Wie können digitale Identitäten sowohl maximal datensparsam als auch skalierbar gestaltet werden?

Matthias Babel – FIM Research Centre, University of Bayreuth
Johannes Sedlmeir – SnT, University of Luxembourg

Why Digital Identities?

Digital Identities play a crucial role on the Internet

„The internet was built without an identity layer"

2018, Sovrin Foundation, Whitepaper: What goes on the ledger

"On the Internet, nobody knows you're a dog."

5 Jul 1993, The New Yorker, by Peter Steiner

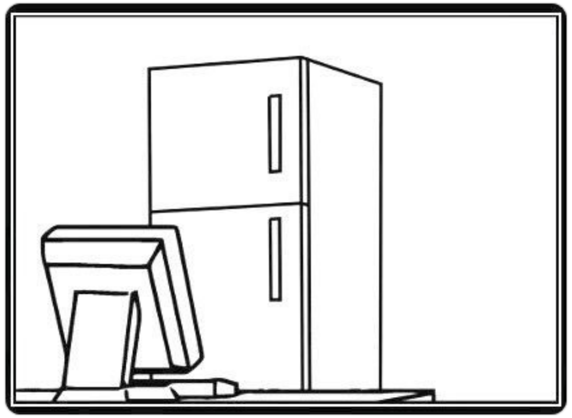
The Internet was built without a way to know who and what you are connecting to.

*Since this essential capability is missing, everyone offering an Internet service has had to come up with a workaround. It is fair to say that today's Internet, absent a native identity layer, is based on a **patchwork of identity one-offs**.*

May 2005, Kim Cameron, The Laws of Identity

Khalil Sehnaoui @sehnaoui Follow

On the Internet of Things, nobody knows you're a fridge ...
#IoT #Privacy #Anonymous



7:46 AM - 8 Dec 2015

Research Center Finance & Information Management | Universität Bayreuth | uni.lu | SnT

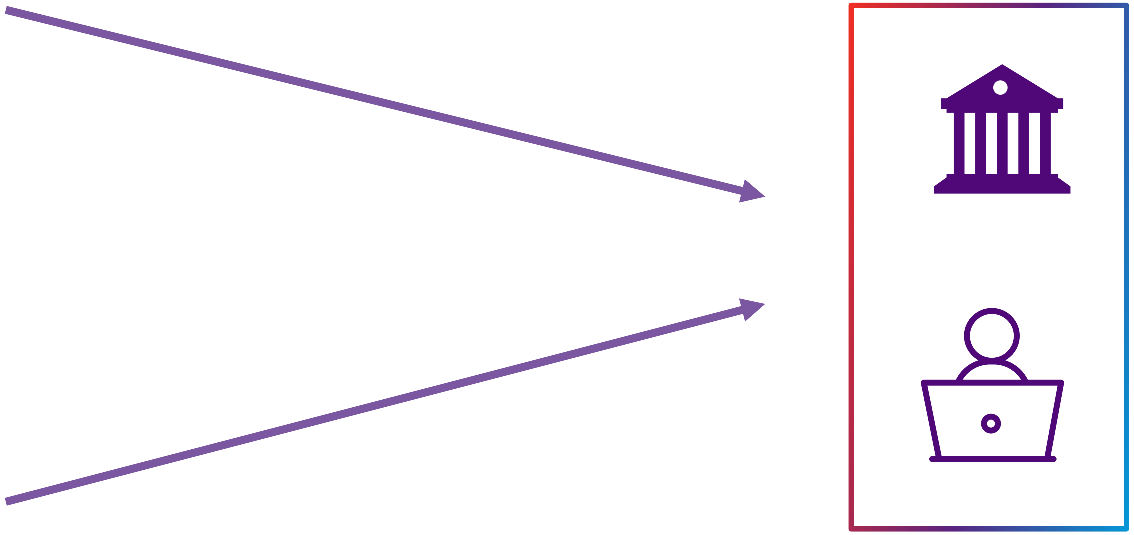
What is identity?

1 **Means for identification and authentication**

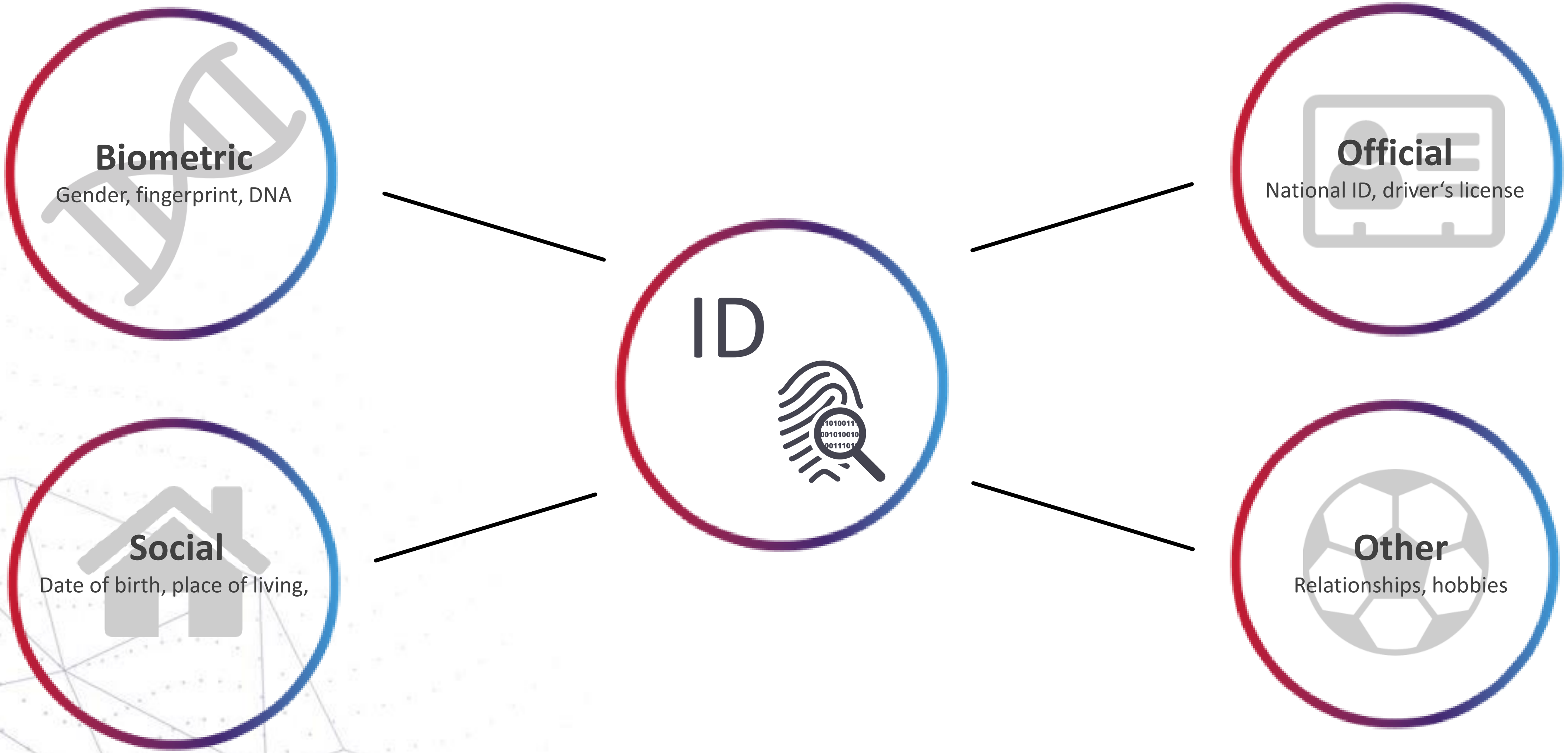
"I am really Alice!"

2 **Consisting of attributes**

Of which only some are relevant in a specific context!
&
Which sometimes need to be verifiable

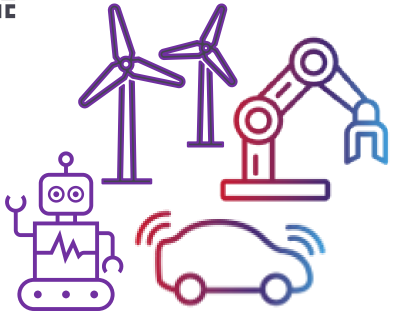


Identity is made up of attributes

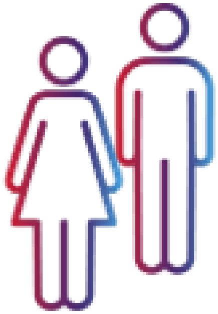


Digital identities play a central role in digital transformation in many areas

Digital identites for organizations



Digital identities for machines



Digital identities for individuals

Interoperability, portability, data minimization, automation

Application areas

(Master) data management

Identification

Authentication and authorization

Verifiable attestations

Today‘s dominant paradigms to digitize identity attributes

FRAGMENTED

Multiple apps and accounts -
insecure and/or inconvenient; little control for users over their identity attributes

FEDERATED

Single sign-on enabled by **corporate identity providers**
High convenience but limited control for users over their identity attributes

CENTRALIZED

Single sign-on enabled by **governmental identity providers**
Moderate convenience and limited control for users over their identity attributes

Tackling the challenges and shortcomings of today‘s digital identity landscape

Identity is monetized by large companies

Prevention of personal data monetization by large companies

Fragmented identity landscape

Avoidance of insecure identity data silos

Fraud and identity theft

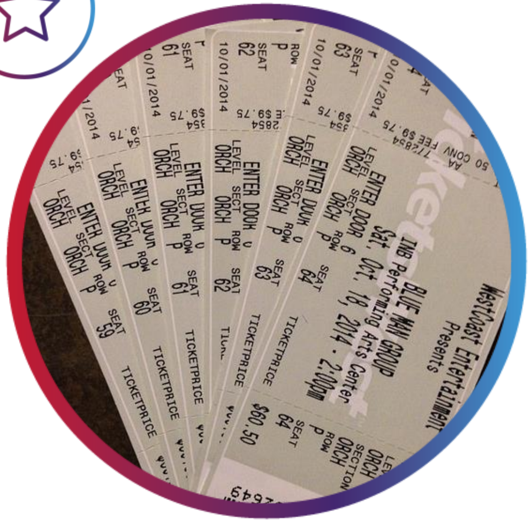
Reduced risk of identity theft

Abuse of personal data, and threats to privacy

Prevention of surveillance by governments and large companies

Good reasons for governments and industries to jointly work on the legal and technical support for secure digital identity systems

Requirements, inspired by today’s physical means of representing identity attributes



Must-have: Combination of machine-readable and machine-verifiable attestations to identity attributes in one app

Should-have: Selective disclosure (incl. comparisons)

Self-Sovereign Identity: Value system

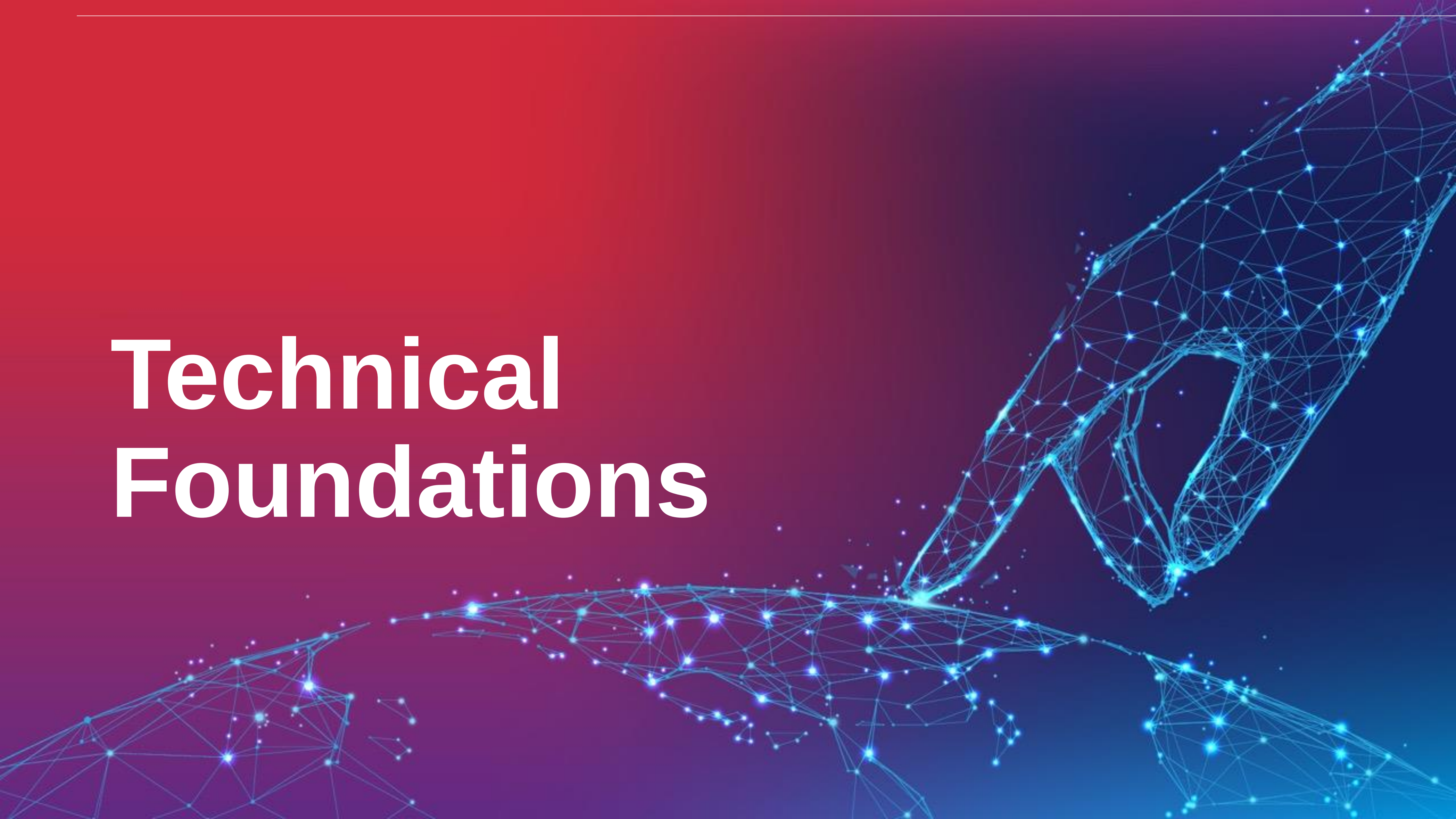
SELF-SOVEREIGN IDENTITY

Users conveniently self-manage their identity attributes in a digital wallet and control the disclosure of identity attributes to third parties

High convenience and control for users over their identity attributes

Initial "definition" of SSI: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html
(Existence, Access, Control, Transparency, Persistence, Portability, Interoperability, Consent, Minimalization)

Technical Foundations



Digital signatures can be used to convert physical proofs of identity into digital form

Digital certificate

Attributes:

Last name: Gabler

First name: Erika

…

Binding Key: 1234

Expiration: 31.10.2020

Signature: 5678

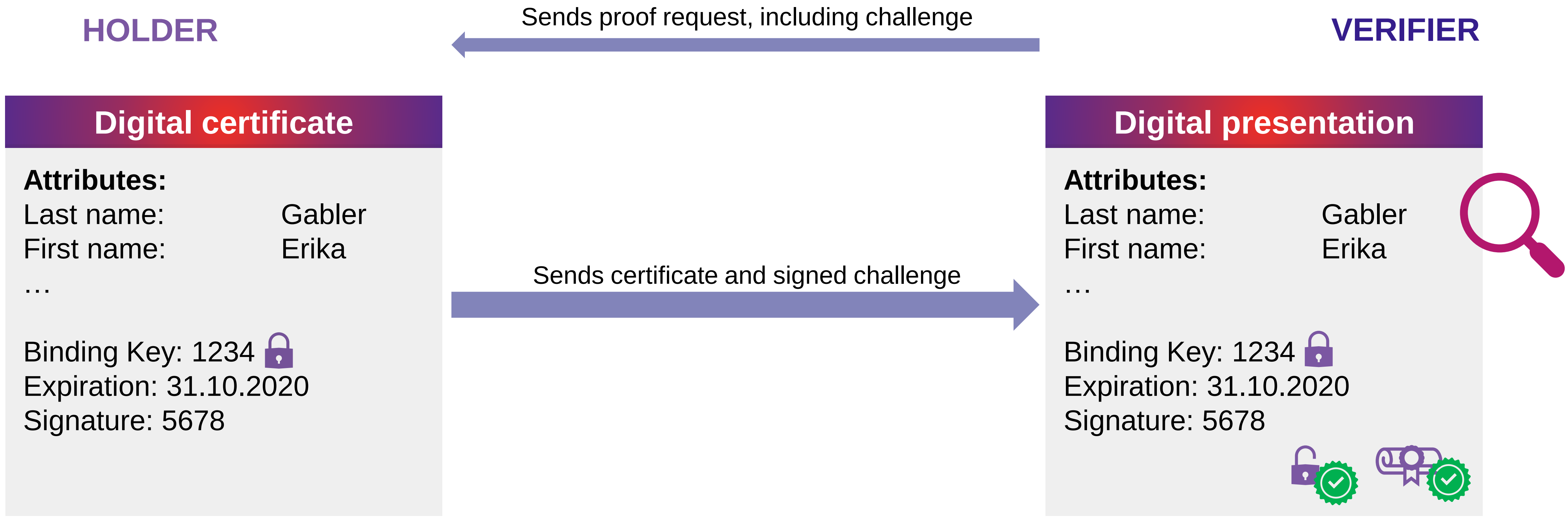
Research Center Finance & Information Management

UNIVERSITÄT BAYREUTH

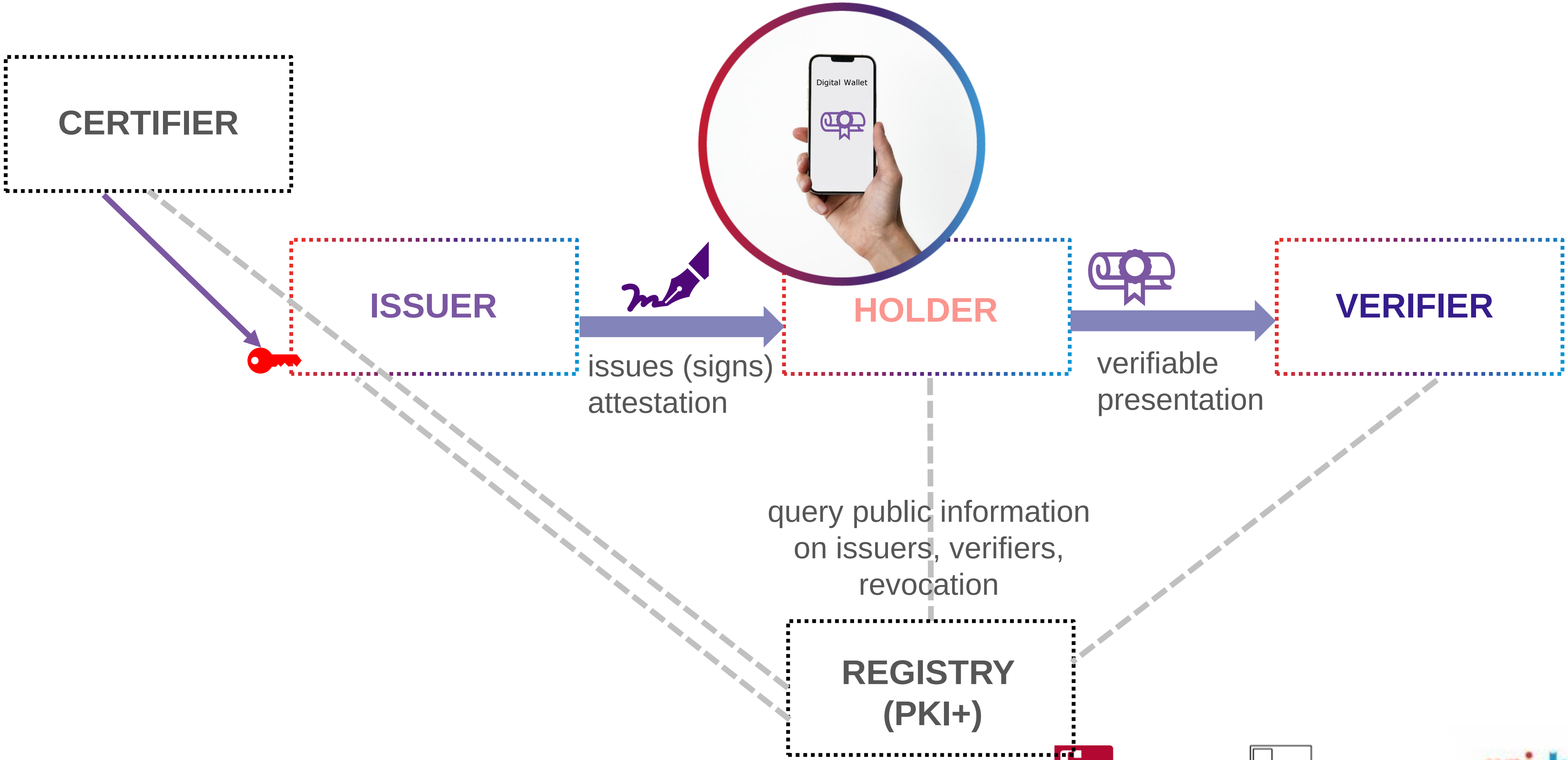
uni.lu

SnT

The verifiable presentation



Summary: Roles in digital identity



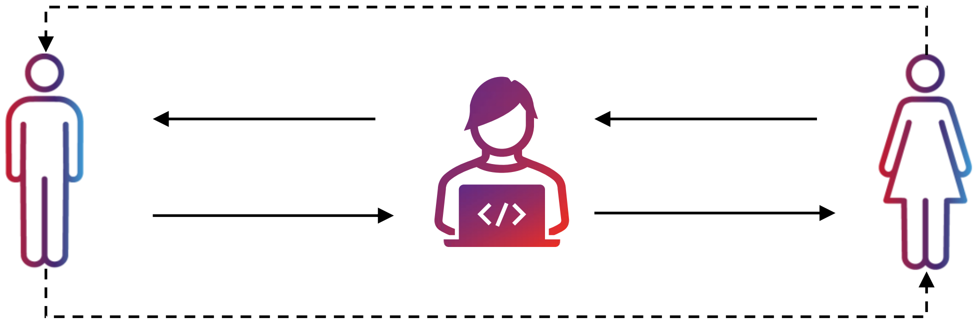
Recap: Challenges of SSI

1. Unique cryptographic identifiers
2. Digitally signed (verifiable) identity attributes
3. Blockchain?
4. Responsibility on the user side

Privacy

"… sophisticated marketing techniques that rely on profiles of individuals [...] being used to manipulate public opinion and elections" (Chaum, 1985, p. 1044)

Man-in-the-middle attacks



Zero-knowledge proofs for more privacy

Digital certificate

Attributes:
Last name: Gabler
First name: Erika
…

Binding Key: 1234
Expiration: 31.10.2020
Signature: 5678

Derived ZKP proves:

- Signature of the certificate is correct
- Certificate has not expired
- Certificate contains certain attribute values
- Checking the binding key…

Digital presentation

Attributes:
Last name: [redacted]
First name: Erika
[redacted]

Binding Key: [redacted]
Expiration: [redacted]
Signature: [redacted]

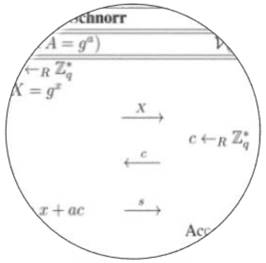
HOLDER ⟷ VERIFIER

Zero-knowledge proofs

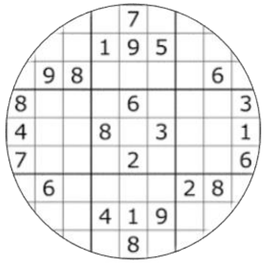
“those proofs that convey no additional knowledge other than the correctness of the proposition in question“ (GMR, 1985)

Examples:

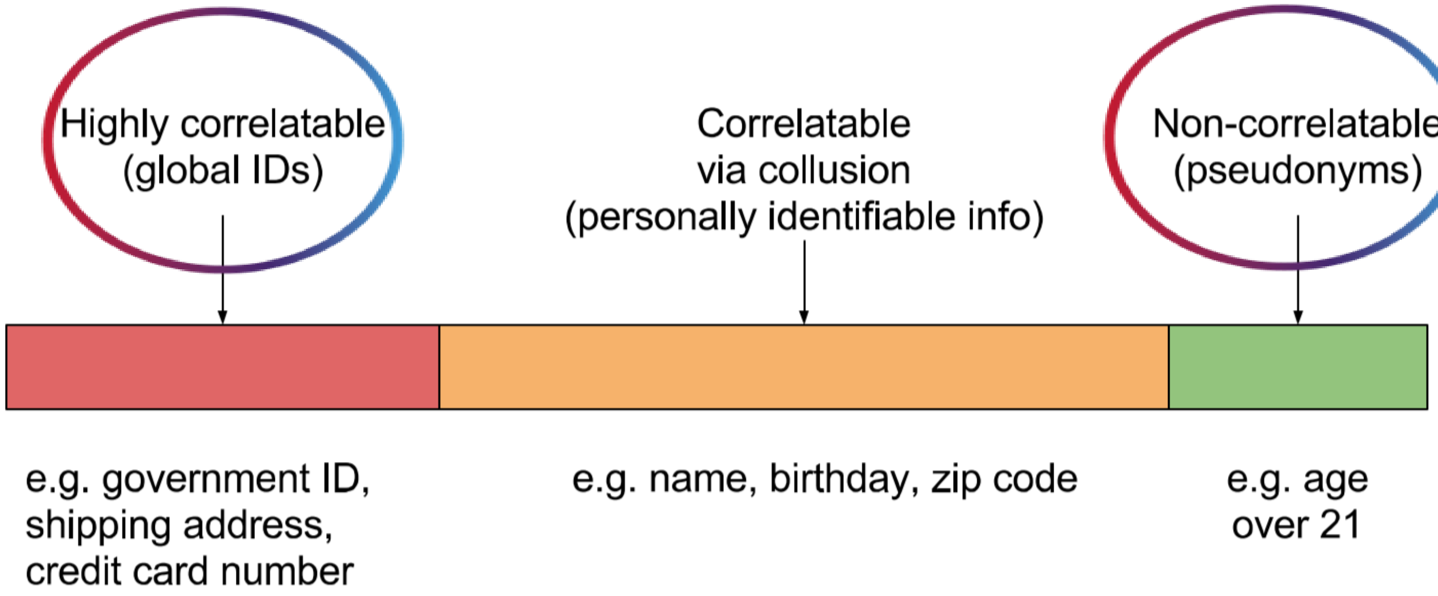
Proof of knowledge of a private key (associated with a public key), without leaking information that would allow or ease reconstructing the secret key.



Proof of knowledge of a solution to a given Sudoku puzzle, without revealing any information that would make it easier for the verifier to solve it.



The best technical solution cannot maintain privacy if one presents highly correlatable data



Taken from https://www.w3.org/TR/vc-data-model/

We researched limitations of privacy-oriented SSI implementations

Bringing data minimization to digital wallets at scale with general-purpose zero-knowledge proofs

Matthias Babel[a,b], Johannes Sedlmeir[b,c,*]

[a]*Branch Business & Information Systems Engineering of the Fraunhofer FIT, Bayreuth, Germany*
[b]*FIM Research Center, University of Bayreuth, Germany*
[c]*Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg*

Abstract

Today, digital identity management for individuals is either inconvenient and error-prone or creates undesirable lock-in effects and violates privacy and security expectations. These shortcomings inhibit the digital transformation in general and seem particularly concerning in the context of novel applications such as access control for decentralized autonomous organizations and identification in the Metaverse. Decentralized or self-sovereign identity (SSI) aims to offer a solution to this dilemma by empowering individuals to manage their digital identity through machine-verifiable attestations stored in a "digital wallet" application on their edge devices. However, when presented to a relying party, these attestations typically reveal more attributes than required and allow tracking end users' activities. Several academic works and practical solutions exist to reduce or avoid such excessive information disclosure, from simple selective disclosure to data-minimizing anonymous credentials based on zero-knowledge proofs (ZKPs). We first demonstrate that the SSI solutions that are currently built with anonymous credentials still lack essential features such as scalable revocation, certificate chaining, and integration with secure elements. We then argue that general-purpose ZKPs in the form of zk-SNARKs can appropriately address these pressing challenges. We describe our implementation and conduct performance tests on different edge devices to illustrate that the performance of zk-SNARK-based anonymous credentials is already practical. We also discuss further advantages that general-purpose ZKPs can easily provide for digital wallets, for instance, to create "designated verifier presentations" that facilitate new design options for digital identity infrastructures that previously were not accessible because of the threat of man-in-the-middle attacks.

Keywords: Anonymous credential, digital certificate, privacy, self-sovereign identity (SSI), verifiable computation, zk-SNARK.

fim Research Center Finance & Information Management
UNIVERSITÄT BAYREUTH
uni.lu SnT

Challenges of Hyperledger AnonCreds

Being "Real" about Hyperledger Indy & Aries / Anoncreds

September 7, 2022 By Kaliya Young

2. LACK OF TECHNICAL RIGOR

- An old, less performant signature algorithm that is not suitable for a new product

3. SERIOUS TECHNICAL, SCALABILITY AND GOVERNANCE ISSUES

- Indy & Aries / Anoncreds was constructed in a way that limited cryptographic agility or "upgradeability" or maintainability or extensibility or portability

1. LACK OF STANDARDIZATION AND WEAK STANDARD ALIGNMENT

https://identitywoman.net/being-real-about-hyperledger-indy-aries-anoncreds/

Core challenges of Hyperledger AnonCreds

Scalable private revocation

Scalability requirements of revocation registries in more detail

ID card

Vaccination passport

Credit card

Number of persons

N = 50,000,000

Revocation registry size	Number of Combinations k $k = N^3/R^3$	Herd privacy (k/N)
10,000	> 100,000,000,000	0.0004
32,768	> 3,500,000,000	< 0.02
100,000	125,000,00	0.4
1,000,000	125,000	400
10,000,000	125	400,000

Status quo in Hyperledger Aries/Indy

Indy Anoncreds status quo:

- Cap at 32,768 credentials per registry
- Relatively high proof generation time

Revocation Registry Size	Tail File Size	Proof Generation Time
3000	768KB	~4sec
10000	2.6MB	~5sec
32768	8.4MB	~7sec

https://github.com/bcgov/indy-tails-server

Alternatives have been discussed, but they are relatively complex and not deployed.

Current Timings

Tested on a 2017 Macbook Pro, with block size 1024. Further optimizations are yet to be applied:

- Create registry metadata: **0.25s**
  - performed once by the issuer when establishing a new registry
- Output a registry state for 1000 blocks, or about 1M credentials: **up to 2s**
  - performed by the issuer when publishing a new registry state
  - output a fully non-revoked block: **1.5ms**
  - output a partially revoked block: **2ms**
- Extract a non-revocation token for a partially-revoked block: **up to 0.5s**
  - performed by the prover after fetching a new registry state
  - TODO: explain method for deriving the witness and accumulator values
  - duration is expected to be reduced significantly
- Prepare a non-revocation token proof of knowledge: **5ms**
  - performed by the prover once per verification
- Verify token proof of knowledge: **12ms**
  - performed by the verifier

https://hackmd.io/kj223D1ZQN29WiusmnPFMA?view

Core challenges of Hyperledger AnonCreds

Scalable private revocation

Hardware-binding without "super cookie"

Private certificate chaining

… and complexity!

CL/BBS+ vs. general-purpose zero-knowledge proofs

MATHEMATICS

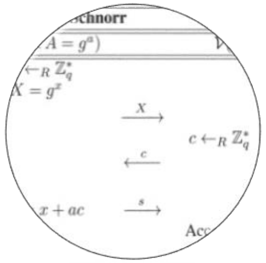
ENGINEERING

Zero-knowledge proofs

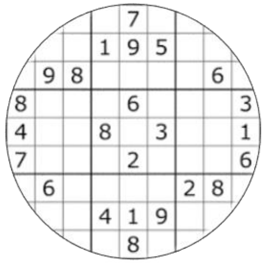
“those proofs that convey no additional knowledge other than the correctness of the proposition in question“
(GMR, 1985)

Examples:

Proof of knowledge of a private key (associated with a public key), without leaking information that would allow or ease reconstructing the secret key.



Proof of knowledge of a solution to a given Sudoku puzzle, without revealing any information that would make it easier for the verifier to solve it.



In general, (succinct) ZKPs certify the correct execution of an algorithm with a (very short) cryptographic attestation while revealing only explicitly selected inputs, intermediary results, and outputs.

Most popular short/efficient ZKPs:
zk-SNARKs (**succinct** non-interactive arguments of knowledge)

Modern zk-SNARK constructions

Polynomial commitment schemes

- KZG
- IPAs
- FRI + Vector commitment

Blinding factors

(Polynomial) interactive oracle proofs

- Linear PCP
- PlonK
- AIR

Efficient zk-SNARK

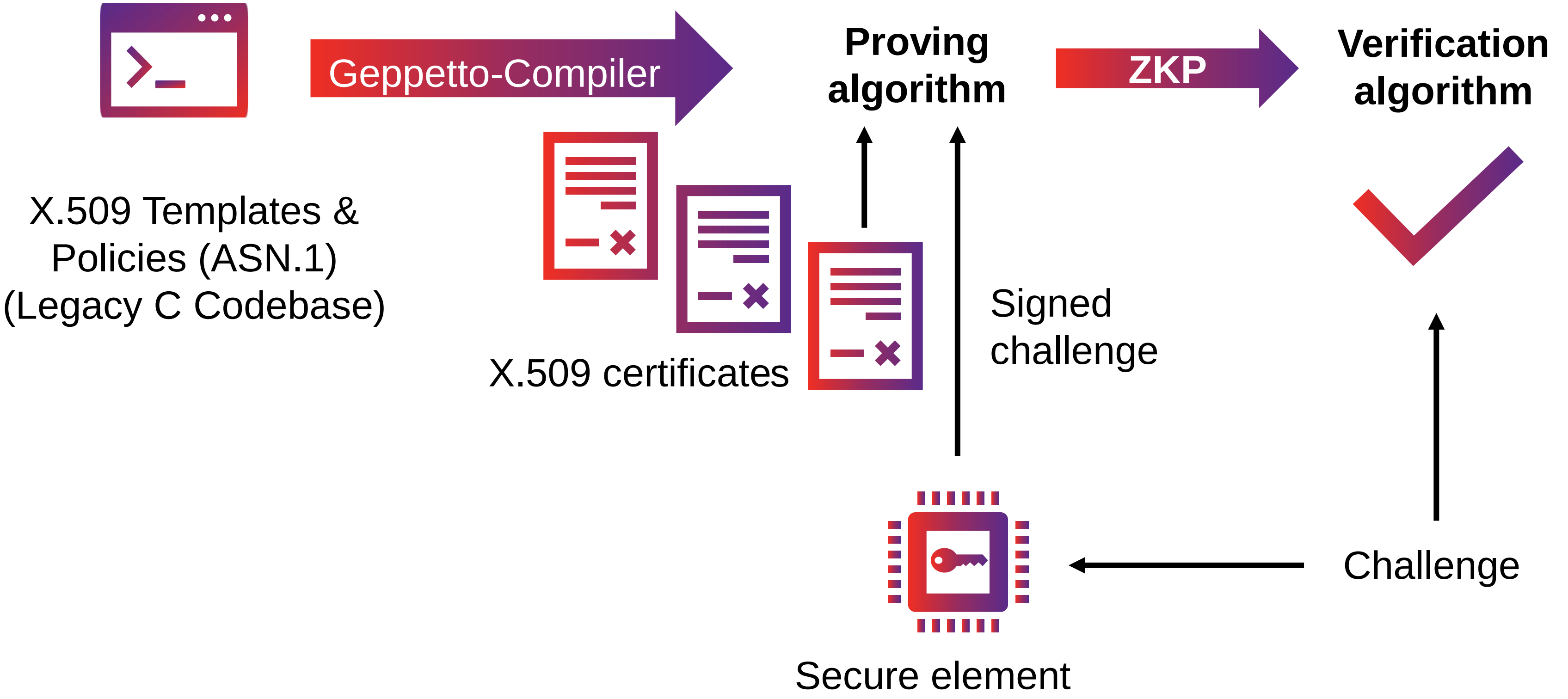
Related work

2016 IEEE Symposium on Security and Privacy

Cinderella: Turning Shabby X.509 Certificates into Elegant Anonymous Credentials with the Magic of Verifiable Computation

Antoine Delignat-Lavaud Cédric Fournet Markulf Kohlweiss Bryan Parno
{antdl,fournet,markulf,parno}@microsoft.com
Microsoft Research

Cinderella in detail



- ✓ Selective disclosure
- ✓ Private credential chaining
- ✓ Private holder binding with secure element (no super cookie)
- ✓ Fast verification (8ms) and constant proof size (288 bytes)
- ✓ OCSP-based proof of non-revocation

- ❑ Slow proving time (2016: ~10 minutes on a desktop PC with 4 cores)
- ❑ Little focus on end-user related applications (e-voting example)

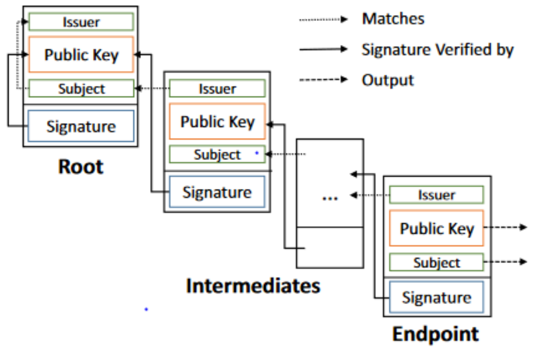
Delignat-Lavaud, A., Fournet, C., Kohlweiss, M. and Parno, B., 2016. Cinderella: Turning shabby X. 509 certificates into elegant anonymous credentials with the magic of verifiable computation. In *IEEE Symposium on Security and Privacy* (pp. 235-254).

Related work

2016 IEEE Symposium on Security and Privacy

Cinderella: Turning Shabby X.509 Certificates into Elegant Anonymous Credentials with the Magic of Verifiable Computation

Antoine Delignat-Lavaud Cédric Fournet Markulf Kohlweiss Bryan Parno
{antdl,fournet,markulf,parno}@microsoft.com
Microsoft Research



- ✓ Private holder binding with secure elements, private credential chaining
- ✓ Covers existing standards (ANS.1) and revocation protocols (OCSP)
- ❑ High proving time (several minutes)
- ❑ Hardly discussion of end-user related application, general predicates, revocation registries, identification of the relying party

zk-creds: Flexible Anonymous Credentials from zkSNARKs and Existing Identity Infrastructure

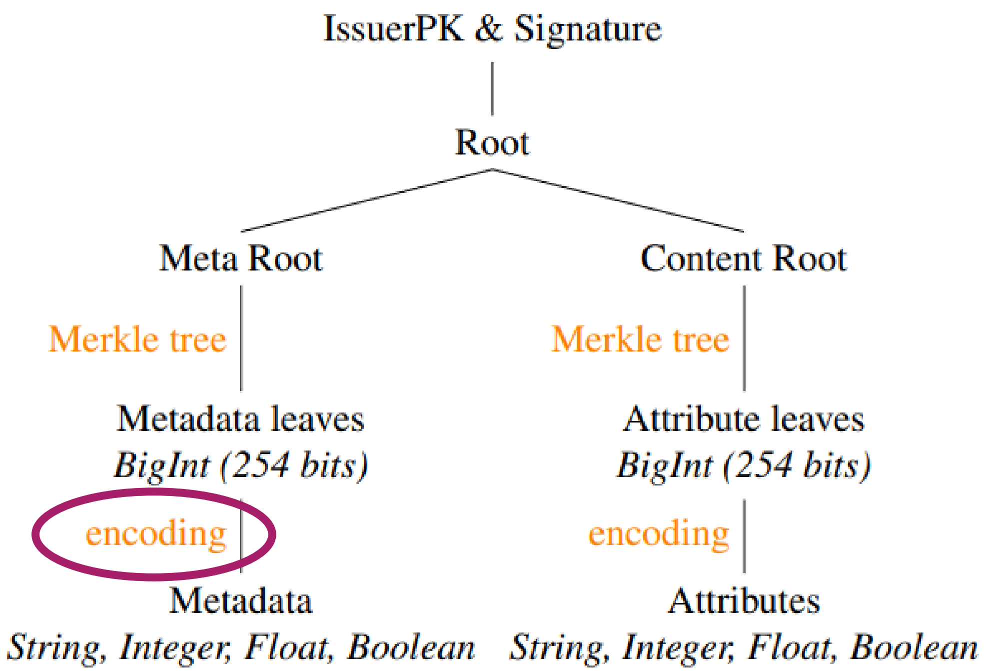
Michael Rosenberg[1], Jacob White[2], Christina Garman[2], and Ian Miers[1]

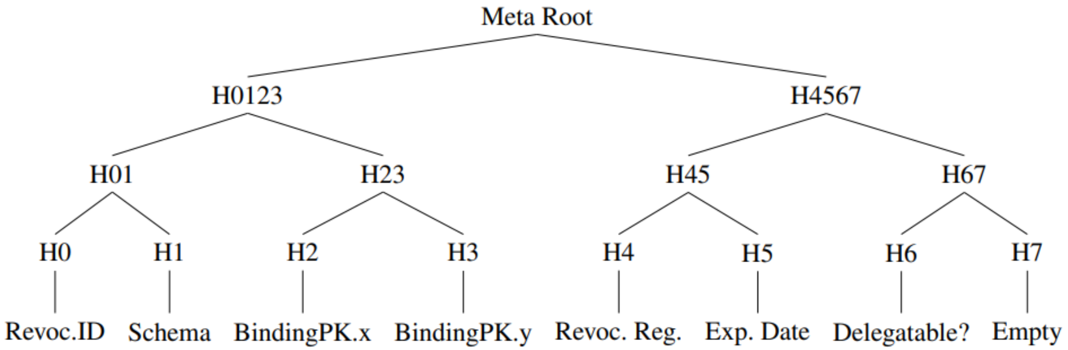
[1]University of Maryland
[2]Purdue University
[1]{micro,imiers}@umd.edu
[2]{clg, white570}@purdue.edu

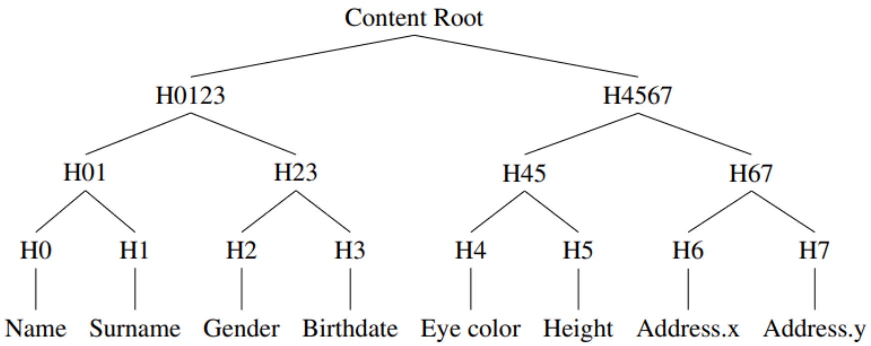
July 4, 2022

- ✓ Discuss arbitrary predicates (also cross-credential)
- ✓ Formal security proofs
- • Use SNARK-friendly primitives
- • Yet another standard
- • Merkle forests instead of digital signatures
- ❑ No discussion of scalable revocation
- ❑ No discussion of the identification of relying parties

Credential design is flexible







Statements to be proved

- Integrity (signature on the Merkle root) with respect to some public key
- Metadata:
  - Holder binding: Capability to sign the verifier's challenge with respect to binding key
  - Revocation: Set-membership for the credential ID (could be also set non-membership) in some accumulator
  - Expiration: Range proof for expiration date with respect to verifier's date
- Attribute data:
  - Selective disclosure: Merkle proofs for attributes
  - Advanced predicates (e.g., polygon inbound proof)

Complexity of the corresponding proofs

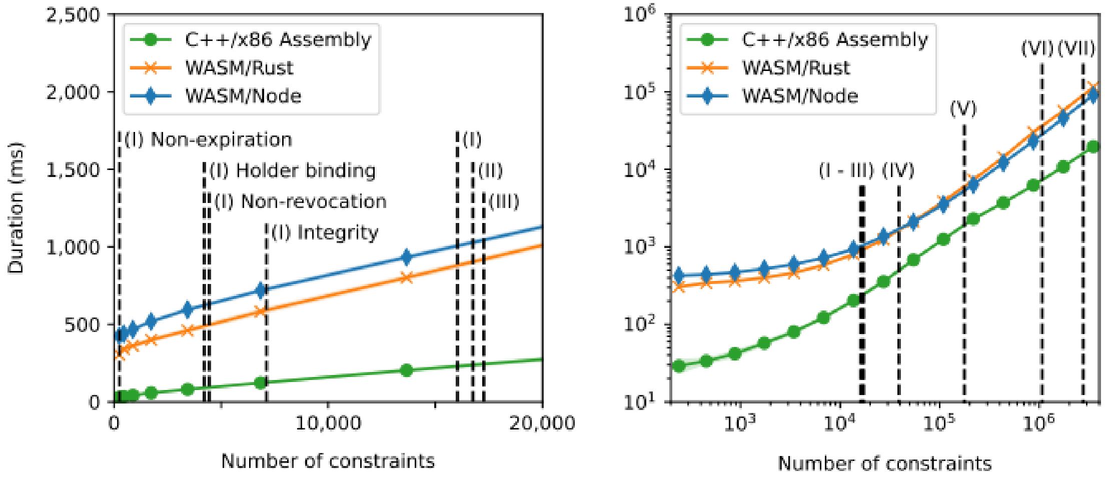
I. Standard, single attribute (Poseidon, EdDSA, 8 attributes, revocation registry size: 2M)
II. All attributes
III. Revocation registry size: 65M
IV. Three chained credentials
V. ECDSA-based holder binding (secure element)
VI. SHA256 and ECDSA everywhere
VII. Three chained credentials with SHA256 and ECDSA

Building blocks		**Number of occurrences in corresponding scenario and contribution to number of constraints**													
Component	**# constraints**	**I**		**II**		**III**		**IV**		**V**		**VI**		**VII**	
		# Occurrences	Constr.	# Occ.	Constr.	# Occ.	Constr.	# Occurrences	Constr.	# Occ.	Constr.	# Occ.	Constr.	# Occ.	Constr.
Selector	5	4 + 13 = 17	85	17	85	22	110	4 + 3 * 13 = 43	215	17	85	17	85	43	215
Range proof	252	1	252	1	252	1	252	3	756	1	252	1	252	3	756
Division with rest	252	1	252	1	252	1	252	3	756	1	252	1	252	3	756
Poseidon hash	240	1 + 4 + 7 + 13 = 25	6,000	28	6,720	30	7,200	4 + 3 * (1 + 7 + 13) + 2 * 2 = 71	17,040	25	6000	0	0	0	0
extractKthBit	1,012	1	1,012	1	1,012	1	1,012	3	3,036	1	1,012	1	1,012	3	3,036
EdDSA signature	4,218	1 + 1 = 2	8,436	2	8,436	2	8,436	1 + 3 * 1 = 4	16,872	1	4,218	0	0	0	0
SHA256 hash	29,636	0	0	0	0	0	0	0	0	0	0	25	740,900	71	2,104,156
ECDSA signature	163,239* (1,508,136)	0	0	0	0	0	0	0	0	1	163,239	2	326,478	4	652,956
Total number of constraints			**16,037**		**16,757**		**17,262**		**38,915**		**175,058**		**1,068,979**		**2,761,875**

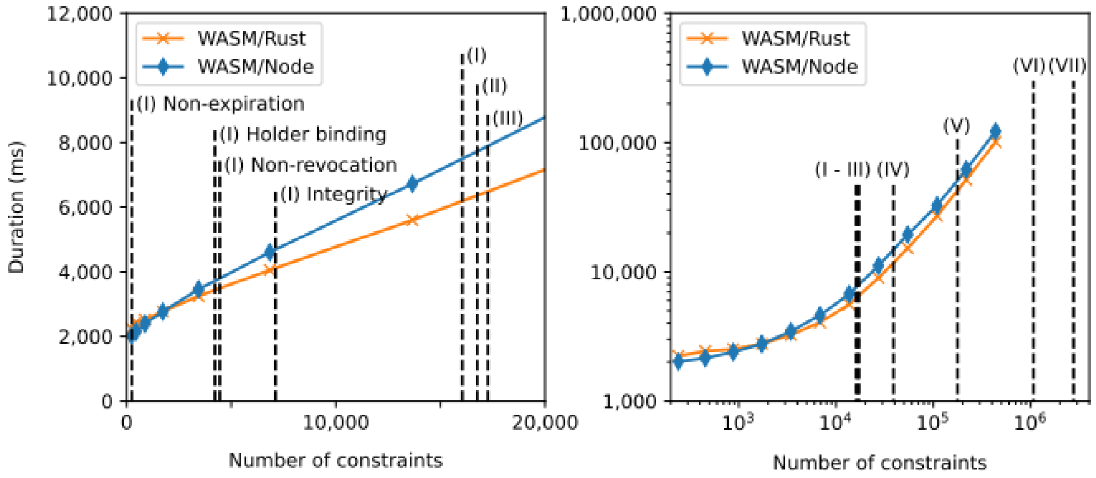
Table 1: Number of constraints for the most relevant basic building blocks and their occurrence in the basic scenarios. * with preprocessing inputs.

Proving performance on a Laptop and a Raspberry Pi

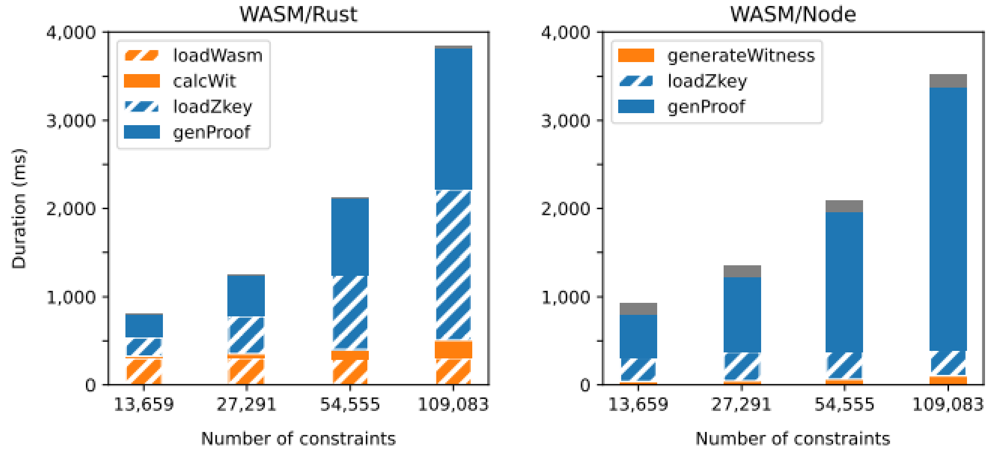
Proving times on a Laptop (Dell Precision 3571)



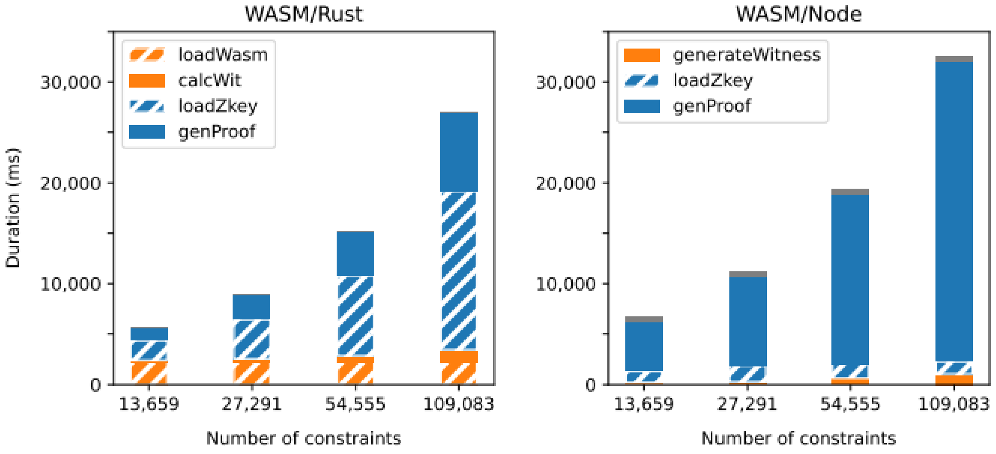
Proving times on a Raspberry Pi 4B



Detailed performance tests on a Laptop (Dell Precision 3571)



Detailed performance tests on a Raspberry Pi 4B



Proving performance on a mobile phone

- Own experiments: Around 5 (high-end device) to 40 (low-end device) seconds for scenarios I – III in the Browser and as a react-native app; around 1 to 3 seconds with Rust on a mobile phone
- Other publication: 6s on a single thread with 62,000 constraints (around 4x more than in scenario I): https://doi.org/10.1109/SP40001.2021.00038
- We are currently trying to deploy Rust-based proof generation libraries (Ark-Circom, Ark-Groth16, Spartan, Plonky2?) on mobile phones. We already succeeded in fast witness generation (C++) and some of the libraries (e.g., Plonky2 is extremely fast with SHA256).
- We keep an eye on new proof systems, new prover implementations, hardware acceleration (GPU support), optimizations of constraint systems for cryptographic primitives, …

One more significant challenge…

Remember why the ID-wallet failed?

- Performance – not a real technical issue
- Threat of man-in-the-middle attacks!

Encryption does not help!

Not an AnonCreds, but a general issue (even for the eID)!

Even a very restrictive certification would face opposition (“signed identity attributes“)!

One solution: Certification of the verifier
→ SSL-certificates, QWAC, CVCA-issued, …

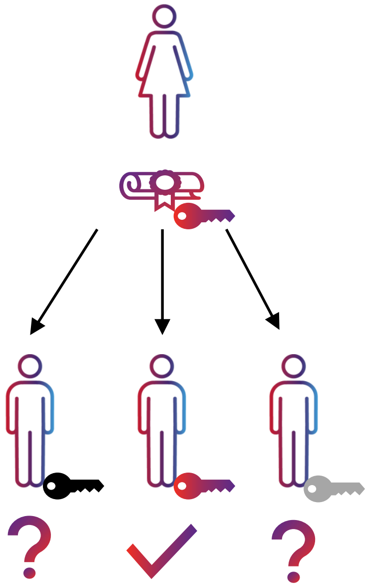
Tradeoff: Either low entry barriers (no “control“) or low security

Different requirements for different attributes are complex to implement/govern and dangerous (“escalation of privileges“)

→ Fundamental problem: Verifiable presentation only bound to challenge, not to verifier‘s identity!

Designated verifier presentations to save the day…

Proof: Either my claims about my credential / identity are correct, or I know the verifier‘s private key (used for encrypted communication). Complex to design in general (e.g., for CL/BBS+), but almost trivial and at negligible performance costs with zk-SNARKs.
Holder creates designated verifier proof and encrypts it with the designated verifier’s public key.



Case A: Attacker puts its own public key (and the replayed nonce) in the proof request → Attacker can decrypt the message with the proof (and would accept the proof). But if the proof is forwarded to the legitimate verifier, this verifier will not accept because it is not the designated verifier.

Case B: Attacker puts the legitimate verifier‘s public key (and the replayed nonce) in the proof request → Attacker cannot decrypt and re-encrypt the proof to send it to the legitimate verifier.

→ Nice side effect: Identity attributes are only verifiable for the legitimate verifier (receiver).

→ Nicer side effect: If we think about the eID, it is almost as concerning as digitally signed data: One can create ZKPs about attributes exchanged in a TLS-based connection (https://doi.org/10.1145/3372297.3417239) (eID!!). Designated verifier ZKPs could fix this problem.

Research Center Finance & Information Management
UNIVERSITÄT BAYREUTH

Next steps / work in progress

Further exploring zk-SNARK performance on mobile phones.

UC security proofs for the construction of anoncreds with plug-in predicates.

Can we find a proof that covers CL/BBS+, SNARK-based approach, and hybrids (Lego-SNARK)?

Optimized designs (commitments or digital signatures, accumulators for revocation, …).

Formal proofs that designated verifier ZKPs can address man-in-the-middle attacks without certification of the relying party and pose less privacy problems when facing TLS oracles.

Implications on user experience: Waiting times, human-readable description of what is revealed?

The Moral Character of Cryptographic Work*

Phillip Rogaway

Department of Computer Science
University of California, Davis, USA
rogaway@cs.ucdavis.edu

December 2015
(minor revisions March 2016)

Abstract. Cryptography rearranges power: it configures who can do what, from what. This makes cryptography an inherently *political* tool, and it confers on the field an intrinsically *moral* dimension. The Snowden revelations motivate a reassessment of the political and moral positioning of cryptography. They lead one to ask if our inability to effectively address mass surveillance constitutes a failure of our field. I believe that it does. I call for a community-wide effort to develop more effective means to resist mass surveillance. I plead for a reinvention of our disciplinary culture to attend not only to puzzles and math, but, also, to the societal implications of our work.

Keywords: cryptography · ethics · mass surveillance · privacy · Snowden · social responsibility

Time for questions

Matthias Babel
matthias.babel@fim-rc.de

fim
Research Center
Finance & Information Management

UNIVERSITÄT BAYREUTH

Johannes Sedmeir
johannes.sedlmeir@uni.lu

uni.lu | SnT

Our updated SSI principles

Control

Only the actual controller has decision-making power over their digital identity

Representation

SSI can represent any entity digitally – human, legal, or technical

Flexibility

No vendor lock-in, focus on interoperable standards, and open-source projects

Usability

Success and durability factors

Security

State-of-the-art cryptographic tools and end-to-end encrypted interactions

Reliability

Guidance for verifiers on whether to trust issuers in a highly dependable infrastructure

Privacy

In each interaction, only the data essential for its purpose is revealed

Authenticity

Credentials are bonded to their initial bearers

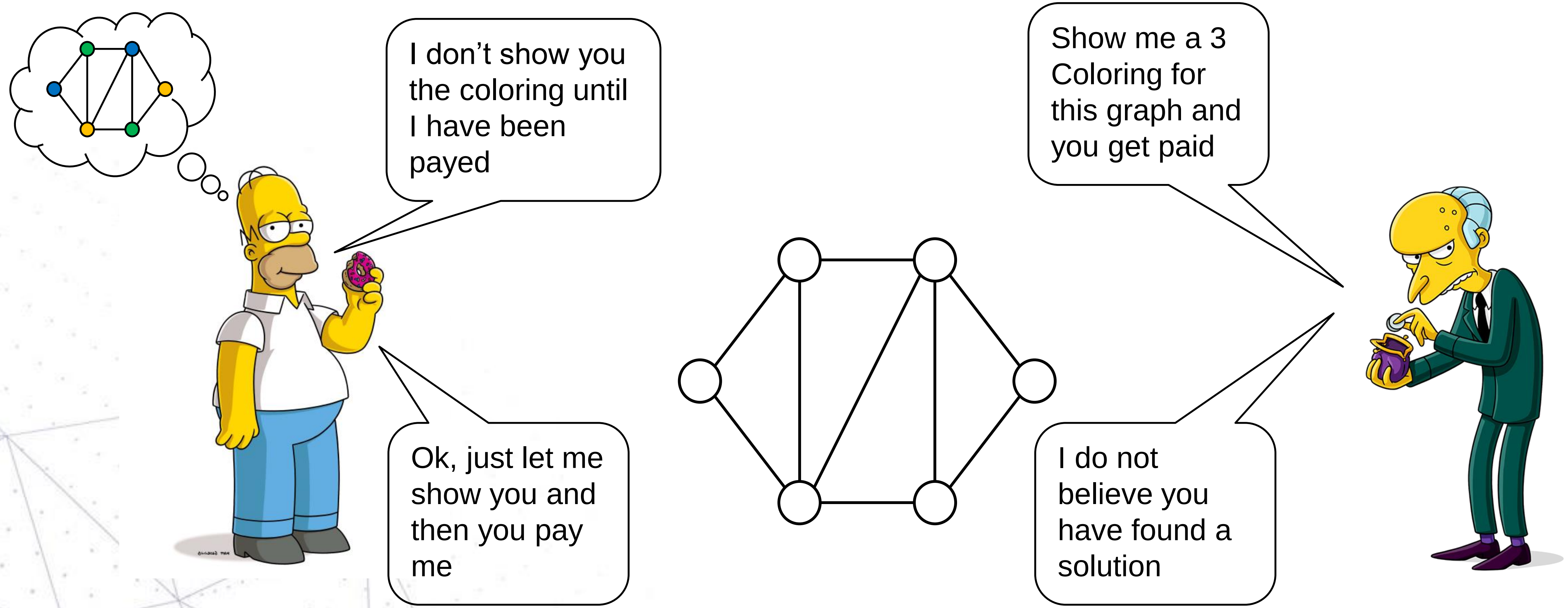
Verifiability

The validity and timeliness of credentials can be checked efficiently

Johannes Sedlmeir, Tom Barbereau, Jasmin Huber, Linda Weigl, & Tamara Roth (2022): Transition pathways towards design principles of self-sovereign identity.
In: Proceedings of the 43rd International Conference on Information Systems (ICIS)

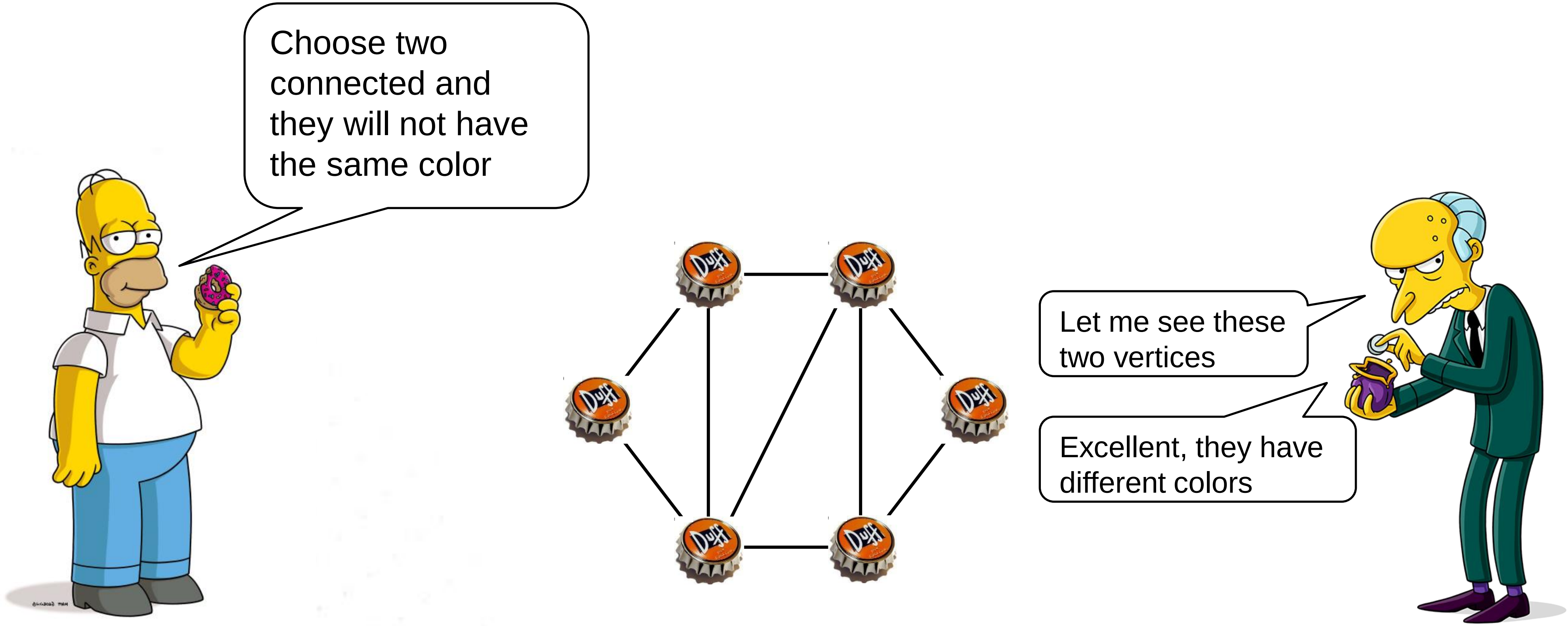
fim Research Center Finance & Information Management | UNIVERSITÄT BAYREUTH | uni.lu | SnT

How do general-purpose ZKPs work?



Graph 3 Coloring: Coloring a graph using only 3 colors,
such that not any two connected vertices have the same color

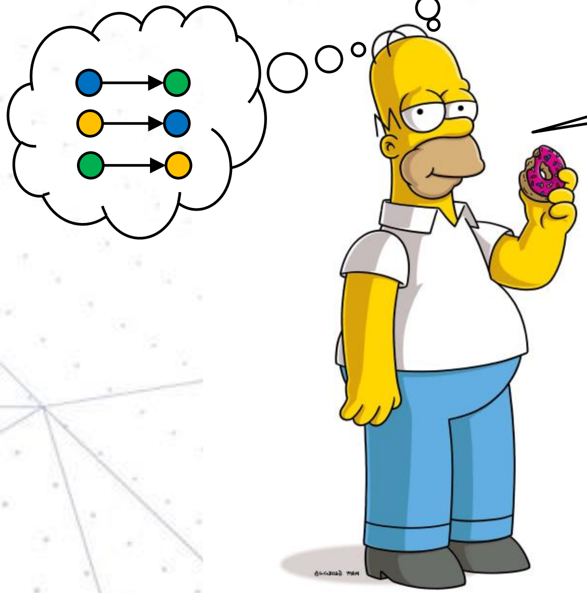
How do general-purpose ZKPs work?

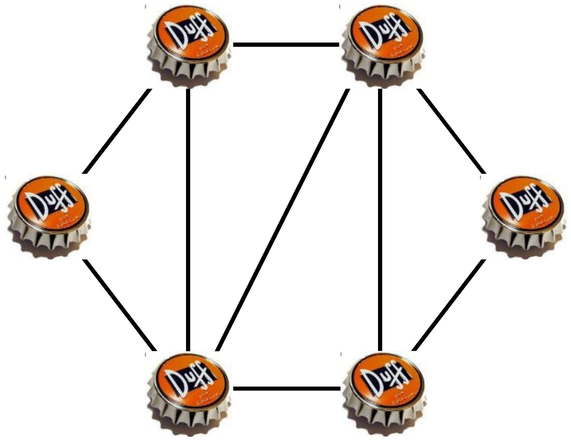


How do general-purpose ZKPs work?

If I show him the same coloring several times, he might figure it out

Choose two connected and they will not have same color



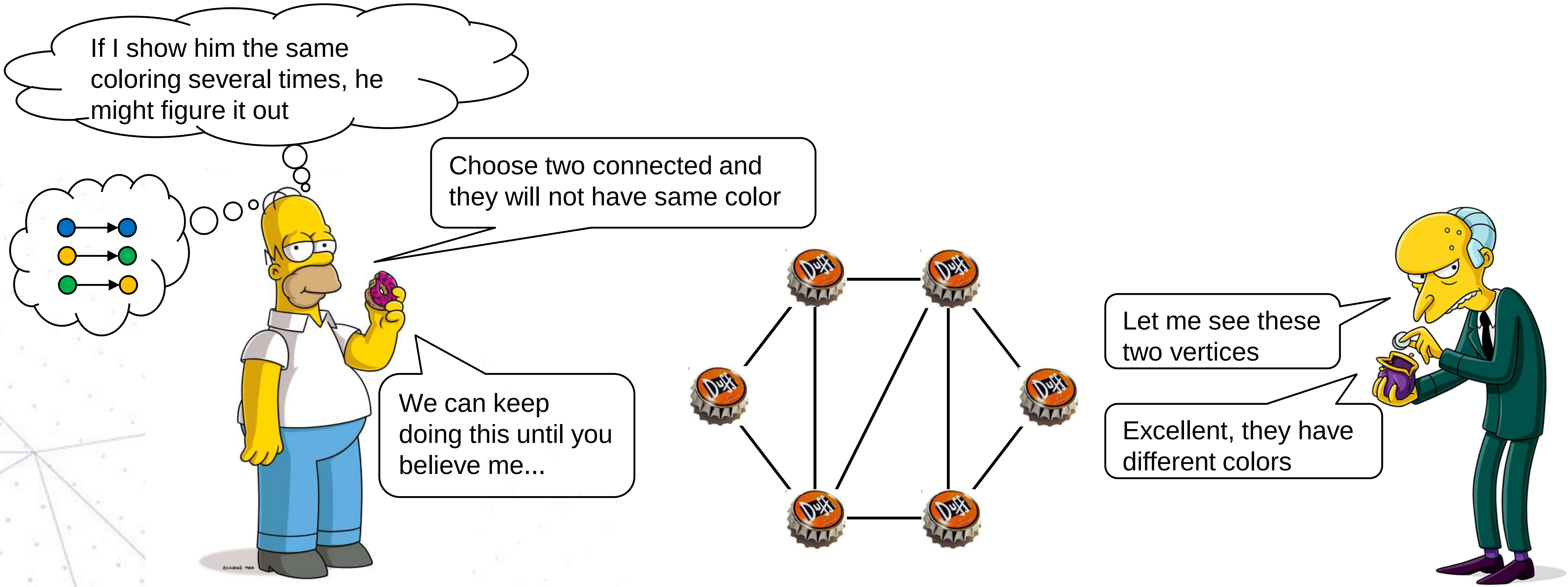


Let me see these two vertices

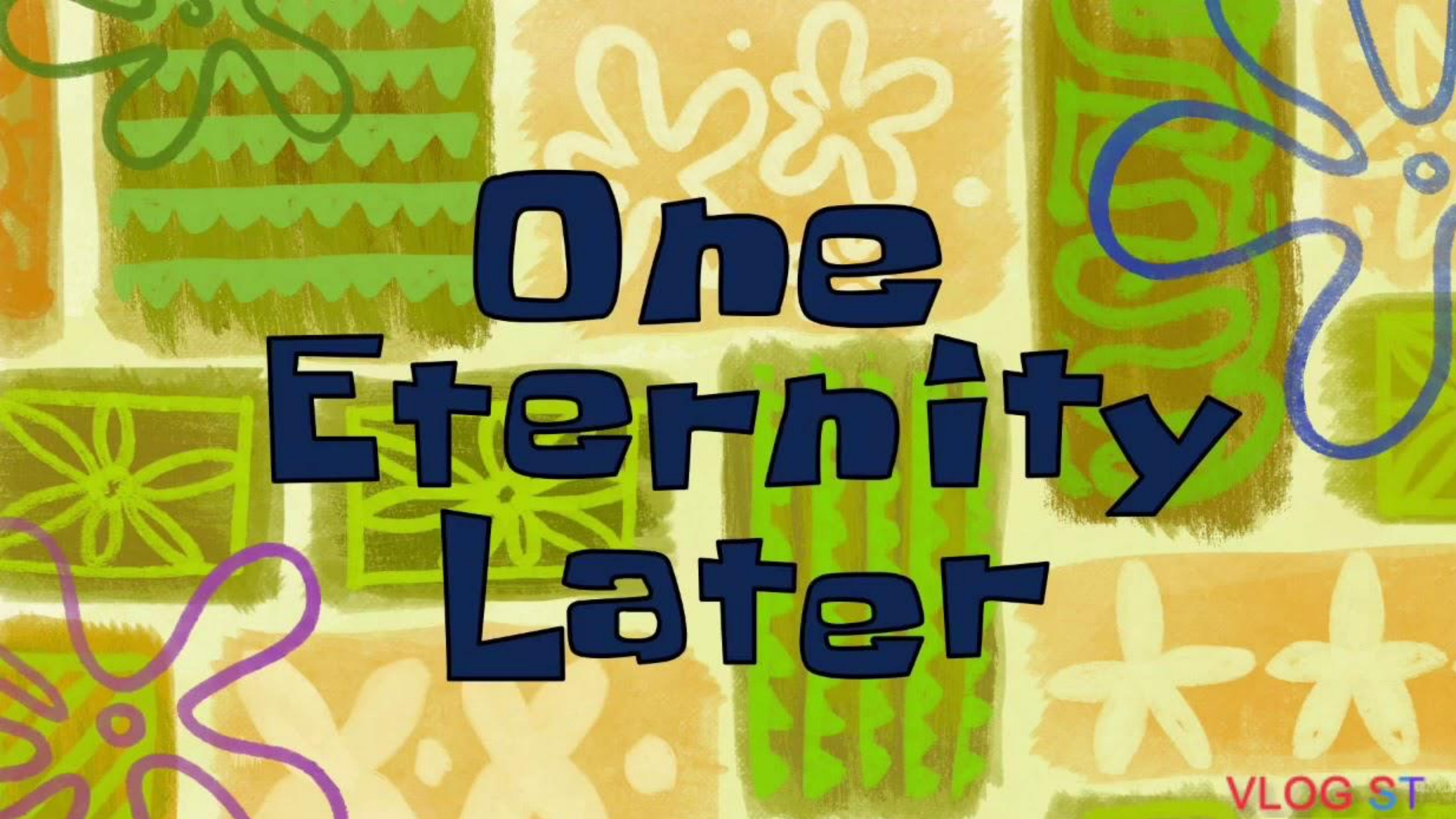
Excellent, they have different colors

Permutation: Renaming (shuffling) of vertices

How do general-purpose ZKPs work?



Permutation: Renaming (shuffling) of vertices


One
Eternity
Later
VLOG ST

How do general-purpose ZKPs work?