Games and the Impossibility of Realizable Ideal Functionality

Michael Backes, Anupam Datta, Ante Derek, John Mitchell, Ajith Ramanathan, Andre Scedrov

Games and the Impossibility of Realizable Ideal Functionality

Michael Backes[1], Anupam Datta[2], Ante Derek[2], John C. Mitchell[2], Ajith Ramanathan[2], and Andre Scedrov[3]

[1] Saarland University backes@cs.uni-sb.de
[2] Stanford University {danupam,aderek,jcm,ajith}@cs.stanford.edu
[3] University of Pennsylvania scedrov@math.upenn.edu

Abstract. A cryptographic primitive or a security mechanism can be specified in a variety of ways, such as a condition involving a game against an attacker, construction of an ideal functionality, or a list of properties that must hold in the face of attack. While game conditions are widely used, an ideal functionality is appealing because a mechanism that is indistinguishable from an ideal functionality is therefore guaranteed secure in any larger system that uses it. We relate ideal functionalities to games by defining the *set* of ideal functionalities associated with a game condition and show that under this definition, which reflects accepted use and known examples, a number of cryptographic concepts do not have any realizable ideal functionality in the plain model. Some interesting examples are multiparty coin-tossing, bit-commitment and shared random sequences. One interpretation of this negative result is that equational approaches based on computational observational equivalence might be better applied to reasoning about game conditions than equivalence with ideal functionalities. Alternatively, generality might be obtained by allowing for various setup assumptions, or by other means.

1 Introduction

Many security conditions about cryptographic primitives are expressed using a form of game. For example, the condition that an encryption scheme is semantically secure against chosen ciphertext attack (IND-CCA2) [1] may be expressed naturally by saying that no adversary has better than negligible probability to win a certain game against a challenger. In this definition, the game itself clearly identifies the information and actions available to the adversary, and the condition required to win the game identifies the properties that must be preserved in the face of attack. Another way of specifying security properties uses ideal functionalities [2–5]. In this approach, usually referred to as Universal Composability [3] (UC) or Reactive Simulatability [6] an idealized way of achieving some goal is presented, possibly using mechanisms such as authenticated channels and trusted third parties that are not basic primitives in practice. An implementation is then considered secure if no feasible attacker can distinguish the implementation from the ideal functionality, in any environment. An advantage of this

approach is that indistinguishability from an ideal functionality leads to composable notions of security [3, 5, 7]. In contrast, if a mechanism satisfies a game condition, there is no guarantee regarding how the mechanism will respond to interactions that do not arise in the specified game.

In this work, we describe a framework for comparing game specifications and ideal functionalities, and prove some negative results about the existence of ideal functionalities in certain settings. While most known primitives have game-based definitions (see, e.g., [8]), it has proven difficult to develop useful ideal functionalities for some natural primitives. Some interesting issues are explored in [9, 10], which describe a series of efforts to develop a suitable ideal functionality for digital signatures. In brief, there is a widely accepted game condition for digital signatures, existential unforgeability against chosen message attacks, formulated in [11]. However, there are many possible ideal functionalities that are consistent with this game condition. For example, a functionality could either explicitly disclose information about messages that were signed in the past, or not disclose this information. More generally, given a game condition, it is often feasible to formulate various functionalities that satisfy the game condition yet reveal varying kinds of "harmless" information that does not seem relevant to the goals of the mechanism.

If we have a game or set of games that define a concept like secure encryption, digital signature, or bit-commitment, then we would like to identify precisely the set of possible ideal functionalities associated with each game condition. Since an ideal functionality is intended to be evidently secure by construction, we propose that an *ideal* functionality must satisfy the given game condition on information-theoretic grounds, rather than as a result of computational complexity arguments. Applied to encryption, for example, this means that an ideal functionality for encryption must not provide *any* information about bits of the plaintext to the adversary. Our definition of *ideal functionality* for a set of game conditions is consistent with all examples we have found in the literature, and reflects the useful idea that it should be easier to reason about systems that use an ideal functionality than about systems that use a real protocol.

Using our definition, we investigate connections between multiparty cointossing, bit-commitment, a form of group signatures, a form of symmetric encryption with integrity guarantees, and the common reference string (CRS) model. One sample result (previously reported in TCC 2006) is that while bit-commitment may be specified using games, there is no realizable ideal functionality for bit-commitment. This may be seen as a negative result about specification using ideal functionality, since there are constructions of bit-commitment protocols that are provably correct under modest cryptographic assumptions (see, e.g., [12]). We also show that there is no realizable ideal functionality for other reasonable and implementable cryptographic primitives, including a form of group signatures and a form of symmetric encryption with integrity guarantees, under certain conditions that allow the encryption key to be revealed after it is used.

Theorem 1. *If $\mathcal{F}$ is an ideal functionality for bilateral bit-commitment, then there does not exist a terminating real protocol $\mathcal{P}$ that securely realizes $\mathcal{F}$.*

Theorem 2. *If $\mathcal{G}$ is a functionality and P is a terminating $\mathcal{G}$-hybrid protocol for bit-commitment which is correct with high probability and provides perfect hiding and perfect binding, then no protocol realizes functionality $\mathcal{G}$.*

Corollary 1. *If $\mathcal{F}$ is a functionality for symmetric encryption providing perfect CCA-security and perfect integrity of ciphertext then $\mathcal{F}$ cannot be realized.*

Corollary 2. *If $\mathcal{F}$ is a functionality for group signatures providing perfect anonymity and perfect traceability then $\mathcal{F}$ cannot be realized.*

The intuition behind our impossibility result is relatively simple. Illustrated using bit-commitment, a good commitment scheme must have two properties: the commitment token must not reveal any information about the chosen bit, while subsequent decommitment must reveal a verifiable relationship between the chosen bit and the commitment token. These are contradictory requirements because the first condition suggests that tokens must be chosen randomly, while the second implies that they are not. Similar "decommitment" issues arise in symmetric encryption or keyed hash, if the encryption key is revealed after some messages using the key have been sent on the visible network. At a more technical level, our proof by contradiction works by showing that if there was a realization of the ideal functionality for bit-commitment, it could be transformed into a protocol for bit-commitment that achieves perfect hiding and binding without using a trusted third party. However, it is well known that such a protocol does not exist [12]. While impossibility results for group signatures and symmetric encryption could be proved by instantiating the general proof method, we present simpler proofs by reducing bit-commitment to these primitives.

In a previous study of ideal functionality for bit commitment, Canetti and Fischlin show that a particular ideal functionality for bit-commitment is not realized by any real protocol [13]. In related work, Canetti [3] shows that particular functionalities for ideal coin tossing, zero-knowledge, and oblivious transfer are not realizable. Canetti et al [14] show that a class of specific functionalities for secure multi-party computation are not realizable, while Canetti and Krawczyk [15] compare indistinguishability-based and simulatability-based definitions of security in the context of key-exchange protocols. Our results are more general since we prove that, given a *game* definition of a primitive, there is *no* realizable ideal functionality associated with that game condition. In addition, our proof is different in that it provides a reduction to a previous negative result independent of universal composability [12], and appears to apply immediately to many primitives.

A different approach for proving impossibility for a class of functionalities is taken in [16, 17]. Using the simulation relations between the symbolic (Dolev-Yao) and the computational model of execution, the authors show that XOR and hashing primitives can never have realizable symbolic (and therefore ideal) functionalities. Informally, for XOR it is shown that if such a functionality existed, it

could be used to compute cryptographic operations and thus would not constitute a Dolev-Yao abstraction of XOR. Since ideal functionalities for XOR might be characterized, in principle, by a set of game conditions, their non-existence might be provable in our framework. As far as hash functions are concerned, our framework seems applicable for proving their impossibility provided that hashes are intended to offer ideal secrecy; for cases where hashes are only meant to offer collision-resistance, establishing impossibility within our framework requires additional work. We thus consider it an interesting future direction to explore the relationship between both approaches.

Recent work also established impossibility results for Dolev-Yao models, i.e. term-based abstractions of cryptography, containing XOR and hashing operations [16, 17]. Achieving these results within our framework requires further investigations: While ideal XOR functionalities might be characterized, in principle, by a game condition game so that their non-existence might be provable in our frame-work, it was shown in [16] that even if such a functionality existed, it could be used to compute cryptographic operations and thus would not constitute a Dolev-Yao abstraction of XOR. As far as Dolev-Yao models with hashes are concerned, our framework seems applicable for proving their impossibility provided that hashes are intended to offer ideal secrecy; for cases where hashes are only meant to offer collision-resistance without secrecy however, establishing impossibility within our framework requires additional work. We thus consider it interesting future work to explore the relationship between both approaches.

A related issue is the choice of so-called "setup assumptions," such as public-key infrastructure, and common reference string. Our negative results hold under some setup assumptions, such as the absence of shared private information, or the presence of a trusted certificate authority (or PKI), and fail for other setup assumptions, such as the assumption of a common reference string. This is expected, since [13] construct a realizable ideal functionality in the common reference string model. We have yet to characterize precisely the set of possible setup assumptions under which our negative results hold.

Another closely related issue is the choice of the interface an ideal functionality is supposed to offer to its surrounding protocols. While different interfaces might be preferable for different protocols, some of these interfaces might give rise to impossibility proofs. For instance, public-key encryption can be suitably defined by an ideal functionality as long as secret keys cannot be imported from or exported to a surrounding protocol via the functionality's interface. If such import or export routines are desirable, e.g., for modelling key distribution or resource-bounded devices that outsource key generation, suitable ideal functionalities do not exist anymore.

The basic impossibility proof has a few simple steps that can be applied to a range of primitives and we aim to develop general sufficient conditions that characterize its applicability. One intriguing direction involves multiparty coin-tossing, which may be shown to have no realizable ideal functionality by the same argument as bit commitment. On the other hand, a form of repeated multi-party coin-tossing produces a common reference string (CRS) among communicating

parties. By the construction of Canetti and Fischlin [13], a CRS enables bit commitment. Therefore, it the existence of a realizable ideal functionality for bit commitment, the existence of a realizable ideal functionality for multi-party coin tossing, and the availability of a common reference string are closely related.

References

1. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Proceeding. Volume 1807 of Lecture Notes in Computer Science., Springer-Verlag (2000) 259–274
2. Abadi, M., Gordon, A.D.: A calculus for cryptographic protocols: the spi calculus. Information and Computation **143** (1999) 1–70 Expanded version available as SRC Research Report 149 (January 1998).
3. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS '01: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science. (2001) 136 Full version available at `http://eprint.iacr.org/`.
4. Lincoln, P., Mitchell, J.C., Mitchell, M., Scedrov, A.: A probabilistic poly-time framework for protocol analysis. In: ACM Conference on Computer and Communications Security. (1998) 112–121
5. Pfitzmann, B., Waidner, M.: Composition and integrity preservation of secure reactive systems. In: ACM Conference on Computer and Communications Security. (2000) 245–254
6. Backes, M., Pfitzmann, B., Waidner, M.: A composable cryptographic library with nested operations. In: CCS '03: Proceedings of the 10th ACM conference on Computer and communications security, ACM Press (2003) 220–230
7. Backes, M., Pfitzmann, B., Waidner, M.: A general composition theorem for secure reactive systems. In: TCC '04: Proceedings of the 1st Theory of Cryptography Conference. Volume 2951 of Lecture Notes in Computer Science., Springer-Verlag (2004) 336–354
8. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332 (2004) `http://eprint.iacr.org/2004/332`.
9. Backes, M., Hofheinz, D.: How to break and repair a universally composable signature functionality. In: Information Security, 7th International Conference, ISC 2004, Proceedings. Volume 3225 of Lecture Notes in Computer Science., Springer-Verlag (2004) 61–72
10. Canetti, R.: Universally composable signature, certification, and authentication. In: CSFW '04: Proceedings of the 17th IEEE Computer Security Foundations Workshop, IEEE Computer Society (2004) 219–233
11. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal on Computing **18**(1) (1989) 186–208
12. Goldreich, O.: Foundations of Cryptography: Basic Tools. Cambridge University Press (2000)
13. Canetti, R., Fischlin, M.: Universally composable commitments. In: Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Proceedings. Volume 2139 of Lecture Notes in Computer Science., Springer-Verlag (2001) 19–40

14. Canetti, R., Kushilevitz, E., Lindell, Y.: On the limitations of universally composable two-party computation without set-up assumptions. In: Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings. Volume 2656 of Lecture Notes in Computer Science., Springer-Verlag (2003) 68–86
15. Canetti, R., Krawczyk, H.: Analysis of key-exchange protocols and their use for building secure channels. In: Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Proceeding. Volume 2045 of Lecture Notes in Computer Science., Springer-Verlag (2001) 453–474
16. Backes, M., Pfitzmann, B.: Limits of the cryptographic realization of Dolev-Yao-style XOR. In: ESORICS'05: Proceedings of the 10th European Symposium on Research in Computer Security. Volume 3679 of Lecture Notes in Computer Science. (2005) 178–196
17. Backes, M., Pfitzmann, B., Waidner, M.: Limits of the reactive simulatability/uc of dolev-yao models with hashes. In: ESORICS'06: Proceedings of the 11th European Symposium on Research in Computer Security. Lecture Notes in Computer Science (2006)