Editors: Alessandro Acquisti, acquisti@andrew.cmu.edu | Hugh Thompson, hthompson@peoplesecurity.com

What Happened to the Crypto Dream?, Part 2

Arvind Narayanan | Princeton University

Part 1 of this article started my examination of why cryptography doesn't seem to have done much for digital privacy, although it has been relatively successful at improving security.[1] I separated two visions:

- "Cypherpunk Crypto"—the dream of wielding crypto as a weapon for social and political change, and
- "Pragmatic Crypto"—a more down-to-earth view that seeks to engineer modest privacy enhancements in specific applications.

I then discussed why the cypherpunk world hasn't materialized.

Here, I look at the Pragmatic Crypto vision, a goal that seems much more plausible. Developing privacy-preserving algorithms and systems constitutes a significant chunk of modern crypto research. Such designs seek to avoid causing privacy breaches any more than is strictly necessary for correct operation of the system. This research field traces its roots to Andrew Yao's "garbled circuit construction," a piece of "crypto magic" dating to the early '80s.[2] (More recently, Craig Gentry's fully homomorphic encryption has been a key addition to the repertoire.[3]) Yao introduced the "millionaires' problem," in which two millionaires seek to determine which of them is richer without revealing their actual wealth. It turns out that using crypto, they can do this while leaking nothing more than the single bit of information they're interested in computing. But the shocker is that there's a two-party protocol to compute an arbitrary function of two secret inputs in a way that reveals nothing more than the function's output.

This is powerful because the two parties can be anybody—a website and a user, two companies, a government and a citizen, or two end users. There are also secure computation protocols with more than two parties, of course. Researchers have used them to demonstrate privacy-preserving versions of functionalities in just about every conceivable domain—voting, electronic health systems and personal genetics, location-based services, and so on. In addition, a long line of research papers have described tailored, more efficient versions of generic multiparty computation in various domains (a Google Scholar search for "privacy preserving", with quotes, showed 21,600 results). Nevertheless, despite the potential, privacy-preserving crypto technologies have largely failed to find their way into practice.

The two most frequent explanations for this failure are that no demand exists for privacy and that crypto is too slow. I reject both—the first is incorrect, and the second is both incorrect and irrelevant. (In contrast, my primary explanation for why Cypherpunk Crypto failed to materialize is insufficient demand.) The actual reasons are varied and complex; throughout the rest of this article I explain them and how to possibly mitigate them. My main focus is on commercial applications that collect personal data, but most of my arguments also apply to noncommercial contexts.

Human Factors

Nontechnical people are largely unaware of the existence or possibility of privacy-preserving computation. This matters for several reasons. Consider, for example, a Stanford project I led on private proximity detection.[4] Our system allowed two friends (say, Facebook friends) to be serendipitously notified on their smartphones when they were near each other (for example, when

 Copublished by the IEEE Computer and Reliability Societies

they were both browsing in a library at the same time). It provided the crypto guarantee that no information about their locations would be revealed to other users or the service provider. Such a modest but useful domain-specific goal is typical of Pragmatic Crypto.

Here's the problem: there's no easy way for a service provider to explain the crypto guarantee to users who have no mental vocabulary for it. Indeed, trying to advertise an apparently impossible location-based service in which the service provider never learns the user's location will likely only backfire and make the app seem less trustworthy! Trying to sell a product to consumers who don't know they need it is a bad enough problem for a business, but this is an especially hopeless version.

But this isn't just about users. There's also an institutional unawareness of privacy-preserving computation at companies. The result is that many companies treat privacy as a legal or compliance problem to be solved by legalese, rather than a (partly) technological one. Policy makers are yet another contingent who suffer from misconceptions about information privacy, including ignorance of crypto. Crypto-for-security has been spurred partly by regulatory incentives such as encryption safe harbors in data breach notification laws, but no corresponding incentives exist for crypto-for-privacy.

How can we fix this? Educating the general public on this topic won't be easy. That said, I feel that everyone taking a college computer science class would gain from acquiring a basic awareness of how to think about digital privacy, including learning about crypto's possibilities. This doesn't require learning the mathematical and algorithmic details.

Another major sticking point is usability, and key management in particular. A much-cited 1999 study found that PGP (Pretty Good Privacy) is essentially unusable for most users.[5] Although the user interface has no doubt improved since then, the underlying conceptual and architectural complexities won't go away. I feel strongly that unless key management is completely invisible, it will always be a nonstarter for consumer devices. Interestingly, some systems such as Off-the-Record Messaging have managed to avoid key management by aiming for a slightly different set of privacy guarantees.

Developers Are People Too!

Perhaps the most underappreciated cause of Pragmatic Crypto's lack of success is engineering complexity. Modern crypto protocols are too complex to implement securely in software, at least without major leaps in developer know-how and engineering practices. A simple example illustrates how embarrassingly bad the situation is. The hash function is the most common crypto primitive, and the length-extension vulnerability is a basic pitfall resulting from the insecure use of hash functions. Every crypto course covers this bug, yet a 2009 study of popular web APIs found 11 whose signature specification was vulnerable to it.[6]

There's an important reason why lack of expertise affects crypto-for-privacy much more than crypto-for-security. People usually want a limited repertoire of security properties. So, enforcing security can be, and typically is, parceled off to well-analyzed modules (such as password-hashing libraries built into various platforms). These modules are written and scrutinized by true experts, of which there are scarily few. However, privacy-preserving computations are domain specific, and they convert the specific functionality into a cryptographic protocol. This makes them poorly modularizable. The idea that a developer who isn't a crypto expert could read a modern paper and understand and implement the protocol in a bug-free way is laughably unrealistic.

The crypto research community could do something about this but has generally chosen not to. I'm aware of two interesting exceptions. The first is a line of research Thomas Ristenpart christened "practice-driven cryptography theory"—that is, theoretical analysis of the security properties of constructions that do get implemented in practice.[7] Although this approach has a long way to go, I can imagine researchers using it to develop cryptosystems with an eye on implementation complexity. Second, some researchers have argued that generic secure two-party computations can be made fast enough for practical use.[8] Again, this insight is only a first step toward a practical tool chain, but perhaps one day we'll be able to avoid custom protocols altogether in many cases.

Misaligned Incentives, Mismatched Models

The next factor is relatively well understood and has often been discussed by privacy scholars outside computer science, but is nevertheless worth noting. Cryptography helps enforce secrecy or confidentiality, but privacy is much more nuanced, better captured by constructs such as contextual integrity.[9] To give just one example of the inadequacy of crypto alone as a technological privacy protection mechanism, consider the need for "breaking the glass" in medical informatics systems. Medical personnel must always be able to override any access control mechanisms in an emergency. Clearly, an auditing and accountability framework is necessary to enforce privacy in such a context. This is arguably an underresearched area of computer

security, and its development in conjunction with crypto could lead to more useful and secure systems.

Perhaps the toughest barrier to Pragmatic Crypto is economics. Whether entirely through inevitable economic pressures or partly by a historical accident, we've ended up in a world where aggregation of personal data is an engine of the digital economy. Although Pragmatic Crypto's goal can be thought of as roughly to prohibit secondary use of data, secondary use is in fact a business imperative. Does this simply mean that no demand exists for privacy? Privacy advocates have argued otherwise—that we have a market failure and that the market underallocates privacy. Behavioral economics provides one explanation. Alessandro Acquisti showed that a triad of problems—incomplete information, bounded rationality, and psychological distortions—means that consumer behavior differs greatly from rational, informed choice.[10] If we accept this argument, I see regulation as the only way to realign incentives with efficient market outcomes. I'm not saying that regulation is always a good idea; rather, no other forces (say, public-relations pressure) seem strong enough to reverse the trend.

My final point is about trust. The trust model in crypto is that the user controls and trusts his or her devices or end nodes and the software running on them, but doesn't trust third parties on the Internet. However, consumer technology has evolved away from this model over the past decade or so. Hardware and software are increasingly vertically integrated and packed together in a way that users can't fully control or modify. This is reinforced by legal restrictions such as the Digital Millennium Copyright Act. Combined with the fact that today's software typically updates automatically, not trusting vendors isn't an option anymore. (Jonathan Zittrain also made this point in his talk "The End of Crypto."[11])

In a related development, users' trust boundaries have evolved away from physical nodes toward brands. Indeed, this "feudal" model is credited with *improving* security. After all, Google's and Amazon's servers are dramatically more secure against theft or intrusion than smartphones and other client devices. The model has also been credited with indirectly improving privacy, to the extent that privacy breaches result from security failures.

In other words, not only does the usual crypto threat model greatly overestimate users' trust in software running on their own devices, it equally underestimates trust in (some) third parties. Many crypto protocols treat service providers as adversaries, a model that's nonsensical in the modern computing environment. Consumers don't seek technological privacy protection against governments and service providers but against their peers, nosy neighbors, stalkers, employers, insurance companies, advertisers, and the like. Even when the adversary in a crypto protocol isn't the service provider, it isn't necessarily practical to use crypto. Often, it's simpler for the trusted service provider to act as a privacy intermediary. Facebook's ad-targeting platform is a good example—it never directly hands over user data to advertisers.

I believe that the two factors I just discussed—misaligned economic incentives and an unrealistic threat model—fundamentally limit the commercial applicability of cryptographic privacy protection technologies. Nevertheless, crypto has an important role in improving privacy, and it hasn't lived up to that potential. I've laid out three avenues for action:

- Improve crypto awareness and education.
- Treat improved usability and minimized implementation complexity (and not just security and performance) as principal design goals of cryptosystems.
- Develop complementary technologies such as accountability mechanisms.

I hope some of these changes, possibly coupled with policy incentives, can move us in the right direction. ■

Acknowledgments

I'm extremely grateful to Joseph Bonneau, Ed Felten, and Vitaly Shmatikov (in no particular order) for comments on a draft, and to the audience at my talks at the Electronic Frontier Foundation and Princeton for useful feedback. Any errors, opinions, and omissions are my own.

References

1. A. Narayanan, "What Happened to the Crypto Dream?, Part 1," *IEEE Security & Privacy*, vol. 11, no. 2, 2013, pp. 75–76.
2. Y. Lindell and B. Pinkas, "A Proof of Yao's Protocol for Secure Two-Party Computation," Electronic Colloquium on Computational Complexity, 2004; http://eccc.hpi-web.de/report/2004/063.
3. C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," *Proc. 41st Ann. ACM Symp. Theory of Computing* (STOC 09), ACM, 2009, pp. 169–178.
4. A. Narayanan et al., "Location Privacy via Private Proximity Testing," *Proc. 2011 Network and Distributed System Security Symp.* (NDSS 11), Internet Soc., 2011; www.internetsociety.org/doc/privacy-private-proximity-testing-paper.
5. A. Whitten and J.D. Tygar, "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0," *Proc. 8th Usenix Security Symp.*, Usenix Assoc., 1999.
6. T. Duong and J. Rizzo, "Flickr's API Signature Forgery Vulnerability," 2009; http://vnhacker.

blogspot.com/2009/09/flickrs-api-signature-forgery.html.
7. Y. Dodis, T. Ristenpart, and T. Shrimpton, "Salvaging Merkle-Damgård for Practical Applications," *Advances in Cryptology—Eurocrypt 2009*, LNCS 5479, Springer, 2009, pp. 371–388.
8. Y. Huang et al., "Faster Secure Two-Party Computation Using Garbled Circuits," *Proc. 2011 Usenix Security Symp.*, Usenix Assoc., 2011.
9. H. Nissenbaum, "A Contextual Approach to Privacy Online," *Daedalus*, vol. 140, no. 4, 2011, pp. 32–48.
10. A. Acquisti, "Privacy in Electronic Commerce and the Economics of Immediate Gratification," *Proc. 5th ACM Conf. Electronic Commerce* (EC 04), ACM, 2004, pp. 21–29.
11. J. Zittrain, "The End of Crypto," *Advances in Cryptology—Crypto 2012*, Springer, 2012, p. 86.

Arvind Narayanan is an assistant professor of computer science at Princeton University. Contact him at arvindn@cs.princeton.edu.