KIM ZETTER. SECURITY 09.24.13 6:30 AM

WIRED

HOW A CRYPTO 'BACKDOOR' PITTED THE TECH WORLD AGAINST THE NSA

1Illustration: alengo/Getty Images

IN AUGUST 2007, a young programmer in Microsoft's Windows security group stood up to give a five-minute turbo talk at the annual Crypto conference in Santa Barbara.

It was a Tuesday evening, part of the conference's traditional rump session, when a hodge-podge of short talks are presented outside of the conference's main lineup. To

draw attendees away from the wine and beer that competed for their attention at that hour, presenters sometimes tried to sex up their talks with provocative titles like "Does Bob Go to Prison?" or "How to Steal Cars – A Practical Attack on KeeLoq" or "The Only Rump Session Talk With Pamela Anderson."

Dan Shumow and his Microsoft colleague Niels Ferguson titled theirs, provocatively, "On the Possibility of a Back Door in the NIST SP800-90 Dual Ec Prng." It was a title only a crypto geek would love or get.

The talk was only nine slides long (.pdf). But those nine slides were potentially dynamite. They laid out a case showing that a new encryption standard, given a stamp of approval by the U.S. government, possessed a glaring weakness that made an algorithm in it susceptible to cracking. But the weakness they described wasn't just an average vulnerability, it had the kind of properties one would want if one were intentionally inserting a backdoor to make the algorithm susceptible to cracking by design.

For such a dramatic presentation — by mathematicians' standards — the reaction to it was surprisingly muted. "I think folks thought, 'Well that's interesting,' and, 'Wow, it looks like maybe there was a flaw in the design,'" says a senior Microsoft manager who was at the talk. "But there wasn't a huge reaction."

Six years later, that's all changed.

Early this month the New York Times drew a connection between their talk and memos leaked by Edward Snowden, classified Top Secret, that apparently confirms that the weakness in the standard and so-called Dual_EC_DRBG algorithm was indeed a backdoor. The Times story implies that the backdoor was intentionally put there by the NSA as part of a $250-million, decade-long covert operation by the

agency to weaken and undermine the integrity of a number of encryption systems used by millions of people around the world.

The Times story has kindled a firestorm over the integrity of the byzantine process that produces security standards. The National Institute of Standards and Technology, which approved Dual_EC_DRBG and the standard, is now facing a crisis of confidence, having been forced to re-open the standard for public discussion, while security and crypto firms scramble to unravel how deeply the suspect algorithm infiltrated their code, if at all. On Thursday, corporate giant RSA Security publicly renounced Dual_EC_DRBG, while also conceding that its commercial suite of cryptographic libraries had been using the bad algorithm as its default algorithm for years.

But beneath the flames, a surprising uncertainty is still smoldering over whether Dual_EC_DRBG really is backdoored. The Times, crypto experts note, hasn't released the memos that purport to prove the existence of a backdoor, and the paper's direct quotes from the classified documents don't mention any backdoor in the algorithm or efforts by the NSA to weaken it or the standard. They only discuss efforts to push the standard through committees for approval.

Jon Callas, the CTO of Silent Circle, whose company offers encrypted phone communication, delivered a different rump session talk at the Crypto conference in 2007 and saw the presentation by Shumow. He says he wasn't alarmed by it at the time and still has doubts that what was exposed was actually a backdoor, in part because the algorithm is so badly done.

"If [NSA] spent $250 million weakening the standard and this is the best that they could do, then we have nothing to fear from them," he says. "Because this was really ham-fisted. When you put on your conspiratorial hat about what the NSA would be

doing, you would expect something more devious, Machiavellian … and this thing is just laughably bad. This is Boris and Natasha sort of stuff."

Indeed, the Microsoft presenters themselves — who declined to comment for this article — didn't press the backdoor theory in their talk. They didn't mention NSA at all, and went out of their way to avoid accusing NIST of anything. "WE ARE NOT SAYING: NIST intentionally put a back door in this PRNG," read the last slide of their deck.

The Microsoft manager who spoke with WIRED on condition of anonymity thinks the provocative title of the 2007 presentation overstates the issue with the algorithm and is being misinterpreted — that perhaps reporters at the Times read something in a classified document showing that the NSA worked on the algorithm and pushed it through the standards process, and quickly took it as proof that the title of the 2007 talk had been right to call the weakness in the standard and algorithm a backdoor.

But Paul Kocher, president and chief scientist of Cryptography Research, says that regardless of the lack of evidence in the Times story, he discounts the "bad cryptography" explanation for the weakness, in favor of the backdoor one.

"Bad cryptography happens through laziness and ignorance," he says. "But in this case, a great deal of effort went into creating this and choosing a structure that happens to be amenable to attack.

"What's mathematically creative [with this algorithm] is that when you look at it, you can't even prove whether there is a backdoor or not, which is very bizarre in cryptography," he says. "Usually the presence of a backdoor is something you can prove is there, because you can see it and exploit it…. In my entire career in cryptography, I've never seen a vulnerability like this."

National Security Agency headquarters, Fort Meade, Maryland. *Photo: Wikipedia*

It's not the first time the NSA has been accused of installing backdoors. Crypto trapdoors, real and imagined, have been part of NSA lore for decades. In some ways the current controversy echoes the long-ago debate over the first U.S. Data Encryption Standard in the 1970s. The NSA was widely suspected of weakening DES to make it more crackable by the agency by tinkering with a table of numeric constants called an S-Box and shortening the algorithm's key length. In 1994, though, the NSA was exonerated when it turned out that the agency had actually changed the S-Box numbers to harden DES against a code-breaking technique that had been known only within NSA at the time.

In 1995, another case came up that seemed to confirm suspicions about the NSA. The Baltimore Sun reported that year that the NSA had inserted a backdoor into cryptographic machines made by the respected Swiss company Crypto AG, apparently substantiating longstanding rumors to that effect.

Then in 1999, Microsoft inadvertently kicked off another controversy when it leaked its internal name for a cryptographic signing key built into Windows NT. The key was called _NSAKEY, spawning speculation that Microsoft had secretly given the agency the power to write and sign its own updates to Windows NT's crypto engine. Microsoft said this was incorrect, that the key was an internal Microsoft key only and that it was called "_NSAKEY" because the NSA was the technical reviewing authority for U.S. export controls. The key was part of Microsoft's compliance with U.S. export laws.

Suspicions about the NSA and backdoors were lingering in 2006 when Shumow and Ferguson began looking at Dual_EC_DRBG after NIST approved it for inclusion in a standard (.pdf). The standard discussed four federally sanctioned random number generators approved for use in encrypting government classified and unclassified-but-sensitive communication.

Each of the four algorithms was based on a different cryptographic design family. One was based on hash functions, one on so-called HMAC (hash-based message authentication code), one on block ciphers and the fourth one was based on elliptic curves. The NSA had been pushing elliptic curve cryptography for a number of years, and it publicly championed the last one — Dual_EC_DRBG — to be included in the standard.

Elliptic curve algorithms are based on slightly different mathematics than the more common RSA algorithm, and the NSA believes they're the future of cryptography, asserting that elliptic curve algorithms are smaller, faster and offer better security.

But as Shumow and Ferguson examined the properties of the elliptic curve random number generator in the standard, to determine how to incorporate it into the Windows operating system, a couple of strange things stood out. First, the random number generator was very slow – two to three orders of magnitude slower than another algorithm in the standard.

Second, it didn't seem to be very secure.

"There was a property [in it] that seemed to make the prediction-resistance of the algorithm not what you would necessarily want it to be," the Microsoft manager says. In non-geek speak, there was a weakness that made the random number generator not so random.

Good random number generation is at the core of encryption, and a weak RNG can undo the entire encryption system. Random number generators play a role in creating cryptographic keys, in opening secure communications between users and web sites and in resetting passwords for email accounts. Without assured randomness, an attacker can predict what the system will generate and undermine the algorithm.

Shumow and Ferguson found that the obstacles to predicting what the random number generator would generate were low. It wasn't a catastrophic problem, but it seemed strange for a security system being promulgated by the government.

Then they noticed something else.

The standard, which contained guidelines for implementing the algorithm, included a list of constants – static numbers – that were used in the elliptic curve on which the random number generator was based. Whoever generated the constants, which served as a kind of public key for the algorithm, could have generated a second set of numbers at the same time – a private key.

Anyone possessing that second set of numbers would have what's known in the cryptography community as "trapdoor information" – that is, they would be able to essentially unlock the encryption algorithm by predicting what the random number generator generated. And, Shumow and Ferguson realized, they could predict this after seeing as few as 32 bytes of output from the generator. With a very small sample, they could crack the entire encryption system used to secure the output.

"Even if no one knows the secret numbers, the fact that the backdoor is present makes Dual_EC_DRBG very fragile," cryptographer Bruce Schneierwrote at the time, in a piece for WIRED. "If someone were to solve just one instance of the algorithm's elliptic-curve problem, he would effectively have the keys to the kingdom. He could then use it for whatever nefarious purpose he wanted. Or he could publish his result, and render every implementation of the random-number generator completely insecure."

No one knew who had produced the constants, but it was assumed that because the NSA had pushed the algorithm into the standard, the agency had generated the numbers. The spy agency might also, then, have generated a secret key.

Schneier called it "scary stuff indeed," but he also said at the time that it made no sense as a backdoor, since it was so obvious to anyone who looked at the algorithm and standard that there was this flaw in it. As a result, developers of web sites and software applications wouldn't use it to help secure their products and systems, he said.

But in fact, many developers did use it.

The U.S. government has enormous purchasing power, and vendors soon were forced to implement the suspect standard as a condition of selling their products to federal agencies under so-called FIPS certification requirements. Microsoft added

support for the standard, including the elliptic curve random-number generator, in a Vista update in February 2008, though it did not make the problematic generator the default algorithm.

Asked why Microsoft supported the algorithm when two of its own employees had shown it to be weakened, a second Microsoft senior manager who spoke with WIRED said that while the weakness in the algorithm and standard was "weird" it "wasn't a smoking gun." It was more of an "odd property."

Microsoft decided to include the algorithm in its operating system because a major customer was asking for it, because it had been sanctioned by NIST, and because it wasn't going to be enabled as the default algorithm in the system, thus having no impact on other customers.

"In fact it is nearly impossible for any user to implement or to get this particular random number generator instantiating on their machines without going into the guts of the machine and reconfiguring it," he says.

Other major companies, like Cisco and RSA, added it as well. NIST in fact provides a lengthy list of companies that have included it in their libraries, though the list doesn't say which companies made it the default algorithm in their library or which products have been developed that invoke the algorithm.

A Cisco spokesman told WIRED that the algorithm was implemented in its standard crypto library around mid-2012, a library that is used in more than 120 product lines, but the algorithm is not the default, and the default algorithm cannot be changed by users. The company is currently completing an internal audit of all of its products that leverage the NIST standard.

RSA, however, made the algorithm the default in its BSafe toolkit for Java and C developers until this week when it told WIRED that it was changing the default

following the renewed controversy over it. The company sent an advisory to developer customers "strongly" urging them to change the default to one of a number of other random number generator algorithms RSA supports. RSA also changed the default on its own end in BSafe and in an RSA key management system. The company is currently doing an internal review of all of its products to see where the algorithm gets invoked in order to change those.

RSA actually added the algorithm to its libraries in 2004 or 2005, before NIST approved it for the standard in 2006 and before the government made it a requirement for FIPS certification, says Sam Curry, the company's chief technology officer. The company then made it the default algorithm in BSafe and in its key management system after the algorithm was added to the standard. Curry said that elliptic curve algorithms were all the rage at the time and RSA chose it as the default because it provided certain advantages over the other random number generators, including what he says was better security.

"Cryptography is a changing field. Some algorithms go up and some come down and we make the best decisions we can in any point in time," he says."A lot of the hash-based algorithms were getting struck down by some weaknesses in how they chose numbers and in fact what kind of sample set they chose for initial seeding. From our perspective it looked like elliptic curve would be immune to those things."

Curry says the fact that the algorithm is slower actually provides it with better security in at least one respect.

"The length of time that you have to gather samples will determine the strength of your random number generation. So the fact that it's slower sometimes gives it a wider sample set to do initial seeding," he says. "Precisely because it takes a little longer, it actually winds up giving you more randomness in your initial seeding, and that can be an advantage."

Despite the renewed controversy over the algorithm and standard, Microsoft managers say they still don't think the weaknesses constitute an intentional backdoor.

Callas agrees. He thinks it is simply bad cryptography that was included in the standard to round-out the selection so that there would be at least one elliptic curve algorithm in the standard.

But one advantage to having the algorithm supported in products like Vista — and which may be the reason the NSA pushed it into the standard — is that even if it's not the default algorithm for encryption on a system, as long as it's an option on the system, an intruder, like the NSA, can get into the system and change the registry to make it the default algorithm used for encryption, thereby theoretically making it easy for the NSA to undermine the encryption and spy on users of the machine.

Schneier says this is a much more efficient and stealth way of undermining the encryption than simply installing a keystroke logger or other Trojan malware that could be detected.

"A Trojan is really, really big. You can't say that was a mistake. It's a massive piece of code collecting keystrokes," he said. "But changing a bit-one to a bit-two [in the registry to change the default random number generator on the machine] is probably going to be undetected. It is a low conspiracy, highly deniable way of getting a backdoor. So there's a benefit to getting it into the library and into the product."

To date, the only confirmation that the algorithm has a backdoor comes in the Times story, based on NSA documents leaked by Edward Snowden, which the Times and two other media outlets saw.

"[I]nternal memos leaked by a former NSA contractor, Edward Snowden, suggest that the NSA generated one of the random number generators used in a 2006 NIST standard — called the Dual EC DRBG standard — which contains a back door for the NSA," the Times wrote.

An editorial published by the Times this weekend re-asserted the claim: "Unbeknown to the many users of the system, a different government arm, the National Security Agency, secretly inserted a 'back door' into the system that allowed federal spies to crack open any data that was encoded using its technology."

But all of the quotes that the Times published from the memos refer to the NSA getting the standard passed by an international standards body; they do not say the NSA intentionally weakened the algorithm and standard, though the Times implies that this is what the memos mean by tying them to the 2007 presentation by Shumow and Ferguson.

NIST has denied any knowledge of a backdoor and has also denied that the NSA authored its standard. The institute has, however, re-opened the standard for public comment as a result of the controversy and "strongly" urged against using the algorithm in question until the matter could be resolved. The public comments period will close Nov. 6.

Even without more explicit confirmation that the weaknesses in the algorithm and standard constitute a backdoor, Kocher and Schneier believe they do.

"It is extraordinarily bad cryptography," says Kocher. "If you look at the NSA's role in creating standards [over the years] and its general cryptographic sophistication, none of it makes sense if there isn't a backdoor in this."

Schneier agrees and says the NSA has done too many other things for him to think, when he sees government-mandated crypto that's weak, that it's just by accident.

"If we were living in a kinder world, that would be a plausible explanation," he says. "But we're living in a very malicious world, it turns out."

He adds that the uncertainty around the algorithm and standard is the worst part of the whole matter.

"This is the worst problem that the NSA has done," Schneier says. "They have so undermined the fundamental trust in the internet, that we don't know what to trust. We have to suspect everything. We're never sure. That's the greatest damage."

Republished for educational purposes only.