# Bruce Schneier: Building Cryptographic Systems

**Charles Severance,** University of Michigan

*Security expert Bruce Schneier discusses security from the perspectives of both the National Security Agency and the National Institution of Standards and Technology.*

Since the 1930s at Bletchley Park, there has been a continuous arms race to both improve and break cryptography. The files leaked by National Security Agency (NSA) contractor Edward Snowden made it clear that governments regularly gather data on average citizens, which makes us wonder if privacy is even possible. Do our carefully designed cryptographic systems protect our information as we expect them to, or are they just thin veils that can easily be pierced by the government? I posed these questions to leading security expert Bruce Schneier. Watch the entire interview at www.computer.org/computingconversations.

See **www.computer.org/computer-multimedia** for multimedia content related to this article.

## CRYPTOGRAPHY AND THE NSA

When asked whether a cryptographic standard like the Advanced Encryption Standard (AES) offers protection against well-funded and highly skilled prying eyes, Schneier replied:

> *One of the things we've learned from the Snowden documents is that broadly applied cryptography gives the NSA trouble, at least at scale. The NSA does a lot of cryptanalysis and breaks a lot of systems. But well-designed and well-implemented cryptography does stymie [the NSA].*

All cryptography can eventually be broken—the only question is how much effort is required:

> *Cryptography forces attackers to have a priorities list. Depending on their time and budget, they'll work their*



0018-9162/16/$33.00 © 2016 IEEE

EDITOR **CHARLES SEVERANCE**
University of Michigan; csev@umich.edu

*way down the priorities list. Your hope is that you're below their budget line. Without cryptography, an organization like the NSA can bulk-collect data on everybody. With cryptography, organizations are forced to be more targeted. That's extraordinarily valuable because it means the FBI will go after criminals, the NSA will go after agents of a foreign power, and the Chinese government will go after US government officials that rise to whatever level it wants to spy on. The cybercriminals will just go after a few of us, and the rest of us are protected.*

In truth, having good cryptography algorithms doesn't automatically ensure security because the algorithms must be realized in real-world systems:

*When we say we trust the cryptography, all we're saying is that we trust the mathematics. Everything I know about cryptography tells me the math is good. Certainly there will be cryptographic advances, and some things will be broken in the future, but by and large the math works. But math has no agency. Math can't do anything—it's equations on a piece of paper. In order for math to do something, some of us need to take that math to write code, embed that code in a program, and embed that program on a computer with an operating system on a network with a user. All those things add insecurity.*

Those who would defeat cryptography rarely attack the mathematics directly; instead, they attack the systems, networks, and humans that implement and use the security:

*There's an important corollary here: complexity is the worst enemy of security. The more complex you make your system, the less secure it's going to be, because you'll have more vulnerabilities and make more mistakes somewhere in the system. We learn again and again when we see analyses of voting systems, embedded systems, cell phones, messaging systems, or email systems that the vulnerability is always outside the cryptography. It's almost always something that the designer, implementers, coders, or users got wrong. The simpler we can make systems, the more secure they are. We recently learned about vulnerabilities in the key agreement protocols that are used to secure a lot of VPNs [virtual private networks] and Internet connections. If you look at where that vulnerability occurred, it was due to a shortcut that allowed for massive pre-computation. The math works great, but the implementation of the math was flawed.*

One way to weaken a security standard is to introduce complexity:

*The Internet Engineering Task Force [IETF] process for Internet standards doesn't really work for security because those standards are compromises made by a committee. They put in all the options to make everyone happy. They put in as much flexibility as necessary to make the system as comprehensive as possible. That approach is anathema to security. Security needs as few options and to be as simple as possible. You don't want to compromise. You want one group to win because that group has a self-contained vision. If you have a piece of this and a piece of that, there's going to be some interaction you didn't notice. And that interaction will be the vulnerability that breaks your system.*

## CRYPTOGRAPHY AND NIST

To make sure the underlying mathematics of cryptography are solid, the National Institute for Standards and Technology (NIST) runs a public multiyear evaluation process where people are invited to submit an algorithm for consideration as the standard. The most recent encryption standard selected was AES in 2001:

*NIST is trying to build standards, and it has a standard for the crypto algorithm, which is currently AES. It was selected using a public process where multiple groups submitted algorithms and NIST, representing the consensus of the community, picked a winner. It wasn't dictated from on high and there were no secret criteria. The AES algorithm was the one that most of us thought was the best. Actually, there were several we thought were good candidates, and NIST picked one. But there is a lot of trust in the process because it is public, open, and international. SHA-3, the new secure hash standard, used the same sort of process.*

Schneier designed and submitted an algorithm called Twofish as one of the entries in the AES competition. Twofish was one of the finalists, but NIST selected an algorithm called Rijndael as the AES standard:

*AES was an interesting process. It started with 64 algorithms, of which 56 met the submission criteria. Then NIST whittled it down to 15 or 16, and then in the next round whittled it down to five,*

*and then eventually to one. So it was a constant winnowing process. My Twofish algorithm made it into the top five. There were no bad algorithms among the finalists. The differences were more about security margin and implementability in the hardware versus embedded systems or otherwise constrained systems. To me, it came down to three algorithms. I thought they were all good choices. Twofish was one of the three and Rijndael (the eventual winner) was another.*

For Bruce, winning the competition was less important than making sure the selected algorithm was something we all could trust:

*While it would have been great to be the winner, I think there was a lot of value in NIST picking a non-US algorithm. By choosing an algorithm created by cryptographers from Belgium, NIST said to the world that it picked what it thought was the best algorithm, not just an American one. That was an important consideration I hadn't thought of at the time. So I can't fault this process at all. It was really fun to participate and I would do it again. I also participated in the SHA-3 competition with an algorithm called Skein. Someone else won, which was fine with me.*

Given that these competitions take several years and it could be more than a decade between competitions, they make a big impact in the security research community:

*These competitions are lots of fun for cryptographers and students. They give students lots of targets. One of the hard things as a crypto student is that you have to learn to break stuff. The only way to learn how to make things is by breaking them. These competitions allow students to start breaking things that have not been broken before. They can publish papers and gain credibility in the field.*

It's a unique aspect of security research that the "coin of the realm" is poking holes in results produced by your colleagues in the field:

*You go to a security or crypto conference and there are going to be papers from people who break each other's stuff, so you need a thick skin. You have to understand that we are all learning. I produce a protocol and you break it. Sure, I'm unhappy, but I've learned something—and so have you and so has everyone else. That knowledge is more important than my particular creation surviving. Anyone can invent a cryptosystem that he or she can't break. The only way to get better at design is by breaking others' designs.*

Once we accept the fact that there is no unbreakable cryptography and certainly no unbreakable computing system, the goal is to get to the point where we have the best possible algorithms and a high level of trust in them. The security field has been well served by the cryptographic algorithms produced through the NIST standardization process. By using an open process and encouraging competition and critique from all participants, we have the best chance of developing solid and trusted cryptographic algorithms.

**CHARLES SEVERANCE** is a clinical associate professor and teaches in the School of Information at the University of Michigan, and is Computer's multimedia editor. Follow him on Twitter @drchuck or contact him at csev@umich.edu.