# The Landscape of Security from Physical Assumptions[1]

Alexander Koch
Competence Center for Applied Security Technology (KASTEL)
Karlsruhe Institute of Technology (KIT), Germany
Email: alexander.koch@kit.edu

***Abstract*—We survey several security assumptions based on physical principles as opposed to more common complexity-theoretic assumptions. This survey focuses on obtaining security guarantees via i) *idealized hardware* and ii) *physical objects*, and specifies how these assumptions have been used for devising cryptographic protocols, such as protocols for secure multi-party computation. Note that due to these assumptions, the protocols are often conceptually simpler, the security is independent of the computational power of an attacker, and the functioning and security is more transparent to humans.**

## I. Introduction

Security assumptions that are based on *physical principles* exhibit quite some advantages when compared to complexity-theoretic assumptions, namely the protocols being conceptually simpler, the security even holding independently of the attacker's computational resources, and the functioning and security being more transparent to humans. Examples of such assumptions are physically isolated or incorruptible hardware components, write-only devices for logging, or scratch-off cards as common in letters for personal PINs. Also, the non-cloneability of quantum states that follows from the principles of quantum theory, is a physical security assumption that is, e.g., used to realize non-cloneable "quantum money".

Using physical hardware or objects allows to circumvent impossibility results in secure computations. For example, tamper-proof hardware tokens due to [41] suffice to construct protocols for general "secure computation" with very strong (composable) security guarantees without the need of a trusted authority to set up, e.g., a public-key infrastructure – something that was thought impossible before, due to [14, 15]. Here, a *secure computation* involves multiple parties, which would like to jointly evaluate a function (such as "who of them is richest"), without giving away anything about the other parties' input values (not obvious from the output and own input).

Moreover, protocols employing physical assumptions may offer qualitatively stronger security guarantees, or other security aspects, such as fairness (informally: if a protocol yields an output, all parties learn it), using different trust models, deniability and non-coercion. A particular advantage that only-digital protocols cannot offer, is to provide a *bridge to reality*. Examples of this are given in [33, 31], where the authors provide a protocol for proving (in zero-knowledge) that a nuclear warhead that is to be disarmed due to an international treaty conforms to a certain prescribed template, without giving away anything about its internal design.

Also, due to our familiarity with the physical world, many protocols that make use of day-to-day objects, such as envelopes or ballots used in cryptographic voting schemes, are often much easier to understand, or are just more transparent than computer hardware executing some program. This might be crucial for a protocol to be even considered for real-world use. An example is the German Constitutional Court's demand that "[w]hen electronic voting machines are deployed, it must be possible for the citizen to check the essential steps in the election act and in the ascertainment of the results reliably and without special expert knowledge." (Judgment of the Second Senate of 03 March 2009). Finally, physical tools such as the wooden boxes, can be used fruitfully for didactics, or to do secure computations without any computer.

The main focus of this paper is to give a (partial) overview over of how physical security assumptions have been used in the area of cryptography and provable security. We will categorize such physical cryptography broadly into three domains (by its main object), namely to obtain security guarantees with i) idealized *hardware* covered in Section II, ii) idealized, easily-manipulateable physical *objects* in Section III and iii) idealized physical *processes* or *properties*. Due to space constraints and its prominence in the literature, we omit the third category in the following, and will *not* survey, e.g. *Wyner's well-known wiretap channels* [77, 20, 49] or quantum cryptography.

## II. Security from Idealized Hardware

Many researchers have suggested using hardware as a facilitator for cryptographic protocols or as a trust anchor. Let us go through the most important hardware assumptions used for cryptographic protocols. Throughout this section, we focus on results that attain security in the strong Universal Composability (UC) framework [13].

### A. Physically Uncloneable Functions

Introduced as "Physical One-Way Functions" by Pappu et al. [69], so-called Physically Uncloneable Functions (PUFs) are simple, stateless hardware modules which serve as a

[1] This invited paper is mainly based on a survey that appeared in [43, Part I]. This work was supported by KASTEL Security Research Labs. This is the author's accepted version (with hyperlinks and more verbose bibliography entries) of an article that will appear at the IEEE Information Theory Workshop (ITW 2021), copyrighted by IEEE. (This will be updated as soon as the article is published, with a link to the publisher's version.)

(noisy) evaluation of a function with high min-entropy output. The used manufacturing process unavoidably and purposefully includes imprecision and slight variations in the chip, leading to a response behavior that is hard to clone. Although the concrete security requirements depend on the cryptographic protocol or application, many definitions include one-wayness and unforgeability of the output. Armknecht et al. [7] give a good survey on these notions and integrates them into one consistent and unified framework. For a survey on technical implementations of PUFs, see [42].

When it comes to the achievable level of security when using PUFs as hardware given to the players, [12] gave the first construction of secure multiparty computation (MPC) in the UC framework that is even unconditionally secure. However, as noted by [68], their construction assumes that the PUFs in use are fully trusted. An adversary that is able to create a hardware chip which looks and behaves like a PUF can easily break the security of their scheme. These malicious PUFs come in two flavors: they can be either *stateful* (i.e. they may log all input-output values; however, note that they neither can communicate back to the adversary, nor does he get access to it after the protocol.) or *stateless*, with the latter being plausibly much more easy to craft in a way that it is as simple (and indistinguishable from the outside) as the PUF that is to be imitated. In the setting of stateful malicious PUFs, [68] present a protocol for secure computation that achieves computational UC-security. *Unconditional* UC-security in the (stateful) malicious PUF-setting has only be attained for commitment protocols, cf. [22]. Indeed, [21] show that unconditional general computation and oblivious transfer is impossible, even in the stand-alone setting. On the positive side, they construct an unconditional and efficient UC-secure protocol for general computation, if one restricts the adversary to only issue stateless PUFs. Recently, [48] achieved UC-secure commitments from fully malicious PUFs in the long-term setting. Long-term security guarantees that the protocol remains secure even if the adversary gets unlimited computational power after the protocol has ended, which is an important security feature in the light of possible future technological advances threatening currently-used cryptography.

### B. Signature Cards

Hofheinz, Müller-Quade, Unruh [36] proposed using trusted signature cards as an alternative setup assumption for UC-secure computation. These are modeled as an ideal hardware functionality which, upon receiving a message as an input by its holder, outputs a signature to it. Using these, one can also obtain UC-security in the long-term setting, as shown in [62]. As further research has shown, one can also handle *untrusted* signature cards, e.g., by restricting to signature schemes which have a unique (non-randomized) signature for each message, and by taking special care for the case when the card aborts dependent on the message to be signed, thereby leaking to the outside that a message from a certain set was to be signed. See [50] for reference, which additionally achieves reusability of the signature card in multiple protocols without impacting security, as in the Global UC (GUC) framework introduced in [16]. ([36] also offer reusability of signature cards, but their signature cards are not modeled in the GUC framework.)

### C. Tamper-Proof Hardware Tokens

Besides hardware modules which compute a random function (such as PUFs) or implement a signing functionality, one can also assume hardware, such as smart or SIM cards or USB authentication tokens, which can execute arbitrary code. These have been proposed by Katz [41] as a setup assumption for UC-secure computation. In contrast to previous setup assumptions, such as a common reference string or a public-key infrastructure, this does not need a trusted central party responsible for establishing the setup assumption. Here, the security relies on two assumptions: i) the code of the token and any internal secrets are completely opaque to the holder, i.e., the token is *tamper-proof* and cannot be brought to reveal its secrets by any engineering measure, and ii) the token cannot communicate with the outside (except possibly through regular protocol messages), in particular, it cannot send any security-relevant information back to its original creator.

Tamper-proof hardware mainly comes in two flavors, dependent on how lightweight the used hardware is supposed to be: i) *stateful* tokens which can reliably store data and keep a non-trivial internal state, ii) *stateless* or (stateful but) *resettable* tokens (such as a smart-card reliant on an external power source) which should still work if an adversary repeatedly cuts off power to the token and thereby resetting its state. Differently from PUFs, malicious tokens are always assumed to be able to keep a state, aiding the purposes of the attack.

Obviously, assuming stateful tokens is a much stronger assumption, leading to strong feasibility results. For example, Goyal et al. [34] show that even non-interactive and unconditionally UC-secure two-party computation is possible with simple stateful tamper-proof hardware tokens against malicious adversaries. Here, non-interactive two-party computation starts with a single charge of tokens being sent from sender to receiver, followed by a computation phase without any communication. In terms of efficiency, [24, 23] were able to reduce the number of necessary stateful tokens to the provable minimum of one and two tokens for interactive and non-interactive two-party computation, respectively, while retaining unconditional UC security in both cases. (Note that using a single token is especially beneficial, as one does not need to take into account the threat of multiple malicious tokens covertly communicating at the receiver's place.)

In the case of stateless or resettable tokens, one is restricted to realizing functionalities that are compatible with being reset at any time in the protocol, called "resettable functionality" in the following. First of all note that (as pointed out in [34]) stateless tokens by themselves cannot achieve unconditional security, as an unbounded adversary can completely learn the behavior of the token (unless one restricts the number of resets, as in [22, 26]). In this setting, due to [27, 25], one can achieve arbitrary resettable two-party functionalities with UC-security using only a single token and the existence of one-way functions. More recently, [35] constructed constant-round adaptively secure protocols which allow all parties to be corrupted. As a variant of the tamper-proof hardware model, [30] suggested to use disposable circuits which can be completely destroyed after the computation, to realize unconditional UC-security computation (with input-dependent abort). This suggestion is especially useful in the context of physical computations [31], such as determining or proving a match of certain genes, where one needs an "information barrier" after the protocol. A good survey on the use of tamper-proof hardware tokens can be found in [67].

### D. *Trusted Hardware with Constrained Functionality*

The above assumptions on secure hardware have first been introduced as a trusted functionality, evoking a search for generalizations to the respective untrusted hardware assumptions. This is understandable, as, e.g., a trusted PUF is a strong assumption. However, if the functionality to be implemented is very simple and can be formally verified as a fixed-function logic circuit, and it might also be plausible to build the hardware yourself or to obtain them in usual electronic stores – making targeted attacks much more difficult – then, such trust assumptions become more plausible. Moreover, often these modules do not carry any secrets themselves, making the tamper-proofness assumptions as above less important. Hence, it is worthwhile to consider such *trusted* hardware modules, in particular if they lead to strong security results. As an example, [3] uses a very simple secure equality check hardware module, to ensure the correct, UC-secure functioning of a parallel firewall setup, protecting against a malicious firewall. Similarly, Achenbach et al. [6] use the assumption of a trusted TAN generator or optical code reader in order to UC-realize a trusted money withdrawal functionality. Interpreted more generally, this is a solution to secure human–computer interface via an untrusted platform, as described in [9].

An example of achieving qualitatively stronger security by using trusted hardware such as data-diodes, air-gap switches and "output interface modules" is the Fortified MPC framework [11]. Here, using the isolation assumptions that come with these channel types, one can define what it means to be (path-)connected to "the outside" at a point in time. The attacker model is then extended from static (physical) corruptions that are performed before parties are invoked, to *remote hacks*, i.e., attacks that are possible during the protocol run, but only when connected to the outside. They give a construction of protocols for MPC with a security notion that is qualitatively stronger than commonly aimed-for adaptive security in that the inputs and outputs of all parties are completely protected (w.r.t. confidentiality and integrity) against remote hacks, unless they happen before the party received its input, or the attacker gains control over all parties.

Secure Oblivious Bingo Voting [4, 5], where the voting machine does not even learn the vote cast by the user, has been realized with a special trusted physical module. Also in the context of voting, Moran, Naor, Segev [61] consider write-once memory as a trusted hardware assumptions to securely store votes even in adversarial environments. These or even simpler write-only devices, such as printer can also be used to achieve secure logging. Next, we discuss a special case of trusted hardware that deserves separate mentioning, due to its refined modeling and more complex functionality.

### E. *Secure Processors*

Recently, processors that allow for *attested execution* in a sandboxed environment, a so-called *enclave*, such as Intel's SGX technology, have been formally modeled by Pass, Shi, Tramèr [70] in the GUC framework. An attested execution of a program $P$ on inputs $i$ outputs not only the result, but also a signature on $P$ and the output, certifying that the program has been correctly executed on this processor, resulting in the respective output. As the execution happens in an enclave, it is protected against tampering and other forms of modification and/or leakage, dependent on how weak the security assumptions are. For example, [70, 76] describe a variant where all internal secrets of the computation are allowed to leak (but not the signing key for the attestation), so-called transparent enclaves. Using these, they were able to construct UC-secure commitments and zero-knowledge proofs, secure in the GUC framework, hence allowing for reusability of the processors after the protocol. Other interesting applications of secure processors are, e.g., given in [32] – implementing functional encryption using Intel SGX enclaves – and [64], which implements obfuscation for RAM programs in a very strong (virtual-black-box) sense. General program obfuscation of this strength has been shown to be impossible in software.

## III. Security from Physical Objects

In contrast to secure hardware, physical objects do not carry any internal logic or programming but are specifically-crafted or day-to-day things that can be used for cryptographic protocols. These are sometimes used as inspiration or analogy, such as when using boxes and locks to explain key exchange protocols, but can also be thought to practically achieve a cryptographic computation, e.g., without a computer, and might thereby offer more tangible and transparent security than the digital counterparts. The most prominent example is in cryptographic voting schemes, where physical ballots or envelopes are modeled with a concrete security goal in mind.

Moreover, one can also use these objects more broadly for recreational cryptography, or didactics, e.g., when illustrating MPC or zero-knowledge proofs to university or high-school students. Finally, some of these are suitable for the theoretical interest of studying unconventional computational models.

### A. *Physical Envelopes and Ballots*

Exploiting physical properties of voting machines or ballots is common in the cryptographic voting community to achieve seemingly contradictory properties such as receipt-freeness (informally, one cannot show others a receipt from which they can derive information about how one has voted, an important property for non-coercibility), and (public) verifiability of the fact that one's vote has been counted. For example, PunchScan

[71] employs ballots which consist of two sheets of paper, fixed together but separable. The sheet on top has holes through which a code for the available options is visible. When voting, an ink punching device large enough to mark both sheets of paper when applied to a holes, is used. Note that the two sheets of the ballot, taken together, determine which hole is to be punched for the voting choice, but when separated, each single sheet does not give away anything about the vote. The protocol is analyzed more formally in [60, Sect. 2.4].

Another voting scheme that uses physical properties is Scantegrity [18, 17], and the schemes of Moran, Naor [59, 58]. One ingredient common to many voting schemes are *tamper-evident seals*, such as envelopes, locked but breakable boxes or scratch-off cards, cf. [57]. Besides voting, they allow to execute many cryptographic primitives, such as oblivious transfer and bit commitment. They distinguish four types, dependent on whether the seals are all indistinguishable, and on whether honest players have the ability to open a locked seal (such as a closed envelope) and achieve distinct feasibility results for the different seal types. In [38], the authors introduce the notion of a "rational attacker" which tries to maximize its utility, and then employ a ballot-box and envelopes to implement unconditionally UC-secure MPC in this rational attacker model (without an honest majority). In [37, 39], they extend their ballot-box method to form the notion of "Verifiably Secure Devices" or "Transparent Computation", which essentially describes a transparent procedure implemented by a human or device employing such a ballot box functionality, to compute the desired functionality.

### B. Cryptography with Playing Cards

Secure multiparty computation can be done with a *deck of physical cards*, as first shown in [10, 19, 65]. In this area of *card-based cryptography*, one designs tangible protocols using a deck of cards with information-theoretic privacy features. The focus has been on designing card-minimal and/or simple protocols for AND and bit-copy protocols [56, 2, 72], as well as lower bounds on the number of cards [47, 40, 44], with the focus being on cards with two symbols, hearts and clubs. Recently, there has been new protocols for the case of a standard deck, where all cards are indistinguishable, cf. [66, 52, 45]. Finally, card-based private function evaluation is given in [46], and the computational power of card-based cryptgraphic protocols is more generally analysed in [28]. A formal model for card-based cryptography was introduced in [55], and an overview on the topic is given in [43].

Famously, Schneier [73] invented a symmetric cipher, called Solitaire, that is executed with cards. While biased and hence not really secure, it is advertised with non-digital features such as plausible deniability (everyone may carry a deck of cards) and fast "secure erasure" (shuffling destroys the key) in case of a physical search. Toponce [75] provides an overview over several alternative card ciphers that have been proposed and which exploit the fact that generating randomness is simple when shuffling cards, to be easy to execute.

### C. Cryptography with Other Objects

A nice introduction to cryptography for the task of securely comparing private values, utilizing different physical objects, which besides cards and envelopes also includes a discussion on using cups and Airline reservation hotlines, is given in [29].

Balogh et al. [8] establish MPC for arbitrary functions using a (large) PEZ dispenser ideal functionality with PEZ candies of two colors, where same-color candies are indistinguishable. Here, each player privately dispenses a number of candies dependent on the input and only this player learns the order and color of its candies. After the protocol, the color of the candy that would be dispensed if pressed once more, encodes the output bit. While certainly evoking amusements, their research was motivated by the question of how far can one can go with a *deterministic* ideal functionality, to which one cannot trivially offload all the computation. [1] improved upon this work w.r.t. several parameters for symmetric functions.

Naor, Shamir [63] invented "visual secret sharing", allowing to create physical transparent slides, each with random (but correlated) dot patterns, such that when both are placed on top of each other, a black-and-white image appears. (There have been many extensions, also to allow colors and detectability if someone uses a malicious transparency.)

Researchers have been creative in employing other objects for MPC, e.g., [54] describes how to compute any function with up to four variables using a 15 Puzzle, and [53] use a dial lock to compute a specific class of functions securely. Moreover, [74] proposes simple protocols using polarized plates. Finally, using marbles and the assumption that when placing them into a bag, they become shuffled, gives rise (or "rice") to nice protocols [51]. They also allow a recreational interpretation of "cooking cryptographers", that take turns in placing ingredients in a pot, to securely compute a logical AND (i.e. solving the "dating problem" by cooking together).

## References

[1] Y. Abe, M. Iwamoto, K. Ohta. "Efficient Private PEZ Protocols for Symmetric Functions". In: *TCC 2019*. Ed. by D. Hofheinz, A. Rosen. LNCS 11891. Springer, 2019, pp. 372–392. DOI: 10.1007/978-3-030-36030-6_15.

[2] Y. Abe et al. "Five-Card AND Protocol in Committed Format Using Only Practical Shuffles". In: *APKC 2018*. 2018, pp. 3–8. DOI: 10.1145/3197507.3197510.

[3] D. Achenbach, J. Müller-Quade, J. Rill. "Universally Composable Firewall Architectures Using Trusted Hardware". In: *BalkanCryptSec 2014*. Ed. by B. Ors, B. Preneel. LNCS 9024. Springer, 2015, pp. 57–74. DOI: 10.1007/978-3-319-21356-9_5.

[4] D. Achenbach et al. "Oblivious Voting: Hiding Votes from the Voting Machine in Bingo Voting". In: *SECRYPT 2016*. Ed. by C. Callegari et al. SciTePress, 2016, pp. 85–96. DOI: 10.5220/0005964300850096.

[5] D. Achenbach et al. “Towards Realising Oblivious Voting”. In: *ICETE 2016*. Ed. by M. S. Obaidat. CCIS 764. Springer, 2017, pp. 216–240. DOI: 10.1007/978-3-319-67876-4_11.

[6] D. Achenbach et al. “Your Money or Your Life—Modeling and Analyzing the Security of Electronic Payment in the UC Framework”. In: *FC 2019*. Ed. by I. Goldberg, T. Moore. LNCS 11598. Springer, Sept. 30, 2019, pp. 243–261. DOI: 10.1007/978-3-030-32101-7_16.

[7] F. Armknecht et al. “Towards a Unified Security Model for Physically Unclonable Functions”. In: *CT-RSA 2016*. Ed. by K. Sako. LNCS 9610. Springer, 2016, pp. 271–287. DOI: 10.1007/978-3-319-29485-8_16.

[8] J. Balogh et al. “Private computation using a PEZ dispenser”. In: *Theor. Comput. Sci.* 306.1–3 (2003), pp. 69–84. DOI: 10.1016/S0304-3975(03)00210-X.

[9] D. A. Basin, S. Radomirovic, M. Schläpfer. “A Complete Characterization of Secure Human-Server Communication”. In: *CSF 2015*. 2015, pp. 199–213. DOI: 10.1109/CSF.2015.21.

[10] B. den Boer. “More Efficient Match-Making and Satisfiability: The Five Card Trick”. In: *EUROCRYPT ’89*. Ed. by J. Quisquater, J. Vandewalle. LNCS 434. Springer, 1990, pp. 208–217. DOI: 10.1007/3-540-46885-4_23.

[11] B. Broadnax et al. “Fortified Multi-Party Computation: Taking Advantage of Simple Secure Hardware Modules”. In: *Proceedings on Privacy Enhancing Technologies* (4 2021), pp. 312–338. DOI: 10.2478/popets-2021-0072.

[12] C. Brzuska et al. “Physically Uncloneable Functions in the Universal Composition Framework”. In: *CRYPTO 2011*. Ed. by P. Rogaway. LNCS 6841. Springer, 2011, pp. 51–70. DOI: 10.1007/978-3-642-22792-9_4.

[13] R. Canetti. “Universally Composable Security: A New Paradigm for Cryptographic Protocols”. In: *FOCS 2001*. 2001, pp. 136–145. DOI: 10.1109/SFCS.2001.959888.

[14] R. Canetti, M. Fischlin. “Universally Composable Commitments”. In: *CRYPTO 2001*. Ed. by J. Kilian. LNCS 2139. Springer, 2001, pp. 19–40. DOI: 10.1007/3-540-44647-8_2.

[15] R. Canetti, E. Kushilevitz, Y. Lindell. “On the Limitations of Universally Composable Two-Party Computation without Set-up Assumptions”. In: *EUROCRYPT 2003*. Ed. by E. Biham. LNCS 2656. Springer, 2003, pp. 68–86. DOI: 10.1007/3-540-39200-9_5.

[16] R. Canetti et al. “Universally Composable Security with Global Setup”. In: *TCC 2007*. Ed. by S. P. Vadhan. LNCS 4392. Springer, 2007, pp. 61–85. DOI: 10.1007/978-3-540-70936-7_4.

[17] R. Carback et al. “Scantegrity II Municipal Election at Takoma Park: The First E2E Binding Governmental Election with Ballot Privacy”. In: *USENIX Security Symposium 2010*. USENIX, 2010, pp. 291–306. URL: http://www.usenix.org/events/sec10/tech/full_papers/Carback.pdf.

[18] D. Chaum et al. “Scantegrity II: End-to-End Verifiability by Voters of Optical Scan Elections Through Confirmation Codes”. In: *IEEE Trans. Information Forensics and Security* 4.4 (2009), pp. 611–627. DOI: 10.1109/TIFS.2009.2034919.

[19] C. Crépeau, J. Kilian. “Discreet Solitary Games”. In: *CRYPTO ’93*. Ed. by D. R. Stinson. LNCS 773. Springer, 1994, pp. 319–330. DOI: 10.1007/3-540-48329-2_27.

[20] I. Csiszár, J. Körner. “Broadcast channels with confidential messages”. In: *IEEE Trans. Inform. Theory* 24.3 (1978), pp. 339–348. DOI: 10.1109/TIT.1978.1055892.

[21] D. Dachman-Soled et al. “Feasibility and Infeasibility of Secure Computation with Malicious PUFs”. In: *J. Cryptol.* 33.2 (2020), pp. 595–617. DOI: 10.1007/s00145-019-09329-9.

[22] I. Damgård, A. Scafuro. “Unconditionally Secure and Universally Composable Commitments from Physical Assumptions”. In: *ASIACRYPT 2013*. Ed. by K. Sako, P. Sarkar. LNCS 8270. Springer, 2013, pp. 100–119. DOI: 10.1007/978-3-642-42045-0_6.

[23] N. Döttling, D. Kraschewski, J. Müller-Quade. *David & Goliath Oblivious Affine Function Evaluation*. 2012. Cryptology ePrint Archive, Report 2012/135.

[24] N. Döttling, D. Kraschewski, J. Müller-Quade. “Unconditional and Composable Security Using a Single Stateful Tamper-Proof Hardware Token”. In: *TCC 2011*. Ed. by Y. Ishai. LNCS 6597. Springer, 2011, pp. 164–181. DOI: 10.1007/978-3-642-19571-6_11.

[25] N. Döttling et al. “From Stateful Hardware to Resettable Hardware Using Symmetric Assumptions”. In: *ProvSec 2015*. Ed. by M. H. Au, A. Miyaji. LNCS 9451. Springer, 2015, pp. 23–42. DOI: 10.1007/978-3-319-26059-4_2.

[26] N. Döttling et al. “General Statistically Secure Computation with Bounded-Resettable Hardware Tokens”. In: *TCC 2015*. Ed. by Y. Dodis, J. B. Nielsen. LNCS 9014. Springer, 2015, pp. 319–344. DOI: 10.1007/978-3-662-46494-6_14.

[27] N. Döttling et al. “Implementing Resettable UC-Functionalities with Untrusted Tamper-Proof Hardware-Tokens”. In: *TCC 2013*. Ed. by A. Sahai. LNCS 7785. Springer, 2013, pp. 642–661. DOI: 10.1007/978-3-642-36594-2_36.

[28] P. Dvořák, M. Koucký. “Barrington Plays Cards: The Complexity of Card-based Protocols”. In: *STACS 2021*. Ed. by M. Bläser, B. Monmege. 187. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 26:1–26:17. DOI: 10.4230/LIPIcs.STACS.2021.26.

[29] R. Fagin, M. Naor, P. Winkler. “Comparing Information Without Leaking It”. In: *Commun. ACM* 39.5 (1996), pp. 77–85. DOI: 10.1145/229459.229469.

[30] B. A. Fisch, D. Freund, M. Naor. “Secure Physical Computation Using Disposable Circuits”. In: *TCC*

*2015*. Ed. by Y. Dodis, J. B. Nielsen. LNCS 9014. Springer, 2015, pp. 182–198. DOI: 10.1007/978-3-662-46494-6_9.

[31] B. Fisch, D. Freund, M. Naor. "Physical Zero-Knowledge Proofs of Physical Properties". In: *CRYPTO 2014*. Ed. by J. A. Garay, R. Gennaro. LNCS 8617. Springer, 2014, pp. 313–336. DOI: 10.1007/978-3-662-44381-1_18.

[32] B. Fisch et al. "IRON: Functional Encryption using Intel SGX". In: *CCS 2017*. Ed. by B. M. Thuraisingham et al. ACM, 2017, pp. 765–782. DOI: 10.1145/3133956.3134106.

[33] A. Glaser, B. Barak, R. J. Goldston. "A zero-knowledge protocol for nuclear warhead verification". In: *Nature* 510 (2014), pp. 497–502. DOI: 10.1038/nature13457.

[34] V. Goyal et al. "Founding Cryptography on Tamper-Proof Hardware Tokens". In: *TCC 2010*. Ed. by D. Micciancio. LNCS 5978. Springer, 2010, pp. 308–326. DOI: 10.1007/978-3-642-11799-2_19.

[35] C. Hazay, A. Polychroniadou, M. Venkitasubramaniam. "Constant Round Adaptively Secure Protocols in the Tamper-Proof Hardware Model". In: *PKC 2017*. Ed. by S. Fehr. LNCS 10175. Springer, 2017, pp. 428–460. DOI: 10.1007/978-3-662-54388-7_15.

[36] D. Hofheinz, J. Müller-Quade, D. Unruh. "Universally Composable Zero-Knowledge Arguments and Commitments from Signature Cards". In: *MORAVIACRYPT 2005* 37 (2007). Ed. by D. Cvrček et al., pp. 93–103. URL: https://tatra.mat.savba.sk/paper.php?id_paper=887.

[37] S. Izmalkov, M. Lepinski, S. Micali. "Verifiably Secure Devices". In: *TCC 2008*. Ed. by R. Canetti. LNCS 4948. Springer, 2008, pp. 273–301. DOI: 10.1007/978-3-540-78524-8_16.

[38] S. Izmalkov, S. Micali, M. Lepinski. "Rational Secure Computation and Ideal Mechanism Design". In: *FOCS 2005*. IEEE Computer Society, 2005, pp. 585–595. DOI: 10.1109/SFCS.2005.64.

[39] S. Izmalkov et al. *Transparent Computation and Correlated Equilibrium*. URL: https://economics.mit.edu/files/1082.

[40] J. Kastner et al. "The Minimum Number of Cards in Practical Card-Based Protocols". In: *ASIACRYPT 2017*. Ed. by T. Takagi, T. Peyrin. LNCS 10626. Springer, 2017, pp. 126–155. DOI: 10.1007/978-3-319-70700-6_5.

[41] J. Katz. "Universally Composable Multi-party Computation Using Tamper-Proof Hardware". In: *EUROCRYPT 2007*. Ed. by M. Naor. LNCS 4515. Springer, 2007, pp. 115–128. DOI: 10.1007/978-3-540-72540-4_7.

[42] S. Katzenbeisser et al. "PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions (PUFs) Cast in Silicon". In: *CHES 2012*. Ed. by E. Prouff, P. Schaumont. LNCS 7428. Springer, 2012, pp. 283–301. DOI: 10.1007/978-3-642-33027-8_17.

[43] A. Koch. "Cryptographic Protocols from Physical Assumptions". PhD thesis. Karlsruhe: KIT, 2019. DOI: 10.5445/IR/1000097756.

[44] A. Koch. "The Landscape of Optimal Card-based Protocols". In: *Journal of Mathematical Cryptology* (Special Issue: Proceedings of MathCrypt 2021). In press.

[45] A. Koch, M. Schrempp, M. Kirsten. "Card-based Cryptography Meets Formal Verification". In: *ASIACRYPT 2019*. Ed. by S. D. Galbraith, S. Moriai. LNCS 11921. Springer, Nov. 25, 2019, pp. 488–517. DOI: 10.1007/978-3-030-34578-5_18.

[46] A. Koch, S. Walzer. *Private Function Evaluation with Cards*. Nov. 16, 2018. Cryptology ePrint Archive, Report 2018/1113.

[47] A. Koch, S. Walzer, K. Härtel. "Card-Based Cryptographic Protocols Using a Minimal Number of Cards". In: *ASIACRYPT 2015*. Ed. by T. Iwata, J. H. Cheon. LNCS 9452. Springer, 2015, pp. 783–807. DOI: 10.1007/978-3-662-48797-6_32.

[48] B. Magri et al. *Everlasting UC Commitments from Fully Malicious PUFs*. 2021. Cryptology ePrint Archive, Report 2021/248.

[49] U. M. Maurer. "Protocols for Secret Key Agreement by Public Discussion Based on Common Information". In: *CRYPTO 1992*. Ed. by E. F. Brickell. LNCS 740. Springer, 1992, pp. 461–470. DOI: 10.1007/3-540-48071-4_32.

[50] J. Mechler, J. Müller-Quade, T. Nilges. "Reusing Tamper-Proof Hardware in UC-Secure Protocols". In: *PKC 2018*. Ed. by M. Abdalla, R. Dahab. LNCS 10769. Springer, 2018, pp. 463–493. DOI: 10.1007/978-3-319-76578-5_16.

[51] D. Miyahara et al. "Cooking Cryptographers: Secure Multiparty Computation Based on Balls and Bags". In: *CSF 2021*. IEEE, 2021, pp. 1–16. DOI: 10.1109/CSF51468.2021.00034.

[52] T. Mizuki. "Efficient and Secure Multiparty Computations Using a Standard Deck of Playing Cards". In: *CANS 2016*. 2016, pp. 484–499. DOI: 10.1007/978-3-319-48965-0_29.

[53] T. Mizuki, Y. Kugimoto, H. Sone. "Secure Multiparty Computations Using a Dial Lock". In: *TAMC 2007*. Ed. by J. Cai, S. B. Cooper, H. Zhu. LNCS 4484. Springer, 2007, pp. 499–510. DOI: 10.1007/978-3-540-72504-6_44.

[54] T. Mizuki, Y. Kugimoto, H. Sone. "Secure Multiparty Computations Using the 15 Puzzle". In: *COCOA 2007*. Ed. by A. W. M. Dress, Y. Xu, B. Zhu. LNCS 4616. Springer, 2007, pp. 255–266. DOI: 10.1007/978-3-540-73556-4_28.

[55] T. Mizuki, H. Shizuya. "A formalization of card-based cryptographic protocols via abstract machine". In: *Int. J. Inf. Secur.* 13.1 (2014), pp. 15–23. DOI: 10.1007/s10207-013-0219-4.

[56] T. Mizuki, H. Sone. “Six-Card Secure AND and Four-Card Secure XOR”. In: *FAW 2009*. 2009, pp. 358–369. DOI: 10.1007/978-3-642-02270-8_36.

[57] T. Moran, M. Naor. “Basing cryptographic protocols on tamper-evident seals”. In: *Theor. Comput. Sci.* 411.10 (2010). Ed. by M. Yung, pp. 1283–1310. DOI: 10.1016/j.tcs.2009.10.023.

[58] T. Moran, M. Naor. “Polling with Physical Envelopes: A Rigorous Analysis of a Human-Centric Protocol”. In: *EUROCRYPT 2006*. Ed. by S. Vaudenay. LNCS 4004. Springer, 2006, pp. 88–108. DOI: 10.1007/11761679_7.

[59] T. Moran, M. Naor. “Receipt-Free Universally-Verifiable Voting with Everlasting Privacy”. In: *CRYPTO 2006*. Ed. by C. Dwork. LNCS 4117. Springer, 2006, pp. 373–392. DOI: 10.1007/11818175_22.

[60] T. Moran, M. Naor. “Split-ballot voting: Everlasting privacy with distributed trust”. In: *ACM Trans. Inf. Syst. Secur.* 13.2 (2010), 16:1–16:43. DOI: 10.1145/1698750.1698756.

[61] T. Moran, M. Naor, G. Segev. “Deterministic History-Independent Strategies for Storing Information on Write-Once Memories”. In: *ICALP 2007*. Ed. by L. Arge et al. LNCS 4596. Springer, 2007, pp. 303–315. DOI: 10.1007/978-3-540-73420-8_28.

[62] J. Müller-Quade, D. Unruh. “Long-Term Security and Universal Composability”. In: *TCC 2007*. Ed. by S. P. Vadhan. LNCS 4392. Springer, 2007, pp. 41–60. DOI: 10.1007/978-3-540-70936-7_3.

[63] M. Naor, A. Shamir. “Visual Cryptography”. In: *EUROCRYPT ’94*. Ed. by A. D. Santis. LNCS 950. Springer, 1995, pp. 1–12. DOI: 10.1007/BFb0053419.

[64] K. Nayak et al. “HOP: Hardware makes Obfuscation Practical”. In: *NDSS 2017*. The Internet Society, 2017. URL: https://www.ndss-symposium.org/ndss2017/ndss-2017-programme/hop-hardware-makes-obfuscation-practical/.

[65] V. Niemi, A. Renvall. “Secure Multiparty Computations Without Computers”. In: *Theor. Comput. Sci.* 191.1-2 (1998), pp. 173–183. DOI: 10.1016/S0304-3975(97)00107-2.

[66] V. Niemi, A. Renvall. “Solitaire Zero-knowledge”. In: *Fundam. Inform.* 38.1,2 (1999), pp. 181–188. DOI: 10.3233/FI-1999-381214.

[67] T. Nilges. “The Cryptographic Strength of Tamper-Proof Hardware”. PhD thesis. Karlsruhe: KIT, 2015. DOI: 10.5445/IR/1000051809.

[68] R. Ostrovsky et al. “Universally Composable Secure Computation with (Malicious) Physically Uncloneable Functions”. In: *EUROCRYPT 2013*. Ed. by T. Johansson, P. Q. Nguyen. LNCS 7881. Springer, 2013, pp. 702–718. DOI: 10.1007/978-3-642-38348-9_41.

[69] R. Pappu et al. “Physical One-Way Functions”. In: *Science* 297.5589 (2002), pp. 2026–2030. DOI: 10.1126/science.1074376.

[70] R. Pass, E. Shi, F. Tramèr. “Formal Abstractions for Attested Execution Secure Processors”. In: *EUROCRYPT 2017*. Ed. by J. Coron, J. B. Nielsen. LNCS 10210. 2017, pp. 260–289. DOI: 10.1007/978-3-319-56620-7_10.

[71] S. Popoveniuc, B. Hosp. “An Introduction to Punch-Scan”. In: *Towards Trustworthy Elections. New Directions in Electronic Voting*. Ed. by D. Chaum et al. LNCS 6000. Springer, 2010, pp. 242–259. DOI: 10.1007/978-3-642-12980-3_15.

[72] S. Ruangwises, T. Itoh. “AND Protocols Using only Uniform Shuffles”. In: *CSR 2019*. 2019, pp. 349–358. DOI: 10.1007/978-3-030-19955-5_30.

[73] B. Schneier. *The Solitaire Encryption Algorithm*. 1999. URL: https://www.schneier.com/academic/solitaire/ (visited on 07/11/2019).

[74] K. Shinagawa et al. “Secure Multi-Party Computation Using Polarizing Cards”. In: *IWSEC 2015*. Ed. by K. Tanaka, Y. Suga. LNCS 9241. Springer, 2015, pp. 281–297. DOI: 10.1007/978-3-319-22425-1_17.

[75] A. Toponce. *Playing Card Ciphers*. 2018. URL: https://aarontoponce.org/wiki/crypto/card-ciphers (visited on 08/23/2019).

[76] F. Tramèr et al. “Sealed-Glass Proofs: Using Transparent Enclaves to Prove and Sell Knowledge”. In: *EuroS&P 2017*. IEEE, 2017, pp. 19–34. DOI: 10.1109/EuroSP.2017.28.

[77] A. D. Wyner. “The wire-tap channel”. In: *Bell. Syst. Tech. J.* 54.8 (1975), pp. 1335–1387. DOI: 10.1002/j.1538-7305.1975.tb02040.x.