Practice-Oriented Provable Security and the Social Construction of Cryptography

Phillip Rogaway*

May 22, 2009†

Abstract

Traditionally, "provable security" was tied in the minds cryptographers to public-key cryptography, asymptotic analyses, number-theoretic primitives, and proof-of-concept designs. In this essay I survey some of the work that I have done (much of it joint with Mihir Bellare) that has helped to erode these associations. I will use the story of practice-oriented provable security as the backdrop with which to make the case for what might be called a "social constructionist" view of our field. This view entails the claim that the body of work our community has produced is less the inevitable consequence of what we aim to study than the contingent consequence of sensibilities and assumptions within our disciplinary culture.

1 Introduction

This essay was written to accompany an invited talk at Eurocrypt 2009. This is not a traditional academic paper; it is only an essay.

The plan. My aim is to accomplish a couple of things. First, I have never provided a general exposition of the research program that I initiated some 15+ years ago with Mihir Bellare, what we rather un-creatively termed *practice-oriented provable security*. So I'd like to do just that. I see practice-oriented provable security as a distinct and recognizable branch of contemporary cryptology, derived from, yet different from, classical provable-security cryptography. But I have no idea if others conceptualize this line of work as such, or if they even associate it to Mihir and me. So telling this story is the first thing I want to do. Second, I would like to tell my story from an atypically *sociological* perspective. My idea is to try to use practice-oriented provable security as the setting with which to illustrate the extent to which our field is *socially constructed*. This weighty-sounding term used to be quite popular (many would say *too* popular) among those in the social sciences. But the term isn't common in *our* community, so I think I better tell you right off what I mean by it.

Social construction. What does it mean when we say that some thing, C, is *socially constructed*? C can be almost anything, like *gender* or *quarks* or *serial killers*, but you might like to think of C as *cryptography*, or *the classical approach for doing provable-security cryptography*. Well, saying that C is socially constructed emphasizes that C (or the idea of C, or the practice of C) need not be as it is. It is not determined by the nature of mathematical truth or physical reality. It wasn't inevitable. Rather, C exists in its present form as the *contingent consequence* of social or historical forces. Not necessarily broad, society-wide forces—the relevant social forces could be much more local. In particular, I believe that the forces most relevant to the shaping of cryptography are those within the *disciplinary culture* of our community. When I say that cryptography is constructed, I am always referring to its construction by the crypto community, contingent on our community's assumptions and beliefs.

*Dept. of Computer Science, One Shields Ave., University of California, Davis, California, 95616, USA. E-mail: rogaway@cs.ucdavis.edu WWW: www.cs.ucdavis.edu/~rogaway/

†First public version: May 6, 2009. Subsequent revisions have been minor.

Constructionism is not the only view of why an area of scholarship is as it is. A different and older explanation for why a scientific field is the way it is goes by the name *scientific realism.*[1] C is the way it is because C captures the truth—or at least a good and ever-improving approximation to the truth. There wasn't a lot of choice about how to do C; its characteristics flow from the nature of the thing. It was pretty much inevitable that C would be as C is now, assuming one is going to have a successful theory at all.

Physicists arrived at the notion of quarks, for example, because quarks are real. They were *discovered* by scientists who were following a particular methodology. Nobody "invented" quarks, contrary to the provocative title of Andrew Pickering's book, *Constructing Quarks.* Another community of physicists—maybe non-humans working on a planet far from here—they would come up with a similar notion in the evolution of their ideas on physics.

Philosophically, most scientists subscribe to realist views. But, ever since Thomas Kuhn's hugely influential *The Structure of Scientific Revolutions* (1962), those who study the history and sociology of science have leaned towards at least somewhat constructionist points of view.

The disagreement between the realist and constructionist camps has been pretty heated—it has even been termed, perhaps a little over-dramatically, the *science wars*. The wars have been fought out in the battlefield of books and papers for the last two or three decades, while most of us cryptographers were blissfully unaware. Constructionism is often seen as the centerpiece of *post-modernism*, in which case the realist/constructionist disagreement is at the heart of the modernist/post-modernist debate.

While the case for constructionism in high-energy physics might be a bit of a stretch, the case for constructionism in technology is an awful lot clearer. What is referred to as the *Social Construction of Technology*, SCOT, hasn't been nearly so controversial. Technology, after all, is the business of the synthetic, and our artifacts, at least, aren't just metaphorically constructed, they are *literally* constructed. Constructionist accounts of how technologies emerge have had little difficulty documenting how social forces or inessential historical choices have shaped them.

Cryptography is a bit of a problem child here, because parts of it seem to rest squarely in engineering and are obviously constructed—nobody would maintain that Rijndael, say, is anything other than construction—but other parts, potentially including all of provable security, seem, possibly, to be better explained by scientific realism.

It is hard to ascertain the extent to which those in our community have realist verses constructionist points of view. Such matters are only vaguely implied by technical work. One case where an author's views are made clear is in Oded Goldreich's essay *On Post-Modern Cryptography* (2006). The views there are decidedly realist. For example, the essay asserts—in italics, without equivocation—that *cryptographic research is indeed part of science*. Oded takes this to be an empirical fact (if not a normative claim as well).

In contrast, I would claim that cryptography, even in its most pure and scientific persona, is quite strongly constructed. My talk will implicitly argue this thesis in the telling of its stories.

2 The Backdrop

Provable security. Any exposition on the development of practice-oriented provable security surely must begin with provable security itself. Provable security begins right around 1982, most especially with the landmark paper of Shafi Goldwasser and Silvio Micali, *Probabilistic Encryption*. The authors were only graduate students at the time, just finishing their PhDs at UC Berkeley. Yet what they did with this paper was to lay the foundations for the entire definition-based, reduction-based approach to doing cryptography—what we are calling *provable security.*

The provable-security approach begins by identifying the cryptographic problem of interest, an often ignored but crucial first step. Next, one gives a precise *definition* for the problem at hand. This entails identifying *what* the adversary can do and *when* it is deemed successful. A *protocol* is then given for problem, that protocol depending on some other, hopefully-simpler protocol for solving some other, hopefully-more-basic problem that has likewise been defined. Evidence of security for the high-level protocol takes the form of a *reduction*, the reduction showing how to transform an adversary attacking the high-level goal into one for attacking the low-level goal.

[1] A better term might be *nonconstructionism*, as terms like *realism* and *scientific realism* have specific philosophical meanings that are not denied by construction as I use the term.

Provable security has had a profound impact on our field. In the proceedings of today's conference, about 27 years after its beginnings, approximately one half of the papers fall within this intellectual tradition. And even this fairly representative count—half of all papers in cryptography—may underestimate the depth of influence of this paradigmatic shift, as major technological changes are invariably ecological, impacting even things quite far removed.

MIT. I myself came into cryptography a few years after the beginning of provable security. I began graduate school in 1985, in the Theory of Computation group at MIT. It was an extraordinarily exciting time to be at MIT studying theoretical computer science and cryptography. To begin with, it was a strangely young and vibrant place. I remember the first class I attended, taught by David Shmoys. He looked about 18 years old. I thought it strange that MIT should let a freshman teach a course. Afterwards, I went to a seminar in which Mike Sipser introduced the day's speaker. Mike looked plausibly old enough to drive a car, but I wasn't really sure.

Some of the students were strangely young as well. I learned that my officemate, Miller Maley, had begun graduate school at age 16. Ravi Boppana was his senior, but not by much. And Ravi seemed not just smart, but wise as well. In general, the place seemed packed with highly impressive people. These included Mihir Bellare, and we became friends. We tried to solve some problems together in those early days, but only seemed to work on things too hard for either of us to do.

The theory group had three cryptographers: Shafi Goldwasser, Silvio Micali, and Ron Rivest. Mike Sipser too, though he was more on the complexity-theory side. Many of the papers that flowed from these people were absolutely visionary, paradigmatic creations. One has only to re-read some of their fantastical titles to re-live a bit of the other-worldliness of the time. How to play mental poker (1982). Paradoxical signatures (1984). Knowledge complexity and interactive proofs (1985). Proofs that yield nothing but their validity (1986). How to play *any* mental game (1987). The MIT cryptographers seemed to live in a world of unbridled imagination. And not *just* the brilliant faculty—the students, too. Joe Kilian would walk up to me and describe the motivation for some wonderfully crazy problem he was thinking about. More often than not, *space aliens* were deeply involved.

Within the Theory of Computation group, we lived our lives by a calendar based on conference deadlines. As the STOC or FOCS deadline approached, the frenzy would rise to fever pitch. At the last conceivable moment, about a dozen copies of each paper would be put into these amazing white envelope that could not be torn. A box of these, sometimes more, would be rushed by car to the Boston airport, to be put on the last plane going out to wherever the papers were due.

As I saw it (and no doubt there was more to see than what I saw), cryptography at MIT did not much countenance pragmatic concerns. The field was, unquestionably, a branch of *theoretical computer science*. The culture of STOC and FOCS was the culture we lived. While a word or two might be uttered in a paper to play-up some far-fetched application, it would be done with a wink of the eye; minimally, practical considerations would have to wait for some distant, less ecstatic, day. My wonderful advisor, Silvio Micali, would wax philosophically about how some definition or proof should go, and it is no exaggeration to claim that, as I saw it, philosophy and beauty had unquestioned primacy over utility in determining what was something good to do.

The reason I have spent time describing the culture of the Theory Group at MIT is that this culture profoundly shaped the early character of our field. And this character lives on. Through mechanisms both obvious and hidden, a discipline carries its history with it, and lives the continuation of that history, no less than does a man.

IBM. After MIT, Mihir took a position at IBM Watson, which actually meant Hawthorne, while I took a position at an IBM development lab. I hadn't planned to do any such thing—I was really too contemptuous of practice to even consider it—but, one day, some crazy people from IBM/Austin, of all places, starting bugging me that I should meet them. I told them I wasn't interested, but this fellow named Bob Blakely insisted on flying out to talk. Bob turned out to be the lead security architect at IBM, although it would take IBM some time to figure that out and give him a commensurate job title. Bob was very persuasive, and I finally decided that a 2–3 year hiatus in industry might work out well. During such a time, I could continue to do *actual* research, but I could also, on the side, do some community-service work, bringing a bit

of the Science of cryptography to the masses of poorly informed security practitioners. I'd inject a healthy dose of crypto-theory into the sad backwaters of crypto-practice.

Linear development. Describing that now, with just a bit of exaggeration, it seems an embarrassingly naïve point of view. Not only naïve, but arrogant, too. In my defense, I would like to offer up that lots of other people, too, seem to have had a roughly similar point of view. What we assumed is that *science* provides the raw material for technological change. Concretely, the theory of cryptography would provide the needed paradigms, techniques, and points of view. Then, these ideas would get selectively picked up, refined, and concretely embodied. They would become objects of material culture—program products, for example, that IBM could sell. Finally, society would happily reap the benefits of our embodied ideas.

The Science-feeds-Technology-feeds-Society view is what Ian Barbour calls the *linear-development model.* The idea is nicely captured in the slogan of the 1933 World Fair:

Science finds — Industry applies — Man conforms

The fair was held in Chicago, during the very year that Hitler came to power in Germany. Redolent of industrial-revolution and even fascist overtones, I find the slogan positively creepy. Yet the idea that it so succinctly captures remains a powerful force. I would claim that the notion has taken on near-mythic proportions, most especially in the USA, providing a kind of implicitly assumed framework in which many scientists see our professional lives.

Two hats. Well, my more senior colleagues at IBM had, of course, long before figured out that, whatever it was we were doing for IBM, it was *not* the injection of some already-worked-out theory into the compliant body of cryptographic practice. What Mihir and I actually observed in how people worked was something completely different. It was what we came to call, in nearly daily phone conversations between the two of us, the *two-hat approach.* Here's how it works.

- You could wear your *theory hat*, in which case you'd try to write a nice paper, preferably one for STOC, FOCS, or CRYPTO; or,
- You could don your *practice hat*, which, fortunately, nobody felt compelled to do all that often. There you'd use your intuitions, intelligence, and predisposition to skepticism to design, attack, or refine some real-world scheme that, against all odds, somehow came your way.

These two modes didn't much interact or interfere. They're just different ways to work.

Whatever it may sound like, I don't want to sound too critical of the two-hat approach. I believe that all of us wear lots of hats—more than two—and that that's usually a good thing. One can do wonderful things wearing each hat one wears. But Mihir and I just didn't feel very comfortable with this two-hat idea. The problem occurs when wearing the practice hat. Then, you're effectively asked to abandon not just your theory-rooted knowledge, which isn't a problem, but, worse, it seemed like you needed to abandon your theory-rooted sensibilities, too. Basic things, like the expectation that you'll formalize things before getting too far trying to construct or analyze a scheme. Ignoring such sensibilities did not sit well. At the same time, I did feel kind of pushed in this two-hat direction by the kind of questions I was being asked that I could not come close to answering.

3 A Model for Entity Authentication (BR93)

The #1 question. When I would talk to people at IBM about how they were using cryptography, or hoping to use it, the first words out of their mouth was usually something like *So—what are your thoughts on Kerberos?* Until I learned better, my answer was always the same: I'd say I had never heard of the thing. This would cause complete consternation; it simply was not possible. I was from MIT. I claimed to have studied cryptography there. Kerberos, I was told, was *the* most important cryptographic contribution to come from MIT since RSA. Ergo I could not *not* know about Kerberos; it just could not be. I am fairly certain that at least one call was placed to the MIT Registrar's office to see if I had *actually* studied cryptography at that school.

The Kerberos protocol. Well, as I'm sure most of you know, Kerberos is an authentication service developed as part of MIT's Project Athena. People like Sam Hartman, Steve Buckley, and Jeff Schiller worked on it. Plus lots of other people you have likewise never heard of. Plenty of undergraduates and paid, professional staff—more than 20 people in all. But, somehow, no Goldwasser, Micali, or Rivest. The irony of this continues to astound me.

At the center of Kerberos is the Kerberos protocol. It's a six-flow protocol among four parties (well, depending on how you count). I won't bother to describe it. It's a pretty complicated protocol, involving nonces, time stamps, these things called tickets, and lots of different shared keys. Two of the parties, however, don't yet share a key, and it's the job of the protocol to see that they are issued one: a short-lived session key.

Kerberos is actually an elaboration and refinement of an earlier three-party protocol for entity authentication and key distribution due to Needham and Schroeder. And besides this Needham-Schroeder protocol and the Kerberos protocol, there turned out to be a veritable zoo of cryptographic protocols of this flavor—as well as techniques for breaking them or proving them secure, under some definition of "secure." The techniques were mostly logical or algebraic in character, effectively abstracting away the underlying encryption primitive. The lineage of many of these techniques goes all the way back to a 1981 FOCS paper by Dolev and Yao. During the early 1990's, the most popular approach was the BAN logic, named for Burrows, Abadi, and Needham. By the time Kerberos came out, there was a large community of people that did this kind of stuff. They had their own conferences, sensibilities, and disciplinary culture.

I was actually quite shocked to learn of the existence of what seemed to me to be a sort of "shadow cryptographic community" in our periphery—not the NSA, I mean, but an actual academic community that did *cryptography*, by any reasonably catholic understanding of the term, but which wasn't represented at CRYPTO, EUROCRYPT, or the like. This shadow-community remains alive and well. In truth, I do not know how or why cryptography bifurcated in this way, but the impact of this early split is vast. And, I suspect, quite regrettable as well.

What problem does Kerberos solve? Back now to Kerberos, I really wanted to figure out what the thing actually was. I of course read the main papers on it. They did a reasonable job explaining the mechanics of the protocol, but they never clearly described what *was* the cryptographic problem they were trying to solve. It was actually quite frustrating. At some point I remember reading a paper on Kerberos by Bill Bryant (1988). It turned out to be a four-act, two-person, miniature play. (I kid you not—I could not have made that up.) The play had but two characters: Athena, described as an up-and-coming system developer, and Euripides, a good-natured adversary. Repeatedly, Athena would refine her protocol, explain the thinking behind it to Euripides, and he'd promptly attack the thing, and send her back to work. After a tiresome number of such scenes, Euripides offers no more complaints, gives Athena the thumbs up, and *Kerberos* has been born.

I appreciated the author trying to communicate his idea in this nonstandard way, I really did, but I must say that, by the final act, I was about ready to strangle him.[2] The kind of iterative, no-assurance-except-that-Euripides-can't-break-it approach is *exactly* what we are trying to *overcome* by having a scientifically-founded field. The play seemed almost celebratory for something that I had thought one was supposed to be embarrassed about.

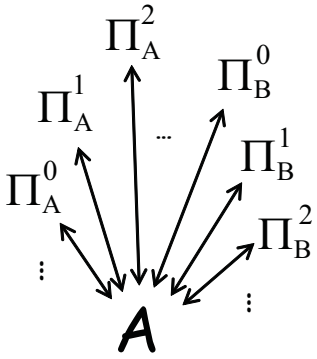
By now convinced that IBM and the rest of their friends in the Open Software Foundation were spending gazillions of dollars on something utterly foundationless, I asked to meet with Jeff Schiller on one of my visits back to MIT. I shared my thoughts with him, explaining that the problem Kerberos solved was completely without foundation, and pointing out that, even if one implements the Kerberos protocol using a semantically secure encryption scheme, as described by [GM] (and adapted for the shared-key setting), *still* it will be trivial to break, as encryption was never the right tool to use. My recollection is that I was perfectly cordial and reasonable, but maybe Jeff thought otherwise, as, just after our meeting, he apparently called up Silvio to ask him just what kind of *lunatic* he had trained.

The BR93 model. Kerberos ultimately seemed too ugly and ill-conceived to work on at that time, but, happily, a paper had recently come out in CRYPTO that offered up a simpler scenario to think about. Bird,

[2] Athena and Euripides too, which might have been unwise.

Gopal, Herzberg, Janson, Kutten, Molva, and Yung (1991) considered *two-party* entity authentication, again in the symmetric setting. They explained how to attack some schemes, including a draft ISO standard; they brought to our attention the idea of *instances* and *sessions*; and they illustrated the idea of *interleaving attacks*. There weren't any actual definitions in the paper, but Mihir Bellare and I saw that we could work up such ideas and get to an actual cryptographic definition.

The model we developed for entity authentication and authenticated key exchange places the adversary squarely at the center of things. It communicates with an infinite sea of oracles. Oracles reify instances. Oracle Π_i^t is meant to model instance t of party i. Oracles are stateful, each initialized to have whatever long-lived keys the corresponding parties are supposed to have. Each oracle can send out messages according to the protocol under study. The adversary communicates with its oracles by way of some repertoire of queries. A Send query, directed to an oracle, is a way to see what the instance will respond with, when you send it some message. A Reveal query, again directed to an oracle, causes it to relinquish its session key, poor thing. A Corrupt query, now directed to a *principal*, results in the loss of all state information of all oracles associated to that principal. A Test query returns either the *session key* that an oracle has within it or else a random key drawn from some specified distribution.



To define an authenticated key exchange (AKE), say, we want to say what it means for an instance to come to have a session key that it shares with only its intended communication partner. So a notion of *partnering* is essential, a way to know which oracle is paired with which. One also needs a notion of *freshness*. A session key housed within some oracle is *fresh* if the adversary cannot know that key by trivial means.

The security of an AKE protocol can be defined by associating a real number to any adversary, that number capturing the adversary's ability to predict if the string returned by pointing of an oracle with a fresh session key is the actual contents of that oracle or a random key drawn from the same distribution.

That's all gone a little quick, I know, but I don't think I need to go over the details more than that to make it plausible that one can give a definition along these lines.

Having defined mutual authentication and authenticated key exchange, we showed how to achieve such ends by simple means. A theorem says, for example, that if you start with a pseudorandom function F and use it in some simple, prescribed way, then the resulting protocol will do the job that that I have sketched: it will be a secure AKE protocol

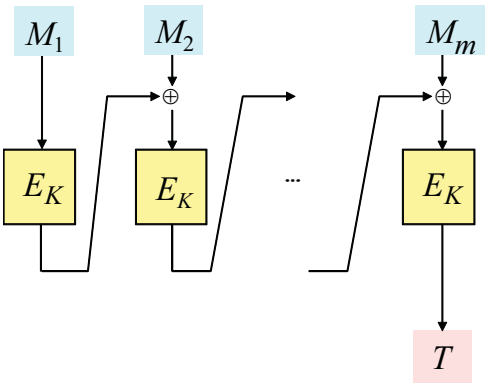
I was quite excited about this paper. In it, we took a problem of enormous practical importance and brought provable security to it, and in a completely practical way. I felt sure that this feat would launch a whole new cottage industry, bringing a lively new problem to my community, reshaping the way that the shadow-community did their stuff, and maybe even unifying the two. But, for years, no such things occurred. In 1995 we went on to do the three-party case. In 2000 we showed how to deal with dictionary attacks, this time with David Pointcheval. In 2001 Canetti and Krawczyk did a paper based on our model, and a follow-up one in 2002, these attending to matters of composability. Our paper finally seemed to get a foothold. Kerberos itself would finally be proven in the computational setting by Backes, Cervesato, Jaggard, Scedrov, and Tsay, but not until 2006, and using the simulation-based framework of Backes, Pfitzmann, and Waidner. By now, BR93 has caught on, even if it did take some time.

4 Provable Security from Blockciphers (BKR94)

The #2 question. Kerberos wasn't the only thing I was asked about at IBM of which I knew absolutely nothing. The second most frequent question was, well, *anything* having to do with symmetric cryptography. I have to admit, with a little embarrassment, that, at MIT, I had somehow failed to pick up anything about classical, symmetric cryptography. It wasn't dealt with in any classes, nor in any provable security research being done. I had heard of block ciphers and DES, but such objects were, frankly, pretty much beneath contempt. I had never heard of a "mode of operation" or a "message authentication code." Real cryptographers just didn't do that kind of stuff. Yet, for some reason, these symmetric things were still big at IBM.

Making a MAC. Mihir and I would soon figure out what this symmetric world was about, and we found it wasn't so bad after all. We decided to start out with MACs. First, one needed such an object in the authentication/key-distribution protocol I just described. Second, Mihir was trying to lend a hand in this plaNET network that IBM was involved in, where a fully parallelizable MAC was needed to authenticate traffic at 1 Gbit a second and beyond.

From a theory point of view, it seemed there was no problem at all: starting with a one-way function, it is easy to make a MAC. Grab enough hardcore bits of a one-way function and you are done. But that kind of answer is, in the end, completely unresponsive to the practical task at hand.

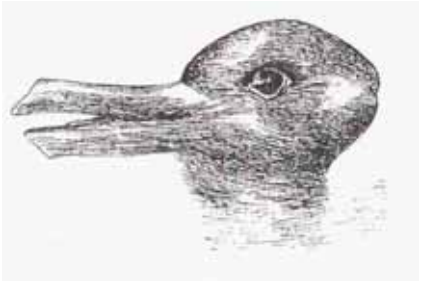


From a practical point of view, the problem would also seem to be solved. Just use the CBC MAC. It's simple and fast, and it's in lots of standards, too. But there was no proof that it was any good, regardless of what you might assume about the blockcipher it was based on.

What Mihir and I wanted to do was to show that the CBC MAC was secure if the block cipher it was based on was secure. Unfortunately, we knew perfectly well that provable security just wasn't possible when starting with an object like a blockcipher. These were finite functions, say mapping 56+64 bits to 64 bits. There is no security parameter present, so one couldn't formalize security. Polynomial time becomes meaningless. There will always be an efficient adversary—indeed a constant-time one—that breaks the blockcipher with a small number of queries. Correspondingly, there will always be a constant-time adversary that breaks your blockcipher-based MAC.

Hmm. In case you didn't catch it, what I just said in the last paragraph was completely bogus. Individual observations are correct—like the existence of an effective, constant-time adversary for the blockcipher and the MAC. But they just don't support the conclusion of the inapplicability of provable security.

Gestalt shifts. It's funny that, even now, when I say something like those prior sentences, I can put myself back in that mindset and it somehow seems correct—even transparently correct. I'll treat you to a visual illusion that you can consider as a metaphor for what I just tried to do. The illusion is something called a "Gestalt shift." This particular image was first used in a philosophical context by Wittgenstein (1953), and later by Hanson (1958) and by Kuhn (1962). You might take it to illustrate how what one sees in one instant—seeing as a duck—can transform into a new way of seeing in another instant—seeing as a rabbit. Most of us will see the drawing as either a duck or as a rabbit, perhaps uncontrollably shifting from one to the other. It is hard to simultaneously see the image as both at once—a duckrabbit—or as neither, just some uninterpreted marks on page. My own feeble brain will do no such thing.



Please regard this only as a metaphor—the cognitive processes associated to the duck/rabbit illusion probably has nothing to do with those associated to how one sees scientific or technical matters in one way, and then, suddenly, another. But see things in very different ways we do, shaped by our expectations and understandings, as instilled within us by our disciplinary culture.[3]

Quantifying blockcipher security. Let's get back to the formalization of the CBC MAC in a reduction-based way. We start by formulating the security of the underlying blockcipher, E. Syntactically, E maps a key K drawn from a finite set and an n-bit string X to the corresponding ciphertext block $Y = E_K(X)$. Each K induces a permutation on n-bit strings. To measure security, an adversary $\mathcal{A}$ is provided an oracle that responds to each query X either with $Y = E_K(X)$, or else $\pi(X)$ for a uniform random permutation π on

[3] What one sees in the duck/rabbit illusion is also shaped by expectations. In one study, on Easter Sunday most children first see a rabbit; on a warm day in June, most children first see a duck.

n-bit strings. The adversary, after interacting with its oracle for some time, outputs a bit, with 1 indicating that the adversary thinks it had the "real" blockcipher oracle and 0 indicating that it thinks it had the random-permutation oracle. The difference in the probabilities is the advantage garnered by $\mathcal{A}$.

In the viewpoint I am describing, we are at this point done defining PRP security: the association of an adversary to a real number *is* the definition of security. Functions meeting the designated syntax are not classified as "good" or not; instead, PRPs are better or worse according to what real number is associated to each adversary.

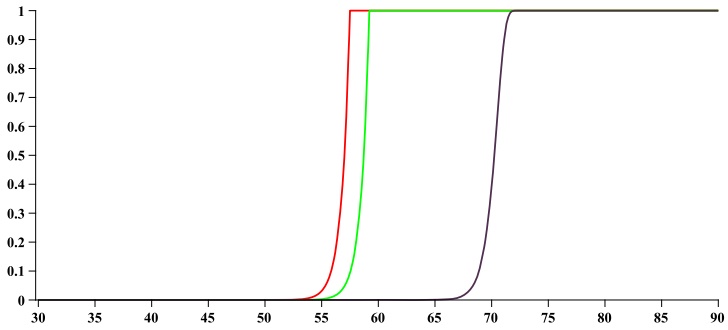
Quantifying MAC security. Similarly, we want to measure how good is a candidate MAC. Syntactically, we regard a MAC as an object that takes a key K from some finite set of keys and a message M from some understood domain and returns, deterministically, a string of some fixed length n. An adversary A is given access to a MAC oracle for the MAC in question, the oracle initialized with a uniformly selected key from the domain. The adversary queries this oracle in an adaptive fashion. At the end, it outputs a pair M^*, T^*. The adversary is said to *forge* if M^* was never queried and T^* is in fact the MAC of M^* under the hidden key K. Security is once again considered to be defined once we have associated a real number to each adversary; there is no security parameter and no absolutist notion for when a MAC is good.

Provable security for the CBC MAC. Following a theorem worked out with Mihir Bellare and Joe Kilian, the security of the CBC MAC can be formalized along the following lines: one asserts that there is a known, efficient, black-box reduction, U, such that for any adversary $\mathcal{A}$ making $q \geq 1$ mn-bit queries against the CBC MAC over some n-bit blockcipher E, adversary $\mathcal{B} = U^{\mathcal{A}}(m, n)$ satisfies

$$\mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) \quad \geq \quad \mathbf{Adv}^{\mathrm{mac}}_{\mathrm{CBC\ MAC}[E]}(\mathcal{A}) - 2m^2q^2/2^n$$

and where $\mathsf{queries}(\mathcal{B}) = m(q+1)$ and $\mathsf{time}(\mathcal{B}) \leq \mathsf{time}(\mathcal{A}) + cqmn$ for an absolutely constant c. In particular, the bound on the quantitative loss in the reduction is well summarized by the $2q^2m^2/2^n$ term, where q is the number of queries, m is the number of blocks in each message, and n is the blocklength.

One of the lovely things about having an explicit formula like that above is that you can graph your results to see what, concretely, they really say. I have started doing that in my own papers, and I encourage people to create a visualization for quantitative security loss when there is a reasonable way to do so. The BKR94 results imply that, for example, an adversary $\mathcal{A}$ attacking the CBC MAC over AES that asks to MAC 2^{55} or fewer messages, each 1 KByte in length, would still have a small chance to forge—about 0.03, as an upper bound. Whatever chance $\mathcal{A}$ has to forge in excess of this value is advantage with which we can break AES, in the sense of distinguishing it from a random permutation, using computational resources comparable to $\mathcal{A}$'s. That's a very concrete and useful kind of thing to be able to say.



Curves corresponding to [BKR94], [BPR05], and an attack. The x-axis is lg q; the y-axis bounds the adversary's advantage.

We can also describe graphically and concretely what improvements mean. Above, I also graphed the better bound from my paper with Mihir and Kryzstof Pietrzak (2005). The improvement looks small in this light, but that's partly because I selected a rather modest length of messages to MAC. Bigger values of m would make the improvement seem more significant.

Finally, on the right, my graph illustrates the efficacy of a natural attack on the CBC MAC. Somewhere between the best security bound and the best attack lies the truth—the actual security of the CBC MAC. Closing this gap, until the two curves are literally one in the same, is a worthwhile game when the primitive is fundamental or widely used.

The real significance of BKR94 is not that it analyzed the CBC MAC. Instead, it helped to bring symmetric cryptography into the provable-security fold. It made MACs a first-class object. It made clear

that blockciphers are a good starting point for doing reductions. And it helped to emphasize that finitary objects could be used within reductions.

5 The Random-Oracle Model (BR93*)

I would like to quickly remind you of a final piece of work of mine from this period: the paper *Random Oracles are Practical* (1993). This seems to have become Mihir and my most well-known piece of work, so perhaps I needn't say much on it. On the other hand, the work has also become our most controversial, so maybe saying something is in order.

The approach advocated in the paper, which builds on Fiat and Shamir (1986) as well as what Mihir and I attribute to folklore, aimed to bring provable security more into the realm of practical, even standardized, schemes. I think it has done this job quite well.

After the RO paper, OAEP was designed and became a popular technique (1994). Boneh and Franklin's scheme for identity-based encryption (2001) was likewise proven secure in the RO model. In fact, literally hundreds of schemes have now been designed or proven secure in the RO model, some of them in standards, and some of them vastly simpler, more efficient, or with weaker complexity assumptions than their non-RO counterparts.

Of course, not everyone likes the RO model. Indeed some people are so anti-RO that they don't even want to call proofs in the RO model "proofs." They'll use words like "heuristic arguments." I personally find this a little bit silly. A proof in the RO model already has a name: it's called a "proof." Something doesn't stop being a proof because you're unhappy with the definition; it stops being a proof when it has a bug.

Serious criticism of the RO model begins with the 1998 paper of Canetti, Goldreich, and Halevi. But concern over random oracles goes back much further. Adi Shamir once mentioned to me that the journal submission for Feige-Fiat-Shamir a (1988) dropped the RO-model proof of the Fiat-Shamir scheme (1986) because a referee wanted the thing removed.

Not surprisingly, I myself think the RO model is intuitive and useful, and find the arguments offered up against it rather uncompelling. Regardless, I am unhappy to see random oracles turned into something ideological, described with words like "fetish." But I do agree this far: that excessive attachment to a model or definition is not a good idea. But this statement holds for *any* definition or model in cryptography—most definitely including our esteemed "standard" model.

It is important to take our models seriously in the sense that we create these things and then we seriously investigate what flows from them. But we are wrong if we convince ourselves that some particular model is objectively "correct"—that it captures, replaces, or defines some objective "reality of security." It is crucial to remember that our models are constructed. The standard model, the RO model, the ideal-cipher model, the Dolev-Yao model, the new model that I find in today's proceedings for dealing with side-channel attacks[4]—every last one of these things is constructed by the hand of man, constructed for some human purpose. So we have to look at how well each model fulfills its purpose. The RO model looks good on these grounds. So do the rest. This isn't an unscientific point of view. It isn't post-modern. It's an acknowledgment of what our community does.

6 Discussion

Why lump together BR93, BR93*, BKR94? I have so far described three works of mine from 1993/1994. One is about entity authentication, one is about a blockcipher mode, and one is about random oracles. Apart from their all coming out at about the same time, why on earth should these disparate papers be grouped together?

In fact, Mihir and I have always regarded these three papers as cut from one cloth. They share a common vision. That vision was the adjustment of provable security to evolve it into a more practically-useful tool. *Practice-oriented* provable security.

[4] The implicit reference is to F. Standaert, T. Malkin, and M. Yung, "A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks," Eurocrypt 2009.

Classical vs. practice-oriented provable security. In the classical approach to provable-security cryptography, our community principally aimed to study abstract security relationships among asymptotically defined objects. The choice of what to use as our starting points, and the choice as to what to reach as an endpoint, was primarily based on aesthetic or philosophical consideration. The community wanted to make conceptually interesting and powerful objects out of objects that were as conceptually simple as possible. All good things to do.

In practice-oriented provable security, we wanted, instead, to study the concrete security relationships among what were often finitary objects. We would be guided, by and large, by practical sensibilities about what problems to consider. We wanted to make efficient and useful objects out of whatever cryptographic practice brought to the table. Also good things to do. But different things.

A big change. As just described, practice-oriented provable security might seem to embody a rather sharp break with the then-prevailing disciplinary culture. To a large extent, I think that's true.

I have already discussed the extent to which, traditionally, provable security was not seen as something to use with finite functions or symmetric cryptography. I have explained that definitions employed a security parameter, and that results were almost always expressed asymptotically.

Throughout the 80's and 90's, the "big" results, as I and other saw them, were plausibility-style ones. The efficiency bar was polynomial time. While the polynomial was usually anonymous, a high-degree one was—I thought—a badge of honor: I figured that only a very smart person could come up with a reduction of complexity $\Omega(n^{20})$, say, or even more. But I started to realize that the efficiency bar for getting work into practice—if one actually wanted to do such a thing—was often light-years away. It wasn't just a question of reducing some polynomial; for something to become "real," there was often a specific bar one had to meet or to exceed. That bar was whatever ad hoc practice could already deliver. If you could sneak under that, you might have your day in court. If not, the obstacles were effectively insurmountable.

Classical provable-security practice was wed to the idea of reducing assumptions. A one-way function was the preferred starting point. I started to realize that this ethos was pushing people away from exploiting the primitives that we actually had. A one-way-function verses a pseudorandom permutation makes a nice example. The theory community had admirably figured out how to make a PRP out of a one-way function—a very hard thing to do. But, ironically, if you wanted a one-way function in practice, one good approach is to make it from a PRP. That's what the UNIX password-encryption facility does.

Finally, it seemed that the new problems that fell under the provable-security scalpel came from one of two sources: either they were problems "well known" within the disciplinary culture, or else they were dreamt up by highly imaginative people. Those are good sources of problems, but I wanted to try to come up with problems by listening to practioners and figuring out they really wanted to do. This leads to a different set of problems. I don't think anyone would ever come up with the notion of AKE by sitting under an oak tree and thinking, nor by talking to other theorists; the problem is worthwhile *because* it serves a community that lives beyond our walls.

Not a big change. There are a variety of counter-arguments to the claim that any of the three papers I have spoken of were significantly counter-cultural.

First, practice-oriented provable security uses, without modification, the conceptual apparatus of provable security. We follow the same definition–protocol–proof approach.

Second, while most provable-security papers expressed their results in asymptotic language, this was always understood to be a matter of choice. Implicit in most every asymptotic claim is a corresponding concrete security statement, and usually it's not hard to extract out that claim. The use or non-use of asymptotics statements is only a question of taste.

Third, one could claim I have actually focused on one particular line of development in cryptography. If one looks at the community at large, it has always been interested in saying practical and concrete things about real-world schemes. Cryptography is not now, and has never been, a monoculture.

Finally, one could make the argument that truly counter-cultural work simply does not occur—can not occur—due to the impossibility of escaping the dialectic of ones professional community while at the same time contributing to it. One can support such a general claim on philosophical grounds, historical grounds, and even linguistic grounds.

I am sympathetic to all of these counter-arguments—they each contain a basic truth—yet each can be countered again. I'll only counter the second point—that concrete security is implicit and extractable, and therefore doesn't really matter.

First, I would point out that, not infrequently, concrete security claims are *not* readily extractable from the papers that appear within our community. I challenge anyone who thinks otherwise to spend a day—or a week—to extract out the concrete security claim, with proof, implicit in Patarin's 2004 paper analyzing six-round Feistel. Second, there is the question of *who* is doing this concrete-security-claim extraction. If the intent is that a would-be user of the result must understand the proof to the extent of sharpening it and reformulating it, I do not think this is a reasonable expectation. Finally, most significantly, the idea that because something is implicit in a work means that it matters little if you do or do not bring that thing out is just not true. It matters not only because scholarship is a communication-based enterprise, but because the way we talk—how we express our results, what we choose to highlight, what we decide to sweep under the rug—these things not only reflect our sensibilities, they also reinforce them. Asymptotic security doesn't just reflect a preference for theoretical, high-level issues—it also buttresses the disciplinary view that that is what counts. As Marshal McLuhan said in his typically pithy way, "We shape our tools, and then our tools shape us" (1964). A predisposition for asymptotic claims in classical provable security helps shape our way of seeing, very much including what problems we see and work on next.

7 Formalizing Symmetric Encryption

The BDJR97 notion. I'd like to move on now to a different example, one that may further illustrate the practical difficulty of escaping ones own disciplinary biases.

By the time Mihir and I had written the papers I've described, in 1993/94, we realized that symmetric encryption too was overdue for a provable-security treatment. In 1997 we finally did this, along with grad students Anand Desai and Eron Jokippi. The main idea was that, lacking a public key, the adversary would need to be provided with an oracle that would effectively enable it to launch a chosen-plaintext attack.

To define the syntax of a symmetric encryption scheme, in comes a key K and a message M. The encryption algorithm is probabilistic—it can flip coins. Alternatively, we allow the encryption algorithm to be stateful—for example, to maintain a persistent counter from one invocation to the next. Having the encryption algorithm maintain state doesn't seem to make sense in the public-key setting, but it makes a lot of sense in the shared key setting.

Decryption works exactly as one would expect. In comes a key and the ciphertext. Out comes, deterministically, the string-valued plaintext.

To formalize security, one could imitate any or all of the notions in Goldwasser and Micali (1982). Our paper gave multiple formulations and proved them equivalent, quantifying the concrete security of each reduction. Here's a simple notion I like, derivative of GM's indistinguishability notion. An adversary $\mathcal{A}$ is given access to an oracle. The oracle may be instantiated in one of two ways. In the first way, the random key K is chosen from the key space and then the oracle encrypts each query M by applying the encryption function to it, keyed by K. Alternative, a key K is chosen and then, in response to each query M, the oracle encrypts not M but a uniform random string of length $|M|$. Adversary $\mathcal{A}$'s effectiveness in attacking the encryption scheme is measured in the natural way.

Necessary? I'd like to ask if it was *necessary* to define symmetric encryption in the way I just did in order to arrive at a desirable, productive theory? I suspect that many other colleagues, if they had taken on the same project at the same point time, would have done things in a similar way. So it seems like the answer might be *yes*.

Nonetheless, I claim that the answer is *No*, that it was *not* necessary to have gone the route we did. I claim that the choice we made was highly *contingent*. It was shaped by the disciplinary culture of the day.

The difficulty in seeing *how* our choices are biased is that we tend not to see or question our most strongly rooted ideas. Deeply embedded disciplinary assumptions can easily assume almost doctrinal unassailability. They become Truth beyond reproach. I gave an example already when I said that, classically, provable security was simply not seen as being applicable to finitary objects.

What are the assumptions with near doctrinal unassailability implicit in the formulation I just gave? There are probably many, not all of which I see, but let me identify two. First: that to be secure, encryption schemes need to be probabilistic (or stateful, now that we're in the symmetric setting). Goldwasser and Micali had convincingly explained that good encryption couldn't be deterministic, and everyone understood this to be so.[5] Second: that encryption is for privacy. Other tools, like MACs or signature schemes, those are for authenticity. Indeed if someone told me in the early-90's that they had designed some symmetric encryption scheme that provided authenticity as well as privacy, I might have suggested that they not call it "encryption," since that is not what encryption was supposed to be about.

We weren't really making conscious choices on these matters; these two assumptions had pretty much receded into the invisible. Ludwik Fleck writes that *Once a structurally complete and closed system of opinions consisting of many details and relations has been formed, it offers enduring resistance to anything that contradicts it* (1935/1979). Fleck's "structurally complete and closed system of opinions" means, for him, a disciplinary culture. The "resistance" that Fleck speaks of can take multiple forms, but the most insidious, it seems to me, is that the assumptions simply become invisible. *What does not fit into the system*, Fleck goes on, *remains unseen.* Ducks are nowhere to be found. Rabbits fill the plains.

Were the BDJR choices "good"? I'd like to suggest that *if* the goal for the notion of symmetric encryption was to create an abstraction boundary that would serve "typical" security needs, *then* the choices do not seem to have been all that good. First, users of encryption schemes quite often *assume* that encryption provides a lot more than privacy. For example, those working with formal models of encryption routinely make assumptions along the lines that "encryption provides authenticity." Protocols like Kerberos have this assumption built right in. A considerable body of anecdotal experience suggests that users, provided a tool that does less than what their intuitions say, will hang themselves every time.

In addition, users of encryption often need to authenticate strings beyond what is being encrypted. A typical example is message headers in the context of a networking protocol. The headers should be authenticated, so as to be checked at the communications endpoint, but they typically can't be encrypted, since intermediate routers won't have the key. Such considerations suggest that authenticating something as an adjunct to what is being encrypted is conceptually desirable.

Finally, the inclusion of state or random coins needs special attention. Anecdotal evidence again makes clear that implementers and protocol designers routinely get wrong the provisioning of random coins. Cryptography books contain wrong advice in this connection. Generating some approximation of random coins can be quite costly, and, in practice, coins are usually harvested rarely and, in implementations, almost certainly outside of an encryption scheme's boundary. Very often, the party that is encrypting a value will already have available to it some nonce—a value used at most once per session—such as a sequence number in a communications packet. For all of these reasons, it makes sense that one assume encryption schemes are themselves *deterministic*, but that the user of a scheme provides to it a nonce.

AEAD. We can reflect in the syntax of an encryption scheme the modified notion of what an encryption scheme should take in.[6] The encryption mechanism is now deterministic. It takes in three inputs: a nonce N, the associated data AD, and the message M. Deterministically, it computes a ciphertext C. For decryption, the scheme takes in a nonce N, associated data AD, and the message C. Deterministically, it computes a string-valued ciphertext C or else an indication that the ciphertext should be deemed invalid. Let's call a scheme like this an *AEAD* mechanism, *authenticated-encryption with associated-data*.

Note that, in our formulation of AEAD, the user is expected to somehow communicate N and AD along with a ciphertext, as these are needed to decrypt and are not implicit in a ciphertext. Also note that, unlike conventional probabilistic encryption, there is no longer any need for the ciphertext to be longer than the plaintext: in a "good" scheme, one would expect the ciphertext to have the same length as the plaintext.

To define security, the adversary is presented with a pair of oracles. There are two possibilities. The first is to give the adversary an encryption oracle and a decryption oracle. Both are initialized with the same

[5] Within IBM, I learned and followed a convention to use the word *encipher*, rather than *encrypt*, to refer to what a deterministic scheme can do. I used to hate to hear the word "encrypt" when someone had in mind a deterministic map.

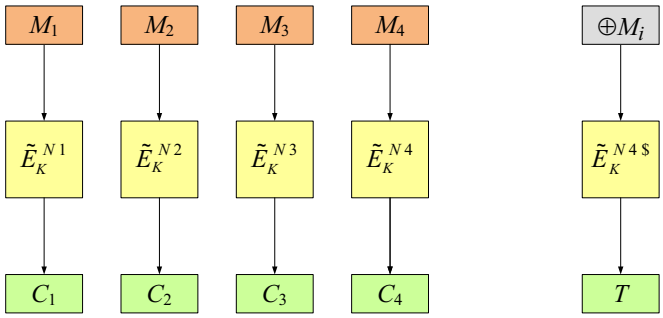
[6] The syntax and security notions I use for AEAD come out of [Rogaway, Shrimpton 06], [Rogaway 02], [Rogaway, Black, Bellare, Krovetz 01], [Bellare, Rogaway 00], and [Katz, Yung 00]. In particular, the nice all-in-one definition for security comes from [Rogaway, Shrimpton 06]. In the past, security was defined with separate notions for privacy and authenticity.

key K, chosen uniformly at random from the key space (assume it is finite). When the adversary asks an encryption query of (N, AD, M) it gets the encryption of M with respect to the given nonce and associated data. When the adversary asks the decryption oracle (N, AD, C) it gets the decryption of C with respect to the given nonce and the associated data. Both are with respect to the initially chosen key K.

Alternatively, when the encryption oracle takes in N, AD, M, it encrypts as before to get a ciphertext C, but then returns $|C|$ random bits. If the encryption is length-preserving, as one would hope, then the oracle just returns $|M|$ random bits, ignoring N and AD. On receipt of a decryption query N, AD, C, the alternative decryption oracle just returns an indication $\perp$ of invalidity.

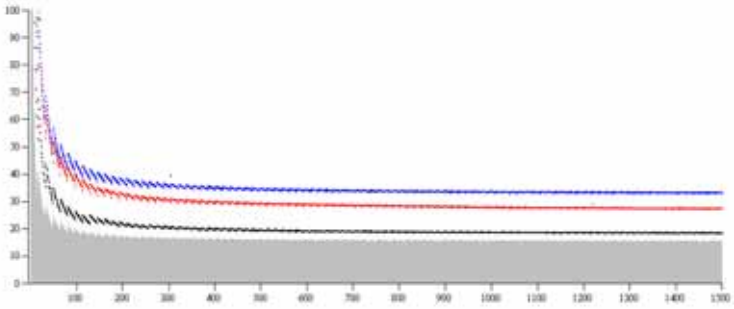
To keep the adversary from winning by trivial means and to ensure that nonces have their intended semantics, we add in the restriction that the adversary shouldn't reuse a nonce in an encryption query, nor ask a decryption query of (N, AD, C) after it earlier asked an encryption query of $(N, AD, \mathit{Something})$ that resulted in a response of C.

Viewed in this way, AEAD security looks very much like another version of CCA security: in fact, CCA2 security is what you get when you replace the $\perp$-returning oracle by a decryption oracle. Perhaps if notions had come out in a different order, AEAD might be called "CCA3" security, instead.



OCB. One of the advantages of seeing AEAD as a single conceptualized problem, rather than as an envisaged solution combining an encryption scheme and a MAC, is that one can now try to optimize performance for the now-defined goal. Here's one solution, called OCB. For simplicity, let's forget about the AD, imagining it to be empty. Let's assume the blockcipher has a blocklength of 128, as would AES. Let's assume that the message we want to encrypt has a length that is a multiple of the 128 bits. Finally, let's use a beautiful abstraction due to Liskov, Rivest, and Wagner (2002) called a *tweakable blockcipher*. The "tweak" is written in the superscript. Every tweak determines, conceptually, a blockcipher independent from that associated to any other tweak. Defining security by modifying our earlier PRP definition is easy. So, in OCB, the message is divided into blocks and the tweak includes the index of the block as well as the nonce. At the very end we do something a little different, bringing in a special tweak. Assuming the underlying tweakable blockcipher is secure (for forward and backward queries both), it is easy to prove that this scheme meets the notion just described.

If you're clever in how we realize the needed tweakable blockcipher, you can instantiate it using a conventional blockcipher in a manner that needs just one blockcipher call plus three more xors of 128-bit strings. That's fairly remarkable. It means that the construction I've just sketched will achieve authenticated encryption at close to the cost of ECB plus four xors per block (an extra one for the checksum), plus a final blockcipher call.



Here's a bit of experimental data recently prepared by Ted Krovetz. It compares the performance of OCB (bottom curve, message length verse cycles per byte, C code, 64-bit x86 processor) and two alternative AEAD schemes: GCM (middle curve) and CCM (top curve, the slowest mode). CCM is a two pass scheme involving one layer of counter mode and one layer of CBC MAC. So it uses about twice the number of AES calls as would ECB or OCB. It is inherently non-parallelizable too, because of the CBC MAC. GCM uses one layer of CTR mode and one universal hash function application. The universal hash function is based on polynomial arithmetic over GF(2^{128}). The blocks of message are treated

as coefficients of a polynomial, and that polynomial is evaluated at key material to determine the hash value. This method is parallelizable and works well in dedicated hardware, but it may not be fast in software, where it typically uses large, key-determined tables.

8 Conclusions

Standards. AEAD isn't only about making faster techniques; it's also about finding the most useful abstraction boundaries. I claim that how we formalize notions can have a major impact on whether or not our notions get used, and if they get used correctly. There are, in fact, no cryptographic standards that follow the BDJR abstraction boundary; conventional modes of operation surface an IV to the user rather than insist that it be random and inaccessible. But there are already standardized schemes that follow the AEAD abstraction boundary that I just described. I just mentioned them: CCM and GCM. While I don't think either scheme is well designed, it is at least the case that we finally—since 2004—have a standardized symmetric encryption scheme that achieves a strong enough notion of symmetric encryption to have a good chance of being correctly used. The schemes have definitions and proofs, the proofs asserting concrete and useful bounds under standard assumptions. That is progress. And it is fairly representative of what's been going on. Practice-oriented provable security has given birth to a great many standards, many of which are extensively used. This includes HMAC, designed by Bellare, Canetti, and Krawczyk (1996), which is not only a NIST standard but a fixture of computer security practice, deployed in every browser. That is progress.

A number of my own schemes, or schemes derivative of them, have made it into cryptographic standards: public-key encryption schemes OAEP and DHIES, signature schemes PSS and PSS-R, message authentication codes CMAC and UMAC, and enciphering schemes XTS and EME2. More than 20 standards in all.

In general, practice-oriented provable security has reshaped the face of cryptographic standards. At this point, provable security has become a de facto requirement for almost all new standards for mid-layer functionality (encryption schemes, MACs, and the like). Even if some of the membership of a standards body doesn't fully understand what is and is not asserted in some proof, still this seems like progress, and a very real way in which our community is having an impact outside of its cocoon.

Benefits of our inquiry. Recognizing the extent to which a field is constructed can be liberating. This is one of the claimed benefits of a constructionist view. As an example, realizing that a belief like "good encryption must be probabilistic" is a pure construction opens the door to a variety of interesting research questions. Papers like [Rogaway, Shrimpton 2006] and [Bellare, Boldyreva, O'Neill 2007] emerge. When disciplinary assumptions lose a bit of their authority, the resulting liberation creates an atmosphere where invention can more easily thrive.

At least for me, it is by now clear that our community is engaged less in a program of discovery than in an ongoing dialectic. That dialectic continually refines, reformulates, and answers anew basic questions like "what is encryption, and what can and should it do?" Once upon a time, I had thought that such questions had been answered.

I am not sure we always act as though we are engaged in a dialectic. This isn't the prevailing model for how scientific inquiry works. The result is that doctrinal unassailability and invisibility can become more common than they need be. Opinions can get strident and entrenched.

The question of interest, ultimately, is if our field is actually healthy. The answer is that I don't really know. There are, subjectively, reasons for concerns. We can be rather isolated and inward-looking. I can't verify many papers that I try to read. I worry that we are having a less profound impact on computer security than one would want to see.

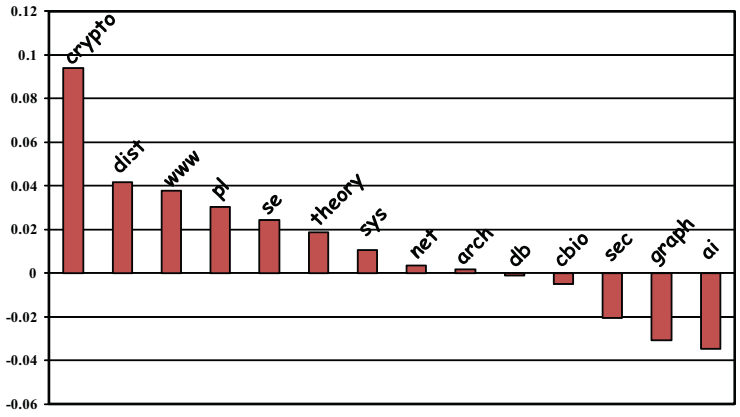
In his Turing award lecture, Adi Shamir predicted that "Crypto research will remain vigorous, but only its simplest ideas will become practically useful" (2002). This seems to me a sad but realistic appraisal of our future. I wonder, though, if the appraisal assumes that our disciplinary community will retain the basic character that it has today. That is not something set in stone.

If our disciplinary culture really does have such a profound impact on the nature of our field, and if we are concerned about the connectedness or utility of our field, maybe it makes sense to seriously consider the character of our community. I think that Neal Koblitz (2007), and before that Koblitz and Menezes (2006,

2007), tried to do this, but it seemed to produce more anger than introspection. Perhaps I can avoid this outcome by focusing on something beyond anecdotal or subjective claims.

Assortativity. Specifically, let me describe some data from colleagues of mine at UC Davis, a paper entitled "Structure and Dynamics of Research Collaboration in Computer Science." What Bird, Barr, Nash, Devanbu, Filkov and Su (2009) did was to select 14 areas of computer science research and, with the help of domain experts, identify the tier-1 conferences in each area. For cryptography, the selected conferences were CRYPTO and EUROCRYPT. Then the authors studied the collaboration network among authors who had published in any tier-1 conference in the 14 areas. All data was harvested from DBLP. The authors looked at various measures from bibliometrics and social-networking theory to try to understanding the structure of the graphs.

One of the measures looked at was assortativity. Given an undirected graph with numeric labels on each vertex, the graph's assortativity quantifies the tendency of vertices to be connected to similarly-valued vertices. A graph with high assortativity is one where there's a strong tendency for nodes to be connected to like-valued nodes. A graph with zero assortativity is one where there is no such tendency. A graph with negativity assortativity is one where nodes tend to be connected to nodes with rather different vertex labels. The assortativity is normally defined by the Pearson coefficient. It's a standard and widely used measure in social networking theory, defined in a well-known paper by Mark Newman (2003).



If one annotates the vertices of the collaboration graph by people's "seniority," measured by the number of years from the current year to the first paper in the dataset, Bird *et al.* find that cryptography has strangely high assortativity. The data is shown here, refreshed by Christian Bird to make it current through 2008. So senior people tend to write papers with senior people, mid-career people write papers with mid-career people, and so on. High assortativity also occurs if you annotate vertices in other ways, like number of publications an author has. In other words, more than in other sub-fields of computer science, in cryptography, tier-1-prolific people tend to write papers with tier-1-prolific people, authors who only rarely have papers in our top-tier venues write papers with authors who only rarely have paper in our top-tier venues.

What does all this mean? So what if we are highly assortative anyway; this isn't exactly an insult kids fling at one another. But in social networking, high assortativity is considered worrisome. It's taken to reflect a lack of diversity in social interchange—a possible sign of cliqueishness or chauvinism. And maybe that is something of concern.

I will leave you to draw your own conclusions. There are always ways to challenge or explain away data like this. But I myself, just from my own anecdotal impressions, was not at all surprised to see that our field came out as highly assortative. You might think if it makes sense to you.

	Practice-Oriented Provable Security	Classical Provable Security
Inspiration for problems	Existing practice	Human imagination
View towards sym crypto	Favorable, engaged	Unfavorable, indifferent
Starting points should be	Efficiently realizable	Conceptually simple
Formalization typically	Concrete	Asymptotic
Random-oracle model	Receptive	Oppositional
Post-paper success	Refs, Stds, Systems	Refs
Cryptography is	Techno-science	Science
New results are	Invented	Discovered
Metaphysics	Nominalism	Inherent structuralism
Development model	Contextualism	Linear development

Summary of POPS. Wrapping up, let me summarize some of the differences between practice-oriented provable security and the more classical thread. Like any summary that tries to capture much with few words, my comments are rather elliptical. But you might say that, in contrast to the classical provable-security approach, practice-oriented provable security tends to takes its problems from existing practice and problems already of known value to practice. We are favorably disposed to symmetric cryptography, and no less likely to pursue a problem because it falls in that tradition. When we choose a problem to start with in a reduction, the crucial question is if there is an efficient existing instantiation. When we give theorems, we prefer to do this in a concrete-security framework. I remain happy with the random-oracle model and, in general, would like to see an expansion of models of interest (also, a less emotive responses to models one does not like). After a paper is done, I am particularly keen to see it find its way into cryptographic standards or computing practice.

From a more philosophical point of view, you might say that practice-oriented provable security is not positioned as simply part of mathematics or science; it attends too closely to engineering concerns, and human concerns, for that. I see the emergence of results as much more invention than discovery. In terms of metaphysical underpinnings, there would seem to be an underlying notion that *man* draws the abstraction boundaries in cryptography, rather than believing that those lines are already present within the world. This is the philosophical position known as *nominalism*, which entails the belief that *we* create the conceptual folds of our universe, we create it through our language and our work products. Finally, more sociological than philosophical, I see the science of cryptography as engaged in an active dialectic with practice: science does not feed technology, but each feeds the other, with society-at-large also sitting at the table.

Avoiding misunderstandings. An essay like this, and a talk even more, offers much opportunity for being misunderstood, and even ruffling some feathers. Let me make a few clarifying points.

- First, I hope it is clear that I have tremendous respect for what has been accomplished by provable-security cryptography, and tremendous admiration for those individuals who have made key contributions to it. Essentially everything I have done in my career flows from provable-security cryptography. Correspondingly, I hope it is clear that my comments on MIT should not be construed as negative; I had an amazing and wonderful education there.
- Second, nothing I have said should be understood to suggest that proofs don't matter, or that rigor isn't necessary, or that intuition is what really counts. I believe the opposite—that we need to attend more closely to rigor, and that we should always question our own intuitions.
- Third, nothing I have said should be understood to deny or even question the existence of objective mathematical truth. However our community may construct our disciplinary universe, we must do so within the inexorable confines of mathematical truth. It seems to be a common misconception about constructionism that it is incompatible with a realist metaphysical view. It is not.
- Fourth, I would claim that the views I have expressed do nothing to rob us of any deserved authority. Most of the stature that we have—and it is not much—we have because of what we can accomplish with our recondite art.
- Fifth, there seems to be some sort of fear that a constructionist view can lead one to regard all ideas as equally valid or worthwhile. It should not. If anything, working to see how forces within our disciplinary culture can shape our field should make us more discerning than before.
- Sixth, nothing I have said allows one to transcend the dialectic of our community and contribute something fundamentally acultural. A bit of detachment may help one see a problem that was not so obvious before, but it probably can't reach much further than that.

– Finally, I do not suggest that challenging the disciplinary culture of a community is somehow useful for its own sake. Work can be good or bad independent of how conforming or counter-cultural it may seem.

The allure of realism. Given all I have said, why might one lean towards scientific realism in trying to understand what it is we really do? There are lots of reasons, such as these: (a) One may see a definition like that of a one-way function and it seems so beautiful and compelling that one is sure that this must be a basic conceptual fold in our disciplinary world. (b) One may spend the bulk of his time trying to prove some result, time spent during which one is more likely to believe, rightly or wrongly, that the question closest at hand, to find or not to find a proof, falls squarely within the realist realm. (c) One may personally identify with the community of mathematicians and scientists, not engineers, because it seems more noble or more consonant with ones sense of self. (d) Having chosen a technical field—perhaps not even liking the humanities or social science—one may look towards science to explain much of what we do, even when this doesn't really work. (e) Finally, intuition and social conditioning play important roles in determining, at least initially, where one sees the boundaries between constructionist and realist space.

We make cryptography.
We make cryptography.

Finally, let me summarize the constructionist aspect of my talk with a suitable image from down the street of our conference venue.[7] To me, it speaks more eloquently the ideas of construction than any words that I could say.

Acknowledgments Many thanks to the Eurocrypt 2009 Program Committee for kindly inviting me to give a talk, an invitation that spurred the writing of this essay. Thanks especially to the Program Chair, Antoine Joux, for not freaking out when he saw my proposed title, complete with the phrase "social construction."

Ideas expressed in this talk are heavily influenced by my years working with Mihir Bellare, and by a thousand conversations with him. Most of the ideas were jointly developed (only the wrong ideas are mine alone). Mihir also gave me valuable, specific comments on multiple earlier drafts of this essay.

My nearly accidental discovery of Ludwik Fleck's fascinating book, *Genesis and Development of a Scientific Fact* (1935/1979), about a year ago, led me to think more seriously on the viewpoints in this essay.

Before and after my talk, I received kind and useful feedback from Whit Diffie, Shafi Goldwasser, Ted Krovetz, Adam O'Neill, Tom Ristenpart, Till Stegers, John Steinberger, and many attendees of Eurocrypt 2009. Many thanks to each of you.

I have received generous support from the NSF over the years. The current essay has been written in connection with NSF CNS 0904380.

[7] Eurocrypt 2009 was held in Cologne, Germany. Construction of the Cologne Cathedral, pictured, began in 1248 and was only completed in 1880, some 632 years later. The Hohenzollern Bridge, also pictured, is more contemporary, rebuilt after World War II.