Report from Dagstuhl Seminar 19042

Practical Yet Composably Secure Cryptographic Protocols

Edited by

Jan Camenisch[1], Ralf Küsters[2], Anna Lysyanskaya[3], and Alessandra Scafuro[4]

1 **Dfinity Foundation – Zug, CH,** jan@dfinity.org
2 **Universität Stuttgart, DE,** ralf.kuesters@sec.uni-stuttgart.de
3 **Brown University – Providence, US,** anna@cs.brown.edu
4 **North Carolina State University – Raleigh, US,** ascafur@ncsu.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 19042 "Practical Yet Composably Secure Cryptographic Protocols".

The workshop's main aim was to enhance the community's understanding of (1) what a good model was for how various protocols and systems co-exist in a larger system; (2) how to model important tasks and security protocols in such a model; (3) how to prove security of protocols in such a model.

Seminar January 20–25, 2019 – http://www.dagstuhl.de/19042
2012 ACM Subject Classification Security and privacy → Cryptography
Keywords and phrases applied cryptography, cryptographic protocols, practical protocols, provably secure protocols, security models, universally composability
Digital Object Identifier 10.4230/DagRep.9.1.88
Edited in cooperation with Sophia Yakoubov

1 Executive Summary

Jan Camenisch
Ralf Küsters
Anna Lysyanskaya
Alessandra Scafuro

We began by having survey talks on four research threads that had laid foundations of such models. Specifically, Ran Canetti presented his Universal Composability model, Dennis Hofheinz presented his work on the GNUC model, Ralf Küsters presented his IITM/iUC model, and Ueli Maurer presented the model of Constructive Cryptography.

Following these tutorials, we had several talks on how specific security goals and protocols are modeled and proved secure. Björn Tackmann presented a way to model a zero-knowledge proof protocol that made statements about knowledge of certain inputs to ideal functionalities. Manu Drijvers presented a way to model the global random oracle that can be used by participants in different protocols in a composable way.

Once the details of the specific models and how to use them were fresh in everyone's minds, we split up into working groups. In order to do this, we first had a discussion on what problems we believed were worth tackling; we proposed many problems, and then agreed to discuss a subset of them.

The topics explored by the working groups are discussed in detail below, in the "results" section of this report. The following additional topics were proposed for discussion (but were not discussed):

- Model asynchrony and time
- Anonymous communication
- Global random oracles in CC
- Secure Message Transfer in various model
- Concrete security in UC/IITM
- Finalise $\mathcal{F}_{\mathsf{sig}}$ (with reasons why certain choices are better than others)

Additionally, we had several talks on recent and ongoing research projects. Marc Fischlin on composition of key agreement; Markulf Kohlweiss on structuring game-based proofs; Ran Cohen on probabilistic termination in cryptographic protocols; Antigoni Polychandrou presented two-round two-party computation; Vassilis Zikas modeling the public ledger functionality; Ran Canetti talking about using the EasyCrypt software to aid in cryptographic proofs and verification.

The following is a summary of the workshop results:

1. The relationship between the UC and IITM model was intensively discussed, concluding that the models are very close and that it is possible to unify the two models. The unification also seamlessly includes JUC, GUC, and SUC.
2. The working group on SNARKs (recursive composition of succinct proofs) achieved initial modeling success and crystallization of what's actually challenging.
3. The working group on modeling $\mathcal{F}_{\mathsf{vrf}}$ and constricting it from $\mathcal{F}_{\mathsf{sig}}, \mathcal{F}_{\mathsf{ro}}$ figured out what the stumbling blocks were and what was fundamental.
4. The working group on $\mathcal{F}_{\mathsf{NIZK}}$ and proofs about signatures in Constructive Crypto started to model typical UC functionality in the Constructive Crypto framework and then inspected how they could be composed.
5. The working group on building threshold primitives from single primitive (e.g. threshold signatures from signatures, threshold encryption from encryption etc) came up with a candidate for a "thresholdizer" functionality, and found some subtleties in defining threshold behavior in the ideal world. The also found a candidate construction to test the validity of the definition.
6. The working group on setup assumptions analyzed the assumptions used for constructing composable protocols in terms of practicality and security provided.
7. The working group on delegating secret keys – discovered a simple interface that can be added to $\mathcal{F}_{\mathsf{sig}}$ to make it possible to delegate from one user to another well-defined user. Next steps are to investigate if it generalizes to other functionalities and to delegation that's based on knowledge transfer rather than explicit authorization of identity.

2 Table of Contents

3 Overview of Talks

3.1 Universally Composable Security: Philosophy, History, Status (Or: Exorcising the devil of detail: A never-ending task)

Ran Canetti (Tel Aviv University, IL)

Main reference Ran Canetti: "Universally Composable Security: A New Paradigm for Cryptographic Protocols", in Proc. of the 42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA, pp. 136–145, IEEE Computer Society, 2001.
URL https://doi.org/10.1109/SFCS.2001.959888

This talk contained an overview of the notion of Universally Composable (UC) security, it's history, variants, and current status.

3.2 Probabilistic Termination and Composability of Cryptographic Protocols

Ran Cohen (MIT – Cambridge, US)

Joint work of Sandro Coretti, Juan Garay and Vassilis Zikas
Main reference Ran Cohen, Sandro Coretti, Juan A. Garay, Vassilis Zikas: "Probabilistic Termination and Composability of Cryptographic Protocols", in Proc. of the Advances in Cryptology – CRYPTO 2016 – 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III, Lecture Notes in Computer Science, Vol. 9816, pp. 240–269, Springer, 2016.
URL https://doi.org/10.1007/978-3-662-53015-3_9
Main reference Ran Cohen, Sandro Coretti, Juan A. Garay, Vassilis Zikas: "Round-Preserving Parallel Composition of Probabilistic-Termination Cryptographic Protocols", in Proc. of the 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland, LIPIcs, Vol. 80, pp. 37:1–37:15, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2017.
URL http://dx.doi.org/10.4230/LIPIcs.ICALP.2017.37

Since the introduction of secure multiparty computation (MPC) in the '80s, it has been a common practice to consider a broadcast channel when designing MPC protocols. Well-known lower bounds show that deterministic broadcast protocols cannot run in a number of rounds sub-linear in the number of corrupted parties. The seminal works of Ben-Or and Rabin showed how to overcome these limitations via randomization, igniting the study of protocols over point-to-point channels with *probabilistic termination (PT)* and expected constant round complexity. However, absent a rigorous simulation-based definition, the suggested protocols are proven secure in a property-based manner, and therefore guarantee limited, if any, composability.

Composing PT protocols affects the round complexity of the resulting protocol in somewhat unexpected ways. For instance, the expected round complexity of the parallel composition of expected-constant-round protocols might be logarithmic in number of instances. Sequential composition of PT protocol also raises subtle issues since the parties fall out-of-sync and cannot start the protocol at the same round.

In this work, we put forth the first simulation-based treatment of MPC with probabilistic termination in the UC framework and prove a universal composition theorem for PT protocols. Our theorem allows one to compile a protocol using deterministic-termination hybrids into a protocol that uses expected-constant-round protocols for emulating these hybrids, preserving

the expected round complexity of the calling protocol. We showcase our definitions and compiler by providing the first composable protocols (with simulation-based security proofs) over point-to-point channels for the following primitives: (1) expected-constant-round perfect Byzantine agreement, (2) expected-constant-round perfect parallel broadcast, and (3) MPC with round complexity independent of the number of parties.

We proceed to analyze whether the techniques used for parallel composition of broadcast (which is a privacy-free functionality) can be generalized for composing in parallel arbitrary MPC protocols, and provide both feasibility and infeasibility results. We show an efficient protocol-compiler that outputs a protocol that realizes the parallel composition of m protocols, without increasing the expected round complexity; moreover, the compiler requires only black-box access to the underlying *protocols*. Using known techniques, a similar result cannot be achieved given only black-box access to the *functions* realized by the protocols.

3.3 The Wonderful World of Global Random Oracles

Manu Drijvers (Dfinity – Zürich, CH)

Joint work of Manu Drijvers, Jan Camenisch, Tommaso Gagliardoni, Anja Lehmann, Gregory Neven
Main reference Jan Camenisch, Manu Drijvers, Tommaso Gagliardoni, Anja Lehmann, Gregory Neven: "The Wonderful World of Global Random Oracles", in Proc. of the Advances in Cryptology – EUROCRYPT 2018 – 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 – May 3, 2018 Proceedings, Part I, Lecture Notes in Computer Science, Vol. 10820, pp. 280–312, Springer, 2018.
URL http://dx.doi.org/10.1007/978-3-319-78381-9_11

The random-oracle model by Bellare and Rogaway (CCS'93) is an indispensable tool for the security analysis of practical cryptographic protocols. However, the traditional random-oracle model fails to guarantee security when a protocol is composed with arbitrary protocols that use the same random oracle. Canetti, Jain, and Scafuro (CCS'14) put forth a global but non-programmable random oracle in the Generalized UC framework and showed that some basic cryptographic primitives with composable security can be efficiently realized in their model. Because their random-oracle functionality is non-programmable, there are many practical protocols that have no hope of being proved secure using it. In this paper, we study alternative definitions of a global random oracle and, perhaps surprisingly, show that these allow one to prove GUC-secure existing, very practical realizations of a number of essential cryptographic primitives including public-key encryption, non-committing encryption, commitments, Schnorr signatures, and hash-and-invert signatures. Some of our results hold generically for any suitable scheme proven secure in the traditional ROM, some hold for specific constructions only. Our results include many highly practical protocols, for example, the folklore commitment scheme H(m|r) where m is a message and r is the random opening information.

3.4 Game-based Composition for Key Exchange

Marc Fischlin (TU Darmstadt, DE)

Joint work of Marc Fischlin, Chris Brzuska, Bogdan Warinschi, Felix Günther

We discuss composition of key exchange protocols with arbitrary symmetric-key protocols, like a secure channel. We use game-based security notions for the primitives and the composed protocol. It turns out that the secure composition requires specific properties of the key exchange protocol such as forward security and public matching of partnered sessions.

3.5 A Bite of GNUC

Dennis Hofheinz (KIT – Karlsruher Institut für Technologie, DE)

Joint work of Dennis Hofheinz, Victor Shoup
Main reference Dennis Hofheinz, Victor Shoup: "GNUC: A New Universal Composability Framework", J. Cryptology, Vol. 28(3), pp. 423–508, 2015.
URL https://doi.org/10.1007/s00145-013-9160-y

GNUC (for "GNUC's not UC") is a framework that allows to model multi-party protocols and to analyze their security properties. This talk highlights two technical design choices made in the GNUC universal composability framework:

- the notion of "efficiency" for protocol machines, and
- the organization of protocols machines in a hierarchical manner.

In particular, we explain that the notion of an "efficient" protocol (or of an "efficient" attack or distinguisher) in fact contains a number of technical pitfalls, and how we avoid those pitfalls in GNUC.

The take-away message of this talk should be

- that there are lots of low-level decisions to be taken when designing a framework for multi-party protocols,
- that these low-level decisions may have high-level consequences (e.g., for the expressiveness or security properties of the resulting framework),
- but that these low-level decisions should not distract from the high-level proof ideas one usually tries to convey when using protocol frameworks.

3.6 State-Separating Proofs for Code-Based Game-Playing Proofs

Markulf Kohlweiss (University of Edinburgh, GB)

The security analysis of real-world protocols involves reduction steps that are conceptually simple but still have to account for many protocol complications found in standards and implementations. Taking inspiration from universal composability, abstract cryptography, process algebras, and type-based verification frameworks, we propose a method to simplify

large reductions, avoid mistakes in carrying them out, and obtain concise security statements. Our method decomposes monolithic games into collections of stateful packages representing collections of oracles that call one another using well-defined interfaces. Every component scheme yields a pair of a real and an ideal package. In security proofs, we then successively replace each real package with its ideal counterpart, treating the other packages as the reduction. We build this reduction by applying a number of algebraic operations on packages justified by their state separation. Our method handles reductions that emulate the game perfectly, and leaves more complex arguments to existing game-based proof techniques such as the code-based analysis suggested by Bellare and Rogaway. It also facilitates computer-aided proofs, inasmuch as the perfect reductions steps can be automatically discharged by proof assistants. We illustrate our method on two generic composition proofs: (1) a proof of self-composition using a hybrid argument; and (2) the composition of keying and keyed components. For concreteness, we apply them to the KEM-DEM proof of hybrid-encryption by Cramer and Shoup and to the composition of forward-secure game-based key exchange protocols with symmetric-key protocols.

3.7 Constructive cryptography and discrete system theory

Ueli Maurer (ETH Zürich, CH)

Joint work of Ueli Maurer, Renato Renner

The talk presented three parts.

1. An abstract resource theory, where resources are elements of a partially ordered set and constructions (of resources from resources), ajoining resources, relaxations, and several other concepts are captured by order-preserving functions (i.e., homomorphisms) satisfying certain axioms, for example (one-sided) commutativity of certain homomorphisms.
2. A theory of discrete probabilistic systems, where most systems discussed in cryptography can be understood as descriptions (in a particular language specific to the context and paper, for example a specific pseudo-code language) of such discrete systems. One can consider system specifications (i.e., sets of probabilistic systems) and define various specification relaxations, including an ϵ-relaxation and the game-relaxation of a system containing a game, where the relaxation is defined as the set of systems behaving like the given system but where nothing is specified if the game is won.
3. Constructive cryptography as an instantiation of a resource theory instantiated with discrete systems. Many examples were presented. A specific example that can probably not be captured by previous concepts in cryptography is authentication amplification, meaning that one constructs an n-bit authenticated channel from a k-bit authenticated channel (for n>k) by use of a hash function. By explaining this using game-relaxation one can make a tight construction statement without need for reductions or a distinguisher or adversary concept, and despite the fact that a single hash function is never collision-resistant.

Joint work with Renato Renner, and also based on joint work and discussions with many other people, in particular Björn Tackmann.

3.8 The IITM Model and its Instantiation iUC: Simple and Expressive Universal Composability

Ralf Küsters (Universität Stuttgart, DE)

Joint work of Ralf Küsters, Max Tuengerthal, Daniel Rausch, Jan Camenisch, Stephan Krenn
Main reference Ralf Küsters, Max Tuengerthal: "The IITM Model: a Simple and Expressive Model for Universal Composability", IACR Cryptology ePrint Archive, Vol. 2013, p. 25, 2013.
URL http://eprint.iacr.org/2013/025

The universal composability paradigm allows for the modular design and analysis of cryptographic protocols. It has been widely and successfully used in cryptography. However, devising a coherent yet simple and expressive model for universal composability is, as the history of such models shows, highly non-trivial.

In this tutorial, we present a coherent model for universal composability, called the IITM model ("Inexhaustible Interactive Turing Machine"). A main feature of the model is that it is stated without a priori fixing irrelevant details, such as a specific way of addressing of machines by session and party identifiers, a specific modeling of corruption, or a specific protocol hierarchy. In addition, we employ a very general notion of runtime. All reasonable protocols and ideal functionalities should be expressible based on this notion in a direct and natural way, and without tweaks, such as (artificial) padding of messages or (artificially) adding extra messages.

The expressivity of the IITM is also reflected in the fact that joint-state and global state composition theorems follow directly from the basic composition theorem of the IITM model. No model extensions or new theorems are necessary. The model also allows for modeling forms of shared state that are out of reach of other models. Moreover, protocols can be modeled where protocol participants are not forced to establish session IDs before the start of the protocol.

Finally, we briefly discuss an instantiation of the IITM model, called iUC, which helps protocol designers in their modeling and analysis tasks.

IITM: http://eprint.iacr.org/2013/025/ (joint work with Max Tuengerthal and Daniel Rausch) iUC: Will soon be made available on eprint (joint work with Jan Camenisch, Stephan Krenn, and Daniel Rausch)

3.9 Multi-protocol UC and its Use for Building Modular and Efficient Protocols

Björn Tackmann (IBM Research-Zurich, CH)

Main reference Jan Camenisch, Manu Drijvers, Björn Tackmann: "Multi-Protocol UC and its Use for Building Modular and Efficient Protocols", IACR Cryptology ePrint Archive, Vol. 2019, p. 65, 2019.
URL https://eprint.iacr.org/2019/065

We want to design and analyze protocols in a modular way by combining idealized components that we realize individually. While this is in principle possible using security frameworks that provide generic composition theorems, we notice that actually applying this methodology in practical protocols is far from trivial and, worse, is sometimes not even possible. As an example, we use a natural combination of zero-knowledge proofs with signature and

commitment schemes, where the goal to have a party prove in zero-knowledge that it knows a signature on a committed message, i.e., prove knowledge of a witness to a statement involving algorithms of the signature and commitment scheme. We notice that, unfortunately, the composition theorem of the widely used UC framework does allow one to modularly prove the security of this example protocol.

We then describe a new variant of the UC framework, multi-protocol UC, and show a composition theorem that generalizes the one from the standard framework. We use this new framework to provide a modular analysis of a practical protocol that follows the above structure and is based on discrete-logarithm-based primitives. Besides the individual security proofs of the protocol components, we also describe a new methodology for idealizing them as components that can then be composed.

4 Working groups

4.1 Extending the UC Signature Functionality with Unpredictability and Applications to Verifiable Random Functions

Markulf Kohlweiss (University of Edinburgh, GB)

Joint work of Markulf Kohlweiss, Dennis Hofheinz, Anna Lysyanskaya, Marc Fischlin, Manu Drijvers, Vassilis Zikas, Celine Chevalier

The universal composability (UC) framework guarantees that a protocol remains secure even when composed with arbitrary other protocols. A composition theorem allows UC secure protocols to be built iteratively by composing protocols that already have been proven UC secure. The proof is performed in a hybrid world with ideal functionalities that describe the guarantees of the component protocols. One such ideal functionality is the $\mathcal{F}_{\mathsf{Sig}}$ functionality that models the security of existentially unforgeable signatures (EUF-CMA).

EUF-CMA secure signatures are by necessity unpredictable. That is, no efficient adversary can compute a valid signature except with negligible probability. This property, however, is not modeled by any existing formulation of the $\mathcal{F}_{\mathsf{Sig}}$ functionality. Existing formulations typically ask the simulator to determine how signatures are computed, either by asking for the signature itself or for signing algorithms that generates the signature.

This state of affairs limits the usefulness of $\mathcal{F}_{\mathsf{Sig}}$ in settings where the unpredictability of signatures is essential, e.g. for the construction of verifiable random functions. A verifiable random function scheme fixes a family of functions $\{f_{\mathsf{sk}}\}$ and a way to sample a public key pair $(\mathsf{sk}, \mathsf{pk})$. The secret key sk is used to evaluate the function on a point x to obtain $y = f_{\mathsf{sk}}(x)$ and a proof π. The public key is used in a verification algorithm $\mathsf{Verify}(\mathsf{pk}, x, y, \pi)$. A secure VRF satisfies two properties:

1. without seeing π, y is indistinguishable from random for any efficient adversary,
2. an efficient adversary cannot compute x, y, π such that $\mathsf{Verify}(\mathsf{pk}, x, y, \pi) = 1$.

Ideally given a functionality for unpredictable signatures $\mathcal{F}_{\mathsf{Sig}}$ we would like to be able to construct a UC protocol emulating a UC functionality for $\mathcal{F}_{\mathsf{vrf}}$ in the random oracle model. This problem is interesting for several reasons:

1. VRFs are an important building block of proof-of-stake ledger (aka blockchain) protocols
2. It points out a current weakness of proof techniques in the UC model with regard to reasoning about rare events in UC, such as reasoning about signature forgeries, hash function collisions, or dishonest majorities.

A technique we explored was to adapt functionaries that commit "suicide" by making themselves trivially distinguishable from the real world into functionalities that become "super-useful". This would make the functionality useful when used in a hybrid proof, in which the suicide in case of the rare event would make it useless otherwise. These techniques appear promising, but did not suffice to construct VRF from unpredictable signatures. Positive (side) outcomes of the discussion were

- a better understanding of a recent $\mathcal{F}_{\mathsf{vrf}}$ functionality. The published variant was buggy and we managed to contact the authors to track down the issue. The paper was subsequently updated.
- identification of interesting research questions: 1. constructing $\mathcal{F}_{\mathsf{vrf}}$ with strong properties in the standard model, 2. idealization of computational entropy using actual entropy.
- an alternative technique for constructing $\mathcal{F}_{\mathsf{vrf}}$ using signatures that can be split into an entropy part and a proof part.

4.2 Relating the UC and IITM Models

Ralf Küsters (Universität Stuttgart, DE), Ran Canetti (Tel Aviv University, IL), Celine Chevalier (University Paris II, FR), Daniel Rausch (Universität Stuttgart, DE), and Björn Tackmann (IBM Research-Zurich, CH)

Two of the most prominent models for analyzing protocols in a universally composable manner are the UC model by Canetti [1] and the IITM model by Kuesters et al. [2] At a very high level, both models follow a similar idea and provide theorems for the secure composition of protocols. However, at a technical level, both models are (sometimes drastically) different. This includes the machine model, connections between machines, addressing of machines, runtime definitions, as well as requirements of the composition theorems. The goal of this working group was to relate both models on this technical level and find a common ground.

We started our discussion by trying to find a common set of rules for protocols which, if they are met, imply that the protocol can be analyzed, proven secure, and composed in both the UC and the IITM model. This meant we had to find the limits of each model, and see whether there are certain types of protocols or features that can be expressed only in one model. Surprisingly, during this discussion, we found out that both models are actually closer related than expected. For (almost) every technical aspect and way of modeling a protocol in one model, we found a way to achieve the same in the other model. We gained many interesting insights in how the same problems are solved in different yet equivalent ways by each model.

We decided that we want to collaborate on a paper as a followup to this working group. The paper shall explore the insights from this working group in more detail and show that the UC model and the IITM model are actually equivalent in terms of expressivity (up to runtime). This has many interesting consequences. For example, it would follow that, as already shown in the IITM model, the composition theorem in the UC model also implies theorems for joint-state and global state composition as special cases.

References

1 Ran Canetti. *Universally Composable Security: A New Paradigm for Cryptographic Protocols*. FOCS, 2001. See https://eprint.iacr.org/2000/067 for the most recent version.
2 Ralf Küsters. *Simulation-Based Security with Inexhaustible Interactive Turing Machines*. CSFW, 2006. See https://eprint.iacr.org/2013/025 for the most recent version.

1. **Key Generation:** Upon getting (KEYGEN, Signer, sid, f) from a party Signer, where f is a polynomial-time deterministic algorithm that, on input sid and a party identity, returns True or False ...
   a. If this is not the first KEYGEN command, ignore this command. Otherwise, continue.
   b. If $f(sid, \text{Signer}) \neq \text{True}$, ignore this command. Otherwise, continue.
   c. Initialize a variable $C = \text{False}$, meant to keep track of whether any signer is corrupt. If $f(sid, \text{P})$ holds for any corrupt party P, set $C = \text{True}$.
   d. Initialize an empty list $\mathbf{W}$ of signed messages.
   e. Send (KEYGEN, Signer, sid) to $\mathcal{S}$.
   f. Get (ALGORITHMS, sid, Sign, Verify) from $\mathcal{S}$, where Sign is a polynomial-time algorithm and Verify is a polynomial-time *deterministic* algorithm.
   g. Send (ALGORITHMS, sid, Verify) to Signer.
2. **Signature Generation:** Upon getting (SIGN, Signer, sid, m) from a party Signer ...
   a. Verify that $f(sid, \text{Signer}) = \text{True}$. If not, ignore this command. Otherwise, continue.
   b. Let $\sigma = \text{Sign}(m)$.
   c. Verify that $\text{Verify}(m, \sigma) = 1$. If not, send $\perp$ to Signer and halt, since this violates correctness. Otherwise, continue.
   d. Output (SIGNATURE, sid, m, σ) to Signer.
   e. Record m in $\mathbf{W}$.
3. **Signature Verification:** Upon getting (VERIFY, Verifier, sid, m, σ, Verify$'$) from a party Verifier ...
   a. If Verify$'$ = Verify, $C = \text{False}$, $\text{Verify}(m, \sigma) = 1$ and $m \notin \mathbf{W}$, send $\perp$ to signer and halt, since this violates soundness. Otherwise, continue.
   b. $\phi = \text{Verify}'(m, \sigma)$.
   c. Output (VERIFIED, sid, m, σ, Verify$'$, v) to Verifier.
4. **Key Delegation:** Upon getting (DELEGATE, Signer, sid, f') from a party Signer, where f' is a polynomial-time deterministic algorithm that, on input sid and a party identity, returns True or False ...
   a. If $f(sid, \text{Signer}) = \text{False}$, ignore this command. Otherwise, continue.
   b. Set $f(\cdot, \cdot) = f(\cdot, \cdot) \vee f'(\cdot, \cdot)$ (that is, set the function f to be the disjunction of the old f and the f' just provided).

Additionally, upon every corruption of party P, the functionality sets $C = C \vee f(sid, \text{P})$.

Figure 1 Ideal Functionality for Digital Signatures With Key Delegation.

4.3 Extending $\mathcal{F}_{\text{sig}}$ to Allow for Key Delegation

Anna Lysyanskaya (Brown University – Providence, US), Celine Chevalier (University Paris II, FR), and Sophia Yakoubov (MIT Lincoln Laboratory – Lexington, US)

In Figure 1, we describe the signature functionality $\mathcal{F}_{\text{sig}}$, with the modification that a signer can delegate their signing ability (e.g. by sharing their secret key) to others. This delegation module can be similarly grafted onto other functionalities, e.g. encryption.

Note that currently, this functionality only describes identity-based delegation (that is, a signer must specify a function on users' identities which determines whether they have the power to sign). Ultimately, it would be desirable to make this function more generic, enabling witness-based delegation.

In magenta, we denote inputs to the ideal functionality that a party (controlled by the environment / adversary) cannot falsify. In blue, we denote places where this functionality differs from the digital signature ideal functionality of Ran Canetti's Universal Composability paper (2005 version).

Future work will include:

1. Grafting this module onto other functionalities.
2. Exploring witness-based delegation.

4.4 Modeling in the Constructive Cryptogaphy Model

Ueli Maurer (ETH Zürich, CH), Jan Camenisch (Dfinity Foundation – Zug, CH), Celine Chevalier (University Paris II, FR), Jens Groth (London, GB), and Daniel Rausch (Universität Stuttgart, DE)

The Constructive Cryptography (CC) model by Maurer [1] is a framework for performing modular proofs. Compared to the widely used Universal Composability (UC) model [2], the CC model takes a much more abstract view and defines only a minimal number of details that are sufficient to obtain modularity. In particular, the CC model does not specify a specific computational model or a runtime notion. Thus it is not directly obvious how ideal functionalities and security proofs as used in the UC model can be modeled in the CC model.

Our working group had two goals. Firstly, we wanted to show that (and how) one can model functionalities and perform security proofs from the UC model also in the CC model, thus verifying the expressiveness of the CC model and making the CC model more accessible to cryptographers that are used to the UC model. Secondly, we wanted to find ways to use the more abstract view of the CC framework for simplifying or generalizing functionalities as well as making security proofs easier to carry out and verify. We used a recent protocol by Jan Camenisch et al. [3] defined in the UC model as a case study. This protocol combines ideal functionalities for zero knowledge (ZK), commitments, and digital signatures in a modular fashion to prove knowledge of a signature on a message in a commitment.

We started by defining an ideal functionality for digital signatures in the CC model. A core insight of this process was that the realization proof of the signature functionality becomes much simpler in the CC model. This is because one can postpone the final reduction to a game based security assumption until after the whole system has been designed, whereas this reduction must be perfomed as part of the realization proof in the UC model. In the next step, we defined an ideal functionality for non-interactive ZK proofs of knowledge in the CC model and sketched a realization, including a corruption model. While this first version of the ZK functionality does not appear to be much simpler than in the UC model, it illustrates how cryptographers can express complex protocols from the UC model in the more abstract CC model.

The results of our working group indicate that we can indeed bridge the gap between the UC and CC model. Furthermore, our experience with the signature functionality makes us confident that we can find additional improvements to simplicity and generality of ideal functionalities and security proofs. Thus, our team decided to follow up in this working group with a paper based on the insights that we gained.

References

1 Ueli Maurer. *Constructive Cryptography – A New Paradigm for Security Definitions and Proofs*. Theory of Security and Applications – Joint Workshop, TOSCA, 2011.
2 Ran Canetti. *Universally Composable Security: A New Paradigm for Cryptographic Protocols*. FOCS, 2001. See https://eprint.iacr.org/2000/067 for the most recent version.
3 Jan Camenisch, Manu Drijvers, and Björn Tackmann. *Multi-Protocol UC and its Use for Building Modular and Efficient Protocols*. IACR Cryptology ePrint Archive, 2019. Available at https://eprint.iacr.org/2019/065 .

4.5 Modular Realization of Threshold Primitives

Alessandra Scafuro (North Carolina State University – Raleigh, US), Stephan Krenn (AIT – Austrian Institute of Technology – Wien, AT), Ralf Küsters (Universität Stuttgart, DE), Daniel Slamanig (AIT – Austrian Institute of Technology – Wien, AT), and Ivan Visconti (University of Salerno, IT)

Research Question. In a (t, n) threshold primitive we want that a cryptographic operation is performed if a subset of t parties agree (and has the credential) to perform a certain operation on a input x. For example, threshold signature require that at least t parties agree on signing a message, threshold encryption requires that at least t parties participate to decrypt a ciphertext c.

The motivation behind threshold schemes is typically robustness, i.e., we want to make sure that if one of more machines fail, the security of a certain operation is still guaranteed. In some settings (such as ring signatures) we additionally want privacy, and we require that the identity of the identity of the t participants is not leaked. We are not considering this setting here.

Definitions of threshold encryption and (ring) signatures exist, in a game-based setting, and only very recently a definition of threshold (ring) signature was also provided as ideal functionality.

Our goal is generalize the problem of computing threshold X having access to primitive X. We want to provide a "wrapper" ideal functionality $\mathcal{F}_{\mathsf{TR}}$ that captures the threshold constraint wrt an operation, making calls to the underlying functionality $\mathcal{F}_{\mathsf{OP}}$ that performs a single operation.

Progress made over the meeting. We have outlined an ideal threshold signature functionality that has access to an instance of signature functionality. During this process we identified potential issues in current definitions of threshold ring signatures and identified edge cases that do not seem to be explicitly captured by current definitions. We also discussed a candidate, proof of concept realization of our new ideal functionality, based on signatures, Merkle tree and succinct non-interactive zero-knowledge argument.

Next step. Our next step is to formally validate our high-level designs.

4.6 Setup Assumptions for Universal Composability

Alessandra Scafuro (North Carolina State University – Raleigh, US), Manu Drijvers (Dfinity – Zürich, CH), Stephan Krenn (AIT – Austrian Institute of Technology – Wien, AT), Arpita Patra (Indian Institute of Science – Bangalore, IN), Antigoni Polychroniadou (Cornell Tech – New York, US), and Daniel Slamanig (AIT – Austrian Institute of Technology – Wien, AT)

Research Question. Setup assumptions are assumptions we make about the world. One example of setup is the assumption that the world has trustworthy parties that can honestly perform certain operations, or that users are able to manufacture physical tokens that behave like black-boxes and completely obfuscate a software.

Setup assumptions are necessary for proving security of protocol in universally composable (UC) sense, thus they are extensively used in any UC-secure protocol.

Naturally, some assumptions might be more practical/realistic than others, at the expenses of requiring more trust on third parties and hence more assumptions about the (honest) behaviour we expect in the real world.

In particular, assuming setups that have local scopes (i.e., they are visible only to the parties participating in the protocol) is quite unrealistic, as in the real world one would assume that the same setup (e.g., the same public key) is re-used in many executions. On the other hands, using global setups, while seemingly a more realistic approach, it introduces global trapdoors that could determine global failure in case the trusted party is compromised.

The literature seem to lack of a thorough analysis and comparison of the existent setup assumptions, especially, in light of global composition requirements.

Progress made over the meeting. During the meeting we discussed a few setup assumptions used in the literature and compared them wrt two parameters: practicality, that is, how easily they could be realized, and trust.

- Common Reference String (CRS model). This setup assumes that for each protocol execution, the participating parties are able to obtain a local common reference string (sampled from a distribution prescribed by the protocol) from a trusted source. The string is fresh and local to the protocol. In the literature, it is typically assumed that the common reference string is sampled by a trusted party. Alternatively, the string can be computed by a set of parties, via a multiparty computation protocol.
  Variation of the CRS model exist, however such models only relax the requirement of trusting a single source.
  Note that the requirement of the CRS being local and fresh is necessary only when proving UC-security. If weaker composability guarantees are required (e.g, simply proving that a protocol is a non-interactive zero-knowledge protocol) then the same CRS could be reused and have a global scope.
  - Practicality: mostly impractical. Since the CRS must be local and freshly sampled upon each protocol execution (or a predetermined set of executions), parties will need continue access to a trusted source of CRS. If the CRS were computed via a MPC protocol, this process would be even more cumbersome,
  - Trust. If maliciously computed the CRS contains trapdoors. Corruption of a CRS source determines loss of security. However, since each protocol is executed with fresh CRS, corruption of a one CRS source has only a local impact and it affects only one execution (assuming that different protocols might have access to different CRS sources).
- Augmented Common Reference String. In this setup, there is a trusted party that publishes a short reference string (i.e., of size independent on the number of parties), and possesses a master secret key that can be used to derive per-party secret keys. Only the reference string is required for executing a protocol. However, corrupted parties can ask the trusted party to provide them with their secret key. In the security proof this secret key is used by the simulator.
- Key Registration Model. In this setup, there is a trusted party that chooses a private and public key pair for each registered party and lets all parties know the value of the public key only. Corrupted parties however can retrieve their own secret key by asking the trusted party.

  - Practicality. Somewhat practical. The key registration authority could be implemented by a third party (post office is often suggested). A potential practical drawback is that the KR authority must be always on-line for registration and key retrieval. This means that a MPC emulation of the KR authority might be problematic since all parties must be on-line. On the other hand, perhaps an MPC implementation would discourage people to cheat since so many parties will then learn that this person is asking for their key.
  - Trust. The key registration authority knows all trapdoors of the system that are used in all executions. As such it represents a single point of global failure.
  - Similar arguments hold for the Augmented Common Reference String model.
- Global Random Oracle. This is a proof methodology rather than a setup assumption since there is no real-world implementation of the random oracle. The global random oracle assumption idealizes properties of hash functions. It assumes that all parties have oracle access to a public random function. In the proof this is translated with the ability of the simulator to extract the queries made to the oracle. There exist two formulations of the Random Oracle Model, one assumes that random oracles are local to the protocol execution. This is somewhat less realistic since in reality the same hash function is used across all execution. The global random oracle model instead assumes that there is a global oracle that all protocol executions share. This limits the power of the simulator in the proof.
  - Practicality. When the RO is implemented with an hash function, it is very practical.
  - Trust/Security. Concrete hash functions do not behave like a random oracle, therefore when RO is replaced with hash function it does not provide provable security guarantees.
- Hardware Assumptions. In this setting parties are assumed to have the ability to manufacture hardware tokens that embed arbitrary functions and behave like a black-box when in the hands of an adversary.
  - Practicality. The exchange and the creation of general purpose hardware token is highly impractical. Some constructions in literature, are based on very specific tokens, such as signature cards, that are easy to obtain from trusted authorities. Such construction are significantly more practical, but their security completely relies on the trust in the authority manufacturing the tokens, somewhat collapsing to the Key Registration Model.
  - Trust. The trust here is posed into the hardware technology as well as the trust into manufacturers.

Concluding thoughts. Known global setups guarantee the highest level of composability, but they seem to provide the most fragile security guarantees since the security of all protocols in the system rely on the security of global trapdoors. If such trapdoors fall into the hands of adversary, the security of every protocol is compromised.

On the contrary, protocols achieving weaker composition guarantees, are less fragile in that they are based on local trapdoors or no trapdoors (e.g., in stand-alone security) This seems to suggest that the highest composability guarantee comes at the price of highest reliance on real world good behaviour of third parties. The global random oracle does not suffer of this problem, since the work that is done by the simulator in the ideal world simply cannot be emulated in the real world by any adversary since random oracles do not exist.

Next steps. The next step would be to write a manuscript analysing more thoroughly possible implementations of each setup assumption, the concrete trust required and the possible "fall-back" security guarantees in presence of compromise.

Participants

- Jan Camenisch
Dfinity Foundation – Zug, CH
- Ran Canetti
Tel Aviv University, IL
- Celine Chevalier
University Paris II, FR
- Ran Cohen
MIT – Cambridge, US
- Manu Drijvers
Dfinity – Zürich, CH
- Marc Fischlin
TU Darmstadt, DE
- Dov Gordon
George Mason University – Fairfax, US
- Jens Groth
London, GB
- Timo Hanke
Dfinity Foundation – Zug, CH
- Dennis Hofheinz
KIT – Karlsruher Institut für Technologie, DE
- Markulf Kohlweiss
University of Edinburgh, GB
- Stephan Krenn
AIT – Austrian Institute of Technology – Wien, AT
- Ralf Küsters
Universität Stuttgart, DE
- Anna Lysyanskaya
Brown University – Providence, US
- Mary Maller
University College London, GB
- Ueli Maurer
ETH Zürich, CH
- Arpita Patra
Indian Institute of Science – Bangalore, IN
- Antigoni Polychroniadou
Cornell Tech – New York, US
- Daniel Rausch
Universität Stuttgart, DE
- Alessandra Scafuro
North Carolina State University – Raleigh, US
- Daniel Slamanig
AIT – Austrian Institute of Technology – Wien, AT
- Björn Tackmann
IBM Research-Zurich, CH
- Muthuramakrishnan Venkitasubramaniam
University of Rochester, US
- Ivan Visconti
University of Salerno, IT
- Sophia Yakoubov
MIT Lincoln Laboratory – Lexington, US
- Vassilis Zikas
University of Edinburgh, GB