Appears in *Proceedings of First International Workshop on Information Security (ISW 97)*, Lecture Notes in Computer Science Vol. 1396, E. Okamoto, G. Davida and M. Mambo eds., Springer Verlag, 1998. Also in Lecture Notes in Computer Science Vol. 1561, I. Damgård ed., Springer Verlag, 1999.

Practice-Oriented Provable-Security

Mihir Bellare[*]

September 1998

[*]Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA. E-Mail: mihir@cs.ucsd.edu. URL: http://www-cse.ucsd.edu/users/mihir. Supported in part by NSF CAREER Award CCR-9624439 and a 1996 Packard Foundation Fellowship in Science and Engineering.

Contents

1 Introduction

This short article is intended to complement my talk. I would like to try to introduce you to a certain, relatively new sub-area of cryptography that we have been calling *practice-oriented provable-security*. It is about applying the ideas of "provably security" to the derivation of practical, secure protocols. I believe it is a fruitful blend of theory and practice that is able to enrich both sides and has by now had some impact on real world security.

A few years ago, provable security was largely known only to theoreticians. This has been changing. We are seeing a growing appreciation of provable security in practice, leading in some cases to the use of such schemes in preference to other ones. Indeed it seems standards bodies and implementors now view provable security as an attribute of a proposed scheme. This means that a wider audience needs an understanding of the basic ideas behind provable security.

This article is directed at practioners and theoreticians alike. For the first I hope it will help to understand what provable security is and isn't, why it is useful, how to evaluate the provable security of a scheme, and where to look for such schemes. For the second group, it can serve to acquaint them with how the ideas with which they are familiar are being applied.

I will begin by describing the basic idea behind provable security. (For many of you, this will be mostly recall, but some novel viewpoints or examples may enter.) Next, I will discuss the practice-oriented approach. I will discuss its main ideas, the problems it has addressed, and briefly survey known results. I hope to leave you feeling there is scope here both for interesting research and for application.

2 Protocols, primitives, proofs and practice

The basic task in cryptography is to enable to parties to communicate "securely" over an insecure channel, namely in a way that guarantees privacy and authenticity of their transmissions. (There are many other tasks as well, but we will begin by thinking about this basic one.)

2.1 Protocols and primitives: the problem

Protocols: the end goal. To enable secure communication, one wants cryptographic *protocols* or *schemes*. For example, an encryption scheme enables users to communicate privately. Such a scheme is specified by a pair $(\mathcal{E}, \mathcal{D})$ of algorithms. The first, run by the sender, takes a *key* and the *plaintext* M to create a *ciphertext* C, which is transmitted to the receiver. The latter applies $\mathcal{D}$, which takes a key and the received ciphertext to recover the plaintext. (Roughly, the security property desired is that an adversary can't learn anything useful about the plaintext given the ciphertext, but we will get into this more later.) They key could be a shared one (this is the private key or symmetric setting) or the keys for encryption and decryption could be different (the public key or asymmetric setting). Designing an encryption scheme means designing the two algorithms $\mathcal{E}$ and $\mathcal{D}$.

Similarly, a message authentication scheme (or protocol) enables parties to tag their data so that the recipient is assured that the data originates with the person claiming to have sent it and has not been tampered with on the way.

The design of such protocols is the end goal for the cryptographer. However, it is not an easy one to reach. What makes it reachable at present is that we have very good *primitives* on which to *base* these protocols.

Primitives: the tools. Julius Caesar also wanted to design protocols. He had a much harder time than we do today, because he didn't have DES or the RSA function.

The latter are examples of what I will call *atomic primitives.* Certainly, they are cryptographic objects of some sort. What is it that distinguishes them from protocols? The distinction is that in their purest and rawest state, atomic primitives don't solve any cryptographic problem we actually care about. We must *use* them appropriately to construct protocols to solve the problems that matter. For example, DES based CBC encryption is a way of using DES to do symmetric encryption. By first hashing a message and then decrypting under RSA we have a possible way to do digital signatures based on the RSA function. (Whether these ways are good or bad ways of accomplishing the goal is another question, to be addressed later.) Thus, atomic primitives are simple building blocks that must be put together to yield protocols.

Good atomic primitives are rare, as are people who understand their workings. Certainly, an important effort in cryptography is to design new atomic primitives and cryptanalyze them and old ones. This, however, is not the part of cryptography I want to talk about. The reason is that the design (or discovery) of good atomic primitives is more an art than a science. On the other hand, I'd like to claim that the design of protocols can be made a science.

The question. We will view a cryptographer as an engine for turning atomic primitives into protocols. That is, we focus on protocol design under the assumption that good atomic primitives exist. Some examples of the kinds of questions we are interested in are these. What is the best way to encrypt a large text file using DES, assuming DES is secure? What is the best way to design a signature scheme using the RSA function, assuming the latter is one-way? How "secure" are known methods for these tasks? What do such questions even mean, and can we find a scientific framework in which to ask and answer them?

The problem. The problem with protocol design is that a poorly designed protocol can be insecure *even though the underlying atomic primitive is good.* An example is ECB (Electronic Code-Book) mode encryption with a block cipher. It is not a good encryption scheme because partial information about the plaintext leaks. Yet this is no fault of the underlying atomic primitive (typically DES). Rather, the atomic primitive was mis-used.

Indeed, lots of protocols are broken. Yet the good atomic primitives, like DES and RSA, have never been convincingly broken. We would like to build on the strength of atomic primitives in such a way that protocols can "inherit" this strength, not loose it!

2.2 Provable security: Reductions

The idea of provable security was introduced in the pioneering work of Goldwasser and Micali [26]. They developed it in the particular context of asymmetric encryption, but it soon spread to be applied to other tasks. (Of these, the most basic were pseudorandomness [16, 40, 25] and digital signatures [27]).

What is provable security? The paradigm is as follows. Take some goal, like achieving privacy via encryption. The first step is to make a formal adversarial model and *define* what it *means* for an encryption scheme to be secure. With this in hand, a particular scheme, based on some particular atomic primitive, can be analyzed from the point of view of meeting the definition. Eventually, one shows that the scheme "works" via a *reduction.* The reduction shows that the *only way* to defeat the protocol is to break the underlying atomic primitive. In other words, there is no need to directly cryptanalyze the protocol: if you were to find a weakness in it, you would have unearthed one in the underlying atomic primitive. So you might as well focus on the atomic primitive. And if we believe the latter is secure, we *know*, without further cryptanalysis of the protocol, that the protocol is secure.

An important sub-part of the last step is that in order to enable a reduction one must also have a formal notion of what is meant by the security of the underlying atomic primitive: what attacks, exactly, does it withstand? For example, we might assume RSA is a one-way function.

Here is another way of looking at what reductions do. When I give you a reduction from the one-wayness of RSA to the security of my protocol, I am giving you a transformation with the following property. Suppose you claim to be able to break my protocol. Let P be the program that does this. My transformation takes P and puts a simple "wrapper" around it, resulting in a protocol P'. This protocol P' provably breaks RSA. Conclusion? As long as we believe you can't break RSA, there could be no such program P. In other words, my protocol is secure.

Those familiar with the theory of NP-completeness will recognize that the basic idea of reductions is the same. When we provide a reduction from SAT to some problem we are saying our problem is hard unless SAT is easy; when we provide a reduction from RSA to our protocol, we are saying the latter is secure unless RSA is easy.

Here, I think, is a beautiful and powerful idea. Some of us by now are so used to it that we can forget how innovative it was. And for those not used to it, it can be hard to understand (or, perhaps, believe) at first hearing, perhaps because it delivers so much. Protocols designed this way truly have superior security guarantees.

Nomenclature. In some ways the term "provable security" is misleading. As the above indicates, what is probably the central step is providing a model and definition, which does not involve proving anything. And one does not "prove a scheme secure:" one provides a reduction of the security of the scheme to the security of some underlying atomic primitive. For that reason, I sometimes use the term "reductionist security" to refer to this genre of work.

The complexity-theoretic approach. The precise formalization of provable security can take many forms. The theoretical literature has chosen, for the most part, to develop it in a complexity theoretic framework where one talks about "polynomial time" adversaries and transformations, and "negligible success probabilities." This approach was convenient for a field striving to develop a technical idea of great depth. Complexity-based cryptography has been remarkably successful, coming up with definitions for many central cryptographic primitives, and constructions based on "minimal assumptions." For a brief introduction to this body of work, refer to the recent survey by Goldreich [24].

In practice? The potential for the idea of provable security to impact practice is large. Yet its actual impact had been disappointingly small, in the sense that these ideas were reflected almost not at all in protocols used in practice. Here are some possible reasons.

In practice, block ciphers are the most popular atomic primitive, especially for private key cryptography. Yet the provable security line of work (prior to the development of the practice-oriented variant) omitted any treatment of schemes based on block ciphers: only number-theoretic atomic primitives were deemed adequate as a basis for protocol design. In particular some of the world's most used protocols, such as CBC MAC [1] or encryption [32, 2], seemed to be viewed as outside the domain of provable security.[1]

The main generic disadvantage of the schemes delivered by the traditional provable security approach is that they are inefficient.[2] This is due in part to the complexity of the constructions. But it is also due in part to a reliance on inefficient atomic primitives. For example, a MAC would

[1] Luby and Rackoff [31] studied the Feistel structure behind DES, but what I am talking about is to look at protocols that use DES and ask about their security.

[2] Typically the gap relative to what is desirable in practice is enormous. In some cases it is small, but still seems enough to preclude usage.

be constructed out of a one-way function like RSA rather than out of a block cipher. This takes us back to the above.

Finally, some aspects of the complexity-theoretic approach unfortunately distanced provable security from practice. For example, practioners need numbers: how many cycles of adversary computation can the scheme withstand, how many bits is the security parameter? These are only loosely captured by "polynomials" or "negligible probabilities." To make provable security useful, reductions and security analyses must be concrete. Theoreticians will say, correctly, that this information can be obtained by looking at their proofs. But this view obscures the importance of working on improving the security of reductions.[3]

Practice-oriented provable security attempts to remedy this by appropriate paradigm shifts.

3 Practice-oriented provable security

Practice-oriented provable security as I discuss it was introduced in a set of papers authored by myself and Phil Rogaway [8, 7, 6]. We preserve and focus on the two central ideas of the provable security approach: the introduction of *notions*, or *definitions* that enable us to think about protocols and atomic primitives in a systematic way, and the idea of doing reductions. But we modify the viewpoints, models, and problems treated. Here are some elements of the approach and work to date.

3.1 Using block ciphers

Block ciphers like the DES are the most ubiquitous tool in practical cryptographic protocol design. However, as indicated above, traditionally nothing was proved about protocols that use them. An important element of our line of work is to integrate block ciphers into the fabric of provable security. On the one hand we analyze existing schemes that use block ciphers to assess how well they meet strong, formal notions of security; on the other hand we design new schemes based on block ciphers and show they meet such notions. In the first category are our analyses of the CBC MAC [7] and analyses of various modes of operation of a block cipher [5]. In the second category are constructions like the XOR MAC [6] or the cascade [4].

Key to these results (and perhaps more important than any individual result) is that we treat block ciphers systematically by formally modeling them in some way. Specifically, the suggestion of [7], followed in the other works, was to model a block cipher as a *finite pseudorandom function* (FPRF) family. (The fundamental notion of a pseudorandom function family is due to Goldreich, Goldwasser and Micali [25]. The finite variant was introduced in [7].) Roughly, we are assuming that as long as you don't know the underlying key, the input-output behavior of a block cipher closely resembles that of a random function.

Thus, the theorems in the mentioned papers say that a scheme (eg. CBC MAC) is secure unless one can detect some deviation from random behavior in the underlying block cipher. Underlying this claim is a reduction, as usual in the provable security approach, showing how to break the cipher given any way to break the scheme based on it.

The idea of treating block ciphers as pseudorandom functions provides a fresh way of looking at block ciphers from both the design and usage perspective. On the one hand, this view can form the basis for analyses of many other block cipher based schemes. On the other hand, we suggest

[3] This is not to say concrete security has always been ignored. One person who from the beginning has systematically addressed concrete security in his works is Claus Schnorr. See any of his papers involving cryptographic reductions.

it be a design criterion for future block ciphers (a view that new efforts such as AES do seem to support) and that existing ciphers should be cryptanalyzed to see how well they meet this goal.

3.2 Concrete security

Practice oriented provable security attempts to explicitly capture the inherently *quantitative* nature of security, via a *concrete* or *exact* treatment of security. Rather than prove asymptotic results about the infeasability of breaking a protocol in polynomial time, we present and prove "exact" or "concrete" reductions. Our results have the form: "If DES withstands an attack in which the adversary gets to see 2^{36} plaintext-ciphertext pairs, then our protocol is secure against an adversary who can run for t steps, for the following value of t." This enables a protocol designer to know exactly how much security he/she gets. And it brings a new dimension to protocols: rather than just being secure or non-secure, one can be "more" secure than another.

For example, the theorem of [7] characterizing the security of the CBC MAC says that an adversary who runs for time t and sees q correctly MACed messages has chance at most $\epsilon + (3q^2n^2 + 1)/2^l$ of correctly forging the MAC of a new message, where l is the block length of the underlying cipher, n is the number of blocks in any message to which the MAC applies, and ϵ captures the security of the cipher, specifically being the chance of detecting a deviation of the cipher from random behavior in time $t + O(nql)$ given nq input-output examples of the cipher under the same key. (This ϵ is of course a function of the key length of the underlying cipher, but the latter does not need to appear explicitly.) Thus, a user sees exactly how the chance of forgery increases with the number of messages MACed.

Another aspect of the concrete security treatment is to try to preserve as much as possible of the strength of the underlying atomic primitive in transforming it to the protocol. This means we aim for reductions as *strong* as possible. This is important because reduction strength translates directly to protocol efficiency in practice. A weak reduction means that to get the same level of security in our protocol we must use larger keys for the underlying atomic primitive, and this means slower protocols. If the reduction is strong, shorter keys will suffice and the protocol is more efficient. Reduction quality plays a significant role in [7, 6, 10, 12, 4, 5] all of which achieve tight or close to tight reductions.

We found that *improving* the concrete security was a rich and rewarding line of work, and thinking about it greatly increases understanding of the problem.

In [5] we also concern ourselves with how different formalizations of a notion (in this case, secure encryption) are affected when concrete security is an issue.

3.3 Security versus attacks

Practitioners typically think only about concrete attacks; theoreticians ignore them, since they prove the security. Under the practice oriented provable security approach, attacks and security emerge as opposite sides of the same coin, and complement each other. Attacks measure the degree of insecurity; our quantitative bounds measure the degree of security. When the two meet, we have completely characterized the security of the protocol.

For example, the security of the CBC MAC shown in [7] is the flip-side of attacks like those of Preneel and Van Oorschot [37]. (The latter say that the CBC MAC can be broken once $2^{l/2}$ messages have been MACed, where l is the block length of the underlying cipher. We say, roughly, that it *can't* be broken when *fewer* than this many messages are MACed.) Thus the results of [7, 37] complement each other very well. Yet, the literature on these subjects does not reflect this duality appropriately.

We found that even when proofs are provided, much is to be gained by finding the best possible attacks. We find *new kinds* of attacks, which break the system as measured by our more stringent notions of security: an encryption scheme is broken of you can tell whether the message encrypted was 0 or 1, not just if you find the key. This is actually important in practice. Meanwhile, these attacks provide, effectively, the lower bounds to our concrete security analyses, telling us whether the proven security is optimal or not. Publications in which we assess the optimality of our reductions via attacks include [6, 4, 5].

3.4 The random oracle model

Sometimes, using pseudorandom function families or one-way functions alone, we are not able to find schemes efficient enough for practice. This is true for example in the case of public key encryption or signatures. In such cases, we turn to the random oracle paradigm.

The random oracle paradigm was introduced in [9] as a bridge between theory and practice. The idea is a simple one: namely, provide all parties —good and bad alike— with access to a (public) function h; prove correct a protocol assuming h is truly random, ie. a random oracle; later, in practice, set h to some specific function derived in some way from a standard cryptographic hash function like SHA-1 [33] or RIPEMD-160 [21].

We used the random oracle paradigm most importantly to design OAEP [10] and PSS [12]. These are schemes for (public key) encryption and signature (respectively), the most popular versions of which use RSA as the underlying primitive. (Both OAEP and PSS are, more accurately, padding or formatting mechanisms which are applied to a message before the appropriate RSA operation is applied.) They are as efficient as previously used or standardized schemes, but, unlike them, provably achieve strong notions of security in the random oracle model, assuming RSA is a one-way function.

RSA Corporation publishes a standard for RSA based encryption called PKCS#1. (It is a widely used standard, implemented in Netscape and other browsers, and used in SSL.) Much publicity was given recently to a chosen-ciphertext attack on PKCS#1 that was discovered by Bleichenbacher [15]. RSA Corporation has now revised the protocol, adopting OAEP in PKCS#1 v2.0 [38]. The rationale for that move is that our protocol had been *proven* to resist chosen-ciphertext attacks (indeed Bleichenbacher's attacks do not work on OAEP, even though at the time of the design of OAEP we had not thought of these specific attacks), and furthermore OAEP is just as practical as the original PKCS#1 protocol.

OAEP is also included in SET, the electronic payment protocol of MasterCard and Visa, where it is used to encrypt credit card numbers. Both OAEP and PSS are being proposed for the IEEE P1363 standard.

What's the point of the random oracle paradigm, and what does it buy you? It buys efficiency, plus, we claim, security guarantees which, although not at the same level as those of the standard provable security approach, are arguably superior to those provided by totally ad hoc protocol design. The last point merits some more discussion.

The random oracle paradigm should be used with care and understanding. It is important to neither over-estimate nor under-estimate what this paradigm buys you in terms of security guarantees. First, one must be clear that this is not standard provable security. The function h that we actually use in the final scheme is not random. Thus the question is: what has it bought us to have done the proof in the first place?

The overly skeptical might say the answer is "nothing." This is not quite true. Here is one way to see what it buys. In practice, attacks on schemes involving a SHA-1 derived h and number theory will often *themselves treat h as random.* We call such attacks *generic attacks.* In other words, cryptanalysis of these "mixed" schemes is usually done by assuming h is random. But then the proofs apply, and indeed show that such generic attacks will fail unless the underlying number-theoretic problems are easy. In other words, the analysis at least provably excludes a certain common class of attacks, namely generic ones.

It is important to choose carefully the instantiating function h. The intuition stated in [9] is that the resulting scheme is secure as long as the scheme and the hash function are sufficiently "independent," meaning the scheme does not itself refer to the hash function in some way. This is a fuzzy guideline which we hope to understand better with time.

An important step in our understanding of the random oracle model was taken by Canetti, Goldreich and Halevi [19]. They indicate that there exist schemes secure in the random oracle model but insecure under any instantiation in which we substitute a function from a small family of efficiently computable functions. Their examples however are somewhat contrived, and this kind of situation does not arise with any of the "real" constructions in the literature.

In comparison with totally ad hoc design, a proof in the random oracle model has the benefit of viewing the scheme with regard to its meeting a strong and formal notion of security, even if this is assuming some underlying primitive is very strong. This is better than not formally modeling the security of the scheme in any way. This explains why the random oracle model is viewed in [9] as a "bridge between theory and practice."

Since we introduced this model, it has been used in other places, for example in the design and analysis of signature schemes [35, 36, 34], hash functions [13] and threshold encryption schemes [23].

3.5 New notions: session key distribution

"Entity authentication" is the process by which a party gains confidence in the identity of a communication partner. It is usually coupled with the distribution of a "session key." These are arguably the most basic problems for secure distributed computation— without a correct solution there can be no meaningful access control or accountability; there cannot even be reliable distribution of work across network resources. Despite a long history and a large literature, this problem rested on no meaningful formal foundation. This is more than an academic complaint: it is an area in which an informal approach has often lead to work which has subsequently been found to be wrong, and in some cases the flaws have taken years to discover.

In [8] we address the two party setting of the problem. It achieves provable security by providing a model, definitions, protocols, and proofs of correctness for these protocols under standard assumptions.

The three party case of this problem may be the most well-known. It was first addressed by Needham and Schroeder in 1978. Its most popular incarnation is the *Kerberos* system. However this system, and existing solutions, suffer from the same problems discussed above. In [11] we provide provably secure protocols for the three party session key distribution problem.

All our protocols are efficient and practical, viable alternatives to current systems. Some have been implemented. Our models have been used to study related key distribution problems, for example in [39].

4 What provable security is and isn't

Now that provable security is moving into practice, there are many people who although not trained as theoreticians, or even deeply interested in the details of research, need to take decisions involving claims about provable security. The kinds of things they need to know are: exactly what provable security provides and doesn't provide; how to compare different provably secure schemes; how to validate a claim of provable security. So I would like to discuss some of these points here.

4.1 On limitations

The above has explained what provable security provides, but it is also important to understand its limitations. The first of these is in the model considered. One must ask what kinds of attacks the model encompasses. In particular, there are classes of attacks that do not fall into the normal fabric of provable security; these include timing attacks [29], differential fault analysis [18, 14], and differential power analysis [30]. (That is, models used in provable security do not encompass these attacks. One should note that this does not mean specific proven secure schemes fall to these attacks. It just means we do not claim to have *proven* that they do not fall to these attacks. In fact if you look at specific schemes, some fall to these attacks, others don't.) But it is worth investigating an extension of provable security that does include these attacks, a line of research suggested by Dan Boneh.

Even when using a proven secure scheme, security can compromised in a number of ways. Sometimes, requirements may have been overlooked: we proved security, but for the wrong problem or in the wrong model. Or, the protocol may be used incorrectly. For example, a setting such as key exchange might require a public key encryption scheme that is secure against chosen-ciphertext attack. It does little good to use a proven secure scheme that is only proven secure against chosen-plaintext attack. This is a question of understanding what requirements a higher level protocol imposes on the lower level primitive.

Or software may be buggy. If you implement the scheme incorrectly, obviously all bets are off. Similarly the environment may be improperly administered leading to loss of passwords or keys. There may be insider attacks. And so on.

4.2 On assumptions

Proven security does not mean that attacks (of the kind modeled) are unconditionally guaranteed to fail. Remember that a scheme is proven secure given some *assumption*. For example, we may have an encryption scheme proven to resist chosen-plaintext attacks as long as the problem of factoring a number product of two primes is computationally infeasible. Or, as in examples above, that a message authentication scheme is secure as long as the underlying cipher behaves like a pseudorandom function family.

If these assumptions fail, of course the proof becomes worthless. (One should note that failure of an assumption does not necessarily lead to attacks on the scheme. It just means that the proof of security is no longer useful.) This means that a proof of security is worth more when the assumption is weaker, ie. less likely to fail. An important parameter of a proof of security is thus the underlying assumption: the weaker the better. In particular this becomes something to consider in comparing schemes. If you have a choice between two schemes, you will of course take into account many things, such as performance, ease of implementation, exportability and so on. But on the security front, if both schemes are proven secure, the one making weaker assumptions is preferable.

Comparing provable guarantees has both a qualitative and quantitative aspect. Even when two schemes are based on the same assumptions, one may have better concrete security. (We discussed the concrete security approach in Section 3.2.) This means that there is less loss of security in the translation from the problem of the assumption to the scheme. An example is the signature schemes FDH and PSS [12]– both are proven secure in a random oracle model assuming RSA is one-way, but the reduction for PSS is tight and that for FDH is not, so the quantitative guarantee of PSS is better.

How does one compare assumptions to see which is weaker? Unfortunately it is not always possible. Indeed, in the bulk of cases, we do not know how to compare the assumptions underlying various proofs of security. But it is still important to know about this and know when they are comparable and when not.

To illustrate these issues let us look at public key encryption secure against chosen-ciphertext attack. We discussed (RSA based) OAEP [10] above: it is proven secure in the random oracle model assuming the RSA function is one-way.

Dolev, Dwork and Naor [22] had designed a scheme that resists chosen-ciphertext attack many years prior to this. Lets call this the DDN scheme. The security of the DDN scheme can be proven assuming RSA is a one-way function. Notice that this assumption is weaker than the one underlying OAEP, since OAEP assumes in addition that we have a hash function that behaves like a random oracle. As a consequence we can say that the provable security guarantee provided in the DDN scheme is superior to that of OAEP. In this case, a security comparison was possible.

More recently, Cramer and Shoup [20] introduced a new proven-secure encryption scheme which we call the CS scheme. Unlike the schemes we have been discussing up to now, it is not RSA based: it assumes the hardness of a certain version of the Diffie-Hellman problem. How does the security of the CS scheme compare to that of OAEP? That is more difficult to assess. The CS scheme does not use the random-oracle paradigm, which is a plus. But it assumes the hardness of the so-called Decisional Diffie Hellman problem. (See [17] for a nice discussion of this problem.) This is a strong assumption, and relatively new and un-studied one compared to the assumption that RSA is one-way. (It would be much more surprising if the RSA assumption failed than if the Decisional Diffie-Hellman assumption failed.) In particular, we do not know how this assumption compares to the assumptions underlying OAEP. So, while the fact that the CS scheme avoids random oracles is a point in its favor, it is not really possible to say that one of these schemes has better security guarantees than the other in practice, because the assumptions are incomparable.

If one had to choose a scheme in practice one would of course also consider cost. OAEP has the same cost as heuristic RSA schemes. The DDN scheme is many orders of magnitude more expensive than any practical scheme, since it involves multiple signatures and zero-knowledge proofs, and thus is likely to be ruled out. The CS scheme is much cheaper than the DDN scheme, but still more expensive than OAEP. (Encryption in OAEP is only a few multiplications if a small RSA exponent is used; while in the CS scheme it is a few exponentiations. Decryption in the CS scheme is about five times as costly as in OAEP.) In some applications, this kind of increase may be tolerable; in others not. There is no unique answer.

4.3 Proofs and definitions

Faced with a choice of protocols claiming to be provably secure, we discussed above some issues involved in comparing them. Another should be mentioned: verification of the claims. A scheme isn't provably secure just because it is claimed to be so. One should check that proper formal definitions of security have been provided so as to know *what* is being proved. One should be able to at least cursorily verify the claims. How? Remember that a reduction consists of an algorithm that is an attacker for the problem we are assuming hard, using as a subroutine an attacker for the scheme. Look at the least for a description of such an algorithm.

5 Going on

The above has discussed provable security and its practice oriented variant in a general way. Next I would like to illustrate the ideas by looking in more depth at a central problem: encryption. The goal is to motivate the need for strong and formal notions of security and then show how to to adapt the seminal notions of [26] (given in the asymmetric setting) to the symmetric setting. With concrete security definitions in hand, we will turn to analyzing popular encryption modes like CBC or CTR and gauge their merits. We want to answer questions like: are the secure? Which is "better"?

I did this in my talk, for the most part following [5], and refer the reader there for this materiel. Some day, I hope to extend this article by the inclusion of this and other materiel.

Acknowledgments

This presentation is based on ideas developed jointly with Phillip Rogaway.

This work is supported in part by NSF CAREER Award CCR-9624439 and a 1996 Packard Foundation Fellowship in Science and Engineering.

References[4]

[1] ANSI X9.9, "American National Standard for Financial Institution Message Authentication (Wholesale)," American Bankers Association, 1981. Revised 1986.

[2] ANSI X3.106, "American National Standard for Information Systems – Data Encryption Algorithm – Modes of Operation," American National Standards Institute, 1983.

[3] M. Bellare, R. Canetti and H. Krawczyk, "Keying hash functions for message authentication," *Advances in Cryptology – Crypto* 96 *Proceedings*, Lecture Notes in Computer Science Vol. 1109, N. Koblitz ed., Springer-Verlag, 1996.

[4] M. Bellare, R. Canetti and H. Krawczyk, "Psuedorandom functions revisited: The cascade construction and its concrete security," *Proceedings of the* 37th *Symposium on Foundations of Computer Science*, IEEE, 1996.

[5] M. Bellare, A. Desai, E. Jokipii and P. Rogaway, "A concrete security treatment of symmetric encryption," *Proceedings of the* 38th *Symposium on Foundations of Computer Science*, IEEE, 1997.

[6] M. Bellare, R. Guérin and P. Rogaway, "XOR MACs: New methods for message authentication using finite pseudorandom functions," *Advances in Cryptology – Crypto* 95 *Proceedings*, Lecture Notes in Computer Science Vol. 963, D. Coppersmith ed., Springer-Verlag, 1995.

[7] M. Bellare, J. Kilian and P. Rogaway, "The security of cipher block chaining," *Advances in Cryptology – Crypto* 94 *Proceedings*, Lecture Notes in Computer Science Vol. 839, Y. Desmedt ed., Springer-Verlag, 1994.

[8] M. Bellare and P. Rogaway, "Entity authentication and key distribution," *Advances in Cryptology – Crypto* 93 *Proceedings*, Lecture Notes in Computer Science Vol. 773, D. Stinson ed., Springer-Verlag, 1993.

[4]The best place to obtain any of my papers listed below is via `http://www-cse.ucsd.edu/users/mihir`. Here you will find the full, and most recent, versions.

[9] M. Bellare and P. Rogaway, “Random oracles are practical: a paradigm for designing efficient protocols,” *Proceedings of the First Annual Conference on Computer and Communications Security*, ACM, 1993.

[10] M. Bellare and P. Rogaway, “Optimal asymmetric encryption – How to encrypt with RSA,” *Advances in Cryptology – Eurocrypt* 95 *Proceedings*, Lecture Notes in Computer Science Vol. 921, L. Guillou and J. Quisquater ed., Springer-Verlag, 1995.

[11] M. Bellare and P. Rogaway, “Provably secure session key distribution– the three party case,” *Proceedings of the* 27th *Annual Symposium on the Theory of Computing*, ACM, 1995.

[12] M. Bellare and P. Rogaway, “The exact security of digital signatures: How to sign with RSA and Rabin,” *Advances in Cryptology – Eurocrypt* 96 *Proceedings*, Lecture Notes in Computer Science Vol. 1070, U. Maurer ed., Springer-Verlag, 1996.

[13] M. Bellare and D. Micciancio, “A new paradigm for collision-free hashing: Incrementality at reduced cost,” *Advances in Cryptology – Eurocrypt* 97 *Proceedings*, Lecture Notes in Computer Science Vol. 1233, W. Fumy ed., Springer-Verlag, 1997.

[14] E. Biham and A. Shamir, “Differential fault analysis of secret key cryptosystems,” *Advances in Cryptology – Crypto* 97 *Proceedings*, Lecture Notes in Computer Science Vol. 1294, B. Kaliski ed., Springer-Verlag, 1997.

[15] D. Bleichenbacher, A chosen ciphertext attack against protocols based on the RSA encryption standard PKCS #1, *Advances in Cryptology – Crypto* 98 *Proceedings*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

[16] M. Blum and S. Micali, “How to generate cryptographically strong sequences of pseudo-random bits,” *SIAM Journal on Computing*, Vol. 13, No. 4, November 1984, pp. 850–864.

[17] D. Boneh, “The decision Diffie-Hellman problem,” Invited paper for the Third Algorithmic Number Theory Symposium (ANTS), Lecture Notes in Computer Science Vol. 1423, Springer-Verlag, 1998. Available at `http://theory.stanford.edu/~dabo/abstracts/DDH.html`.

[18] D. Boneh, R. DeMillo and R. Lipton, “On the importance of checking cryptographic protocols for faults,” *Advances in Cryptology – Eurocrypt* 97 *Proceedings*, Lecture Notes in Computer Science Vol. 1233, W. Fumy ed., Springer-Verlag, 1997.

[19] R. Canetti, O. Goldreich, and S. Halevi, “The random oracle methodology, revisited,” *Proceedings of the* 30th *Annual Symposium on the Theory of Computing*, ACM, 1998.

[20] R. Cramer and V. Shoup, “A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack,” *Advances in Cryptology – Crypto* 98 *Proceedings*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

[21] H. Dobbertin, A. Bosselaers and B. Preneel, “RIPEMD-160: A strengthened version of RIPEMD,” *Fast Software Encryption*, Lecture Notes in Computer Science 1039, D. Gollmann, ed., Springer-Verlag, 1996.

[22] D. Dolev, C. Dwork and M. Naor, “Non-malleable cryptography,” *Proceedings of the* 23rd *Annual Symposium on the Theory of Computing*, ACM, 1991.

[23] R. Gennaro and V. Shoup, “Securing threshold cryptosystems against chosen-ciphertext attack,” *Advances in Cryptology – Eurocrypt* 98 *Proceedings*, Lecture Notes in Computer Science Vol. 1403, K. Nyberg ed., Springer-Verlag, 1998.

[24] O. Goldreich, "On the foundations of modern cryptography," *Advances in Cryptology – Crypto* 97 *Proceedings*, Lecture Notes in Computer Science Vol. 1294, B. Kaliski ed., Springer-Verlag, 1997.

[25] O. Goldreich, S. Goldwasser and S. Micali, "How to construct random functions," *Journal of the ACM,* Vol. 33, No. 4, October 1986, pp. 792–807.

[26] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. of Computer and System Sciences*, Vol. 28, April 1984, pp. 270–299.

[27] S. Goldwasser, S. Micali and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM Journal of Computing,* Vol. 17, No. 2, April 1988, pp. 281–308.

[28] ISO 8372, "Information processing – Modes of operation for a 64-bit block cipher algorithm," International Organization for Standardization, Geneva, Switzerland, 1987.

[29] P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," *Advances in Cryptology – Crypto* 96 *Proceedings*, Lecture Notes in Computer Science Vol. 1109, N. Koblitz ed., Springer-Verlag, 1996.

[30] P. Kocher, "Differential power analysis," `http://www.cryptography.com/dpa/index.html`.

[31] M. Luby and C. Rackoff, "How to construct pseudorandom permutations from pseudorandom functions," *SIAM J. Computation,* Vol. 17, No. 2, April 1988.

[32] National Bureau of Standards, NBS FIPS PUB 81, "DES modes of operation," U.S Department of Commerce, 1980.

[33] National Institute of Standards, FIPS 180-1, "Secure hash standard," April 1995.

[34] K. Ohta and T. Okamoto, "On concrete security treatment of signatures derived from identification," *Advances in Cryptology – Crypto* 98 *Proceedings*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

[35] D. Pointcheval and J. Stern, "Security proofs for signatures," *Advances in Cryptology – Eurocrypt* 96 *Proceedings*, Lecture Notes in Computer Science Vol. 1070, U. Maurer ed., Springer-Verlag, 1996.

[36] D. Pointcheval and J. Stern, "Provably secure blind signature schemes," *Advances in Cryptology – ASIACRYPT* 96 *Proceedings*, Lecture Notes in Computer Science Vol. 1163, M. Y. Rhee and K. Kim ed., Springer-Verlag, 1996.

[37] B. Preneel and P. van Oorschot, "MD-x MAC and building fast MACs from hash functions," *Advances in Cryptology – Crypto* 95 *Proceedings*, Lecture Notes in Computer Science Vol. 963, D. Coppersmith ed., Springer-Verlag, 1995.

[38] RSA Laboratoris, "PKCS," `http://www.rsa.com/rsalabs/pubs/PKCS/`.

[39] V. Shoup and A. Rubin, "Session key distribution using smart cards," *Advances in Cryptology – Eurocrypt* 96 *Proceedings*, Lecture Notes in Computer Science Vol. 1070, U. Maurer ed., Springer-Verlag, 1996.

[40] A. C. Yao, "Theory and applications of trapdoor functions," *Proceedings of the* 23rd *Symposium on Foundations of Computer Science*, IEEE, 1982.