KRAKEN

BROKERAGE AND MARKET PLATFORM FOR PERSONAL DATA

D4.2
Final research report on cryptographic protocols for privacy-preserving data markets and SSI systems

www.krakenh2020.eu

This project has received funding from the European Union’s Horizon 2020 (H2020) research and innovation programme under the Grant Agreement no 871473

D4.2
Final research report on cryptographic protocols for privacy-preserving data markets and SSI systems

Grant agreement	871473
Work Package Leader	TUG
Author(s)	Sebastian Ramacher (AIT)
Contributors	Daniel Slamanig (AIT), Karl Koch (TUG), Tilen Marc (XLAB)
Reviewer(s)	Juan Carlos Perez Braun (ATOS), Giancarlo Degani (INFOCERT)
Version	Final
Due Date	31/05/2022
Submission Date	27/05/2022
Dissemination Level	Public

Release History

Version	Date	Description	Released by
v0.0	22/02/2022	Initial version	Karl Koch (TUG)
v0.1	20/04/2022	Add T4.3 paper (Privacy-Preserving eID Derivation to Self-Sovereign Identity Systems with Offline Revocation)	Karl Koch (TUG)
v0.2	05/05/2022	Add sub-section about benchmarking of MPC programs & protocols as future work	Karl Koch (TUG)
V0.3	06/05/2022	Added subsection on efficient lattice-based inner-product functional encryption.	Tilen Marc (XLAB)
v0.4	10/05/2022	Add all the remaining publications from AIT, introduction and contribution	Sebastian Ramacher (AIT), Daniel Slamanig (AIT)
v0.5	12/05/2022	Update references and bibliography	Sebastian Ramacher (AIT)
V0.6	23/05/2022	Integrate comments from ATOS and INFOCERT review	Sebastian Ramacher (AIT)
v1.0	27/05/2022	Submitted version	ATOS

Table of Contents

List of Acronyms

Acronym	Description
(Q)ROM	(Quantum) Random Oracle Model
(t)SE	(True) Simulation extractability
(T)SPHF	(Trapdoor) Smooth-projective hash function
(u)L-TSPHF	(updatable) lighter trapdoor smooth projective hash function
0-RTT	Zero Round-Trip Time
ABC	Attribute-based credentials
AES	Advanced Encryption Standard
BFKEM	Bloom Filter key encapsulation mechanism
BLS	Boneh-Lynn-Shacham
CA	Consortium Agreement
CCA	Chosen Ciphertext Attack
CPA	Chosen Plaintext Attack
CRS	Common Reference String
DFPE	Dual-Form Puncturable Encryption
DID	Decentralized Identifier
ECDSA	Elliptic Curve Digital Signature Algorithm
FE	Functional Encryption
FO	Fujisaki-Okamoto
GDPR	General Data Protection Regulation
HHK	Hofheinz, Hövelmanns, and Kiltz
HVZK	Honest verifier zero-knowledge
IACR	International Association for Cryptologic Research
IBE	Identity-Based Encryption
IND-CCA2	Indistinguishability under adaptive chosen ciphertext attacks
KEM	Key-encapsulation mechanism
MPC	Multi-Party Computation
MQDSS	Multivariate Quadratic Digital Signature Scheme
NIST	National Institute of Standards and Technology
NIZK	Non-Interactive Zero Knowledge
NP	Nondeterministic polynomial complexity class
PE	Puncturable Encryption
PKE	Public-key encryption
PQC	Post-Quantum Cryptography
PRF	Pseudorandom Function
QA-NIZK	Quasi-adaptive non-interactive zero-knowledge
RLWE	Ring Learning With Errors
SIS	Short Integer Solution
SNARKs	Succinct Non-Interactive Arguments of Knowledge
SSH	Secure Shell Protocol
SSI	Self-Sovereign Identity
STARK	Succinct Transparent Argument of Knowledge

TLS	Transport Layer Security
UMAC	Universal Message Authentication Code
US	Updatable Signature
WP	Work Package
ZKP	Zero-Knowledge Proof
ZK-PoK	Zero-Knowledge Proof of Knowledge

List of Figures

Executive Summary

The work in Work Package 4 (WP4) is mainly concerned with the cryptographic tools employed as part of KRAKEN. Within WP4, the focus lies on the cryptographic design and analysis, as well as efficient and secure implementations of thereof. This deliverable, *Final research report on cryptographic protocols for privacy-preserving data markets and SSI systems*, describes the research efforts as conducted to build a privacy-preserving and authenticity-preserving KRAKEN architecture. The goal of this deliverable is to give a high-level overview of research results on cryptographic tools integrated and built for the KRAKEN architecture and on results on research questions motivated by the unique challenges of a privacy-preserving data marketplace. This deliverable extends the interim research report, *D4.1 Progress report on cryptographic protocols for privacy-preserving data markets and SSI systems,* with all research results that have been obtained as part of WP4.

The research results presented as part of D4.2 are motivated by the requirements and needs defined in other work packages. Specifically, work packages 2 (for the overall architecture), 3 (for the self-sovereign identity aspects) and 5 (for the data marketplace) and their deliverables D2.2 *Intermediate KRAKEN architecture,* D2.4 *KRAKEN intermediate technical design,* D3.1 *Self-sovereign identity solution. First release*, and D5.1 *Initial Pilot Marketplaces User Stories*, serve as main inputs for this deliverable.

This report relates to Tasks 4.1 to 4.3 of WP4: Task 4.1 provides design and research in the area of end-to-end-secure data sharing; between, e.g., a data producer and a data consumer via a data marketplace (like the one of KRAKEN). Task 4.2 tackles design and research questions in the area of privacy-preserving as well as authenticity-preserving data analytics, of, e.g., many data producers offering their data to a (dedicated) data consumer, such that the consumer gets only the analysis result, and all that while the (KRAKEN) marketplace neither learns the input data nor the result. Task 4.3 provides research and enhancements in the area of cryptographic aspects of self-sovereign identity (SSI). Finally, we outline ongoing research and future work within these areas.

1 Introduction

1.1 Purpose of the document

KRAKEN is comprised of three core pillars: the self-sovereign identity (SSI) paradigm, the data marketplace, and cryptographic tools. The goal of Work Package 4 (WP4) is to provide the cryptographic tools to support the core functionality of the data marketplace which are driven by the needs and requirements of the other two pillars. Therefore, cryptographic primitives, schemes, and protocols are analyzed, developed, and implemented efficiently and securely to support the applications envisioned for the data marketplace and the SSI systems. One central aspect of the KRAKEN use-cases is their focus on privacy; thus the cryptographic tools are developed with this feature in mind.

For the data marketplace, KRAKEN envisions use-cases that are supported and partly enabled by the cryptographic protocols and schemes. Secure multi-party computation (MPC), functional encryption (FE), and other cryptographic tools to perform computations on protected data enable us to build a data-analytics-as-a-service platform. With non-interactive zero-knowledge proofs, such systems can additionally ensure correctness of the performed computations. In combination with suitable digital signature schemes, we are able to lay the theoretical foundations to build an end-to-end confidential and authentic data flow in the marketplace.

Cryptography is also a key aspect in other pillars of the marketplace. As users should be enabled to have fine-grained control over their data, cryptographic schemes with fine-grained access control built-in are of interest. Sophisticated encryption schemes such as proxy re-encryption, puncturable encryption, attribute-based encryption, and others enable interesting use-cases for end-to-end secure provider-to-producer data sharing solutions. Furthermore, signature schemes, privacy-respecting variants thereof, and zero-knowledge proofs are the central building blocks of SSI systems and are thus of interest for KRAKEN.

The purpose of this document is to give an overview of the research results obtained regarding the objectives set out for WP4 and specifically on those covered by Tasks 4.1 to 4.3. The objectives focus on cryptographic tools for end-to-end secure data sharing, authenticity of data analytics, confidentiality of privacy-sensitive data in outsourced computations, and the building blocks of SSI systems. The research topics are also driven by the requirements identified and derived in Work Package 5 (cf. D5.1 *Initial Pilot Marketplaces User Stories [25]*, D5.4 *Final KRAKEN marketplace integrated architecture document*) and Work Package 3 (cf. D3.1 *Self sovereign identity solution. First release* [26]) and the architecture designed in Work Package 2 (cf. D2.2 *Intermediate KRAKEN architecture* [23], D2.4 *KRAKEN intermediate technical design* [24], D2.5 *KRAKEN final technical design*). As this deliverable discusses cryptographic building blocks that will be integrated in KRAKEN, it also serves as input for implementation efforts in Task 4.4 and is therefore input to its final deliverable, D4.4 *Final implementation of cryptographic libraries*.

1.2 Task Overview

We will give a short overview of the goals of the three tasks T4.1 – T4.3.

T4.1 End-to-end secure data-sharing capabilities. The first task is concerned with the technical solutions to securely share data between two parties. The goal is to ensure confidential data exchange in an end-to-end secure manner. Hence, no party other than the intended receiver should be able to gain access to the data. In the context of a data marketplace architecture (cf. D2.4 [24]), end-to-end security poses interesting challenges. In a data marketplace, the data from a provider passes potentially many different services until it reaches the data producer, i.e., its intended receiver. As the data may be stored – even if only temporarily – on different services operated within and outside the data marketplace, achieving end-to-end security is paramount. Otherwise, a data marketplace would run into legal issues when any of the services is compromised or dishonest and confidential data is leaked.

Feature-wise, KRAKEN wants to enable all data producers and data providers to exchange the data asynchronously. Meaning that a data providers may not know the producers when they upload their data. Therefore, public keys that could otherwise be used to encrypt data specifically for the provider cannot assumed to be available at this point. Hence, the data marketplace needs to provide suitable cryptographic means to support this feature. In this context, encryption schemes that allow fine-grained access control become of interest.

Beside end-to-end security and fine-grained access control, other cryptographic properties become of interest in the setting of data marketplace. As ciphertexts are potentially stored over longer periods of time, forward-secrecy and everlasting security are interesting security properties. For the former, data is protected even in the event of key leakage. For the latter, one is mainly concerned with the risks posed by quantum computers and efficient attacks on number-theoretical problems that become practical with powerful-enough quantum computers.

Task 4.2 Authenticity-preserving and privacy-preserving data analytics. The second task is focused on the core pillars of a data marketplace that offers data-analytics-as-a-service type functionality (c.f. Figure 1). In this setting, the goal is to provide a service which prevents data producers access to the plain data but provides them with the possibility to buy a statistical analysis of the data of potentially many providers. In this scenario, there are many different security goals to be considered: first of all, beyond the information that is revealed from result of the analysis, the data producer should not learn anything else on the data provider's potentially sensitive data. Conversely, except the producer, none of the providers nor the services, i.e., the data processors, should learn the result of the computation. Besides these confidentiality guarantees, authenticity also plays a significant role. Furthermore, the data coming from the providers needs to be authentic, and in addition, the data producer needs to be ensured that the computation has been performed correctly. Consequently, the authenticity and integrity of all the data that is processed by the system can be verified. Thereby, potentially misbehaving data processors can be identified as they would break those guarantees.

As candidates of cryptographic schemes and protocols to achieve the desired functionalities, secure multi-party computation and zero-knowledge proofs among others have been identified as promising technologies. Secure multi-party computation achieves the desired confidentiality guarantees, whereas zero-knowledges proofs and, in particular, Succinct Non-Interactive Arguments of Knowledge (SNARKs) enable us to ensure integrity of the processed data. We want to note, though, that combining these two technologies is a non-obvious task. Hence, we investigate their combination for specific KRAKEN use-cases.

We also want to note that the trust assumptions required to deploy MPC or SNARKs in such a system are of interest. Especially for SNARKs, the setup of the common reference string requires a trusted third party. Therefore, reducing these trust assumptions while still achieving the central properties of a SNARK – zero-knowledge and knowledge soundness – renders the deployment of SNARKs more practical.

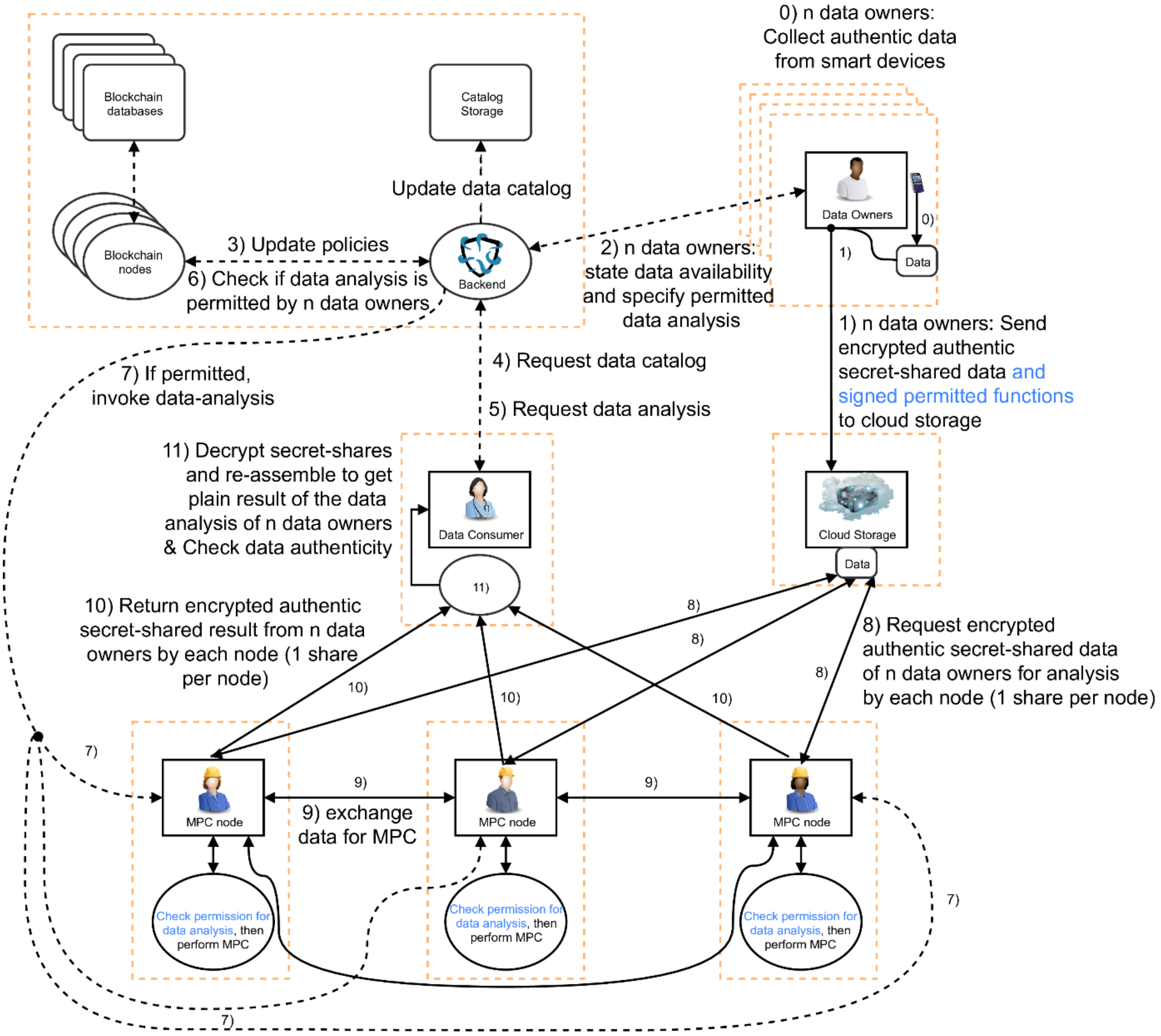
0) n data owners: Collect authentic data from smart devices
Blockchain databases
Catalog Storage
Update data catalog
Data Owners
0)
Data
1)
Blockchain nodes
3) Update policies
6) Check if data analysis is permitted by n data owners
Backend
2) n data owners: state data availability and specify permitted data analysis
1) n data owners: Send encrypted authentic secret-shared data and signed permitted functions to cloud storage
4) Request data catalog
7) If permitted, invoke data-analysis
5) Request data analysis
11) Decrypt secret-shares and re-assemble to get plain result of the data analysis of n data owners & Check data authenticity
Data Consumer
Cloud Storage
Data
11)
10) Return encrypted authentic secret-shared result from n data owners by each node (1 share per node)
8)
8)
8)
8) Request encrypted authentic secret-shared data of n data owners for analysis by each node (1 share per node)
10)
10)
10)
7)
9)
9)
MPC node
MPC node
MPC node
9) exchange data for MPC
Check permission for data analysis, then perform MPC
Check permission for data analysis, then perform MPC
Check permission for data analysis, then perform MPC
7)
7)

Figure 1: Architecture for privacy-preserving analytics in a data marketplace.

Task 4.3 Cryptographic aspects of SSI systems. Self-Sovereign Identity (SSI) systems aim to give the user control over their (digital) identity. Such identity systems are built from digital signatures schemes. Since SSI systems are built on top of personal data, privacy-preserving features are of high priority. Therefore, SSI systems are extended with zero-knowledge proofs to provide the users with mechanisms to disclose some of their attributes in a privacy-preserving manner. As such, methods that support, improve, and make such approaches more practical are of interest for this task. These research questions relating to zero-knowledge proofs and SNARKs in particular, are also relevant for SSI systems.

Furthermore, integrating zero-knowledge proofs in identity systems with already pre-defined digital signature schemes is not straightforward. Many of the standardized and currently employed signature schemes are not suitable for proofs of knowledges of signed messages or similar types of statements. Therefore, it is either required to select suitable signature schemes that support composition with SNARKs or the use of specifically designed cryptographic protocols for such use-cases. Attribute-based credential systems, for example, provide such a functionality while also enjoying additional, privacy-friendly features such as unlinkability.

1.3 Overview of Our Contributions

The contributions in WP4 can be clustered as follows:

Privacy-preserving data marketplaces. Finally, in this research cluster we studied the design and architecture of decentralized data marketplaces built on top of MPC. In such marketplaces unique challenges arise from the decentralized nature of the computation. Additionally, due to the MPC assumptions, the computing nodes need to be provided by distinct entities to uphold the security guarantees. Therefore, these aspects need to be considered in an architecture.

For the data-analytics-as-a-service component, cryptographic methods to compute on encrypted data (or otherwise protected data) are of central interest to KRAKEN. We investigated schemes to support this use-case in different directions. One of our contributions focuses on practical improvements to functional encryption. Further work is focused on selecting and designing suitable symmetric primitives for the use in MPC computations. Additionally, two additional works in progress investigate the combination of MPC with SNARKs to ensure data authenticity throughout the full data flow as well as additional features for functional encryption schemes for marketplaces.

We have published three academic papers in this research cluster (cf. Sections 2.3, 3.6, and 3.10) and three contributions are currently work in progress (cf. Sections 4.1, 4.2, and 4.4)

SSI systems. In this cluster we investigate various aspects relevant for SSI systems. We identified several interesting questions. First, we are interested in the use of credential systems in a multi-issuer scenario. In that case, the identity of the issuer might already leak sensitive information. Second, we design a privacy-preserving SSI system that supports the derivation of SSI credentials from eID. We extend this system with revocation features that are also designed to be privacy-preserving. Related to revocation, we also investigated how efficient and compact revocation schemes can be implemented using MPC-techniques in a distributed system common in SSI systems such as Sovrin [19].

Our contributions in this line of research consist of three academic papers (cf. Sections 2.4, 2.5, and 3.9) that are supported with prototypical implementations to gauge their efficiency.

Updatable and subversion-resistant zero-knowledge proofs. In this line of research, we are interested in the properties of zero-knowledge proof systems related to subversion-resistance, i.e., the soundness and zero-knowledge properties hold even if the common reference string is subverted, and updatability, i.e., that the common reference string can be updated so that all trapdoors held by some participants are invalidated. Consequently, the trust assumptions for the generation of the common reference string can be reduced. In a highly decentralized setting such as a marketplace or SSI system removes an otherwise central and trusted authority for the setup phase from the architecture.

So far, we have published three academic papers in this line of research (cf. Sections 2.1, 2.2, and 3.7) and one contribution is work in progress (cf. Section 4.3).

End-to-end data sharing. In the context of data-sharing, we investigate properties of (public-key) encryption schemes that are of particular interest to the data sharing work flows in the marketplace. The first property is related to fine-grained access control. As users should be enabled to upload data to to a suitable cloud-storage location without a priori defining the possible set of receivers, we require more advanced primitives that support more flexible delegation mechanisms of the encryption rights. As second feature, we are interested in forward secrecy. Especially when ciphertexts are stored over longer periods of time, key leakage becomes an issue. With a forward-secret encryption scheme, some of the risks of key leakage can be mitigated as in such systems keys lose the ability to decrypt ciphertexts from a certain time epoch.

Our contributions in this area consist of two academic papers (cf. Sections 3.1, and 3.4).

Digital signatures. Digital signatures are one of the core components of secure communication but also for SSI systems. For the former, the integration of signature schemes secure against quantum adversaries is becoming more and more important. Hence, we have investigated so-called post-quantum secure digital signature schemes built from assumptions and techniques that are not susceptible to efficient attacks by classical or quantum computers. Our contributions in this area consist of two academic publications (cf. Sections 3.2, 3.3).

For the latter, we are interested also in efficient methods to delegate certain signing rights. Thereby, we can enable users of our system to temporarily delegate access rights and other functionality that is bound to successful authentication to other users. Furthermore, we investigate methods to support key rotation in signature schemes without the need to reissue signatures under new keys. Such signature schemes can improve the handling of long-term keys that need to re-issue out of caution or due to key compromise. Our contributions in this area have been published as two academic paper (cf. Section 3.8, and 3.5) and are partially supported by prototype implementations.

1.4 Structure of the document

The remainder of this document is organized as follows:

- Section 2 describes our peer-reviewed and published contributions obtained since D4.1 [1],
- Section 3 describes our peer-reviewed and published contributions from D4.1 [1],
- Section 4 describes ongoing research and results that are currently under submission.

2 New Results since D4.1

In this section we present all the research results that were obtained since the finalization of D4.1. For each result, we present an extended abstract in the following subsections. The full versions of the papers are available as open access on the International Association for Cryptologic Research's (IACR) ePrint service. If no open access version is available yes, such versions will be provided until the end of KRAKEN and will be reported in the final dissemination report of the project, deliverable *D6.8 Final communication, dissemination and standardization report*.

2.1 Updatable Trapdoor SPHFs: Modular Construction of Updatable Zero-Knowledge Arguments and More

Zero-knowledge (ZK) proofs were introduced by Goldwasser, Micali and Rackoff [43] and play a central role in both the theory and practice of cryptography. A long line of research has led to efficient pairing-based zero-knowledge Succinct Non-interactive ARguments of Knowledge (zk-SNARKs) and succinct Quasi-Adaptive Non-Interactive Zero-Knowledge arguments (QA-NIZKs) in the common reference string (CRS) model. QA-NIZKs [45] are a relaxation of NIZK proofs where the CRS is allowed to depend on the specific language for which proofs have to be generated. In general, SNARKs (QA-NIZKs) are succinct, in fact, they allow to prove that circuits of arbitrary size (for linear languages) are satisfied with a constant-size proof. They are also concretely very efficient, 3 group elements is the best SNARK construction for arithmetic circuits [44] and 1 group element is the best QA-NIZK construction for linear languages [46]. Recently, Campanelli et al. [39] proposed LegoSNARK, a toolbox for commit-and-prove zk-SNARKs (CP-SNARKs), where they use succinct QA-NIZKs as efficient zk-SNARKs for linear subspace languages.

For the practical application of zero-knowledge primitives, an important question is the generation of the CRS. While in theory it is simply assumed that some trusted party will perform the CRS generation, such a party is hard to find in the real-world. Recently, there has been an increasing interest to reduce the trust in the generator of the CRS. Existing approaches are (1) the use of multi-party computation to generate the CRS in a distributed way, e.g., [38], or (2) the use of CRS checking algorithms in subversion NIZKs [8], zk-SNARKS [33] [41] and QA-NIZKs [34]. Here, although the prover does not need to trust the CRS, the zero-knowledge property (so called subversion ZK) is still preserved. However, the verifier still needs to trust the CRS generator. Abdolmaleki et al. [34] later studied the Kiltz-Wee QA-NIZKs [46] in a variant of the bare public-key model, where some part of the CRS (the language parameter) is generated by a trusted party, but the rest of the CRS can be generated maliciously by some untrusted party (from the prover's perspective). Finally, (3) there is the recent approach of a so-called updatable CRS [12]. Here, everyone can update a CRS such that the updates can be verified and ZK holds in front of a malicious CRS generator and the verifier can trust the CRS (soundness holds) as long as one operation, either the CRS creation or one of its updates, have been performed honestly. So, a verifier can do a CRS update on its own and then send the updated CRS to the prover. Note that this updating inherently requires communication of the prover and the verifier in such an updatable SNARK/QA-NIZK setting, a fact that will be useful for our work.

Our starting point Smooth Projective Hash Functions (SPHFs) [40], which can be viewed as honest-verifier zero-knowledge (HVZK) arguments for the membership in specific languages. HVZK is a weakened variant of ZK, which only needs to hold for honest verifiers. Benhamouda et al. [35] [37] showed how one can construct ZK instead of HVZK arguments in the CRS model by introducing so called Trapdoor SPHFs (TSPHFs). Recently, Abdolmaleki et al. [32] defined a variant of TSPHF with an untrusted setup, so called smooth zero-knowledge hash functions, and show how to construct 2-round

ZK arguments in the plain (or sub-version ZK arguments in the CRS) model under a non-falsifiable assumption

In this work we revisit the notion of TSPHFs proposed by Benhamouda et al. [35] [37]. We present a new approach which we call lighter TSPHFs (L-TSPHFs), allowing instantiations in bilinear groups that are more efficient than known TSPHFs, as all three hashing algorithms hash, projhash, and thash yield hash values in source groups of the bilinear group. We present a framework for updatable L-TSPHFs (uL-TSPHFs) which is inspired by updatable zk-SNARKs. Our updatable L-TSPHF framework is a generic building block which can be used to modularly design updatable primitives. Our instantiation of an uL-TSPHF is directly based on an L-TSPHF together with a suitable additive updating procedure of the trapdoor in the CRS and extraction based on the BDH-knowledge assumption [33]. We then show as the main application of uL-TSPHFs the construction of updatable QA-ZK arguments. When compared with the only existing construction of updatable QA-ZK proofs in [47] (which is ad-hoc), we can significantly reduce the proof as well as the communication size and in particular obtain succinct proofs. Moreover, we present a concrete instance for proving the correct encryption of a valid Waters signature. Finally, as another interesting application, we show how to construct an updatable two-round Password-Authenticated Key-Exchange protocol from uL-TSPHFs, which allows to reduce trust in the setup of the PAKE.

Publication details: **Updatable Trapdoor SPHFs: Modular Construction of Updatable Zero-Knowledge Arguments and More.** *Behzad Abdolmaleki (external), Daniel Slamanig (AIT).* In Information Security and Privacy. ACISP 2021. Lecture Notes in Computer Science, vol 13083. Springer, Cham. https://doi.org/10.1007/978-3-030-90567-5_3. Open access: https://eprint.iacr.org/2021/1360

2.2 Subversion-Resistant Quasi-adaptive NIZK and Applications to Modular Zk-SNARKs

Zero-knowledge (ZK) proofs introduced by Goldwasser, Micali and Rackoff [43] are cryptographic protocols between two parties called the prover and the verifier with the purpose that the prover can convince the verifier of the validity of a statement in any language in NP without revealing additional information. Besides this zero-knowledge property, such a system needs to provide soundness, i.e., it must be infeasible for the prover to provide proofs for false statements. While ZK proofs, in general, may require many rounds of interaction, an interesting variant is non-interactive zero-knowledge (NIZK) proofs. They require only a single round, i.e., the prover outputs a proof, and this proof can then be verified by anybody. A long line of research [52] [53] [54] [56] [57] [49] [44] has led to efficient pairing-based succinct NIZKs called zero-knowledge Succinct Non-interactive ARguments of Knowledge (zk-SNARKs), which are NIZK arguments with i) a stronger notion of soundness called knowledge soundness and, more importantly, ii) in which proofs, as well as the computation of the verifier, are succinct, i.e., ideally a small constant amount of space and computation respectively. Due to these latter properties, zk-SNARKs are a suitable tool to preserve privacy within cryptocurrencies and distributed ledger technologies, most notably used within Zcash [36], and they increasingly attract interest outside of academia. Our focus is on quasi-adaptive NIZK (QA-NIZK) arguments [45], i.e., NIZKs in which the common reference string (CRS) depends on a language parameter and they have many applications and have been intensively studied.

For practical applications of (QA-)NIZKs and zk-SNARKs, an important question is the generation of the CRS. While in theory it is simply assumed that some mutually trusted party will perform the CRS generation, in many real-world settings (such as fully decentralized systems) there typically does not exist such a trusted party. Recently, there has been an increasing interest to reduce trust in the generator of the CRS. One of these lines of work is subversion zero-knowledge initiated by Bellare et al. in [8], where the zero-knowledge property even holds when the CRS is generated maliciously, i.e., the CRS generator is subverted. Following this initial work, Abdolmaleki et al. [33] [48] as well as Fuchsbauer [41] investigated subversion zk-SNARKs. More recently, Abdolmaleki et al. (ALSZ) in [34] initiated the study of subversion zero-knowledge QA-NIZK (Sub-ZK QA-NIZK for short). While the latter is an important step, it leaves a number of open problems such as weakening the requires assumptions, stronger soundness guarantees and demonstrating impact for real-world applications.

In this work we first investigate the most efficient QA-NIZK constructions of Kiltz and Wee (KW) [46] and the asymmetric QA-NIZKs by González et al. (GHR) [50] in a subverted setup. In contrast to ALSZ, which relies on a new non-standard knowledge assumption for their subversion zero-knowledge property, our Sub-ZK QA-NIZK can be shown to have this property under the more well-known Bilinear Diffie-Hellman Knowledge of Exponents (BDH-KE) assumption [33][48]. Moreover, we present a Sub-ZK QA-NIZK version of GHR by relying on the same BDH-KE assumption.

Then, we investigate the construction of Sub-ZK QA-NIZK that satisfies a stronger notion of knowledge soundness and in particular a weakened version of simulation extractability (SE) called true-simulation extractability (tSE) [55]. Our work is the first treatment of tSE Sub-ZK QA-NIZK and we present unbounded tSE Sub-ZK QA-NIZKs based on KW (also in the non-subversion setting).

Finally, we consider applications of Sub-ZK QA-NIZK and in particular their integration into modular zk-SNARK frameworks. One such popular framework is LegoSNARK [39], a framework for Commit-and-Prove zk-SNARKs (CP-SNARKs) with the aim of constructing a "global" SNARK for some computation C via the linking of "smaller" specialized SNARKs for different subroutines that overall compose to C. The main idea is that by letting each subroutine of C be handled by a different proof system one can choose the one that maximizes a metric (e.g., efficiency) that is important for the concrete application. We will show how to integrate subversion primitives into LegoSNARK. In particular, we show how to integrate our Sub-ZK QA-NIZKs instead of their non-subversion counterparts. Together with the results on subversion (SE) zk-SNARKs [33] [41] [51] [42] [58], we thus make an important step towards a complete subversion (SE) variant of the LegoSNARK framework.

Publication details: **Subversion-Resistant Quasi-adaptive NIZK and Applications to Modular Zk-SNARKs.** *Behzad Abdolmaleki (external), Daniel Slamanig (AIT).* In Cryptology and Netowrk Security. CANS 2021. Lecture Notes in Computer Science, vol 13099. Springer, Cham. https://doi.org/10.1007/978-3-030-92548-2_26. Open access: https://eprint.iacr.org/2020/364

2.3 Efficient Lattice-Based Inner-Product Functional Encryption

Function encryption (FE) is a generalization of traditional public-key encryption overcoming the all-or-nothing property of it. FE allow an authorized user holding a functional key to evaluate a computation on the encrypted message m and obtain the result of the computation f(m) without revealing any additional information about the message m, beyond the result. The functionality provided by this primitive can be useful in practical scenarios such as cloud computing and computation over encrypted data without interactions. While many general-purpose FE schemes exist, they suffer from a severe inefficiency and cannot be used in practical scenarios.

On the other hand, a research area emerged with the goal of designing FE with limited but still wide classes of functionalities that are efficient enough to be implemented and used in practice. In particular, FE schemes for inner product (linear functions) have been developed to be as efficient as possible. Nevertheless, an open problem on this basic inner-product functionality remained: can it be efficiently instantiated based on a quantumly secure assumption. While it was known that it can been instantiated on the Learning with Errors (LWE) assumption, this construction is arguably impractical for real-world scenarios. The Ring Learning with Errors (RLWE) assumption provides quantum-resistance security while in comparison with LWE assumption gives significant performance and compactness gains. In this work we present the first RLWE-based inner-product scheme.

To obtain a truly efficient FE schemes we need to carefully choose strategies in the security proofs to optimize the size of the parameters. More precisely, we develop two new results on ideal lattices. The first result is a variant of the RLWE, that we call multi-hint extended RLWE, where some hints on the secret and the noise are given. We present a reduction from RLWE problem to this variant. The second tool is a special form of Leftover Hash Lemma (LHL) over rings, known as Ring-LHL.

To argue on the practicality of the scheme and to demonstrate the efficiency we provide an optimized implementation of RLWE-based IPFE scheme. We show how the scheme can be used in a practical classification task with a machine learning model on encrypted data. We explain how our scheme can be used to securely evaluate functions on multiple data in parallel. For example, in the case of classifying images (785-dimensional vectors), one can encrypt up to 4092 images in under 0.4s and classify them all in under 0.2s, using liner regression.

Furthermore, we extend our scheme to scenarios where multiple clients are encrypting their data in a centralized or decentralized approach. We present new compilers that, combined with some existing ones, can transfer a single-input FE to its (identity-based, decentralized) multi-client variant with linear size of the ciphertext (w.r.t the number of clients).

Publication details: **Efficient lattice-based inner-product functional encryption.** *Jose Maria Bermudo Mera (external), Angshuman Karmakar (external), Tilen Marc (XLAB), Azam Soleimanian (external).* In Proceedings of the 2022 IACR International Conference on Public-Key Cryptography (PKC '22). Springer-Verlag, Berlin, Heidelberg. Open access: https://eprint.iacr.org/2021/046.

2.4 Issuer-Hiding Attribute-Based Credentials

Anonymous credential systems and their attribute-based extensions (ABCs) allow users to receive digital certificates (credentials) certifying certain pieces of personal information (attributes) from issuers. A user can then present her credential to a verifier in a way that respects the user's privacy while giving high authenticity guarantees to the verifier. That is, the user can decide, on a fine-granular basis, which information about their attributes they want to disclose to the verifier, while no further information, including metadata, is revealed. In particular, different actions of the same user can only be linked through the disclosed information. In the most general case, the verifier can publish arbitrary predicates (Boolean formulas) over attribute values that users need to satisfy for authentication (e.g., a user is older than 21, comes from a specific country, or has a certain name), and receives cryptographic evidence that such attribute values were certified by the given issuer.

ABC systems have in common that the privacy guarantees only hold with respect to a single issuer key: whilst not being able to link actions of a single user, a verifier learns the public key of the issuer of the underlying credential. Even though this seems to be a natural property at first glance, it turns out that this approach leads to a tradeoff between scalability and user privacy. As an example, consider a state-wide electronic identity system with millions of users. In order to give users the highest level of privacy, all citizens' personal credentials need to be issued under the same public key. In case of a compromise of the secret key, all previously issued keys need to be invalidated, potentially requiring millions of certificates to be re-issued under a new key. Alternatively, different keys could be used for groups of citizens, e.g., randomly, per time period, or per administrative region. However, as the issuer's public key is revealed upon presentation, this approach dramatically reduces the size of the anonymity set of a specific user.

In this work, we tackle this problem by introducing the notion of issuer-hiding attribute-based credential systems. In such a system, the verifier can define a set of acceptable issuers in an ad-hoc manner, and the user can then prove that her credential was issued by one of the accepted issuers -- without revealing which one. We then provide a generic construction, as well as a concrete instantiation based on a structure preserving signature scheme and simulation-sound extractable NIZK, for which we also provide concrete benchmarks in order to prove its practicability. The online complexity of all constructions is independent of the number of acceptable verifiers, which makes it also suitable for highly federated scenarios.

Publication details: **Issuer-Hiding Attribute-Based Credentials.** *Jan Bobolz (external), Fabian Eidens (external), Stephan Krenn (AIT), Sebastian Ramacher (AIT), Kai Samelin (external)*. In: Cryptology and Network Security - 20th International Conference, CANS 2021, Vienna, Austria, December 13-15, 2021, Proceedings. Open access: https://eprint.iacr.org/2022/213.

2.5 Privacy-Preserving eID Derivation to Self-Sovereign Identity Systems with Offline Revocation

"Digital identities play a vital role in an increasingly digital world. These identities often rely on central authorities to issue and manage them. Central authorities have the drawback of being a central trusted party, representing a bottleneck and single point of failure with exclusive control of identity-related data." - [29]. This central authority could mean in the KRAKEN context, for instance, that the national government of the respective users is the central identity provider; or, as with many web services, that the KRAKEN marketplace itself issues user credentials.

"Self-sovereign identity (SSI) tackles those problems by utilizing distributed ledger technology and making users the sovereign owners of their identity data. Nevertheless, SSI, as recent technology, still lacks qualified identity data. This is especially a problem since sensitive services like eGovernment or banking services require identity data issued by a qualified identity provider; thus, SSI-based identities cannot be used for these services." - [29]. Also in KRAKEN, we leverage SSI to have more user-centric means of authentication. However, if the KRAKEN marketplace needs qualified identity data, like an identity proof from the user's respective government, the KRAKEN marketplace - or the SSI-system's nodes - usually sees the full identity assertion. Hence, these entities see more than actually needed; for instance, it might be enough to see the user's name and birth date, and not also, e.g., the user's driver license(s) or national person identifier. In such a case, the birth date could also be hidden, and only proving, e.g., that the user's age is at least 18. Moreover, a user might want to revoke the SSI credentials; e.g., when the name changes or one of the devices got lost. In such a revocation case, usually the SSI-system nodes get to know more about the attributes than needed; hence, there is a lack of doing it in a privacy-preserving way.

"In this paper, we propose a concept for deriving identity data from an existing identity system into an SSI in a fully privacy-preserving way by additionally supporting offline verification. This way, we enable a chain of trust from the existing identity system to the SSI system by introducing a novel trust model. Our concept utilizes novel cryptographic primitives to support efficient and privacy-preserving identity showing as well as revocation. To underline the feasibility of our concept, we implement a proof system and benchmark the related use cases." - [29]. Figure 2 shows the overall concept of our solution. The concept consists of four stages: (1) Import Identity Data, (2) Obtain Attestation, (3) Revocation of SSI Credentials, and (4) Showing of SSI Credentials to Verifier.

For importing the identity data into the SSI world, the user (1) creates a zero-knowledge proof (ZKP) for the identity-assertion's signature (hash's pre-image), and (2) gets an attestation from the SSI-system's nodes. If a user decides to revoke the SSI credentials, a corresponding request is sent to the

SSI-system's nodes. Finally, for, e.g., authenticating towards a verifier, the user creates another individual ZKP for the desired attributes to show using the received attestation, and another ZKP for proving that the credentials have not been revoked. To verify the proofs, the verifier can either use offline-stored keys, or request those keys online. Furthermore, in addition to an offline check of the revocation status, a verifier could also ask the (online) SSI-system nodes.

We implemented and benchmarked this concept based on Bellman [31] in Rust: [30]. The benchmarks resulted in the following times for proving and verification:

- Attestation
  - Proof creation: ~15 seconds
  - Verification: ~20 ms
- Showing
  - Proof creation using the Hash functions (1) SHA-256 and (2) Poseidon [16] for various elements in the revocation-accumulator's Merkle tree
    - SHA-256:
      - ~2 seconds for 32 elements
      - ~8 seconds for 2048 elements
    - Poseidon:
      - ~350 ms for 32 elements
      - ~470ms for 2048 elements
  - Verification: ~3.7 ms

The performance bottlenecks are the proof creations. For the proof creation during the attestation phase, we need to adhere to the supported hash function by the *"traditional"* Identity Provider (IdP); otherwise, we would not be compatible anymore, assuming these IdPs do not often change. For the proof creation during the showing phase, however, we can choose a suitable hash function. The *"traditional"* Hash function SHA-256 is rather slow when the depth of the Merkle tree grows; hence SHA-256 is not practical for our showing phase. Poseidon, on the other hand, is a *modern* hash function designed with requirements of ZKPs in mind. As such, Poseidon's proof-creation time only grows slightly linearly, ~20ms per added depth of the Merkle tree; thus, Poseidon is practical for our showing phase. Figure 3 shows the difference between SHA-256 and Poseidon for the proof creation of our showing phase, for several levels of the Merkle-trees depth. *"In all cases, the proofs are the size of three group elements. With our choice of curve, the proof takes 192 bytes if the elements are stored in compressed form."* - [29]

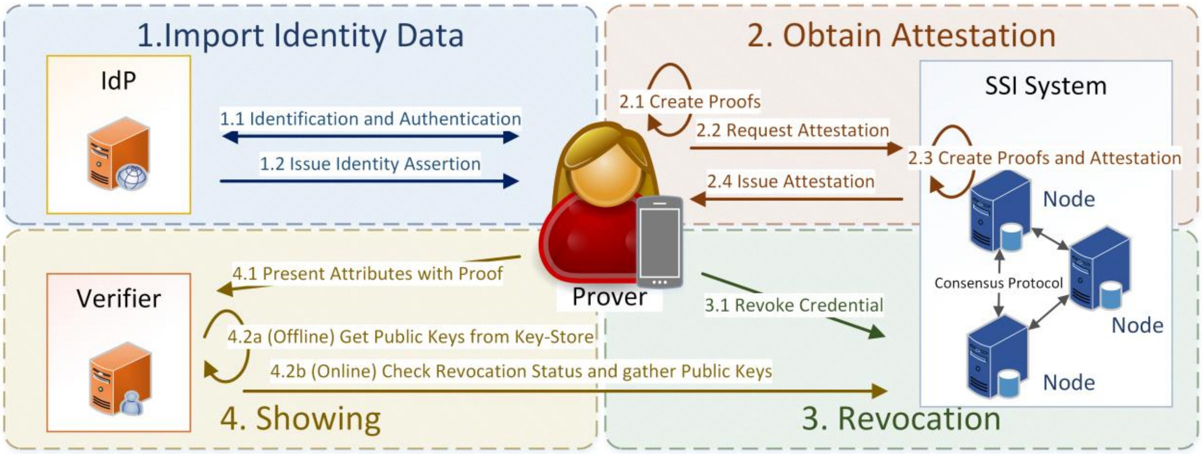
1.Import Identity Data
IdP
1.1 Identification and Authentication
1.2 Issue Identity Assertion
2. Obtain Attestation
2.1 Create Proofs
2.2 Request Attestation
2.3 Create Proofs and Attestation
2.4 Issue Attestation
SSI System
Node
Consensus Protocol
Node
Node
Prover
Verifier
4.1 Present Attributes with Proof
4.2a (Offline) Get Public Keys from Key-Store
4.2b (Online) Check Revocation Status and gather Public Keys
3.1 Revoke Credential
4. Showing
3. Revocation

Figure 2: Architectural overview of the proposed concept including the actors and main process flows within the four defined stages: (1) Import Identity Data, (2) Obtain Attestation, (3) Revocation, and (4) Showing.

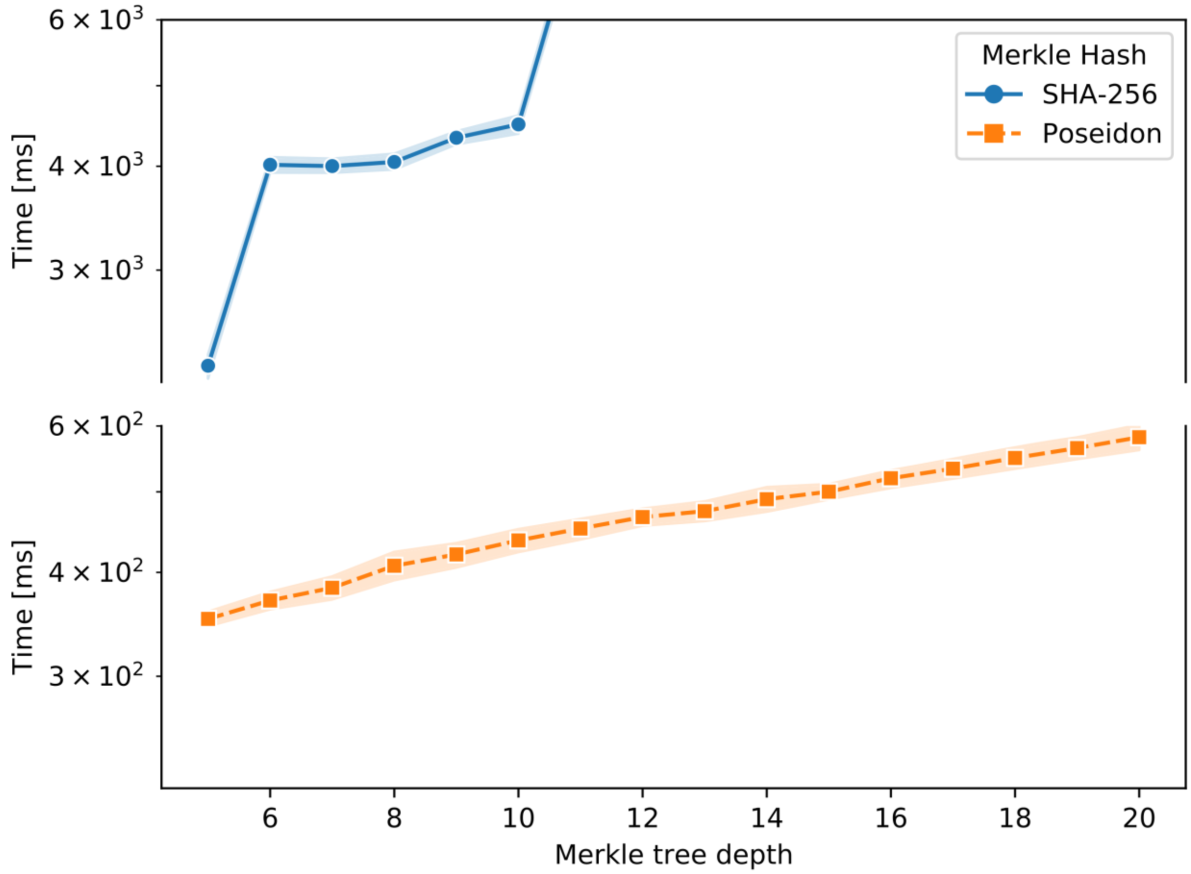

Figure 3: Benchmarks for proof creation during showings for various depths of the Merkle tree. The time is displayed as mean and standard deviation over 100 runs.

Publication details: **Privacy-Preserving eID Derivation to Self-Sovereign Identity Systems with Offline Revocation.** Andreas Abraham *(TUG), Karl Koch (TUG), Stefan More (TUG), Sebastian Ramacher (AIT), Miha Stopar (XLAB)*. In: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2021, pp. 506-513.

3 Results from D4.1

To give a complete picture of all research results related to challenges raised by KRAKEN, data marketplaces in general and SSI, we restate all results from D4.1 [1]. The following extended abstracts and summaries are taken verbatim from D4.1. The full versions of all papers are all available as open access either on IACR's ePrint service or on arXiv.

3.1 CCA-Secure (Puncturable) KEMs from Encryption With Non-Negligible Decryption Errors

Public-key encryption (PKE) schemes or key-encapsulation mechanisms (KEMs) are fundamental cryptographic building blocks to realize secure communication protocols. In particular, PKE is essential for non-interactive end-to-end secure data exchange. There are several known transformations that generically turn weakly secure schemes (e.g., indistinguishability against chosen plaintext attacks) into strongly (i.e., indistinguishability against chosen ciphertext attacks) secure ones. While most of these transformations require the weakly secure scheme to provide perfect correctness, i.e., every well-formed ciphertext can be decrypted, Hofheinz, Hövelmanns, and Kiltz (HHK) [3] have recently shown that variants of the Fujisaki-Okamoto (FO) transform can work with schemes that have negligible correctness error in the (quantum) random oracle model ((Q)ROM). While many recent schemes in the NIST PQC use variants of these transformations, some of their Chosen Plaintext Attack (CPA)-secure versions even have a non-negligible correctness error and so do not satisfy the requirements to apply the techniques of HHK.

In this work, we study the setting of generically transforming PKE schemes with potentially large, i.e., non-negligible, correctness error to ones having negligible correctness error. In an asymptotic setting, this question was studied by Dwork, Naor and Reingold [4]. Our goal is to come up with practically efficient compilers in a concrete setting. First, we show how to generically transform weakly secure deterministic or randomized PKEs into Chosen Ciphertext Attack (CCA)-secure KEMs in the (Q)ROM using variants of the HHK techniques. This applies to essentially all candidates of the NIST PQC based on lattices and codes with non-negligible error. In our extensive analysis, we show that our techniques improve some of the code-based candidates. Second, we apply our techniques to identity-based encryption (IBE) schemes from lattices and codes with (non-)negligible correctness error. Thereby we generically achieve the first post-quantum secure Bloom Filter KEMs which were proposed by Derler et al. [2] and inherently have a non-negligible correctness error. BFKEMs are a building block to construct fully forward-secret zero round-trip time key-exchange protocols.

Publication details: **CCA-Secure (Puncturable) KEMs from Encryption with Non-Negligible Decryption Errors.** *Valerio Cini (AIT), Sebastian Ramacher (AIT), Daniel Slamanig (AIT), Christoph Striecks (AIT).* In: Moriai S., Wang H. (eds) Advances in Cryptology – ASIACRYPT 2020. ASIACRYPT 2020. Lecture Notes in Computer Science, vol 12491. Springer, Cham. Open access: https://eprint.iacr.org/2020/1548.

3.2 An Attack on Some Signatures Schemes Constructed From Five-Pass Identification Schemes

Many popular signature schemes are constructed by taking an interactive identification scheme and making it non-interactive by using the Fiat-Shamir transformation, a decade old standard technique. While the security of the Fiat-Shamir transformation is well understood for traditional 3-pass identification schemes (an identification scheme consisting of 3 messages in total), an increasing number of proposed signature schemes are instead built from 5-pass identification protocols. In this work, we investigate the concrete security of signature schemes build from 5-pass identification schemes. We show a generic attack that uses the nature of how parallel repetitions are used to boost the soundness of the identification scheme to cryptographic security levels by splitting the attack cost between the different phases of the identification scheme. While our attack reduces the concrete security of schemes, it still has exponential runtime and can be mitigated by increasing the number of internal parallel repetitions of the identification scheme.

We apply our attack to MQDSS, a second-round candidate in the current NIST post-quantum standardization project and show that a forgery for their proposed 128-bit parameter set can be produced with about 2^95 hash function calls. The designers acknowledged our attack and in turn increased the number of internal repetitions by about 40%, following our proposal. However, this change in turn reduced the performance of MQDSS and it did not advance into the third round of the NIST post-quantum standardization project, highlighting the practical impact of this work.

Finally, we generalize the attack and apply it to other schemes from the literature. The parameter sets of these schemes already have been updated to take our attack into account.

Publication details: **An Attack on Some Signature Schemes Constructed from Five-Pass Identification Schemes.** *Daniel Kales (TUG), Greg Zaverucha (external).* In: Cryptology and Network Security - 19th International Conference, CANS 2020, Vienna, Austria, December 14-16, 2020, Proceedings. Open access: https://eprint.iacr.org/2020/837.

3.3 Banquet: Short and Fast Signatures from AES

Existing post-quantum signatures can be based on different hardness assumptions such as lattice problems, code-based cryptography, or the hardness of solving multivariate quadratic equation systems. A very conservative choice is to build signatures only from symmetric-key primitives such as block ciphers and hash functions. These constructions include Picnic and SPHINCS [20], both candidates in the ongoing NIST PQC project.

Picnic is built using the novel approach of proving knowledge of a block cipher secret key for a given public plaintext-ciphertext pair and the internal complexity of the used block cipher is the main factor in the final size of the signature. Picnic therefore uses LowMC internally, a relatively recent design which is optimized for evaluations in contexts such as the used proof system. LowMC provides performance improvements of up to 5x when compared to using standard primitives such as AES, however as a tradeoff, LowMC has not received the 20 years of combined cryptanalysis that AES has.

In our work, we propose a Picnic-style signature scheme based around AES instead of LowMC. We build on previous work, BBQ [21], and improve on their ideas by proposing a new proof system that works well with the internal structure of the AES Sbox (and the field inversion contained therein). This results in signatures from conservative and standardized primitives that approach Picnic's signature sizes with lower performance or can match Picnic's performance at the cost of larger signatures. In comparison to previous AES-based signatures, we improve on the current state-of-the-art (BBQ) by a factor of more than 2 in signature size and provide an open-source implementation.

Publication details: **Banquet: Short and Fast Signatures from AES.** *Carsten Baum (external), Cyprien Delpech de Saint Guilhem (external), Daniel Kales (TUG), Emmanuela Orsini (external), Peter Scholl (external), Greg Zaverucha (external).* In: Public-Key Cryptography - PKC 2021 - 24th IACR International Conference on Practice and Theory of Public Key Cryptography, Virtual Event, May 10-13, 2021, Proceedings, Part I. Open access: https://eprint.iacr.org/2021/068.

3.4 Fine-Grained Forward Secrecy: Allow-List/Deny-List Encryption and Applications

Forward secrecy is an important feature for modern cryptographic systems and is widely used in secure messaging such as Signal and WhatsApp as well as in common Internet protocols such as Transport Layer Security (TLS), IPSec, WireGuard or Secure Shell Protocol (SSH). The benefit of forward secrecy is that the damage in case of key-leakage is mitigated. Forward-secret encryption schemes provide security of past ciphertexts even if a secret key leaks, which is interesting in settings where cryptographic keys often reside in memory for quite a long time and could be extracted by an adversary, e.g., in cloud computing. The recent concept of PE [5] provides a versatile generalization of forward-secret encryption: it allows to puncture secret keys with respect to ciphertexts to prevent the future decryption of these ciphertexts.

We introduce the abstraction of allow-list/deny-list encryption schemes and classify different types of PE schemes using this abstraction. Based on our classification, we identify and close a gap in existing work by introducing a novel variant of PE which we dub Dual-Form Puncturable Encryption (DFPE). DFPE significantly enhances and, in particular, generalizes previous variants of PE by allowing an interleaved application of allow- and deny-list operations.

We present a construction of DFPE in prime-order bilinear groups, discuss a direct application of DPFE for enhancing security guarantees within Cloudflare's Geo Key Manager, and show its generic use to construct forward-secret IBE and forward-secure digital signatures.

Publication details: **Fine-Grained Forward Secrecy: Allow-List/Deny-List Encryption and Applications.** *David Derler (external), Sebastian Ramacher (AIT), Daniel Slamanig (AIT), Christoph Striecks (AIT).* In: Financial Cryptography and Data Security. FC 2021. Lecture Notes in Computer Science, Springer, Cham. 2021 (to appear). Open access: https://eprint.iacr.org/2019/912.

3.5 Updatable Signatures and Message Authentication Codes

Cryptographic objects with updating capabilities have been proposed by Bellare, Goldreich and Goldwasser [15] under the umbrella of incremental cryptography. They have recently seen increased interest, motivated by theoretical questions [10] as well as concrete practical motivations [11], [12], [13]. In this work, the form of updatability we are particularly interested in is that primitives are key-updatable and allow to update old cryptographic objects, e.g., signatures or message authentication codes, from the old key to the updated key at the same time without requiring full access to the new key (i.e., only via a so-called update token).

Inspired by the rigorous study of updatable encryption by Lehmann and Tackmann [11] and Boyd et al. [14], we introduce a definitional framework for updatable signatures (USs) and universal message authentication codes (UMACs). We discuss several applications demonstrating that such primitives can be useful in practical applications, especially around key rotation in various domains, as well as serve as building blocks in other cryptographic schemes. We then turn to constructions and our focus is on ones that are secure and practically efficient. In particular, we provide generic constructions from key-homomorphic primitives (signatures and Pseudorandom Functions (PRFs)) as well as direct constructions. This allows us to instantiate these primitives from various assumptions such as Decisional Diffie-Hellman or Computational Diffie-Hellman (latter in bilinear groups), or the Ring Learning With Errors ((R)LWE) and the Short Integer Solution (SIS) assumptions. As an example, we obtain highly practical US schemes from Boneh-Lynn-Shacham (BLS) signatures or UMAC schemes from the Naor-Pinkas-Reingold PRF.

Publication details: **Updatable Signatures and Message Authentication Codes.** *Valerio Cini (AIT), Sebastian Ramacher (AIT), Daniel Slamanig (AIT), Christoph Striecks (AIT), Erkan Tairi (external).* In: Garay J.A. (eds) Public-Key Cryptography – PKC 2021. PKC 2021. Lecture Notes in Computer Science, vol 12710. Springer, Cham. 2021. Open access: https://eprint.iacr.org/2021/365.

3.6 Privacy-preserving Analytics for Data Markets using MPC

"Data markets have the potential to foster new data-driven applications and help growing data-driven businesses. When building and deploying such markets in practice, regulations such as the European Union's General Data Protection Regulation (GDPR) impose constraints and restrictions on these markets especially when dealing with personal or privacy-sensitive data." [17]. Also in KRAKEN we deal with personal data, and the protection of this personal data is very important from a security as well as a privacy point of view; (1) to keep the users' security and privacy intact and, furthermore, (2) to be GDPR-compliant.

In KRAKEN we leverage Functional Encryption (FE) and Multi-Party Computation (MPC) to enable privacy-preserving data analytics. Users encrypt (FE) or secret-share and then encrypt (MPC) their data before uploading it to an (external) cloud. To ensure the data-origin's authenticity, we leverage Group Signatures. Group signatures have the (nice) property, that users can authenticate their data by signing it and yet they stay anonymous within the group. Only a kind of "opening authority", like a judge, could identify a user, e.g., in a dispute during a lawsuit. On the other hand, if this is an issue, the group's "master key" could be, e.g., thrown away or only used via MPC. A data producer only gets the analytics' result, yet it is still possible to verify the data-origin's authenticity and if the correct function has been applied. This verification is achieved by leveraging zero-knowledge proofs of knowledge, which is further explained, e.g., in Section 3.10. Moreover, with our solution, the KRAKEN marketplace does not learn about the users' data nor the analytics' result in the MPC case.

"In this paper, we present a candidate architecture for a privacy-preserving personal data market, relying on cryptographic primitives such as multi-party computation (MPC) capable of performing privacy-preserving computations on the data. Besides specifying the architecture of such a data market, we also present a privacy-risk analysis of the market following the LINDDUN methodology." - [17]. Figure 4 gives an overview of KRAKEN's entities and crypto components for the data-analytics-via-MPC case including the concrete choice of cryptographic building blocks and their interaction. Figure 1 gives an overview of KRAKEN's entities and data flows of the MPC case; from data gathering to data-analytic results.

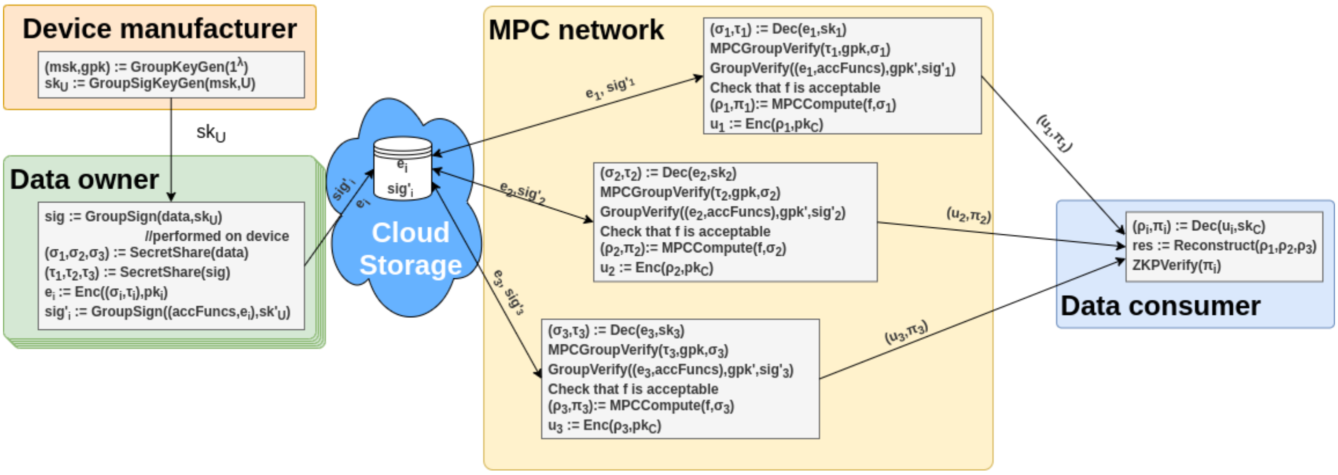
Device manufacturer
$(msk,gpk) := GroupKeyGen(1^{\lambda})$
$sk_U := GroupSigKeyGen(msk,U)$
sk_U
Data owner
$sig := GroupSign(data,sk_U)$
//performed on device
$(\sigma_1,\sigma_2,\sigma_3) := SecretShare(data)$
$(\tau_1,\tau_2,\tau_3) := SecretShare(sig)$
$e_i := Enc((\sigma_i,\tau_i),pk_i)$
$sig'_i := GroupSign((accFuncs,e_i),sk'_U)$
sig'_i
e_i
e_i
sig'_i
Cloud Storage
MPC network
e_1, sig'_1
$(\sigma_1,\tau_1) := Dec(e_1,sk_1)$
$MPCGroupVerify(\tau_1,gpk,\sigma_1)$
$GroupVerify((e_1,accFuncs),gpk',sig'_1)$
Check that f is acceptable
$(\rho_1,\pi_1) := MPCCompute(f,\sigma_1)$
$u_1 := Enc(\rho_1,pk_C)$
e_2,sig'_2
$(\sigma_2,\tau_2) := Dec(e_2,sk_2)$
$MPCGroupVerify(\tau_2,gpk,\sigma_2)$
$GroupVerify((e_2,accFuncs),gpk',sig'_2)$
Check that f is acceptable
$(\rho_2,\pi_2) := MPCCompute(f,\sigma_2)$
$u_2 := Enc(\rho_2,pk_C)$
e_3, sig'_3
$(\sigma_3,\tau_3) := Dec(e_3,sk_3)$
$MPCGroupVerify(\tau_3,gpk,\sigma_3)$
$GroupVerify((e_3,accFuncs),gpk',sig'_3)$
Check that f is acceptable
$(\rho_3,\pi_3) := MPCCompute(f,\sigma_3)$
$u_3 := Enc(\rho_3,pk_C)$
(u_1,π_1)
(u_2,π_2)
(u_3,π_3)
$(\rho_i,\pi_i) := Dec(u_i,sk_C)$
$res := Reconstruct(\rho_1,\rho_2,\rho_3)$
$ZKPVerify(\pi_i)$
Data consumer

Figure 4: Overview of KRAKEN's entities and crypto components for the data-analytics-via-MPC case.

Publication details: **Privacy-Preserving Analytics for Data Markets Using MPC.** *Karl Koch (TUG), Stephan Krenn (AIT), Donato Pellegrino (TX), Sebastian Ramacher (AIT)*. In: Friedewald M., Schiffner S., Krenn S. (eds) Privacy and Identity Management. Privacy and Identity 2020. IFIP Advances in Information and Communication Technology, vol 619. Springer, Cham. Open access: https://arxiv.org/abs/2103.03739.

3.7 Lift-and-Shift: Obtaining Simulation Extractable Subversion and Updatable SNARKs

Zero-knowledge proofs and in particular succinct non-interactive zero-knowledge proofs (so called zk-SNARKs) are getting increasingly used in real-world applications, with cryptocurrencies being the prime example. Simulation extractability (SE) is a strong security notion for zk-SNARKs which informally ensures non-malleability of proofs. The high importance of this property is underpinned by various attacks against the malleability of cryptographic primitives in the past [6]. Another problematic issue for the practical use of zk-SNARKs is the requirement of a fully trusted setup, as especially for large-scale decentralized applications because finding a trusted party that runs the setup is practically impossible or requires large-scale ceremonies including many different parties to set up the parameters [7]. Quite recently, the study of approaches to relax or even remove the trust in the setup procedure has been initiated [8]. This line of research introduced subversion-resistant und updatable Non-Interactive Zero Knowledges (NIZKs) (and zk-SNARKs). For subversion resistance, one considers subversion soundness, i.e., soundness holds even if the Common Reference String (CRS) is subverted, and subversion zero-knowledge, i.e., zero-knowledge holds even if the CRS is subverted. Note however, that it is impossible for both notions to holds simultaneously. For updatable NIZKs the approach is different. There the idea is that it is possible to update the CRS such that knowledge of trapdoors with respect to an old CRS will not help in breaking soundness or zero-knowledge of the new CRS. So far SE-SNARKs that are subversion-resistant or updatable are only constructed in an ad-hoc manner and no generic techniques are available.

We are interested in generic techniques for constructing updatable and subversion-resistant SE-SNARKs. Therefore, we firstly revisit the only available lifting technique due to Kosba et al. [9] (called COCO) to generically obtain SE-SNARKs. By exploring the design space of many recently proposed SNARK- and succinct transparent argument of knowledge (STARK)-friendly symmetric-key primitives we thereby achieve significant improvements in the prover computation and proof size. Unfortunately, the COCO framework as well as our improved version (called OCOCO) is not compatible with updatable SNARKs. Consequently, we propose a novel generic lifting transformation called LAMASSU. It is built using different underlying ideas compared to COCO (and OCOCO). In contrast, it only requires key-homomorphic signatures (which allow to shift keys) covering well studied schemes such as Schnorr or Elliptic Curve Digital Signature Algorithm (ECDSA). This makes LAMASSU highly interesting, as by using the novel concept of so-called updatable signatures, we can prove that LAMASSU preserves the subversion and in particular updatable properties of the underlying zk-SNARK. This makes LAMASSU the first technique to also generically obtain SE subversion and updatable SNARKs. As its performance compares favorably to OCOCO, LAMASSU is an attractive alternative that in contrast to COCO is only based on well-established cryptographic assumptions.

Publication details: **Lift-and-Shift: Obtaining Simulation Extractable Subversion and Updatable SNARKs Generically.** *Behzad Abdolmaleki (external), Sebastian Ramacher (AIT), Daniel Slamanig (AIT).* In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20). Association for Computing Machinery, New York, NY, USA, 2020. Open access: https://eprint.iacr.org/2020/062.

3.8 Short-Lived Forward-Secure Delegation for TLS

Delegations are an important concept in many trust management systems, enabling more flexible authorization mechanisms than static access control. Delegations are also an integral part of many real-world processes in government and industry, so supporting delegations is crucial for systems used to digitalize those processes. One simple example for a delegation in the context of KRAKEN is if the subject of some data-set wants to enable another entity to manage or share these data. Doing this on the KRAKEN marketplace requires that the other entity (delegate) can use an SSI credential to act on behalf of the delegator.

When a delegate acts on behalf of a delegator, they have to authenticate themselves as the delegator to access some restricted resources. In some use cases this can be realized by issuing a credential to the delegate which authorizes her/him to represent the delegator – requiring modifications to the VC verification logic to support those special credentials. Other use cases require that the delegate has access to the secret key of the delegator's Decentralized Identifier (DID), which represents a security issue. To curb this problem, multiple workarounds exist to realize a delegation of the authentication.

In this paper, we present a solution that works without authorization-credentials, renders key sharing unnecessary and reduces the need for workarounds. By adapting identity-based signatures to this setting, our solution offers short-lived delegations. Additionally, by enabling forward-security, existing delegations remain valid even if the delegator's secret key leaks. To demonstrate the scheme's feasibility, we provide an implementation of the scheme. We furthermore show the scheme's versatility by discussing an integration into a TLS stack. We also evaluate the performance of the scheme's implementation, concluding that an efficient implementation incurs less overhead than a typical network round trip.

Publication details: **Short-Lived Forward-Secure Delegation for TLS.** *Lukas Alber (TUG), Stefan More (TUG), Sebastian Ramacher (AIT).* In CCSW'20: Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop, Virtual, November 2020. Open access: https://arxiv.org/abs/2009.02137.

3.9 Multi-Party Revocation in Sovrin: Performance through Distributed Trust

Cryptographic accumulators are a common building block in identity systems, e.g., to accumulate all allowed identifiers into a central value called the *accumulator*. Parties can then prove that their identity is included in this accumulator using zero-knowledge techniques to show that they indeed belong to some specific group.

Cryptographic accumulators providing constant-size accumulation values are traditionally built using trapdoor functions, such as ones based on RSA or bilinear pairings. However, these trapdoors have some practical considerations that must be kept in mind when used. First, the trapdoor value used during the initial setup must be destroyed and not be known to any party after that point, otherwise the security of the system is not guaranteed. Second, without the knowledge of the secret trapdoor, many operations in the accumulator algorithms are much more expensive to compute. This puts some limits on the size of the sets that are accumulated in practice, where operations on large sets can take hours.

To fix both issues, we propose using our Multi-Party Linear-Secret-Shared Accumulators. They use multiple independent parties to generate the trapdoor secret in a secret-shared form, solving the issue of requiring a trusted party to set up the public parameters. Even further, instead of forgetting the secret trapdoor information, the parties can work together in the online phase to compute operations on the accumulator using the secret-shared trapdoor information from the setup phase. This allows the accumulation of large sets while still offering practical performance.

Publication details: **Multi-Party Revocation in Sovrin: Performance through Distributed Trust.** *Lukas Helminger (TUG), Daniel Kales (TUG), Sebastian Ramacher (AIT), Roman Walch (TUG)*. In: Topics in Cryptology - CT-RSA 2021 - The Cryptographers' Track at the RSA Conference 2021, San Francisco, CA, USA, May 17-20, 2021, Proceedings. Open access: https://eprint.iacr.org/2020/724.

3.10 Poseidon: A New Hash Function for Zero-Knowledge Proof Systems

In the KRAKEN architecture, which is also envisioned by [17], data consumers only get the analysis result, and yet they are able to verify the data-origin's authenticity. Furthermore, they also want to verify if the correct function on the data has been applied. For the verification of both data-origin's authenticity and the correct function, among other things, zero-knowledge proofs of knowledge (ZK-PoK) are leveraged, specifically succinct non-interactive arguments of knowledge (SNARKs). Another use case for ZK-PoKs is the proof of the knowledge of a hash's preimage.

"A zero-knowledge proof of knowledge (ZK-PoK) is a two-party protocol between a prover and a verifier, which achieves two intuitively contradictory goals: it allows the prover to convince the verifier that she knows a secret piece of information, while at the same time revealing no further information than what is already revealed by the claim itself." - [17] Thus, e.g., for the hash's preimage, we are able to prove the knowledge of the preimage without actually showing it to the verifier. The verifier only gets the hash and the ZK proof. The proof of a hash's preimage is used, e.g., for identity proofs based on an identity assertion from a legal authority [18]. Whereas "traditional" standardized hash functions, such as SHA-256, are very efficient for creating a hash in a "traditional" setting, those hash functions are not well suited for, e.g., a ZK proof of a hash's preimage. This new requirement heralds the dawn of a new era in (modern/current) cryptography: ZK-friendly hash functions. Or as also said by this paper: *"The area of practical computational integrity proof systems, like SNARKs, STARKs, Bulletproofs, is seeing a very dynamic development with several constructions having appeared recently with improved properties and relaxed setup requirements. Many use cases of such systems involve, often as their most expensive part, proving the knowledge of a preimage under a certain cryptographic hash function, which is expressed as a circuit over a large prime field. A notable example is a zero-knowledge proof of coin ownership in the Zcash cryptocurrency, where the inadequacy of the SHA-256 hash function for such a circuit caused a huge computational penalty."* [16].

"In this paper, we present a modular framework and concrete instances of cryptographic hash functions which work natively with GF(p) objects. Our hash function Poseidon uses up to 8x fewer constraints per message bit than Pedersen Hash. Our construction is not only expressed compactly as a circuit but can also be tailored for various proof systems using specially crafted polynomials, thus bringing another boost in performance. We demonstrate this by implementing a 1-out-of-a-billion membership proof with Merkle trees in less than a second by using Bulletproofs." - [16]. Furthermore, our developed ZK-friendly hash function, Poseidon, is used in a proposal for a modern privacy-preserving eID and self-sovereign identity system. This proposal is ongoing research and part of our future work (cf. Section 5.1).

Publication details: **Poseidon: A New Hash Function for Zero-Knowledge Proof Systems**. *Lorenzo Grassi (external), Dmitry Khovratovich (external), Christian Rechberger (TUG), Arnab Roy (external), Markus Schofnegger (TUG)*. In: 30th USENIX Security Symposium (USENIX Security 2021). Open access: https://eprint.iacr.org/2019/458.

4 Conclusion

The data-marketplace architecture and SSI systems raise a wide variety of research questions. As part of the work in KRAKEN, we have studied some of these questions both in theoretical and applied directions. An overview of the scientific contributions motivated by KRAKEN is available in this document.

Beside the already published contributions, some of the results are still in preparation or currently under submission. In this section we give a short overview on these results. These contributions focus on secure computation on encrypted data and zero-knowledge proofs and their secure combination to build a verifiable and secure multi-party computation system. The outcome of the submissions will be reported in deliverable *D6.8 Final communication, dissemination and standardization report* at the end of the project.

4.1 A Verifiable Multiparty Computation Solver for the Assignment Problem

The assignment problem is an essential problem in many application fields and frequently used to optimize resource usage. The problem is well understood and various efficient algorithms exist to solve the problem. However, it was unclear what practical performance could be achieved, for privacy-preserving implementations based on multiparty computation (MPC) by leveraging more efficient solution strategies than MPC-based simplex solvers for linear programs. We solve this question by implementing and comparing different optimized MPC algorithms to solve the assignment problem for reasonable problem sizes. Our empirical approach revealed various insights to MPC-based optimization and we measured a significant (50x) speedup compared to the known simplex-based approach. Furthermore, we also study the overhead introduced by making the results publicly verifiable by means of non-interactive zero-knowledge proofs. By leveraging modern proof systems, we also achieve significant speedup for proof and verification times compared to the previously proposed approaches as well as compact proof sizes.

Publication details: **A Verifiable Multiparty Computation Solver for the Assignment Problem**. *Thomas Lorünser (AIT), Florian Wohner (AIT), Stephan Krenn (AIT). Under submission.*

4.2 (Inner-Product) Functional Encryption with Updatable Ciphertexts

We propose a novel variant of functional encryption which supports ciphertext updates, dubbed ciphertext updatable functional encryption (CUFE). Such a feature further broadens the practical applicability of the functional encryption paradigm and is carried out via so-called update tokens. However, allowing update tokens requires some care for the security definition as we want that updates can be done by any semi-trusted third party and only on ciphertexts. On the technical level, we build on recent functional encryption schemes with fine-grained access control and linear operations on encrypted data [59] and introduce an additional ciphertext updatability feature.

Our contribution is three-fold:

1. We define our new primitive with a security notion in the adaptive indistinguishability setting. Within CUFE, functional decryption keys and ciphertexts are labeled with tags such that only if the tag of the decryption key and the ciphertext match, then decryption succeeds. Furthermore, we allow ciphertexts to switch their tags to any other tag via update tokens. Such tokens are generated by the main secret key holder and can only be used in the desired direction.

2. We present a generic construction of CUFE for any functionality as well as predicates different from equality testing on tags. This construction satisfies the strongest version of our security model. However, therefore we have to rely on the existence of (probabilistic) indistinguishability obfuscation.
3. We present a practical construction of CUFE for the inner-product functionality from standard assumptions (i.e., LWE) in the random-oracle model. Proving security for such a construction turned out to be non-trivial, particularly when revealing keys for the updated challenge ciphertext is allowed. Overall, such a construction enriches the set of known inner-product functional-encryption schemes with the additional updatability feature of ciphertexts.

Publication details: **(Inner-Product) Functional Encryption with Updatable Ciphertexts.** *Valerio Cini (AIT), Sebastian Ramacher (AIT), Daniel Slamanig (AIT), Christoph Striecks (AIT), Erkan Tairi (external). Under submission.*

4.3 Composable and Simulation-Extractable Compact NIZKs with Updatable Common Reference Strings

Non-interactive zero-knowledge proofs (NIZKs) are a powerful cryptographic primitive. Especially within the cryptocurrency space, they increasingly see real-world adoption in large and complex systems. Due to the specifics of this setting, i.e., the requirement for efficient verifiers and compact proofs, succinct NIZK arguments of knowledge (so called zk-SNARKs) are of particular interest, and research into zk-SNARKs within academia and industry is currently exploding.

The increasing complexity and scale of deployed cryptographic systems comes with the desire to provide various strong properties. One important property that is growing in importance is the composition of secure protocols, e.g., via using the Universal Composability (UC) framework. This allows arbitrary composition of such protocols with other cryptographic building blocks and supports modular constructions. An additional requirement that turns out to be important for NIZKs is ensuring non-malleability of proofs, which can be achieved via the property of simulation extractability (SE). Moreover, when relying on a common reference string (CRS), it is desirable to reduce the trust that needs to be put into the generation of this CRS, which can be achieved via the notions of subversion or updatable CRS.

In this work, we are interested in zk-SNARKs or more generally compact NIZKs providing all these desired properties. This is a tricky task as the UC framework rules out several natural techniques for such a construction. Our main result is to show that achieving these properties is indeed possible in a generic and modular way when slightly relaxing the succinctness properties of zk-SNARKs to those of a compact NIZK, in particular by increasing the proof size of the zk-SNARK by the size of the witness w (and thus obtaining proofs of size |w|+|π|). We will argue that for many practical applications this overhead is perfectly tolerable. Our starting point is a recent framework by Abdolmaleki et al. called Lamassu [58] and for our approach we need to develop a new tool called black-box extractable key-updatable public-key encryption that might be of independent interest.

Publication details: **Composable and Simulation-Extractable compact NIZKs with Updatable Common Reference Strings.** *Behzad Abdolmaleki (external), Noemi Glaeser (external), Sebastian Ramacher (AIT), Daniel Slamanig (AIT). In preparation.*

4.4 Benchmarking of Multi-Party-Computation Programs & Protocols with Various Settings in Envisioned Environments

Theoretically, every program with any input and amount of computation nodes can be run secretly with MPC. However, in reality it is a question about performance: efficiency and practicability. For efficiency, e.g., the runtime or memory consumption are relevant metrics. The shorter the runtime, the higher the likelihood, that the MPC program can be used in practice; the same holds for memory consumption. For practicability, e.g., the rounds of communication and corresponding amount of data, or amount of MPC nodes are relevant metrics. On the one hand, the metrics about practicability usually have a direct impact on the runtime; on the other hand, the more MPC nodes, the higher the chance, that the program execution fails (for synchronous MPC protocols, which are the most popular and useful ones at the moment).

Since MPC programs need to be efficient and practical for real-life applications, reliable benchmarks, which consider the mentioned metrics, become increasingly important.

Hence, we aim to develop an MPC benchmarking framework which allows to test the performance of any MPC program with various settings in the envisioned environment. The settings comprise, e.g., the MPC engine (SCALE, MP-SPDZ, etc.) or MPC protocol (passive vs. active adversaries, honest vs. dishonest majority, etc.). The environment comprises, e.g., the number of players or network (local area network, wide area network, etc.).

Furthermore, due to the close inspection of different MPC programs and their performance for various settings, we might gain new insights into efficient MPC computation blocks for certain programs and envisioned environments.

Publication details: **Benchmarking of Multi-Party-Computation Programs & Protocols with Various Settings in Envisioned Environments.** *Karl Koch (TUG), Christian Rechberger (TUG), Dragoș Rotaru (external). In preparation.*

5 Bibliography & References

[1] KRAKEN Consortium: D4.1: Progress report on cryptographic protocols for privacy-preserving data markets and SSI systems. 2021.

[2] David Derler, Tibor Jager, Daniel Slamanig, Christoph Striecks: Bloom Filter Encryption and Applications to Efficient Forward-Secret 0-RTT Key Exchange. EUROCRYPT 2018.

[3] Dennis Hofheinz, Kathrin Hövelmanns, Eike Kultz: A Modular Analysis of the Fujisaki-Okamoto Transformation. TCC 2017.

[4] Cynthia Dwork, Moni Naor, Omer Reingold: Immunizing Encryption Schemes from Decryption Errors. EUROCRYPT 2004.

[5] Matthew D. Green, Ian Miers: Forward Secure Asynchronous Messaging from Puncturable Encryption. IEEE S&P 2015.

[6] https://www.coindesk.com/study-finds-mt-gox-lost-386-bitcoins-due-transaction-malleability (accessed on July 2021).

[7] Sean Bowe, Ariel Gabizon, Matthew D. Green: A Multi-Party Protocol for Constructing the Public Parameters of the Pinoccio zk-SNARK. FC 2018 Workshops.

[8] Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. NIZKs with an untrusted CRS: Security in the face of parameter subversion. In Jung Hee Cheon and Tsuyoshi Takagi, editors, ASIACRYPT 2016, Part II, volume 10032 of LNCS, pages 777–804. Springer, Heidelberg, December 2016.

[9] Ahmed Kosba, Zhichao Zhao, Andrew Miller, Yi Qian, Hubert Chan, Charalampos Papamanthou, Rafael Pass, Abhi Sehlat, Elaine Shi: COCO: A Framework for Building Composable Zero-Knowledge Proofs. ePrint 2015/1093

[10] Prabhanjan Ananth, Aloni Cohen, Abhishek Jain: Cryptography with Updates. EUROCRYPT 2017

[11] Anja Lehmann, Björn Tackmann: Updatable Encryption with Post-Compromise Security. EUROCRYPT 2018

[12] Jens Groth, Markulf Kohlweiss, Mary Maller, Sarah Meiklejohn, and Ian Miers. Updatable and universal common reference strings with applications to zk-snarks. In Shacham, H., Boldyreva, A. (eds) Advances in Cryptology - CRYPTO 2018, volume 10993 of LNCS, pages 698-728. Springer, 2018.

[13] Michael Klooß, Anja Lehmann, Andy Rupp: (R)CCA Secure Updatable Encryption with Integrity Protection. EUROCRYPT 2019

[14] Colin Boyd, Gareth T. Davies, Kristian Gjøsteen, Yao Jiang: Fast and Secure Updatable Encryption. CRYPTO 2020

[15] Mihir Bellare, Oded Goldreich, Shafi Goldwasser: Incremental Cryptography: The Case of Hashing and Signing. CRYPTO 1994

[16] Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, Markus Schofnegger: Poseidon: A new hash function for zero-knowledge proof systems. 30th USENIX Security Symposium, 2021.

[17] Karl Koch, Stephan Krenn, Donato Pellegrino, Sebastian Ramacher: Privacy-preserving Analytics for Data Markets using MPC. IFIP International Summer School on Privacy and Identity Management, 2020.

[18] Andreas Abraham, Felix Hörandner, Olamide Omolola, Sebastian Ramacher: Privacy-Preserving eID Derivation for Self-Sovereign Identity Systems. International Conference on Information and Communications Security (ICICS), 2019.

[19] https://sovrin.org/ (accessed on July 2021).

[20] Daniel J. Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, Peter Schwabe: The SPHINCS+ Signature Framework. CCS 2019

[21] Cyprien Delpech de Saint Guilhem, Lauren De Meyer, Emmanuela Orsini, Nigel P. Smart: BBQ: Using AES in Picnic Signatures. SAC 2019

[22] Mina Deng, Kim Wuyts, Riccardo Scandariato, Bart Preneel, Wouter Joosen: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requir. Eng. 16(1): 3-32 (2011)

[23] KRAKEN Consortium: D2.2 Intermediate KRAKEN architecture. 2020

[24] KRAKEN Consortium: D2.4 KRAKEN intermediate technical design. 2020

[25] KRAKEN Consortium: D5.1 Initial Pilot Marketplaces User Stories. 2020

[26] KRAKEN Consortium: D3.1 Self sovereign identity solution. First Release. 2021

[27] https://z.cash/technology/jubjub/ (accessed on July 2021)

[28] https://ec.europa.eu/cefdigital/wiki/display/EBSIDOC/1.3.2.+Data+modelling+ESSIF+v2 (accessed on July 2021)

[29] Andreas Abraham, Karl Koch, Stefan More, Sebastian Ramacher, Miha Stopar: Privacy-Preserving eID Derivation to Self-Sovereign Identity Systems with Offline Revocation. 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2021, pp. 506-513.

[30] https://github.com/sebastinas/ppeid-bench (accessed on April 2022)

[31] https://github.com/matter-labs/bellman (accessed on April 2022)

[32] Behzad Abdolmaleki, Hamidreza Khoshakhlagh, and Helger Lipmaa. Smooth zero-knowledge hash functions. Cryptology ePrint Archive, Report 2021/653, 2021. https://eprint.iacr.org/2021/653.

[33] Behzad Abdolmaleki, Karim Baghery, Helger Lipmaa, and Michal Zajac. A subversion resistant SNARK. In Tsuyoshi Takagi and Thomas Peyrin, editors, ASIACRYPT 2017, Part III, volume 10626 of LNCS, pages 3–33. Springer, Heidelberg, December 2017.

[34] Behzad Abdolmaleki, Helger Lipmaa, Janno Siim, and Michal Zajac. On QA-NIZK in the BPK model. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, PKC 2020, Part I, volume 12110 of LNCS, pages 590–620. Springer, Heidelberg, May 2020.

[35] Fabrice Benhamouda, Olivier Blazy, Céline Chevalier, David Pointcheval, and Damien Vergnaud. New techniques for SPHFs and efficient one-round PAKE protocols. In Ran Canetti and Juan A.

Garay, editors, CRYPTO 2013, Part I, volume 8042 of LNCS, pages 449–475. Springer, Heidelberg, August 2013.

[36] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In 2014 IEEE Symposium on Security and Privacy, pages 459–474. IEEE, 2014

[37] Fabrice Benhamouda and David Pointcheval. Trapdoor smooth projective hash functions. Cryptology ePrint Archive, Report 2013/341, 2013. http://eprint.iacr.org/2013/341.

[38] Eli Ben-Sasson, Alessandro Chiesa, Matthew Green, Eran Tromer, and Madars Virza. Secure sampling of public parameters for succinct zero knowledge proofs. In 2015 IEEE Symposium on Security and Privacy, pages 287–304. IEEE, 2015.

[39] Matteo Campanelli, Dario Fiore, and Anaïs Querol. LegoSNARK: Modular design and composition of succinct zero-knowledge proofs. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, ACM CCS 2019, pages 2075–2092. ACM Press, November 2019.

[40] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, EUROCRYPT 2002, volume 2332 of LNCS, pages 45–64. Springer, Heidelberg, April / May 2002.

[41] Georg Fuchsbauer. Subversion-zero-knowledge SNARKs. In Michel Abdalla and Ricardo Dahab, editors, PKC 2018, Part I, volume 10769 of LNCS, pages 315–347. Springer, Heidelberg, March 2018.

[42] Helger Lipmaa. Simulation-extractable snarks revisited. Cryptology ePrint Archive, Report 2019/612, 2019. https://eprint.iacr.org/2019/612.

[43] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. SIAM Journal on computing, 18(1):186–208, 1989.

[44] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, EUROCRYPT 2016, Part II, volume 9666 of LNCS, pages 305–326. Springer, Heidelberg, May 2016.

[45] Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In Kazue Sako and Palash Sarkar, editors, ASIACRYPT 2013, Part I, volume 8269 of LNCS, pages 1–20. Springer, Heidelberg, December 2013.

[46] Eike Kiltz and Hoeteck Wee. Quasi-adaptive NIZK for linear subspaces revisited. In Elisabeth Oswald and Marc Fischlin, editors, EUROCRYPT 2015, Part II, volume 9057 of LNCS, pages 101–128. Springer, Heidelberg, April 2015.

[47] Helger Lipmaa. Key-and-argument-updatable QA-NIZKs. In Clemente Galdi and Vladimir Kolesnikov, editors, SCN 20, volume 12238 of LNCS, pages 645–669. Springer, Heidelberg, September 2020.

[48] Behzad Abdolmaleki, Helger Lipmaa, Janno Siim, and Michal Zajac. On subversion-resistant snarks. J. Cryptol., 34(3):17, 2021.

[49] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, EUROCRYPT 2013, volume 7881 of LNCS, pages 626– 645. Springer, Heidelberg, May 2013.

[50] Alonso González, Alejandro Hevia, and Carla Ràfols. QA-NIZK arguments in asymmetric groups: New tools and new constructions. In Tetsu Iwata and Jung Hee Cheon, editors, ASIACRYPT 2015, Part I, volume 9452 of LNCS, pages 605–629. Springer, Heidelberg, November / December 2015.

[51] Jens Groth and Mary Maller. Snarky signatures: Minimal signatures of knowledge from simulation-extractable SNARKs. In Jonathan Katz and Hovav Shacham, editors, CRYPTO 2017, Part II, volume 10402 of LNCS, pages 581–612. Springer, Heidelberg, August 2017.

[52] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, EUROCRYPT 2006, volume 4004 of LNCS, pages 339–358. Springer, Heidelberg, May / June 2006.

[53] Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, ASIACRYPT 2010, volume 6477 of LNCS, pages 321–340. Springer, Heidelberg, December 2010.

[54] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, EUROCRYPT 2008, volume 4965 of LNCS, pages 415–432. Springer, Heidelberg, April 2008.

[55] Kristiyan Haralambiev. Efficient Cryptographic Primitives for Non-Interactive Zero- Knowledge Proofs and Applications. PhD thesis, New York University, 2011.

[56] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In 24th ACM STOC, pages 723–732. ACM Press, May 1992.

[57] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In Ronald Cramer, editor, TCC 2012, volume 7194 of LNCS, pages 169–189. Springer, Heidelberg, March 2012.

[58] Behzad Abdolmaleki, Sebastian Ramacher, Daniel Slamanig. Lift-and-Shift: Obtaining Simulation Extractable Subversion and Updatable SNARKs Generically. CCS 2020.

[59] Michel Abdalla, Dario Catalano, Romain Gay, Bogdan Ursu. Inner-Product Functional Encryption with Fine-Grained Access Control. Asiacrypt 2020.

KRAKEN

Atos

FONDAZIONE BRUNO KESSLER

AIT AUSTRIAN INSTITUTE OF TECHNOLOGY

sic

LYNKEUS. STRATEGY CONSULTING | BLOCKCHAIN & SMART CONTRACTS | DATA ANALYTICS

XLAB

TX

KU LEUVEN CiTiP CENTRE FOR IT & IP LAW

IAIK TU Graz

InfoCert TINEXTA GROUP

@KrakenH2020

Kraken H2020

www.krakenh2020.eu

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871473