FINE-GRAINED AND CONTROLLED REWRITING IN BLOCKCHAINS

Chameleon Hashing Gone Attribute-Based

David Derler (DFINITY), Kai Samelin (TÜV), Daniel Slamanig (AIT), Christoph Striecks (AIT)

TALK OUTLINE

- Motivate problem on **editing/re-writing** distributed ledgers (DLs)
- Solution in form of a **new cryptographic primitive**:

Policy-Based Chameleon Hashing (PBCH)

- **Instantiation** from known cryptographic building blocks
- High-level example for **fine-grained redactable transactions** in DLs
- First performance **evaluations**

RESEARCH IN DISTRIBUTED LEDGERS TECHNOLOGIES

- Massive progress beyond Bitcoin, very hyped in recent years

- Signs that hype is turning into extensive research within the *cryptographic* community
  - **(Cryptographic) research centers** are established, e.g., CBR Stanford, CBRC Aarhus, ABC Austria

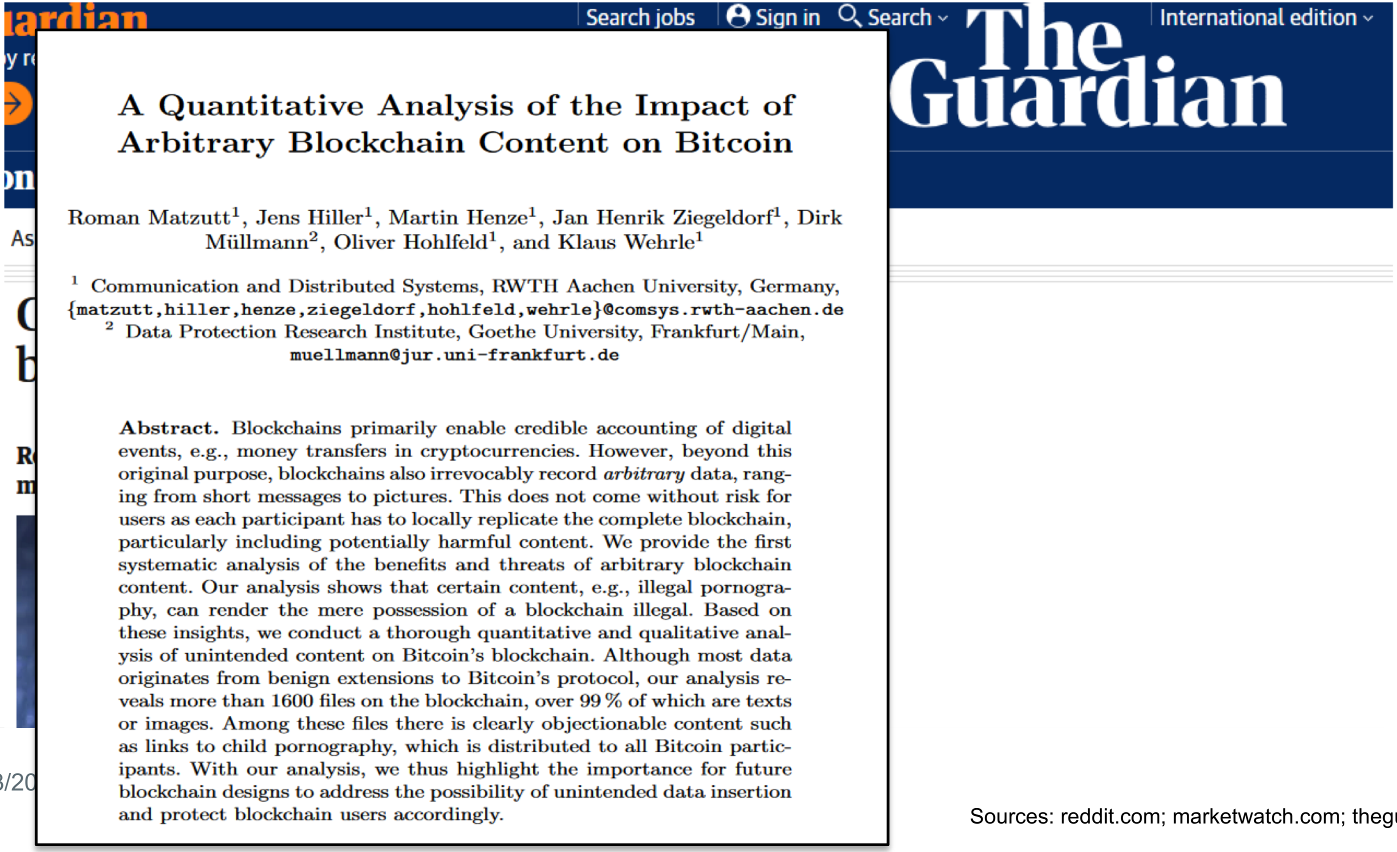
- **Many Cryptographic building blocks** are applied to DLs
  - ZK-SNARKs, Multi-Signatures, Verifiable Random Functions/Delay Functions/Secret Sharing, Threshold Signatures, Multi-Party Computation, …

- Less research is known on **rewriting DLs** …

  » … wait, isn't that counterintuitive?

IMMUTABLE DATA IN THE BLOCKCHAIN

AUSTRIAN INSTITUTE OF TECHNOLOGY



A Quantitative Analysis of the Impact of Arbitrary Blockchain Content on Bitcoin

Roman Matzutt[1], Jens Hiller[1], Martin Henze[1], Jan Henrik Ziegeldorf[1], Dirk Müllmann[2], Oliver Hohlfeld[1], and Klaus Wehrle[1]

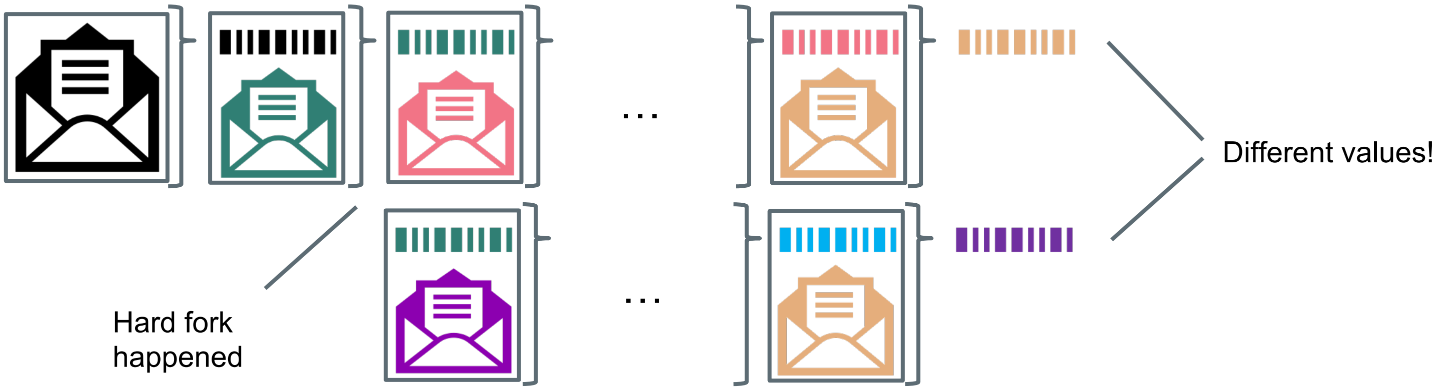
[1] Communication and Distributed Systems, RWTH Aachen University, Germany, {matzutt,hiller,henze,ziegeldorf,hohlfeld,wehrle}@comsys.rwth-aachen.de
[2] Data Protection Research Institute, Goethe University, Frankfurt/Main, muellmann@jur.uni-frankfurt.de

Abstract. Blockchains primarily enable credible accounting of digital events, e.g., money transfers in cryptocurrencies. However, beyond this original purpose, blockchains also irrevocably record *arbitrary* data, ranging from short messages to pictures. This does not come without risk for users as each participant has to locally replicate the complete blockchain, particularly including potentially harmful content. We provide the first systematic analysis of the benefits and threats of arbitrary blockchain content. Our analysis shows that certain content, e.g., illegal pornography, can render the mere possession of a blockchain illegal. Based on these insights, we conduct a thorough quantitative and qualitative analysis of unintended content on Bitcoin's blockchain. Although most data originates from benign extensions to Bitcoin's protocol, our analysis reveals more than 1600 files on the blockchain, over 99 % of which are texts or images. Among these files there is clearly objectionable content such as links to child pornography, which is distributed to all Bitcoin participants. With our analysis, we thus highlight the importance for future blockchain designs to address the possibility of unintended data insertion and protect blockchain users accordingly.

Sources: reddit.com; marketwatch.com; theguardian.com

JUST DO A HARD FORK …

- Simple solution: **hard forks**, but *not* really useful (i.e., chain from change point has to be “re-written”)



AUSTRIAN INSTITUTE OF TECHNOLOGY

RESEARCH MOTIVATION OF DL EDITS

- Ateniese, Magri, Venturi, Andrade (EuroS&P 2017) motivated to **rethink** immutable DLs:
  - **Illegal** or **improper content** occurs, **intellectual properties** unclear
  - New versions of **smart contracts** unclear
  - **Right to be Forgotten** may be legally required, e.g., by the **EU's GDPR**
  - **But:** redactions should be rare events
- Ateniese et al. proposed a solution on **block level** using **chameleon hashing** replacing essential ingredient of DLs, i.e., hash function
- Deuber et al. (S&P 2019) propose alternative solution also on **block level**

In **this work**, focus is on **transaction-level** rewriting.

PROTOTYPE OF EDITABLE BLOCKCHAINS

SEPTEMBER 20, 2016

Accenture Debuts Prototype of 'Editable' Blockchain for Enterprise and Permissioned Systems

Invention addresses blockchain 'immutability' challenges for permissioned systems, including the legal 'right to be forgotten,' human error, illegal actions

Co-developers Accenture and Dr. Giuseppe Ateniese register U.S. and E.U. patents

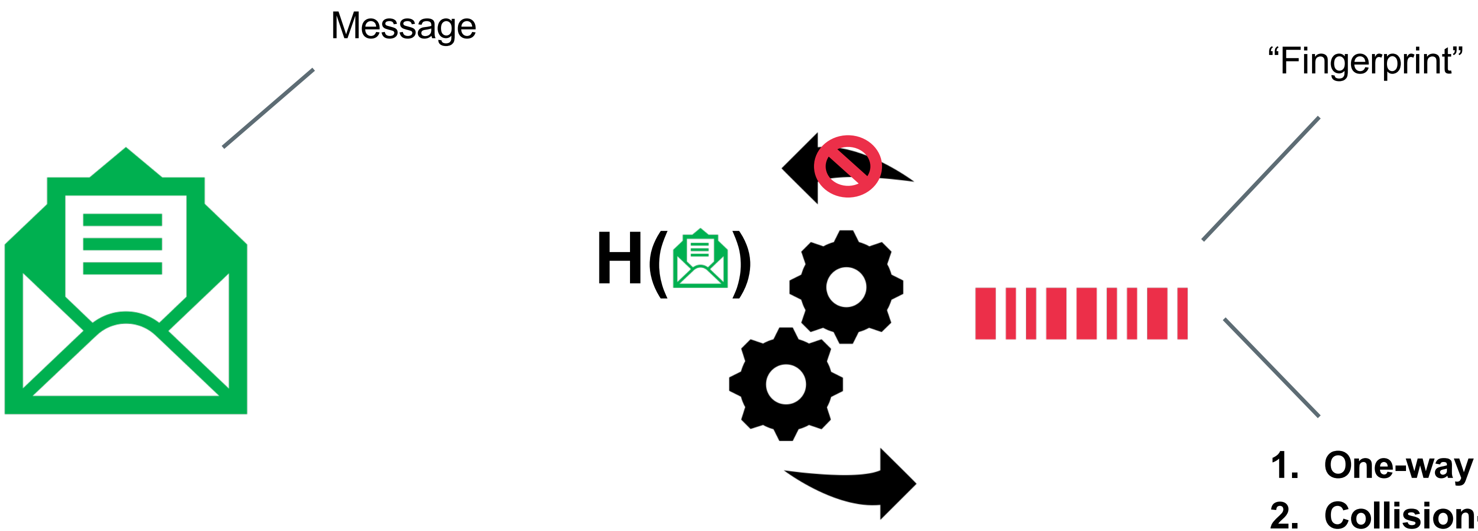
wrong and to meet new and changing regulatory and legal requirements, like the 'right to be forgotten' and other data-privacy and retention rules. An editable form of blockchain will make the technology more practical and useful for enterprise systems and accelerate its adoption. It combines the confidence that comes from immutability with the pragmatism required in an imperfect world."

"The clever work of the bitcoin creators and leaps of progress in applied cryptographic research are opening the door to bold new uses of blockchain," said Dr. Giuseppe Ateniese, a leading

CHAMELEON HASHING

Finding collisions for hash functions (if you know a trapdoor)

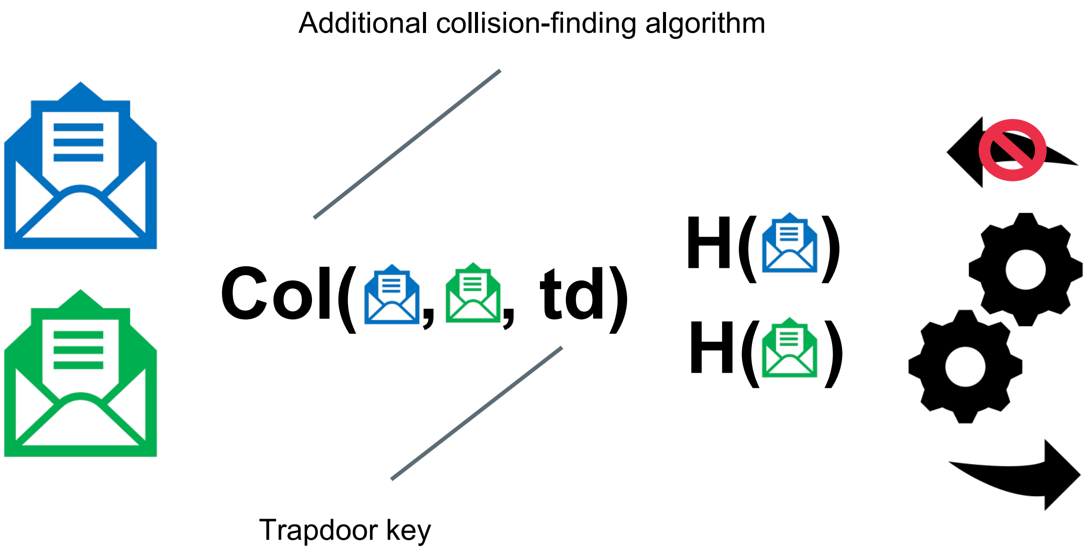
PRIMER: CRYPTOGRAPHIC HASH FUNCTIONS

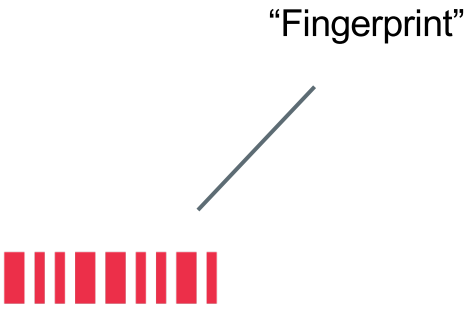


1. **One-way**
2. **Collision-resistant**
3. **Short output**

Hash function are a central ingredient to DLTs, e.g., RIPEMD-160 used in Bitcoin

CHAMELEON HASH (CH) FUNCTIONS





1. One-way
2. Collision-resistant
3. Short output

*… but only if **td** is unknown.*

CHAMELEON HASH (CH) FUNCTIONS

- Very useful cryptographic primitive envisioned by *Krawczyk and Rabin* (NDSS 2000), based on work by *Brassard, Chaum, Crépeau* (JCS 1988)

- Application in many research areas:
  - On-/offline digital signatures, tightly secure signatures, sanitizable signatures, identity-based encryption, direct anonymous attestation, distributed hashing, and in **editable blockchains**

- **Problem:** coarse-grained, if one is in possession of the trapdoor *td*, all security guarantees are lost

MAIN RESULT:

POLICY-BASED CHAMELEON HASHING

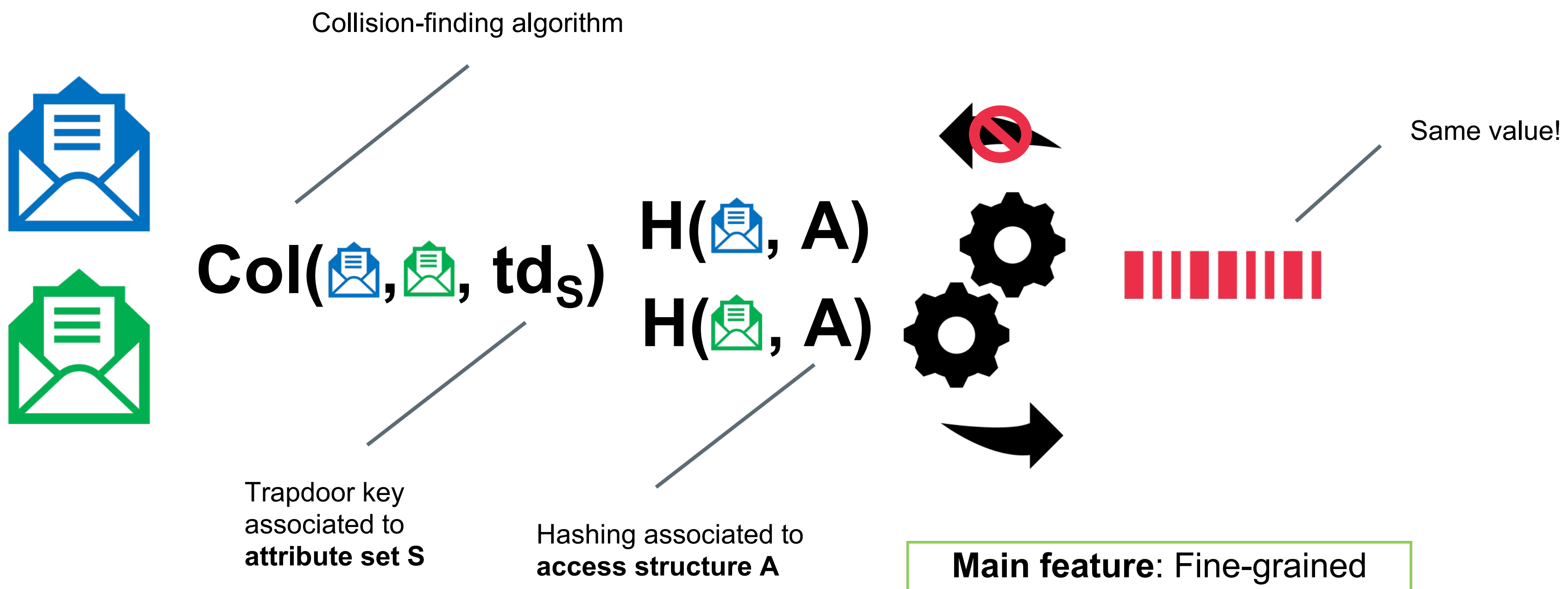
A new primitive for **fine-grained** hash-collision finding

POLICY-BASED CHAMELEON HASHING (PBCH)

- Enhances Chameleon Hashing with **attributes** and **access structure/policies**
- **Attributes** can be any string, e.g., “Scientist”, “Research”, “Engineer”
- **Access structures** can be seen as Boolean formulas, e.g., (“Research” AND “Scientist”) OR “Engineer”
- Attributes fulfill an access structure if the Boolean formula evaluates to 1/true

Mimics **fine-grained** collision finding
for **chameleon hashing** *and* **strong security guarantees**.

POLICY-BASED CHAMELEON HASHING (PBCH)

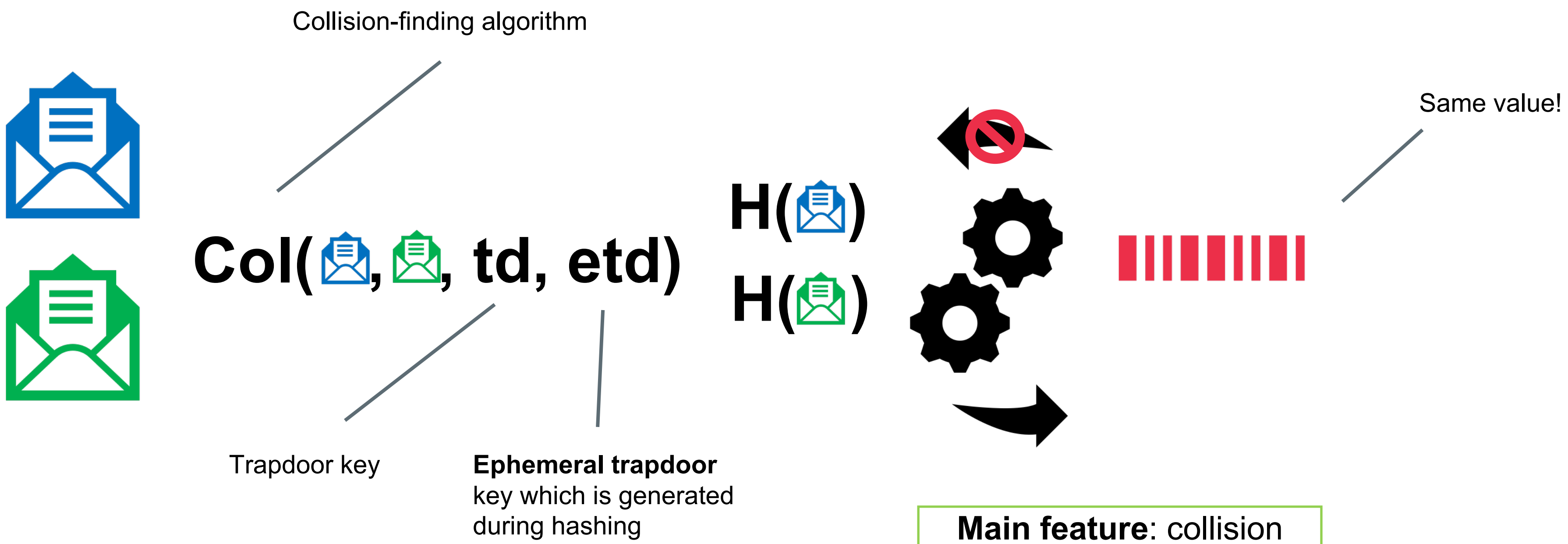


Main feature: Fine-grained collision finding if attribute set **S** fulfills the access structure **A**

INSTANTIATING PBCH

Combining Chameleon Hashing (with Ephemeral Trapdoors) and Attribute-Based Encryption

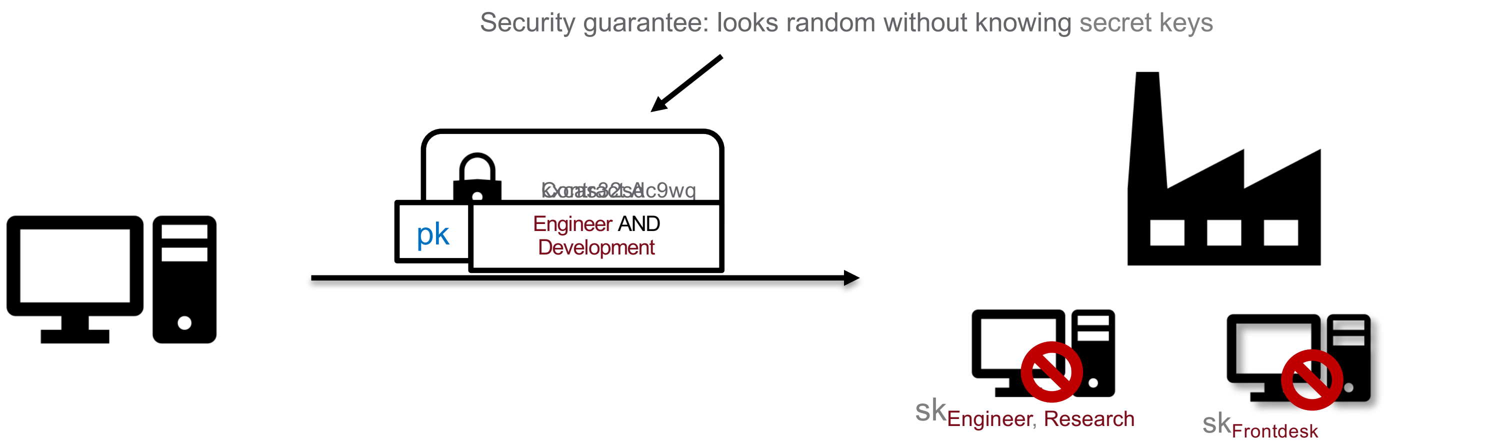
INGREDIENT 1: CHAMELEON HASHING WITH EPHEMERAL TRAPDOORS (CHET)

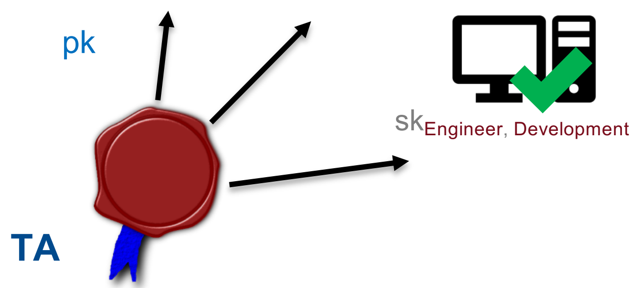


Main feature: collision finding possible if **td** and **etd** are present.

Due to Camenisch et al. (PKC 2017)

INGREDIENT 2: ATTRIBUTE-BASED ENCRYPTION (ABE)

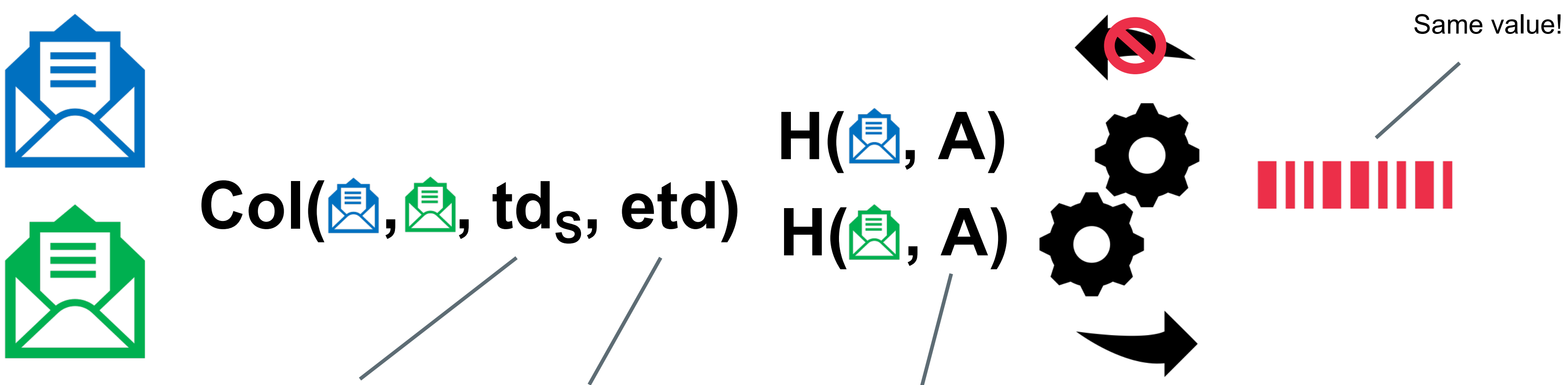




Properties:

- **Enables fine-grained one-to-many communication**
- Enforces access control on the cryptographic level
- Need of pk-related authority **TA** that distributes secret keys

PUTTING EVERYTHING TOGETHER



Trapdoor key associated to **ABE secret key for attribute set S**

Ephemeral trapdoor

Hashing also **encrypts etd for access structure A with ABE**

Main feature: collision finding possible if ABE secret key for **S** that fulfills access structure **A** for encrypted **etd** is known.

POLICY-BASED CHAMELEON HASHING (PBCH)

$\mathsf{Gen}(k)$: Outputs the secret key $sk_{\mathsf{PBCH}} \leftarrow (msk_{\mathsf{ABE}}, sk_{\mathsf{CHET}})$ and public key $pk_{\mathsf{PBCH}} \leftarrow (pk_{\mathsf{ABE}}, pk_{\mathsf{CHET}})$.

$\mathsf{Key}(sk_{\mathsf{PBCH}}, S)$: Outputs a secret key $sk_S \leftarrow (sk_{\mathsf{CHET}}, sk_{\mathsf{ABE},S})$.

$\mathsf{Hash}(pk_{\mathsf{PBCH}}, m, A)$: Outputs a hash $h \leftarrow (h_{\mathsf{CHET}}, C_A)$ and randomness $r \leftarrow r_{\mathsf{CHET}}$, for $(h_{\mathsf{CHET}}, r_{\mathsf{CHET}}, etd) \leftarrow \mathsf{Hash}_{\mathsf{CHET}}(pk_{\mathsf{CHET}}, m)$ and $C_A \leftarrow \mathsf{Enc}(pk_{\mathsf{ABE}}, A, etd)$.

$\mathsf{Verify}(pk_{\mathsf{PBCH}}, m, h, r)$: Return 1 if $\mathsf{Verify}_{\mathsf{CHET}}(pk_{\mathsf{CHET}}, h, h_{\mathsf{CHET}}, r_{\mathsf{CHET}})$, else 0.

$\mathsf{Col}(sk_S, m, m', h, r)$: Outputs randomness $r' \leftarrow \mathsf{Adapt}_{\mathsf{CHET}}(sk_{\mathsf{CHET}}, etd, m, m', h, r_{\mathsf{CHET}})$, for $etd \leftarrow \mathsf{Dec}_{\mathsf{ABE}}(sk_{\mathsf{ABE},S}, C_A)$.

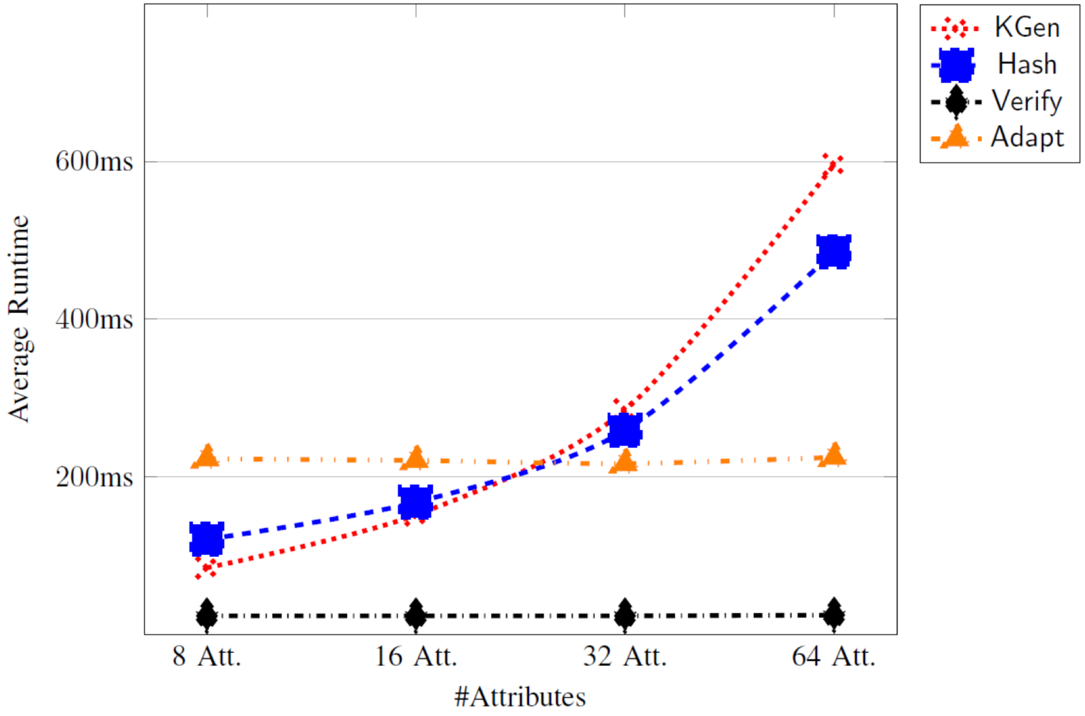
Ephemeral trapdoor *etd* can only be accessed with ABE **secret key for attributes** which fulfill **the ciphertext access structure**.

PBCH EVALUATION AND HIGH-LEVEL EXAMPLE

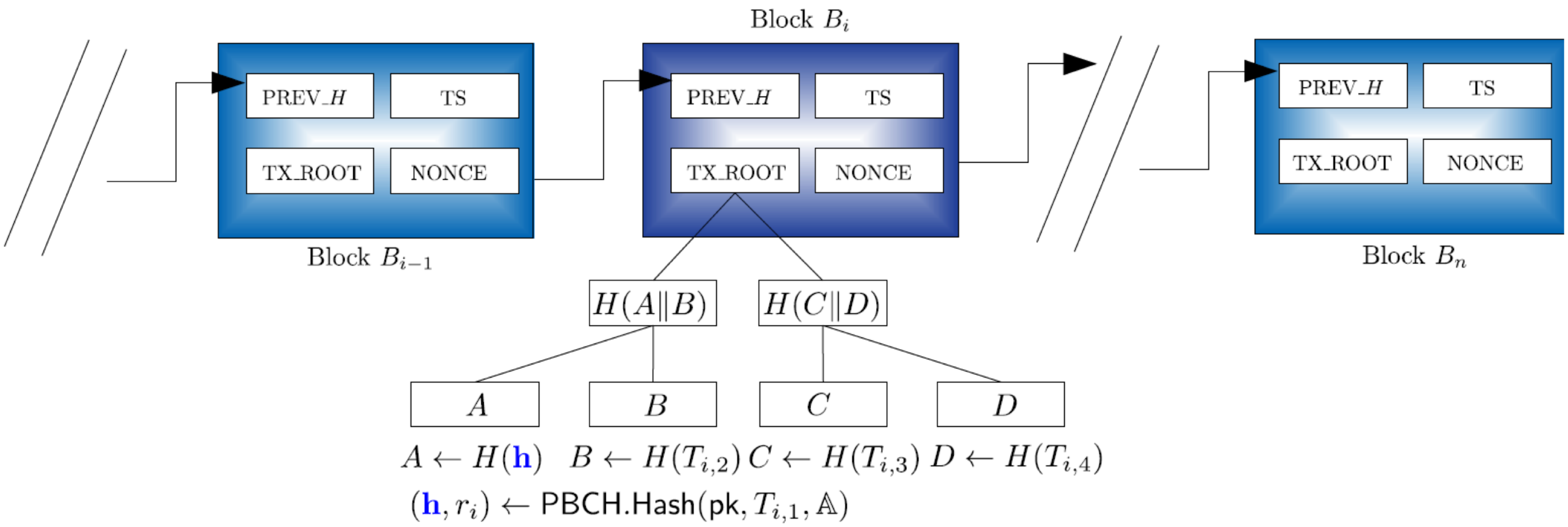
PBCH Proof-of-Concept Implementation

FIRST EVALUATION RESULTS

- Python 3.5.3 based on Charm framework v0.5
- Intel Core i7-7600U @ 2.8 GHz with 16 GB RAM
- ABE is instantiated with FAME (CCS 2017)
- Own CHET implementation
- Results under weaker security variant (at most doubling of Hash running time expected)



HIGH-LEVEL EXAMPLE



CONCLUSION

- **Editing/re-writing** DLs important aspect to consider
  - Possible on block level and transaction level

- New primitive **Policy-Based Chameleon Hashing (PBCH)** to allow fine-grained re-writing on the **transaction** level in DLs
  - Yields the **first instantiation** of its kind

- First performance **evaluations** and **high-level example** presented, details in the full version

- Open question: efficient integration into real-DL setting

THANK YOU!

Christoph.Striecks@ait.ac.at