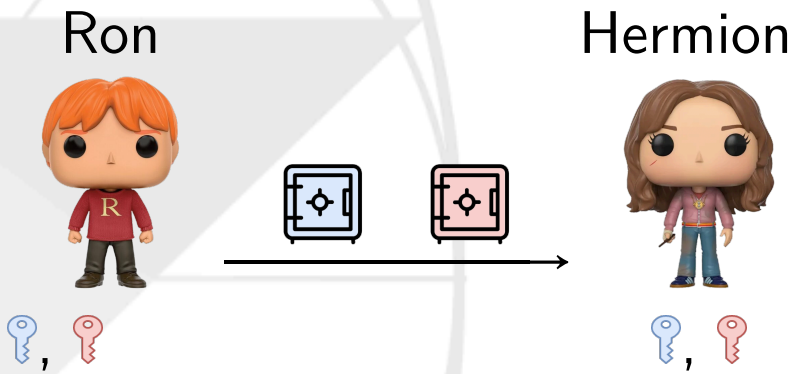


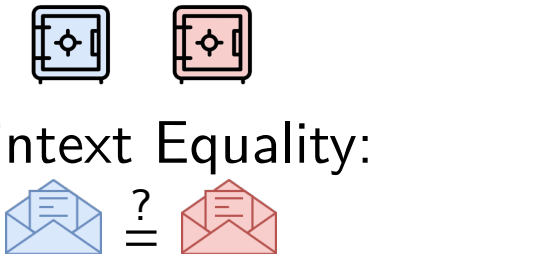


- $\mathsf{pk}_1 = \mathsf{pk}_2$ and Ron knows sk_1

  $\rightsquigarrow$ HPEQ, PEQ

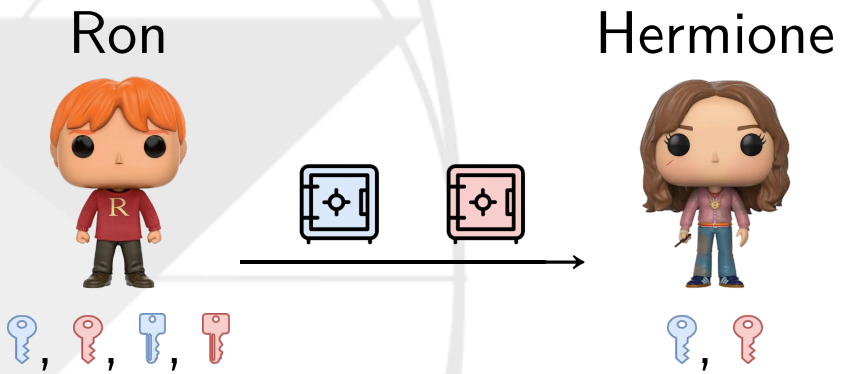
Generic Plaintext Equality and Inequality Proofs

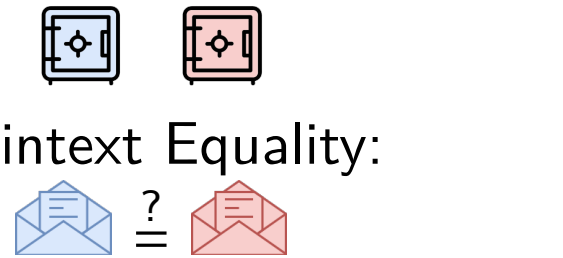




- $k_1 = k_2$ and Ron knows k_1
  $\rightsquigarrow$ HPEQ, PEQ
- $k_1 \neq k_2$ and Ron knows k_1 & k_2

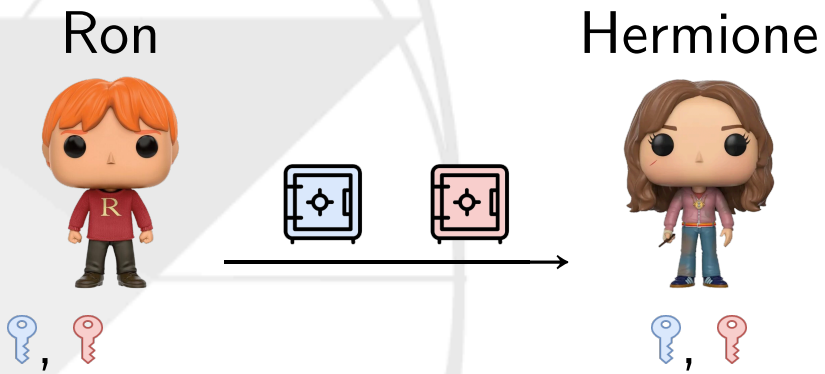
Generic Plaintext Equality and Inequality Proofs

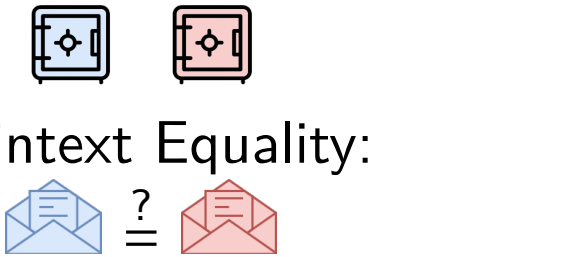




- $=$ and Ron knows
  $\rightsquigarrow$ HPEQ, PEQ
- $\neq$ and Ron knows &
  $\rightsquigarrow$ MATCHPEQ, SIGPEQ

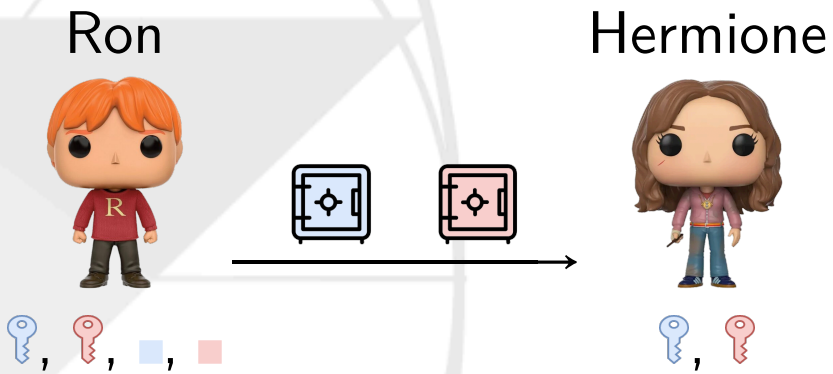
Generic Plaintext Equality and Inequality Proofs

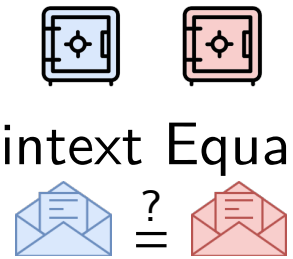




- $=$ and Ron knows
  $\rightsquigarrow$ HPEQ, PEQ
- $\neq$ and Ron knows &
  $\rightsquigarrow$ MATCHPEQ, SIGPEQ
- $\neq$ and Ron knows the encryption's random coins

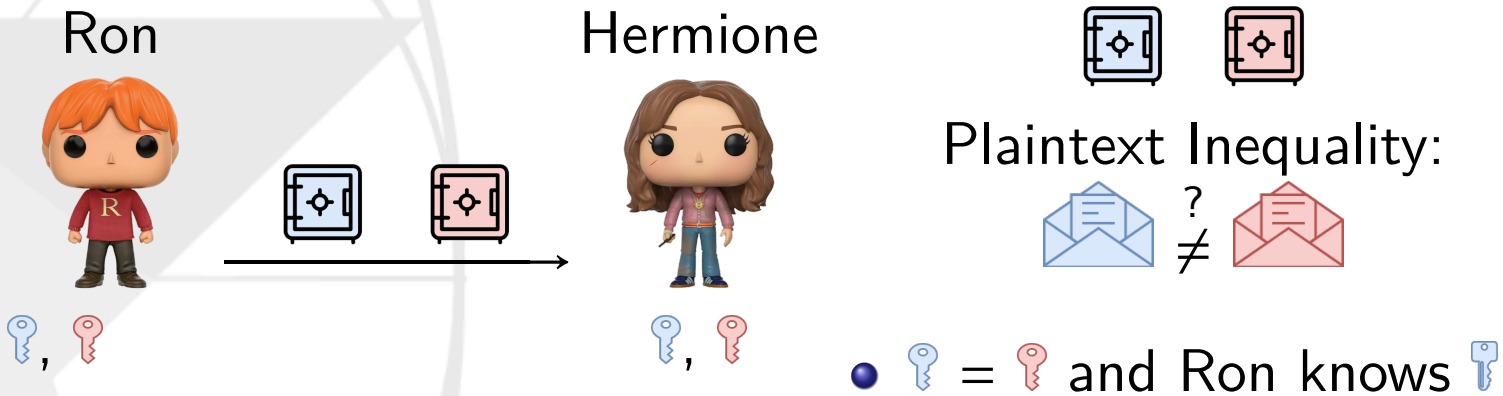
Generic Plaintext Equality and Inequality Proofs



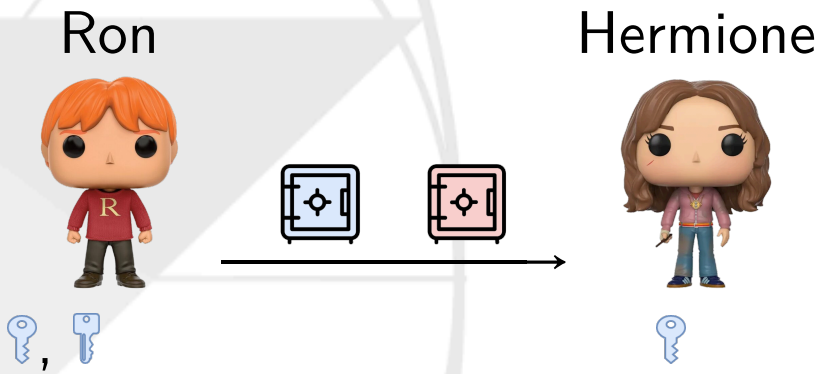


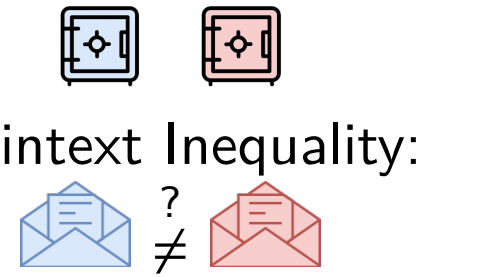
- 🔑 = 🔑 and Ron knows 🔑
  ⇝ HPEQ, PEQ
- 🔑 ≠ 🔑 and Ron knows 🔑 & 🔑
  ⇝ MATCHPEQ, SIGPEQ
- 🔑 ≠ 🔑 and Ron knows the encryption's random coins
  ⇝ RSPEQ

Generic Plaintext Equality and Inequality Proofs



Generic Plaintext Equality and Inequality Proofs





- = and Ron knows
  ⇝ HPINEQ, PINEQ

Generic Plaintext Equality and Inequality Proofs

- Randomisation requirements:
  - Plaintext Inequality (HPINEQ, PINEQ)

    $\rightsquigarrow$ Ciphertexts: 
  - Plaintext Equality (HPEQ, PEQ)

    $\rightsquigarrow$ Ciphertexts and messages: 
  - Sigma protocols for plaintext equality (MATCHPEQ, SIGPEQ, RSPEQ)

    $\rightsquigarrow$ Ciphertexts, messages keys (or random coin decryption):

How Generic our Protocols are?


ElGamal


Goldwasser-Micali


Paillier


Damgård


Cramer-Shoup


DS Cramer-Shoup

How Generic are our Protocols?

ElGamal

IND-CPA

DLP

Goldwasser-Micali

IND-CPA

QRP

Paillier

IND-CPA

DCRA

Damgård

IND-CCA1

DLP

Cramer-Shoup

IND-CCA1

DDH

DS Cramer-Shoup

RCCA

DDH

How Generic are our Protocols?

ElGamal	Goldwasser-Micali	Paillier	Damgård	Cramer-Shoup	DS Cramer-Shoup
IND-CPA	IND-CPA	IND-CPA	IND-CCA1	IND-CCA1	RCCA
DLP	QRP	DCRA	DLP	DDH	DDH
PEQ PINEQ MATCHPEQ SIGPEQ RSPEQ	PEQ PINEQ MATCHPEQ RSPEQ	PEQ PINEQ SIGPEQ RSPEQ	PEQ PINEQ MATCHPEQ SIGPEQ RSPEQ	PEQ PINEQ RSPEQ	PINEQ

Conclusions

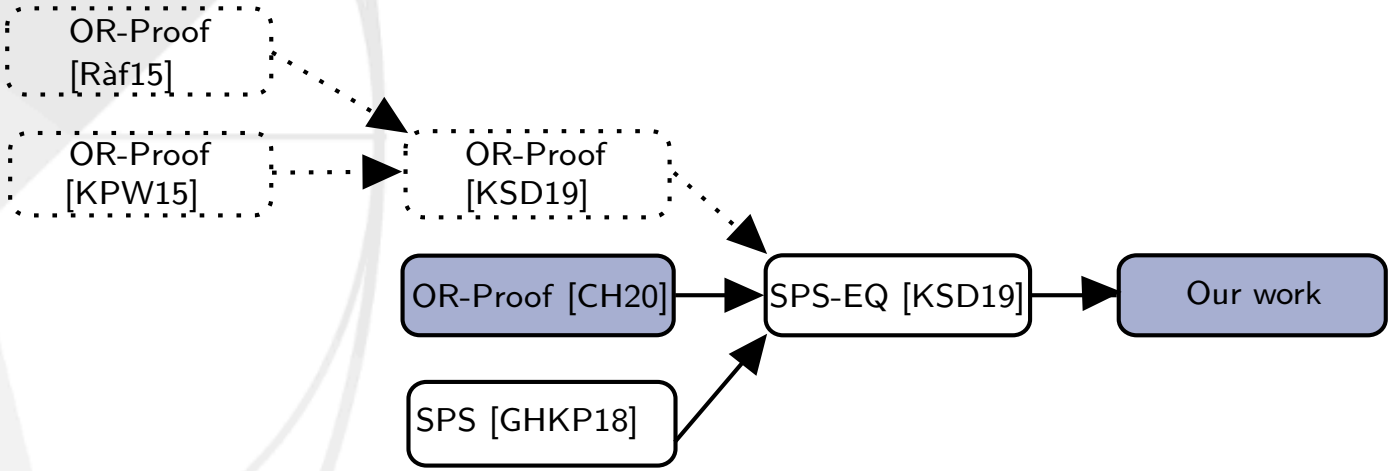
- Intuitive constructions from generic randomisable encryption
- Non-interactive variants for sigma protocols via Fiat-Shamir
- Applicable to real-world problems in a “plug & play” manner

Outline

SPS-EQ: Intuition

- Controlled form of malleability: 
- Message space can be partitioned into equivalence classes
  - *e.g.*, $m \in \mathbb{G}^\ell \sim_{\mathscr{R}} m' \in \mathbb{G}^\ell \iff \mu \in \mathbb{Z}_p^*$ s.t. $m' = \mu m$
- Unforgeability holds with respect to classes
- Message-signature pairs in the same class are unlinkable
- Extended to consider equivalence classes on the key space [BHKS18,CL19]

SPS-EQ: Our construction



SPS-EQ: Comparison with previous work

Scheme	$\lvert\sigma\rvert$	$\lvert pk\rvert$	Sign	Verify	ChgRep	Assumptions
[GHKP18]	$8\lvert\mathbb{G}_1\rvert + 6\lvert\mathbb{G}_2\rvert$	$2\lvert\mathbb{G}_1\rvert + (9+\ell)\lvert\mathbb{G}_2\rvert$	28E	**9**P	N/A	SXDH
[KSD19]	$8\lvert\mathbb{G}_1\rvert + 9\lvert\mathbb{G}_2\rvert$	$(\mathbf{2}+\ell)\lvert\mathbb{G}_2\rvert$	29E	11P	19P+38E	SXDH
Our work	$\mathbf{9}\lvert\mathbb{G}_1\rvert + \mathbf{4}\lvert\mathbb{G}_2\rvert$	$(\mathbf{2}+\ell)\lvert\mathbb{G}_2\rvert$	**10**E	11P	**19**P+**21**E	extKerMDH, SXDH

$\approx 25\%$ **shorter signature's size**

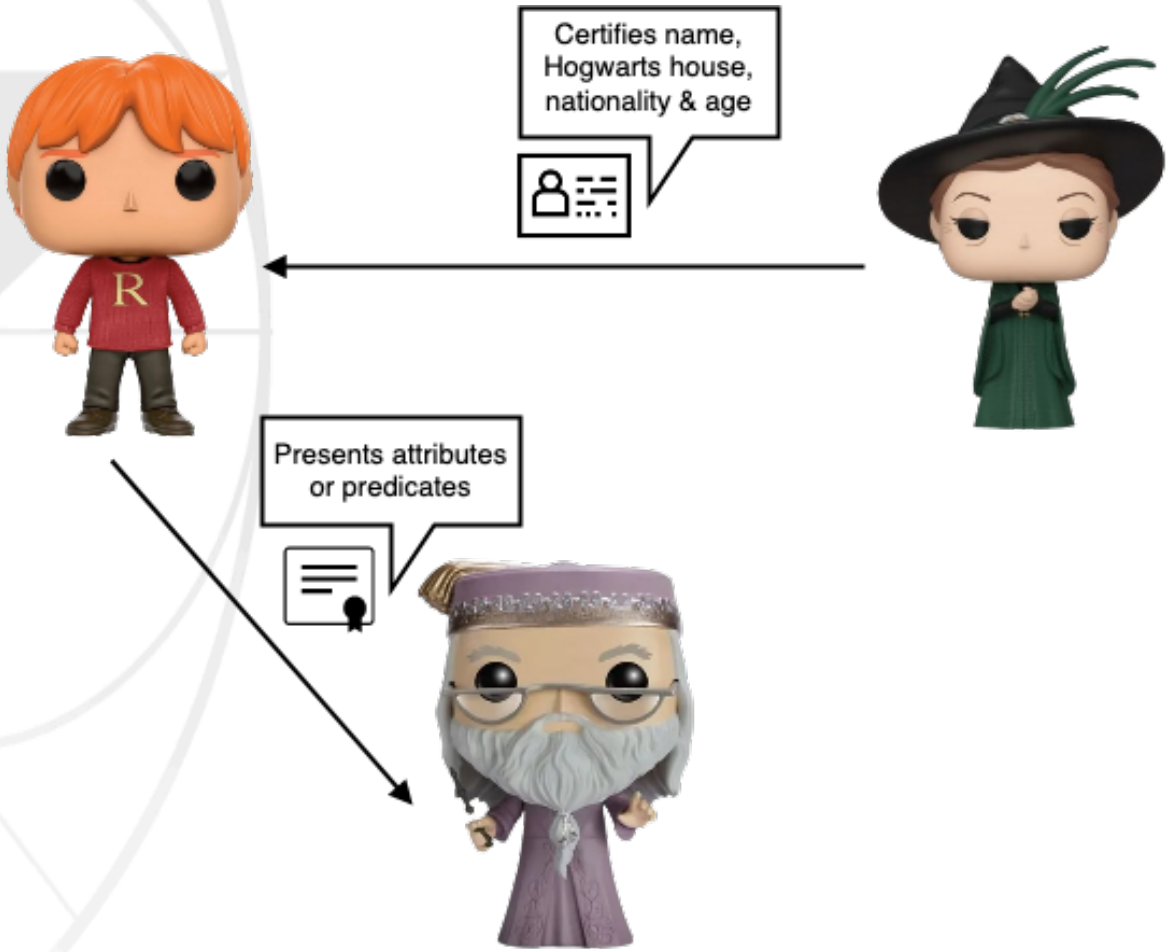
$\approx 50\%$ **less exponentiations for Sign and ChgRep**

Conclusions

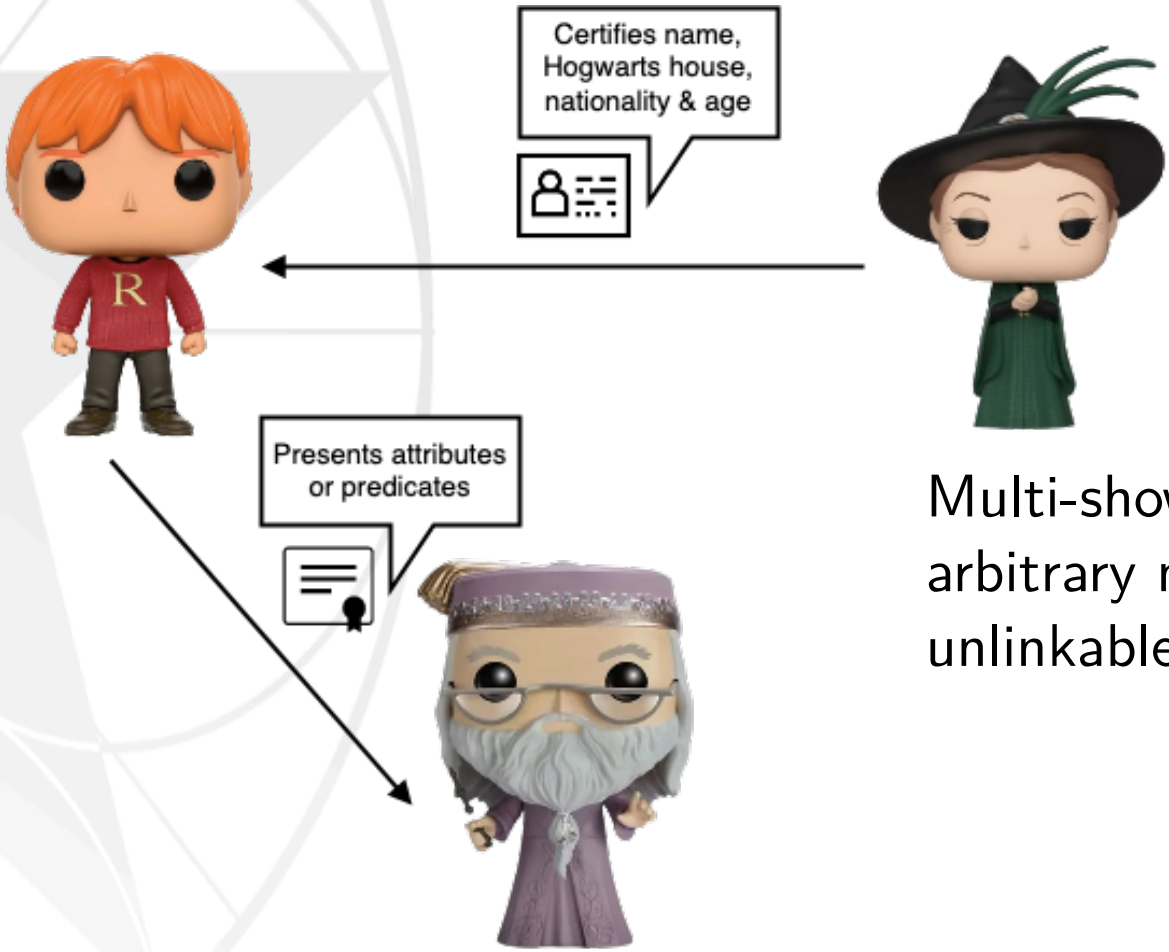
- Improved efficiency of SPS-EQ under standard assumptions
- First mercurial signature under standard assumptions
- We require a common reference string
- Our mercurial signature only achieves a weak form of perfect adaption

Outline

Attribute-based credentials: Interactions

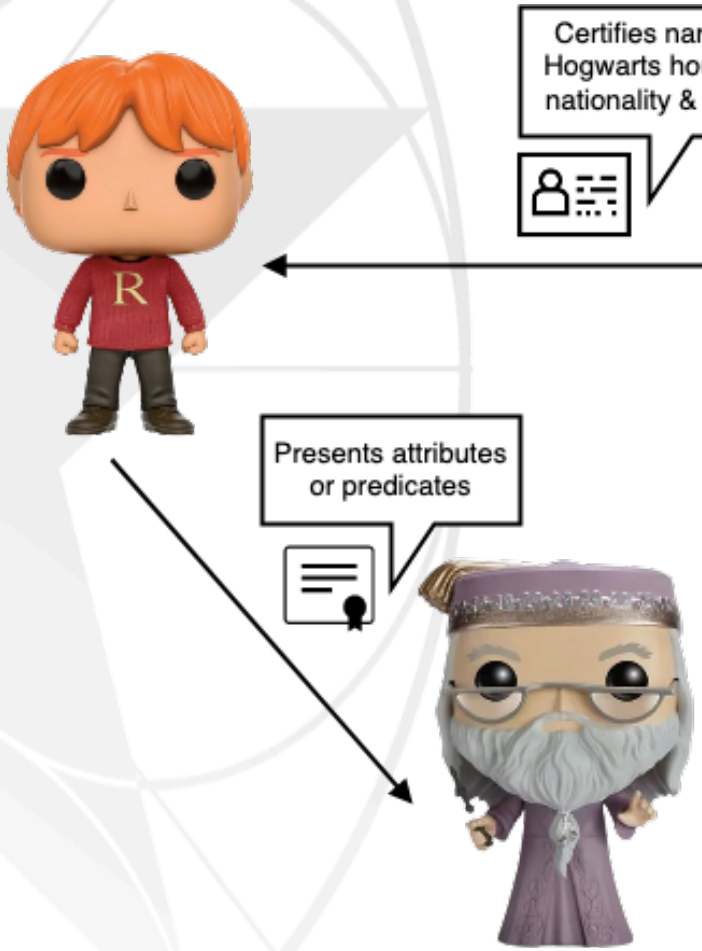


Attribute-based credentials: Interactions



Multi-show ABC's: arbitrary number of unlinkable showings

Attribute-based credentials: Interactions



Multi-show ABC's: arbitrary number of unlinkable showings

Multi-authority ABC's: single credential for attributes issued by multiple authorities

Attribute-based credentials: Differences

Expressiveness

Efficiency

Communication

Security model

Revocation

Attribute-based credentials: Lines of work

- Camenisch-Lysyanskaya signatures [CL04]: Idemix [Zur13] and [TG20]
- Aggregatable signatures: [CL11] and [HP20]
- Sanitizable signatures: [CL13]
- Redactable signatures: [CDHK15] and [San20]
- Structure-Preserving Signatures on Equivalence Classes: [DHS15] and [FHS19]
- **All previous constructions leak the issuer's identity**

Attribute-based credentials: The framework from [FHS19]

- Credentials are **signatures** on (randomizable) accumulators (**set of attributes**)
- Joint randomization of a message-signature pairs
- **Constant-size** showings
- Security properties:

Unforgeability

Anonymity

- Drawback: **only allows selective-disclosure of attributes**

Attribute-based credentials: Towards improved constructions

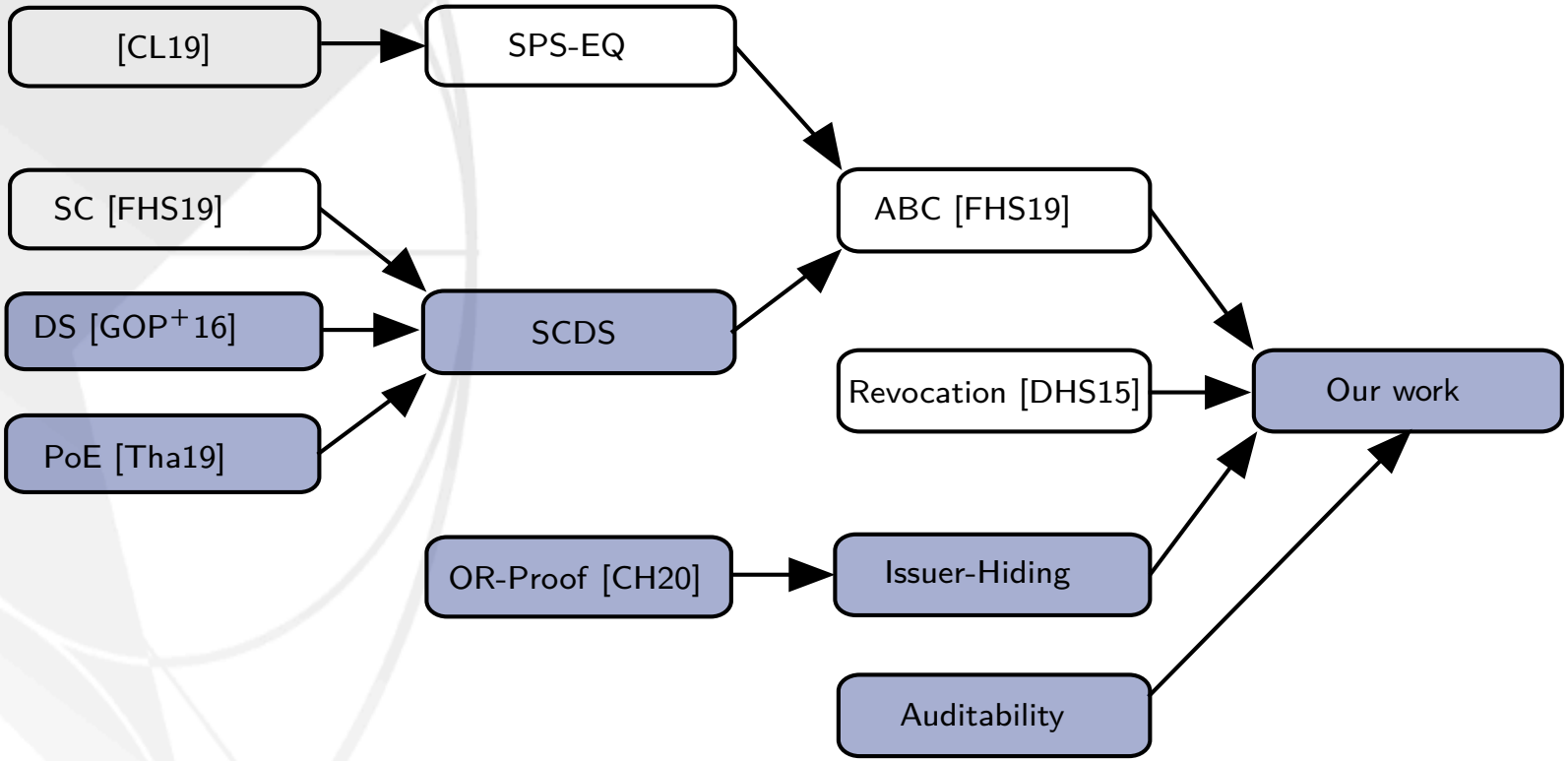
- We focused on improving the following aspects:

$\rightsquigarrow$ Extending the accumulator

$\rightsquigarrow$ Leveraging user/verifier costs

$\rightsquigarrow$ Including issuer-hiding and auditing features

Building blocks



Protego: Efficient, Revocable and Auditable Anonymous Credentials

- Extends [FHS19], including issuer-hiding and auditability
- Security properties:


Unforgeability

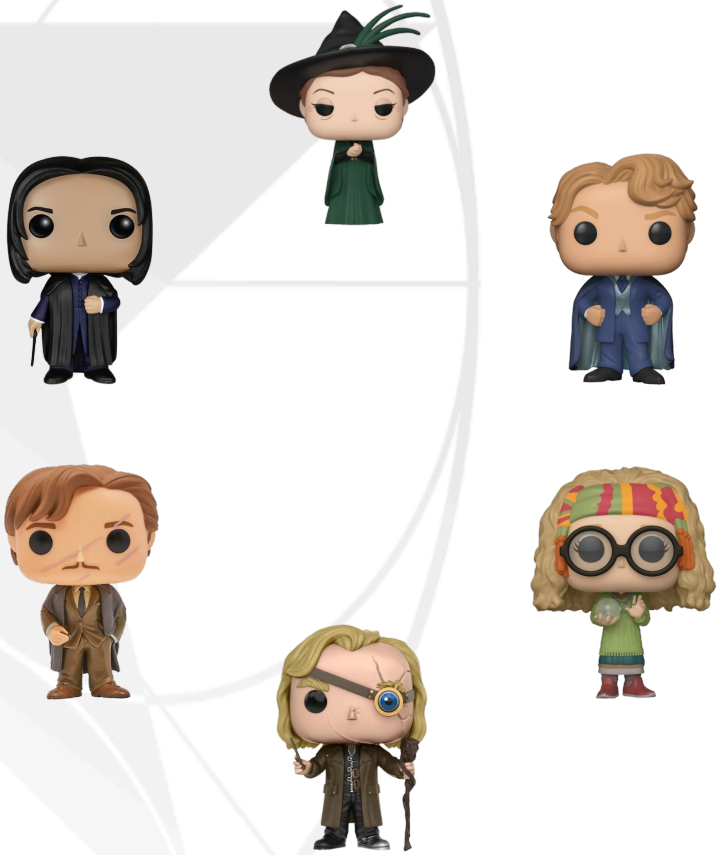

Anonymity


Issuer-hiding


Auditability

- Two flavours: Protego and Protego Duo

Protego



Issuer-hiding (**PKC '22**):

- Randomize the credential and issuer's public key consistently
- Use a fully adaptive 1-out-of-n NIZK argument (OR-proof)

Protego Duo

Issuer-hiding based on [BEK+21]:

- Define access policies as signatures on the issuers public keys
- Consistently randomize the credential with the access policy

Protego: Comparison with previous work

Scheme	[CL04]	[CL11]	[CL13]	[CDHK15] & [FHS19]	[TG20]	[San20]	[HP20]	**Our work**
	Issuing n-attr. credential							
Comm.	$O(n)$	$O(n)$	$O(n)$	**O(1)**	$O(n)$	**O(1)**	$O(n)$	**O(1)**
User	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Issuer	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
	Showing k-of-n attributes (selective disclosure)							
|ek|	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n^2)$	$O(n)$	$O(n)$
Comm.	$O(n)$	$O(1)$	$O(k)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$
User	$O(n)$	$O(n)$	$O(k)$	$O(n-k)$	$O(n-k)$	$O(n-k)$	**O(1)**	$O(\max\{n-k,k\})$
Verifier	$O(n)$	$O(n)$	$O(k)$	$O(k)$	$O(k)$	$O(k)$	$O(n)$	**O(1)**

Protego: Demonstration

Conclusions

- We extended previous work with new functionalities
- We obtained an efficient and versatile ABC scheme
- Our implementation confirms the feasibility of our proposals

Outline

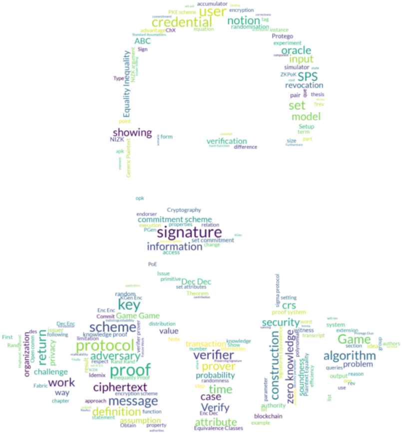
Conclusions

- Framework for generic randomisable encryption
- Generic plaintext equality and inequality proofs
- New SPS-EQ constructions under standard assumptions
- Extended previous ABC improving efficiency and functionalities
- Implemented our contributions showing their feasibility

Future Work

- Design non-interactive protocols for plaintext inequality
- Improve the number of rounds
- Build generic plaintext inequality tests ($<, \leq, \geq, >$)
- Constructions in the standard model without a common reference string
- Mercurial signatures satisfying a stronger notion of perfect adaption
- Explore other notions of equivalence classes and the use of aggregatable signatures with SPS-EQ


merci
thank you
gracias
obrigado
danke
grazie
go raibh maith agat
dziekuje
terima kasih
sagolun
chokrane
謝謝
mochchakkeram
dakujem
sukriya
bedankt
arigato
ngiyabonga
спасибо
мерси
barka
taiku
leat
tapadh
hvala
vinaka
grazzi
ありがとう
dank je
tesekkur ederim


credential
user
notion
oracle
input
SPS
revocation
set
model
showing
verification
Equality Inequality
signature
information
commitment scheme
key
scheme
protocol
adversary
proof
ciphertext
message
definition
assumption
return
privacy
challenge
work
verifier
prover
probability
time
case
Verify
attribute
security
construction
zero knowledge
soundness
Game
algorithm
problem
crs
Dec Dec