Can Elliptic Curve Cryptography Be Trusted?

A Brief Analysis of the Security of a Popular Cryptosystem

Many smart card, cell phone, Internet of Things (IoT) and Bitcoin businesses have already implemented elliptic curve cryptography (ECC), and for good reason. This asymmetric encryption and decryption method is shown by the US National Institute of Standards and Technology (NIST) and third-party studies to significantly outperform its biggest competitors, offering significantly shorter keys, lower central processing unit (CPU) consumption and lower memory usage.[1, 2]

As security is an instrumental aspect of cryptography, it is important to evaluate every cryptogram carefully—not only for efficiency, but also for imperviousness against all kinds of cryptographic attacks. There are multiple ways to assess the security capabilities of ECC to determine if it is a worthwhile venture.

What vulnerabilities or possible weaknesses in design exist with ECC? Can ECC withstand the test of time, and what implementation issues does it face?

ECC for Security

Although there is no such thing as a perfect, widely applicable and unbreakable cryptosystem, there are many ways to keep data safe when at rest and when in motion. There exist a variety of classes of cryptoalgorithms, including hashing algorithms, symmetric cryptoalgorithms and asymmetric cryptoalgorithms. ECC, just like RSA, falls under the asymmetric algorithm (public/private key) classification. This type of cryptogram solves a variety of problems, one of which is allowing two nodes or individuals who have never communicated to each other before to pass information to each other in a secure manner. These algorithms are also a crucial cog in the mechanism of many protocols, standards, services and infrastructures. Bitcoin, X.509/PKI, Transport Layer Security/Secure Sockets Layer (TLS/SSL), Internet Key Exchange (IKE), Secure Shell (SSH), Domain Name System Security Extensions (DNSSEC), Pretty Good Privacy/Gnu Privacy Guard (PGP/GPG), Secure/Multipurpose Internet Mail Extensions (S/MIME), RFC 3161, most things with digital signatures (e.g., digitally signed portable document formats [PDFs]), Z and Real Time Transport Protocol (ZRTP), and Secure Internet Live Conferencing (SILC) all deeply rely on asymmetric encryption and decryption in one way or another.

Once it is established that asymmetric encryption is needed, it is time to choose the best-fitting tool. The statistics look great for ECC. NIST-recommended key-size tables depict the shorter key advantage ECC has. For an equivalent symmetric key size of 80 bits, RSA requires 1,024 bits, while ECC requires 160 bits (a 3:1 ratio). When the symmetric key size grows to 256 bits, the ratio jumps up to 64:1. Thus, elliptic curves are computationally lighter for longer keys.[3] Further studies show that the time different processors take to encrypt and/or decrypt data can be 400 times faster for ECC than for an equivalent RSA length.[4]

The security side of ECC is complex. As of today, there are numerous standards defining and governing it, including the American National Standards Institute's (ANSI) X9.62, the Institute of Electrical and Electronics Engineers' (IEEE) P1363, the Standards for Efficient Cryptography Group (SECG), NIST's Federal Information Processing Standards (FIPS) 186-2, ANSI X9-63, Brainpool, the US National Security Agency's (NSA's) Suite B, and ANSI FRP256V1.

ECC is adaptable to a wide range of cryptographic schemes and protocols, such as the Elliptic Curve Diffie-Hellman (ECDH), the Elliptic Curve Digital Signature Algorithm (ECDSA) and the Elliptic Curve Integrated Encryption Scheme (ECIES). The mathematical inner workings of ECC cryptography and cryptanalysis security (e.g., the Weierstrass equation that describes elliptical curves, group theory, quadratic twists, quantum mechanics behind the Shor attack and the elliptic-curve discrete-logarithm problem) are complex.

Veronika Stolbikova
Currently works as a principal infrastructure analyst (information security risk management) at Quintiles. Her areas of interest include security posture and vulnerability assessments, security risk management, secure development, and cryptography.

Do you have something to say about this article?
Visit the *Journal* pages of the ISACA web site *(www.isaca.org/journal)*,find the article and click on the Comments link to share your thoughts.

Currently Known Attacks

There are a significant number of potential vulnerabilities to elliptic curves, such as side-channel attacks and twist-security attacks. These attacks threaten to invalidate the security ECC aims to provide to private keys.

Side-channel attacks generally occur when measurements are made on the physical implementation of a cryptosystem, resulting in leaks of information. Side-channel analysis includes a variety of attacks, such as simple timing attacks, simple power attacks, differential power attacks and fault analysis.[5] During timing attacks, for instance, the malicious user measures the difference in time between observed peaks in power consumption with an oscilloscope. Relying on the fact that different operations or input values have a significant time variance, the attacker can deduce the secret key. Power attacks, on the other hand, are similar to timing attacks except for the fact that the actual shape and amplitude of voltage peaks is analyzed by the attacker. A variety of power attacks exist, including simple power analysis (SPA) and differential power analysis (DPA).

Simple countermeasures exist for all types of side-channel attacks. Both timing and simple-power attacks can be prevented with the implementation of the Montgomery power ladder (a scalar multiplication technique used to compute) into the ECC instead of using one of the other similar techniques (e.g., double-and-add, sliding window). Not only do Montgomery ladders have the advantage of providing fast scalar multiplication for ECC, but they also tend to behave regularly, masking the computation against timing and simple power-side-channel attacks.[6] Unfortunately, not all existing ECC curves support the use of ladders. The number of curves that do not support this technique is vast (e.g., Anomalous, NIST P-224, BN [2,254], BrainpoolP256t1, ANSSI FRP256v1), so it is important to check if one's ECC implementation uses a curve that both implements and supports Montgomery ladders.[7] Furthermore, simple timing attacks can be prevented by inserting dummy adds into the algorithm to act as an ignored variable; this makes the number of process operations to be performed the same regardless of the value of the secret key.[8] DPA-type side-channel attacks can be prevented in a variety of manners, including adding significant entropy to the secret key, disguising group points and using randomized projective coordinates.[9]

Another category of attacks on elliptic curves is known as twist-security (fault) attacks. Such attacks usually succeed when several conditions are met, and they all lead to the leakage of the victim's private key. Typically during a twist attack, the malicious party shares a carefully selected public key that does not lie on the agreed-upon ECC curve and that will lead to a shared key that can be easily reversed. After the victim computes a shared key (computed out of the victim's private key and the malicious public key) and computes a hash out of the shared key, the malicious party is able to extract the victim's secret key. Twist attacks can be broken down into many subcategories including small-subgroup attacks, invalid-curve attacks and invalid-curve attacks against Montgomery ladders. Small-subgroup attacks make it possible to simply enumerate the victim's private key by using a carefully selected point of small order as the public key. During the much more severe invalid curve attacks, the attacker picks a point of small

order that lies on an elliptical curve with a different constant coefficient. However, as invalid-curve attacks are limited by the use of ladders such as the aforementioned Montgomery ladder, specific twist attacks exist against those as well.[10] However, twist-security attacks generally are fairly easily mitigated by careful choices of curves and validation of various parameters.

Possible Future Attacks

While quantum computing is already facing a large variety of problems, such as its poor decoherence rates, error correction issues, state preparation issues and problems with quantum gates,[11] its advancement may bring additional challenges to ECC once it becomes a technological reality instead of the theoretical concept it is today. As quantum computers continue making strides in development, businesses must consider if quantum computers have potential implications on their ECC implementations.

Quantum computing will provide two major cryptanalytic weapons: Shor's and Grover's algorithms (and variations thereof). Shor attacks make factoring easy, essentially making it trivial for the attacker to uncover the secret key in an asymmetric cryptosystem. Grover attacks make brute-forcing easier by creating a uniform superposition over all possible inputs, destructively interfering states that are invalid and, consequently, finding inputs that satisfy a given function. Shor's and Grover's algorithms may have major implications not only for ECC, but also for asymmetric cryptography altogether. Furthermore, ECC's advantage in shorter key lengths in classical computing will prove to be a disadvantage in quantum computing. ECC will be easier to break than RSA cryptosystems due to a lower qubits (quantum equivalents of traditional bits) requirement.[12] While quantum computers present a frightening threat to ECC and asymmetric cryptography, this is not imminent, as quantum computers need to first overcome some very difficult physical limitations.

Issues With ECC Implementation

History has shown that, although a secure implementation of the ECC curve is theoretically possible, it is not easy to achieve. In fact, incorrect implementations can lead to ECC private key leaks in a number of scenarios. Such leaks can occur when incorrect results are calculated and when the input does not end up on the selected curve. Furthermore, they can happen when branch-timing errors occur or when cache-timing errors occur. In a nutshell, a lot of things can go wrong while ECC is being implemented.[13]

There are numerous examples of how failed implementation of ECC algorithms resulted in significant vulnerabilities in the cryptographic software. A great example is that of the Sony ECDSA security disaster. Although Sony used ECDSA to sign software for their PlayStation game console, they did not properly implement the algorithm. Using static parameters instead of random ones made Sony's implementation of the algorithm solvable and subsequently useless.[14]

Furthermore, there are examples of improper implementation of ECC in OpenSSL that resulted in common vulnerabilities, such as Common Vulnerability and Exposure (CVE)-2014-3572, CVE-2014-0076 and CVE-2008-5077. These vulnerabilities range from omission of the server key exchange message to malformed signatures. Worse, such issues can lead to an unauthenticated, remote attacker gaining access to Secure Sockets Layer (SSL) private keys. Improper implementation issues are a frightening security issue and must be tackled through security code review, static code analysis and penetration testing.

Possible NSA Backdoor

Over the last 10 years, there has been serious media and security community speculation that the NSA inserted a backdoor into one of the ECC standards, undermining its strength.[15] While there are currently many other third-party Cryptographically Secure

Pseudo-random Number Generator (CSPRNG) and ECC standards in existence that remain outside of the scope of this issue, the suspicions first fell on the Dual Elliptic Curve Deterministic Random Bit Generator (Dual_EC_DRBG) elliptic curve pseudo-random generator that was used in the algorithm. One of the weaknesses publicly identified at the time had all the markings of a purposefully designed CSPRNG backdoor.[16] A 2013 Reuters report of a secret US $10 million deal with RSA only served to fuel these fires.[17] After this revelation and much public debate, Dual_EC_DRBG was excluded from the standards and is no longer used.

> **"There is no guarantee that any one team could efficiently find all existing and yet-to-be-discovered weaknesses in a cryptosystem."**

However, there are now similar suspicions about NIST Standard Curves. Since the Edward Snowden revelations, there has been significant concern that the ECC pseudo-random number generator was fabricated to inject an NSA backdoor into ECC cryptography.[18] However, the debate is still ongoing on this subject. Some cryptographers suspect that curves were deliberately chosen as having a mathematical weakness known only to the NSA. Others argue that some security considerations were not widely understood at the time the NIST curves were introduced and that some security issues were due to NIST using the US Secure Hash Algorithm 1 (SHA1) to generate algorithm parameters.

Test of Time

All cryptographers work toward a common goal: to create a cryptosystem that is too hard to break. In a sense, one could consider a cryptosystem's resistance capability to malicious attacks as its quality. However, while other products, such as cars, can be tested for quality by their own manufacturer or approved third parties, there is no guarantee that any one team could efficiently find all existing and yet-to-be-discovered weaknesses in a cryptosystem. Thus, the security community generally recommends opening up new cryptoalgorithms for the world to test the system against various types of threats. Only cryptosystems that can survive extensive community testing over time can be considered as having withstood the test of time. Equally, most security analysts strongly advise against using security through obscurity (relying on the algorithm not to be known to the attacker).[19]

ECC's strength can be analyzed by determining how well it has withstood the test of time. For example, ECC has faced multiple successful and unsuccessful brute-force attacks. In 2004, a team of mathematicians with 2,600 computers that were used over a period of 17 months completed the Certicom Elliptic Curve Cryptography (ECC) 2-109 challenge.[20] In 2009, the 112-bit prime ECDLP was solved using 200 PlayStation 3 consoles.[21] However, to date, cryptanalysts believe that the 160 bit-prime field ECC should remain secure against public attempts until at least 2020.[22]

For the first 30 or so years of ECC's existence, elliptical curves in cryptography were analyzed and experimented with mostly for theoretic and aesthetic reasons. However, during the 1990s, ECC rose in popularity. This resulted both in publicity backlash and significant scrutiny of ECC by opponents attempting to find flaws in it. While the debate between RSA and ECC continued, the latter cryptosystem finally achieved status as an accepted standard. In the end, however, ECC did not significantly rise to fame until the NSA published "The Case for Elliptic Curve Cryptography" in 2005.[23] Nonetheless, it can be said that ECC has been available for everyone to test for quite some time now and that the public should be fairly comfortable that ECC is not merely based on security through obscurity.

Conclusion

Despite the significant debate on whether there is a backdoor into elliptic curve random number generators, the algorithm, as a whole, remains fairly secure. Although there are several popular vulnerabilities in side-channel attacks, they are easily mitigated through several techniques. Quantum attacks loom over ECC, but they are yet to be widely available. Although twist-security attacks can threaten ECC, they can be militated against. Furthermore, although longer ECC keys are broken into publicly every now and then, the same is true for all other popular algorithm types. But no matter how secure ECC is theoretically, it must be properly implemented. History has shown that such a thing

is not trivial, as large teams and corporations have failed to achieve this goal. Above everything else, the aforementioned reality highlights the necessity for proper testing of both security and proper implementation of the algorithm.

Endnotes

1 National Security Agency (NSA), "The Case for Elliptic Curve Cryptography," USA, 2015
2 Lauter, K.; "Elliptic Curve Cryptography for Wireless Security," Microsoft Corp., 2004, *www.msr-waypoint.com/en-us/um/people/klauter/ieeefinal.pdf*
3 *Op cit*, NSA
4 *Op cit*, Lauter
5 Bar-El, H.; "Introduction to Side Channel Attacks," Discretix Technologies Ltd., 2003
6 Joyce, M.; S.-M. Yen; *et al.*; "The Montgomery Powering Ladder," Cryptographic Hardware and Embedded Systems, CHES 2002, volume 2523, *Lecture Notes in Computer Science*, Springer-Verlag, 2003, p. 291-302, *https://choucroutage.com/Papers/SideChannelAttacks/ches-2002-joye.pdf*
7 Bernstein, D.; T. Lange; *et al.*; "SafeCurves: Choosing Safe Curves for Elliptic-curve Cryptography," *http://safecurves.cr.yp.to*
8 Kadir, S.; A. Sasongko; *et al.*; "Simple Power Analysis Attack Against ECC Processor on FPGA Implementation," 2011, *http://140.98.202.196/xpl/articleDetails. jsp?arnumber=6021757&reload=true&searchWithin=%22Authors%22:.QT.Sasongko,%20A..QT.&newsearch=true*
9 Coron, J. S.; "Resistance Against Differential Power Analysis for Elliptic Curve Cryptosystems, Cryptographic Hardware and Embedded Systems," *Lecture Notes in Computer Science*, vol. 1717, 1999
10 *Op cit*, Bernstein
11 Ponnath, A.; "Difficulties in the Implementation of Quantum Computers," 2006, *http://arxiv.org/pdf/cs/0602096.pdf*
12 Yan, S. Y.; *Quantum Attacks on the Public-Key Cryptosystems*, Springer, USA, 2013
13 *Op cit*, Bernstein
14 The Central Scrutinizer, "Sony's PS3 Security Is Epic Fail—Videos Within," PSX-Scene Forum, 29 December 2010, *http://psx-scene.com/forums/content/sony-s-ps3-security-epic-fail-videos-within-581/?s=68e141dc91333038e2223ee86e3c748f*
15 Schneier, B.; "Did NSA Put a Secret Backdoor in New Encryption Standard?," *Schneier on Security* blog, 15 November 2007, *https://www.schneier.com/essays/archives/2007/11/did_nsa_put_a_secret.html*
16 Schneier, B.; "The NSA Is Breaking Most Encryption on the Internet," *Schneier on Security* blog, 5 September 2013, *https://www.schneier.com/blog/archives/2013/09/the_nsa_is_brea.html#c1675929*
17 Menn, J.; "Exclusive: Secret Contract Tied NSA and Security Industry Pioneer," *Reuters*, 20 December 2013, *www.reuters.com/article/2013/12/20/us-usa-security-rsa-idUSBRE9BJ1C220131220*
18 Hales, C.; "The NSA Back Door to NIST," *Notices of the AMS*, vol. 61, no. 2, *www.ams.org/notices/201402/rnoti-p190.pdf*
19 Douligeris, C.; D. N. Serpanos; "Network Security: Current Status and Future Directions," IEEE, 2007
20 Certicom, "Certicom Announces Elliptic Curve Cryptography Challenge Winner," 27 April 2004, *https://www.certicom.com/news-releases/300-solution-required-team-of-mathematicians-2600-computers-and-17-months-*
21 Bos, J.; M. Kaigara; *et al.*; "PlayStation 3 Computing Breaks 2^60 Barrier 112-bit Prime ECDLP Solved," *Laboratory for Cryptological Algorithms*, 25 November 2015, *http://lacal.epfl.ch/112bit_prime*
22 Bos, J.; M. Kaigara; *et al.*; "On the Security of 1024-bit RSA and 160-bit Elliptic Curve Cryptography," Microsoft Research, 1 September 2009, *http://lacal.epfl.ch/files/content/sites/lacal/files/papers/ecdl2.pdf*
23 Koblitz, A. H.; N. Koblitz; A. Menezes; "Elliptic Curve Cryptography: The Serpentine Course of a Paradigm Shift," *Journal of Number Theory*, vol. 131, iss. 8, 2011, p. 781-814 *www.sciencedirect.com/science/article/pii/S0022314X09000481*