

Research Statement

Yehuda Lindell

Dept. of Computer Science
Bar-Ilan University, ISRAEL.
lindell@cs.biu.ac.il
www.cs.biu.ac.il/~lindell

July 11, 2005

The main focus of my research is the theoretical foundations of cryptography. This research is mainly concerned with the feasibility or infeasibility of securely realizing computational and cryptographic tasks. The answer to whether or not a task can be securely realized often depends on the assumed power of the adversary and the network model (this model includes, for example, any trusted setup assumptions like those used in setting up a public-key infrastructure). Our aim is to obtain security against the most powerful (yet realistic) adversary, while minimizing the need for trust. Beyond questions of feasibility, the theory of cryptography is also concerned with understanding the computational resources that are needed for solving cryptographic tasks (much like the questions that are central to the fields of algorithms and complexity). The resources most often considered are computational complexity (i.e., measuring the time required to compute some cryptographic function like encryption) and communication complexity (i.e., the number of rounds and bandwidth required for a secure interactive protocol).

Secure Multi-Party Computation and Protocol Composition

In the setting of multi-party computation, sets of parties with private inputs wish to jointly and securely compute some function of their inputs. Security requirements on such a computation are that the outputs received by the parties are correctly distributed, and furthermore, that the privacy of each party's input is preserved as much as possible. This encompasses any distributed computing task and includes computations as simple as coin-tossing and broadcast, and as complex as electronic voting, electronic auctions, electronic cash schemes and anonymous transactions. Classically, research on secure multi-party computation focused on the *stand-alone* setting, where a single set of parties execute a single protocol in isolation. In the late 80's it was shown that in this setting, *any* function can be securely computed. However, in modern network settings, it is usually the case that many different sets of parties run many different protocol executions at the same time. Furthermore, an active adversary can try to utilize these many different executions in order to somehow "break" the security of the individual protocols. A protocol that is secure in such a multi-execution setting is said to remain secure under *protocol composition*.

Security under composition. Unfortunately, protocols that are secure in the stand-alone setting do not necessarily remain secure under composition. In fact, in joint work with Lysyanskaya and Rabin, we show that there exist multi-party tasks (specifically, authenticated Byzantine agreement with a third or more corrupted parties) which have secure protocols for the stand-alone setting, but which cannot be solved by *any* protocol in the setting of composition [1].

In light of the above, a highly important research project is the reestablishment of the feasibility results of the 80's for secure computation under composition. Until recently, very little was known about this question. I have made a significant contribution to the understanding of what can and cannot be securely computed under composition, and under what assumptions. First, in joint work with Canetti and Kushilevitz, we proved that large classes of two-party functions cannot be securely computed under the specific definition

of universal composability [4]. The importance of this definition is that it implies security under *concurrent general composition*, where a secure protocol runs concurrently with arbitrary other protocols (as in an Internet-like setting). Following this, I presented a series of impossibility results that hold for *any* definition implying security under concurrent general composition, as well as for *concurrent self composition*, where a single protocol is run many times concurrently (but nothing else is run in the network) [5, 7, 8]. In addition to the above negative results, I also present protocol constructions (of course, where impossibility does not hold). Specifically, together with Canetti, Ostrovsky and Sahai, we show that in the common reference string model, any multi-party function can be securely computed under the definition of universal composability, for any number of corrupted parties [2]. (In the common reference string model, all parties are given access to a string chosen in a trusted setup phase.) Thus, at the price of accepting trust for a one-time setup of the reference string, security in Internet-like settings can be achieved. (Note that our impossibility results apply to the *plain model*, where no trusted setup is assumed. Therefore, they do not rule out the existence of protocols in the common reference string model.) I also show that in the plain model, with no setup assumptions, any two-party function can be securely computed under *bounded concurrent self composition* (in this model, there is some fixed a-priori bound on the number of executions of the secure protocol) [5]. Unfortunately, this model requires a problematic *global* assumption regarding the number of concurrent executions. In addition, secure protocols for this model suffer from lower bounds on their complexity, as demonstrated in [5, 8]. It is therefore of interest to search for alternative, more reasonable, models where security may be obtained. One such model is the *timing model* where it is assumed that parties have local clocks that proceed at approximately the same rate (note that there is no requirement on clock synchronization, but just that they do not run at radically different rates). In this model, we show that any multiparty functionality can be securely computed under concurrent general composition, as long as the messages of other arbitrary protocols are (locally) delayed by a fixed number of time units [9].

In summary, the above-mentioned works provide a rather complete picture of the feasibility and infeasibility of secure computation under composition. In the plain model, broad impossibility applies to both concurrent general and concurrent self composition. However, assuming a common reference string, it is possible to obtain universally composable protocols for any multi-party function (recall that such protocols are secure under general composition). Furthermore, in the case of no trusted setup phase, it is possible to achieve *bounded concurrent self composition*, and concurrent general composition in the timing model. A survey on the different notions of security under composition and what is known about them can be found in the introduction of my book on the topic [3].

- [1] Y. Lindell, A. Lysyanskaya and T. Rabin. On the Composition of Authenticated Byzantine Agreement. In the *34th ACM Symposium on the Theory of Computing (STOC)*, pages 514–523, 2002.
- [2] R. Canetti, Y. Lindell, R. Ostrovsky and A. Sahai. Universally Composable Two-Party and Multi-Party Secure Computation. In the *34th ACM Symposium on the Theory of Computing (STOC)*, pages 494–503, 2002.
- [3] Y. Lindell. *Composition of Secure Multi-Party Protocols – A Comprehensive Study*. Lecture Notes in Computer Science Vol. 2815, Springer-Verlag, 2003.
- [4] R. Canetti, E. Kushilevitz and Y. Lindell. On the Limitations of Universally Composable Two-Party Computation Without Set-Up Assumptions. To appear in the *Journal of Cryptology*. An extended abstract appeared in *Advances in Cryptology – EUROCRYPT 2003*, Springer-Verlag (LNCS 2656), pages 68–86, 2003.
- [5] Y. Lindell. Bounded-Concurrent Secure Two-Party Computation Without Setup Assumptions. In the *35th ACM Symposium on the Theory of Computing (STOC)*, pages 683–692, 2003.
- [6] Y. Lindell. Brief Announcement: Impossibility Results for Concurrent Secure Two-Party Computation. In the *22nd ACM Symposium on the Principles of Distributed Computing (PODC)*, page 200, 2003.
- [7] Y. Lindell. General Composition and Universal Composability in Secure Multi-Party Computation. In the *44th IEEE Symposium on the Foundations of Computer Science (FOCS)*, pages 394–403, 2003.

- [8] Y. Lindell. Lower Bounds for Concurrent Self Composition. In the *1st Theory of Cryptography Conference (TCC)*, Springer-Verlag (LNCS 2951), pages 203–222, 2004.
- [9] Y. Kalai, Y. Lindell and M. Prabhakaran. Concurrent General Composition of Secure Protocols in the Timing Model. In the *37th STOC*, 2005.

Secure computation in the stand-alone model. In addition to the above works on protocol composition, I have also considered the basic, stand-alone case. In particular, I have demonstrated the feasibility of obtaining *constant-round* protocols for securely computing any two-party functionality [10] (in previous protocols, the number of rounds was at least linear in the security parameter). Together with Goldwasser, we have also shown that, in contrast to popular belief, secure broadcast is not needed for obtaining secure multi-party computation [11]. This shows that it is not necessary to assume that parties have access to a physical broadcast channel or can run authenticated Byzantine agreement protocols. This enables us to reduce the required setup assumptions for secure computation (even in the stand-alone setting) and, due to [1], also has ramifications on protocol composition. Continuing in the above line of research that attempts to understand what setup assumptions are truly needed for obtaining secure protocols, together with Barak, Canetti, Pass and Rabin, we have shown that meaningful secure computation can be carried out *without* authenticated channels [12]. We note that until this work, all known protocols assumed that all parties are connected via authenticated point-to-point channels. Beyond the theoretical question of what can be obtained with no setup assumptions whatsoever, this result has a number of applications. For just one example, we show that in a partially authenticated network (like the Internet today where servers typically have certificates for digital signatures, but clients do not), it is possible to carry out secure computation while obtaining a meaningful security guarantee.

One of the most fundamental results in the field of secure computation is the two-party, constant-round protocol of Yao for the semi-honest adversarial case (FOCS 1986). Despite its importance, a full and explicit proof of security has never been provided for the protocol. In [13], we remedy this situation and provide a complete description and proof of Yao's protocol.

- [10] Y. Lindell. Parallel Coin-Tossing and Constant-Round Secure Two-Party Computation. In the *Journal of Cryptology*, 16(3):143–184, 2003. (An extended abstract appeared in *Advances in Cryptology – CRYPTO 2001*, Springer-Verlag (LNCS 2139), pages 171–189, 2001.)
- [11] S. Goldwasser and Y. Lindell. Secure Computation Without Agreement. In the *Journal of Cryptology*, 18(3):247–287, 2005. (An extended abstract appear in the *16th International Conference on Distributed Computing (DISC)*, Springer-Verlag (LNCS 2508), pages 17–32, 2002.)
- [12] B. Barak, R. Canetti, Y. Lindell, R. Pass and T. Rabin. Secure Computation Without Authentication. To appear in *CRYPTO 2005*.
- [13] Y. Lindell and B. Pinkas. A Proof of Yao's Protocol for Secure Two-Party Computation.

Efficient secure computation. Feasibility results for secure computation are important for understanding what can and cannot be achieved *in principle*. However, a positive feasibility result does not necessarily provide us with a protocol that is efficient enough for use in practice (and in fact, rarely does). Therefore, once feasibility has been established, the next important step is to construct protocols that can actually be used. In this direction, I am particularly interested in questions related to *privacy preserving data mining*. In joint work with Pinkas, we have presented highly efficient secure protocols for the problem of decision tree learning, with an application to privacy preserving data mining [14]. I have also written a short chapter on the topic of privacy preserving data mining that is directed to a general audience [15].

- [14] Y. Lindell and B. Pinkas. Privacy Preserving Data Mining. *Journal of Cryptology*, 15(3):177–206, 2002. (An extended abstract appeared in *Advances in Cryptology – CRYPTO 2000*, Springer-Verlag (LNCS 1880), pages 36–54, 2000.)
- [15] Y. Lindell. Secure Computation for Privacy Preserving Data Mining. In the *Encyclopedia of Data Warehousing and Mining*, Idea Publishing Group, 2005.

Zero Knowledge

Zero-knowledge proofs have the amazing property that a prover can convince a verifier that a statement is correct, without revealing any information beyond the correctness of the statement. This is defined by showing that a verifier can “simulate” its entire conversation with the prover by itself; thus, anything learned from the prover could be obtained by itself. Zero-knowledge proofs have been demonstrated for all languages in \mathcal{NP} and have become a central tool in cryptographic constructions. In joint work with others, I have studied different aspects of zero-knowledge and have presented both lower bounds and constructions.

- [16] B. Barak, O. Goldreich, S. Goldwasser and Y. Lindell. *Resettably-Sound Zero-Knowledge and its Applications*. In the *42nd IEEE Symposium on the Foundations of Computer Science*, pages 116–125, 2001.
- [17] B. Barak and Y. Lindell. *Strict Polynomial-Time in Simulation and Extraction*. In the *SIAM Journal on Computing (SICOMP)*, 33(4):783–818, 2004.
- [18] B. Barak, Y. Lindell and S. Vadhan. *Lower Bounds for Non-Black-Box Zero Knowledge*. To appear in the *Journal of Computer and System Sciences (FOCS special issue)*. An extended abstract appeared in the *44th IEEE Symposium on the Foundations of Computer Science (FOCS)*, pages 384–393, 2003.

Password Based Authentication

One of the most basic problems of cryptography is that of authentication and session-key generation. In this problem, two parties interact over an insecure network in order to obtain a secret key that can be used for secret and reliable communication. In order to prevent the adversary from learning the key, the parties must share some initial secret information. In practice, this secret information is most commonly a low-entropy password that can be remembered by humans. Together with Goldreich, we present the *first* protocol for password-based authenticated session-key generation that does not rely on any additional setup assumptions, and has a rigorous proof of security [19]. Thus, our work constitutes the first proof of feasibility for this important problem. In joint work with Gennaro, we also present efficient protocols for password-based session-key generation, in the setting where all parties have access to a common reference string [20]. Further extending this work to the setting of composition, together with Canetti, Halevi, Katz and MacKenzie, we show the existence of *universally-composable* password-based protocols for session-key generation in the common reference string model [21]. These protocols also guarantee security when passwords are arbitrarily chosen by users (and are not necessarily uniformly distributed in some dictionary).

- [19] O. Goldreich and Y. Lindell. *Session-Key Generation using Human Passwords Only*. In *Advances in Cryptology – CRYPTO 2001*, Springer-Verlag (LNCS 2139), pages 408–432, 2001.
- [20] R. Gennaro and Y. Lindell. *A Framework for Password-Based Authenticated Key Exchange*. In *Advances in Cryptology – EUROCRYPT 2003*, Springer-Verlag (LNCS 2656), pages 524–543, 2003.
- [21] R. Canetti, S. Halevi, J. Katz, Y. Lindell and P. Mackenzie. *Universally Composable Password-Based Key Exchange*. In *Advances in Cryptology – EUROCRYPT 2005*, Springer-Verlag (LNCS 3494), pages 404–421, 2005.

Secure Encryption

As we have mentioned, a central focus of theoretical cryptography is that of proving feasibility of cryptographic tasks. One of the most basic cryptographic tasks is that of encryption, and as such, feasibility results for this task are of great importance. In [22], I present a simpler proof of the existence of public-key encryption schemes that are secure against adaptive chosen-ciphertext attacks. This simplification makes it possible to teach this important result in a graduate course, something that would have previously been much more difficult.

- [22] Y. Lindell. *A Simpler Construction of CCA2-Secure Public-Key Encryption Under General Assumptions*. In *Advances in Cryptology – EUROCRYPT 2003*, Springer-Verlag (LNCS 2656), pages 241–254, 2003. To appear in the *Journal of Cryptology*.

Expected versus Strict Polynomial-Time in Cryptography

The standard notion of efficient computation is that of probabilistic polynomial-time. However, in many cases in cryptography, *expected* polynomial-time strategies are used. In fact, there are many protocols for which the only simulation strategies that are known run in expected polynomial-time. I am interested in questions related to this, and in [10, Appendix A] I began looking at what happens to zero-knowledge protocols when the verifier (and not just the simulator) is allowed to run in expected polynomial-time. Together with Katz, we consider this problem in general and offer first solutions to the problems that arise due to the use of expected polynomial-time simulation strategies [23]. A different approach to this question was taken in [17], where together with Barak, we show that there are inherent limitations to (black-box) strict polynomial-time simulation strategies. On the positive side, we also show that using nonblack-box techniques, it is possible to obtain constant-round zero-knowledge proofs of knowledge with strict polynomial-time extractors.

- [23] J. Katz and Y. Lindell. Handling Expected Polynomial-Time Strategies in Simulation-Based Security Proofs. In the *2nd Annual Theory of Cryptography Conference (TCC)*, Springer-Verlag (LNCS 3378), pages 128–149, 2005.