Control System Data Authentication and Verification Using Elliptic Curve Digital Signature Algorithm

Kenneth A. Fischer

Code 955, NSWCCD-SSES
1000 Kitty Hawk Avenue, Philadelphia PA 19112
Kenneth.a.fischer@navy.mil

Abstract— *Recent endeavours such as the Smart Grid and the Navy's Next Generation Integrated Power System, along with attacks on control systems such as Stuxnet, have highlighted the need for improved communications. Control system components such as PLCs and HMIs can no longer rely on simple heartbeat logic algorithms in order to verify communications. We can no longer rely on parity and checksum algorithms to determine that messages are coming through intact and unmodified. Advanced cryptographic algorithms for data authentication and verification are needed in messaging protocols between Programmable Logic Controllers (PLCs), Human-Machine Interfaces (HMIs), and sensors.*

Cryptographic algorithms such as RSA or the Digital Signature Algorithm (DSA) appear to provide a solution to this need on the surface except that the bit sizes required for implementing these solutions is not feasible for implementation in control system equipment. Elliptic Curve DSA (ECDSA) looks to be a promising solution due to the smaller key sizes which allow for smaller storage requirements and faster signature generations. The implementation of ECDSA can be complicated, but techniques such as using specialized prime field and binary curves as well as variations on ECDSA such as EC-KDSA can greatly increase the efficiency of the algorithm for use in control systems.

Index Terms—*PLC, HMI, Stuxnet, Smart Grid, NGIPS, ECDSA, Machinery Control System*

1. INTRODUCTION

Increasing demands in all sectors of an industrial society have led to an ever increasing need for more sophisticated controls and monitoring equipment and software. Control systems, once consisting of simple transmitters and relays have evolved into complex systems containing dozens of controllers communicating with each other, each containing tens of thousands of lines of code, for even the simplest processes. Complex Human-Machine Interface (HMI) mechanisms designed to give system owners and operators enhanced capabilities to remotely operate, maintain, and troubleshot equipment are being developed and deployed. At the core of most modern control systems is the Programmable Logic Controller (PLC), a device whose power lies in the ability of a Control System Engineer to quickly and easily implement complex control schemes at minimal cost. As a result, PLCs (originally designed to replace relay panels) have become prevalent in virtually every industrial environment from pharmaceutical plants to electrical power distribution systems.

The need for PLCs will significantly expand in the coming years, as countries with mature economies work tirelessly to develop new sophisticated power distribution networks required to support our growing economy. Our existing power grids were designed decades ago, with the main aim of delivering electricity from large power stations to households and businesses. The increasing efficiency and reliability requirements necessary to support our developing civilization in the face of increasing energy demands and the real threat of domestic terrorism and foreign aggression require significant modernization of these power distribution networks. The new "Smart Grid", as it commonly called, will be characterized by a two-way flow of electricity and information creating a widely distributed energy network. The control system required to support this energy network will be of an unheard of scale, the design of which will introduce significant challenges never before addressed.

In related efforts, the US Navy has been rapidly migrating to ship designs with propulsion, auxiliary, and weapons systems with significantly higher energy requirements than in the past. To address these requirements, modern ship designs such as the USS ZUMWALT DESTROYER (DDG1000) class are using Integrated Power Systems (IPS) that provide electrical power to propulsion and electrical loads from a common set of sources. To provide direction for future IPS development, the Navy initiated the Next Generation Integrated Power Systems (NGIPS) effort to provide smaller, simpler, more affordable, and more capable systems for all Navy ships.

The NGIPS effort is remarkably similar to the Smart Grid effort in multiple respects, and in both there is an increasing consensus that the controls communication infrastructure needs fundamental changes. In an automated electrical system, damage to a complex communication network, a hostile terrorist act, or even a failing component giving erroneous data can result in a control system taking improper actions that could result in large scale power failures on land and weapons, propulsion, or a complete electrical failure at sea or worse. In 2012 NSWCCD-SSES engineers uncovered a case (referred to as “Case 1” hereafter) where erroneous data communicated from a failing control system component over ControlNet resulted in a complete loss of propulsion and steering control whenever a ship was placed into full speed. This could have resulted in the ship colliding into the shore if it were not for conveniently placed Emergency Stop pushbuttons. It has become clear to controls engineers that more sophisticated methods are needed for verifying the integrity of the data and commands being issued to and from control systems.

Implementing control systems on a large, highly integrated scale introduces significant challenges partly because control system networks were not designed with security being primarily in mind. Historically, control system networks were designed to be completely physically isolated from other networks and therefore securing those control system networks seemed unnecessary. Instead, control system networks were designed to have maximum throughput with minimal to non-existent data loss. In recent years though control systems have gradually been getting connected to the Internet, mostly via corporate network systems, in order to meet business and maintenance requirements.

In order to secure networks, IT administrators have been applying traditional security measures to prevent attackers from gaining access to the corporate networks thus protecting control system networks. The last year particularly has highlighted the deficiencies with this model, as viruses such as Stuxnet have become rapidly prevalent. There is also significantly more risk in a compromised control system than a compromised corporate system. For example, an attacker could compromise the control system of a nuclear power plant resulting in a failure of the reactor cooling system. Therefore control system designers are realizing that not only do we need improved algorithms to verify that control system data is accurate, we need algorithms to verify that the data and commands to the control systems are authenticated (i.e. coming from a valid, recognized source).

In this paper, requirements for securing control systems are first surveyed in Section 2 with a particular focus on large scale complex systems such as those in use for modern naval applications and Smart Grid applications. Section 3 discusses current practices in the field of security and cryptography, which are then critiqued from a Control Systems Engineering perspective. Section 4 discusses our proposed solution utilizing a modified variant of the Elliptic Curve Digital Signature Algorithm (ECDSA), highlighting the advantages for its use in Machinery Control System (MCS) applications and describing a path forward for implementation by Control System Engineers. Section 5 describes the current challenges and areas of future work. Section 6 summarizes and concludes the paper.

2. CONTROL SYSTEM SECURITY REQUIREMENTS

Controls engineers have long recognized the need to verify that components within a control system are communicating and that the failure of communications between control system components should result in critical high priority alarms with possible equipment shutdowns. Since control system communications operate in real time, 24 hours a day, 7 days a week, algorithms are needed to detect a failure in communications as soon as it occurs. Traditionally, “heartbeat” logic is implemented between each pair of communication devices.

As long as a communications failure alarm does not occur, then the data being transmitted between the two PLCs is considered to be both valid and sourced between the communicating pair. This kind of logic has proven to be very effective for general network health monitoring. Issues in communication, primarily in the physical or transport layer, can be easily detected using this method. For control system networks that are physically isolated from any other network, this is generally sufficient to implement an effective control scheme. Unfortunately, this method does not protect against any kind of atypical equipment failure (or potential sabotage) such as that documented in Case 1 described earlier. It also does not protect against sophisticated attacks such as a "man-in-the-middle" attack launched by enemy forces.

2.1. Literature Review on Smart Grid Control System Application Requirements

A number of papers have been written to introduce the Smart Grid concepts and provide a general overview of the requirements and challenges involved in developing a Smart Grid.

Bouhafs, Mackay, and Merabti (2012) [1] identified a number of general requirements including communications and electrical generation needed in order to fully realize the Smart Grid vision. They noted that underlying communications protocols will need to be more flexible and enable horizontal data exchange between controllers and remote terminal units (RTUs). The current "heartbeat" logic concept would not be useful in an implementation where data could flow from a source through multiple sources to a target since it only verifies the link between pairs and not the data itself. They went on to note that in the event the Internet is used to connect equipment in the Smart Grid strong encryption and authentication measures must be taken to ensure the security of the data in transit.

Yan, Qian, Sharif, and Tipper (2012) [2] noted that it is necessary to have guaranteed Quality of Service (QoS) for the communications and networking technology. In particular they highlighted latency, bandwidth, interoperability, scalability, and security requirements. Of particular interest is the author's analysis of bandwidth requirements which showed that there will be significant challenges in this area. Therefore, adding a significant number of bits in any communications protocol for control systems could have a profoundly negative impact on the operation of the Smart Grid as a whole. The authors also noted that the effort required to provision symmetric keys (i.e. keys between each pair of communicating devices) into thousands of devices would be too expensive or insecure. They noted that the development of key and trust management schemes for large network deployments would be required. While Navy systems are small enough that they would not suffer from the same kinds of limitations, it seems obvious that a solution must be developed for Navy systems that would be applicable to all future controls systems including the Smart Grid, particularly in support of modernized shore power connections for naval systems.

Yan, Qian, Sharif, and Tipper (2012) [3] in a related paper noted that new functions in the Smart Grid such as demand response introduce significant new cyber-attack vectors such as a malware that initiates a massive coordinated and instantaneous drop in demand. This attack could result in substantial damage to distribution, transmission, and generation facilities. Research ongoing at NSWCCD-SSES has also noted this risk as applicable to Navy systems, particularly in combat scenarios with the use of advanced weapon systems such as the railgun. The authors also noted that a major difference between Smart Grid controls communication and the Internet is that the controls data is significantly more concerned with message delay and timing constraints.

Liu, Ning, and Reiter (2009) [4] in their work presented a notable example of a new type of attack, called false data injection attacks, that highlights the very real risk of attacks targeting data integrity.

Baumeister (2011) [5] noted that most information systems uses a Public Key Infrastructure (PKI) solution, but that the nature of power grid systems creates additional PKI requirements not present in traditional information systems. This same statement can be generalized to apply to all control systems. For example, Baumeister noted that control systems must make informed decisions regularly, and that it is unreasonable to expect a control system to go down or revert to a less efficient predecessor every time a certificate is unavailable. For example, what happens when a certificate from a sensor expires? In an information system, the impact of expired certificates is insignificant and they can be renewed when discovered. However, in a control system this could cause the process (such as electric flows) to be incorrectly altered.

In response to the number of concerns related to the Smart Grid and Cyber Security, NIST established the Smart Grid Interoperability Panel (SGiP) Cyber Security Working Group which published NISTIR 7628 (2010) [6]. This document broke down the various kinds of communications that would be prevalent in a full international Smart Grid system into a number of categories. SGiP then identifies the unique security requirements for each of these categories, focusing on the three areas of confidentiality, integrity, and availability. Most, but not all of the categories identified by SGiP are directly or indirectly applicable to control systems. In reviewing the categories, it appears that all of them have significant overlap with NGIPS efforts as well as industrial control systems in general. Going through the requirements of these categories as identified by SGiP, the primary concerns are data integrity and authentication. Data encryption can be useful in some circumstances, however it can be shown that it is not as critical as the other two requirements for most types of control system communications.

2.2. Literature Review on NGIPS Requirements

Most of the literature focusing on the NGIPS effort has focused on areas such as electrical generation, propulsion, power conversion and distribution, energy storage, and zonal survivability. In NAVSEA (2007) [7], the NGIPS architecture is broken up into seven modules types. The PCON module is of particular interest to controls engineers, as it consists of the software and communications protocols necessary to operate the system.

Doerry (2009) [8] noted a number of functions that PCON should implement, stating that the software should be developed for robustness in anticipation of future changes both in the life of a ship and for modifying for use across multiple ship classes. The functions identified are remarkably similar to the control system functions required for the development of a Smart Grid, with the notable exception of Quality of Service (QoS) and Mission Priority Load Shedding. As a result, the same need for data authentication and verification in the Smart Grid would be applicable to NGIPS, particularly in functions such as maintenance support where it becomes increasingly common for ships to transmit data to and from shore based services for software upgrades and maintenance / troubleshooting support.

Desired requirements for QoS also introduce the need to ensure that commands being transmitted across the ship for electrical service are genuine. As noted by Doerry, a typical cause of a QoS failure is the shifting of electrical power sources from ship to shore, and that communications will be required with the terrestrial power system command and control centers. Failure of the ship and shore to properly establish valid communications could result in power instabilities for both.

The increasing prevalence of computer viruses specifically targeting control systems will introduce new challenges to the mission readiness of a ship in times of war. By attacking PCON, an enemy may be able to cause a control system to incorrectly transfer loads which could result in a failure of propulsion or weapon systems (or both) at a critical moment. Modern weapon systems produce substantial electrical loads that may require realigning of the ship's electrical distribution prior to being operational.

The Navy has been putting in significant effort to develop open architecture approaches in the development of control system software to support not only NGIPS development but also to support development of control systems fleet wide. Doerry, Scherer, Cohen, and Guertin (2011) [9] pointed out that information assurance and security needs to be thought of at the outset of any new MCS design, stating that confidentiality, integrity, and availability of data must be assured. They also highlight that the software should perform error detection (and error correction if possible) along with filtering of the sensor data

3. CURRENT PRACTICES

Within the field of cryptography there are multiple solutions providing various degrees of secure communication. In order to be effectively used to establish secure communications these solutions have the following fundamental objectives: Confidentiality, Data Integrity, Data Origin Authentication, Entity Authentication, and Non-repudiation. There are essentially two main categories of cryptographic solutions, symmetric-key cryptography and public-key cryptography.

3.1. Symmetric-key Cryptography

Symmetric-key Cryptography includes schemes such as the Data Encryption Standard (DES) (now obsolete), RC4, and the Advanced Encryption Standard (AES) to achieve confidentiality. They may also be used with a message authentication code (MAC) algorithm such as HMAC to achieve data integrity and data origin authentication.   In a typical symmetric-key cryptography scheme two parties already share a secret key *k* that has been communicated to the parties by some other means (typically a physical secure channel such as a trusted courier, or by using a public-key cryptography scheme to negotiate a shared secret key). Party A wishing to transmit to B uses one of the previously mentioned schemes to compute a ciphertext $c = ENCk(m)$ to be sent to B. B then receives the message and uses the same *k* (and knowing the same scheme used to encrypt m used by A) to recover the plaintext message $m = DECk(c)$.

If data integrity and data origin authentication are desired, then the same principles apply, however instead of encrypting the message *m* into ciphertext *c* a tag *t* is first computed where $t = MACk(m)$ of the plaintext message using a MAC algorithm (of which there are many) and the key. The plaintext message and the tag are both transmitted, and the receiver can use the plaintext message to compute its own tag *t'*. If

$t = t'$ then the receiver can accept the message as having originated from the source.

While symmetric-key cryptography can be very efficient, the key distribution and key management problems tend to render it ineffective for large scale systems communicating to multiple partners [10]. In a network of *N* entities, each entity may have to maintain keying material with each of the other *N-1* entities. Some symmetric-key systems attempt to alleviate this problem by using an online trusted third party that distributes the keys as required, however for control systems this creates a single critical point of failure that will be unacceptable as control systems become more and more distributed and de-centralized such as in the Smart Grid. For control systems on Navy ships such a single point of failure would be unsatisfactory regardless of the control system design.

3.2. Public-key Cryptography

Public-key cryptography began in 1975 to address the aforementioned limitations in symmetric-key cryptography. Unlike symmetric-key schemes, public-key schemes require the keying material that is exchanged to only be authentic, but not secret. Additionally, instead of each pair of entities sharing a secret key, each entity selects a single pair of keys (e, d) consisting of a public key e and a related private key d. The entity keeps the private key a secret from all other entities and shares the public key with all other entities. The keys are mathematically related but share the property that it is computationally infeasible to determine the private key solely from knowledge of the public key. Deriving the private key from the public key is equivalent to solving a computational problem that is believed to be intractable.

The most commonly used public-key cryptography scheme is RSA, named after its inventors Rivest, Shamir, and Adleman. It was first proposed in 1977 shortly after the discovery of public-key cryptography. In RSA, the public key consists of a pair of integers (n, e) where n is the modulus. The modulus is a product of two randomly generated (and secret) primes p and q which are of the same bitlength. RSA encryption and signature schemes use the fact that $m^{ed} = m \ (mod\ n)$. The hardness in breaking RSA is based on the integer factorization problem, i.e. determining the secret primes p and q from the public key for large values of bitlength l. In the RSA digital signature generation and verification algorithms, as in all signature schemes, the signer first generates a cryptographic hash H which acts in a similar manner as the tag in symmetric-key encryption. The signer then generates the signature and transmits the message m along with the signature s to a verifying party.

In 1976 Diffie and Hellman proposed developing a key agreement protocol based on the discrete logarithm problem (DLP) [10], which like the integer factorization problem used in RSA is computationally infeasible to solve. Discrete logarithms are group-theoretic analogues of ordinary logarithms. For example, an ordinary logarithm $log_a(b)$ is a solution of the equation $a^x = b$ for x. In a discrete logarithm, you have a group G which consists of a range of integer values from 0 to n-1. If a and b are elements in the group then a solution of x of the equation $a^x = b$ is called a discrete logarithm to the base a of b in the group G. In a discrete logarithm public-key cryptography system a key pair is associated with a set of domain parameters (p, q, g).

In 1984 ElGamal described discrete logarithm public-key encryption and signature schemes, and since then many different variants have been proposed leading up to the establishment of the Digital Signature Algorithm (DSA) [10]. DSA was proposed in August 1991 by the U.S. National Institute of Standards and Technology (NIST) and was specified in a U.S. Government Federal Information Processing Standard (FIPS 186), adopted in 1993. A minor revision was issued in 1996 as FIPS 186-1, which was expanded further in 2000 as FIPS 186-2 and again in 2009 as FIPS 186-3.

The weakness of public-key cryptography is that the security of an algorithm cannot exceed its key length (measured in bits) since any algorithm can be cracked by brute force. A key therefore should be sufficiently large enough such that a brute force attack is infeasible – i.e. it would take too long to execute. If there is some indicator that an attack may exist to feasibly break a key for a particular algorithm in an efficient manner for some bit length, then the size of the key is increased to provide additional security. The key size to security level ratio is not the same for all categories of algorithms

As of 2003 RSA Security claims that 1024-bit RSA keys are equivalent in strength to 80-bit symmetric keys, 2048-bit RSA keys to 112-bit symmetric keys and 3072-bit RSA keys to 128-bit symmetric keys. RSA claims that 1024-bit keys are likely to become crackable sometime between 2006 and 2010 and that 2048-bit keys are sufficient until 2030. An RSA key length of 3072 bits should be used if security is required beyond 2030. NIST key management guidelines further suggest that 15360-bit RSA keys are equivalent in strength to 256-bit symmetric keys.

These key lengths, while implementable in Information / Corporate systems, are infeasible in Control Systems where processing power and data storage is limited. Therefore an alternative public-key algorithm is needed that provides the benefits of algorithms such as RSA and DSA without the excessive key lengths required by these algorithms.

3.3. Limitations in Control Systems

Information / Corporate Systems will typically consist of x86-based architecture computers running either Windows or Linux operating systems and a host of other software programs provided by multiple vendors to provide an integrated solution. At the heart of the Control System are Programmable Logic Controllers (PLCs), which use vendor specific developer environments to write software following IEC 61131-3 guidelines (ladder logic, function blocks, etc) to implement a solution that is both easy and cheap to design and is very effective for controls.

The downside of these PLCs is that they tend to have significantly less processing power and storage capabilities as they are designed to run very specific software programs extremely efficiently, non-stop, for 20 years or more. As a result these processors would not be capable of performing the complex mathematical operations required in a timely manner for algorithms like RSA. PLC devices were simply not designed to process such large bit sized integers. If a PLC manufacturer were to choose to develop such a device capable of processing 10K to 15K bit integers then control network protocols, particularly ControlNet, DeviceNet, Profibus, and other Fieldbus protocols would need fundamental changes in order to handle the increased overhead resulting from transmitting such long keys and the resulting digital signatures.

An alternative to PLCs are VERSAmodule Eurocards (VME) which add significant complexity to the design of a control system but have greater processing power and contain the same input / output processing capabilities as PLCs. Unfortunately, the increased processing power is still not sufficient to generate and verify signatures over such large key sizes to enable timely communications between control system components.

Another alternative to PLCs are SoftPLCs. SoftPLCs are essentially programmed in the same manner as regular PLCs, but contain additional underlying base code designed to interface with an operating system (typically Windows NT based operating systems) in order to run the IEC 61131-3 code on an x86-based architecture.

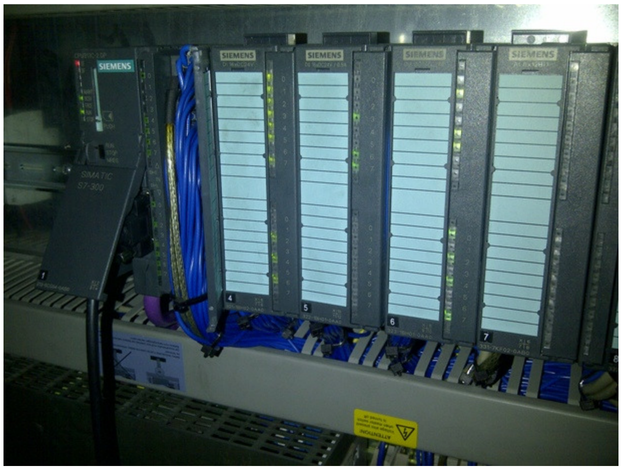


Figure 1. PLC Rack

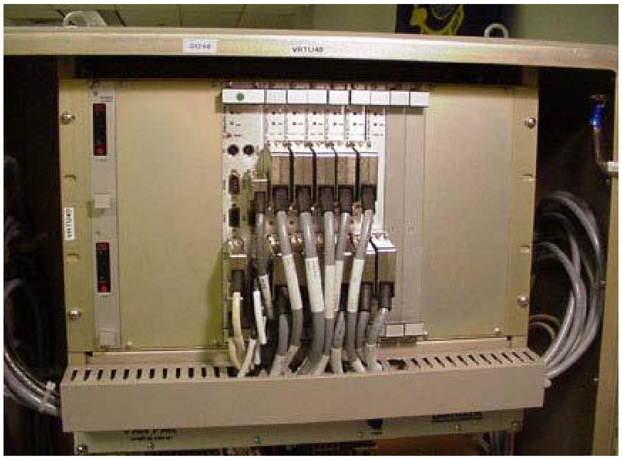


Figure 2. VME Rack

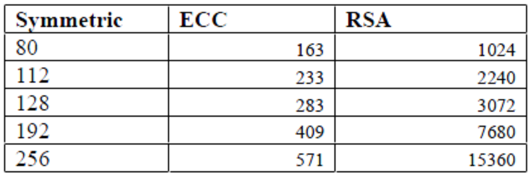
Since VME cards can be obtained that use the x86 architecture, in recent years the Navy has been implementing control systems on ship classes that use SoftPLCs running on VMEs to obtain the best of both worlds. This can be a complicated and expensive solution that is still more in the research and development stage and will likely not be implemented in either the Smart Grid or regular industrial control systems. However it is possible from a research perspective to perform cryptography testing on SoftPLCs using VMEs to do “proof of concept” testing in order to determine the validity of a solution before expending significant resources in developing an independent and complete PLC solution.

4. ECDSA In Machinery Control Systems

4.1. Introduction

Elliptic curve public key cryptosystems were first independently proposed by V.S. Miller (1985) [11] and by N. Koblitz (1987) [12] and have the advantage of requiring significantly smaller key sizes than other public-key algorithms to obtain equivalent security. Figure 3 below shows a chart of comparable key sizes for equivalent levels of security.

Elliptic curve public key cryptosystems have only begun to recently be used in commercial systems, and adoption has been slow. This is primarily due to concerns about intellectual property, as a number of optimizations and special algorithms used to increase efficiency have been patented in recent years. Despite these concerns, elliptic curve cryptography (ECC) has grown resulting in its inclusion in standards by accredited standards organizations such as ANSI (American National Standards Institute) [13, 14], IEEE (Institute of Electrical and Electronics Engineers) [15], ISO (International Standards Organization [16, 17], and NIST (National Institute of Standards and Technology [18]. The Elliptic Curve Digital Signature Algorithm (ECDSA) is an ECC-based algorithm equivalent of DSA.



Symmetric	ECC	RSA
80	163	1024
112	233	2240
128	283	3072
192	409	7680
256	571	15360

Figure 3. Comparable Key Sizes (in bits) [19]

4.2. Mathematical Foundation

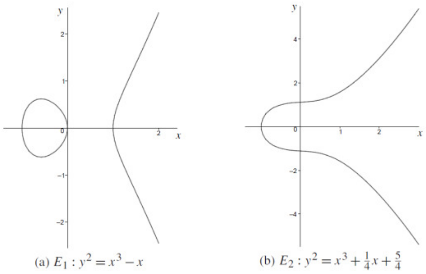
Elliptic curves are most commonly shown in the form of the simplified Weierstrass equation in the form of:

$$y^2 = x^3 + ax + b$$

where

$$4a^3 + 27b^2 \neq 0$$

This condition is critical to ensure that the elliptic curve is "smooth", i.e. that there are no points at which the curve has two or more distinct tangent lines. The curves shown in Figure 4 illustrate examples of elliptic curves satisfying this condition.



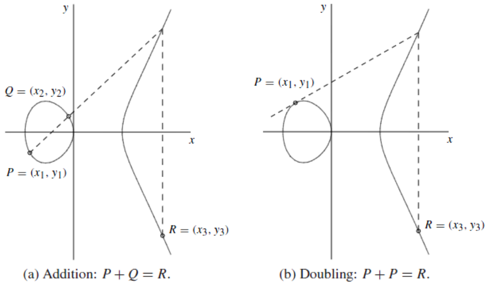
(a) $E_1 : y^2 = x^3 - x$

(b) $E_2 : y^2 = x^3 + \frac{1}{4}x + \frac{5}{4}$

Figure 4. Sample Elliptic Curves [10]

The security of ECC is based on the elliptic curve discrete logarithm problem (ECDLP), which arises when elliptic curves are used over finite fields. The ECDLP is [10]: given an elliptic curve *E* defined over a finite field F_q, a point $P \in E(F_q)$ of order *n*, and a point $Q \in \langle P \rangle$, find the integer $l \in [0, n\text{-}1]$ such that $Q = lP$. The integer *l* is called the discrete logarithm of *Q* to the base *P*, denoted $l = log_P Q$. The elliptic curve domain parameters for cryptographic schemes should be carefully chosen in order to resist all known attacks on the ECDLP. However, since the methods for computing solutions to the ECDLP are much less efficient than methods used for computing solutions to integer factorization (used in RSA) ECC can provide the same level of security as RSA with smaller key lengths, and scales much better at higher levels of security than RSA. This is critical when considering the limited available storage on control system devices.

When an elliptic curve *E* is defined over a field (call it *K*) there exist rules for adding two points in *E(K)* to give a third point in *E(K)*. This operation is commonly known as point addition. Furthermore, there also exist rules for doubling a point as to obtain another point, an operation commonly known as point doubling. Figure 5 shows a geometric representation of both of these rules.



(a) Addition: $P + Q = R$.

(b) Doubling: $P + P = R$.

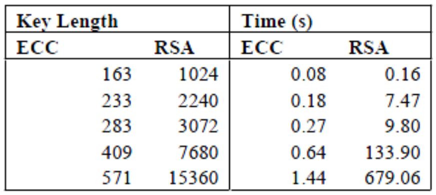
Figure 5. Geometric Representation of Point Addition and Point Doubling [10]

4.3. Comparison of ECC to RSA

As discussed previously, the primary need for control systems is to verify data integrity and authentication. This need is fulfilled in corporate / non-control systems through the use of DSA. The large bit sizes required result in communication latencies (resulting from the combined time of generating the signature by the sender and verifying the signature by the receiver) that are unacceptable in control system applications. These latencies will only continue to increase as key sizes scale exponentially in the future.

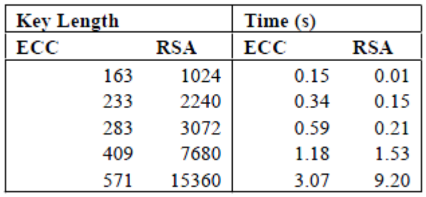
Elliptic curves offer a promising alternative solution for control system applications. Comparative studies have been performed on the timing of ECC vs RSA utilizing an Intel Pentium 4 2.0 GHz machine with 512 MB of RAM on a 100KB text file used as a message [19]. The results show that ECC outperforms RSA significantly in key generation (over 470 times faster at higher key sizes), and performs signature generation faster than RSA for higher key sizes (approximately 3 times faster). RSA outperforms ECC in signature verification significantly for all key sizes. These results are shown in Figures 6 through 8.

For control system applications, these results indicate that for higher key sizes ECC would slightly outperform alternative algorithms in a typical message transmission between two controllers. ECC would vastly outperform alternative algorithms each time keys are refreshed (which would be required anywhere from every hour to every day depending on the application).



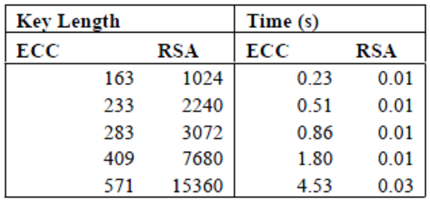
Key Length		Time (s)	
ECC	RSA	ECC	RSA
163	1024	0.08	0.16
233	2240	0.18	7.47
283	3072	0.27	9.80
409	7680	0.64	133.90
571	15360	1.44	679.06

Figure 6. ECC vs RSA Key Generation [19]



Key Length		Time (s)	
ECC	RSA	ECC	RSA
163	1024	0.15	0.01
233	2240	0.34	0.15
283	3072	0.59	0.21
409	7680	1.18	1.53
571	15360	3.07	9.20

Figure 7. ECC vs RSA Signature Generation [19]



Key Length		Time (s)	
ECC	RSA	ECC	RSA
163	1024	0.23	0.01
233	2240	0.51	0.01
283	3072	0.86	0.01
409	7680	1.80	0.01
571	15360	4.53	0.03

Figure 8. ECC vs RSA Signature Verification [19]

4.4. Control System Applications

For control system applications it is proposed that data traveling across any controls network is first signed by the sending controller at the application layer. The signature is then transmitted across the control network along with the data and is subsequently verified by the receiving controller at the application layer. This will allow the receiving controller to verify both the authenticity and integrity of the data, and will allow both controllers to utilize any available communications protocol to transmit. Signing and verifying the data at the application layer will also allow intermediary controllers to relay data silently without additional overhead which would impact transmission time.

For existing control systems, ECC can be retrofitted into the software by creating an IEC 61131-3 compatible function block that evaluates the data to be transmitted and generates the corresponding signature. A similar style block would be used on the receiving controller to evaluate the data received and verify the signature.

5. Challenges and Future Work

Signature verification is of particular concern in looking at implementing ECC signature algorithms for control systems. At stronger levels of security with larger key sizes, ECDSA will outperform RSA for the total message transmission (including both signature generation and verification) since ECC signature verification timing scales linearly while RSA signature generation timing scales exponentially. In order for ECDSA to become an effective algorithm for control system implementations, the timing of signature verification will need to be improved.

5.1. Domain Parameter Selection

In designing an improved implementation of ECDSA for control systems the most critical component is the selection of the elliptic curve type (either prime field curves or binary field curves) and the corresponding domain parameters that define the curve. In general, for elliptic curves over a finite field $\mathbb{F}_{q^m}$, the following domain parameters are required to be specified:

$$D = (q, FR, S, a, b, P, n, h)$$

Where:

q – field order
FR – field representation
S – seed, used if the elliptic curve was generated randomly
a & b – coefficients in the field $\mathbb{F}_{q^m}$ that define the equation over the field
P – the base point $P=(x_p, y_p) \in \mathbb{F}_{q^m}$ that has prime order
n – the order of P
h – the cofactor $h=\#E(\mathbb{F}_{q^m}) / n$

In order to increase efficiency of cryptographic implementations and to prevent all known attacks, various standardized domain parameters have been developed for elliptic curves over both prime and binary fields. These standardized, or "special", curves have been published by the SECG [20] and are recommended by NIST for use in U.S. government applications. However, in order to guard against future attacks against these curves one might decide to generate a new curve randomly which has a validation process that proves the new curve resists all known attacks on the ECDLP. Fortunately algorithms exist to accomplish this very task [10].

The conventional wisdom of ECC has been, as described by Koblitz [21]:

- For greatest security choose parameters as randomly as possible
- It is safest to choose the defining equation to have random coefficients
- It is okay to use special curves for reasons of efficiency if you insist, however that choice may one day come back to bite you

Recent work on isogenies in elliptic curve cryptography has shown that there are various scenarios in which a special curve is better than a random curve. Isogenies, simply put, allow one to transport the discrete logarithm problem from one curve to another. It is "random self-reducible" within a set of endomorphism classes with small conductor gaps. Work in this area has shown that we need to assume that some version of a Weil Descent attack or another approach someday will lead to a faster-than-sqrt attack on a small but non-negligible portion of random curves [21].

It is unknown at this time whether random curves are truly more secure than special curves. Therefore, for control systems for the Smart Grid and NGIPS following the NIST recommendation seems to be the most prudent.

5.2. Reducing Computational Cost

There are a number of mathematical techniques useful for improving the efficiency of ECC by reducing the computational cost. For example, the formulas for point addition and point doubling require field inversions and field multiplications. These are complex operations for the very large fields typically used in cryptographic applications. If inversion in a field K is significantly more expensive than multiplication (and it typically has a cost of roughly 80 field multiplications), then the use of a technique known as projective coordinates may be advantageous to use. There are a number of projective coordinate approaches worth further investigation such as affine coordinates, Jacobian coordinates, Jacobian-affine coordinates, Chudnovsky coordinates, mixed Jacobian-Chudnovsky coordinates, and mixed Chudnovsky-affine coordinates. Each system has its own advantages which will require detailed investigation and analysis.

In ECC point multiplication (the computation of kP where P is a point on the curve and k is an integer) dominates the execution time of ECC schemes. There are three cases where point multiplication occurs:

- kP where precomputation must be online
- kP for P known in advance and precomputation may be offline
- $kP + lQ$ where only the precomputation for P may be done offline

The last two cases are motivated by ECDSA, where signature generation requires a calculation kP where P is fixed, and signature verification requires a calculation $kP + lQ$ where P is fixed and Q is known a priori.

There are a number of mathematical techniques that can be used in order to increase the efficiency of point multiplications. Some methods, such the "sliding-window methods", require that extra memory be available. Additionally, if the point P is fixed and some storage is available, then the point multiplication kP can be accelerated by pre-computing some of the data dependent on P using a type of fixed-base windowing method such as that proposed by Brickell, Gordon, McCurley, and Wilson [10]. Shamir's Trick is yet another method used specifically to speed up the calculation of $kP + lQ$ by performing simultaneous multiple point multiplication [10].

There are also a number of other alternative elliptic curve signature schemes, such as Elliptic Curve ElGamal Signatures (ECES) and Abbreviated ECES Signatures (AECES). A particularly promising variation of ECDSA is known as the Elliptic Curve Korean Certificate-based Digital Signature Algorithm (EC-KCDSA). In EC-KCDSA the signer's private key is an integer $d \epsilon_R [1, n-1]$ as is in ECDSA, but the public key is instead $Q= d^{-1}P$ (instead of dP). This allows for the design of signature generation and verification procedures that do not require performing modular inversion and therefore could potentially be more applicable in meeting control system needs should ECDSA prove impractical. EC-KCDSA has been proven secure under the assumptions that the discrete logarithm problem is intractable and that the hash function is a random function.

Another promising variation, proposed by Antipa et al (2005) [23], involves reconstructing the ephermeral elliptic curve point R from the signature component r. In other words one converts the ECDSA signature (r, s) over some message m to a new ECDSA* signature (R, s). Antipa et al provide a general procedure for this change which accepts the ECDSA signature as an input, performs the reconstruction/conversion, and returns either acceptance or rejection of the signature. This speeds up ECDSA signature verification by 35-40% at the cost of only a small number of bits appended to traditional ECDSA signatures.

Further analysis and testing is required on both EC-KCDSA and ECDSA* for control system applications. Currently the EC-KCDSA and ECDSA* algorithms are non-compliant with any of the existing ECDSA standards.

5.3. IEC 61131-3 and IEC 61499 ECC Implementations

As discussed previously, PLC hardware is specifically designed to run software complaint with IEC 61131-3. Newer PLC products are also capable of executing IEC 61499 complaint software, designed to eventually replace IEC 61131-3. An implementation of an ECDSA variant written in software compliant with these IEC standards could presumably utilize the underlying PLC hardware in the most efficient manner possible. Such an implementation could require minimal to no changes in PLC hardware by PLC equipment manufactures except possibly an increase in memory to allow for precomputation storage (although PLC manufacturers have been increasing on board memory steadily for years). This would facilitate the ability of industry and the Navy to incorporate ECDSA into existing systems at significantly reduced cost.

6. Conclusion

This paper has shown that algorithms for control system data authenticity and integrity will be critical for next generation control systems, particularly in Navy and Smart Grid applications. Traditional protocols used widely in corporate / non-control systems will be unsuitable for control system applications due to the exponential growth of key sizes. The ECDSA offers a promising alternative to these protocols, however improving signature verification will be critical for its success in control system applications. Developing improved implementations to reduce computational cost and determining ideal domain parameters will help to improve signature verification. Pursuit of more radical approaches such as the EC-KCDSA and ECDSA* variants will likely be required in order to develop and optimize a control system solution. An IEC complaint implementation of the final algorithm would expedite adoption by industry and the Navy and reduce cost.

References

[1] Bouhafs, F., Mackay, M., Merabti, M. (2012). "Links to the Future." IEEE Power and Energy Magazine 1540-7977/12, pp. 24-32

[2] Yan, Y., Qian, Y., Sharif, H., Tipper, D. (2013) "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements, and Challenges." IEEE Communications Surveys & Tutorials, Vol. 15 #1, pp. 5-20

[3] Yan, Y., Qian, Y., Sharif, H., Tipper, D. (2012) "A Survey on Cyber Security for Smart Grid Communications." IEEE Communications Surveys & Tutorials, Vol. 15 #1, pp. 998-1010

[4] Liu, Y., Ning, P., Reiter, M. (2009) "False data injection attacks against state estimation in electric power grids." In Proc. ACM Conference on Computer and Communications Security (CCS 09)

[5] Baumeister, T. (2011) "Adapting PKI for the Smart Grid." IEEE SmartGridComm, 978-1-4577-1702-4/11

[6] NISTIR 7628 Volume 1 (2010) "Guidelines for Smart Grid Cyber Security: Vol. 1, Smart Grid Cyber Security Strategy, Architecture, and High-Level Requirements."

[7] Naval Sea Systems Command (2007) "Next Generation Integrated Power System Technology

Development Roadmap." Ser 05D/349 of 30 Nov 2007

[8] Doerry, N., CAPT USN, " Next Generation Integrated Power Systems for the Future Fleet," Presented at the Corbin A. McNeill Symposium, United States Naval Academy, Annapolis, MD, March 30, 2009

[9] Doerry, N., Scherer, T., Cohen, J., Guertin, N., "Open Architecture Machinery Control System ," Presented at ASNE Intelligent Ships Symposium 2011, May 25-26, 2011, Philadelphia, PA. Also Published in ASNE Naval Engineers Journal, Mar 2012, Vol 124 No. 1, pp. 101-114.

[10] Hankerson, D., Menezes, A., Vanstone, S. (2004) Guide to Elliptic Curve Cryptography

[11] Miller, V.S. (1985). "Use of elliptic curves in cryptography." Advances in Cryptology Proc. Crypto '85, LNCS 218, H.C. Williams, Ed., Springer-Verlag, pp. 417-426

[12] Koblitz, N. (1987). "Elliptic curve cryptosystems." Mathematics of Computation, Vol. 48, No. 177, p. 279-287

[13] ANSI X9.62 (1999). "Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)

[14] ANSI X9.63 (2000- Working Draft). "Public Key Cryptography for the Financial Services Industry: Elliptic Curve Key Agreement and Key Transport Protocols."

[15] IEEE 1363-2000 (2000) "Standard Specifications for Public-Key Cryptography."

[16] ISO/IEC 14888-3 (1998). "Information Technology – Security Techniques – Digital Signatures with Appendix – Part 3: Certificate Based Mechanisms."

[17] ISO/IEC 15946 (1999 – Committee Draft). "Information Technology – Security Techniques – Cryptographic Techniques Based on Elliptic Curves."

[18] NIST FIPS Pub 186-3 (2009). "Digital Signature Standard."

[19] Jansma, N., Arrendondo, B. (2004). "Performance Comparison of Elliptic Curve and RSA Digital Signatures." (http://nicj.net/files/performance_comparison_of_elliptic_curve_and_rsa_digital_signatures.pdf) Accessed 12th July 2012

[20] SECG SEC 2 Version 2.0 (2010). "SEC 2: Recommended Elliptic Curve Domain Parameters."

[21] Koblitz, N. (2010) "My Last 24 Years in Crypto: A Few Good Judgments and Many Bad Ones" (http://2010.eccworkshop.org/slides/Koblitz.pdf) Accessed: 24th June 2012

[22] Jansma, N., Arrendondo, B. (2004). "Performance Comparison of Elliptic Curve and RSA Digital Signatures." (http://nicj.net/files/performance_comparison_of_elliptic_curve_and_rsa_digital_signatures.pdf) Accessed 12th July 2012

[23] Antipa, A., Brown, D., Gallant, R., Lambert, R., Struik, R., Vanstone, S. (2005). "Accelerated Verification of ECDSA Signatures." (http://www.mathnet.or.kr/mathnet/preprint_file/cacr/2005/cacr2005-28.pdf) Accessed: 14th July 2012

ACKNOWLEDGEMENTS

The author would like to thank Dr. Richard Perry, Villanova Unversity, Professor of Electrical and Computer Engineering, for his support and mentorship in this work.

Kenneth A. Fischer, received a BS in Chemical Engineering from the University of Delaware and is pursuing an MS in Computer Engineering from Villanova University. Mr. Fischer has over 10 years of automation and controls experience in pharmaceutical, power generation, food and beverage, specialty chemical, and naval applications. He is currently employed with NSWCCD-SSES Code 955 providing engineering support to the INLS, ENFMC, LCS, and DDG1000 programs.

"The views expressed herein are the personal opinions of the author and are not necessarily the official views of the Department of Defense or any military department thereof."