Applications of Elliptic Curve Cryptography

A light introduction to elliptic curves and a survey of their applications

R. Harkanson
Computer Science dept.
University of Nevada Las Vegas
Las Vegas, NV, USA
harkanso@unlv.nevada.edu

Y. Kim
Computer Science dept.
University of Nevada Las Vegas
Las Vegas, NV, USA
yoohwan.kim@unlv.edu

ABSTRACT

Elliptic curve cryptography (ECC) is a relatively newer form of public key cryptography that provides more security per bit than other forms of cryptography still being used today. We explore the mathematical structure and operations of elliptic curves and how those properties make curves suitable tools for cryptography. A brief historical context is given followed by the safety of usage in production, as not all curves are free from vulnerabilities. Next, we compare ECC with other popular forms of cryptography for both key exchange and digital signatures, in terms of security and speed. Traditional applications of ECC, both theoretical and in-practice, are presented, including key exchange for web browser usage and DNSSEC. We examine multiple uses of ECC in a mobile context, including cellular phones and the Internet of Things. Modern applications of curves are explored, such as iris recognition, RFID, smart grid, as well as an application for E-health. Finally, we discuss how ECC stacks up in a post-quantum cryptography world.

CCS CONCEPTS

• **Security and Privacy** → **Cryptography**; *Key Management*; Public key techniques

KEYWORDS

Elliptic Curve Cryptography (ECC), Diffie-Hellman (DH), Digital Signature Algorithm (DSA), Elliptic Curve Diffie-Hellman (ECDH), Elliptic Curve DSA (ECDSA)

ACM Reference format:

R. Harkanson and Y. Kim. 2017. Applications of Elliptic Curve Cryptography. In *The 12th Annual Cyber and Information Security Research Conference 2017, Oak Ridge, TN, USA, April 04-06, 2017 (CISRC '17)*, 7 pages.
DOI: 10.1145/3064814.3064818

CISRC '17, April 04-06, 2017, Oak Ridge, TN, USA
 ISBN 978-1-4503-4855-3/17/04...$15.00
DOI: http://dx.doi.org/10.1145/3064814.3064818

1 INTRODUCTION

The demand for computer security is greater today than ever before, as computers are used in many fields where information assurance is a must. Parties must be able to store, send, and receive data from other authenticated parties securely while assuring data integrity. This includes all data, from mobile phones to corporate servers, from storing personal information to connecting with banks, information must be assured.

Elliptic Curve Cryptography (ECC) is a relatively new form of cryptography, dating back to 1985, and has come into wide use since 2005. Its most common utilization is for key exchange, replacing the popular standard Diffie-Hellman (DH) key exchange protocol, or rather expanding upon it. Message signing can be achieved by combining ECC with the Digital Signature Algorithm (DSA). Coupled with something like ElGamal, ECC can be used for actual encryption as well. There are many applications implementing elliptic curves and that is the focus of this writing.

We explore the mathematical structure of Elliptic Curve Cryptography, the shape of curves and which curves are safe and unsafe to use in practice. We see how ECC compares to other competing forms of cryptography and discuss the strengths and weaknesses of using elliptic curves. We investigate how ECC is being used for traditional applications, mobile applications, and newly discovered modern uses of elliptic curves. Finally, we discuss the future of ECC, including how well it will hold up in a post-quantum computing world.

2 STRUCTURE

2.1 Shapes

Mathematically, elliptic curves are cubic curves that are equivalent to tori, topologically. Despite their name, they are not closely related to the ellipse, however they get their name from the elliptic integral.

The Weierstrass normal form, the basic general elliptic curve used for cryptography, is of the form

$$y^2 = x^3 + ax + b \tag{1}$$

which is visualized in Figure 1.

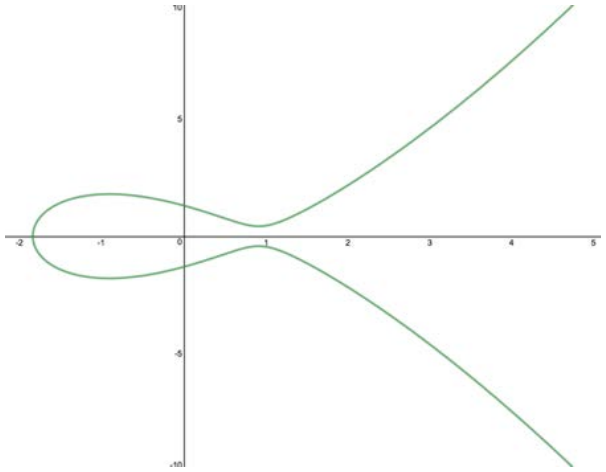

Figure 1: A simple elliptic curve visualization.

Curves of this form are defined by different values for a and b. By modifying these values, the visualization of the curve can expand, contract, or pinch off to be two separate pieces. Curves used for cryptography, in practice, are often defined with very large integer values for a and b.

To avoid a singularity, $4a^3 + 27b^2 \neq 0$ must also be satisfied [1]. By defining the point at infinity to be 0, we can now define a mathematical abelian group [2] over the elliptic curve. Abelian groups have certain properties that give rise to the mathematical operations defined over the curve, making it a useful tool for cryptography and simple for computation. These properties are: commutativity, associativity, closure, an inverse element for every element, and an identity element which is 0.

Elliptic curves may take other forms as well. A couple notable examples are Montgomery curves and Edwards curves. These different forms are also applicable for cryptography.

2.2 Operations

There are two main mathematical operations, with another inferred, defined over elliptic curves which take a single or two different points on the curve as arguments and result in a new point which is also on the curve. These operations are what make ECC a functional form of cryptography, which will be covered in section 2.3.

2.2.1 Point Addition. Point addition is akin to the secant function of two points, resulting in a third point on the curve. For the general elliptic curve form, point addition is defined as

$$P + Q = R \tag{2}$$

where $P = (x_p, y_p)$, $Q = (x_q, y_q)$, and $R = (x_r, y_r)$. The coordinates of R are calculated as

$$x_r = \lambda^2 - x_p - x_q, \;\; y_r = \lambda(x_p - x_r) - y_p \tag{3}$$

where

$$\lambda = \frac{y_q - y_p}{x_q - x_p} \tag{4}$$

2.2.2 Point Doubling. Point doubling is akin to the tangent of the starting point on the curve which results in a second point on the curve. For the general elliptic curve form, point doubling is the same as point addition, except P is coincident with Q and where

$$\lambda = \frac{3x_p^2 + a}{2y_p} \tag{5}$$

where a is the same a from the general curve equation.

2.2.3 Point Multiplication. Point multiplication is simply point addition, possibly with point doubling, applied n times, where n is a scaler positive integer value.

$$nP = \sum_{i=1}^{n} P = R \tag{6}$$

The mathematics of elliptic curves easily lends itself to modular arithmetic. This ensures that the curve gets defined over a field which makes ECC computation feasible. To move from elliptic curves defined over real numbers to finite fields, the mathematical expressions simply need to be modulo p, where p is a prime number, to equally distribute values over the finite modular field, a property of modulo primes.

2.3 Cryptography

Elliptic curves were first suggested and shown to be viable for cryptography independently in 1985 by Neal Koblitz and Victor Miller. At the time, Koblitz expected these curves to be more secure than other cryptosystems because he realized that the elliptic curve variety of the discrete logarithm problem was harder to solve than the standard discrete logarithm problem [3]. Miller saw that elliptic curves can be applied to Diffie-Hellman key exchanges and saw the potential for this field of mathematics to be applied to cryptography [4].

The strength of ECC comes from the mathematical trap door function it utilizes. Just as RSA depends on the factoring problem, cryptography with elliptic curves, just like DH, relies upon the discrete logarithm problem, specifically the elliptic curve discrete logarithm problem. The power of this, and any other trap door function, is that it can be easily solved with the proper information. But without knowing key pieces of the puzzle, the problem is infeasible to solve quickly enough to be significant, even by the most powerful computers of today. Aside from a few minor performance tweaks, attackers are essentially reduced to using brute force attacks. The survival of certain trap door functions in a post-quantum computing world will be discussed later on in this paper.

Elliptic curves, when used for cryptography, are defined over a finite field, as opposed to the real numbers. When combined with modular arithmetic, curves lend themselves to integer computations. Also, now working within a modular field, patterns that exist when curves are defined over real numbers which could be used to speed up attacks now disappear. The logarithm problem that exists for real curves becomes the more-difficult discrete logarithm problem for finite modular curves.

The two major families of elliptic curves used in cryptography are binary curves and prime curves [5]. For hardware applications, binary curves are actually preferred due to the smaller equations and the lack of carries. In software, prime curves are usually faster. Also, prime curves are thought to be more secure as there are certain improvements that aid in attacking binary curves.

2.4 Safe Curves

Not all elliptic curves are safe for production use. Since the rise of ECC there have been certain curves that are believed to be safe as well as curves that are shown to be unsafe. When it comes to curve safety, we are addressing curves that have been shown to be mathematically sound, clear from vulnerabilities, and are surviving the test of time.

Another consideration while choosing a safe curve, in recent years, is to select curves that are free from influence of the NSA or other government organizations. Studies have been done to come up with lists of curves that satisfy these safety requirements [6].

Curves shown to be safe become more standardized as they gain widespread use. One such curve designed to address the curve safety problem very early on is Curve25519. Created by mathematician Daniel J. Bernstein, this curve uses a 32 byte (256 bit) public and private key. The shared secret is 32 bytes as well and is used to authenticate and encrypt messages between the communicating parties [7].

Not only was Curve25519 designed to be safe, but it was also designed to be quick, even on older hardware from before the period the curve was designed. The curve hit record speeds for its day, more than twice as fast as alternatives at the time. Even today, Curve25519 remains a standard curve used for ECC.

3 COMPARISON

ECC, though being mathematically more challenging to understand and implement, offers a nearly-exponential advantage over RSA. This means, for the same amount of bits composing the key, ECC is much harder to crack than RSA, which clearly offers an advantage, so long as a proven curve is being implemented. NIST shows how ECC stacks up against RSA when it comes to bit size of keys for equal security (Table 1) [8].

Table 1: ECC vs RSA: Equivalent Key Sizes

ECC Key Size (bits)	RSA Key Size (bits)	Ratio
163	1024	1:6
256	3072	1:12
384	7680	1:20
512	15360	1:30

Specifically, studies have been done showing how ECC and RSA compare within the domain of QR Code Authentication. The paper describes an authentication process with a one-time password utilizing QR Codes with either RSA3072 or Elliptic Curve Diffie-Hellman (ECDH) and AES256. The comparison is between RSA3072 and ECC-256, specifically, because the two are equivalent in terms of the security level they both provide. The time it takes for one-time password encryption increases with the length of characters. The overall performance of the process shows that ECC can achieve results much quicker than RSA for the same level of encryption. Table 2 shows this comparison. The tests were run on the same Windows Phone SDK emulator with VWGA 512 MB memories [9].

Table 2: Process Performance

Process	RSA3072	ECDH and AES256
Key derivation and SHA-256	7 s	0.8 µs
Byte array conversion	113 ms	48 ms
Shared secret key computation	-	52 ms
Encryption	0.489 ms	0.0285 ms
Total	7.1135 s	100.0286 ms

Most traditional usages of ECC in use today are for either key exchange with DH, the previous paragraph being one example, or for digital signatures when coupled with DSA, named Elliptic Curve Digital Signature Algorithm (ECDSA). ECDSA signature and verification speeds, per bit, are slightly more expensive than standard DSA. From an OpenSSL speed test with a 2.5 GHz Intel Core i7 Mid 2014 MacBook Pro with 16 GB of 1600MHz DDR3 RAM, we obtained the following results for DSA (Table 3) followed by the ECDSA results (Table 4) for the different curves in the popular OpenSSL library [10].

Table 3: OpenSSL DSA Speed

Bits	Sign	Verify
512	56 µs	48 µs
1024	111 µs	108 µs
2048	322 µs	348 µs

Table 4: OpenSSL ECDSA Speed

Bits (Curve)	Sign	Verify
160 (secp160r1)	0.1 ms	0.2 ms
192 (nistp192)	0.1 ms	0.3 ms
224 (nistp224)	0.1 ms	0.1 ms
256 (nistp256)	< 0.1 ms	0.1 ms
384 (nistp384)	0.2 ms	0.8 ms
521 (nistp521)	0.4 ms	0.7 ms
163 (nistk163)	0.2 ms	0.4 ms
233 (nistk233)	0.3 ms	0.5 ms
283 (nistk283)	0.5 ms	0.9 ms
409 (nistk409)	1.1 ms	1.5 ms
571 (nistk571)	2.3 ms	3.5 ms
163 (nistb163)	0.2 ms	0.4 ms
233 (nistb233)	0.3 ms	0.6 ms

283 (nistb283)	0.5 ms	1 ms
409 (nistb409)	1.1 ms	1.6 ms
571 (nistb571)	2.3 ms	3.7 ms

4 TRADITIONAL APPLICATIONS

Within the past decade, ECC has become more and more common on the web. All major mainstream operating systems and web browsers have some form of support for ECC. Also, major server software and security libraries have support for ECC as well. With widespread support comes widespread adoption.

4.1 Key Exchange

The top 100 most visited websites, as of this writing, on Alexa [11], the major website ranking service, have been sampled to discover what form of cryptography each used, if any, on their landing page. We found that, of the top 100, 26 pages did not offer a certified HTTPS connection and 2 additional sites offered an HTTPS connection without a certificate. This leaves 72 websites offering an HTTPS connection with certificate. Of these 72 websites, 69 of which utilized some form of ECC for key exchange. Specifically, 53 websites used Elliptic Curve Diffie-Hellman Ephemeral RSA (ECDHE_RSA) and 16 websites used Elliptic Curve Diffie-Hellman Ephemeral ECDSA (ECDHE_ECDSA) for key exchange. Ephemeral keys are temporary keys generated on the spot that do not need to be authenticated. This leaves only 3 websites of the top 100 that were not implementing ECC of any kind, all 3 used RSA. The 2 sites that offered and HTTPS connection without a certificate both used ECDHE_RSA.

4.2 DNSSEC Validation

The domain name system (DNS), the system which translates web queries into IP addresses that the computer understands, is another area where information assurance and security is a concern. DNS Security Extensions (DNSSEC) was developed to protect DNS servers and transactions from attacks including distributed denial-of-service attacks (DDoS attacks). Researchers suggest that DNSSEC implementing RSA as a signature algorithm is the root of some attacks. It is believed that ECC, although slower at validation, could also be applied to the DNS server to increase security. By modeling DNS resolution and comparing ECC digital signature benchmarks it was shown that even the most computationally intense ECC schemes do not exceed the capacity of a modern CPU core. With ECC DNSSEC can protect against amplification attacks as well as packet fragmentation, making DNS servers more secure and just as reliable [12].

4.3 Signature Server

Due to the light-weight mathematical equations used by ECC, graphics processing units (GPUs) with their limited instruction set can often out-perform CPUs due to the extra bulk of a larger instruction set. One team exploited this to their advantage by making a cryptographic accelerator with a GPU. W. Pan et al. demonstrated that GPUs can be utilized to speed up ECDSA and create a functional universal signature server that is also capable of key agreement and encryption with ECC. They call their server GUESS (GPU-accelerated Universal Elliptic-curve Signature Server). They were able to use multiple GPU threads to achieve a higher throughput to exceed existing prototypes and products with significantly higher performance. In the future, the team plans on testing and improving GUESS and expanding the library of curves in its arsenal [13].

5 MOBILE APPLICATIONS

Considering the growing mobile device market in the past decade, more people have mobile device than ever before. Also considering the many functions these mobile devices can have, private information is sure to be present. Most of these devices have networking capabilities as well, and where there is a connection to the Internet there are attackers trying to exploit security weaknesses. In addition to phones and tablets, the Internet of Things (IoT) is becoming larger than ever. People with smart homes have their lights, air conditioners, garage doors, and doorbells all connected to the Internet. With IoT blurring the boundaries between the digital world and the physical one, security is a must.

5.1 Authentication

Originally, a team of researchers developed a two-factor authentication system for mobile devices, specifically for location-based services. However, another team of researchers, A. G. Reddy et al., analyzed the proposed method and saw many limitations and flaws, including imperfect mutual authentication and vulnerability to insider attacks. The team proposed a new method to replace the older one for the same application, making use of ECC [14].

The proposed protocol with ECC uses two-factor authentication as well and is suitable for practical application by making use of light-weight operations. This new proposition was verified with the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool and shown to lack the security threats that existed in the protocol being replaced.

The new protocol shows user anonymity and untraceability, resistance to replay attacks and man-in-the-middle attacks, protection against stolen smartcard attacks, resistance to foreign agent impersonation and user impersonation attacks, resistance to offline and online password guessing attacks, resistance to privileged insider attacks, resistance to session key compromised attacks, and forward security.

5.2 MANET

A mobile ad hoc network (MANET) is a collection of mobile nodes with the ability to communicate with each other, through each other, wirelessly using radio waves. MANETs are eligible for attack through various means, such as worm hole, black hole, and rushing attacks. To prevent against unauthorized access, P. D. Nikam and V. Raut utilized ECC and Enhanced Adaptive

Acknowledgment (EAACK) to improve the security of MANETs [15].

The researchers have designed and developed an intrusion-detection system specifically for MANETs which demonstrates high malicious behavior detection while not affecting the network performance greatly. They call the system Enhanced Adaptive Acknowledgment with Elliptic Curve Algorithm. The protocol uses light weight ECC Digital Signatures to require all acknowledgment packets to be signed before they are sent and verified once accepted. By using ECC they were able to evade attackers from forging acknowledgement packets, thus avoiding potential attacks.

5.3 Internet of Things

The IoT is another emergent trend in the past few years that is made possible by the large number of wireless networks around the world. Like any newer technology, many IoT devices are open to security threats. One team of researchers set out to solve a part of the IoT security problem by introducing a one-time password (OTP) authentication scheme for IoT devices using ECC. ECC was chosen because it is light-weight, and some IoT devices do not have enough computing power to utilize anything much larger. V. L. Shivraj et al. reviewed the suitability of one-time passwords for IoT devices then developed a scheme using identity based ECC and Lamport's OTP algorithm. The scheme they developed is less resource-hungry than competing schemes used for similar tasks. Also, the new scheme can be up-scaled to Smart City, Smart Home and Smart Infrastructure application [16].

Another team of researchers have also looked into development of an algorithm for IoT devices based on ECDH. T. K. Goyal and V. Sahula took advantage of ECC's smaller-key-size-for-equal-security and quick computations to reduce power consumption, which is important for IoT devices that could have a limited amount of power to start with. Another benefit of the algorithm is the low memory and bandwidth reserve required. The team has also done proper analysis of power and performance as well as comparisons to DH and RSA. They have found that ECDH is superior to the other tested algorithms in terms of power and area [17].

6 MODERN APPLICATIONS

Aside from traditional and mobile applications, many different researchers are extending the reach of elliptic curves to applications outside of the traditional context. There are many different papers from researchers who have implemented ECC in less-conventional settings.

6.1 Smart Grid

The smart grid aids electrical power transmission by including status information along with the power being transmitted. Equipped with this sophisticated means of communication, the system can make informed choices, such as power routing. This greatly improves the efficiency of the grid. Data transmission must be secure and reliable through a smart grid. To ensure the security of communication, many key distribution methods have been proposed, however many cannot provide anonymity or have unsatisfactory performance. D. He et al. have proposed an anonymous key distribution (AKD) scheme for the smart grid using identity-based ECC [18].

With their AKD scheme, the grid can provide smart meter anonymity and mutual authentication without needing a trusted party. The new scheme has greater performance than other AKD schemes proposed in the past. In addition, this new smart grid AKD scheme's computational costs are much cheaper than previous schemes.

6.2 Vehicular Communication

Autonomous, self-driving cars (autos) have shown great improvement over recent years. Currently, given that sensors are appropriately mounted and all equipment is functioning properly, autos have the ability to accurately calculate stopping distances and safe driving speeds much more efficiently and faster than a human would ever be able to. These vehicles operate much faster than humans can think and can react quicker to data that is constantly being fed to it. Besides sensors, another important data source could be from a network with other autos. When a vehicle is traveling at 75 miles per hour down a freeway, it would be a tragedy if a malicious hacker were able to interfere with communication.

Currently, this communication is used for emergency situations and location privacy. With insecure wireless communication, false messages can be transmitted which can lead vehicles astray. A. Dua et al. have proposed a scheme for secure smart city vehicle message communication. Previously, large key size has made secure communication difficult to implement. By using ECC the team was able to develop a method that is able to use smaller key sizes which could grant equal or greater security than previously proposed cryptographic mechanisms for vehicle communication. The solution is mathematically simple and comes at a low computation cost. The scheme also provides mutual authentication and confidentiality with forward security. Analysis has shown that the proposed scheme is suitable for the smart city environment. With their scheme, man-in-the-middle attacks and brute force attacks are not possible in polynomial time due to the elliptic curve discrete logarithm problem, making the system a secure method of communication which can save lives [19].

6.3 RFID

Radio Frequency Identification (RFID) has been around for quite some time now. It is a technology used for automatic identification via radio waves and is used in a large variety of applications. Recently, RFID has been used for authentication. Due to the wireless nature of RFID there are a variety of security concerns that must be addressed.

M. Benssalah et al. used an FPGA to validate RFID messages for authentication using ECC. This time, the ECC implementation is used for actual encryption by means of ECC-ElGamal. Specifically, the team showed the effectiveness of the

implementation with car key systems. The paper outlines the entire process from encryption to decryption as it is useful for any type of access control beyond just car key systems. With ECC, the transmission of access credentials does not need to be transferred in plaintext which thwarts many attack attempts [20].

6.4 Iris Pattern Recognition

For iris pattern recognition, S. V. Vishnubhatla developed a hashing algorithm based off of ECC. The test input were images from the UBIRIS database that were run through the system. The hashing was done on grayscale images using Python with the OpenCV library. After results were collected, analysis showed that the elliptic curve hashing algorithm outperformed the standard MD5 and SHA-1 hashing algorithms and is accurate at a rate of 99.5% [21].

The hash is a sponge hash that is cleared by NIST standards to be used commercially [22]. Sponge hash is a multi-step hash with a single input that is an array of numbers generated by the input from the iris, the colorful center of the eye that contains the pupil at the center. Though other algorithms are computationally expensive, the mathematics of the elliptic curve hash is less complicated. The final results show that the entropy of the hashing algorithm is statistically greater than the other hashing algorithms tested, which means the elliptic curve hash has a lower chance of collisions.

6.5 E-Health Applications

G. Sahebi et al. have designed a framework utilizing ECC for its fast speed, smaller keys, and greater security for E-health applications such as sensors and wearables. Secure and Efficient Elliptic Curve Cryptosystem (SEECC) is their proposition developed to select secure, efficient curves from all available curves. By choosing these safer curves, security is enhanced. The method also increases the efficiency of ECC by means of a parallel genetic algorithm. The team's multiple case studies have shown that the proposed parallel genetic algorithm had increased accuracy at quicker speeds than previous methods for selecting safe curves in a field where data security and privacy is a must [23].

7 Future of ECC

In present day and in the near future, ECC has been shown to be a safe, secure, reliable, and fast form of cryptography, whether it be for digital signatures, key exchange, hashing, or encryption. It is fast enough for production use on servers, for desktop web browsers, and even for mobile and embedded systems. ECC is even an acceptable form of cryptography for modern applications that extend further than computers and mobile phones, as was shown for the smart grid, the IoT, and others. But how long will ECC be able to stand in the future?

7.1 Post-Quantum Cryptography

The quantum computer's analog to a classical bit is a qubit. Qubits can be in a state of 0 or 1, just like classical bits. But due to the nature of quantum mechanics, qubits in a state of superposition can take the form of, both, 0 and 1 simultaneously until the desired output is acquired. This property translates to a performance boost that is not possible by classical computers. The major problem holding quantum computing back is stability. Although we have been finding better ways to increase the stability of qubits in lab settings, quantum computers are not quite ready for mass production.

When properly implemented, quantum computers pose a great threat to public key cryptography. Researching post-quantum cryptography will quickly show allusions and applications for Shor's Algorithm or some variation of it [24]. Initially, Shor's Algorithm was developed to quickly factor integers into their prime factors using qubits. In 2001, researchers at IBM have factored 15 into 3 and 5 with Shor's Algorithm using only 7 qubits [25]. In the years since, researchers have factored even larger numbers with early quantum computers using Shor's and other related algorithms. The largest being 56153 in 2012 [26].

Shor's Algorithm can be used to solve mathematical problems that many common forms of cryptography rely on. The trap door functions, mentioned previously in section 2.3, no longer take excessive amounts of time to solve. Even in the absence of private keys or information that is required to solve these problems with classical computation, a fully realized quantum computer can solve the same problem in a reasonably short amount of time.

Even the additional security of ECC is vulnerable to the power of quantum computing. When quantum computers become more stable and accessible the elliptic curve discrete logarithm problem will not be much of a problem at all. Researchers have shown that approximately 1000 qubits are all that is necessary to solve a 160-bit elliptic curve cryptography key, when focusing on prime curves [27].

7.2 Continued Development of ECC Algorithms

So why bother improving classical cryptography, including ECC? As mentioned, production quantum computers are still years off from being in every office and on every desktop around the world. Until then, it is still important to assure information and improve cryptography until that day comes. And when that day comes, there are already algorithms being developed that are designed to be immune to the unusual power of quantum computers. Some of these algorithms include: lattice-based cryptography [28], multivariate cryptography [29], and even a new application of elliptic curves called supersingular elliptic curve isogeny cryptography [30].

8 CONCLUSIONS

Elliptic curves have many advantages compared to other public key cryptography algorithms. For cryptography, ECC requires smaller key sizes for equivalent security with RSA, it is more immune to attacks, and it can be applied to many applications. In this paper, we have surveyed the mathematics behind Elliptic

Curve Cryptography, including structure and operations. We briefly surveyed the history of Elliptic Curves as used for cryptography and choosing safe curves for production use. Next, we compared ECC to other forms of cryptography, in terms of equal security and speed, for applications involving key exchange and digital signatures. We then looked into the current applications of ECC and how they are used for traditional computing usage. Next, we surveyed multiple mobile applications for ECC, followed by newer applications that have been recently proposed. Finally, we explored the future of Elliptic Curves and how they may still be used in a post-quantum cryptography world.

REFERENCES

[1] James S. Milne. 2006. *Elliptic Curves*. BookSurge Publishing, Charleston, SC.

[2] Alfred J. Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. 2002. Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field. *IEEE Transactions on Information Theory* 39, 5 (Aug. 2002), 1639-1646. DOI: https://doi.org/10.1109/18.259647

[3] Neal Koblitz. 1987. Elliptic Curve Cryptosystems. *Mathematics of Computation* 48, 177 (Jan. 1987), 203-209. DOI: https://doi.org/10.1090/S0025-5718-1987-0866109-5

[4] Victor S. Miller. 1985. Use of Elliptic Curves in Cryptography. In *CRYPTO '85 Advances in Cryptology*. Springer-Verlag, 417-426.

[5] M. Sudhakar, R.V. Kamala, and M.B. Srinivas. 2007. A Unified, Reconfigurable Architecture for Montgomery Multiplication in Finite Fields GF(p) and GF(2^n). In *20th International Conference on VLSI Design held jointly with 6th International Conference on Embedded Systems (VLSID'07)*. IEEE, Bangalore, India, 750-755. DOI: https://doi.org/10.1109/VLSID.2007.27

[6] Jean Karim Zinzindohoué, Evmorfia-Iro Bartzia, and Karthikeyan Bhargavan. 2016. A Verified Extensible Library of Elliptic Curves. In *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*. IEEE, Lisbon, 296-309.

[7] Daniel J. Bernstein. 2006. Curve25519: New Diffie-Hellman Speed Records. In *International Workshop on Public Key Cryptography*. Springer, Berlin, Heidelberg, 207-228. DOI: https://doi.org/10.1007/11745853_14

[8] Elaine B. Barker, William C. Barker, William E. Burr, William T. Polk, and Miles E. Smid. 2007. *Recommendation for Key Management – Part 1: General (Revision 3)*. National Institute of Standards & Technology, Gaithersburg.

[9] Non Thiranant, Young Sil Lee, and Hoonjae Lee. 2015. Performance Comparison Between RSA and Elliptic Curve Cryptography-Based QR Code Authentication. In *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, Gwangiu, 278-282. DOI: https://doi.org/10.1109/WAINA.2015.62

[10] OpenSSL. 2016. OpenSSL Cryptography and SSL/TLS Toolkit. (December 2016). Retrieved from https://www.openssl.org/

[11] Alexa. 2016. Alexa Top 500 Global Sites. (December 2016). Retrieved from http://www.alexa.com/topsites

[12] Roland van Rijswijk-Deij, Kaspar Hageman, Anna Sperotto, and Aiko Pras. 2016. The Performance Impact of Elliptic Curve Cryptography on DNSSEC Validation. *IEEE/ACM Transactions on Networking* PP, 99 (Sep. 2016), 1-13. DOI: https://doi.org/10.1109/TNET.2016.2605767

[13] Wuqiong Pan, Fangyu Zheng, Yuan Zhao, Wen-Tao Zhu, and Jiwu Jing. 2016. An Efficient Elliptic Curve Cryptography Signature Server With GPU Acceleration. *IEEE Transactions on Information Forensics and Security* 12, 1 (Aug. 2016), 111-122. DOI: https://doi.org/10.1109/TIFS.2016.2603974

[14] Alavalapati Goutham Reddy, Ashok Kumar Das, Eun-Jun Yoon, and Kee-Young Yoo. 2016. A Secure Anonymous Authentication Protocol for Mobile Services on Elliptic Curve Cryptography. *IEEE Access* 4 (July 2016), 4394-4407. DOI: https://doi.org/10.1109/ACCESS.2016.2596292

[15] Pranjali Deepak Nikam and Vanita Raut. 2015. Improoved MANET Security Using Elliptic Curve Cryptography and EAACK. In *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE, Jabalpur, 1125-1129. DOI: https://doi.org/10.1109/CICN.2015.221

[16] V L Shivraj, M A Rajan, Meena Singh, and P Balamuralidhar. 2015. One time password authentication scheme based on elliptic curves for Internet of Things (IoT). In *2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW)*. IEEE, Riyadh, 1-6. DOI: https://doi.org/10.1109/NSITNSW.2015.7176384

[17] Tarun Kumar Goyal and Vineet Sahula. 2016. Lightweight Security Algorithm for Low Power IoT Devices. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, Jaipur, 1725-1729. DOI: https://doi.org/10.1109/ICACCI.2016.7732296

[18] Debiao He, Huaqun Wang, Muhammad Khurram Khan, and Lina Wang. 2016. Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography. *IET Communications* 10, 14 (Sep. 2016), 1795-1802. DOI: https://doi.org/10.1049/iet-com.2016.0091

[19] Amit Dua, Neeraj Kumar, Mukesh Singh, M. S. Obaidat, and Kuei-Fang Hsiao. 2016. Secure Message Communication Among Vehicles Using Elliptic Curve Cryptography in Smart Cities. In *2016 International Conference on Computer, Information and Telecommunication Systems (CITS)*. IEEE, Kunming, 1-6. DOI: https://doi.org/10.1109/CITS.2016.7546385

[20] Mustapha Benssalah, Mustapha Djeddou, and Karim Drouiche. 2016. Design and Implementation of a New Active RFID Authentication Protocol Based on Elliptic Curve Encryption. In *2016 SAI Computing Conference (SAI)*. IEEE, London, 1076-1081. DOI: https://doi.org/10.1109/SAI.2016.7556111

[21] Sasank Venkata Vishnubhatla. 2015. An Elliptic Curve Algorithm for Iris Pattern Recognition. In *2015 Annual Global Online Conference on Information and Computer Technology (GOCICT)*. IEEE, Louisville, KY, 51-59. DOI: https://doi.org/10.1109/GOCICT.2015.19

[22] Bart Preneel. 1993. *Analysis and Design of Cryptographic Hash Functions*.

[23] Golnaz Sahebi, Amin Majd, Masoumeh Ebrahimi, Juha Plosila, Jaber Karimpour, and Hannu Tenhunen. 2016. SEECC: A Secure and Efficient Elliptic Curve Cryptosystem for E-health Applications. In *2016 International Conference on High Performance Computing & Simulation (HPCS)*. IEEE, Innsbruck, 492-500. DOI: https://doi.org/10.1109/HPCSim.2016.7568375

[24] Peter W. Shor. 1997. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing* 26, 5 (Oct. 1997), 1484-1509. DOI: https://doi.org/10.1137/S0097539795293172

[25] Lieven M. K. Vandersypen, Matthias Steffen, Gregory Breyta, Costantino S. Yannoni, Mark H. Sherwood, and Isaac L. Chuang. 2001. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature* 414, 6866 (Dec. 2001), 883-887. DOI: https://doi.org/10.1038/414883a

[26] Nikesh S. Dattani and Nathaniel Bryans. 2014. Quantum factorization of 56153 with only 4 qubits. (November 2014). Retrieved from https://arxiv.org/abs/1411.6758v3

[27] John Proos and Christof Zalka. 2003. Shor's discrete logarithm quantum algorithm for elliptic curves. *Quantum Information & Computation* 3, 4 (July 2003), 317-344.

[28] Chaohui Du and Guoqiang Ba. 2016. High-Performance Software Implementation of Discrete Gaussian Sampling for Lattice-Based Cryptography. In *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*. IEEE, Chongqing, 220-224. DOI: https://doi.org/10.1109/ITNEC.2016.7560353

[29] Li Tian and Wansu Bao. 2010. A Medium Field Multivariate Public Key Signature Scheme with External Perturbation. In *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*. IEEE, Jinggangshan, 753-757. DOI: https://doi.org/10.1109/IITSI.2010.149

[30] Brian Koziel, Reza Azarderakhsh, Mehran Mozaffari Kermani, and David Jao. 2016. Post-Quantum Cryptography on FPGA Based on Isogenies on Elliptic Curves. *IEEE Transactions on Circuits and Systems I: Regular Papers* 64, 1 (Oct. 2016), 86-99. DOI: https://doi.org/10.1109/TCSI.2016.2611561