
Photo by Martin Gundersen

Cryptography after the Aliens Land

Bruce Schneier
Harvard University

Quantum computing is a new way of computing—one that could allow humankind to perform computations that are simply impossible using today's computing technologies. It allows for very fast searching, something that would break some of the encryption algorithms we use today. And it allows us to easily factor large numbers, something that would break the RSA cryptosystem for any key length.

This is why cryptographers are hard at work designing and analyzing "quantum-resistant" public-key algorithms. Currently, quantum computing is too nascent for cryptographers to be sure of what is secure and what isn't. But even assuming aliens have developed the technology to its full potential, quantum computing doesn't spell the end of the world for cryptography. Symmetric cryptography is easy to make quantum-resistant, and we're working on quantum-resistant public-key algorithms. If public-key cryptography ends up being a temporary anomaly based on our mathematical knowledge and computational ability, we'll still survive. And if some inconceivable alien technology can break all of cryptography, we still can have secrecy based on information theory—albeit with significant loss of capability.

At its core, cryptography relies on the mathematical quirk that some things are easier to do than to undo. Just as it's easier to smash a plate than to glue all the pieces back together, it's much easier to multiply two prime numbers together to obtain one large number than it is to factor that large number back into two prime numbers. Asymmetries of this kind—one-way functions and trap-door one-way functions—underlie all of cryptography.

To encrypt a message, we combine it with a key to form ciphertext. Without the key, reversing the process is more difficult. Not just a little more difficult, but astronomically more difficult. Modern encryption algorithms are so fast that they can secure your entire hard drive without any noticeable slowdown, but that encryption can't be broken before the heat death of the universe.

With symmetric cryptography—the kind used to encrypt messages, files, and drives—that imbalance is exponential, and is amplified as the keys get larger. Adding one bit of key increases the complexity of encryption by less than a percent (I'm hand-waving here) but doubles the cost to break. So a 256-bit key might seem only twice as complex as a 128-bit key, but (with our current knowledge of mathematics) it's 340,282,366,920,938,463,463,374,607,431,768,211,456 times harder to break.

Public-key encryption (used primarily for key exchange) and digital signatures are more complicated. Because they rely on hard mathematical problems like factoring, there are more potential tricks to reverse them. So you'll see key lengths of 2,048 bits for RSA, and 384 bits for algorithms based on elliptic curves. Here again, though, the costs to reverse the algorithms with these key lengths are beyond the current reach of humankind.

This one-wayness is based on our mathematical knowledge. When you hear about a cryptographer "breaking" an algorithm, what happened is that they've found a new trick that makes reversing easier. Cryptographers discover new tricks all the time, which is why we tend to use key lengths that are longer than strictly necessary. This is true for both symmetric and public-key algorithms; we're trying to future-proof them.

Quantum computers promise to upend a lot of this. Because of the way they work, they excel at the sorts of computations necessary to reverse these one-way functions. For symmetric cryptography, this isn't too bad. Grover's algorithm shows that a quantum computer speeds up these attacks to effectively halve the key length. This would mean that a 256-bit

continued on p. 86

 Copublished by the IEEE Computer and Reliability Societies

continued from p. 88

key is as strong against a quantum computer as a 128-bit key is against a conventional computer; both are secure for the foreseeable future.

For public-key cryptography, the results are more dire. Shor's algorithm can easily break all of the commonly used public-key algorithms based on both factoring and the discrete logarithm problem. Doubling the key length increases the difficulty to break by a factor of eight. That's not enough of a sustainable edge.

There are a lot of caveats to those two paragraphs, the biggest of which is that quantum computers capable of doing anything like this don't currently exist, and no one knows when—or even if—we'll be able to build one. We also don't know what sorts of practical difficulties will arise when we try to implement Grover's or Shor's algorithms for anything but toy key sizes. (Error correction on a quantum computer could easily be an unsurmountable problem.) On the other hand, we don't know what other techniques will be discovered once people start working with actual quantum computers. My bet is that we will overcome the engineering challenges, and that there will be many advances and new techniques—but they're going to take time to discover and invent. Just as it took decades for us to get supercomputers in our pockets, it will take decades to work through all the engineering problems necessary to build large-enough quantum computers.

In the short term, cryptographers are putting considerable effort into designing and analyzing quantum-resistant algorithms, and those are likely to remain secure for decades. This is a necessarily slow process, as both good cryptanalysis transitioning standards take time. Luckily, we have time. Practical quantum computing seems to always remain "ten years in the future," which means no one has any idea.

After that, though, there is always the possibility that those algorithms will fall to aliens with better quantum techniques. I am less worried about symmetric cryptography, where Grover's algorithm is basically an upper limit on quantum improvements, than I am about public-key algorithms based on number theory, which feel more fragile. It's possible that quantum computers will someday break all of them, even those that today are quantum resistant.

If that happens, we will face a world without strong public-key cryptography. That would be a huge blow to security and would break a lot of stuff we currently do, but we could adapt. In the 1980s, Kerberos was an all-symmetric authentication and encryption system. More recently, the GSM cellular standard does both authentication and key distribution—at scale—with only symmetric cryptography. Yes, those systems have centralized points of trust and failure, but it's possible to design other systems that use both secret splitting and secret sharing to minimize that risk. (Imagine that a pair of communicants get a piece of their session key from each of five different key servers.) The ubiquity of communications also makes things easier today. We can use out-of-band protocols where, for example, your phone helps you create a key for your computer. We can use in-person registration for added security, maybe at the store where you buy your smartphone or initialize your Internet service. Advances in hardware may also help to secure keys in this world. I'm not trying to design anything here, only to point out that there are many design possibilities. We know that cryptography is all about trust, and we have a

lot more techniques to manage trust than we did in the early years of the Internet. Some important properties like forward secrecy will be blunted and far more complex, but as long as symmetric cryptography still works, we'll still have security.

It's a weird future. Maybe the whole idea of number theory–based encryption, which is what our modern public-key systems are, is a temporary detour based on our incomplete model of computing. Now that our model has expanded to include quantum computing, we might end up back to where we were in the late 1970s and early 1980s: symmetric cryptography, code-based cryptography, Merkle hash signatures. That would be both amusing and ironic.

Yes, I know that quantum key distribution is a potential replacement for public-key cryptography. But come on—does anyone expect a system that requires specialized communications hardware and cables to be useful for anything but niche applications? The future is mobile, always-on, embedded computing devices. Any security for those will necessarily be software only.

There's one more future scenario to consider, one that doesn't require a quantum computer. While there are several mathematical theories that underpin the one-wayness we use in cryptography, proving the validity of those theories is in fact one of the great open problems in computer science. Just as it is possible for a smart cryptographer to find a new trick that makes it easier to break a particular algorithm, we might imagine aliens with sufficient mathematical theory to break all encryption algorithms. To us, today, this is ridiculous. Public-key cryptography is all number theory, and potentially vulnerable to more mathematically inclined aliens. Symmetric cryptography is so much nonlinear muddle, so easy to make more complex, and so easy to increase key length, that this future is unimaginable. Consider an AES variant with a 512-bit block and key size, and 128 rounds. Unless mathematics is fundamentally different than our current understanding, that'll be secure until computers are made of something other than matter and occupy something other than space.

But if the unimaginable happens, that would leave us with cryptography based solely on information theory: one-time pads and their variants. This would be a huge blow to security. One-time pads might be theoretically secure, but in practical terms they are unusable for anything other than specialized niche applications. Today, only crackpots try to build general-use systems based on one-time pads—and cryptographers laugh at them, because they replace algorithm design problems (easy) with key management and physical security problems (much, much harder). In our alien-ridden science-fiction future, we might have nothing else.

Against these godlike aliens, cryptography will be the only technology we can be sure of. Our nukes might refuse to detonate and our fighter jets might fall out of the sky, but we will still be able to communicate securely using one-time pads. There's an optimism in that. ■

Bruce Schneier is a lecturer and Fellow at the Harvard Kennedy School, and special advisor to IBM Security. His new book is *Click Here to Kill Everybody: Security and Survival in a Hyper-connected World*. Contact him via www.schneier.com