Spring 3-2020

Mobile Identity, Credential, and Access Management Framework

Peggy Renee Camley

MOBILE IDENTITY, CREDENTIAL, AND ACCESS MANAGEMENT FRAMEWORK

A doctoral dissertation submitted to Dakota State University in partial fulfillment of the requirements for the degree of

Doctor of Science
in
Cyber Security

March, 2020

By
Peggy Renee Carnley

Dissertation Committee:
Dr. Pam Rowland - Chair
Dr. David Bishop
Dr. Sikha Bagui
Dr. Matt Miller

Dakota State University

Completion of Dissertation Final Defense Form

Student Name: Peggy Renee Carnley **Student ID:** 7462498

Title of Dissertation: Mobile Identity, Credential, and Access Management (ICAM)

Location of Final Defense: virtual

Date and Time of Final Defense: 3/23/2020 1PM CT

Date Time

Semester of Intended Graduation: Sp2020

The above-named student has:

□ Satisfactorily passed his/her final defense without revisions to the dissertation

□ Satisfactorily passed his/her final defense with revisions to the dissertation (page 2)

Revisions due:______________________

date

□ Had his/her final defense deferred and will be rescheduled (Please attach letter)

□ Not satisfactorily completed his/her dissertation defense (Please attach letter)

Pam Rowland Pam Rowland April 20, 2020

Signature (chairperson) Date

David Bishop David Bishop April 20, 2020

Signature (member) Date

Matt Miller Matt Miller April 20, 2020

Signature (member) Date

Sikha Bagui Sikha Bagui April 20, 2020

Signature (member) Date

Name of Student: Renee Carnley

Date of Final Defense: 3/23/2020

The following revisions are **required** for completion of dissertation (attach a separate sheet if necessary):

Date revisions required back to Dissertation Committee for approval: _n/a_______________
Date

The following revisions are *suggested* (but not required):

Pam Rowland ______________________________ April 20, 2020
Date

Name: Pam Rowland

Peggy Renee Carnley ______________________________ April 22, 2020
Name: Peggy Renee Carnley
Date

Wayne Pauli ______________________________ April 22, 2020
Graduate Program Coordinator
Name: Wayne Pauli
Date of Approval

Mark Hawkes ______________________________ April 22, 2020
Dean of Graduate Programs
Name: Mark Hawkes
Date of Approval

ACKNOWLEDGMENT

As an employee of the Department of Defense (DoD), I work for an Air Force (AF) organization that is responsible for providing capabilities to the warfighter more quickly than the regular DoD acquisition process can. The military is currently filled with millennials who grew up with a computer type advice in their hands practically from birth. They have an expectation that the DoD has the latest and greatest cutting-edge computer systems. And in some cases, they're correct but in others they are not. The biggest area where the DoD is lacking is the ability to communicate via mobile devices especially on the public internet. My dissertation research was inspired by the DoD's existing Public Key Infrastructure System and the move to a Federal identity card using personal identity verification (PIV) (U.S. Government, 2019) . Even though the Federal government was making moves to a more secure identity, it still did not account for use on the public internet and personal mobile devices. This is due to a lack of security especially that of a trusted digital identity. How can identities be trusted? How can we securely communicate on the public internet? These are the questions that inspired my dissertation topic.

I could not have worked full time and gone to school full time without the help of my family. I would have never made it through the core curriculum of the program without the assistance of my advisor, Dr. Wayne Pauli. My dissertation committee's support, encouragement, and guidance got me through the dark, lonely, frustrating days of dissertation challenges and failures. I owe much thanks to each and every one of them. Thank you!

ABSTRACT

Organizations today gather unprecedented quantities of data from their operations. This data is coming from transactions made by a person or from a connected system/application. From personal devices to industry including government, the internet has become the primary means of modern communication, further increasing the need for a method to track and secure these devices. Protecting the integrity of connected devices collecting data is critical to ensure the trustworthiness of the system. An organization must not only know the identity of the users on their networks and have the capability of tracing the actions performed by a user but they must trust the system providing them with this knowledge. This increase in the pace of usage of personal devices along with a lack of trust in the internet has driven demand for trusted digital identities. As the world becomes increasingly mobile with the number of smart phone users growing annually and the mobile web flourishing, it is critical to implement strong security on mobile devices. To manage the vast number of devices and feel confident that a machine's identity is verifiable, companies need to deploy digital credentialing systems with a strong root of trust. As passwords are not a secure method of authentication, mobile devices and other forms of IoT require a means of two-factor authentication that meets NIST standards. Traditionally, this has been done with Public Key Infrastructure (PKI) through the use of a smart card. Blockchain technologies combined with PKI can be utilized in such a way as to provide an identity and access management solution for the internet of things (IoT). Improvements to the security of Radio Frequency Identification (RFID) technology and various implementations of blockchain make viable options for managing the identity and access of IoT devices. When PKI first began over two decades ago, it required the use of a smart card with a set of credentials known as the personal identity verification (PIV) card. The PIV card (something you have) along with a personal identification number (PIN) (something you know) were used to implement two-factor authentication. Over time the use of the PIV cards has proven challenging as mobile devices lack the integrated smart card readers found in laptop and desktop computers. Near Field Communication (NFC) capability in most smart phones and mobile devices provides a mechanism to allow a PIV card to be read by a mobile device. In addition, the existing PKI system must be updated to meet the demands of a mobile focused internet. Blockchain technology is the key to modernizing PKI. Together, blockchain-based PKI and NFC will provide an IoT solution that will allow industry, government, and individuals a foundation of trust in the world wide web that is lacking today.

Keywords—Blockchain technology, Identity, Credential, and Access Management (ICAM), Internet of Things (IoT), Mobile Devices, Public Key Infrastructure (PKI), Near Field Communication (NFC), smart card.

DECLARATION

I hereby certify that this project constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Peggy Renée Carnley

Peggy Renee Carnley

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

1 INTRODUCTION

1.1 Background of the Problem

Despite almost all of our daily activities involving the internet, the internet has no trusted digital identity (Stokkink & Pouwelse, 2018). Currently, cyberspace is the one domain a person can literally be anything or anyone they choose. Social media accounts are self-sovereign meaning the user controls their credentials. The user creates a name, a birthdate, address, and career. Some are authentic. Some are fake. There is no way to validate a person is who they say they are in a self-sovereign system. In an ideal world, we would all trust each other. In the world as we know it, we are constantly bombarded with phishing emails, fake websites, and other forms of social engineering to scam us in one way or another. Across the globe more than two out of five consumers have been the victim to a fraudulent online incident. The United States (US) has the highest rates when compared to the rest of the world. Over fifty percent of businesses worldwide saw an increase in fraud over the last year, while over eighty percent of US businesses reported an increase (Experian, 2019). Individuals, businesses, and government have a strong desire and need for trusted digital identities (Stokkink & Pouwelse, 2018).

Most online identity verification is done through the use of a password. Passwords are a form of single factor authentication (SFA) which means only one method of identifying an individual's request for access is given. Passwords are used for logging into social media accounts, bank accounts, email accounts, and online shopping accounts. We are encouraged to use a strong password and make it unique for every online service. Most major industry such as governments, Universities, and financial institutions follow the National Institute of Standards and Technology (NIST) guidelines regarding secure passwords. These guidelines typically required strong passwords with a minimum length of fourteen characters that are a combination of letters, numbers and special characters. Despite password creation requirements, passwords remain weak and easily deducible because people continue to use predictable patterns and common words, such as names and birth dates (L. Bosnjak, 2018). There are many limitations on passwords or single-factor authentication such as:

- Password Capture –there are many methods of obtaining passwords.
- Keylogger malware – captures a user's keystrokes.
- Social engineering to manipulate a person's password through phishing emails, fake websites, social networks, phone calls, or other similar methods.
- Gaining access to unencrypted or weakly encrypted stored passwords.
- Passwords that are written down.
- Password Guessing – Using default passwords, dictionary words, and other probable passwords over and over to attempt authentication.
- Password Cracking – Using analysis and cryptographic password hashes to reveal the password.
- Password Resetting – Using interception or manipulation to reset a user's existing password (Ferraiolo, Cooper, Regenscheid, Scarfone, & Souppaya, 2016).

These are all reasonable threats that a hacker can exploit to obtain unauthorized access to a system or network and have that users' rights. Most hackers target administrators and others who have high level privileges in a system in order to be most effective with their attack because once an attacker has gained access into a system, the practice is similar every time:

- Establish continued access.
- Escalate or obtain administrator privileges.
- Network mapping.
- Port scanning to find open ports.
- Locate valued data – sometimes hard to determine and sometimes already known what to look for.
- Retrieve the data.
- Remain undetected for as long as possible (Verizon, 2019).

In 2018, approximately 330 million Twitter accounts had their passwords exposed in plain text, GOMO app had the information of approximately 50.5 million users exposed on a publicly accessible server via port 80 with no login requirement, and 1.5 million medical records were exposed from a password hacking breach (ENISA, 2019). According to studies conducted by Crowd Research Partners in 2018, the most common culprit of insider threat is accidental exposure by employees. They found the accidental breeches were due 56% to weak or reused passwords, no passwords accounted for 44%, and password sharing practices accounted for 44% for insider threats (Crowd Research Partners, 2019). Hacktivists, people who gain unauthorized access computer files or networks in order to further social or political ends, used SQL injection, unpatched system vulnerabilities, and password stealing as their main techniques to hack a website. LokiPWS, a webbot used as a password stealer, distribution increased by more than 300% and data breaches increased by 28% in 2018 (ENISA, 2019). Then there are other well-known password stealers such as Mimikatz (WatchGuard, 2018). Last year, one could find approximately 500,000 email accounts with passwords that were priced at US $90 in the Dark Web (ENISA, 2019).

The events of 2018 reveal that re-using the same password in various services is a serious security issue and should be avoided. It is also evident that passwords are not the optimal method of preventing entry. We have long been aware of password weakness. Back in 2012, LinkedIn lost 117 million passwords hashed with SHA-1. This stimulated a thorough analysis of how government and military employees use passwords compared to other organizations by WatchGuard Technologies, Inc. Government employees are considered to follow strong password practices typically greater than the rest of the population. For this study, the Security Analysts used the 55 million hashes obtained from the 117 million leaked and using the well-known dictionary realuniq.lst from CrackStation.net were able to crack 52% of them. Out of these they accounted for government addresses based on their emails identified by ".gov" or ".mil" and were able to crack 50% of them in under two days. The results determined that government and military users were only 2% better at picking strong passwords than non-government and military users (WatchGuard, 2018).

The cognizance of password weakness has become so popular that NIST made some controversial changes to their password guidelines with the release of Special Publication 800-63-3. They basically stated to try not to use passwords at all. NIST removed the periodic password change requirements as well as the password complexity requirement of mixing letters (upper & lower case), symbols, and numbers (Paul A. Grassi, 2017). Password-only authentication is not adequate for an application; therefore, it is recommended to use it in combination with other security mechanisms. The European Union Agency for Network and Information Security (ENISA) recommends enabling two factor-authentication whenever applicable as two factor-authentication can prevent account takeover (ENISA, 2019). Two-factor authentication can mitigate or lessen the exposure to lost or stolen passwords (Verizon, 2018). Public Key Infrastructure (PKI) has sustained itself over the past 20 years as the de facto standard for providing electronic trust via two-factor authentication (2FA) for desktop and other stationary devices (Ragjendran, 2017).

1.2 Statement of the problem

The pace of usage of personal devices has increased within industry, driving demand for trusted digital identities. As the world becomes increasingly mobile with the number of smart phone users growing annually and the mobile web flourishing, it is critical to implement strong security on mobile devices. As passwords and OTPs are not a secure method of authentication, mobile devices and other forms of IoT require a means of two-factor authentication that meet NIST standards. When PKI first began over two decades ago, it required the use of a smart card with a set of credentials known as the personal identity verification (PIV) card. The PIV card (something you have) along with a personal identification number (PIN) (something you know) were used to implement two-factor authentication. Over time the use of the PIV cards has proven challenging as mobile devices lack the integrated smart card readers found in laptop and desktop computers. In addition, the existing PKI system must be updated to meet the demands of a mobile focused internet.

1.3 Objectives of the project

The objectives of this project are to update the existing PKI framework to provide a secure means of authentication utilizing the PIV card when accessed from a mobile device. Currently, non-mobile electronics utilize Public Key Infrastructure as a method of 2FA. A USB smartcard reader allows those sitting in front of a laptop or desktop computer the capability of providing 2FA. Utilizing the near field communication (NFC) functionality built into most smart phones provides a method to develop authenticating with something you have and something you know from a mobile device meeting the requirements of 2FA. The NFC capabilities within smartphones allows an adequate replacement for USB smartcard readers. Improving on the existing centralized PKI framework by decentralizing it through blockchain technology, a new PKI framework will provide a secure method of authentication. This research will answer the question on how PKI can be improved to provide a trusted digital identity for mobile devices. The research will break down the overarching question to find answers to these questions:

1.3.1 Digital Identity

- Can an identity be verified as belonging to a real person or thing supplying the digital identity?
- Can a claimed identity be securely linked to a single, unique identity?
- Can the evidence supplied by an identity be validated as genuine (e.g. not counterfeit or spoofed)?
- Can the digital identity be validated as existing in the real world?

1.3.2 NFC

- Can the encryption size be reduced to fit on an NFC card?
- Can a smartphone with NFC enabled be a smart card reader?
- Is there a secure communication channel between the card and device?
- What drain is there on the battery life?
- Can the existing standard PKCS #11 which specifies an interface to cryptographic smart cards be used?
- Can the existing standard PKCS #12 which specifies a format that allows storing and transferring security-sensitive data such as private keys and certificates be used?

1.3.3 Blockchain

- Can PKI become a distributed system with blockchain?
- Can blockchain allow separate PKI systems to function as one?
- Can blockchain perform identity brokering?
- Can PKI systems from competitors be trusted?
- Can PKI provide a trusted identity for the internet?
- Can blockchain provide a trusted third-party certification authority?
- Can the size of the blockchain be hosted by enough peers?

CHAPTER 2

2 LITERATURE REVIEW

2.1 AUTHENTICATION

In order to ensure that a user on a network is genuine, there must be a method of identifying them. The data being transferred across a network as well as the systems on a network need to be identified as well. Origin, destination, time of transmission/reception, content and so on are data items requiring integrity preservation (Batten, 2012). The process of ensuring trust in a user, system, or data is called authentication. An authentication factor is a way of confirming the identity of a subject. The three authentication factors are “something you know”, “something you have”, and “something you are.” “something you know” is defined as an item that the subject has knowledge of, “something you have” is an item that is in the possession of the subject, and “something you are” is a biometric characteristic of the subject. Single-factor authentication is the act of identifying a user as authentic with one out of three of the authentication factors. Two-factor authentication is the act of identifying a user as authentic with two out of three of the authentication factors. Multi-factor authentication is the act of identifying a user as authentic using all three authentication factors. The more authentication factors used for authentication the stronger the security and trust in the subject’s identity (Ballad, Ballad, & Banks, 2011).

2.2 DIGITAL IDENTITY

The built-in anonymity of cyberspace makes identity one of the largest challenges that cybersecurity experts face (Rivera, Robledo, Larios, & Avalos, 2017). Managing and having trust in the identity of a user is desired knowledge by governments, industry, and individuals. In the technologically rooted social and business environments of the modern world, identity can be faked or impersonated. People want to know that the cute person they have been chatting with on social media or flirting with on dating sites is who they say they are. People want to trust that the email they just opened is really from their bank as it claims. Industry and governments want to know that the person they allowed onto their network or access to their websites is who they say they are. A digital identity sometimes referred to as an electronic identification (eID) is the cyberspace equivalent to a person or entities real life identity. An entity can be industry, government, or a thing. Basically, anything that connects to a network requires a digital identity. The Oxford Dictionary (Oxford, 2019) defines identity as:

- The fact of being who or what a person or thing is.
- The characteristics determining who or what a person or things is.
- [as modifier] (of an object) serving to establish who the holder, owner, or wearer is by bearing their name and often other details such as a signature or photograph.
- A close similarity or affinity
- (also identity operation)
- An element of a set which, if combined with another element by a specified binary operation, leaves that element unchanged.
- The equality of two expressions for all values of the quantities expressed by letters, or an equation expressing this, e.g. $(x+1)^2 = x^2 + 2x + 1$.

Therefore, a digital identity can be expressed as the fact of being who or what a person or thing is online. A digital identity is the characteristics determining who or what a person or thing is online. Or serving to establish who the holder, owner, or wearer is by bearing their name and often other details such as a signature or photograph online.

2.3 Cryptography

Cryptography is the science of securing data. There is a long history throughout the centuries of obscuring messages to prevent unwanted parties from reading them. The process of encoding a message in such a way that makes it unreadable to those not authorized is called encryption. Modern cryptography is founded on mathematical theory and computer science principles. The purpose of cryptography is to prove three fundamental things about a message: authentication, integrity, and confidentiality. Authentication proves that a message was sent by who claims to have sent it. Integrity proves that a message has not been altered in unauthorized ways. Confidentiality proves that a message is read by only those intended to read it (Batten, 2012).

There are two forms of cryptography: symmetric and asymmetric. Symmetric cryptography is a method of encryption where the sender and receiver share the same key. Symmetric cryptography utilizes simpler and faster encryption but requires securely exchanging the key which becomes a challenge. Asymmetric cryptography uses a public and private key to encrypt and decrypt data. The public key can be shared with anyone. The private key is known only to owner (Paar & Pelzl, 2010).

2.4 Public Key Cryptography

Public key cryptography is the same as asymmetric cryptography. The two terms can be used interchangeably. Public key cryptography began in 1976 with a paper publication by Whit Diffie and Martin Hellman describing a method of establishing a common key in a secure manner over an insecure channel (Paar & Pelzl, 2010).

Each entity (person or device) that uses public-key cryptography has a key pair that consists of a public key and a private key. Private keys are secret and known only to their owners. They are protected by a passphrase and can be stored on separate hardware cryptographic devices such as smart cards. Private keys are used for proving the identity of the entity. Public keys are made known openly and are distributed to all hosts with which the entity wants to securely communicate. The two keys are mathematically dependent but the private key cannot be derived from the public key. The data encrypted with the public key can only be decrypted with the private key and vice versa. Since the public key is shared freely a method to ensure the authenticity of the public key is created through a public key certificate. A public key certificate is an electronic document used to prove the ownership of a public key. This ensures trust that the public key belongs to the entity it is associated with. This is most commonly done using the X.509 standard. X.509 is a standard defining the format of the digital certificates used to validate ownership of a public key. This standard allows interoperability among numerous tools and applications among vendors (Adams & Lloyd, 2003).

2.4.1 Digital Certificate

Digital certificates are electronic credentials that binds a user, computer, or service’s identity to a public key by providing information about the subject of the certificate, the validity of the certificate, and applications and services that can use the certificate. Digital certificates can be used for authentication, encryption, and digital signing. Certificates issued in PKIs are structured to meet these objectives based on standards established by the Public-

Key Infrastructure (X.509) Working Group (PKIX) of the Internet Engineering Tasks Force (IETF) (Adams & Lloyd, 2003).

2.4.2 Root Certificate

The root certificate is the foundation of trust for PKI which is why it is also called the root of trust. In order for a certificate to be trusted it must originate from a trusted source. A certificate is signed by a root certificate issued by a program run under strict guidelines. Many well-known root certificates are distributed in operating systems such as Microsoft and Apple. A root certificate is invaluable because any certificate signed with its private key is automatically trusted (Mayes & Markantonakis, 2017).

2.4.3 X.509 Certificate

X.509 is a standard defining the most commonly used format of public key certificates through a series of Requests for Comments (RFC). X.509 for PKI follows RFC 5280. Each root CA will have a trusted root certificate that is used to sign each key pair generated for participates in the PKI system. The X.509 certificate for PIV authentication and its associated private key is defined in FIPS 201 and is used to authenticate the card and the cardholder. The PIV authentication private key and its corresponding certificate are accessed solely over the contact interface. The control rule for read access is set so that the certificate can always be read without any access control restrictions. This cryptographic function is protected with a PIN. Without a verified PIN submission there can be no private key operations using the PIV Authentication key. The X.509 certificate for digital signature and its associated private key support the use of digital signatures such as that utilized for signing documents.

2.4.4 Message Authentication Code (MAC)

A message authentication code (MAC) is used for data integrity. A MAC provides a cryptographic checksum that is computed by the sender and appended to the message. The receiver will compute a MAC and compare the MAC computed to the MAC attached to the message. If they are the same, data integrity can be trusted. If they differ, the data has been tampered with in some way (Paar & Pelzl, 2010). Although this technique employs symmetric encryption, it depends on a shared key known to both the sender and receiver. Therefore, public key cryptography can be used to compute this shared key (Adams & Lloyd, 2003).

2.4.5 Digital Signature

A digital signature is a means of identifying the sender of a digital message similar to the way a written signature on a paper or document proves authorship. Using public key cryptography, a sender, Bill, uses his private key to digitally sign a message. The receiver, Jill, uses Bill's public key to confirm the messages signature so that she can trust it came from Bill. If Bill wants to send a private message to Jill, Bill encrypts the message with Jill's public key. Only Jill can decrypt the message with her private key ensuring that only Jill reads the message. Bill digitally signs the message with his private key so that Jill can assure herself the message came from Bill. This is how public key cryptography allows a means of authenticating the identity of each other (Batten, 2012).

2.5 Smart Cards

A smart card is a piece of plastic with a microprocessor that can read and write data, contain a unique identifier, be used to process transactions, and add security. These cards are read by a smart card reader. The first plastic cards utilized a magnetic strip. The magnetic strip could be seen on every credit and debit card but their use has been replaced with a smart card that has a chip (Mayes & Markantonakis, 2017). Today's credit and debit cards have implemented a chip in addition to the magnetic strip. The smart card that contains a chip hosts a small microprocessor between two pieces of plastic that can store and process data via an integrated circuit (Chirico, 2014). As security features improved the magnetic strip was not seen to be secure as it can be copied therefore magnetic chip cards are no longer seen as smart cards as they do not meet smart card security requirements. To meet the modern definition of a smart card, the card must:

- Have a unique identifier,
- Be capable of use in an automated electronic transaction,
- Be used primarily as added security
- Not easily forgeable or able to be copied.
- Be capable to securely store data,
- Be capable of running a variety of security algorithms and functions (Oxford, 2019).

A smart card is tamper resistant and can be used for highly secure storage making it beneficial to house secret keys. A smart card implementing the proper security controls is read only once written to and therefore cannot be copied and then misused. The ability to have a unique identifier, contain encryption keys, and be tamper resistant makes smart cards the ideal option for multi-factor authentication (Hansmann, Nicklous, Schack, & Seliger, 2000).

2.5.1 History

Diners Club started the first credit card in the early 1950s. It was a plastic card that had the name of the cardholder printed on the front. With possession of this little plastic card, you could buy goods or services on credit at selected hotels or restaurants (Hansmann, Nicklous, Schack, & Seliger, 2000). Like all good ideas the plastic card could be forged or stolen so protection was implemented by visual features such as security printing and the signature panel. Eventually, these rudimentary security features became inadequate and they added a magnetic stripe on the back of the card which allowed for digital data to be stored. This increased handling costs for merchants and banks by making machine-readable card accessories necessary. Banks implemented the additional security feature of requiring a user to enter a pin at its automatic teller machines (ATMs) when presenting a debit card. Over time, it became apparent the magnetic-stripe technology had a major flaw. The data stored on the stripe can be read, deleted and rewritten at will by anyone with access to a suitable magnetic card reader/writer (Rankl & Effing, 2010). The idea of the smart card began with a patent filed by two German inventors, Jurgen Dethloff and Helmut Grotrupp in 1968. During the 1970's CII-Honeywell-Bull demonstrated the first prototype microprocessor-based smart card (Chirico, 2014). In 1974 Roland Moreno recorded his smart card patents that stimulated much growth with smart card development. Moreno's provided the semiconductor industry with the means to provide low cost integrated circuit supplies.

2.5.2 OpenCard Framework

The OpenCard Framework (OCF) is a standard Java framework for working with smart cards. The OCF is an open source framework found available on the internet at no cost. The OCF originated from the OpenCard Consortium through work done by IBM and Gemplus (CardContact, 2019). The architecture was created to provide a set of standards and commonality for card operating system providers, card terminal vendors, and card issuers (Chen, Java Card Technology for Smart Cards, 2000).

2.5.3 Java Card

The Java Card technology combines a subset of the Java programming language with a runtime environment optimized for smart cards and similar small-memory embedded devices. The Java Card platform is a smart card platform enabled with Java Card technology (also called a “Java Card”). This technology allows for multiple applications to run on a single card and provides facilities for secure interoperability of applications. Applications for the Java Card platform are called applets (Java Card Applications) (Chirico, 2014).

The Java Card Technology is a virtual machine on a smart card or other tamper-resistant security chips that utilizes the Java language. The Java Card Virtual Machine (JCVM) is split into two parts that consists of a part that runs off-card and the other that runs on-card. On the JCVM on-card portion is the Java Card bytecode interpreter. On the off-card portion is the Java Card converter (Chen, Java Card Technology for Smart Cards, 2000). The Java Card Technology utilizes the JCVM as an operating system that allows Java Card applets to run on a variety of smart cards. This is similar to how a Java applet runs on different computers (Chirico, 2014). A Java Card may be contact or contactless or both. Contact means the smart card uses a chip that houses a microprocessor while contactless means the smart card uses embedded integrated circuits that store and process data via near field communication (Chen, Java Card Technology for Smart Cards, 2000).

2.6 Personal Identity Verification (PIV)

The US Federal government created a Personal Identity Verification (PIV) credential to be used to access Federally controlled facilities and information systems. A PIV credential uses public key cryptography utilizing a public key certificate following multi-factor authentication following something you know, something you have, and something you are. The PIV credential is standardized to have specific information using technology w hich is interoperable (United States Government, 2019).

2.7 Public Key Infrastructure

Public Key Infrastructure (PKI) is the origin of a persistent security infrastructure whose services are implemented and delivered using public key cryptography (Adams & Lloyd, 2003). A PKI will provide the policies, roles, software, hardware, and procedures necessary to create, manage, distribute, use, store, and revoke digital certificates. The most important aspect of PKI is that it establishes the identity of people, devices, and services (Ballad, Ballad, & Banks, 2011).

2.7.1 Registration Authority (RA)

A registration authority (RA) is the standard name for the entity responsible for initial authentication of an individual. The RA's can be widely dispersed geographically to establish and confirm the identity of an individual as part of the initialization process. The identity is confirmed via physical presence and associated picture identification such as driver's license or passport. The RA is responsible for the unique association between certificate and person.

2.7.2 Credential Service Provider (CSP)

A credential service provider (CSP) establishes and maintains the enrollment records and binding authenticators of a digital identity. They were formed in order to meet the challenge of linking a digital identity to a single precise person or thing. A CSP can meet one of three identity assurance levels (IALs) (Grassi, et al., 2017).

- IAL1: Invalid or unverified digital identity. There does not have to be a link to a real person or thing.
- IAL2: There is some proof that the digital identity is real. The proof can be remote or physically-present identity proofing.
- IAL3: The digital identity is verified and validated by an authorized and trained CSP representative by the physical presence of the real person or thing (Grassi, et al., 2017).

The CSP responsibilities are sometimes delegated to a RA where they would maintain a close relationship in working together. The responsibilities of the CSP include collecting as much evidence as possible to validate an applicant and determine authenticity, validity, and accuracy. Identity validation consists of three primary acts: gathering suitable identity evidence, sanctioning the evidence is genuine and authentic, and corroborating that the data submitted as identity evidence is real, current and belongs to an actual object. The strength of evidence validity criteria is as follows:

Strength	Qualities of Identity Evidence
Unacceptable	No acceptable identity evidence provided.
Weak	• The issuing source of the evidence did not perform identity proofing. • The issuing process for the evidence means that it can reasonably be assumed to have been delivered into the possession of the applicant. • The evidence contains: • At least one reference number that uniquely identifies itself or the person to whom it relates, OR • The issued identity evidence contains a photograph or biometric template (of any modality) of the person to whom it relates.
Fair	• The issuing source of the evidence confirmed the claimed identity through an identity proofing process. • The issuing process for the evidence means that it can reasonably be assumed to have been delivered into the possession of the person to whom it relates. • The evidence: o Contains at least one reference number that uniquely identifies the person to whom it relates, OR o Contains a photograph or biometric template (any modality) of the person to whom it relates, OR o Can have ownership confirmed through KBV. • Where the evidence includes digital information, that information is protected using approved cryptographic or proprietary methods, or both, and those methods ensure the integrity of the information and enable the authenticity of the claimed issuing source to be confirmed. • Where the evidence includes physical security features, it requires proprietary knowledge to be able to reproduce it. • The issued evidence is unexpired.
Strong	• The issuing source of the evidence confirmed the claimed identity through written procedures designed to enable it to form a reasonable belief that it knows the real-life identity of the person. Such procedures are subject to recurring oversight by regulatory or publicly-accountable institutions. For example, the Customer Identification Program guidelines established in response to the USA PATRIOT Act of 2001 or the Red Flags Rule, under Section 114 of the Fair and Accurate Credit Transaction Act of 2003

	• (FACT Act). • The issuing process for the evidence ensured that it was delivered into the possession of the subject to whom it relates. • The issued evidence contains at least one reference number that uniquely identifies the person to whom it relates. • The full name on the issued evidence must be the name that the person was officially known by at the time of issuance. Not permitted are pseudonyms, aliases, an initial for surname, or initials for all given names. • The: o Issued evidence contains a photograph or biometric template (of any modality) of the person to whom it relates, OR o Applicant proves possession of an AAL2 authenticator, or equivalent, bound to an IAL2 identity, at a minimum. • Where the issued evidence includes digital information, that information is protected using approved cryptographic or proprietary methods, or both, and those methods ensure the integrity of the information and enable the authenticity of the claimed issuing source to be confirmed. • Where the issued evidence contains physical security features, it requires proprietary knowledge and proprietary technologies to be able to reproduce it. • The evidence is unexpired.
Superior	• The issuing source of the evidence confirmed the claimed identity by following written procedures designed to enable it to have high confidence that the source knows the real-life identity of the subject. Such procedures are subject to recurring oversight by regulatory or publicly accountable institutions. • The issuing source visually identified the applicant and performed further checks to confirm the existence of that person. • The issuing process for the evidence ensured that it was delivered into the possession of the person to whom it relates. • The evidence contains at least one reference number that uniquely identifies the person to whom it relates. • The full name on the evidence must be the name that the person was officially known by at the time of issuance. Not permitted are pseudonyms, aliases, an initial for surname, or initials for all given names. • The evidence contains a photograph of the person to whom it relates. • The evidence contains a biometric template (of any modality) of the person to whom it relates.

	• The evidence includes digital information, the information is protected using approved cryptographic or proprietary methods, or both, and those methods ensure the integrity of the information and enable the authenticity of the issuing source to be confirmed. • The evidence includes physical security features that require proprietary knowledge and proprietary technologies to be able to reproduce it. • The evidence is unexpired.

Table 1: Strengths of Identity Evidence (Grassi, et al., 2017)

Homeland Security Presidential Directive-12 (HSPD-12) mandated a common identification standard to be adopted governing the interoperable use of identity credentials. Federal Information Processing Standard 201 (FIPS201), Personal Identity Verification (PIV) of Federal Employees and Contractors, was developed to establish standards for identity credentials.

2.7.3 Certification Authority (CA)

A certification authority (CA) is an entity that certifies an identity with a public key. Basically, the CA is the method of generating the digital identity through providing a certificate. Certification is a binding that occurs in the form of a signed data structure called a public-key certificate. A CA is an authority on the process of certification. The issuing CA digitally signs certificates ensuring integrity therefore a CA must be trusted by both the issuer of the certificate and the owner of the certificate (Ballad, Ballad, & Banks, 2011). A CSP must provide security management services for key generation and storage.

2.7.4 Validation Authority (VA)

The Validation Authority (VA) is the authentication system within PKI. It verifies the validity of a digital certificate by following the requirements of the X.509 standard. A VA manages the certificate revocation list (CRL) issued by the CAs and provides online certificate status protocol (OSCP) function. Some VAs may also provide access control and authorization services (Ballad, Ballad, & Banks, 2011).

2.7.5 Certificate Revocation

A certificate may become revoked for a number of reasons. A person could change their name like some do upon marriage. Perhaps the CA issued an improper certificate or the private key has become compromised. Whatever the reason, when it is necessary to invalidate a certificate it must be revoked. This is usually done with a certificate revocation list (CRL). The VA would check the CRL as part of its process before authenticating. The issue with the current system is that the CRL is generated and published periodically potentially allowing an invalid certificate to be authorized when it should not (Adams & Lloyd, 2003).

2.7.6 PKI Architecture Overview

The authentication and authorization validation process of MFA requires strong trust that must have meaning and be quantifiable. Since trust is more of a social construct, giving it meaning and finding measurements within an electronic system proves challenging. PKI's reliability on the correct usage of a public/private key pairs depends upon there being a chain of trust among certificate authorities (CA). A public key certificate is issued as the public component of these key pairs and are often associated with common access cards (CACs). These CA's are the third-party servers providing the certification path to authentication. Path validation and path construction are essential to the proper management of trust within PKI (Rahoof, 2017).

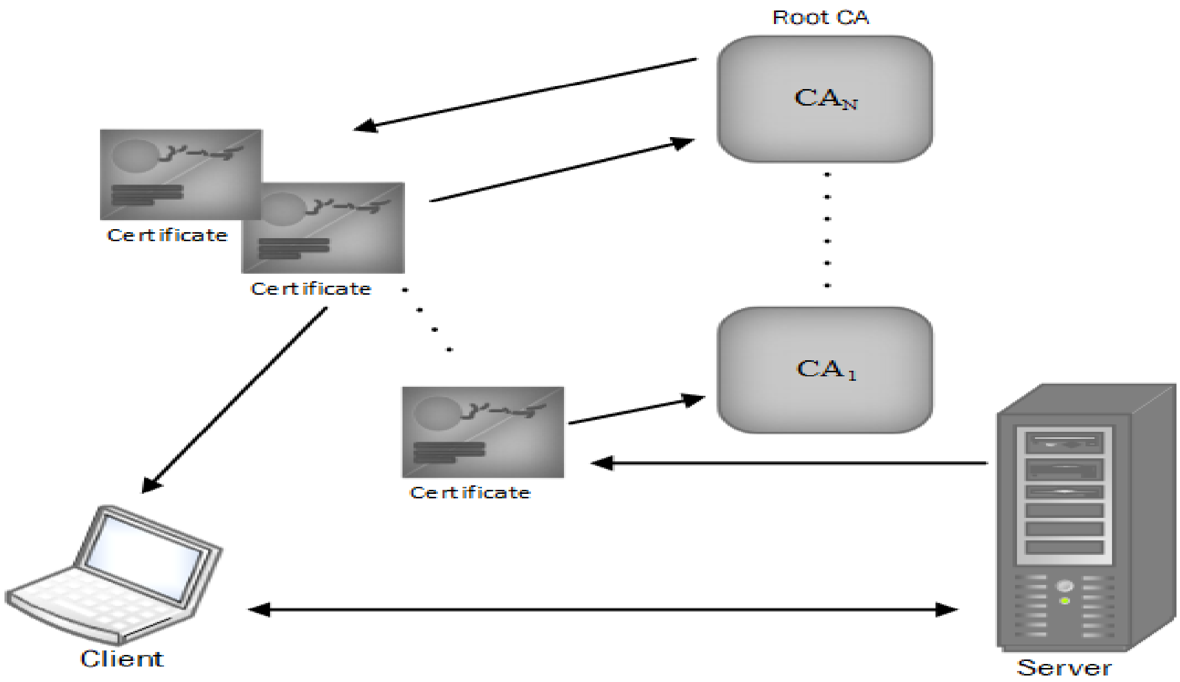


Figure 1: Service Authentication

Before allowing a user access to a system or network, the authenticity of the public key presented must be assured. A validation authority (VA) is a trusted server providing a means of verifying the validity of a digital certificate. The trusting entity sends a certificate to the VA server that validates the public key certificate (PKC). This process typically occurs on the client side and requires the use of software that can support the protocols and algorithms. An organization's use of a VA in addition to establishing policies provide confidence in who is and who is not allowed on their systems (Ma, 2011).

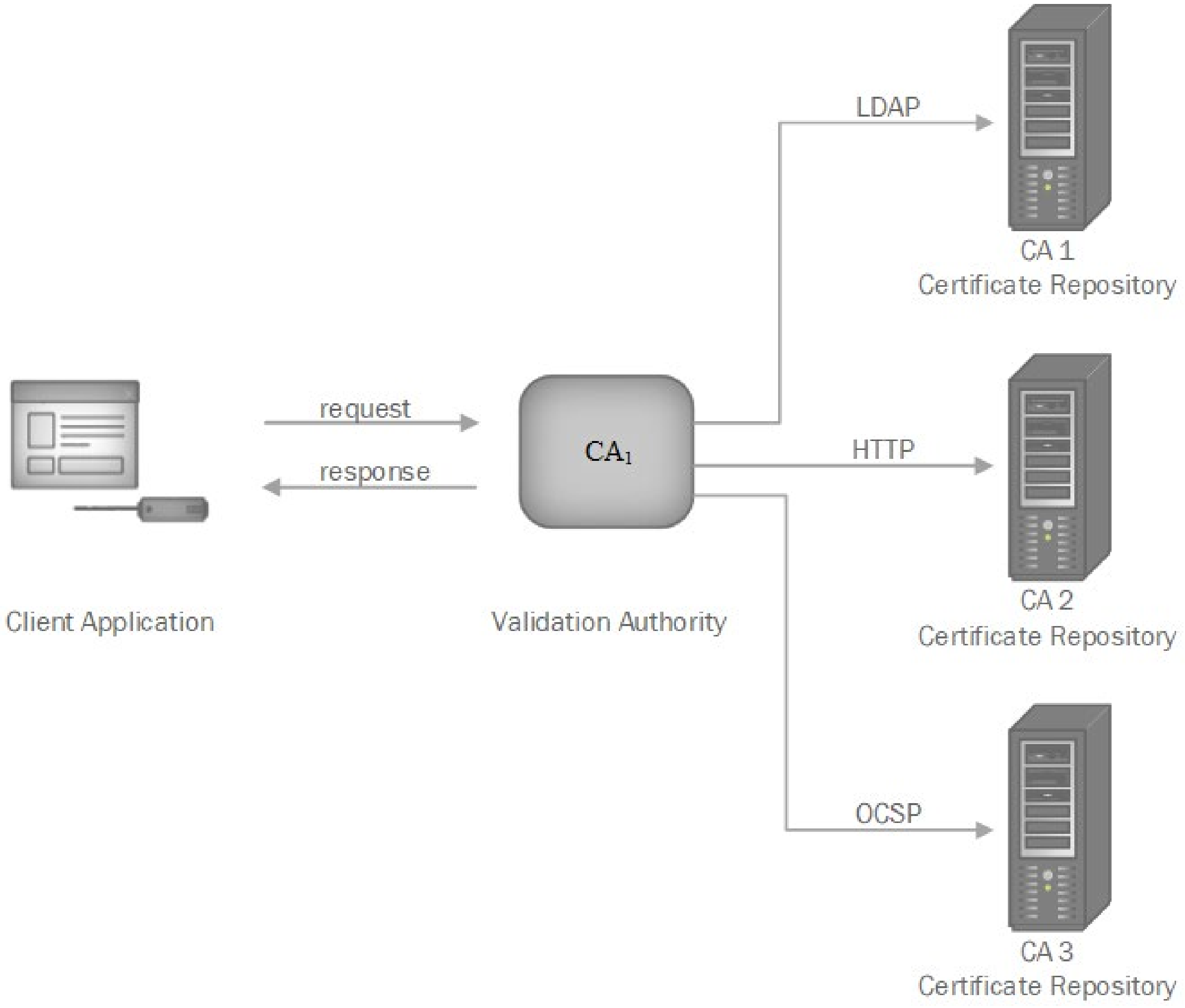


Figure 2: Validation Authority Schematic Diagram

Path construction is the process of building a CA certification path. Constructing this path is generally more difficult than path validation (Rahoof, 2017). These paths are defined and based upon the X.509 PKI standard [8]; further details can be gleamed from examination of that standard. Path construction typically begins with a root CA that generates its own self-signed certificate. Once the root CA is established it binds the identity and public key of an intermediate CA. The intermediate CA launches the next CA in the path and so on and so on until the path reaches the end-user who seeks a certificate (Ma, 2011).

2.8 Blockchain

In as much as email ushered in a new way of sending letters to businesses and people, blockchain has provided a new way of storing transactions and other kinds of data (a database). Databases have been used to centrally store data for decades. Blockchain provides a decentralized method of storing data. Data is entered into a block and then added to a chain of blocks. This structure forms a blockchain. Each block in the chain is represented by a cryptographic hash that contains its own hash as well as the hash for the previous block (Gates, 2017). Blockchain is defined as "a peer-to-peer, distributed ledger that is cryptographically-secure, append-only, immutable, and updatable only via consensus or agreement among peers" (Bashir, 2018). The foundation of the blockchain began with a paper published by Stefan Konst in 2000 that provided instructions for implementing cryptographically secured chains. Blockchain offers several key benefits such as transparency, trust, cost reduction, transaction improvements, and security (Gates, 2017).

Blockchain became popular due to the fact that it was the technology behind Bitcoin. Bitcoins are a type of electronic cash used as a digital currency on the internet. Bitcoin spelled with an uppercase 'B' references the cryptocurrency payment network, protocols, and blockchain. When spelled with a lowercase 'b', bitcoin refers to the units of bitcoin. For example, Sally is sending Bob 1.5 bitcoins (Ethereum, Bitcoin, Blockchain, and Cryptocurrencies Resources, 2018). In order to explain how blockchains function bitcoins will be used.

2.8.1 Creating a blockchain

A block is an assortment of transactions which are arranged logically. A transaction is the transference of digital currency from a sender's account to a receiver's account. A block can consist of more than one transaction (Bashir, 2018). Block 0, or the genesis block, is the first block on the blockchain. The genesis block within Bitcoin was hardcoded at the time of creation with the message "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks." Each block contains a hash of the previous block's message hash linking each block together in a chain and providing additional security that the previous block's transaction has not been tampered with. The Bitcoin blockchain uses the SHA-256 algorithm as it's hash. The SHA-256 algorithm generates a unique, fixed-size 256-bit hash (Gates, 2017).

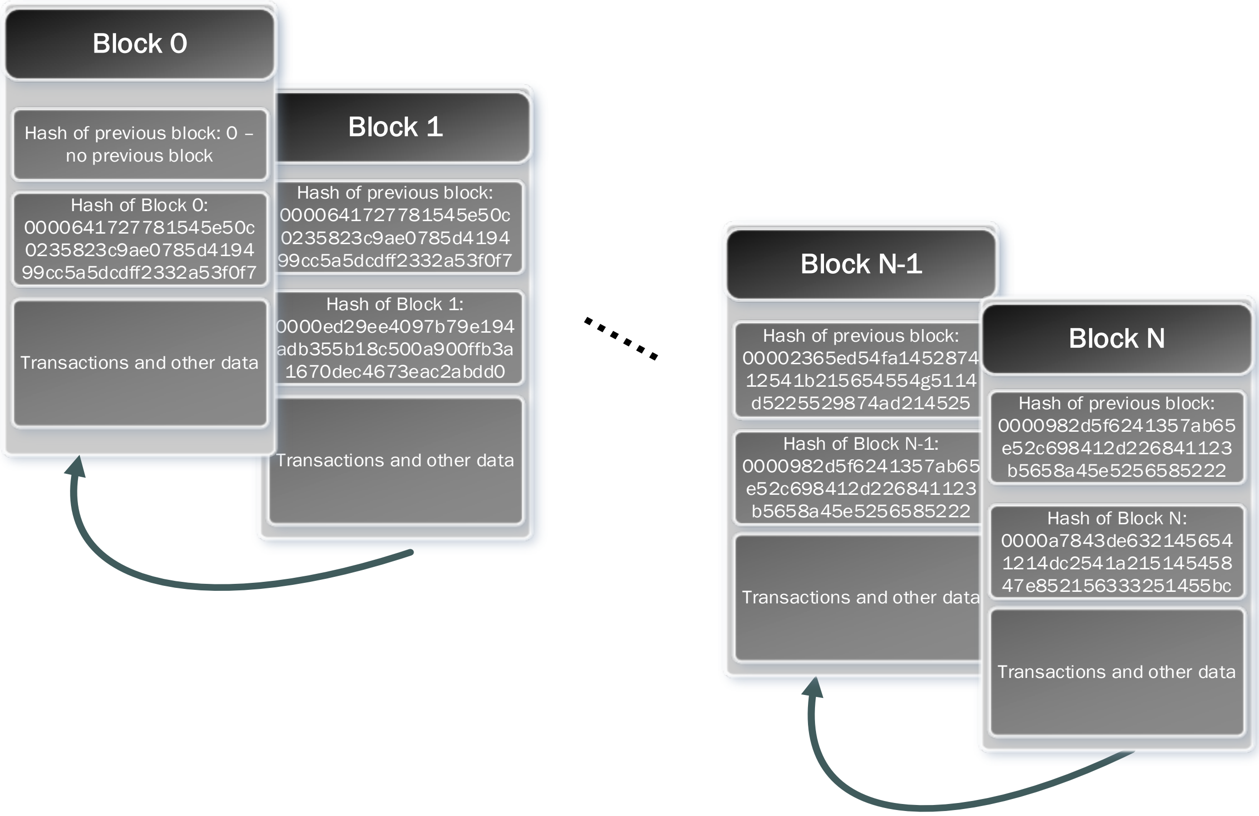

Block 0
Hash of previous block: 0 – no previous block
Hash of Block 0: 000064172778154 5e50c0235823c9ae0785d419499cc5a5dcdff2332a53f0f7
Transactions and other data
Block 1
Hash of previous block: 0000641727781545e50c0235823c9ae0785d419499cc5a5dcdff2332a53f0f7
Hash of Block 1: 0000ed29ee4097b79e194adb355b18c500a900ffb3a1670dec4673eac2abdd0
Transactions and other data
Block N-1
Hash of previous block: 00002365ed54fa145287412541b215654554g5114d5225529874ad214525
Hash of Block N-1: 0000982d5f6241357ab65e52c698412d226841123b5658a45e5256585222
Transactions and other data
Block N
Hash of previous block: 0000982d5f6241357ab65e52c698412d226841123b5658a45e5256585222
Hash of Block N: 0000a7843de6321456541214dc2541a21514545847e852156333251455bc
Transactions and other data

Figure 3: The Generic Structure of a blockchain

2.8.2 Centralization vs Decentralization vs Distributed

Centralization is the concentration of control of an action or process under a single authority. Basically, this means that a data transaction must be verified by a trusted third party. For example, when shopper uses a credit card for a purchase, the store is trusting payment for the goods provided in another party other than the shopper. This centralization of credit gives financial institutions prominent authority because these institutions as trusted third parties become the final decision makers on a shopper's creditworthiness. It is extremely challenging to correct inaccuracies of information stored in a centralized database (D. Richard Kuhn, 2001).

Decentralization is a fundamental aspect of blockchain technology. The trusted third party relied upon in a centralized system such as PKI is unnecessary. Instead, blockchain utilizes a consensus mechanism to validate transactions (Bashir, 2018). Every participant in the system makes its own decisions. Due to the viewability and validation by anyone, blockchain provides transparency and trust. It allows a system of trust between parties without requiring an intermediary. Negotiations can be performed between individuals for practically anything; property, money, digital files, etc. (Gates, 2017). The responsibility and control of the correctness of the data stored in a blockchain-based system falls upon the individuals involved in the transactions. This means the network must agree and decide. Within Bitcoin, changes must be agreed to by a certain majority of the network. This may be 50% but could be as high as 70 to 80% of the network. There is a risk of a 51% attack on a blockchain network if a malicious user(s) controls more than 50% of the computers on the network. There is potential for collaboration among users on a blockchain network to influence current or future development through a 51% attack (Bashir, 2018).

A distributed system is where all of the parties work together as a single coherent system. It has qualities of centralization and decentralization. There is still a central authority that has some control over the other parties in governing processes yet the other parties can make many of their own decisions and work autonomously. Distribution improves availability, reliability, fault tolerance, performance, and scalability (Bashir, 2018).

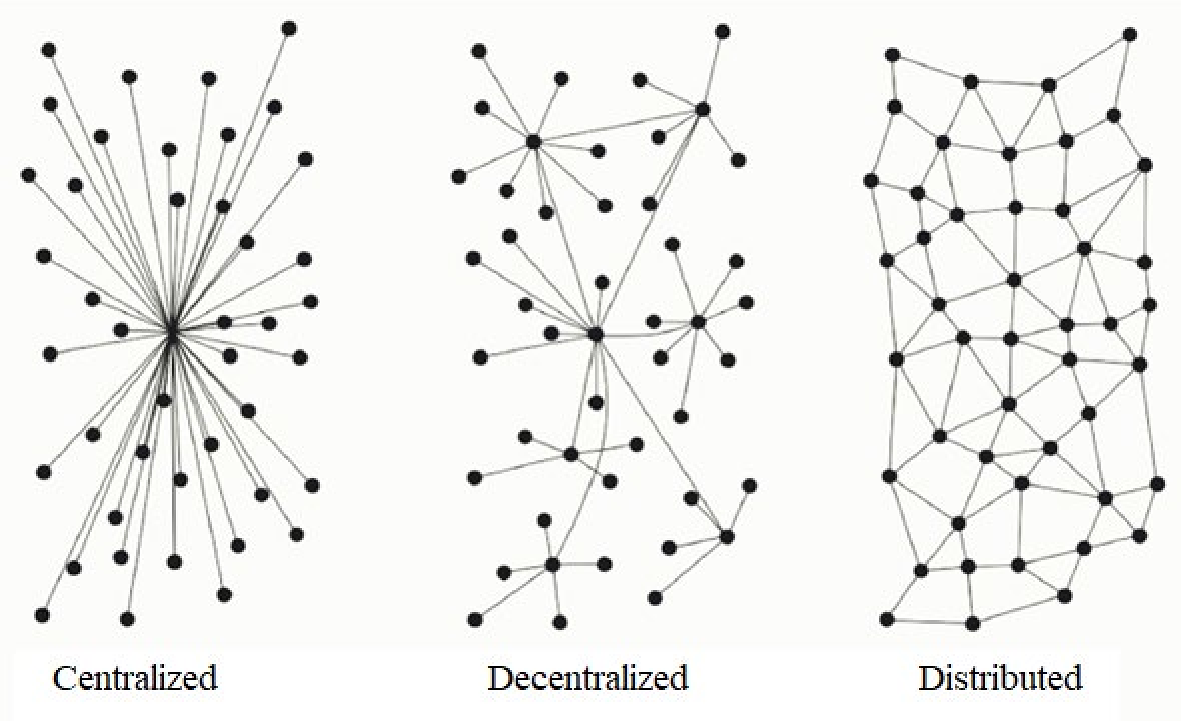


Figure 4: Graphical Representation of Centralized, Decentralized, and Distributed

2.8.3 Public Blockchains

A public blockchain is available to anyone who wants to participate in the blockchain. There is no one exclusive power over the blockchain. Everything is open to all of those using the blockchain (White, 2018).

2.8.4 Private Blockchains

A private blockchain has an organization or consortium who controls the permission to write data onto the blockchain. There are a set of rules that govern the blockchain. Transactions are not allowed if they violate the rules and regulations (White, 2018).

2.8.5 Permissioned Blockchains

A permissioned blockchain only allows those authorized to have a role within the blockchain. They are the middle ground between public and private blockchains. The verification is performed by predetermined nodes. They use cryptography to give permissions to those using the blockchain (White, 2018).

2.9 Blockchain-based PKI

The PKI framework as it currently exists has vulnerabilities. Reporting unauthorized certificates is a time consuming and labor-intensive effort that leaves a CA open to a man-in-the-middle (MITM) attack. If the CA's are not operating correctly, the introduction of encryption has no value. Blockchain-based PKI techniques provides methods to secure the CA vulnerabilities immediately in real time (Matsumoto S. &., 2017). Blockchain is a data structure that utilizes public-key cryptography in the creation of tamper-proof digital signatures that may be shared among parties. Basically, they are online ledgers that provide decentralized and transparent data sharing (Kshetri). Blockchains are the technology behind bitcoins that have been successfully used in E-commerce. Blockchains rely on cryptographic proof instead of trust negating the use of a trusted third-party and allowing anonymity in online transactions (Yakubov).

In order to affectively implement blockchain within two-factor authentication, establishing trust would be necessary to instantiate security measures against interference, breach, and eavesdropping (Robey, 2017). A considerable vulnerability to PKI applications and platforms is their dependence on a centralized cloud. The PKI in its current form is centralized relying on trusted third-parties. Decentralizing and incorporating blockchains provides the means of overcoming several of the problems linked with the centralized cloud approach. Provenance and other startups are using blockchain to promote trust in product transactions from source to the customer (Kshetri). Blockchains can cryptographically sign transactions and verify the originator's cryptographic signature to guarantee a message's origin (Kshetri). Blockchains also provide secure traceability of certifications and other relevant data in supply chains. Blockchain's public availability ensures transactions can be linked to identify vulnerable mobile devices (Robey, 2017). Suitable for registering time, location, price, parties, and data as they move through the supply chain, blockchain based MFA systems will help strengthen mobile device security (Kshetri).

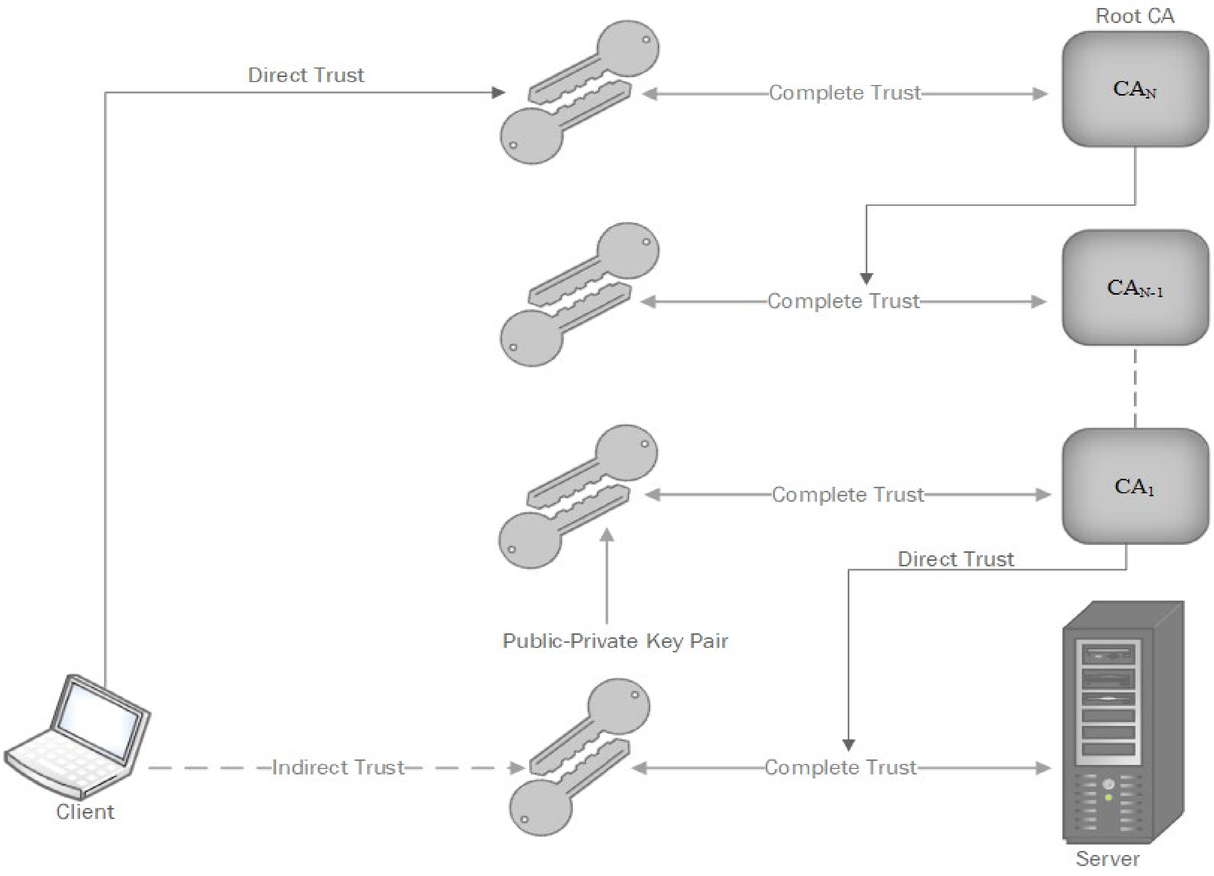


Figure 5: PKI Trust Structure Path Construction

2.10 RFID

Radio Frequency Identification (RFID) technology is a low powered system that transmits wirelessly. The tags are generally passive devices meaning they have no power source while the readers are a more complicated computing device with sufficient power, memory, communication interfaces, and its own clock. RFID began as a way to replace barcodes but blossomed to include a wide variety of applications such as toll transponders, passports, credit cards, access badges, pet tracking devices, pharmaceuticals, clothes, library books, and much more (Gritzalis). This has led to RFID becoming the preferred method of providing wireless communication between IoT devices. This has increased the need to commission a secure method of authentication that involves MFA. Electronic Product Code Class 1 Generation 2 (EPC C1 G2) standard is the most widespread RFID standard projected to provide secure authentication for RFID users (Kshetri).

Lightweight authentication protocols incorporating simple cryptographic functions have been developed to provide an authentication method (Ma, 2011). RFID systems incorporate RFID tags and RFID readers. To utilize a PKI MFA, each tag needs its own public/private key pair with a public key certificate. The primary purpose of RFID tags is to allow identification by readers. A reader that has become the possession of a malicious user (i.e. stolen, lost, compromised) can be used to identify and track tags. Therefore, it is more critical to have trust in the reader than the tag (Gritzalis). One possible way of providing trust despite the risks associated with the reader is via near field communication.

2.11 NFC

Near field communication (NFC) is a more simplistic implementation of the RFID technology. NFC involves two wireless devices operating via short range frequencies within 5-10 cm. There are two modes: active and passive. An active mode device starts the communication. These devices are referred to as the initiator. The initiator generates its own power and sends information by amplitude shift keying. Within passive mode the device is referred to as the target and uses the radio frequency (RF) field from the initiator as power for its communication (Matsumoto S. R.). Within NFC, the lines between reader and tag are blurred eliminating the primary issues of RFID PKI usage. For example, NFC-enabled smartphones can switch between being a reader and being a tag. While sending the smartphone acts as the tag and while receiving it acts as the reader.

2.11.1 Gemalto

Gemalto has been making contact smart cards for the DoD and other agencies for years. Gemalto uses the java card for their smart cards. Java card is an industry standard technology platform developed by Sun Microsystems (now Oracle) to enable Java-based applications that run on smart cards. These Java-based applications are called applets. Java card helps developers build, test and deploy smart card-based applications quickly and efficiently with an object-oriented programming model and off the shelf development tools. Since Java Card 3.0, the card has been extended to support a Web application model with servlets running on the card, and TCP/IP as basic protocol. The Virtual machine and Runtime Environment have been upgraded as well to support multithreading and hierarchical class loaders. The Java Card platform can run on contact and contactless devices since it runs on secure elements that power the Card Emulation mode in NFC. The java card operating system they use is called JLEP. This information was obtained from email and phone conversations with Gemalto employees. It was challenging to get more information. Gemalto clammed up when it was discovered I was a dissertation student with no budget to buy anything from them. They ghosted my emails and phones calls after revealing my student status.

2.11.2 NXP Semiconductors

NXP Semiconductors makes a smart card using Java Card Open Platform (JCOP) operating system. The JCOP operating system has a Java Card Virtual Machine (JCVM) and lots of information can be found on it compared to JLEP. NXP's smart cards are more popular in Europe than the United States. The most popular smart card using NXP's JCOP is MiFare (NXP, 2019).

2.11.3 Google Titan

Using special firmware created by Google, Titan is a security key that provides a 2FA solution for logging into accounts on desktop and mobile devices. They have a variety of keys built on FIDO open standards so they can be used with many apps and services. Titan has the ability to operate via NFC, USB, and Bluetooth (Google, 2019).

2.11.4 Yubico

Yubico has incorporated NFC into their YubiKey and given it smartcard capabilities. They perform RSA or ECC signing and decryption via the private key stored on the Yubikey following PKCS#11 standards. They come in a variety of styles and work with both Android and Apple smartphones.

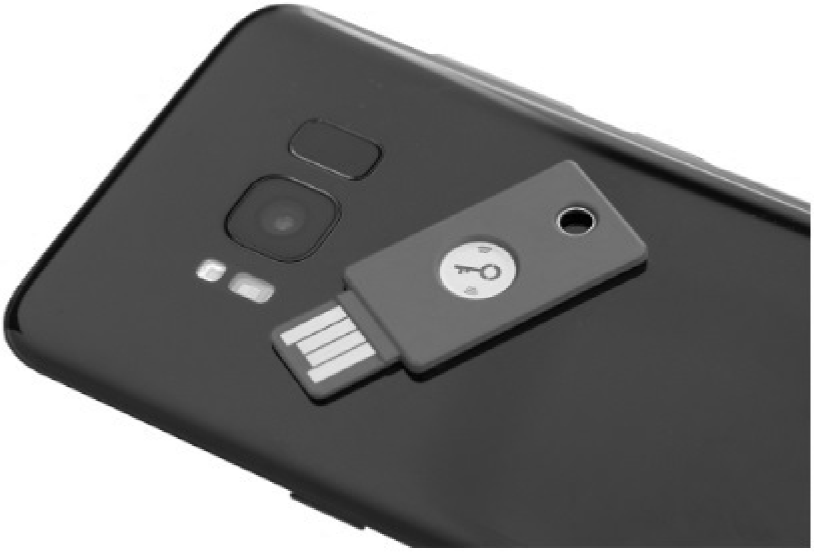


Figure 6: Yubico's YubiKey with NFC

Just tap the YubiKey to the back of a NFC enabled smartphone and the phone acts as the PIV smartcard reader. The Yubikey is being used for authentication for mobile devices, apps, and websites. This provides better security than a one-time passcode and is much more convenient as well.

Figure 7: YubiKey Touch and Go Capabilities with Mobile App

2.11.5 PKI utilizing NFC

A cryptographic challenge response protocol based on PKC and PKI has been developed for protecting NFC tags from attacks. This proposed framework consists of using symmetric cryptography (Matsumoto S. &., 2017).

To enhance security, a secure protocol is presented with the NFC chip. The intent is to add an extra layer of security within NFC-enabled systems by incorporating a data/information processing unit. The security protocol includes a processing stack. This stack consists of handshaking, scheme, certificate verification, signature verification, and an alert mechanism.

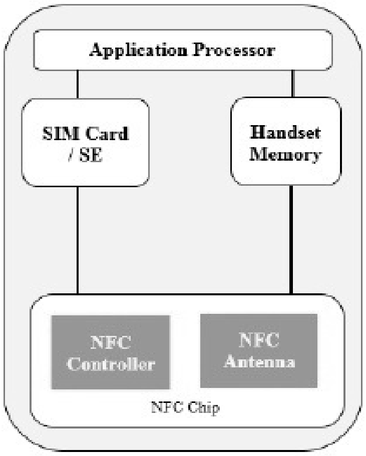


Figure 8: A typical NFC architecture

The process begins by the handshaking scheme asking for a certificate. If the certificate and the signature match, data is stored for further processing. If at any time there is an error i.e. the certificate and the signature cannot be verified, the data is discarded from the system and alert messages are transmitted.

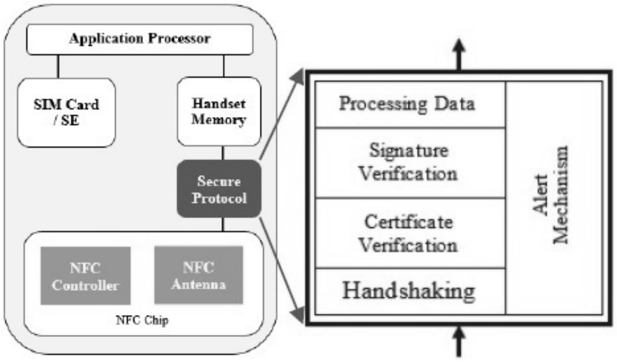


Figure 9: The proposed NFC system

The proposed NFC system was tested and found to adequately protect against tag manipulation and data insertion. There are minor increases to the processing time the larger the signature size used. Thus, to save processing time use a smaller signature (Matsumoto S. &., 2017). Robust authentication is a requirement for MFA. Most leading services provide strong authentication through symmetric cryptography such as Advanced Encryption Standard (AES) or asymmetric cryptography such as Elliptic Curve Cryptography (ECC). Asymmetric solutions such as ECC are complex to implement and often inefficient. Researchers have discovered a secure NFC with a flexible architecture call Cryptographic Protected Tags (CRYPTA). The latter works passively using a low-area design that utilizes as few resources as possible. This passive implementation provides a secure NFC/RFID that may be used in NFC-enabled smart phones (Plos). Authenticity and confidentiality are used to provide end-to-end communication between a client and a server; therefore, a server is required to authenticate its identity to a client and vice-versa (Matsumoto S. R.). The CRYPTA tag provides strength in authentication through an analog antenna that demodulates and modulates the data, extracts the power supply, and provides a stable clock and reset signal (Plos). The framing logic is the portion that handles the time critical low-level commands. The cryptographic operations are processed within the crypto unit and is accessed by the microcontroller via micro-code patterns. The tag's power is supplied from the RF field and provides the interface for the data, clock, and reset. Smart cards often use 32-b controllers that have high area and power consumption, CRYPTA uses an 8-b microcontroller with a low chip area and low power consumption making CRPTA more efficient than anything currently in use (Plos). The only downside to CRYPTA is that it is a proposed real-world RFID system that includes all hardware components needed for a practical chip fabrication. While the scientists who designed the system invented a prototype that tested well in the lab (Plos), more testing will be needed to prove the viability of it as an IoT solution.

2.12 Zero Trust Architecture

A Zero Trust Architecture (ZTA) provides no implicit trust to a system based on their physical or network location. Access to data resources is only allowed when the resource is required, and authentication to both users and devices is performed before the connection is established. The ZTA is a response to enterprise network trends that include remote users and cloud-based assets that are not located within an enterprise-owned network boundary. The ZTA focuses on protecting resources, not network sectors, as network location is no longer seen as the prime component to the security stance of the resource.

CHAPTER 3

3 SYSTEM DESIGN

This chapter provides a formal characterization of the Identity, Credential, and Access Management (ICAM) Framework, describes each component of the framework and the detailed steps in its construction. A framework for identity cannot be formed without a solid understanding of identity within IoT. NIST Special Publication 800-63-3 Digital Identity Guidelines were followed. NIST Publications were followed and attributed as necessary in describing the framework. As demonstrated in Chapter 2, industry and government want to know who is on their networks and what they are doing while on them. The PKI systems that worked so well in the 20th century do not meet the mobile demands of the 21st century. In today's dynamic world, PKI must become decentralized and identity must be digitized in such a manner that an individual feel assured they have control over their personal data. As identity and security were an afterthought of the internet, safely maneuvering cyberspace relies on frameworks that will provide the needed infrastructure to protect individuals, industries, and government's data. The ICAM Framework although theoretical provides updates to PKI that establishes a digital identity for our mobile world that no other framework has given.

3.1 RESEARCH APPROACH

This study utilized the design science research (DSR) method to develop the ICAM theoretical framework. The DSR method is a creative research paradigm that concentrates on the development and performance of artifacts with the explicit intention of improving the functional performance of the artifact. Compared to other research methodologies, DSR is more pragmatic and is a quest for understanding and improvement. The DSR method uses design as a research method or technique targeting the improvement or innovation of information and communication technology (ICT) artifacts (Vaishnavi & Keuchler, Jr, 2015).

Design science research is the design and investigation of artifacts that are goal-oriented. The technical research goal defines the problems and designs an artifact so that it contributes to solving these problems. The engineering cycle is the design problem solving process that consists of tasks: problem investigation, treatment design, treatment validation, treatment implementation, and implementation evaluation. (Wieringa, 2014).

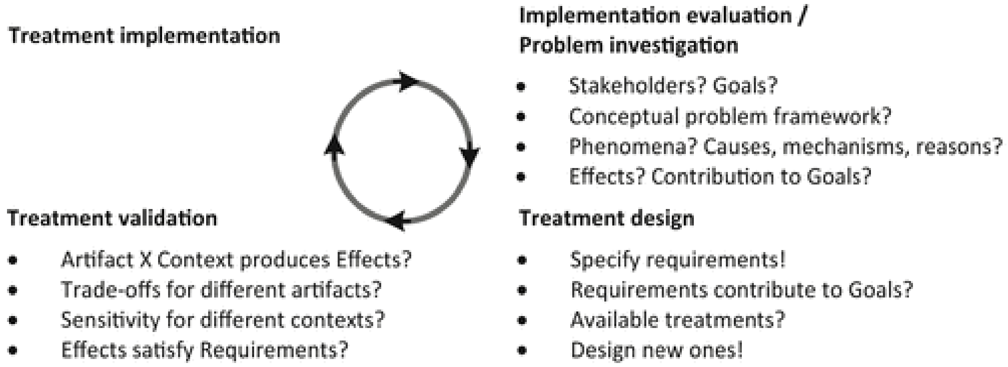

Treatment implementation
Implementation evaluation / Problem investigation
Stakeholders? Goals?
Conceptual problem framework?
Phenomena? Causes, mechanisms, reasons?
Effects? Contribution to Goals?
Treatment validation
Artifact X Context produces Effects?
Trade-offs for different artifacts?
Sensitivity for different contexts?
Effects satisfy Requirements?
Treatment design
Specify requirements!
Requirements contribute to Goals?
Available treatments?
Design new ones!

Figure 10: DSR Engineering Cycle (Wieringa, 2014)

Due to the problem-solving or performance improving aspects of DSR, it is sometimes called improvement research. Emphasis is placed on contributing to the body of knowledge surrounding the problem set with the goal of providing potential solutions to the areas of greatest concern (Vaishnavi & Keuchler, Jr, 2015). Therefore, this research is focused on the development of an artifact that facilitates solutions of critical interest within the research community or society or both in regards to having a trusted digital identity that will work in our mobile world.

3.1.1 Treatment

The DSR method sidesteps calling an artifact a solution in order to prevent impartiality to an artifacts result. Instead artifacts are considered treatments. Treatment suggests that the artifact is interacting with a problem and allows the researcher to gage the effectiveness is “treating” the problem. Treatments are designed and documented in a specification. A design is a decision about what to do and a specification is a documentation of that decision (Wieringa, 2014).

3.1.2 Implementation

An implementation can mean different things to different people depending upon context. For the purposes of this research, an implementation takes the standard DSR definition and defines implementation as the application of the treatment to the original problem set. Implementation begins to appear after the first few iterations of the engineering cycle. The treatment is designed then evaluated returning to the design phase as often as necessary until the evaluation and validation phases show satisfactory results (Wieringa, 2014).

3.1.3 Validation and Evaluation

Treatment validation requires that it contribute to the technical research goals if implemented. As shown in Figure 6: DSR Engineering Cycle (Wieringa, 2014) validation comes before implementation and involves a thorough examination of the effects of the treatment upon the problem set. Based on the validation process a design theory is developed which envisions the outcome if the treatment were implemented (Wieringa, 2014).

Evaluation is a constant progression throughout the engineering cycle that consist of multiple micro-evaluations at every design aspect choice. Evaluation is performed after implementation. Implementation within DSR does not consist of using it in the real world. The goal of evaluation is to investigate how effective a treatment is through simulation, modeling, or theorizing (Vaishnavi & Keuchler, Jr, 2015). The ICAM framework was created through theorizing from simulations to create an artifact that can used for future work.

3.2 LIMITATIONS

The ICAM Framework will decentralize the existing centralized PKI system. Within the context of this research, the validation and evaluation of the decentralization occurred in limited locations. The framework is theoretically designed to successfully function in an unlimited number of sites. Due to the robust nature and substantiation of decentralization within blockchain technology in real world implementations, the limitation in number should not be a problem. Consideration to validating and evaluating the framework in larger numbers is mentioned in future work.

3.3 THE ICAM COMPONENTS

The ICAM system can be broken down into three major components: the frontend subsystem, the issuance and management subsystem, and the access control subsystem. The frontend contains the PIV card, the card reader, and the PIN input device. The Issuance and management subsystem comprise the components responsible for the identity proofing and registrations, the card and key issuance and management, and PKI directory and certificate status servers. The access control subsystem consists of the logical and physical access control systems, the protected data, and the authorization data.

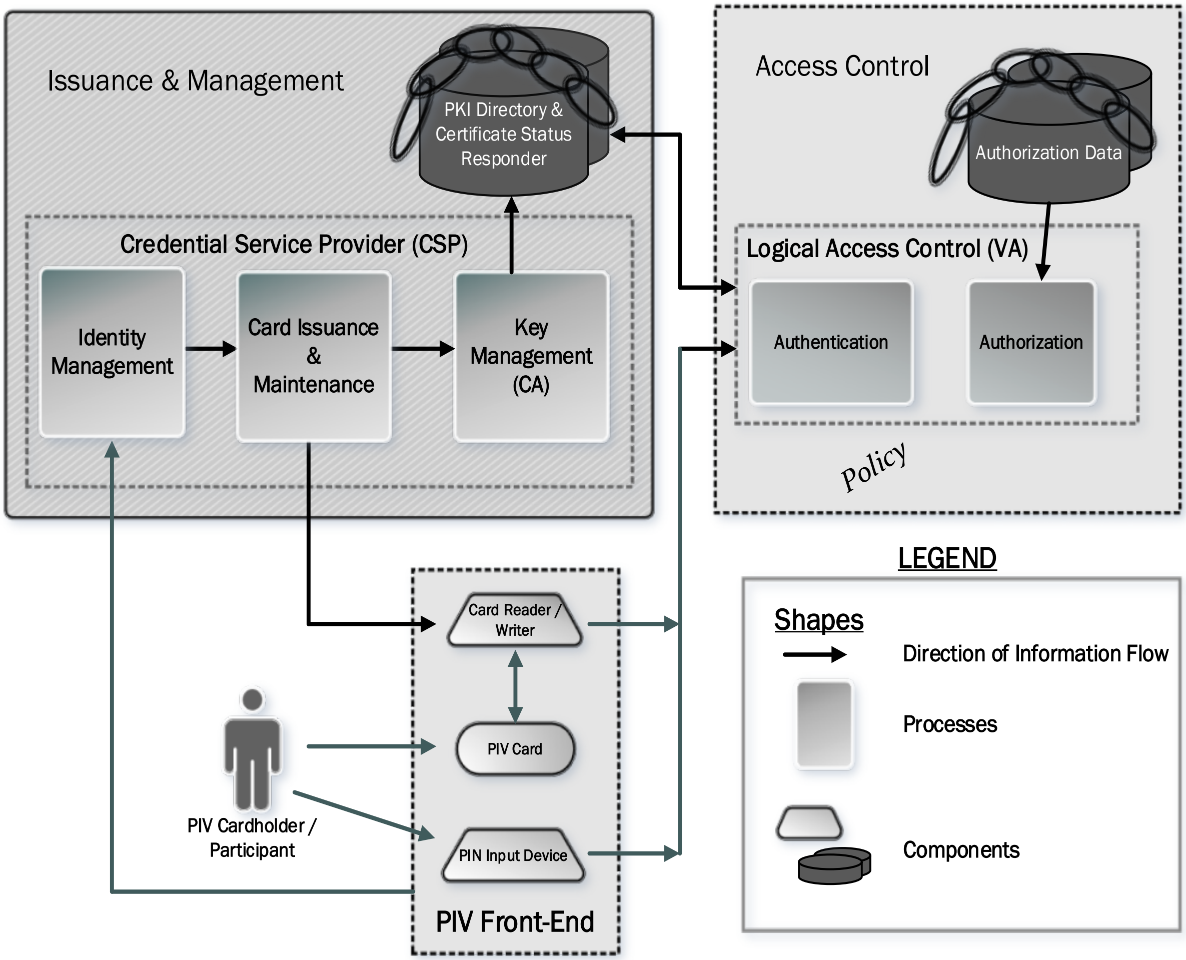


Figure 11: ICAM PIV System Overview

3.4 ISSUANCE AND MANAGEMENT SUBSYSTEM

The issuance and management subsystem can be broken down into two major components; identity management and credential management. Identity management allows an organization to establish, maintain, and terminate identities. Credential management allows an organization to issue, track, update, and revoke digital credentials for identities.

3.4.1 Identity Management

The foundation of any system managing identity is trust in the authenticity of that identity. In the United States (U.S.), when a citizen is born, the parents provide the hospital with a name. The hospital issues a notarized birth certificate with the person's name, date of birth, and U.S. citizenship. That person now has an identity for life. Any time the person wants to prove who they are, they simply provide their birth certificate. With a birth certificate, a U.S. citizen can obtain a U.S. passport as it is unlawful to enter or exit the United States without one. The U.S. citizen also resides in a state or U.S. territory that will issue a state identification (ID) that most citizens obtain in the form of a driver's license. Citizens, government agencies, and industry trust birth certificates, passports, and driver's licenses as identification for an individual. Since digital identities require the same level of trust, identities within ICAM begin with a credential service provider that vets the person's identity through the showing of a birth certificate, passport, or driver's license/ state ID.

3.4.1.1 Identity Proofing

Identity proofing is the process by which an identity is first established. This process can be simple or complicated depending on the IAL that is required of the identity. The ICAM Framework strongly encourages IAL 3 for proofing the identity of the person or object. This means the digital identity is verified and validated by an authorized and trained credential service provider by the physical presence of the real person or thing.

3.4.1.2 Creation

Establishing a digital identity record within the system composed of attributes that define a person or entity. Each identity must be associated with an identifier. An identifier is a unique attribute that can be used to locate a specific identity. For example, a state may have more than one Susan Jones but the Department of Motor Vehicles (DMV) provides each with a different driver's license number.

3.4.1.3 Maintenance

Once a record of the identity is stored within the system it must be maintained. Identity lifecycle management needs to occur ensuring accurate and current attributes within an identity record over its life cycle. For example, current home address and phone number is in the system if the person moves, etc.

3.4.1.4 Identity Resolution

If for whatever reason, there is more than one identity record for the same person or entity, the identity must be reconciled. There must be only one identifier per person or entity for the system to maintain its credibility and trustability. Extreme care must be taken to ensure that no loopholes exist in the identity proofing to allow a malicious user that ability to create a second or third identity record.

3.4.1.5 Deactivation

Identity record deactivation is critical for similar reasons as identity resolution. When a person or entity should no longer have access to the infrastructure for whatever reason, their identity record must be deactivated immediately. This maintains confidence and assurance that an identity can be trusted.

3.4.2 Credential Management

Just like there must be a trusted agency and system to manage identities, there must be a trusted agency and system to manage credentials, a CSP. The CSP will issue, track, update, and revoke credentials for an entity within an organization. Following HSPD-12, the PIV credential is the one ICAM Framework uses as digital proof of an entity’s identity. The CSP will function at IAL3. With the ICAM Framework following the criteria for superior identity evidence, individuals, government, and industry can be assured that an identity is authentic and has not been forged. They can feel confident that any information they have been provided is correct and pertains to the real-life object. When a CSP is IAL3 and meets superior strength for identity evidence the goal of establishing a connection between a claimed identity and a real-life object is confirmed through following the highest levels of identity verification. There are 5 phases of credential management, sponsorship, registration, issuance, maintenance, and revocation.

3.4.2.1 Sponsorship

Before an entity is provided a digital certificate that allows access to a network or system, the entity must be sponsored. Via an authorized role through a registered entity following the principle of least privilege, an entity is granted permission for admittance to only the devices or applications in which there is a strong validated reason to access. This authorization should be granted via documentation that is stored at registration in case the need should ever arise to confirm or investigate the reasons access was granted. The authorization will include the identity of the entity, the systems, devices, and application the entity is allowed to access, and the duration this permission is allowed. Every sponsorship is allowed a maximum of two years. This is to ensure a continuous monitoring of all entities permissions in order to evaluate the risks to internal assets and business functions in order to provide better security.

3.4.2.2 Registration

A typical registration would consist of the entities name, address, sponsorship, and identity validation. If the entity is a non-person such as a system, application, or mobile device then the address would consist of the entities mac address instead of a physical address. The identity of the entities must be confirmed through an organization following the identity management of the ICAM Framework. The sponsorship is a first check on a valid entity but the entities identity should be confirmed again at registration as a defense in depth process to ensure the truth of identities allowed to register.

3.4.2.3 Issuance aka the Certification Authority (CA)

Issuance of a digital identity is done through the CA. Once identity has been confirmed real and has sponsorship, a digital identity is issued on a PIV card. The PIV card is a contact/contactless java card that follows the FIPS201 standards. This digital identity is a signed certificate. The CA will issue a signed certificate following the X.509 standard that certifies an entity’s ownership of a public key. This public key will be added to the blockchain. The private key remains in the entity’s possession securely stored on the PIV card.

The ICAM PIV X.509 Certificate structure follows NIST and X.509 standards because it works therefore no point in reinventing the wheel. The only difference is the addition of the Blockchain name and blockchain public key.

Version
Serial number
Algorithm ID
Issuer
Validity --Not Before Date -- Not After Date
Subject
Subject Public Key Info
Public Key Algorithm
Subject Public Key
Issuer Unique Identifier
Subject Unique Identifier
Extensions
Blockchain Unique Identifier
Blockchain Public Key
Certificate Signature Algorithm
Certificate Signature

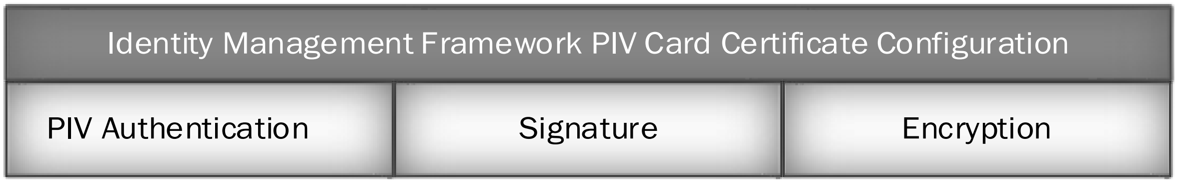
Table 2: Structure of ICAM PIV X.509 Certificate

An example of an ICAM X.509 certificate:

```
Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
  Signature Algorithm: sha256WithRSAEncryption
```

```
Issuer: C=BE, ST=East Flanders, L=Ghent, O=KU Leuven - Campus Ghent,
  O=Computer Science Department, OU=MSEC, CN=MSEC Tutorial Client CA
Validity
        Not Before: May 12 13:38:13 2014 GMT
        Not After : May 12 13:38:13 2015 GMT
Subject: C=BE, ST=East Flanders, L=Ghent, O=KU Leuven - Campus Ghent,
   O=Computer Science Department, OU=MSEC, CN=Alice in Wonderland
   /emailAddress=alice@msec.be
Subject Public Key Info:
   Public Key Algorithm: rsaEncryption
        Public-Key: (2048 bit)
        Modulus:
                00:be:2e:3d:32:72:4a:92:ff:67:7b:df:7c:89:83:
                ...
                72:91
        Exponent: 65537 (0x10001)
X509v3 extensions:
        X509v3 Basic Constraints:
                CA:FALSE
        X509v3 Subject Key Identifier:
                5D:1D:32:41:95:72:C6:CA:9C:E6:91:4B:32:50:C7:6E:14:68:F9:CA
X509v3 Authority Key Identifier:
        keyid:8B:D5:5E:F2:84:62:04:E4:91:25:78:74:87:14:5F:F2:F0:20:AC:2E
        DirName:/CN=MSEC Tutorial Root CA/OU=MSEC/O=KU Leuven - Campus Ghent/
O=Computer Science Department/L=Ghent/ST=East Flanders/C=BE
serial:02
X509v3 Key Usage:
        Digital Signature, Non-Repudiation, Key Encipherment
X509v3 Extended Key Usage:
        TLS Web Client Authentication
Signature Algorithm: sha256WithRSAEncryption
        3e:be:89:73:ed:92:ff:f2:89:2b:98:0a:46:e8:26:b7:af:53:
        ...
        a7:4a:ec:89
```

There are three certificates on the PIV card; encryption certificate, signature certificate, and PIV-auth certificate. The PIV-Auth will be used for authentication, Signature will be used for email and document signing, and encryption will be used for encryption (obviously).



Identity Management Framework PIV Card Certificate Configuration		
PIV Authentication	Signature	Encryption

Table 3: ICAM PIV Card Certificate Configuration

3.4.2.3.1 Blockchain

The CA issuing the certificates is also responsible for the blockchain. Since trust in the identity is the primary aim of the ICAM Framework, a private blockchain is instantiated. Only vetted agencies such as the CAs through the CSPs may make changes to the blockchain. A vetted agency is one that can be trusted by confirmation through a certificate issued to them from the root CA.

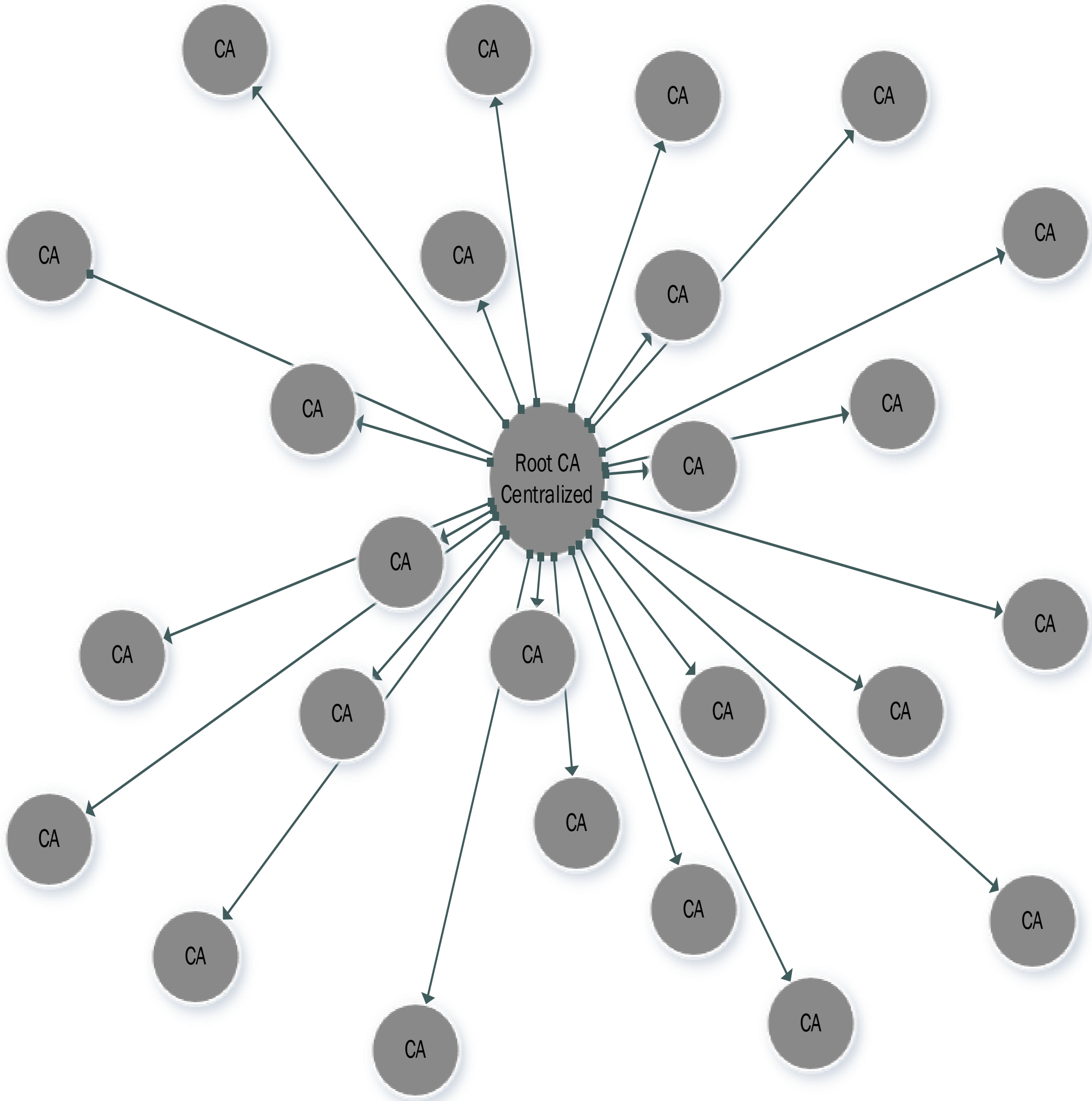


Figure 12: The root certificate authority is centralized

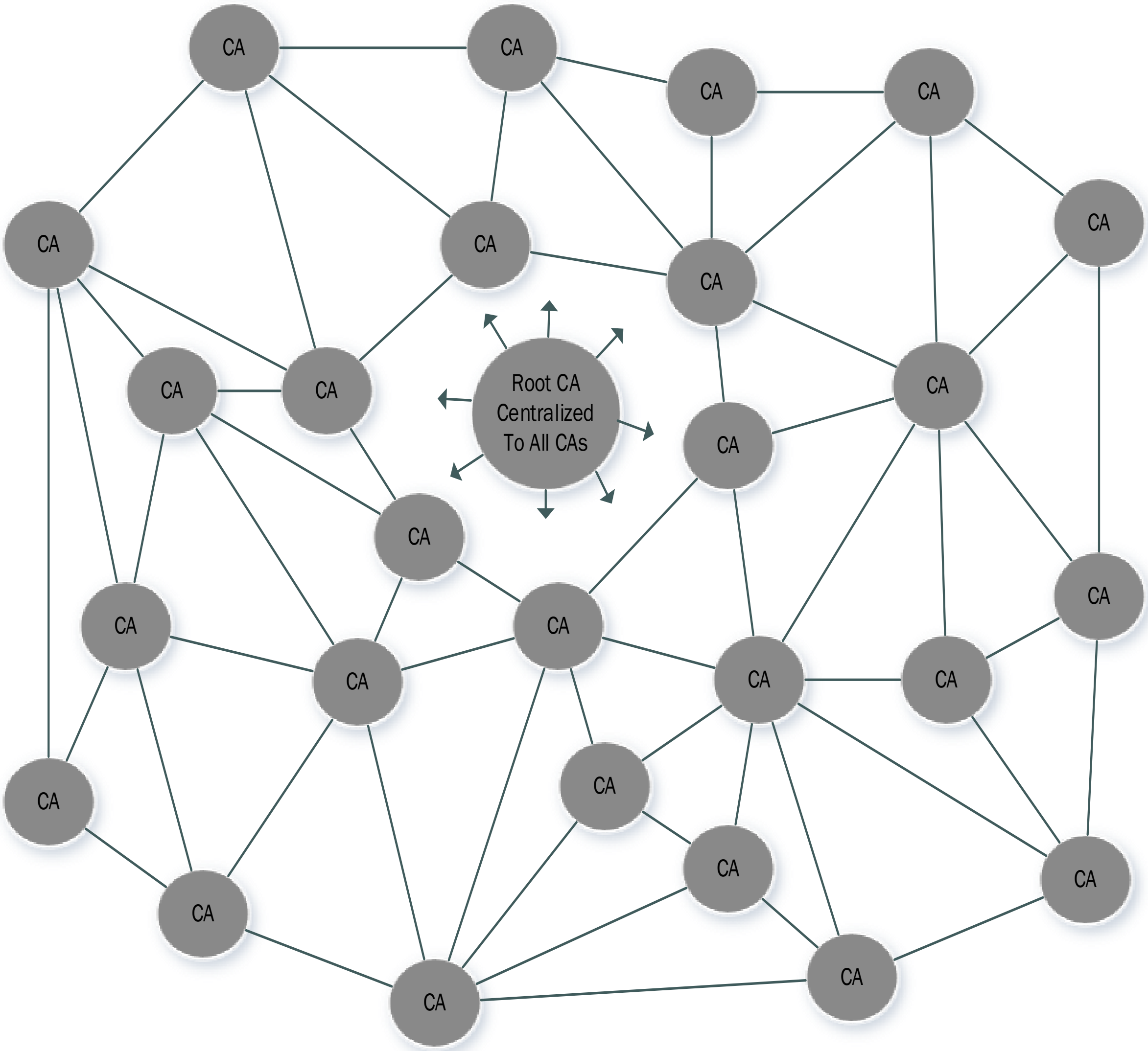


Figure 13: All other CA's are distributed

There are two blockchains, an active blockchain and a revoked blockchain. The active blockchain contains the list of entities, their access privileges, and their public keys that are allowed onto the network/system. The revoked blockchain contains a list of entities and their certificates public keys that are no longer allowed onto the network/system. Every ten minutes if a change to the directory has occurred, the CA adds a new block and chains it to the previous block. This heavily encrypted new block contains a ledger of all of the public keys of authorized entities and their permissions. This block is signed with the CAs private key. Since the blockchain is private and the CA is a trusted entity within the system, no mining is necessary. The new block is sent across the network to allow all nodes to sync.

3.4.2.4 Maintenance

Maintaining the blockchains of digital certificates is the most critical area to ensure success of the ICAM framework. Certificates expire, get revoked, or people lose them. All of this requires care and attention to the process. Maintenance is comprised of 5 key subsets: renewal, reset, suspension, blocking, and reissuance. Naturally, the blockchain kept as is with new blocks added every ten minutes would grow larger than the system could efficiently handle. Therefore, the root CA will monitor system productivity and make structural changes to the blockchains as needed. As there is only one root CA this maintains the integrity of the blockchains and system efficiency.

3.4.2.4.1 Renewal

When a certificate expires the entity must go through the same process that they did initially in order to obtain new credentials. In order for the entity not to lose access to the systems and services they need, the renewal process should be started before the certificate expires. All expired certificates will be kept linked to the entity for accurately identifying the entities history. Since history is not a primary feature of the system, records of old certificates will be kept in a traditional database that can be accessed as needed by the system. Renewing a certificate must insure no duplicate digital identities therefore the reset step follows renewal.

3.4.2.4.2 Reset

The reset process is similar to identity resolution in the Identity Management system. There must be only one active certificate per entity for the system to maintain its credibility and trustability. The reset process consists of changing the old certificate to the new certificate throughout the system. A thorough check and recheck will be performed to ensure the system functions with the new certificate being the only certificate tied to the entity.

3.4.2.4.3 Suspension

There are times when an entities certificates will need to be suspended. This means that the entity cannot continue to use the certificate for a period of time. The suspension time period must be specified with a start and end date. If the suspension is not resumed before the end date, the certificate is revoked.

3.4.2.4.4 Blocking

The ICAM Framework is intended to provide trust in the identity of entities using a system. Since no system is perfect, instances may occur when a digital credential must be blocked. This means that the entity may not access some or all of the network, applications, and systems. An update to the blockchain occurs instantly. There is no waiting the usual 10 minutes to ensure the strictest security possible to the system.

3.4.2.4.5 Reissuance

Reissuance is very similar to renewal but occurs before the expiration of an existing certificate. This could occur if the entity's PIV Card is lost or stolen. If the PIV has been stolen, reissuance and blocking should occur. Reissuance follows all steps from the renewal process.

3.4.2.5 Revocation

If an entity loses their privileges to the system, their certificates can no longer be used. The reason for the revocation is documented and digitally signed by an authorized entity. The revoked blockchain is updated.

3.5 ACCESS CONTROL SUBSYSTEM

The access control subsystem provides access management which allows only those entities permitted the ability to perform an action on a particular resource. The access control subsystem is where the verification authority (VA) is housed. The three most common access management services are policy administration, authentication, and authorization.

3.5.1 Verification Authority (VA)

Before access is granted to any system the VA is the system that will confirm the validity of the certificates by checking against the active blockchain and the revoked blockchain. The VA is a node on the network of those receiving the blockchains. It makes no changes to the blockchains. It only reads them. All this system does is compare the certificates from an incoming request to the blockchains. Having the blockchains sent to the VA as part of the blockchain peer-to-peer network reduces the amount of network traffic required for certificate validation. The lists are there at the system where access is being requested.

3.5.2 Policy Administration

The ICAM Framework does not establish specific policy administration. This is left up to the system owners who are using the ICAM Framework as only they know the laws, regulations, rules, and organizational access policies to put into effect for their system. Policy administration is mentioned within the ICAM Framework to ensure that access policies are established for the system. The rules and regulations for organizational access policies should be clearly defined and documented. Credential management will follow policy administration guidelines.

3.5.3 Authentication

The VA will check that the credential presented via a PIV card was issued by a trusted organization, the credential’s expiration date, and if the credential is on the revoked blockchain. When the entities identity is confirmed through the VA using its credential to be valid, the entity will confirm that the credential belongs to them by entering their PIN. When both the certificate and PIN are confirmed (two-factor authentication), the entities identity is considered authenticated.

3.5.4 Authorization

Just because an entity has been confirmed authentic does not mean that entity has access to the system. Once authenticated, the entities authorizations listed on the active blockchain are checked. If the system the entity requested access to is on the list, authorization is granted. If the system is not on the list, the entity is denied authorization.

3.5.5 Access Management

Policy administration determines the rule sets that govern access to resources. The linking and unlinking of access permissions for the entity to a resource must be followed. For example, policy sets the length of time before the entity needs to reauthenticate themselves. This could be a prompting to reenter the PIN assuring the entity is the genuine. Or locking the entity out of the resource and reinitiating the request for access.

3.6 FRONTEND SUBSYSTEM

The frontend subsystem consists of the PIV card, the PIV reader, and the PIN input device. The user will present their PIV card to the PIV reader and be prompted by the PIN input device to enter their PIN. The PIV reader and the PIN input device communicates with the access control subsystem to establish authentication and authorization.

3.6.1 PIV Card

The PIV card will be a contact/contactless Java Card that follows NIST Special Publication 800-73. The PIV card will contain a cardholder unique identifier (CHUID) that is digitally signed by the authorizing CA and the card authentication. The card authentication is a certificate containing the key pair that is used to verify that the PIV credential was issued by an authorized entity, has not expired, and has not been revoked.

The following two electronic elements authenticate the identity of the PIV card owner. The PIV card will contain a photograph that is digitally signed allowing confirmation that the printed photo on the card has not been altered and PIV authentication. The PIV authentication is a certificate and key pair that is used to verify that the PIV credential was issued by an authorized entity, has not expired, has not been revoked, and the holder of the credential is the same individual it was issued to.

The following electronic elements are for usage by the PIV card holder. A digital signature which is a certificate and key pair that allows the PIV card owner to digitally sign a document or email, providing both integrity and non-repudiation. Encryption which is a certificate and key pair allowing the PIV card owner to digitally encrypt documents or email thus providing confidentiality through ensuring only authorized parties can read the document or email.

The PIV card contact and contactless portions will mimic each other by each containing all of the information listed above. The contact (chip) portion of the Java Card is to allow functionality with existing PKI readers on desktops and laptops while the contactless (NFC) portion of the Java Card provides the means to expand PKI to mobile devices.

3.6.2 PIV Reader

The PIV card reader is the piece of hardware that helps read the card. The card reader supplies power to the chip or NFC and allows the device's operating system to talk to the PIV credential chip or NFC operating system. Most smart mobile devices are equipped with the ability to read NFC (contactless portion of the PIV card). Existing PKI systems can continue to operate as they always have by using their current contact PIV card readers.

3.6.3 PIN Input Device

Middleware is the computer software or drivers that allow the computer (whether desktop or mobile) to interact with the PIV credentials to support authentication, digital signatures, encryption, and integrations with the personal identification number (PIN) device. On a standard desktop or laptop, the PIN input device will be a keyboard. On a mobile device, this could be a software tool that is part of the middleware or a separate piece of software that works in conjunction with the middleware. If the mobile device is a smartphone, the PIN input device could be the phone's keyboard, on-screen or physical.

3.7 Digital Identity Summary

The next iteration of DoD PIV cards from Gemalto will contain NFC capabilities. But since the DoD PKI system does not have a method to handle mobile devices, the NFC portion within the DoD PIV cards will not be utilized until a mobile PKI system is setup. The ICAM Framework provides that solution. YubiKey has already demonstrated how NFC can securely handle PIV credentials. NFC can securely contain the necessary credentials for digital identity. Therefore, the ICAM will require the use of Java Cards that are contact/contactless.

Having dual functionality will allow existing PKI systems to continue to operate as they have while providing the additional capabilities of mobile devices.

3.8 ZTA

The new Zero Trust Architecture (ZTA) reshapes traditional network defense by moving network defenses from wide network perimeters to narrowly focusing on individual or small groups of resources. The ICAM Framework ensures the user is who they say they are using ZTA for trusting no one. The system will believe the person is who they say they are only once proven. Policy administration should ensure the tenants of ZTA are being followed by its governance.

3.9 Blockchain-based Digital Credentials

Within blockchain-based digital credentials is a certification path that leads back to the root CA called the chain of trust. The root CA is where the chain of trust begins. Every CA must be trusted within the ICAM Framework or the chain of trust becomes broken. The root certificate is self-signed but every other certificate issued is signed by a CA. The CA's certificates are signed by the root CA.

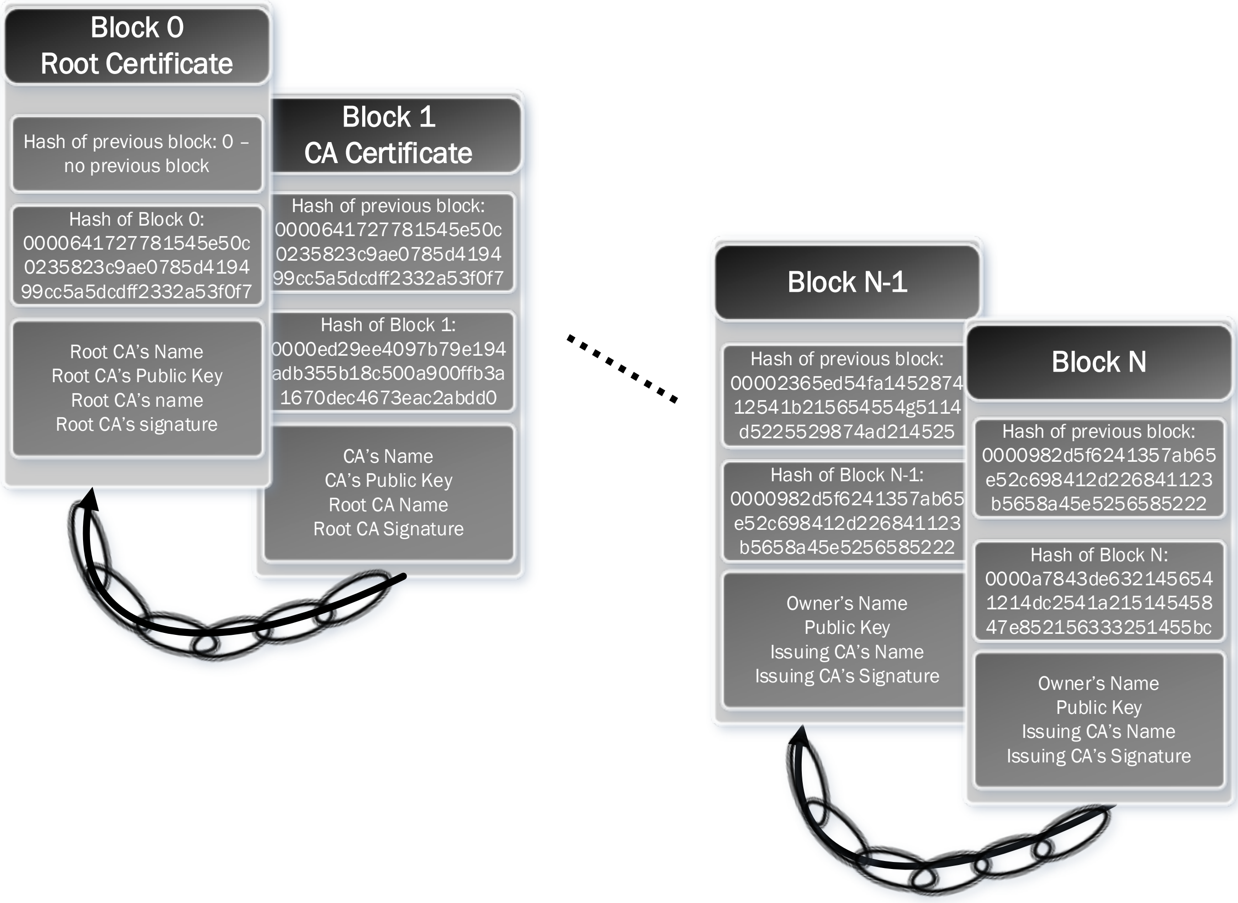

Block 0
Root Certificate
Hash of previous block: 0 – no previous block
Hash of Block 0: 000641727781545e50c 0235823c9ae0785d4194 99cc5a5dcdff2332a53f0f7
Root CA's Name
Root CA's Public Key
Root CA's name
Root CA's signature
Block 1
CA Certificate
Hash of previous block: 000641727781545e50c 0235823c9ae0785d4194 99cc5a5dcdff2332a53f0f7
Hash of Block 1: 0000ed29ee4097b79e194 adb355b18c500a900ffb3a 1670dec4673eac2abdd0
CA's Name
CA's Public Key
Root CA Name
Root CA Signature
Block N-1
Hash of previous block: 00002365ed54fa1452874 12541b215654554g5114 d5225529874ad214525
Hash of Block N-1: 0000982d5f6241357ab65 e52c698412d226841123 b5658a45e5256585222
Owner's Name
Public Key
Issuing CA's Name
Issuing CA's Signature
Block N
Hash of previous block: 0000982d5f6241357ab65 e52c698412d226841123 b5658a45e5256585222
Hash of Block N: 0000a7843de632145654 1214dc2541a215145458 47e852156333251455bc
Owner's Name
Public Key
Issuing CA's Name
Issuing CA's Signature

Figure 14: Chain of Trust

3.9.1 What is being decentralized (distributed)?

What systems within PKI is being decentralized for the ICAM Framework? The root CA is centralized but the rest are distributed. There are many CAs that can be spread out among a country or the world. Each CA is a node on the peer-to-peer network. Only the CAs are allowed to make changes to the blockchains. The VAs are distributed as well. There are many VAs spread out among the resources requiring access control. The CAs and VAs do not have to reside together but in some instances they could.

CHAPTER 4

4 CASE STUDY (RESULTS AND DISCUSSION)

In developing a proof of concept that would answer the research questions it was necessary to create two proof of concepts; a digital identity on NFC and a digital identity on blockchain-based PKI. This allowed the research to focus on the questions and reveal emerging knowledge that contributed to the ICAM framework through each iteration of the DSR methodology. It also allowed for the two proof of concepts to provide one artifact that becomes the object of study.

The most challenging part in developing the ICAM Framework was in figuring out a way to allow blockchain to revoke certificates. There was research and exploration in having a single blockchain that would contain everything. In the end, it was found that one blockchain that had valid and invalid certificates is harder to implement and has the potential for a revoked certificate to be missed. Therefore, two separate blockchains would be better; an active blockchain and a revoked blockchain. All certificates will be checked to ensure they are not on the revoked blockchain in much the same way as the certificate revocation list (CRL) works on the standard PKI system.

4.1 Digital Identities on NFC

Yubico’s Yubikey 5 NFC was used to demonstrate this proof of concept. The Security Key NFC from Yubico uses NFC for tap-and-go authentication over the FIDO U2F and FIDO2/WebAuthn protocols on Android phones. While FIDO2 is not the same as PKI, it demonstrates that NFC can be used contain a digital identity for authenticating users on their mobile devices.

FIDO uses public key cryptography which is issued to a user upon registration but the registration is self-sovereign. An individual saying who they are will not be a trusted digital identity. The digital identity can be linked to a single, unique identity and validated as genuine but the Yubikey demonstrates that public key cryptography will fit on an NFC card, can be read by a smartphone, and has no noticeable drain on the battery life. The Yubico FIPS series demonstrates that FIPS 140-2 validation can be integrated at some point with NFC capabilities to provide the highest authenticator assurance level 3 per NIST SP800-63B guidance.

Figure 15: YubiKey Touch-and-Go

4.1.1 Question Results

Digital Identity

- Can an identity be verified as belonging to a real person or thing supplying the digital identity?
  No, according to the tenets of ZTA, the digital identity using Yubico's YubiKey are not a verified identity. The digital identity is self-sovereign.
- Can a claimed identity be securely linked to a single, unique identity?
  Yes, a claimed identity can be securely linked to a single, unique identity.
- Can the evidence supplied by an identity be validated as genuine (e.g. not counterfeit or spoofed)?
  No, according to the tenets of ZTA, the identity cannot be validated as genuine.
- Can the digital identity be validated as existing in the real world?
  No, according to the tenets of ZTA, the identity is not validated as existing in the real world.

The iterations of testing the digital identity on a YubiKey as it exists with Google's FIDO U2F brought the importance of the ZTA to the ICAM framework. Trust is the most critical component in ensuring success in a PKI. In order to change the above no answers to yes, the identity management component of the CSP was instantiated within the ICAM framework.

The identity management component of the Issuance and Management Subsystem identity proofing phases ensures that the entity or person is real by requiring the physical presence and showing a valid form of identification such as a passport or driver's license. Connecting to database systems like the police or TSA officers use to check against counterfeits provides evidence that the identity is genuine and exists in the real world. The additional phases of identity management ensure ZTA throughout the lifecycle of an identity.

The CA's as they currently operate do not have any formal oversight so are usually only trusted within a business or organization. In order for all CA's to be trusted by all business', governments, and individuals and to ensure no spoofing or counterfeiting of the certificates, all CA's are vetted and registered to a centralized certification authority as part of the credential management component of the Issuance and Management Subsystem. This is the root certificate or root of trust that signs the certificates that the distributed CA's are issued. The additional phases of the credential management component ensure trust throughout the certificate's lifecycle.

In order to provide defense in depth trust in identities, authorization to the various systems, subsystems, and components of the Issuance and Management Subsystem, is only provided if the user's role requires it per the tenets of least privilege.

NFC

- Can the encryption size be reduced to fit on an NFC card?
  Yes, the YubiKey can use public key cryptography following FIPS AAL3 standards according to NIST SP800-63B guidelines.
- Can a smartphone with NFC enabled be a smart card reader?
  Yes, the smartphone could read the YubiKey every time with no problems.
- Is there a secure communication channel between the card and device?

Yes, on a Samsung Android device such as the one used for testing; Samsung Knox provides direct control of the NFC chip embedded in the device. The Knox Platform provides security hardening for every aspect of mobile device operation. It enables trust on mobile endpoints with Samsung's Real-Time Kernel Protection (RKP) that is considered one of the best kernel protection technologies available from any mobile device vendor (Samsung, 2019).

- What drain is there on the battery life?

  The mobile device used for testing was a Samsung Galaxy Note 10+. On this device there was no noticeable drain on the battery life. The battery percentage was documented within an excel spreadsheet at the start and end of an hour period of using the YubiKey to log into various Google accounts. Some differences could be attributed to receiving text messages or phone calls during that hour time period. It seemed to remain steadily at 1% if no texts or phone calls came in during the hour of testing.

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Average
Battery Start	76.00%	96.00%	77.00%	72.00%	75.00%	86.00%	1.80%
Battery End	75.00%	94.00%	74.00%	71.00%	73.00%	85.00%	
Usuage	1.00%	2.00%	3.00%	1.00%	2.00%	1.00%	

Table 4: Battery Life

Overall, the results showed that NFC is a viable option for updating PKI's front-end to allow functionality with mobile devices. The smart cards currently in use today Java 3 are already capable of being both contact and contactless making it an easier transition away from traditional PKI front-end usage. More details on how the YubiKey was installed and tested can be found in Appendix A.

4.2 Digital Identities on Blockchain-Based PKI

Remme's Protocol was used as the blockchain-based PKI. Only the free services of Remme was utilized but it was enough to demonstrate that PKI can become a distributed system with blockchain. REMChain is the first public blockchain to be built on top of Protocol. REMChain is not open-source but its success demonstrates that blockchain-based PKI can support millions of users, perform identity brokering, allow separate PKI systems from competitors to work together, and provide a trusted third-party CA.

4.2.1 Background

I began using some source code from GitHub on pki-blockchain (wshbair & alyakubov, 2019). The developers had created a Proof of Concept blockchain-based PKI implementation. Starting with their code would be faster than starting from scratch and why reinvent the wheel? In working with their code, I quickly realized that I would be very limited in creating a network. In conducting more research, I learned about Remme. Remme is a blockchain-based PKI protocol that can be used for issuing and managing public keys. Remme core is built on Hyperledger Sawtooth platform and allows flexibility in language choices during development. Remme also supports JS and .Net programming (Roman-tik, 2019).

The Remme protocol follows the X.509 standards for its public keys. The web server authenticates its client with the help of certificates and the associated private key. The current limitation is that the Remme Protocol on works on Ubuntu 18.04 (Remme, 2019).

Digital Identity

- Can an identity be verified as belonging to a real person or thing supplying the digital identity?

  No, according to the tenets of ZTA, the digital identity using Remme are not a verified identity. The digital identity is self-sovereign.
- Can a claimed identity be securely linked to a single, unique identity?

  Yes, a claimed identity can be securely linked to a single, unique identity.
- Can the evidence supplied by an identity be validated as genuine (e.g. not counterfeit or spoofed)?

  No, according to the tenets of ZTA, the identity cannot be validated as genuine.
- Can the digital identity be validated as existing in the real world?

  No, according to the tenets of ZTA, the identity is not validated as existing in the real world.

To turn these no answers into a yes, it was found that Remme's public key cryptography was lacking similar qualities that the YubiKey does in regards to establishing a trusted digital identity. In addition, the very public blockchain nature of Remme led to the

ICAM framework being built on a private based blockchain. This ensures that only verified entities are allowed access and permissions to make changes to the blockchain. Again, this is following the tenets of ZTA. More details on how Remme was installed and tested can be found in Appendix B.

Blockchain

- Can PKI become a distributed system with blockchain?
  Yes, running Remme's tests showed that PKI can become distributed with blockchain.
- Can blockchain allow separate PKI systems to function as one?
  Yes, theoretically it should allow separate PKI systems to function as one. This is because Remme has been implemented within RemChain. RemChain being a public blockchain has demonstrated success in separate PKI systems functioning as one.
- Can blockchain perform identity brokering?
  Yes, RemChain proves successful in performing identity brokering but it does not follow the tenets of ZTA
- Can PKI systems from competitors be trusted?
  No, not as RemChain or Remme currently exist due to a lack of vetting in identities and the openness of the public blockchain
- Can Blockchain-Based PKI provide a trusted identity for the internet?
  Yes, with changes to the type of blockchain used and processes that follow ZTA
- Can blockchain provide a trusted third-party certification authority?
  Yes, again with changes to the type of blockchain and ZTA
- Can the size of the blockchain be hosted by enough peers?
  Unsure of size limit. The blockchain testing worked well with numbers up to 10,000 blocks. The size did not reach the numbers it might when the entire internet works together to form a trusted web. More testing needs to be performed in future work.

In order for the blockchain-based PKI to form a web of trust, the tenets of ZTA needed to be implemented within the ICAM framework. A private, permissioned blockchain was chosen in order to ensure trust in the entities accessing the blockchain and making modifications. Roles will be limited based on least privilege separation of duties. For example, a VA system would only be allowed to authenticate and authorize an entity based on their certifications and permissions granted. The VA would only be allowed to read the revoked blockchain and not make any changes.

4.3 Future Work

The next step for the theoretical ICAM framework is to find a company to be the client as part of Technical Action Research (TAR). The researcher plays three roles: designer, helper, and researcher. The researcher desires to learn something about a technique by using it to help a client. Technology drives TAR not problems. In TAR, a client with an experimental artifact is the object of study. Therefore, a client uses the ICAM framework as the artifact and provides the funding to implement a real world blockchain-based PKI using NFC.

Instead of relying on Yubikeys actual Java 3 cards programmed specifically for the ICAM framework need to be used. The Java 3 card provides contact and contactless capabilities allowing the card to continue working with existing systems while making room for updating the system. Rather than using Remme for the blockchain-based PKI it must be created from scratch or start with one such as Fluree because a distributed permission blockchain-based PKI system needs to be established. This is distributed because all participating parties in the ICAM framework system need to work together and form a cohesive system of trust among each other. Permissioned blockchains maintain trust by only allowing certain actions to be performed by participants identified as needing to perform those actions. This ensures the tenets of zero trust are being followed.

CHAPTER 5

5 CONCLUSIONS

"Closing the identity gap is an enormous challenge. It will take the work of many committed people and organizations coming together across different geographies, sectors and technologies. But it's exciting to imagine a world where safe and secure digital identities are possible, providing everyone with an essential building blocks to every right and opportunity they deserve." – Peggy Johnson Executive VP, Business Development, Microsoft Corporation

Industry and government want to know who is on their networks and what they are doing while on them. People want to trust that the person they are talking with online is who they say they are and that the email is actually from the company it claims to be. The only way to achieve this is through trusted digital identities. The PKI systems that worked so well in the 20th century do not meet the mobile demands of the 21st century.

In today's dynamic world, PKI must become distributed and identity must be digitized in such a manner that an individual feel assured they have control over their personal data. As identity and security were an afterthought of the internet, safely maneuvering cyberspace relies on frameworks that will provide the needed infrastructure to protect an individual's, an industry's, and government's data. Blockchain-based PKI opens the door to a standard that allows everyone willing to participate to run on a federated infrastructure. This enables PKI to truly be a "public" system. Governments and industry following the same set of standards utilizing certificates signed by the same root certificate authority can all belong to the permission based blockchain where their CSPs are part of the distributed network. The ICAM Framework provides updates to PKI that establishes a digital identity for our mobile world that no other framework has given.

REFERENCES

Adams, C., & Lloyd, S. (2003). *Understanding PKI Concepts, Standards, and Deployment Considerations.* Boston, MA: Pearson Education, Inc.

Ballad, B., Ballad, T., & Banks, E. K. (2011). *Access Control, Authenticatin, and Public Key Infrastructure.* Sudbury, MA: Jones & Bartlett Learning, LLC.

Bashir, I. (2018). *Mastering Blockchain (second edition).* Birminghan, UK: Packt Publishing Ltd.

Batten, L. M. (2012). *Public Key Cryptography.* Piscataway, NJ: Wiley.

CardContact. (2019, December 15). *OpenCard Framework*. Retrieved from CardContact smart system architects: http://www.gemalto.com/techno/opencard/

Chen, Z. (2000). *Java Card Technology for Smart Cards.* Palo Alto, California: Sun Microsystems, Inc.

Chen, Z. (2000). *Java Card Technology for Smart Cards.* California: Sun Microsystems, Inc.

Chirico, U. (2014). *Smart Card Programming A comprehensive guide to smart card programming in C/C++, Java, C#, VB.NET.* Morrisville, NC: Lulu Press.

Crowd Research Partners. (2019). *Insider Threat 2018 Report.* CA Technologies.

D. Richard Kuhn, V. C.-J. (2001). *Introduction to Public Key Technology and the Federal PKI Infrastructure.* U.S. Government.

DISA. (2018, April 20). *milcloud*. Retrieved from DISA: https://www.disa.mil/Computing/Cloud-Services/MilCloud

DISA. (2018, April 18). *Our Work/DISA 101*. Retrieved from DISA: https://www.disa.mil/en/About/Our-Work

ENISA. (2019). *ENISA Threat Landscape Report 2018.* Heraklion, Greece: ENISA. doi:DOI 10.2824/622757

Ethereum. (2019, March 10). *Ethereum Docs*. Retrieved from Ethereum Homestead: http://www.ethdocs.org/en/latest/

Ethereum Foundation. (2019, March 10). *Ethereum*. Retrieved from Ethereum: https://www.ethereum.org/

Ethereum, Bitcoin, Blockchain, and Cryptocurrencies Resources. (2018). Wise Fox Publishing.

Evered, G. I. (1978, Dec). An Assessment of the Scientific Merits of Action Research. *Administrative Science Quarterly Vol 23 No 4*, pp. 582-603.

Experian. (2019). *2019 Global Identity and Fraud Report.* Experian Information Solutions, Inc.

FedRAMP. (2018, May 05). *about us*. Retrieved from fedramp: https://www.fedramp.gov/about/

Ferraiolo, H., Cooper, D. A., Regenscheid, A. R., Scarfone, K. A., & Souppaya, M. P. (2016). *Best Practices for Privileged User PIV Authentication.* NIST Pubs. doi:https://doi.org/10.6028/NIST.CSWP.04212016

Gates, M. (2017). *Blockchain: Ultimate Guide to Understanding Blockchain, Bitcoin, Cryptocurrencies, smart contracts adn the future of money.* Wise Fox Publishing.

Google. (2019, 09 March). *Near Field Communication Overview*. Retrieved from Android Developers: https://developer.android.com/guide/topics/connectivity/nfc

Google. (2019, October 5). *Titan Security key*. Retrieved from Google Cloud: https://cloud.google.com/titan-security-key

Google. (2019, August 7). *Turn on 2-Step Verification*. Retrieved from Google Account Help: https://support.google.com/accounts/answer/185839?hl=en

Grassi, P. A., Fenton, J. L., Lefkovitz, N. B., Choong, Y.-Y., Greene, K. K., Danker, J. M., & Theofanos, M. F. (2017, June). *NIST Special Publication 800-63A.* Retrieved from NIST: https://doi.org/10.6028/NIST.SP.800-63a

Gritzalis, D. N. (n.d.). User-aided reader revocation in PKI-based RFID systems. *Journal of Computer Security*, pp. 1147-1172.

Hansmann, U., Nicklous, M. S., Schack, T., & Seliger, F. (2000). *Smart Card Application Development Using Java.* Berlin: Springer.

IETF. (2019, March 09). *About*. Retrieved from IETF: https://www.ietf.org/

IETF. (2019, march 10). *Internet X.509 Public key Infrastructure Certificate and CRL Profile*. Retrieved from IETF: https://www.ietf.org/rfc/rfc2459.txt

Johannes A. buchmann, E. K. (2013). *Introduction to Public Key Infrastructures.* New York: Springer.

Kshetri, N. (n.d.). Can blockchain stregthen the internet of things. *IT Professional, 19*(4), pp. 68-72. doi:https://doi.org/10.1109/MITP.2017.3051335

L. Bosnjak, J. S. (2018). Brute-force and dictionary attack on hashed real-world passwords. *2018 41st International Conventnion on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1161-1166. doi:10.23919/MIPRO.2018.8400211

Leonard, H. (2012, December 5). *BII Mobile Insights*. Retrieved from More proof that the web is going mobile: https://www.businessinsider.com/the-world-is-going-mobile-2012-12

Ma, Y. (2011, August 4). Research on the solution of PKI interoperability based on validation authority. *Proceedings from the 2011 International Conference on Computer Science and Service System (CSSS).*

Matsumoto, S. &. (2017). IKP: turning PKI around with decentralized automated incentives. *Proceedings from the 2017 IEEE Symposium on Security and Privacy*, pp. 410-426. Retrieved from https://doi.org/10.1109/SP.2017.57

Matsumoto, S. R. (n.d.). Authentication challenges in a global environment. *ACM Transactions on Privacy and Security*, pp. 1-38.

Mayes, K., & Markantonakis, K. (2017). *Smart Cards, Tokens, Security, and Applications (second edition).* Cham, Switzerland: Springer.

NXP. (2019, September 25). *Products*. Retrieved from NXP: https://www.nxp.com/

Oxford. (2019, Oct 29). *Identity*. Retrieved from Lexico: https://www.lexico.com/en/definition/identity

Paar, C., & Pelzl, J. (2010). *Understanding Cryptography.* New York : Springer.

Paul A. Grassi, J. L. (2017). *Digital Identity Guidelines.* National Institute of Standards and Technology . National Institute of Standards and Technology Special Publication. doi:https://doi.org/10.6028/NIST.SP.800-63-3

Plos, T. H. (n.d.). Security-enabled near-field communication tag with flexible architecture supporting asymmetric cryptography. *IEEE Transactions on Very Large Scale Integration Systems*, pp. 1965-1974.

Ragjendran, B. (2017). Evolution of PKI Ecosystem. *2017 International Conference on Public Key Infrastructure and Its Applications (PKIA)*, 9-10.

Rahoof, P. N. (2017, April 24). Trust structure in public key infrastructure. *Proceedings of the 2017 2nd International Conference on Anti-Cyber Crimes*. doi:https://doi.org/10.1109/Anti-Cybercrime.2017.7905295

Rankl, W., & Effing, W. (2010). *Smart Card Handbook.* West Sussex, United Kingdom: John Wiley & Sons Ltd.

Remme. (2019, September 15). *Remme Protocol Documentation*. Retrieved from Remme: http://docs.remme.io

Rivera, R., Robledo, J. G., Larios, V. M., & Avalos, J. M. (2017). How Digital Identity on Blockchain can contribute in a smart city environment. *2017 International Smart Cities Conference (ISC2)*, 1-4.

Robey, C. (2017). Whom doyou trust part 2 blockchain technology & smart contracting. *Contract Management*, pp. 18-27.

Roman-tik. (2019 , September 12). *Remmeauth protocol*. Retrieved from Github: https://github.com/Remmeauth/remprotocol

Samsung. (2019, September 16). *The Samsung Knox Platform*. Retrieved from Samsung Knox Documents: https://docs.samsungknox.com/knox-platform-for-enterprise/admin-guide/why-use-knox.htm#h2_0

Seffers, G. I. (2017, August). DISA Moves Beyond Conventional Biometrics. *SIGNAL*, 27-28. Retrieved from www.afcea.org/signal

Stokkink, Q., & Pouwelse, J. (2018). Deployment of a Blockchain-Based Self-Sovereign Identity. *IEEE Conference on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics*, (pp. 1336-1342).

U.S. Government. (2019, June 06). *PIV Usage Guides*. Retrieved from ID Management: https://piv.idmanagement.gov/

United States Government. (2019, Oct 24). *PIV Usage Guides.* Retrieved from ID Management: https://piv.idmanagement.gov/

Vaishnavi, V. K., & Keuchler, Jr, W. (2015). *Design Science Research Methods and Patterns Innovating Information and Communication Technology.* Boca Raton, FL: CRC Press.

Verizon. (2018). *2018 Data Breach Investigations Report.* Verizon . Retrieved from https://enterprise.verizon.com/resources/reports/DBIR_2018_Report_execsummary.pdf

Verizon. (2019). *2019 Data Breach Investigations Report.* Retrieved from https://enterprise.verizon.com/resources/reports/2019-data-breach-investigations-report.pdf

WatchGuard. (2018). *Internet Security Report Q2 2018.*

White, A. K. (2018). *Blockchain: Discover The Technology Behind Smart Contracts, Wallets, Mining, and Cryptocurrency.* Abraham K White.

Wieringa, R. (2014). *Design Science Methodolgy.* New York: Springer.

wshbair, & alyakubov. (2019, August 2). *pki-blockchain*. Retrieved from github: https://github.com/snt-sedan/pki-blockchain

Yakubov, A. S. (n.d.). A blockchain-based PKI management framework. *Proceedings from IEEE/IFIP Network Operations and Management Symposium*.

Yubico. (2019, August 12). *YubiKey for Mobile*. Retrieved from Yubico: http://yubico.com

ZDNet. (2018, May 05). *2017s biggest hacks, leaks, and data breaches*. Retrieved from ZDNet: https://www.zdnet.com/pictures/biggest-hacks-leaks-and-data-breaches-2017/21/

APPENDIX A: YUBIKEY

YubiKey 5 NFC

YubiKey is really easy to set up to use to protect Google accounts. I've set mine up to use with all of my accounts. I utilize the Yubikey on both my desktop and my mobile device.

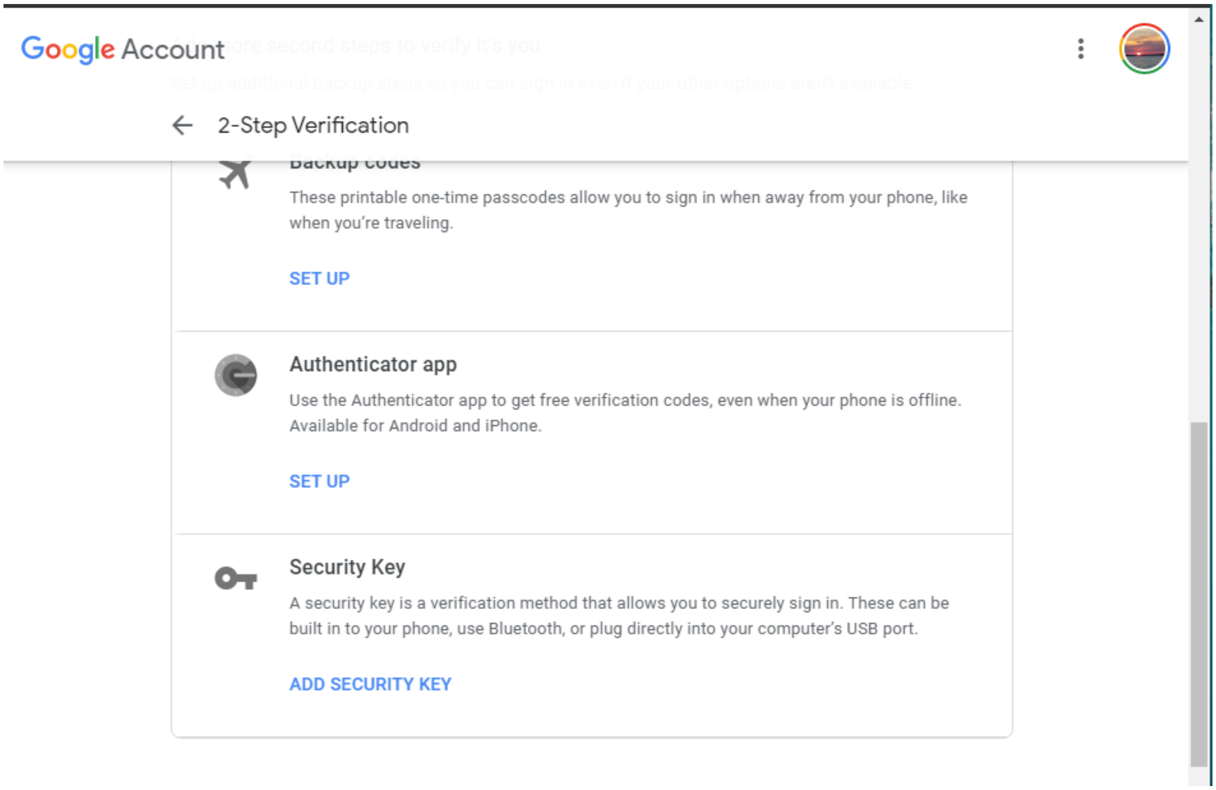
Turn on 2-Step Verification

First turn on 2-step verification. Go to your google account and on the left navigation panel, click security. On the signing in to Google panel, click 2-step verification (Google, 2019).

Figure 16: YubiKey 5 NFC

Add a Security Key

While logged into your Google account, go to Security Key under 2-Step. Click Add Security Key.



Add more second steps to verify it's you

Set up additional backup steps so you can sign in even if your other options aren't available.

Backup codes

These printable one-time passcodes allow you to sign in when away from your phone, like when you're traveling.

SET UP

Authenticator app

Use the Authenticator app to get free verification codes, even when your phone is offline. Available for Android and iPhone.

SET UP

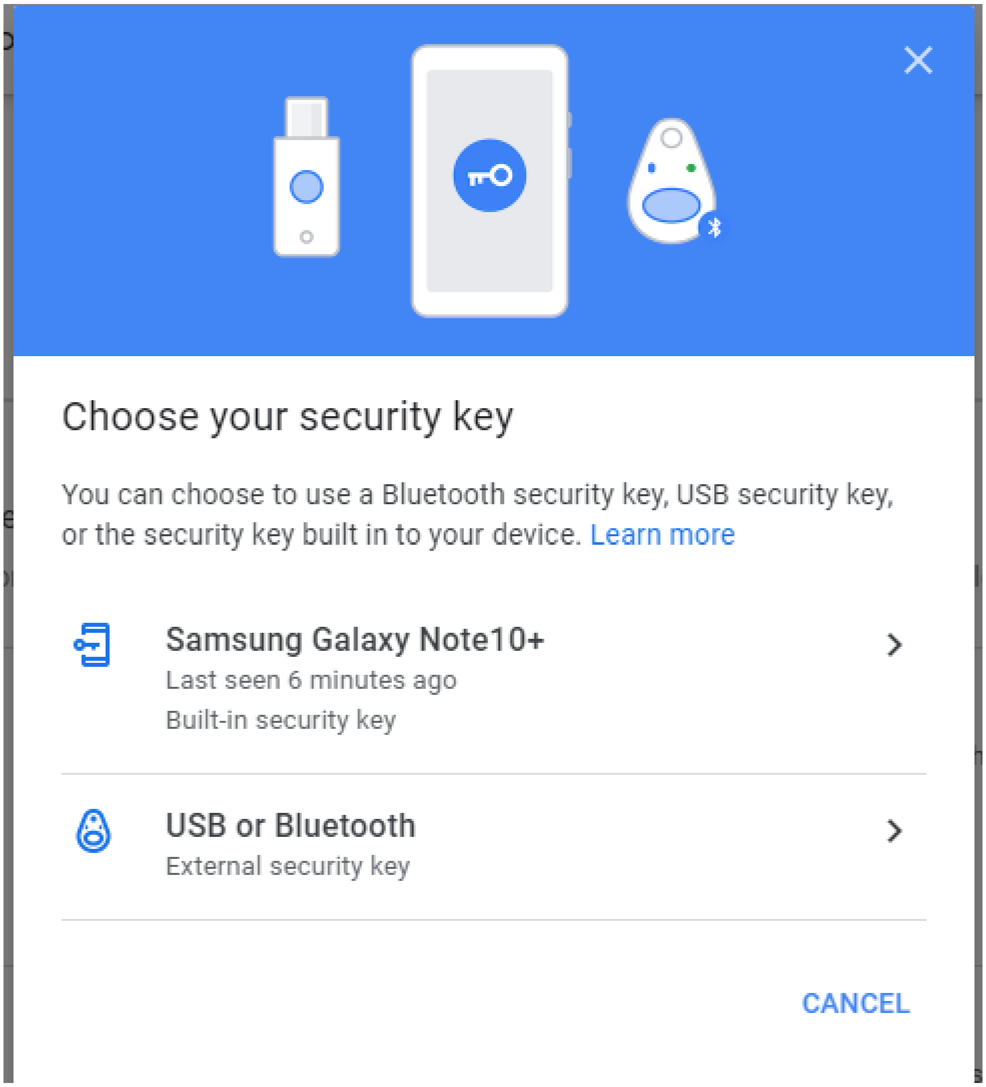
Security Key

A security key is a verification method that allows you to securely sign in. These can be built in to your phone, use Bluetooth, or plug directly into your computer's USB port.

ADD SECURITY KEY

Figure 17: Add Security Key

Choose USB or Bluetooh security key from the list.



Choose your security key

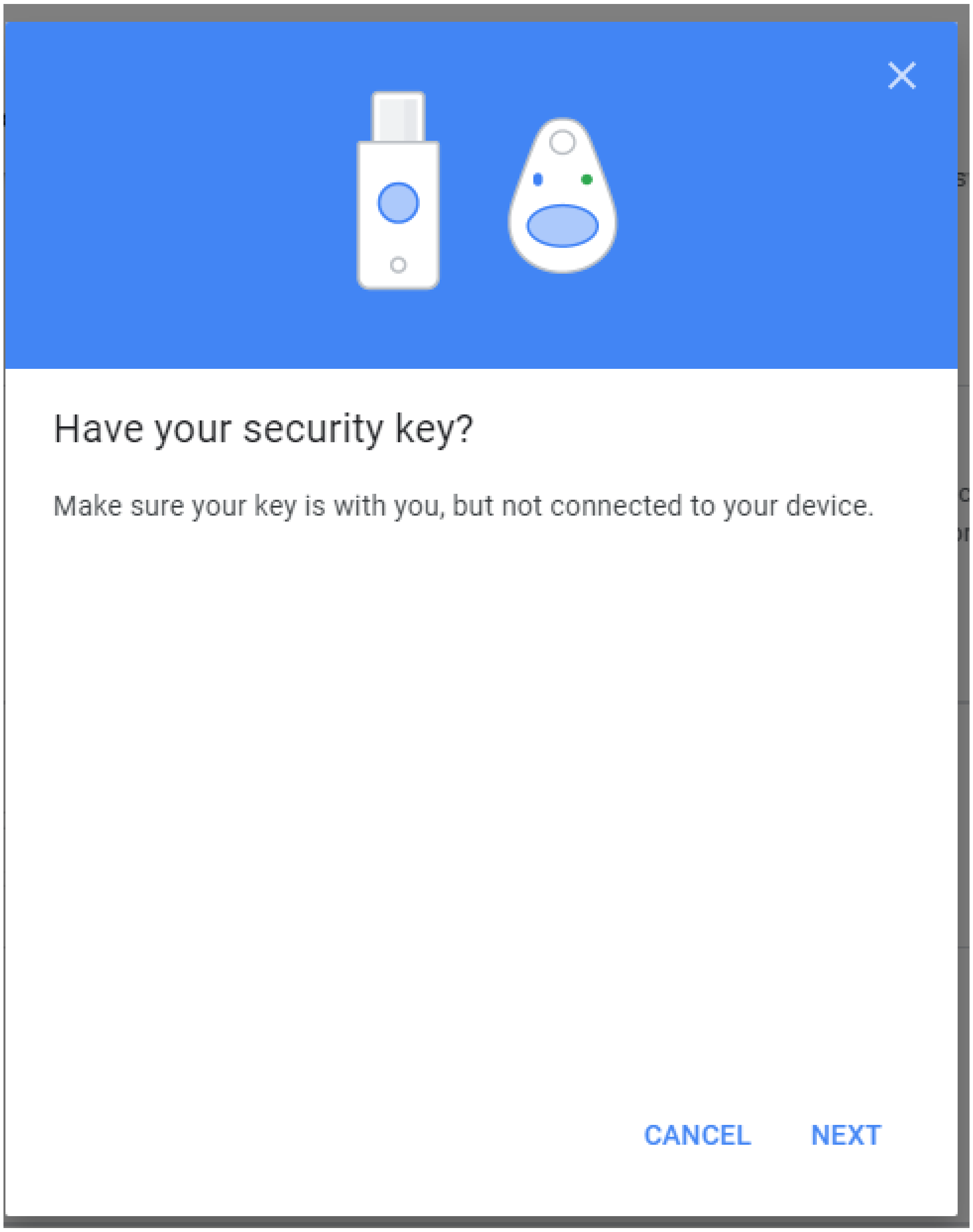
You can choose to use a Bluetooth security key, USB security key, or the security key built in to your device. Learn more

Samsung Galaxy Note10+
Last seen 6 minutes ago
Built-in security key

USB or Bluetooth
External security key

Figure 18: Choose Your Security Key

Next you'll be asked to ensure your YubiKey is with you but not connected. Then click next.

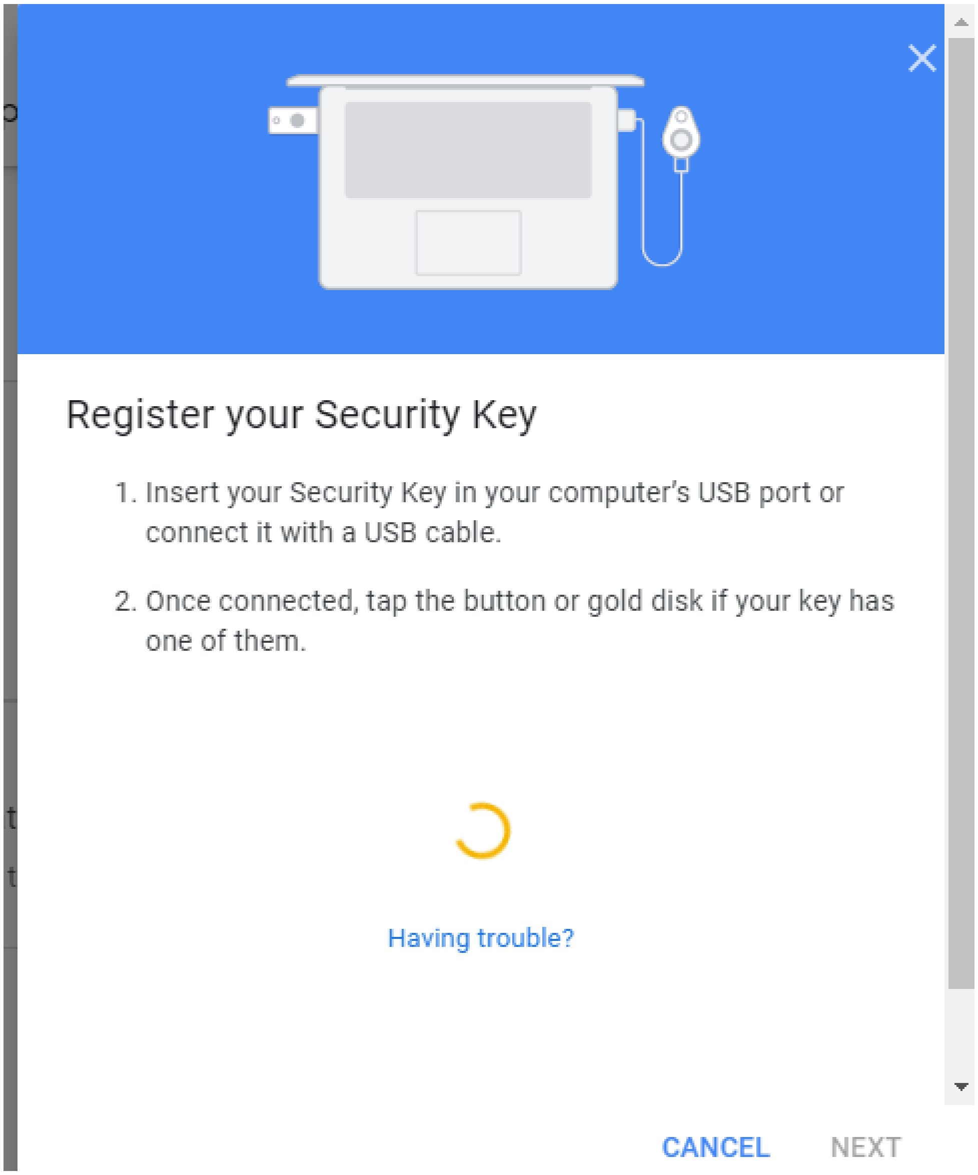


Have your security key?

Make sure your key is with you, but not connected to your device.

Figure 19: Ensure YubiKey is with you but not connected

When prompted insert your YubiKey into a USB port on the computer. Then touch the gold disk that is lit up on your YubiKey.



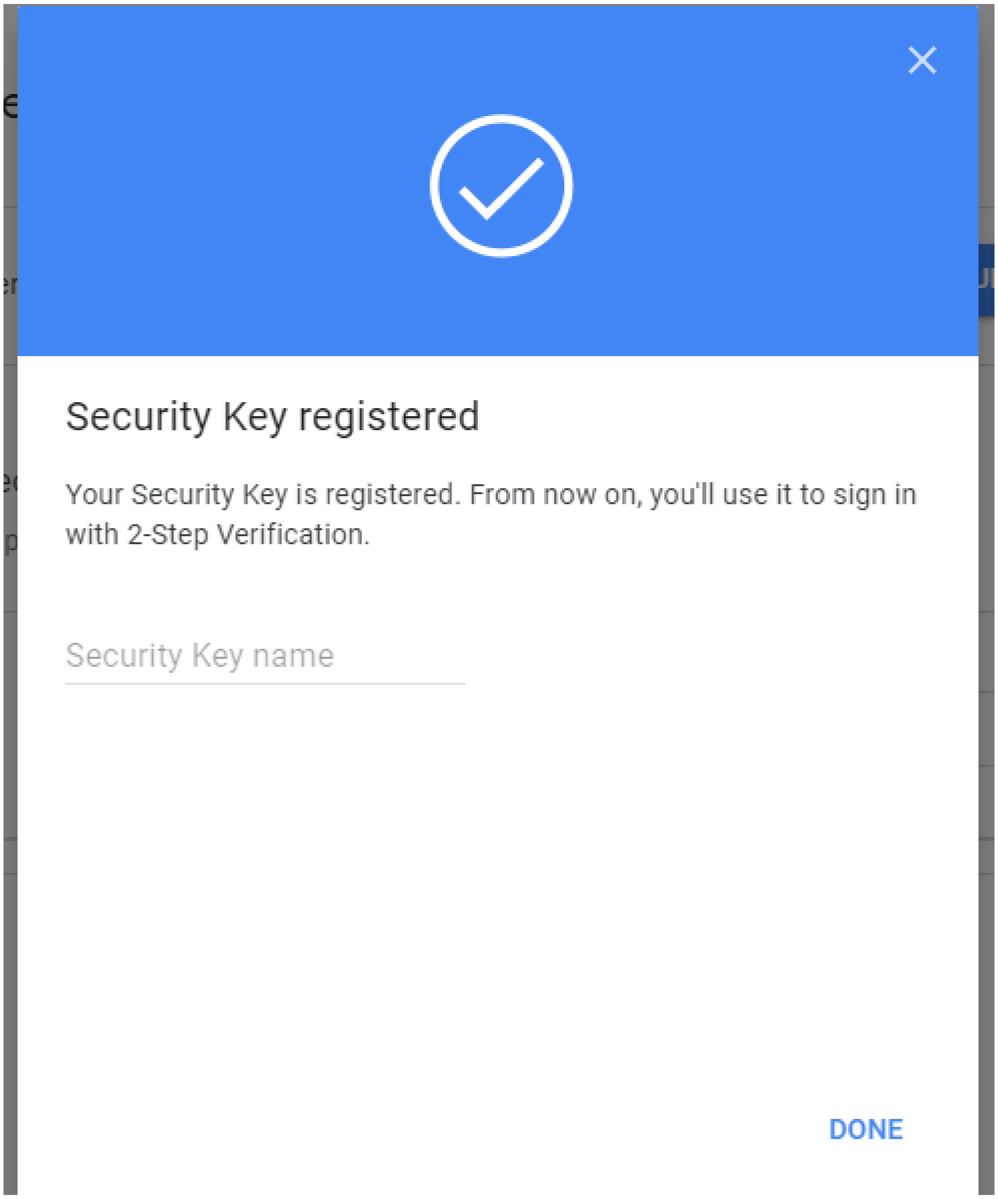
Register your Security Key

1. Insert your Security Key in your computer's USB port or connect it with a USB cable.

2. Once connected, tap the button or gold disk if your key has one of them.

Figure 20: Instructions to register YubiKey

Your YubiKey will be registered. Assign a name for your Yubikey and click done.



Security Key registered

Your Security Key is registered. From now on, you'll use it to sign in with 2-Step Verification.

Figure 21: Security Key Registered

Logging into Your Google Account

Now that a security key is registered with your google account, to log in on a computer you insert your Yubikey and touch it. To log in from a mobile device, you just tap your YubiKey (Yubico, 2019).

APPENDIX B: REMME

Installing Remme

All previous version of Remme Protocol must be uninstalled before installing again.

Uninstall Binaries

```
$ sudo dpkg -r remnode
```

Step 1: Install Binaries

On Ubuntu 18.04:

```
$ wget
https://github.com/Remmeauth/remprotocol/releases/download/v0.1.0/remmeprotocol_0.1.0-
ubuntu-18.04_amd64.deb && \
      sudo dpkg -i ./remmeprotocol_0.1.0-ubuntu-18.04_amd64.deb
```

Step 2: Boot node and wallet

Start remvault

```
$ remvault &
```

You will see an output similar to the one below:

```
info  2019-08-12T13:16:38.388 remvault  http_plugin.cpp:625        add_handler        ] add
api url: /v1/remvault/stop
info  2019-08-12T13:16:38.389 remvault  http_plugin.cpp:625        add_handler        ] add
api url: /v1/node/get_supported_apis
info  2019-08-12T13:16:38.389 remvault  wallet_api_plugin.cpp:73     plugin_startup      ]
starting wallet_api_plugin
info  2019-08-12T13:16:38.389 remvault  http_plugin.cpp:625        add_handler        ] add
api url: /v1/wallet/create
info  2019-08-12T13:16:38.389 remvault  http_plugin.cpp:625        add_handler        ] add
api url: /v1/wallet/create_key
info  2019-08-12T13:16:38.389 remvault  http_plugin.cpp:625        add_handler        ] add
api url: /v1/wallet/get_public_keys
```

Press enter to continue.

Start remnode

This command loads all the basic plugins, set the server address, enable CORS (with no restrictions and development logging) and add some contract debugging and logging.

```
$ remnode -e -p rem \
    --plugin eosio::producer_plugin \
    --plugin eosio::chain_api_plugin \
    --plugin eosio::http_plugin \
    --access-control-allow-origin='*' \
    --contracts-console \
    --http-validate-host=false \
    --verbose-http-errors >> remnode.log 2>&1 &
```

Note: In the above configuration, CORS is enabled for * for development purposes only, you should never enable CORS for * on a node that is publicly accessible!

Step 3: check that remnode is producing blocks

Run the following command:

```
tail -f remnode.log
```

You will see an output similar to the one below:

```
1929001ms thread-0   producer_plugin.cpp:585       block_production_loo ] Produced block
0000366974ce4e2a... #13929 @ 2018-05-23T16:32:09.000 signed by eosio [trxs: 0, lib:
13928, confirmed: 0]
1929502ms thread-0   producer_plugin.cpp:585       block_production_loo ] Produced block
0000366aea085023... #13930 @ 2018-05-23T16:32:09.500 signed by eosio [trxs: 0, lib:
13929, confirmed: 0]
```

```
1930002ms thread-0   producer_plugin.cpp:585       block_production_loo ] Produced block
0000366b7f074fdd... #13931 @ 2018-05-23T16:32:10.000 signed by eosio [trxs: 0, lib:
13930, confirmed: 0]
1930501ms thread-0   producer_plugin.cpp:585       block_production_loo ] Produced block
0000366cd8222adb... #13932 @ 2018-05-23T16:32:10.500 signed by eosio [trxs: 0, lib:
13931, confirmed: 0]
1931002ms thread-0   producer_plugin.cpp:585       block_production_loo ] Produced block
0000366d5c1ec38d... #13933 @ 2018-05-23T16:32:11.000 signed by eosio [trxs: 0, lib:
13932, confirmed: 0]
1931501ms thread-0   producer_plugin.cpp:585       block_production_loo ] Produced block
0000366e45c1f235... #13934 @ 2018-05-23T16:32:11.500 signed by eosio [trxs: 0, lib:
13933, confirmed: 0]
1932001ms thread-0   producer_plugin.cpp:585       block_production_loo ] Produced block
0000366f98adb324... #13935 @ 2018-05-23T16:32:12.000 signed by eosio [trxs: 0, lib:
13934, confirmed: 0]
1932501ms thread-0   producer_plugin.cpp:585       block_production_loo ] Produced block
00003670a0f01daa... #13936 @ 2018-05-23T16:32:12.500 signed by eosio [trxs: 0, lib:
13935, confirmed: 0]
1933001ms thread-0   producer_plugin.cpp:585       block_production_loo ] Produced block
00003671e8b36e1e... #13937 @ 2018-05-23T16:32:13.000 signed by eosio [trxs: 0, lib:
13936, confirmed: 0]
1933501ms thread-0   producer_plugin.cpp:585       block_production_loo ] Produced block
0000367257fe1623... #13938 @ 2018-05-23T16:32:13.500 signed by eosio [trxs: 0, lib:
13937, confirmed: 0]
```

Press ctrl + c to close an output.

Step 4: Check the Wallet

Run the following command, we need to validate the installation and check if wallet is working as intended:

```
$ remcli wallet list
```

You will see an output similar to the one below:

```
$ Wallets:
[]
```

Step 5: Check remnode endpoints

Run the following command, this will check that the RPC API is working correctly:

```
$ curl http://localhost:8888/v1/chain/get_info
```

Installing Testchain

Step 1: install binaries

Previous versions need to be uninstalled before installing. On Ubuntu 18.04:

```
$ wget
https://github.com/Remmeauth/remprotocol/releases/download/v0.1.0/remmeprotocol_0.1.0-
ubuntu-18.04_amd64.deb && \
      sudo dpkg -i ./remmeprotocol_0.1.0-ubuntu-18.04_amd64.deb
```

Step 2: boot node and wallet

Start remvault:

```
$ remvault &
```

You will see an output similar to the one below:

```
info  2019-08-12T13:16:38.388 remvault  http_plugin.cpp:625        add_handler        ] add
api url: /v1/remvault/stop
info  2019-08-12T13:16:38.389 remvault  http_plugin.cpp:625        add_handler        ] add
api url: /v1/node/get_supported_apis
info  2019-08-12T13:16:38.389 remvault  wallet_api_plugin.cpp:73     plugin_startup      ]
starting wallet_api_plugin
info  2019-08-12T13:16:38.389 remvault  http_plugin.cpp:625        add_handler        ] add
api url: /v1/wallet/create
```

```
info  2019-08-12T13:16:38.389 remvault  http_plugin.cpp:625          add_handler          ] add
api url: /v1/wallet/create_key
info  2019-08-12T13:16:38.389 remvault  http_plugin.cpp:625          add_handler          ] add
api url: /v1/wallet/get_public_keys
```

Press enter to continue

Step 3: Download testchain settings

```
$ wget https://testchain.remme.io/genesis.json
```

Step 4: Create Configuration File

Create `data` and `config` folders.

```
$ mkdir data && mkdir config
```

Create `config/config.ini` and put the following settings into it:

```
plugin = eosio::chain_api_plugin
plugin = eosio::net_api_plugin
http-server-address = 0.0.0.0:8888
p2p-listen-endpoint = 0.0.0.0:9876
p2p-peer-address = 167.71.88.152:9877
verbose-http-errors = true
```

These config options should get you into the basic operation mode with your node API available at port `8888`. P2p-peer-address points to the other nodes where to fetch the new blocks from (you may specify multiple entries, `167.71.88.152` is the address of a node hosted by `Remme`).

Start remnode

```
$ remnode --config-dir ./config/ --data-dir ./data/ --delete-all-blocks --genesis-json
genesis.json
$ remnode --config-dir ./config/ --data-dir ./data/ >> remnode.log 2>&1 &
```

The command above will run the node in the background and will save its output to the `remnode.log` file. At this point, you must be ready to start and connect your node to the network. If your node is connected and synced, this command should return you the information about the chain:

```
$ remcli get info
  {
    “server_version”: “96796929”,
    “chain_id”:
“93ece941df27a5787a405383a66a7c26d04e80182adf504365710331ac0625a7”,
    “head_block_num”: 680455,
    “last_irreversible_block_num”: 680121,
    “last_irreversible_block_id”:
“000a60b93d787895c905e36d7cf8d37a2bbed21d6f4b04f55645aefe459a32c0”,
    “head_block_id”:
“000a62074d3b6919262d90beecdffcc021fca03dc9ecd01ce4bfb91f8af36720”,
    “head_block_time”: “2019-08-12T15:08:58.500”,
    “head_block_producer”: “remproduce21”,
    …
  }
```

`remcli` (analog of cleos in EOSIO terms) is a command-line tool that has a rich variety of functions. It has nearly everything that you may need to interact with the blockchain. You may start getting familiar with it by running `remcli -help`.

Step 5: Become a Block Producer

To become a block producer you need to register your account via a system smart contract by calling the action `regproducer`, vote for someone or yourself, set up your node as a full node (described above) and prepare it for block production (so it starts to produce blocks in case you make it to the `top21`).

```
$ remcli system regproducer YOURACCOUNTNAME YOURPUBLICKEY
https://yourdomain.com
$ remcli system voteproducer prods YOURACCOUNTNAME YOURACCOUNTNAME
```

In your node config file, add these options:

```
plugin = eosio::producer_plugin
plugin = eosio::producer_api_plugin
producer-name = YOURACCOUNTNAME
signature-provider = YOURPUBLICKEY=KEY:YOURPRIVATEKEY
```

Once you run remnode, these config options should get you into block producer operation mode with your node. Once your block producer account gets into the top21 list, your node will automatically start producing blocks. Please pay attention that on the contrary to `EOS` network, `Block Producers` on `testchain` are required to validate the token swaps between the chains and have to run an additional bot (along the `remnode`) that monitors external blockchains (e.g. `Ethereum`).

Token swap

Download sources:

```
$ git clone -b block-producer-swap-bot --single-branch
https://github.com/Remmeauth/remprotocol.git && \
    cd remprotocol/block_producer_swap_bot
```

If you use `Ubuntu 18.04`, install dependencies with the following command

```
$ sudo ./scripts/ubuntu18.04_install.sh
```

Create configuration file with the following command:

```
$ nano ./config.ini
```

Paste into the config file the following content:

```
[NODES]
```

```
remnode=127.0.0.1:8888
eth-provider=wss://ropsten.infura.io/ws/v3/<your infura id>
[REM]
swap-permission=<permission to authorize init swap actions>@active
swap-private-key=<private key to sign init swap actions>
```

Replace remnode, eth-provider, swap-permission, swap-private-key with your remnode host and port, a link to Ethereum node with websocket connection, your account and private key to authorize init swap actions (for example your block producer account name and private key for signing blocks). Tutorial for creating Infura API key.

Save config file with Ctrl+O. Press Enter. Close config file with Ctrl+X.
To start approving swaps run the following command:

```
$ sudo ./scripts/run.sh >> swap.log 2>&1 &
```

Monitoring

Another option to check if your node has completed a correct setup is through monitoring. While starting the node, the monitoring has also been installed and started. Completing this step is required.

Monitoring is a process of tracking application performance to detect and prevent issues that could occur with your application on a particular server. For the monitoring, we will use `ELK stack`. It is an open source, feature-rich metrics dashboard and graph editor stack.

Step 1: install Docker

Ubuntu

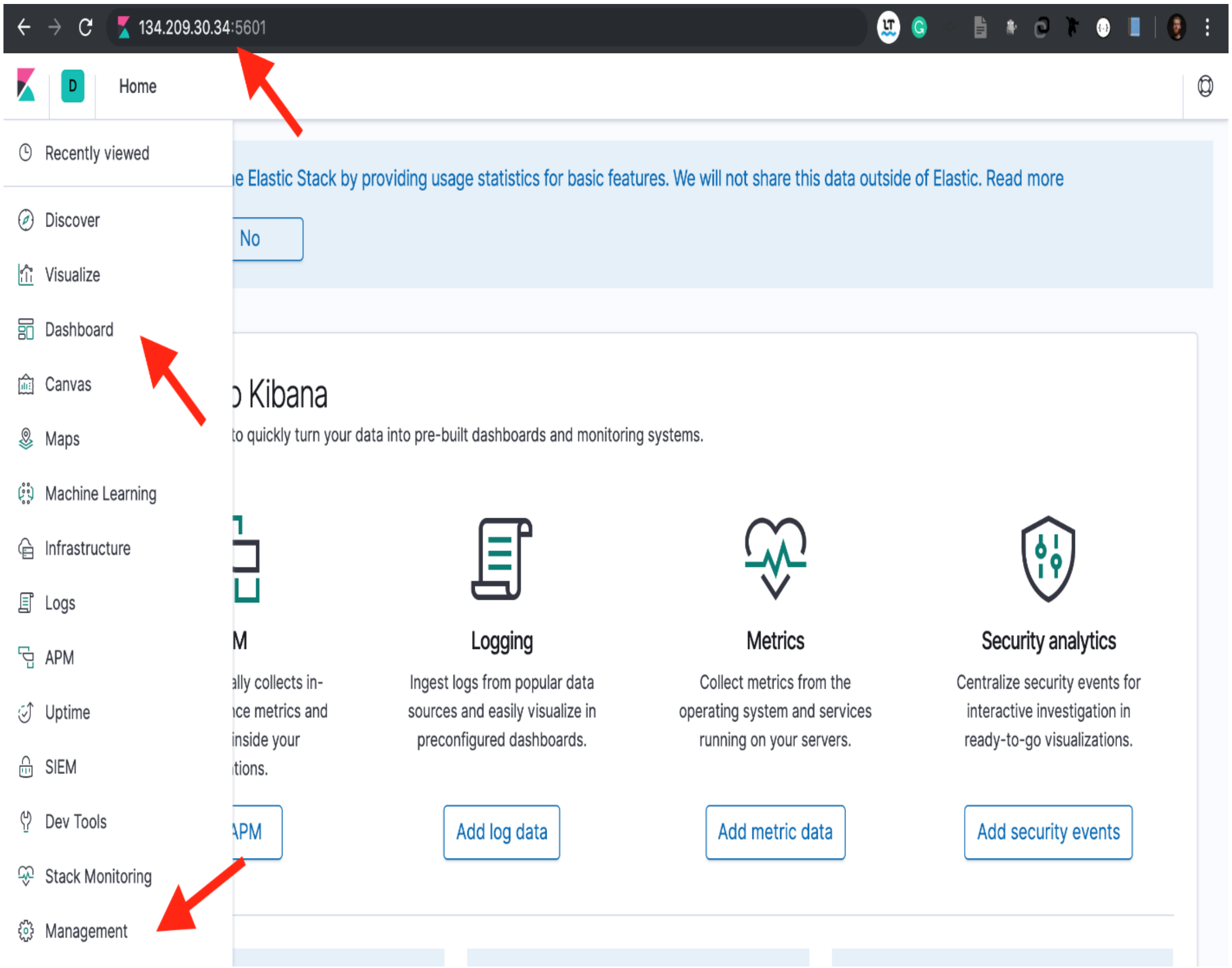
```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add && \
    sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable" && \
    sudo apt-get update && sudo apt-get install docker-ce -y && \
    sudo curl -o /usr/local/bin/docker-compose -L
"https://github.com/docker/compose/releases/download/1.23.2/docker-compose-$(uname -s)-
$(uname -m)" && \
    sudo chmod +x /usr/local/bin/docker-compose
```

Step 2: start the project

Download the project and start it with the following command:

```
$ git clone https://github.com/Remmeauth/protocol-monitoring && \
    cd protocol-monitoring && \
    sudo docker-compose -f docker-compose-linux.yml up -d && \
    curl -X POST "localhost:5601/api/saved_objects/_import" -H "kbn-xsrf: true" --form
file=@export.ndjson
```

Step 3: visualization and graphs



ie Elastic Stack by providing usage statistics for basic features. We will not share this data outside of Elastic. Read more

› Kibana

to quickly turn your data into pre-built dashboards and monitoring systems.

ally collects in-
ice metrics and
inside your
tions.

Ingest logs from popular data sources and easily visualize in preconfigured dashboards.

Collect metrics from the operating system and services running on your servers.

Centralize security events for interactive investigation in ready-to-go visualizations.

Figure 22: Visualizations and Graphs on Dashboard

On confirming the stack is started, navigate to Kibana at http://<ip-address>:5601. To see visualization and graphs, go to Dashboards -> [Metricbeat System] Host overview ECS.

Dashboards

The toll named Metricbeat collects the following data: filesystem per host, system overview, CPU, filesystem, memory, network, overview, processes.

Technical notes

The following summarizes some important technical considerations:

- The Elasticsearch instances uses a named volume esdata for data persistence between restarts. It exposes HTTP port 9200 for communication with other containers.
- Environment variable defaults can be found in the file .env.
- The Elasticsearch container has its memory limited to 1g. This can be adjusted using the environment parameter ES_MEM_LIMIT. Elasticsearch has a heap size of 1g. This can be adjusted through the environment variable ES_JVM_HEAP and should be set to 50% of the ES_MEM_LIMIT. Users may wish to adjust this value on smaller machines.
- The Elasticsearch password can be set via the environment variable ES_PASSWORD. This sets the password for the Elastic and Kibana user.
- The Kibana container exposes the port 5601.
- All configuration files can be found in the extracted folder ./config.
- The Metricbeat container mounts both /proc and /sys/fs/cgroup on Linux. This allows Metricbeat to use the system module report on disk, memory, network and cpu of the host.
- On systems with POSIX file permissions, all Beats configuration files are subject to ownership and file permission checks. The purpose of these checks is to prevent unauthorized users from providing or modifying configurations that are run by the Beat. The owner of the configuration file must be either root or the user who is executing the Beat process. The permissions on the file must disallow writes by anyone other than the owner. As we mount our configurations from the host, where the user is likely different than that used to run the container and the beat process, we disable this check for all beats with -strict.perms=false.

Customizing the Stack

With respect to the current example, we have provided a few simple entry points for customization:

- The example includes an .env file listing environment variables which alter the behaviour of the stack. These environment variables allow the user to change:
  - ELASTIC_VERSION - the Elastic Stack version (default 7.2.0)

  - ES_PASSWORD - the password used for authentication with the elastic user. This password is applied for all system users i.e. kibana and logstash_system. Defaults to changeme.
  - DEFAULT_INDEX_PATTERN - The index pattern used as the default in Kibana. Defaults to metricbeat-*.
  - ES_MEM_LIMIT - The memory limit used for the Elasticsearch container. Defaults to 1g. Consider reducing for smaller machines.
  - ES_JVM_HEAP - The Elasticsearch JVM heap size. Defaults to 1024m and should be set to half of the ES_MEM_LIMIT.
- Modules and Configuration - All configuration to the containers is provided through a mounted ./config directory. Where possible, this exploits the dynamic configuration loading capabilities of Beats. For example, an additional module could be added by simply adding a file to the directory ./config/beats/metricbeat/modules.d/ in the required format.

Shutting down the stack

The following command will exit the containers and ensure they are shut down gracefully.

```
$ sudo docker-compose -f docker-compose-linux.yml stop
```

To remove all containers, including their mounted named volumes, use the following

command:

```
$ sudo docker-compose -f docker-compose-linux.yml down -v
```