ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM

by

THOMAS HILTON JOHNSON III

A thesis submitted to the Graduate Faculty of
Elizabeth City State University
in partial fulfillment of the
requirements for the Degree of
Master of Science in Mathematics

Elizabeth City, North Carolina

May 2021

APPROVED BY

Julian A.D. Allagan, Ph.D.	Glen Bowman, Ph.D.
Malcolm D'costa, Ph.D.	Kenneth L. Jones, Ph.D.

Dipendra C. Sengupta, Ph.D.
Chair, Thesis Committee

ABSTRACT OF THESIS

ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM

Elliptic curve cryptography has been a remarkable development in the history of cryptography thanks to the properties provided by the implementation of elliptic curve cryptography. Elliptic curve cryptography has proven to be adaptable given its broad presence across various electronics of differing sizes and capabilities. Elliptic curve cryptography is known for utilizing the discrete logarithm problem, modern algebra, and elliptic curves that have encouraged continued research into elliptic curve cryptography in its advantages and restrictions. The current comprehension of elliptic curves as well as continuing advancements and employments of elliptic curve cryptography have yielded the elliptic curve digital signature algorithm (ECDSA) which has become an alternative to the primary Digital Signature Algorithm. The ECDSA provides advantages of elliptic curve cryptography to the function of the digital signature algorithm to authenticate and protect transmissions between involved parties. Such can be explored in the securing of bitcoin related transactions where the completely electronic, decentralized currency would face a multitude of cyber threats and require safeguards to remain trustworthy.

DEDICATION

I would like to dedicate this thesis to my family that has been with me from the beginning and has helped me along the way. Without them I do not know where I would be in life. It was through their wisdom and love that I have been nurtured into the person I am today. I know there is a wealth of wisdom, experience, and perspective that were imparted to me to assist me on this journey through life and will no doubt continue to assist me in the future.

Thanks to the family at Gray Legal Group, PLLC for many fun summers and supporting me throughout most of my life. Thanks for the celebrations, life advice and nurturing.

Special thanks to Mr. Wall, who maintained an excellent Exploring Technology course in high school that kept me engaged in the topics. I thank you for encouraging the ambition I had then I hope that you are well and safe in paradise. Know that you left a legacy that continues to grow today.

Thanks to my cousin Lee Godly Jr., Aunt Louise, and Uncle George for paving a legacy here at Elizabeth City State University that I would also take part in.

Special thanks to the network that I have been so fortunate to be a

part of that helped me along the way. I would like to give thanks to Dr. Jones who encouraged my curiosity in mathematics and introduced me to the idea of double majoring in mathematics along with computer science in my undergraduate career.

Thanks to Steve Coleman for providing me with a high school internship that encouraged STEM exploration and introduced me to peers with similar aspirations within my community.

Thanks to Dr. Lamara Warren, Dr. Apu Kapadia, Pat Shaffer, EJ Seong, and Tyler Dell for an excellent first internship experience. Being able to travel to Indiana was a privelege that I enjoyed and was only made better by being part of a team that enabled me to grow both personally and professionally.

Thanks to the Science Gateways Community Institute, Texas Advanced Computing Center, and their respective staff for providing an excellent second internship experience. Thanks to Rosalia Gomez and Dr. Ritu Arora for the resources and opportunities available during the interval of the internship. Thanks to Carlos, Gerald and Anubhaw for being great and supportive colleagues for this internship experience.

Thanks to NCAR, CISL and SIParCS staff for providing the resources and infrastructure to enable successful internships in Boulder, Colorado. Special thanks to Dr. Lauer, Ms. Do, Mr. Foust, and Dr. Vanderwende, Mrs. Mickelson, Mr. Dobbins, and Dr. Dennis for wonderful and informative set of internship experiences.

Thanks to Mr. Close and his enthusiasm in sharing knowledge. Also, I

thank him for being a firm supporter of mathematics and the other STEM fields.

ACKNOWLEDGEMENT

Special thanks to Dr. Dipendra Sengupta for encouraging my interest in the field of cryptography through the Applied Cryptography and Data Security course where my first research project in cryptography was conducted. I also thank Dr. Dipendra Sengupta for being the advisor for this thesis and offering guidance and expertise in mathematics and MATHEMATICA.

Thanks to Dr. Allagan for being an excellent coordinator for the Graduate Program in Applied Mathematics.

Thanks to Dr. Jones for being an astounding Chair for the Department of Mathematics, Computer Science and Engineering Technology.

A huge thank you to the thesis defense committee for their time and evaluations.

I would like to thank the Department of Mathematics, Computer Science and Engineering Technology for the opportunity to pursue my undergraduate and masters level education at Elizabeth City State University.

Contents

List of Figures

1 Introduction

Cryptography is the study of methods of protecting information as said information crosses public channels. In public key cryptography the original message will start out as plaintext, then proceed to undergo an algorithm that converts the plaintext to a ciphertext, the ciphertext itself is to be made unreadable or difficult to read across the public channel. The ciphertext is to be decoded by the receiver to be reverted back to the plaintext. Keys are used to ensure that between the receiver and the sender, that the receiver is able to revert the ciphertext to plaintext through the cryptographic algorithm used to produce the ciphertext as well as the provided key. For public key cryptography, every person involved possesses their own personal public and private keys. The public key is the key that a person will share with others for enabling the encrypting of messages being received by said person. The private is used by said person to revert encryption on the transmitted ciphertext and reveal the plaintext.

2 Related Literature

There exists a plethora of current literature regarding ECDSA and its related topics. This is due to the widespread implementation of ECDSA as well as the general tenability of ECDSA and elliptic curve cryptography.

Wang, Yu, Zhang, Piao, and Liu embarked on the research endeavor that produced "ECDSA weak randomness" in Bitcoin to apply scienctific

insight to a concern with bitcoin's application of ECDSA [14]. As of the composition of this article, the arbitrary value foible was still present and potentially exploitable [14]. This foible exposed the values that composed private keys leaving the undesirable exposure for bitcoin transactions and management [14]. With governments now authorizing bitcoin to be a viable form of currency, this foible in bitcoin ECDSA has posed another consideration for bitcoin usage [14]. This foible is not confined merely to ECDSA, as similar phenomenon have been identified to endanger digital signature algorithm and ECDSA's RSA equivalent [14]. A notable mention of exploitation is the usage of spam transaction attacks and data that may suggest such exploits were more prevalent than initially assumed [14].

The foible with the arbitrary values can emerge in two scenarios which propose different scales of threat as well as magnitudes of exposure [14]. One scenario is that the arbitrary value of one exchange is equivalent to that of a previous exchange although the public keys for the exchanges in question are relatively unique [14]. Within this scenario, it is observed participants of the exchanges could successfully attempt to deduce the other participants withheld private keys, endangering the viability of their bitcoin financials [14]. The second scenario is formed once there are at least a pair of exchanges that sustain equivalent public keys which would imply that the same private key is sustained as well allowing for determining the private key for said set of exchanges [14]. The initial preventative measure applied was RFC 6979, yet the weak randomness foible stands to persist to the present for ECDSA

[14].

"Securing Bitcoin wallets via a new DSA/ECDSA threshold signature scheme" is a research article Noting the prevalence of criminal activities that revolve around illegal procurement of bitcoin from individuals and entities that legally acquired bitcoin [15]. The research also provides suggested countermeasures to reduce the criminal procurement of bitcoin at various scales which will be beneficial to the larger bitcoin economy [15]. The reasoning for why these criminal procurements of bitcoin are such an issue has been condensed three principles embedded in the system that allows bitcoin to exist as established by the authors [15]. The devised countermeasure is a threshold wallet that applies a the feature of two-factor authentication to attempt to reduce the likelihood of criminal procurement of bitcoin wallets or bitcoin themselves [15]. The threshold wallet, by the experiments conducted, does not severely hinder the bitcoin exchange system which is contributing suggestions to integrate threshold wallets into at least a portion of the bitcoin cyberinfrastructure [15].

Tessler and Byrnes' exploration delves into the ramifications of bitcoin being used in conjunction with quantum computing assets as well as quantum computing assets being applied in an offensive against bitcoin [16]. The general overview of emerging quantum computing assets is that there will be minimized effects, until there quantum computing assets will be scaled upwards to enable significant feats to be accomplished [16]. Bitcoin mining by their estimates will be relatively unchanged since SHA-256 has no satisfac-

tory means of being reverse engineered to obtain the value of the argument submitted to SHA-256 [16]. The main method for mining with quantum computing assets for the bitcoin cyberinfrastructure is through employing the quantum computing algorithm known as Grover search [16]. As the current computing assets stand, contemporary computing assets when tuned for mining bitcoin are still displayed to be more effective than the quantum computing assets that were available at the time of this research from Tessler and Byrnes [16]. In terms of cryptographic defense, foibles are noted towards Shor's algorithm when referencing elliptic curve cryptography alone [16]. For bitcoin's cyberinfrastructure, various mechanisms are employed that act as encumbrances to deter or reduce any offensive possibilities that were suggested at the writing of this article in the context of bitcoin's tenability [16]. This is compounded with the time-dependent viability of an offensive within bitcoin mechanisms [16]. Not to mention, the cryptocurrency culture has already begun preparations for possible quantum computing offensives [16]. There is an examination done on the opportunities that could be revealed if a cryptocurrency is devised that is primarily reliant and tuned for satisfactory quantum computing assets at a later date [16].

Johnson, Menezes, and Vanstone are responsible for the in-depth research article entitled "The Elliptic Curve Digital Signature Algorithm (ECDSA)" which was published under Certicom [13]. It is apparent that the research of Johnson, Menezes, and Vanstone is to both go over the breadth of the subject of ECDSA, while providing the notes and precautions for the best practices

[13]. One such practice includes a set of options to employ for acquiring viable elliptic curves for a stellar foundation for the remainder of ECDSA architecture to rest upon [13]. Another practice that is given no small amount of focus is the ensuring that participants involved in any ECDSA exchanges have undertaken the diligent task to ensure all participants involved are properly following acceptable archetypes [13]. A notable gem is the outline of DSA architectures for better comprehension of the mathematical phenomenon that grant higher tenability at relatively enormous scaling [13]. The trio of mathematical conundrums for DSA architectures are noted to include elliptic curve, discrete logarithm, as well as integer factorization [13].

An extensive amount of concentration is allocated to delineating the buttressing concepts and mathematical fields for the continuation of the core focus of ECDSA [13]. One more mention for elliptic curves within this research article is the inclusion of Hasse's Inequality to obtain an interval in which the quantity of elements of the elliptic curve can be determined [13]. In discussing fields, the research article spends ample time introducing the symbolism involved in effectively converting the fields to viable and practical software machinations [13]. The symbolic options available are in turn emphasized to allow for further comprehension of the overall development of the viable software equivalent from the ECDSA theory being explained [13]. Where appropriate, there are samples provided for enhancing comprehension which at times includes demonstrations [13]. When addressing the possible foibles of components of ECDSA, Johnson, Menezes, and Vanstone provide

a thorough list of possibilities that may be available as potential exploits to contemplate [13]. These exploits include, but are not limited to, usage of programming algorithmic techniques, the employment of stellar hardware configurations, attempts to tackle the SHA-1 or its equivalent RIPEMD-160, or exploitations in a diverse assortment of other components essential to the operation of ECDSA [13].

Malvik and Witzoee wrote "Elliptic Curve Digital Signature Algorithm and its Applications in Bitcoin", providing a general summation of ECDSA in its most rudimentary points before going into bitcoin [17]. Notably, the designated elliptic curve being employed with the bitcoin cyberinfrastructure is mentioned to be Secp256k1, categorized to be a Koblitz elliptic curve [17]. Two additional statements are made in regard to this particular elliptic curve [17]. The first is that at the time of the writing of Malvik and Witzoee's endeavor, the Secp256k1 elliptic curve has minimal exploits [17]. That statement is made in conjunction with another statement that Secp256k1 was not endorsed to be a stellar option by NIST for a sustainable ECDSA architecture [17]. The efficiency of any ECDSA architecture rests on the restrictions that revolve around the measure of celerity that the elliptic curve operations may be carried out [17]. The factors that contribute to this are best summarized as the definition of the elliptic curve in turn provides a set of factors that can streamline or lengthen the time expense for elliptic curve operations [17].

A means to raise efficacy in relation to this bottleneck is the employment

of an endomorphism to get a more effective equivalent for the Secp256k1 elliptic curve, without having to abandon the original Secp256k1 elliptic curve [17]. In the case of Secp256k1, the endomorphism identified by Malvik and Witzoee is the Frobenius map [17]. The Frobenius map accelerates the elliptic curve operations to be over 33% quicker than the standard operations lacking the employment of endomorphisms [17]. Malvik and Witzoee endorse the efficacy of the Koblitz category of elliptic curves thanks to the boost in acceleration for operations thanks to the employment of endomorphisms [17]. The Koblitz elliptic curve category have been deemed adequate to be suggested for usage by administrative bodies [17]. Further endorsement for the Koblitz elliptic curve category comes from the Koblitz elliptic curve category being found within the NIST Digital Signature Standard for reference in federal implementations of digital signature architectures [17]. Koblitz elliptic curves are given higher endorsement as the means to build an instance of a Koblitz elliptic curve has eased anxiety from the potentiality that the seed determined for elliptic curve development can lead to inherent foibles in other elliptic curve categories [17].

Ziegeldorf, Matzutt, Henze, Grossmann, and Wehrle sought to reconfirm cryptocurrencies as an instrument that promotes protection of the finaincial data of users in "Secure and anonymous decentralized Bitcoin mixing" [18]. Some of the article is spent refreshing on the topics of blockchain, cryptocurrencies, and interrelated subjects that would assist in comprehending the research article [18]. To prevent users from being simply identified by

existing foibles in the structure of the blockchain on which cryptocurrencies have a stable foundation, the authors present the software CoinParty as a suitable remedy [18]. CoinParty within this article is a revamped iteration of a previous iteration that is also mentioned for a baseline contrast for the progression of the CoinParty iteration of focus in this research article [18]. Restoring confidence in cryptocurrency has become more critical since Bitcoin alone became valued at over four billion USD within the year of 2015 [18]. The extent of the potential exploits can be stretched to the obtainment of IP addresses of the participants of the cryptocurrency infrastructure [18].

CoinParty stands as a mixing service that jumbles up a collection of cryptocurrency transfers similar to other cryptocurrency mixer software, yet aims to preserve the concealment of the participants included in said transfers while maintaining a credible system within the cryptocurrency domain [18]. The six principles providing a collection of metrics to evaluate CoinParty are designated to be "Applicability and Usability, "Scalability", "Deniability", "Correctness", "Cost-efficiency", and "Anonymity" [18]. The previously stated principles are used to justify why other mixing software fall short of what can be attained by CoinParty or that the suggested mixing prototype did not progress to an implementation stage [18]. CoinParty is structured to have a threshold mechanism embedded within to better manage the potentialiality of a large quantity of participants that desire to exploit the blockchain or launch an offensive against the blockchain [18]. The ECDSA architecture in turn is integrated with the threshold mechanism as well [18].

Significant time is allocated to present a definitive contrast between CoinParty with any other potential alternatives [18]. Ziegeldorf, Matzutt, Henze, Grossmann, and Wehrle affirm CoinParty is a hybridized product in reference to the amalgamated and unamalgamated mixing software where CoinParty's appeal is extracted from both design constructs [18]. Another affirmation and clarification that is made is CoinParty's harmonious nature when integrated with the available cryptocurrency options that have Bitcoin's ECDSA architecture embedded within the cryptocurrency's cyberinfrastructure [18].

Extance's work, "The future of cryptocurrencies: Bitcoin and beyond", displays the general timeline of cryptocurrency advancement with a fixation on Bitcoin [19]. There are points made concerning the potential assimilation of cryptocurrencies into the collection of assets and instruments of standing financial constructs [19]. One such financial construct being JP Morgan Chase, which already had a verified curiosity in the potential integration of Bitcoin cyberinfrastructure [19]. Bitcoin has specifically acquired a tarnished reputation from being utilized for or being associated with crimes or illegal ventures [19]. Furthermore, cryptocurrency remains a novel financial instrument as well as electronic-based advancement that makes the potential legislation that could concentrate on cryptocurrencies a discussion starting matter [19]. Extance takes time to also detail briefly some controversial matters that cryptocurrencies have brought about since popularization has skyrocketed [19].

The circumvention of tinier scale cryptocurrencies have demonstrated

that essentially all cryptocurrencies, unless countermeasures are embedded to mitigate such, possess a foible where the faction possessing over one-half of the cryptocurrency mining assets will be able to actively ignore the mechanisms meant to reduce unfair or criminal actions within the cryptocurrency cyberinfrastructure [19]. These instances are designated to be "51% attacks", and are dreaded in particular for anyone involved in cryptocurrencies [19]. Countermeasures to these offensive exploits have already been materialised at differing phases of progression [19]. Another controversial matter is the energy expense incurred to mine cryptocurrency, which has endorsed the production of cryptocurrency that acts to buttress something of long-term value such as genetic research to make cryptocurrency more than merely an economic instrument [19].

Sarath, Jinwala, and Patel's research endeavor provides a notable attempt to look into the ECDSA architecture to examine the idiosyncracies that have emerged from the differing permutations that have been tuned from the rudimentary ECDSA architecture [20]. Since the year 2000, ECDSA has been deemed to meet the criterion to be approved by ISO, NIST, IEEE as well as ANSI [20]. Two points of interest are made immediately, the first of which is the intricity of the abstraction of the ECDSA architecture [20]. Second is the ECDSA maintains a hurdle in practice for gaining a tenable elliptic curve at initiation of the ECDSA's construction [20]. The essential components of ECDSA architecture are given condensed sections in conjunction with a proof for validation before the permutations of ECDSA architectures

are examined [20].

The first permutation that is investigated is noted to sustain the private key for some fixed interval [20]. Due to the sustainment of the private key for a fixed interval, exploits on recurrence of the private key have led to the first permutation having a severe foible [20]. The first permutation is stated to be more appropriate for usage on less capable electronics [20]. The second permutation was identified to be best suited in the instance that the participants attempting to authenticate an endorsement lack computational assets that would be required for other permutations from modifications of the authentication sequence [20]. The second permutation and first permutation are concluded to share the same foible [20]. The third permutation augments tenability of the ECDSA architecture by a duo of concealed variables that eliminate the potential acquiring the private key by exploiting the recurrence of said duo of concealed variables [20]. The third permutation is noted to exceed the the safeguards offered by the rudimentary ECDSA architecture [20].

The Elliptic Curve German Digital Signature Algorithm (ECDGSA) is the fourth permutation to be investigated which, similar to the third permutation, is tenable when the concealed variable(s) is/are recurrent [20]. In line with the first permutation, ECGDSA will sustain a private key for a fixed interval [20]. The ECDGSA is evaluated to be more efficient than the deployment of rudimentary ECDSA architecture in overall expense [20]. The fifth permutation is spread across two subsections as there is the source form

of the fifth permutation of ECDSA, then an altered state is delineated after adding more safeguards to compensate for existing foibles within the the fifth permutation [20]. Two more permutations attempting to tackle specific scenarios or mandates are also delineated, followed by table for viewing all the permutations within the article [20]. The final notes of the article pinpoint the suggested scenarios for employing the permutations from the authors' judgements [20].

Satoshi Nakomoto is the name or alias of the individual or collective that have earned recognition for their brainchild of Bitcoin, and the rudimentary architecture that buttresses cryptocurrency today [4]. Development of Bitcoin was spurred forward as a countermeasure to the foible of validation and safeguarding of the exchange of currency within the structure of the internet [4]. Nakomoto states that a pivotal contribution to the safeguards of the cryptocurrency cyberinfrastructure is that there is predominance of honorable participants active to dishonorable active participants [4]. The concept of the cryptocurrency "coin" is implemented in the form of a conglomeration of the endorsements of the relevant participants, effectively providing a means by which to validate the "coin['s]" passage [4]. To reduce the probability of a coin being subject to unauthorized replication for invalid exchanges, the cryptocurrency system participants are notified of exchanges then the exchanges are ensured to be authentic by participants within the confines of the cryptocurrency system [4].

By inserting chronological markers within the conglomeration of hashed

data that composes the cryptocurrency coin, an authentic detailing of when exchanges were initiated is retained [4]. The means of doing so is accomplished by Nakamoto in the adaptation of a preceding "proof-of-work" mechanism [4]. Said mechanism was devised in reference to Adam Back's Hashcash [4]. The proof-of-work mechanism can then be seen to make the trial of circumventing the existing ledger for manipulation toilsome [4]. The trial itself would engage methods that require an exponential time complexity to even hope to carry out with contemporary computing assets, which becomes more daunting with the ledger expanding from further exchanges [4]. As an added countermeasure, the novel additions to the ledger can have their expense requirements heightened in response to the participants of the cryptocurrency infrastructure utilizing refined or numerous computing assets [4]. Proof-of-work will manage the progression of the ledger with simple conditions to determine how existing disputes within the blockchain will be resolved [4]. Redundancies have been embedded to enable the blockchain to continue functioning even with potential delays or disputes arising throughout [4].

Another mechanism for preventing offenses against the Bitcoin infrastructure is that the infrastructure itself advocates for honorable participation over any other possible involvement [4]. To better illustrate such advocation, the similarities of attempting to manipulate the Bitcoin blockchain are provided in reference to the Gambler's Ruin conundrum as an example is displayed [4]. Concerns for the management of storage assets are allayed in a detailed, brief elaboration on the means employed to minimize the strain on storage

devices [4]. Nakamoto has also considered the diversity that is inherent in the exchanges that will take place with Bitcoin as Bitcoin advances [4]. In regard to safeguarding personal data from being accessible through the ledger, Nakamoto has given consideration to this in stating constructs that filter data as well as possible foibles that could be revealed [4].

"Secure Implementation of ECDSA Signatures in Bitcoin" is an article delineating the ECDSA, its rudimentary concepts, Bitcoin, and the integration of ECDSA within Bitcoin from DI WANG [23]. ECDSA architecture has the secp256k1 elliptic curve embedded in the cyberinfrastructure, yet the suggestion from NIST has been the secp256r1 elliptic curve in contrast [23]. Three critical components of the Bitcoin infrastructure is the SHA256, RIPEMD160 and Merkle trees [23]. SHA256 is known to be a hash function. Merkle trees are a data structure that can be observed to buttress authentication systems by storing hashes [23]. Merkle trees diminish the expenditures that are incurred from disputes originating from alterations of the continuing Merkle tree [23].

P. Kocher is presented an offense for the expanse of cryptography through the concentration on the hardware infrastructure versus solely the software infrastructure [23]. This category of offense is titled the side channel offense [23]. The crutch of a successful side channel offense is that there is data concerning the efficacy or the other hardware properties are analyzed, removing the necessity of aiming directly for reverting encryption methods safeguarding plaintext [23]. For example, the interval that is expended for

computations could be repurposed to assist with the obtainment of the private key [23]. Another example is using a sustained intake of the energy expenditure for supplemental data that can be employed in parallel to other hacking methods to boost an existing offensive to undermine cryptographic constructs [23]. Energy expenditure has been definitively noted to produce unintentional insight as to the characteristic values that are embedded in the executed encryption architecture [23]. Another means of initiating a side channel offense is to purposely generate disruptions to divulge the critical values of the cryptographic architecture thanks to the data that will be attained [23]. There are safeguards that can be employed to restrict the probability of success of side channel offenses, yet there should be an acknowledgement that such will dampen the efficacy of the cryptographic construct at play [23].

Over the multiple explorations that have been conducted for elliptic curve and Bitcoin, there has been a wealth of knowledge extracted from such explorations. Some of the explorations for ECDSA and its integration into Bitcoin have yielded the option to embed other cryptographic constructs to retain the safeguards in Bitcoin or eliminate foibles that ECDSA could produce [23]. One option is to embed a Lamport endorsements mechanism in lieu of the contemporary ECDSA mechanism [23]. The employment of Lamport endorsements can be traced to V. Buterlin in the early 2010s [23]. The second option that has been devised, while being deemed of the same likeness of Lamport endorsements, of another classification of endorsements

[23]. The classification the second option fits into has been determined to be Guy Fawkes [23]. The machinations that led to the second option was traced to A. Miller and J. Binneau, devised inside the same decade as Lamport endorsements option of Buterlin [23]. WANG notes that these two options demonstrate that ECDSA is not the sole means of cryptocurrency continuing to possess sustainable endorsement mechanisms [23].

2.1 Groups

Let G be a non-empty set with the binary operation $*$, the following conditions must hold for G to be considered to be a group:

1: $*$ is associative meaning that for $a, b, c \in G$ $(a * b) * c = a * (b * c)$, $\forall a, b, c \in G$.

2: There exists an identity element e such that for $a, e \in G$, $a*e = e*a = a, \forall a, e \in G$.

3: For each element $a \in G, \exists a^{-1}$ such that $a*a^{-1} = a^{-1}*a = e, \forall a, a^{-1}, e \in G$. a^{-1} is called the inverse of a.

A group is called abelian or commutative if for $a, b \in G, a*b = b*a, \forall a, b \in G$. A cyclic group G is a group that can be generated by a single element a, so that every element in G has the form a^n for $n \in \mathbb{Z}$.

The order of an element a of a group is the smallest positive integer n such that $a^n = e$, where e denotes the identity element of the group. The order of the group G is denoted by $|G|$ is the number of elements in G. If the order of G is prime, then G is a cyclic group.

2.2 Cryptocurrency

Cryptocurrency is a form of electronic currency that was originally structured to be decentralized so that individual people rather than consolidated financial institutions could maintain the funds traded amongst each other. Cryptocurrencies are dependent on ledgers distributed amongst all participating members of the cryptocurrency exchange to conduct, verify, and validate transactions that occur amongst the members of the cryptocurrency exchange. The rudimentary infrastructure for cryptocurrency is open source meaning that anyone could potentially develop their own cryptocurrency after proper setup of said infrastructure.

One thing to keep in mind with cryptocurrency is that cryptocurrency does not have a tangible equivalent, cryptocurrency is existent only in the digital domain which simplifies exchanges with cryptocurrency [1]. Despite said constraint, cryptocurrency has been pursued for a variety of motivations including as a financial option in attempt to profit if or when the cryptocurrency prospers in the markets [1]. Being decentralized, cryptocurrency lacks the guarantees and safeguards that are associated with conventional currencies [1]. Cryptocurrencies are known to be volatile in nature further complicating the processes of becoming involved in cryptocurrency trading [1].

Cryptocurrency activity is meant to be anonymous but is not by any means the method to completely remove people from being connected to exchanges [1].

2.2.1 Blockchain

Blockchain is a term that is attributed to a tool to be applied ubiquitously to any and all issues or complications within the domains dealing with currency as well as domains concerning technological implementations [2]. The denotation of blockchain concerns cyberinfrastructure consisting of "peer-to-peer networking, consensus mechanisms, and ... hash-linked data structures" [2]. Each computer or node is going to be transmitting to participating nodes of the same network infrastructure instead of a single node or collection of nodes manipulating transmissions amongst all the nodes as the main authority [2], sustaining no node(s) as possible foibles due to exerting authority over nodes. There are also programs that allow the involved nodes to remain consistent in the contents of the existing ledgers and ensure the participating nodes can validate the exchanges occurring and would have occurred [2]. This sustains the consistency across all of the involved nodes by a defined set of conditions to prevent inconsistencies [2]. The amendable and appendable ledger is "will make alterations evident," if from an unauthorized party yet it is up to the individuals or parties with authority in the blockchain network to rectify such [2].

Blockchain can be used for various forms of electronic information [2]. For the consensus, there are are a trio of attributes that are examined to characterize the blockchain infrastructure that has been established [2]. These three attributes concern where the protections are concentrated for the blockchain, if the electronic information in question is accessible to everyone, and the

exclusivity involved in accessing the blockchain [2].

2.2.2 Bitcoin

Bitcoin viewed as the first and possibly the most renown cryptocurrency that exists in the world today. Bitcoin is the brainchild of an individual or group referred to as Satoshi Nakamoto, and the original bitcoin blockchain detailed in the their white paper *Bitcoin: A Peer-to-Peer Electronic Cash System* [3]. Bitcoin was devised for providing a means to electronically transfer currency amongst two entities without the necessity of a additional entity being incorporated at any step of the transferal [4]. In place of the third entity being the authority of the credibility of the transfer, a ledger of a blockchain would serve to take into account all the transmissions of currency amongst the community of the blockchain [4]. New devices can become part of the blockchain or separate from said blockchain, with the freshest ledger being the referenced for any neophyte or readmitted devices [4].

The drive to conceive the blockchain infrastructure is derived from the observation of monetary entities being intensely incorporated into the electronic transmission [4]. This depth of involvement by monetary entities goes all the way to settling the conflicts that can generated as a consequence of the potential disagreements amongst those involved in the transmission [4]. Since the monetary entities have tacked on supplemental tolls on top of the value of currency that will be transmitted, electronic transmissions become more expensive and there can be limited reliability in regard to the

two groups or individuals that are directly included in the transmission [4]. It should also be mentioned that there is no certainty or absolute assurance of the credibility or safeguarding of each and every transmission being made [4]. Cryptocurrency, as proposed by Nakamoto, was to eliminate the dependency on an unnecessary third entity so that two individuals or groups could commit to the transmission of currency with the assurance being placed on a logical security model in a blockchain [4].

Through the implementation of electronic endorsement, the cryptocurrency is established as a construct formed from continuing endorsements of the individuals or groups participating in a series of engagements in transmitting currency [4]. Hashing methods as well as public key cryptography are essential for this proposed cryptocurrency construct [4]. The amendable ledger that is ubiquitous amongst active participants of the blockchain is essential to maintaining the credibility of all present and previous transmissions of currency [4]. Attempting to alter the ledger will readily increment expense and toil that will be necessitated to overwrite the existing credibility of the ledger as transmissions in the blockchain persist [4]. The blockchain is fault tolerant and resistant to incorrrect computations occurring within a minority of the participants [4].

The blockchain is devised in such a way that the motivation is to maintain the credibility of the blockchain rather than overwrite the credibility of the ledger [4]. The cryptocurrency can have established parameters that can constrain the parameters maintaining the cryptocurrency such as the quan-

tity of units that are permitted within the blockchain [4, 23]. This resolves severe controversies of inflation that can occur in contemporary currency [23]. Individuals or groups can have a layer of safeguard for their identities as the ledger does not directly implicate whom was participating the transmission of currency, and the public and private keys are freshly made for each transmission [4].

The mining mechanism expending an estimated $\frac{1}{6}$ of an hour within the Bitcoin cyberinfrastructure is critical to the dual set of aims to be accomplished for the sustainability of Bitcoin [23]. The mining will grant consistent opportunity for authentic exchanges to be confirmed within the infrastructure [23]. The other aim that is fulfilled is that spawning of supplemental coins within the Bitcoin infrastructure [23].

A possible controversy with Bitcoin commerce is centered on the exchanges that take place as Bitcoin is adapted. An offensive that may be setup within Bitcoin to bypass the safeguards for removing unauthentic exchanges by predominance of confirmation of the Bitcoin participants [23]. This is due to the foible existing in the embedded scripting languages that exist within the Bitcoin cyberinfrastructure that provide opportunity to tamper with endorsements [23]. This offense has been titled the "Transaction malleability" [23].

While Bitcoin is the premiere cryptocurrency, there have been a large quantity of similar initiatives generated all of which are traceable to Satoshi Nakamoto's machination. Cryptocurrencies have been observed, investi-

gated, studied, and created to such an extent that there exist platforms that allow new enthusiasts to construct their own cryptocurrency with ease. Cryptocurrencies have even been launched by corporations or in response to icons or elements of popular culture.

2.3 Cryptography

Since the early emergence of human civilization information has proven to be invaluable in the correct context to interested groups or individuals. For those who are the intentional sender and receiver of said information, the information should be transmitted without error. For anyone external to such a transmission, it is critical the information is safeguarded. To this end, cryptography was born and has since been advanced through the ages as resources and concepts developed in the advancement of human civilization. Cryptography at its basics involve a sender, a receiver, the plaintext, the ciphertext, encryption, and decryption. The plaintext is the authentic message to be transmitted from sender to receiver before undergoing encryption. Encryption is a process by which information in the plaintext is morphed into ciphertext. The encryption is done through the use of a repeatable process, or algorithm, that will in turn allow for decryption of the ciphertext after said ciphertext is transmitted to the receiver. Decryption is the process of taking the ciphertext and revert it back to the plaintext. While cryptography is aimed to safeguard the information, there are times when said information is coveted by unintended persons or groups. That in turn spurs said persons

or groups to attempt to gain the plaintext even if that requires decrypting the ciphertext.

Cryptographic algortithms are designed to convert plaintext to ciphertext by a defined set of rules. In its simplest terms, the sender will encrypt the message then the receiver will decrypt said message operating in context of the same cryptographic algorithm. Over the advance of history, there has been a number of cryptographic algorithms that have been devised within the growing field of cryptography. Said cryptographic algorithms are identified in two distinct families: asymmetric and symmetric.

2.3.1 Symmetric Cryptography

Symmetric cryptography is constructed in such a way that there is a singular key that can be utilized for encryption of the plaintext and decryption of the ciphertext. An example would be the Caesar cipher, where the key is the is the quantity of characters shifted to yield the ciphertext. To decrypt the Caesar cipher with the key, you would merely shift in the opposite direction by the same number of characters. Since there is one key involved that serves for both encryption and decryption purposes, the apparent foible of symmetric cryptography is that the key must be guarded otherwise anyone with access to the key can decrypt the ciphertext. There is also the consideration of devising a method to transmit the key between the receiver and sender without any unintended entities acquiring access to the key. The involvement of a sole key for the encryption and decryption actions lead to symmetric

cryptographic algortihms being valued less than asymmetric cryptography for safeguarding the plaintext contents to be transmitted [5].

Examples of symmetric cryptogrphic algorithms are the hill cipher, playfair cipher.

2.3.2 Asymmetric Cryptography

Asymmetric cryptography implements a dual collection of keys to provide a means of encryption and decryption without the vulnerabilities exhibited by symmetric cryptography. One key is denoted to be the public key and is distributed by the user amongst the network of participants that encrypted transmissions will take place. Any messages to be sent to the user will be encrypted by other participants by the user's public key. The public key acts to enable encryption, but the user and every other participant will possess a private key. The private key is related to the public key in that any information encrypted with that specific public key can be decrypted by the private key that was created with that public key. The private key is safeguarded by each individual owner to so that only the person the private key belongs to may be able to successfully decrypt ciphertext to plaintext from the usage of the associated public key. Aymmetric cryptographic algorithms are the preferred option when attempting to guard plaintext details through encryption from unauthorized persons or entities that would exploit the plaintext [5]. Examples of asymmetric cryptographic algorithms are RSA and elliptic curve cryptography.

2.3.3 Why Cryptography Continues to Develop

Cryptography is a field that does not cease in its growth due to the constant struggle between interests to safeguard information and interests to exploit said information. Those working in cryptography to guard plaintext must consistently dedicate their work to devising fresh methods or upgrading preceding methods in hopes of outpacing the opposition that seeks to obtain the plaintext by undoing the encryption that yields the ciphertext [5]. To make the internet a viable means of transmitting details from one point to another with confidence, the progress of cryptography must strive ahead to to provide adequate defensive methods to safeguard plaintext as transmitted ciphertext [5].

2.4 Digital Signature Algorithm

2.4.1 Hash

Before addressing the digital signature algorithm, the concept of hashing must be addressed. A hash is an output of a predetermined magnitude that is produced from a hashing function [5]. The idea of hashing in cryptography is that the hash will be used as both an identifier and tool for verification for transmitted details or as a means for ensuring passphrases are tenable within a database [5].

2.4.2 Digital Signature

The digital signature is was designed as a tool for ensuring validity in the context of technology and data, being held as an equal to the physical signature a person may bestow [6]. Digital signatures should remain functional both after communications are completed and beyond the context of time of said communications as long as the content involved can be fetched [6], further emphasizing the equivalence to a physical signature.

2.4.3 Digital Signature Algorithm (DSA)

DSA is composed of a pair of functions at the most rudimentary concept in which one function confirms the genuineness of the digital signature that is the product of the partner function [6]. Where asymmetric cryptography becomes involved is in regard to said pair of functions applying the public key for confirming the genuineness of the digital signature, while the private key is crucial to producing the digital signature within the implementation [6]. The plaintext is morphed to into the output of a hashing function that is delivered to the person wanting to confirm the digital signature in conjunction with delivery of the plaintext and digital signature in question [6].

Before attempting to use the digital signature algorithm, proper context for the implementation of the digital signature algorithm is established by determining the factors that will be crucial to the creation of the dual set of keys [6]. This is critical in the implementation of both the digital signature algorithm, and the later discussed elliptic curve digital signature algorithm, since

without this cohesive context amongst all participants there will be inconsistencies reflected in the factors governing each participants' implementation of the digital signature algorithm [6]. Following such, dual keys are created amongst the participants followed by confirmation of the exclusive possession of the dual collection of keys amongst said participants [6]. Any plaintext to be delivered to any other participant will have its hash output produced for later confirmation that the plaintext's contents was preserved throughout the time period the communication took place [6]. Additional methods may be used to further manage communications over the long term, such as the creation of an arbitrary value to identify specific instances of plaintext [6]. Once the preparations for the implementation of the digital signature algorithm are satisfied, a digital signature may be produced then reviewed to examine if there were miscalculations when producing said digital signature algorithm or for other purposes related to the confirmation of genuineness [6].

2.4.4 Factors Necessary for the Digital Signature Algorithm

Five key components are identified as being unavoidable for the implementation of the digital signature algorithm [6]. Said components are the function utilized for hashing, the dual set of keys, the value to be given to each instance of a communication that remains confidential, the factors that establish the context of the digital signature algorithm, as well as the plaintext to be subjected to the digital signature algorithm [6]. The private key will be denoted

as a, the public key is to be denoted as b, and the non-clonable arbitrary value to be given to each instance of a communication denoted as c [6].

A non-composite value denoted as r shall be acquired in which the number of bits composing r are denoted by S [6]. The limitations of r are established as $2^{S-1} < r < 2^S$ [6]. Then t will denote a non-composite factor of r in which H is the quantity of bits that compose t [6]. The limitations of t are established as $2^{H-1} < t < 2^H$ [6]. Taking the finite field or group where the designated operation is multiplication with order r, f is the element that is also a generator of a finite subfield or subgroup possessing order of t within said finite field or group [6]. The limitations of f are described as $1 < f < r$ [6]. The limitation of the private key in respect to the previously detailed parts of the digital signature algorithm implementation is that $0 < a < t$ as well as the lifespan of a being constrained to however long digital signatures are being produced [6]. The public key can be delineated as $b = f^a \ mod \ r$ and has a lifespan of how ever long the digital signatures of the private key will be confirmed [6]. The non-clonable value that is associated with each instance of communication has the limitation of being $0 < c < t$ [6]. The value c will have a multiplicative inverse c^{-1} under $mod \ t$ that will also contribute to producing digital signatures [6].

2.4.5 Using the Digital Signature Algorithm

To start, obtain the smaller of the two values of H and the quantity of bits that compose the product attained from usage of the hashing function

denoted as *hashbitquantity* as the value, denoted as Q, will be critical later [6]. The components of the digital signature compose a tuple (i, j), in which i is attained by the method of $i = (f^c \ mod \ r) \ mod \ t$ [6]. j is obtained by two methods, the first is acquiring u by extracting the quantity of bits Q from the left hand side of the hash produced from the communication contents and the hash function [6]. j is then acquired by $j = (c^{-1}(u + ai)) \ mod \ t$ [6]. Before communication contents are transferred to the receiver a confirmation is made of $i, j \neq 0$, so that in the case such occurs a fresh c is attained to restart digital signature production [6].

Digital signatures can be confirmed through a related process dependent on parameters that are transferred over by the sender [6]. The receiver will be transferred duplicates of (i, j), b to be denoted as (i', j'), b' from the participant who created the digital signature, which is possible when the receiver is operating within the same context of factors as the sender of the digital signature so progress may move forward [6]. To initiate, $0 < i', j' < t$ will be proven to be fact, and if neither i', j' violate such, the procedure may continue to advance [6].

A series of new values will be required to continue [6]. Let $g = (j')^{-1} \ mod \ t$, k denotes smaller of the values of the H and *hashbitquantity* retrieved from employing hashing [6]. Let $m_1 = gk \ mod \ t$ while $m_2 = (i' \times g) \ mod \ t$ [6], then $n = ((g^{m_1} \times b^{m_2}) \ mod \ r) \ mod \ t$ [6]. The inverse of j' will be acquired succeeded by ensuring that k is in the form of numeric value of the group $\mathbb{Z}$ [6]. Confirmation of the digital signature concludes if the $n = i'$, where

any other possibility is subject to being dismissed to optionally reattempt communication [6].

2.5 Elliptic Curves

Elliptic curves are mathematical constructs that can be delineated by what is referred to as the Weierstrass equation. The general Weierstrass equation is known to be condensable through the usage of set factors from the core structure of $E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ [7]. To attain the condensed version, we initiate the system of steps with completing the square solely on the left side of equation: $(y + \frac{a_1x}{2} + \frac{a_3}{2})^2 = x^3 + (a_2 + \frac{a_1^2}{4})x^2 + a_4x + (\frac{a_3^2}{4} + a_6)$. Such is followed by allowing $y_1 = y + \frac{a_1x}{2} + \frac{a_3}{2}$, allowing $A = a_2 + \frac{a_1^2}{4}$, setting $B = a_4$, and establishing $C = \frac{a_3^2}{4} + a_6$. Such will allow for the reconstruction of $(y + \frac{a_1x}{2} + \frac{a_3}{2})^2 = x^3 + (a_2 + \frac{a_1^2}{4})x^2 + a_4x + (\frac{a_3^2}{4} + a_6)$ to the version structured as $y_1^2 = x^3 + Ax^2 + Bx + C$. Establishing $x = x_1 - \frac{a_2}{3}$ will allow for yet another alteration to the structure of the Weierstrass equation to yield: $y_1^2 = (x_1 - \frac{A}{3})^3 + A(x_1 - \frac{A}{3})^2 + B(x_1 - \frac{A}{3}) + C = x_1^3 - x_1^2A + x_1\frac{A^2}{3} - \frac{A^3}{9} + Ax_1^2 - 2x_1^2 - 2x_1\frac{A^2}{3} + \frac{A^3}{3}Bx_1 - B\frac{A}{3} + C$ so that we may press forward. The proceeding step is employing simplification to gain: $y^2 = x_1^3 - x_1\frac{A^2}{3} + Bx_1 - B\frac{A}{3} + C = x_1^3 + x_1(B - \frac{A}{3}) + (-B\frac{A}{3} + C)$ from the previous version. We then examine that by setting $c = B - \frac{A^2}{3}$ as well as $d = -B\frac{A}{3} + C$ we can get the condensed mathematical structure of $y_1^2 = x_1^3 + cx_1 + d$ from the general Weierstrass equation $E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$.

2.5.1 Elliptic Curves Constructed on the Group of Real Numbers

Elliptic curves can be formed on various groups, allowing different set of factors that will govern the structure of said elliptic curves. On R, the elliptic curve is observed to be formed by an assortment of coordinates that visually demonstrate its characteristics. Elliptic curves in the context of $\mathbb{R}$ reveals the two divisions that exist for elliptic curves within the condensed structure of $y = x^3 + cx + d$. The attribute on which the divisions are founded is whether the elliptic curve possesses a singularity or not. The singularity is understood to be a‘ coordinate that is a component of the elliptic curve where $\frac{d}{dx}$ as well as $\frac{d}{dy}$ are both reckoned as 0, meaning the elliptic curve in question will be singular [8]. Otherwise, said elliptic curve is has displayed a non-singular nature [8]. It is known that the singularity is tied to the phenomenon of isomorphism that makes the singular elliptic curve a foible, not safeguard, in the context of cryptography [8]. When the equation of $4c^3 + 27d^2 = 0$ is examined to be fact, the elliptic curve in question is determined to be singular [9]. When the equation of $4c^3 + 27d^2 \neq 0$ is examined to be fact, the elliptic curve in question is determined to be non-singular [9].

2.5.2 Examples of Singular as well as Non-Singular Elliptic Curves on $\mathbb{R}$

Below will be examples of singular and non-singular elliptic curves over $\mathbb{R}$. First are the non-singular elliptic curves:

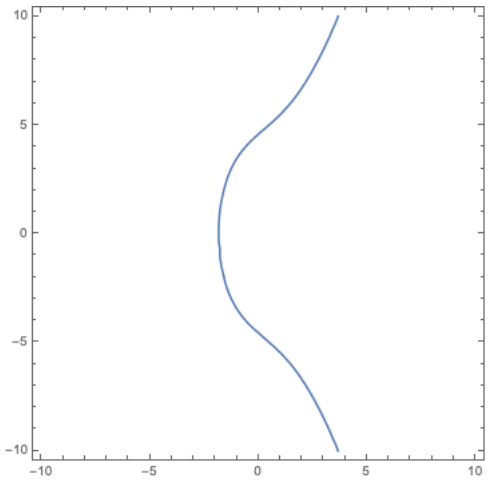

Figure 1: A non-singular elliptic curve modeled by $y^2 = x^3 + 8x + 21$, $-10 \leq x \leq 10$, $-10 \leq y \leq 10$

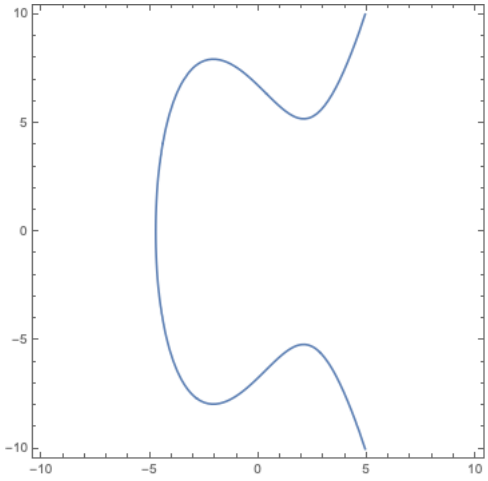

Figure 2: A non-singular elliptic curve modeled by $y^2 = x^3 - 13x + 45$, $-10 \leq x \leq 10$, $-10 \leq y \leq 10$

Next are examples of singular elliptic curves:

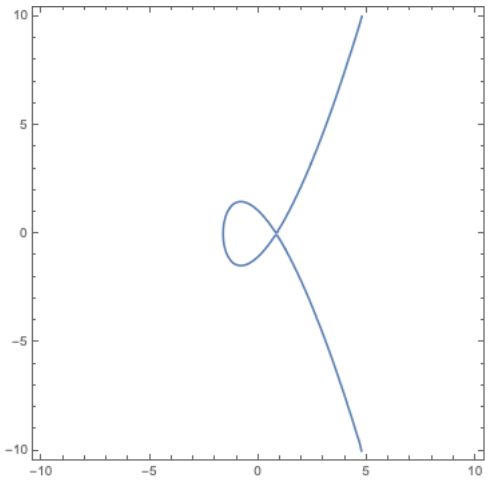

Figure 3: A singular elliptic curve modeled by $y^2 = x^3 - 2x + \sqrt{\frac{32}{27}}$, $-10 \leq x \leq 10$, $-10 \leq y \leq 10$

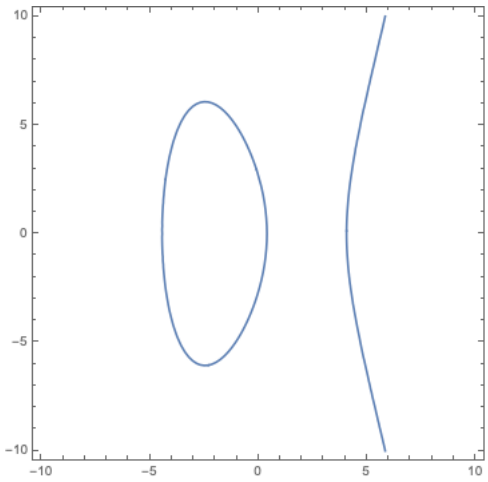

Figure 4: A singular elliptic curve modeled by $y^2 = x^3 - \sqrt{\frac{1323}{4}}x + 7$, $-10 \leq x \leq 10$, $-10 \leq y \leq 10$

These figures were constructed through the usage of the software MATHEMATICA.

2.5.3 Set Up for Operations on an Elliptic Curve

Elliptic curve operations possess an identity that is designated to be the coordinate at infinity, ∞, and is essential to the employment of elliptic curve operations. The inverse of a coordinate on $\mathbb{R}$ is considered to be a reflection while taking into account the horizontal axis of the coordinate plane. For example, say that there is a coordinate B that can be denoted as $B = (x_B, y_B)$ would possess an non-negative counterpart $-B$ denoted as $-B = (x_B, -y_B)$ when the elliptic curve in question constrained in the group of real numbers [9]. Whenever the scenario of $B + -B$ arises, ∞ is the predictably yielded as a consequence [9].

2.5.4 Operations of an Elliptic Curve

When dealing with elliptic curve operations, there are a trio of scenarios that are formed based on the inputs for the operations. Said elliptic curve operations can be done strictly on the coordinate plane if the elliptic curve lies on $\mathbb{R}$, otherwise the elliptic curve operations depend on specific formulas in the context of the group to calculate.

The first scenario is when a coordinate A is doubled or $A + A$. When this happens, on the coordinate plane a contiguous linear connection will be formed in the context of $\mathbb{R}$ [8]. The output, coordinate C, is the mirror in reference to the horizontal axis of wherever said line makes contact with a novel coordinate,$-C$, that is a portion of the elliptic curve in question [8]. There are two sub-cases that will exist here. For when the y-value of the

coordinate A is equivalent to 0, $-C$ will be ∞. For when the y-value of the coordinate A is not equivalent to 0, C will be the consequence of the transformation of $-C$ [8]. The algebraic assets to attain $-C$ are the rise over run of the contiguous line for the coordinate A or $m = \frac{(3(x_A^2)+c)}{2(y_A)}$, $x_C = m^2 - 2(x_A)$, and $y_{-C} = -y_A + m(x_A - x_C)$, then attain C from $-C$ [8].

The second scenario is when $A + -A$ occurs [8]. Thanks to the properties of the group and the respective addition operation, ∞ will always be yielded [8].

The third scenario is when there are two novel coordinates A and B being utilized and $B \neq -A$. To acquire $-C$ by geometric applications a linear connection is formed for A and B [8]. Upon extending this connection, the next instance the connection encounters the elliptic curve will be denoted as the coordinate $-C$ [8]. A reflection with respect to the horizontal axis would be sufficient to yield the coordinate C from the attained coordinate $-C$ [8]. Another situation can emerge when either the coordinates A or B are indeed the tangent coordinate of the elliptic curve in consideration of the linear connection made. Then, the tangent coordinate is also defined to be $-C$. Mirror $-C$ to yield C. In the application of algebra, first m, the rise over run of the contiguous linear connection, is established as $m = \frac{(y_A - y_B)}{(x_A - x_B)}$, followed by $x_C = m^2 - x_A - x_B$ as well as $y_{-C} = -y_A + m(x_A - x_B)$ to acquire the coordinate $-C$ to obtain coordinate C [8].

2.5.5 Examples of Elliptic Curve Operations Over $\mathbb{R}$ as the Field or Group

The first example will make use of the non-singular elliptic curve E: $y^2 = x^3 + 8x + 21$ and the coordinates $A = (1, \sqrt{30})$, $B = (6, \sqrt{285})$.

Example 1. *Initiate by obtaining* $m = \frac{(\sqrt{30}-\sqrt{285})}{(1-6)} \approx$ *2.281.*
Succeeded by the obtainment of the x-value of coordinate $-C$*:* $x_{-C} = 2.281^2 - 1 - 6 \approx$ *-1.797.*
Concluded by the obtainment of the y-value of coordinate $-C$*:* $y_{-C} = \sqrt{30} + 2.281(1 - 6) \approx$ *-5.928.*
Now coordinate C *can be observed to be* $(-1.797, -5.928)$.

The second example will make use of the non-singular elliptic curve E: $y^2 = x^3 - 13x + 45$ and the coordinate $A = (3, \sqrt{33})$ will be doubled.

Example 2. *Initiate by acquiring* $m = \frac{(3(3^2)-13)}{2(\sqrt{33})} \approx$ *1.219.*
Then acquire the x-value of coordinate $-C$*,* $x_{-C} = 1.219^2 - 2(3) =$ *-4.514.*
To acquire the y-value of coordinate of $-C$*,* $y_{-C} = -\sqrt{33} + 1.219(3 + 4.514) =$ *3.415.*
To finish, coordinate C *can be determined to be* $(-4.514, 3.415)$.

2.5.6 Elliptic Curves in the Context of Finite Fields $\mathbb{Z}^+$

When elliptic curve operations are implemented over $\mathbb{Z}$ rather than $\mathbb{R}$, similar algebra may be used with some adaptation reliant on the finite field or group. Non-composite values that exceed the value of 3 will be denoted as p, so that

the elliptic curve operations can be in the context of $\mathbb{Z}_p$ [9]. The modification of the condensed Weierstrass mathematical structure $y^2 = x^3 + cx + d$ that is necessary for implementation within the context of the of the finite field of $\mathbb{Z}_p$ are $y^2 = x^3 + cx + d \ mod \ p$ in combination with the necessity of $4c^3 + 27d^2 \not\equiv 0 \ mod \ p$ for non-singular elliptic curves [9]. The consideration must also be made that both c and $d \in \mathbb{Z}_p$ [9]. The element ∞ is retained from $\mathbb{R}$ to $\mathbb{Z}_p$ [9]. The coordinates that are elements of the set are defined as (x, y) in which the values of x and y are $\in \mathbb{Z}_p$ [9].

Dealing with elliptic curves in the context of finite fields alters the nature of the elliptic curve slightly. Such may best be seen in examples of visualizing the coordinates of the elliptic curve over finite field $\mathbb{Z}_p$.

The first example for an elliptic curve over $\mathbb{Z}_p$ is $y^2 = x^3 + 4x + 6 \ mod \ 23$ (we can verify that $4(4)^3 + 27(6)^2 \ mod \ 23$ is 9). Here we can see that set of coordinates satisfying $y^2 = x^3 - 4x + 6$ over $\mathbb{Z}_{23}$ are: (0, 11), (0, 120, (5, 6), (5, 17), (6, 4), (6, 19), (7, 3), (7, 20), (9, 9), (9, 14), (11, 1), (11, 22), (13, 1), (13, 22), (14, 0), (16, 70), (16, 16), (19, 8), (19, 15), (20, 6), (20, 17), (21, 6), (21, 17), (22, 1), (22, 22), ∞. The order of this group is 30.

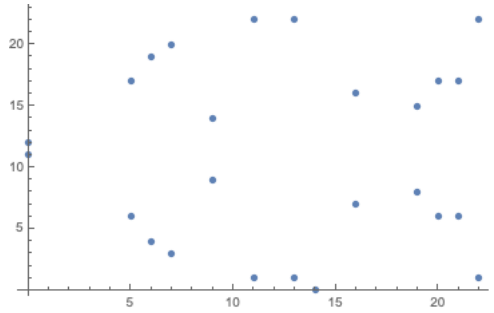

Figure 5: The elements of the elliptic curve $y^2 = x^3 + 4x + 6$ over $\mathbb{Z}_{23}$

The second example for an elliptic curve over $\mathbb{Z}_p$ is $y^2 = x^3 + 9x + 4$ *mod* 17 (we can verify that $4(9)^3 + 27(4)^2$ *mod* 17 is 16). It can be observed that the set of coordinates satisfying $y^2 = x^3 + 9x + 4$ over $\mathbb{Z}_{17}$ are: (0, 2), (0, 15), (2, 8), (2, 9), (4, 6), (4, 11), (5, 2), (5, 15), (6, 6), (6, 11), (7, 6), (7, 11), (9, 7), (9, 10), (12, 2), (12, 15), (14, 1), (14, 16), ∞. The order of the group is 19.

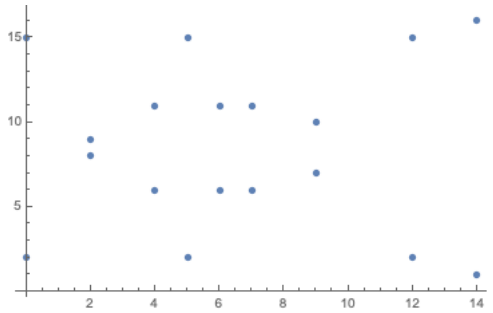

Figure 6: The elements of the elliptic curve $y^2 = x^3 + 9x + 4$ over $\mathbb{Z}_{17}$

The third example for an elliptic curve over $\mathbb{Z}_p$ is $y^2 = x^3 + 22x + 56$ *mod* 101 (we can verify that $4(22)^3 + 27(56)^2$ *mod* 101 is 4). It can be observed that the set of coordinates satisfying $y^2 = x^3 + 22x + 56$ over $\mathbb{Z}_{101}$ are: (0, 37), (0, 64), (1, 33), (1, 68), (4, 39), (4, 62), (6, 0), (8, 21), (8, 80), (10, 8), (10, 93), (11, 35), (11, 66), (13, 32), (13, 6), (14, 49), (14, 52), (15, 23), (15, 78), (18, 27), (18, 74), (22, 49), (22, 52), (25, 24), (25, 77), (26, 23), (26, 78), (28, 0), (32, 20), (32, 81), (33, 37), (33, 64), (37, 35), (37, 66), (39, 21), (39, 80), (41, 47), (41, 54), (42, 5), (42, 96), (46, 38), (46, 63), (51, 2), (51, 99), (52, 2), (52, 99), (53, 35), (53, 66), (54, 21), (54, 80), (55, 48), (55, 53), (58, 10), (58, 91), (59, 17), (59, 84), (60, 23), (60, 78), (63, 10), (63, 91), (65, 49),

(65, 52), (66, 12), (66, 89), (67, 0), (68, 37), (68, 64), (70, 40), (70, 61), (71, 26), (71, 75), (72, 28), (72, 73), (74, 22), (74, 79), (75, 47), (75, 54), (81, 10), (81, 91), (84, 18), (84, 83), (85, 31), (85, 70), (86, 47), (86, 54), (89, 36), (89, 65), (94, 8), (94, 93), (96, 15), (96, 86), (97, 45), (97, 56), (98, 8), (98, 93), (99, 2), (99, 99), (100, 29), (100, 72), ∞. The order of this group is 100.

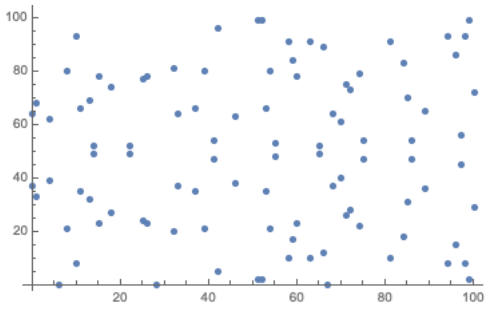

Figure 7: The elements of the elliptic curve $y^2 = x^3 + 22x + 56$ over $\mathbb{Z}_{101}$

These figures were constructed through the usage of the software MATHEMATICA.

2.5.7 Elliptic Curve Operations Over $\mathbb{Z}_p$

Going over some of the general rules that carry over from $\mathbb{R}$ counterparts we understand that coordinate A will consistently be the result of $A + \infty$ and $\infty + A$, $A \in y^2 = x^3 + cx + d \ mod \ p$ [9].

The first scenario that will be reexamined in the context of $\mathbb{Z}_p$ is the addition of novel coordinates of the elliptic curve. Given coordinates A, $B \in y^2 = x^3 + cx + d \ mod \ p$, defining the coordinates $A = (x_A, y_A)$ and $B = (x_B, y_B)$ with solution $C = (x_C, y_C)$, we first calculate the rise over run $m = \frac{y_B - y_A}{x_B - x_A}$

$mod\ p$ [9]. Now we can successfully acquire $x_C = m^2 - x_A - x_B\ mod\ p$ followed by $y_C = m(x_A - x_C) - y_A\ mod\ p$ to obtain coordinate C [9].

The second scenario to be reexamined in the context of $\mathbb{Z}_p$ is the doubling of a coordinate A, or $A + A$ [9]. Given the coordinate $A \in y^2 = x^3 + cx + d\ mod\ p$ to be defined as $A = (x_A, y_A)$ the rise over run of the the contiguous line, $m = \frac{3x_A^2 c}{2y_A}\ mod\ p$, is found to acquire $x_C = m^2 \times 2x_A\ mod\ p$ [9]. This is succeeded by obtaining $y_C = m(x_A - x_C) - y_A\ mod\ p$ to conclude with the coordinate C [9].

The third scenario to be reexamined concerns ∞ as the result of the setup of $A + -A$, for which A, $-A$, ∞, $\in y^2 = x^3 + cx + d\ mod\ p$ [9].

2.5.8 Generating an Estimation for the Quantity of Coordinates that Will Satisfy $y^2 = x^3 + cx + d$ Over $\mathbb{Z}_p$

In establishing the order of the group formed by the elliptic curve over $\mathbb{Z}_p$, there are a few means of achieving such with different levels of success. There are means to compute the quantity of coordinates that satisfy $y^2 = x^3 + cx + d$ over $\mathbb{Z}_p$ using computing assets. The drawback of enumeration by brute force is problematic as it does little to preserve the overall availability of computing assets as the enumeration scales upwards with the parameters. Smaller variations of $y^2 = x^3 + cx + d$ $\mathbb{Z}_p$ will be faster to enumerate than larger counterparts. A more efficient method of estimating the quantity of elements for $y^2 = x^3 + cx + d$ $\mathbb{Z}_p$ is through employing Hasse's Inequality. To utilize Hasse's Inequality, we start by initially delineating Hasse's Inequality

to be $|p + 1 - N_p| \leq 2\sqrt{p}$ in which p is understood as the non-composite number [10]. It then stands that N_p is the quantity of coordinates that fulfill the parameters established for $y^2 = x^3 + cx + d\ \mathbb{Z}_p$ [10]. An augmentation can be employed to obtain the unraveled Hasse's Inequality delineated as $p + 1 - 2\sqrt{p} \leq N_p \leq p + 1 + 2\sqrt{p}$ to dissolve the absolute value portion of Hasse's Inequallity [10].

Two examples for employing Hasse's Inequality will be demonstrated:

Example 3. *Given* $y^2 = x^3 + 33x + 19$ *mod* 103 *(can verify that* $4(33)^3 + 27(19)^2$ *mod 103 is 25), we will acquire an estimate for the quantity of elements in the set of coordinates that fulfill the parameters for the elliptic curve group.*

$103 + 1 - 2\sqrt{103} \leq N_{103} \leq 103 + 1 + 2\sqrt{103}$

$84 \leq N_{103} \leq 124$

The quantity of elements in the set takes a value between 84 and 124.

The actual number of elements as checked by MATHEMATICA is 101.

Example 4. *Given* $y^2 = x^3 + 12x + 71$ *mod* 251 *(can verify that* $4(12)^3 + 27(111)^2$ *mod 251 is 227), we can obtain an estimate for the quantity of elements in the set of coordinates that fulfill the parameters for the elliptic curve group.*

$251 + 1 - 2\sqrt{251} \leq N_{251} \leq 251 + 1 + 2\sqrt{251}$

$220 \leq N_{251} \leq 284$

The quantity of elements in the set takes a value between 220 and 284.

The actual number of elements as checked by MATHEMATICA is 234.

2.6 Delving Into Elliptic Curve Cryptography

Elliptic curve cryptography is a subfield of cryptography that originated from combining the principles of group theory and the mathematical structures of elliptic curves for the purpose of fulfilling cryptographic goals. Elliptic curve cryptography belongs to the family of asymmetric cryptography, making a considerably more fortified than symmetric cryptography counterparts. For a bit of the history, elliptic curve cryptography can be traced to contemplations of Victor Miller and Neal Koblitz within the year of 1985 [8]. The crutch of elliptic curve cryptography safeguards lies in the inherent inclusion of the discrete logarithm problem from the possibilities of repetitive elliptic curve addition operations being utilized [8]. While acquiring a coordinate D from a coordinate A being added to itself over some h iterations ($h \times A$) is reasonable, reverse engineering such will become incrementally toilsome in reveal the value of h when employed well [8]. This safeguard is sustained in various scenarios including under the conditions that unauthorized parties manage to distinguish $y^2 = x^3 + cx + d \ mod \ p$ along with the coordinate A [8].

The discrete logarithm problem is crucial to comprehending the appeal that elliptic curve cryptography has for safeguarding information. The discrete logarithm problem emerges in a context analogous or equivalent to the simplicity of exponentiating using computational resources versus attempting to initiate and efficiently carry through on logarithmic implementations [8]. The best possibility for an unauthorized party is that the value of h can

be distinguished for $h \times A = D$ without having progressing to the extreme of processing the entire set of coordinates that are in the set of $y^2 = x^3 + cx + d$ $mod\ p$ [8]. Note that this takes into assumption the unauthorized party has previously managed to distinguish coordinate A as well as $y^2 = x^3 + cx + d$ $mod\ p$. Utilizing this brute force method will diminish in efficiency in relation to the rising magnitude of p once p is gargantuan [8]. Another safeguard is to examine A to validate that A does not make h simple to obtain from the elliptic curve group [8]. In the matter of cybersecurity, incrementing the time expense that is necessary for being able to obtain plaintext form ciphertext can limit if not prevent unauthorized parties from being able to access the information in the plaintext during the lifespan said information is still of reasonable value.

There exists a multitude of methods to employ elliptic curve cryptography. The following explanation will be on the ElGamel variant.

Each participant in the transmission of the ciphertexts must have the same established elliptic curve $y^2 = x^3 + cx + d\ mod\ p$ is agreed upon all relevant participants, then a value $f \in \mathbb{Z}$ that is restricted by $1 < f < p$ is determined by each participant to act as their personal private key [11]. Each participant proceeds to acquire a coordinate α from the group $y^2 = x^3 + cx + d\ mod\ p$ to compute another coordinate β [11]. β is defined to be $\beta = \alpha \times f$ [11]. The public keys of the participants that are released are a composition of β, α, and p [11]. Given x $\in y^2 = x^3 + cx + d\ mod\ p$, while $z \in \mathbb{Z}$ under the restriction $1 < z < p$ is the sender's private key, the

encryption can be observed to be carried out as $e_z(x, z) = (z \times (\alpha), x + z \times \beta)$ $= (y_1, y_2)$ [12]. The decryption counterpart for the encryption is carried out as $d_z(y_1, y_2) = y_2$ - $f \times y_1$ [12].

2.6.1 Examples of Elliptic Curve Cryptography

Two examples of elliptic curve cryptography shall be demonstrated:

Example 5. *Let Alice and Bob be two participants in an encrypted transaction where Alice is attempting to send an encrypted transmission to Bob. The elliptic curve to be used* $y^2 = x^3 + 33x + 19$ *over* $\mathbb{Z}_{103}$.
Bob's private key is $f = 53$. *The coordinate that Bob chooses as part of his public key is* $\alpha = (79, 82)$. *Bob then calculates* $\beta = 53 \times (79, 82) = (92, 73)$. *Bob exposes Bob's public key of* $\beta = (92, 73)$, $\alpha = (79, 82)$, *as well as* $p = 103$.
Alice acquires Alice's own private key of $z = 29$. *The plaintext that Alice desires to transmit is* $x = (63, 102)$.

Now Alice will begin the encryption process to send Bob the encrypted ciphertext to protect the plaintext.
$e_z((63, 102), 29) = (29 \times (79, 82), (63, 102) + 29 \times (92, 73)) = ((48, 50), (63, 102) +$ $(93, 53) = ((48, 50), (45, 54)) = (y_1, y_2)$.
Bob receives the transmission from Alice containing the ciphertext. To retrieve the plaintext Bob must initiate and carry through with the decryption process.
$d_z((48, 50), (45, 54)) = (45, 54) - 53 \times (48, 50) = (45, 54) - (93, 53) = (45, 54) +$

$(93, 50) = (63.102)$.

Now Bob has the plaintext from the ciphertext that was transmitted by Alice.

Example 6. *Alice an Bob are once again trying to transmit messages to one another. This time Bob is attempting to transmit a message to Alice utilizing elliptic curve cryptography. The elliptic curve that they have agreed upon is* $y^2 = x^3 + 12x + 71$ *over* $\mathbb{Z}_{251}$.

Alice's private key is assigned as $f = 179$*, and the coordinate Alice chooses is* $\alpha = (198, 144)$. *Alice then takes* α *and* f *to be able to compute* $\beta = (198, 144) \times f = (234, 15)$. *Alice now prepares to reveals Alice's public key that is composed of* $\beta = (234, 15)$, $\alpha = (198, 144)$, *and* $p = 251$.

After Alice reveals Alice's public key, Bob can now initiate preparations to be send the intended message. Bob's own private key is assigned as $z = 132$. *Bob's personal selection of coordinate is established to be* $\gamma = (121, 219)$. *The plaintext that Bob desires to transmit is* $x = (157, 136)$.

Bob initiates the encryption process to be able to generate the ciphertext to be transmitted to Alice.

$e_z((157, 136), 132) = 132 \times (198, 144), (157, 136) + 132 \times (234, 15) = ((234, 15),$
$(157, 136) + (9, 54)) = ((234, 15), (43, 246)) = (y_1, y_2)$

Now that Alice has received the successful ciphertext transmission form Bob, Alice can focus on the decryption process to be able to acquire the plaintext.

$d_z((234, 15), (43, 246)) = (43, 246) - 179 \times (234, 15) = ((43, 246) - (9, 54)) =$
$(43, 246) + (9, 197)) = (157, 136)$

Now Alice has managed to acquire the plaintext from the transmitted cipher-

text.

These examples were completed through the assistance of MATHEMATICA.

Thanks to the versatility and reduced overhead involved with elliptic curve cryptography, elliptic curve cryptography has become an indispensable asset to the assortment of cryptographic algorithms available.

2.7 Elliptic Curve Digital Signature Algorithm or ECDSA

The ECDSA is traced to Scott Vanstone who was attempting to fulfill a request for the National Institute of Standards and Technology [13]. The ECDSA is considered an equivalent to the DSA with the implementation buttressed by the group defined by an elliptic curve over the field $\mathbb{Z}_p$ [9]. Similar to elliptic curve cryptography, the ECDSA generally has increased protection for each bit used in the keys generated than DSA that is buttressed by the typical discrete logarithm problem [13]. This difference manifests in minimized overhead when establishing the same measure of defence in the keys for ECDSA in contrast to DSA [13]. Another consideration is that for electronics or computing systems that manage limited computing assets, ECDSA proves to be more effective under these restrictions than DSA [13].

3 Methodology: Implementing ECDSA

The first phase of ECDSA is the creation of the keys that will be involved in the ECDSA process. This is initiated in selection an elliptic curve $y^2 = x^3 + cx + d$ that is established within the context of a field $\mathbb{Z}$ [9]. The order of the elliptic curve group should have an enormous non-composite value for a factor to raise tenability, let this factor be denoted as a [9]. Acquire a coordinate $J \in y^2 = x^3 + cx + d \ mod \ p$ that has the characteristic of an order that is equivalent to the value of a [9]. Determine the value of b from the restriction $1 \leq b \leq a - 1$ in such a way that b is random or sufficiently randomized [9]. Acquire K by implementing the formula $K = b \times J$ then construct the public key from $y^2 = x^3 + cx + d \ mod \ p$, J, a, K but retain b to serve the purpose of being the relevant private key [9]. The previous algorithmic sequence is performed by all relevant participants in the transmissions that are to be subject to ECDSA [9].

This next portion, or phase 2, will detail the algorithmic sequence for some participant, Alice, to create an endorsement of a transmission [9]. To begin, Alice will determine the value of n from the restriction $1 \leq n \leq a - 1$ and that n is randomly obtained or sufficiently randomized [9]. Get the coordinate result from the formula $n \times J = M$, then assign $t = x_M \ mod \ a$, then check to ensure that $t \neq 0$ [9]. Now the inverse of n, n^{-1}, should be found within the context of $n^{-1} \ mod \ a$ for the next portion [9]. Now $u = n^{-1}(v(B) + bt) \ mod \ a$ is found, in which B is the content of the transmission

[9]. The presence of v denotes usage of the Secure Hash Algorithm otherwise referred to as SHA-1 [9]. As a precaution, there should be an effort to be certain that $u \neq 0$ or phase 2 should be restarted to ensure that ECDSA is implemented properly [9]. Now the endorsement for Alice's transmission content B is constructed of the tuple (t, u) for the recipient to validate once transmission content is obtained [9].

Phase 3 is the validation process once the transmission completes and Bob has the transferred content from Alice [9]. Bob will check that $0 < t < a$ and $0 < u < a$ are satisfied with t and u being whole numbers [9]. Obtain i by employing the formula $i = u^{-1}\ mod\ a$ [9]. Acquiring the produced hash from $v(B)$ is also a requirement [9]. Acquire the following through their respective formulas: $q_1 = v(B)i\ mod\ a$, $q_2 = ti\ mod\ a$, $q_1J + q_2K = L = (x_L, y_L)$, to set $g = x_L\ mod\ a$ [9]. Bob will consider the endorsement for the transmission content to be valid only when g is equivalent to t, otherwise the endorsement is viewed as invalid [9].

3.1 An Example of Implementing ECDSA

The two examples detailed below are for the general setup of ECDSA displayed on elliptic curve groups. As detialed throughout this work, there are precautions to be undertaken to produce tenable ECDSA implementations.

Example 7. *Let Alice and Bob be two participants in a transmission where Bob sends some written content to Alice and wants to endorse said written*

content so that Alice knows the transmission originates from Bob. For this endorsement, Bob and Alice both agree that they should employ the usage of ECDSA.

The elliptic curve that is agreed upon amongst Bob and Alice is the elliptic curve $y^2 = x^3 + 12x + 71$ *over* $\mathbb{Z}_{251}$. *This sets the value of* $a = 235$. *The coordinate* $J = (1, 109)$ *as well as* $b = 44$ *are determined by Bob. Now Bob can compose* $K = 44 \times (1, 109) = (208, 86)$. *Bob then provides the public key of* $y^2 = x^3 + 12x + 71$ *mod* 251, $J = (1, 109)$, $a = 235$, $K = (208, 86)$, *while at the same time retaining* b *as Bob's private key.*

Bob now will be creating an endorsement for the transmission to Alice. n *is found to be 23, so* $M = 23 \times (1, 109) = (132, 188)$, *from which* $t = 132$ *mod 235 is extracted. This is succeeded by Bob obtaining* $n^{-1} = 92$ *mod 235. The content Bob is trying to send is* $B =$ *"Did you get the code?" which has a hash of 9150019620529558939600795613016611310682693645 80. Bob then moves to* $u = (92 \times (9150019620529558939600795613016611310682693645 80 + 44(132)))$ *mod 235 = 126. The tuple for Bob's endorsement is therefore* $(132, 126)$.

Now Bob transmits the content to send to Alice along with Bob's own endorsement. Alice now checks that $0 < t < 235$ *and* $0 < u < 235$ *hold:* $0 < 132 < 235$ *and* $0 < 168 < 235$. *Alice will now find* $i = 126^{-1}$ *mod* $235 \equiv 166$ *while gaining the product of the hash function taking the transmission content as an argument* $9150019620529558939600795613016611310682693645 80$. *The following sequence of formulas are then employed:*

$q_1 = (9150019620529558939600795613016611310682693645 80 \times 166)$ *mod 235= 100,*

q_2 *= (132* $\times$ *166) mod 235 = 57,*

$L = 100 \times (1, 109) + 57 \times (208, 86) = (136, 70) + (190, 39) = (132, 188)$

In comparing g to t, Alice finds that g = t, therefore confirms the endorsement that was attached to the transmission contents. Bob is inferred to be the originator of the transmission.

Example 8. *Alice wants to send a message to Bob now, where the elliptic curve agreed amongst Alice and Bob is* $y^2 = x^3 + 33x + 19$ *mod* 103 *setting the value of a = 17 (Alice wants to use the largest prime factor of the order of the group which is 102). The coordinate chosen for J is* $J = (0, 15)$ *with b= 12 for Alice's private key. Alice then calculates* $K = 12 \times (0, 15) = (96, 75)$. *Alice then provides the public key:* $y^2 = x^3 + 33x + 19$ *mod* 103, $a = 17$, $J = (0, 15)$, $K = (96, 75)$ *as b= 12 is retained as Alice's private key.*

Alice will follow by finding n = 3, thus $M = 3 \times (0, 15) = (14, 54)$ *to extract* $t = 14$ *mod 17. Alice then obtains* $n^{-1} = 6$ *mod 17. The content that Alice is sending is B = "244" which has a hash of 769770500080508414547100290342667917626445259 3. Alice goes forward with* $u = (6 \times (769770500080508414547100290342667917626445259 3 + 12 \times 14))$ *mod* $17 \equiv 14$. *Now Alice sends the endorsement* $(14, 14)$ *along with B to Bob.*

Bob observes $0 < 14 < 17$ *as well as* $0 < 14 < 17$, *thus ensures the validity of* (t, u). $i = 14^{-1}$ *mod* $17 \equiv 11$ *is found along with* $v(B) =$

769770500080508414547100290342667917626445259 3.

Now $q_1 = 769770500080508414547100290342667917626445259 3 \times 11$ *mod 17* $\equiv$ *8,* $q_2 = 14 \times 11$ *mod* $17 \equiv$ *1, Bob then proceeds* $8 \times (0, 15) + 1 \times (96, 75) = (98, 48) + (96, 75) = (14, 54)$ *to extract g = 14. Bob finds g = t as 14 = 14 confirming Alice to be the originator of the transmission.*

4 Discussion

ECDSA remains an indispensable tool in the modern cryptographic toolset for the value of its attributes and performance. When the the order of the elliptic curve is a composite value, then the Hollman and Pohlig devised method diminish the time expense to acquire the private key [9]. To eliminate such a foible, the order of the elliptic curve should be a non-composite value [9]. The more common Pollard's Rho offense is noticed to have a time complexity of $\sqrt{\frac{\pi \times m}{2}}$, in which m is the greatest prime factor for the value of the order of the elliptic curve [9, 13]. The time complexity is measured in the quantity of elliptic curve operations that will be computed to yield the answer to the implementation of the elliptic curve discrete logarithm problem [9, 13]. Pollard's Rho has been accelerated thanks to the contributions of Wiener, Zuccherato, Vanstone, Lambert, and Gallant to $\frac{\sqrt{\pi \times m}}{2}$ [9, 13].

4.1 Possible Exploits

When attempting to setup exploits in opposing a community or group of participants that are employing ECDSA with exploits that may only target a singular private key per deployment the quantity of deployments must be replicated in relation to the quantity of targeted private keys [9]. The sum of coordinates for the elliptic curve necessitates larger allocations of computing assets than that of comparable encryption systems [9]. The brute force method of of attempting to decrypt private keys for ECDSA is identified as the Naive Exhaustive Search [13]. Naive Exhaustive Search will simply try to acquire h by repeated, sequential engagements of elliptic curve multiplication so that the private key b is deciphered [13]. The time complexity extends in accordance to the order of the elliptic curve involved, making Naive Exhaustive Search horribly inefficient as ECDSA scales upwards [13]. The next evolution of the Naive Exhaustive Search was the development of the Baby-Step Giant-Step Algorithm [13]. Baby-Step Giant-Step Algorithm exacerbates the strain on the memory assets to attempt to diminish the time expense in deciphering the private key [13]. The time complexity is shrunken to the $\sqrt{s}$, where s is the order of the elliptic curve group [13].

Pohlig-Hellman designed an exploit that diminishes the issue of time complexity for elliptic curves that maintain an order that is characterized by high factorability [13]. Employing the Chinese Remainder Theorem, the non-composite factors of the order of the elliptic curve group can be used to recontextualize the tenable conundrum to one that is less taxing on comput-

ing assets [13].

Pollard's Rho offense has also served for the development of two other options for launching offenses against ECDSA architecture [13]. The first is Pollard's Rho being adapted for concurrently running instances, which can be a potential means of bringing the time complexity down to a fraction of what said time complexity would be [13]. The Parallelized Pollard's Rho offense's time complexity can be examined to be $\frac{\sqrt{\pi \times m}}{2u}$ [13]. Note that u is the quantity of processors that are utilized to concurrently run the Pollard's Rho offense [13]. Parallelized Pollard Rho's offense is determined to be the optimal selection to employ for effectively all ECDSA architectures until another is suggested that is characterized by enhanced swiftness in execution [13]. The second option is Pollard's Lambda offense, which is dependent on the chance a foible embedded within set up of the ECDSA architecture when deployed [13]. Pollard's Lambda offense can be executed concurrently across a multitude of available processors as well [13].

Following a ramification of Pollard Rho offense being successfully carried out, there is an opportunity to initiate a Multiple Logarithms offense [13]. The prerequisite of deploying the Multiple Logarithms offense is that an equivalent coordinate J as well as elliptic curve exists for targets of the Pollard Rho's offense [13, 22]. Any similar offenses on replications of the elliptic curve and J can be observed can be aided by applying previous work to shave off time for any following deployments [13, 22]. A natural drawback of the Multiple Logarithms offense is that the offense can only be deployed

once a successful Pollard Rho's offense is deployed [13]. Another is the precondition that an equivalent elliptic curve and J are part of said deployments [13].

A foible in a specific classification of elliptic curves termed supersingular elliptic curves exists for another possible exploit [13]. The detriment of tenability manifests in the ECDSA architecture's elliptic curve discrete logarithm conundrum being recontextualized to the more favorable discrete logarithm conundrum [13]. The time complexity for obliterating the tenability of ECDSA when the ECDSA set up is founded on a supersingular is diminshed to that of DSA and RSA [13].

A foible can emerge within ECDSA architecture when using specific non-composite values in relation to the field the elliptic curve is structured upon intertwines cardinality and the order of the elliptic curve group itself [13]. The offense is termed the Semaev-Smart-Satoh-Araki offense, with the precondition delineated to be that the field cardinality is equivalent to the order of the elliptic curve group [13].

An elliptic curve that is structured upon anything other than a non-composite field can present innate foibles in the ECDSA architecture [13]. The instrument to be utilized within this scenario is the Weil descent allowing potentially acceleration of the timeline required for the offense to be successful beyond that achievable by Pollard's Rho in specific cases [13]. The Weil descent offensives have posed a severity of threat to such an extent that only non-composite fields have been recommended to remove the potential

foible that can be exploited with Weil descent [13].

4.1.1 Hash Offensives

There are suggestions to launch offensives with the aim being concentrated on the hash functions that will be applied within the ECDSA architecture [13]. The two attributes that enable the hash function to be toilsome to decipher is being insusceptible to collisions, or two inputs for the hash producing a homogeneous hash result, and preimaging, or finding the input that can generate a peculiar output of the hash function [13].

Another concern that has been raised are ECDSA architectures that maintain a method of producing values that are not arbitrary in nature [13].

4.1.2 Quantum Computing Offensives

Quantum computing assets have been considered the nemesis of elliptic curve cryptography thanks to Shor's Algorithm [23]. The threat of quantum computing assets are confined to the boundaries established by the current magnitude of capabilties and infrastructure of existing quantum computing assets.

4.2 Comparison with DSA and RSA as well and ECDSA Specific Advantages

To juxtapose ECDSA to DSA to RSA, a key detail is the necessity to implement similar cryptographic safeguards relative to one another. To achieve

equivalent tenability of the 1024-bit well-structured constructions of DSA and RSA, the ECDSA would necessitate at the least a 160-bit non-composite value order for the coordinate J with larger bit lengths noticably heightening the tenability [9]. According to Jurisic and Menezes, when measuring the connection of the quantity of bits that composes the key to that of the measure of protection granted by the asymmetric cryptographic architecture the elliptic curve-centered architecture is unrivaled [9]. Compounding such with the diminished necessity for an enormous quantity of bits for the creation of sustainable keys too displays the appeal and reason for the demand for ECDSA architecture in cryptographic domains [9].

Another means of evaluating the ECDSA to the DSA architecture concerns the the intricacy of the algorithms necessary to launch an offensive against it. ECDSA would necessitate computational efforts that would reach an exponential time complexity easily [21]. The RSA and DSA architectures are founded upon the integer factorization as well as the discrete logarithm conundrums respectively [21]. Integer factorization as well as discrete logarithm conundrums are subject to the number sieve method to attempt to obtain the resolve both conundrums [22]. The integer factorization and discrete logarithm conundrums are subject to attacks that necessitate an expense below that of an exponential time complexity [21]. Therefore, when adequately set up, ECDSA will be more tenable when measuring the time expense of offenses that will be utilized against it relative to DSA and RSA [21]. Condensed bit lengths of the keys involved produces less overhead for ECDSA

than that of the RSA [22]. For the processes involving commensurate keys for ECDSA and RSA, in all but the validation phase the ECDSA was observed to be faster within the two [22]. More research has been conducted on the expanse of the discrete logarithm conundrum along with its adaptations which have not brought any new potentialities for a general foible to exploit in ECDSA architecture [13].

4.3 ECDSA Acceleration

In terms of accelerating the overall ECDSA architecture, there a multitude of ways to do so. To add what has already been inscribed inside of this article, further options to accelerate the ECDSA architecture can be practiced. By implementing ECDSA over $\mathbb{F}_{2^k}$, or finite field of characteristic two, acceleration beyond the time expense of Z_p can be attained [9]. ECDSA can be set up with tinier elliptic curves thanks to the nature of the conundrum at the core of ECDSA, promoting acceleration of the ECDSA architecture [21].

4.4 ECDSA Suggestions

Some of the involved institutions within the field of cryptography that have presented attributes of desired elliptic curves for ECDSA set up include, but are not restricted to, FIPS, ISO/IEC, SEC, and IEEE [13].

5 Conclusion and Future Work

ECDSA has become a critical component of the modern cryptographic ecosystem. Having been derived from multiple topics in mathematics, ECDSA stands as a respected and reliable architecture for endorsements and ensuring the validity of data. When ECDSA is being measured and critiqued along with RSA and DSA for the cryprographic viability, ECDSA remains an invaluable option. The benefits of the ECDSA is key to ECDSA being persevered as a valued cryptographic signature architecture not only in general, but additionally within the sustainability of the consistently growing cryptocurrency cyberinfrastructure. ECDSA has been prominently included within the Bitcoin cyberinfrastructure for cryptographic purposes for a considerable span of time.

The research on ECDSA is continuing in the persevering motivations of those attempting to discover novel or remove present exploits that jeopardize the information that is safeguarded by ECDSA architectures. Said research will continue to be a priority to ensure that the cyberinfrastructure will be able to function successfully over continued usage.

5.1 Future Work

Future exploration into this topic will likely continue in regard to the examining the impact of quantum computing assets at a later date. Given enough progress, quantum computing assets could present a unquestionable adver-

sary for ECDSA through the underlying elliptic curve cryptography within the ECDSA architecture. Another future component to explore is the significance of ECDSA in the continued existence of cryptocurrencies. Given the burst in the variety of cryptocurrencies as well as the ease of implementing variants upon existing cryptocurrencies, observing the continued usage or emergence of alternatives for ECDSA will also be a target of inquiry. If alternatives to ECDSA are implemented on a noticable scale, then an examination can be made as to the benefits and costs of embedding the alternatives into cryptocurrency systems instead of ECDSA. Another possibility would be to develop a basic blockchain using a programming language to further expand on the inquiries of ECDSA within blockchains and the systems built atop said blockchains such as cryptocurrencies.

There could be new suggestions for strengthening ECDSA that can lead to more nuances emerging within ECDSA architecture, or the creation of distinguished variants of ECDSA altogether. Further exploration into these variants would likely provide a better perspective of the advancements and progression of ECDSA within the field of cryptography.

References

[1] Consumer Information. 2020. What To Know About Cryptocurrency. Federal Trade Commission. [online] Available at: `https://www.consumer.ftc.gov/articles/what-know-about-cryptocurrency`. [Accessed 10 October 2020].

[2] Van Valkenburgh, P., 2020. What's A Blockchain, Anyway? - Coin Center. [online] Coin Center. Available at: `https://www.coincenter.org/`

education/blockchain-101/whats-a-blockchain/. [Accessed 10 October 2020].

[3] 2020. Frequently Asked Questions. [online] Bitcoin Project. Available at: https://bitcoin.org/en/faq#what-is-bitcoin. [Accessed 10 December 2020].

[4] Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. [online] Bitcoin Project. Available at: https://bitcoin.org/bitcoin.pdf. [Accessed 10 December 2020].

[5] M. Gençoğlu, "Importance of Cryptography in Information Security", OSR Journal of Computer Engineering, vol. 21, no. 1, pp. 65-66,67, 2019. Available: http://www.iosrjournals.org/iosr-jce/papers/Vol21-issue1/Series-2/I2101026568.pdf. [Accessed 13 December 2020].

[6] National Institute of Standards and Technology. "Digital Signature Standard (DSS)", FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, vol. 186, no. 4, pp. 9-15, 18-20, 2013. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf. [Accessed 14 December 2020].

[7] C. RITZENTHALER, Introduction to elliptic curves. Campus Beaulieu: University of Rennes 1, 2020, pp. 3-5. https://perso.univ-rennes1.fr/christophe.ritzenthaler/cours/elliptic-curve-course.pdf

[8] Kak, A., 2020. Elliptic Curve Cryptography And Digital Rights Management. https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture14.pdf

[9] Jurisic, A. Menezes, A. Elliptic curves and cryptography, 1st ed. n.a.: Dr. Dobb's Journal, 1997, pp. 4, 6-10. http://www.cs.nthu.edu.tw/~cchen/CS4351/jurisic.pdf

[10] L. Washington, "Elliptic Curves", Discrete Mathematics and Its Applications, p. 107, 2008. Available: https://koclab.cs.ucsb.edu/teaching/ecc/eccPapers/Washington-ch04.pdf. [Accessed 30 January 2021].

[11] R. Sunuwar and S. Ketan Samal, Elgamel Encryption using Elliptic Curve Cryptography. Lincoln, Nebraska: University of Nebraska-Lincoln, 2015, p. 4.

[12] D. Boneh, R. Chen, G. Gong and S. Jiang, "Chapter 7: ElGamal and Elliptic Curve", Waterloo, 2021.

[13] Johnson, D., Menezes, A., Vanstone, S., "The Elliptic Curve Digital Signature Algorithm (ECDSA)", Certicom Corporation, Waterloo, 2001.

[14] Z. Wang, H. Yu, Z. Zhang, J. Piao and J. Liu, "ECDSA weak randomness in Bitcoin", Future Generation Computer Systems, vol. 102, pp. 507-513, 2020. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17330030. [Accessed 5 February 2021].

[15] S. Goldfeder et al., Securing Bitcoin wallets via a new DSA/ECDSA threshold signature scheme. 2015, p. 1. http://stevengoldfeder.com/papers/threshold_sigs.pdf

[16] L. Tessler and T. Byrnes, Bitcoin and quantum computing. N/A: Cornell University, 2018, pp. 1-5. https://arxiv.org/abs/1711.04235v2

[17] A. Malvik and B. Witzoee, Elliptic Curve Digital Signature Algorithm and its Applications in Bitcoin. Santa Barbara: University of California, Santa Barbara, 2015, p. 3. https://koclab.cs.ucsb.edu/teaching/ecc/project/2015Projects/Malvik+Witzoee.pdf.

[18] J. Ziegeldorf, R. Matzutt, M. Henze, F. Grossmann and K. Wehrle, "Secure and anonymous decentralized Bitcoin mixing", Future Generation Computer Systems, vol. 80, pp. 448-450, 2018. Available: https://www.sciencedirect.com/science/article/pii/S0167739X16301297. [Accessed 9 February 2021].

[19] A. Extance, "The future of cryptocurrencies: Bitcoin and beyond", Nature, vol. 526, no. 7571, pp. 21-23, 2015. Available: https://www.nature.com/news/the-future-of-cryptocurrencies-bitcoin-and-beyond-1.18447. [Accessed 10 February 2021].

[20] G. Sarath, D. Jinwala and S. Patel, "A Survey on Elliptic Curve Digital Signature Algorithm and its Variants", Computer Science Information

Technology (CS IT), pp. 121-136, 2014. Available: https://airccj.org/CSCP/vol4/csit42111.pdf. [Accessed 10 February 2021].

[21] D. Johnson and A. Menezes, "Elliptic Curve DSA (ECDSA): An Enhanced DSA", in 7th conference on USENIX Security Symposium, San Antonio, 1998, p. 13. https://static.usenix.org/publications/library/proceedings/sec98/invited_talks/menezes/ecdsa.ps.

[22] S. Levy, Performance and Security of ECDSA. Santa Barbara: University of California, Santa Barbara, 2015, pp. 2-3. https://koclab.cs.ucsb.edu/teaching/ecc/project/2015Projects/Levy.pdf

[23] D. WANG, Secure Implementation of ECDSA Signatures in Bitcoin. London: University College London, 2014, pp. 10, 13-14, 16,18-20,24-27. http://www.nicolascourtois.com/bitcoin/thesis_Di_Wang.pdf

A Appendix

A.1 MATHEMATICA

Usage of the software package MATHEMATICA is present within the span of this paper. MATHEMATICA is owned and maintained by Wolfram Research. It has a variety of capabilities in regards to building mathematical functions, providing calculations, or providing graphical components to better visualize mathematical implementations or concepts.
Wolfram Mathematica. Wolfram, 2021.

John McGee. "Adding Points on an Elliptic Curve."
http://demonstrations.wolfram.com/AddingPointsOnAnEllipticCurve/. Wolfram Demonstrations Project Published: March 7 2011