6-11-2017

Design and Implementation of a Direct/Indirect Hybrid Trust Model for Secure Authentication in a Mobile Ad Hoc Network

Immanuel Amirtharaj
Santa Clara University, iamirtharaj@scu.edu

Eric Bonilla
Santa Clara University, ebonilla@scu.edu

Parker Newton
Santa Clara University, pnewton@scu.edu

SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

Date: June 6, 2017

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Immanual Amirtharaj
Eric Bonilla
Parker Newton

ENTITLED

Design and Implementation of a Direct/Indirect Hybrid Trust Model for Secure Authentication in a Mobile Ad Hoc Network

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREES OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING
BACHELOR OF SCIENCE IN WEB DESIGN AND ENGINEERING
BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING AND MATHEMATICS

yuhong Xin

Thesis Advisor

Department Chair

Design and Implementation of a Direct/Indirect Hybrid Trust Model for Secure Authentication in a Mobile Ad Hoc Network

by

Immanuel Amirtharaj
Eric Bonilla
Parker Newton

Submitted in partial fulfillment of the requirements
for the degrees of
Bachelor of Science in Computer Science and Engineering
Bachelor of Science in Web Design and Engineering
Bachelor of Science in Computer Science and Engineering and Mathematics
School of Engineering
Santa Clara University

Santa Clara, California
June 11, 2017

Design and Implementation of a Direct/Indirect Hybrid Trust Model for Secure Authentication in a Mobile Ad Hoc Network

Immanuel Amirtharaj
Eric Bonilla
Parker Newton

Department of Computer Engineering
Santa Clara University
June 11, 2017

ABSTRACT

We investigated the problem of cryptographic key authentication in a mobile ad hoc network (MANET). Using the theory of digital trust, we propose an authentication scheme for MANETs that includes a hybrid trust model between the direct and indirect approaches. Our hybrid trust model supplies trust data to a decentralized web of trust in order to authenticate nodes in a MANET. We ran some simulations of our authentication scheme to verify its security and investigate potential trust threshold values. Also, we designed and implemented a proof-of-concept iOS application that implements our authentication scheme. Some future work includes investing several implications of the mobility aspect of MANETs on trust management, such as the maximum levels of trust concatenation.

Table of Contents

List of Figures

Chapter 1

Introduction

Many modern technologies such as Bitcoin, Spotify, and various file-sharing services are based on a distributed network architecture. A distributed network is a network that partitions jobs among the nodes. Specifically, a mobile ad hoc network (MANET) is a type of distributed network that contains small-scale personal computers and embedded systems. A chief cryptographic concern in the design of MANETs is the ad hoc authentication of the nodes. In this context, by the authentication of nodes in a MANET, we mean key authentication without a centralized certificate authority. We propose introducing the concept of digital trust to authentication in a distributed system. By building on past work in the area of trust theory, we designed an authentication scheme that implements a direct/indirect hybrid trust model to supply trust data to a web of trust.

An existing solution is the PGP web of trust scheme. However, PGP implements a system of qualitative trust levels. By providing a system of quantitative trust values established through algorithms that conform to basic axioms of trust theory, our solution offers trust management with increased granularity and accuracy. Our authentication scheme also addresses several attacks on MANETs, including the Sybil and Newcomer attacks.

Our project consists of theoretical research, simulations, and software design and implementation. We designed and implemented a proof-of-concept mobile iOS application that demonstrates our authentication scheme. The mobile application reads mocked network data from a web back-end component, and attempts to authenticate and send a message containing a string to another node. A web front-end component displays a visualization of the mocked network web of trust graph.

Chapter 2 introduces the background of public key cryptography, the theory of groups, rings, fields, and elliptic curves, several prominent public key cryptosystems we use in our project, and webs of trust. Chapter 3 addresses current innovations in digital trust theory. Chapters 4 and 5 address our contributions of our project and the results we collected. Chapter 6 discusses the software

design and architecture of our iOS and web proof-of-concept application. We address several societal implications of our project in Chapter 7. Finally, we provide some concluding remarks in Chapter 8.

Chapter 2

Introduction to Cryptographic Authentication

2.1 Introduction to Public Key Cryptography

Public key cryptography is a major area of cryptography that has many applications to topics such as symmetric key agreement and authentication. This paper assumes the reader is familiar with the concepts of symmetric key cryptography and hash algorithms.

Definition 2.1. ***Public key cryptography*** *is a form of cryptography in which each user has their own set of public and private keys.*

Definition 2.2. *A* ***public key*** *is a cryptographic key that is published. A* ***private key*** *is a cryptographic key that is secret and known only by the user to which it belongs. We say that such a user is the* ***holder*** *of the public and private keys.*

Axiom 2.3. *Let A be an entity. Then, the holder of A's private key is A.*

This axiom may seem trivial, but nonetheless is an underlying principle of cryptography. Often, we discuss verifying that the holder of A's public key is indeed the holder of A's private key. But, how do we know that the only holder of A's private key is in fact A? We assume that is the case; otherwise, many of our results in cryptography would also not hold. Non-repudiation is an area of cryptography which addresses the scenario of some user purposefully leaking his or her private key in order to claim fraud and nullify a legally binding document. Now, we will discuss the concept of symmetric key agreement.

Definition 2.4. *Let A and B be two entities that wish to exchange cipher text encrypted with a symmetric key cryptosystem. The process of A and B determining the same symmetric key on-demand and without having to be in the same location is called* ***symmetric key agreement****.*

There are two important factors to address with symmetric key agreement. The first is the on-demand agreement on a symmetric key. Historically, a symmetric key had to be agreed on by two entities before the communication could be transmitted. However, the correct application of

symmetric key agreement ensures that there does not need to be a previous communication between the two entities. That is, the symmetric key agreement occurs ad hoc, or on-demand. The second factor is that with public key cryptography, the two entities do not even have to be in the same location to agree on a symmetric key.

Let A and B be two entities that wish to agree on a symmetric key. The general process is that A will compute a symmetric key, and then encrypt the symmetric key with B's public key. So, now only B can decrypt the encrypted message, because the message can only be decrypted with B's private key.

Now that the two entities have successfully agreed on a symmetric key and exchanged encrypted messages, we consider the problem of verifying the authenticity of the messages and the identities of the users.

Definition 2.5. *Let A and B be two entities, and suppose A sends an encrypted message to B. The process of B verifying that the message originated from the holder of A's private key is called* ***message authentication****. The process of B verifying that the holder of A's public key is the same entity as the holder of A's private key is called* ***key authentication****.*

Authentication is an extremely important concept in cryptography that provides secure communication. Correct authentication of messages ans users' keys ensures that a malicious third party has not not impersonated another user or forged public and private keys. The authenticity of messages and users' keys are verified with a mechanism called a digital signature.

Definition 2.6. *A* ***digital signature*** *is a cryptographic mechanism that is used to verify that a message originated from some entity.*

Let A and B be two entities, and suppose A encrypts a message for B. A computes a hash of the message, and encrypts the resulting value with A's private key. B then independently computes a hash of the decrypted message, and then decrypts the digital signature with A's public key. B compares the value of the hash to that of the decrypted digital signature, and if the two values are equal, and since by axiom 2.3 only A could have encrypted the digital signature, then B is assured of the integrity and authenticity of the message.

The next natural extension of the concept of authentication addresses how public and private keys are created and managed. A user's public keys are published in a document called a certificate.

Definition 2.7. *A* ***certificate*** *is a document that associates a set of public keys with a set of private keys.*

In centralized network architecture, a certificate is issued by a trusted authority called a certificate authority (CA). The CA creates the public and private keys for that user, and then encrypts a hash of the document with the CA's private key. The resulting signature is then published on the certificate.

Let A and B be entities, and C be a CA. Suppose C creates a certificate for A, and that some user B wishes to verify the authenticity of A's certificate. B decrypts the digital signature with C's public key, and then computes the hash of the document. B then compares the value of the decrypted digital signature to that of the hash. If the values are equal, then B is assured of the integrity and authenticity of A's certificate. Because the certificate originated from C, a trusted CA, then B is assured that the holder of A's private key is the holder of A's public key. By axiom 2.3, B is assured that A is the holder of A's public key.

The theoretical background of public key cryptography algorithms involves a great deal of abstract algebra and number theory. Specifically, group theory, ring theory, field theory, and Galois theory are several areas of abstract algebra with numerous applications to public key cryptography.

2.2 Introduction to Group Theory

Group theory is an area of abstract algebra that includes the study of the algebraic structure of groups and results that generalize properties of groups. Group theory can be applied to study many objects in mathematics. Many of these objects form the theoretical basis of public key cryptography. In this section, we will present a brief overview of some introductory group theory material with few proofs. The following group, ring, and field theory results are presented in Judson [4] and Long [5].

Definition 2.8. *A **group** is a set G and a binary operation $* : G \times G \to G$ defined on G with the following properties:*

1. $\exists\ e \in G$ *such that* $\forall g \in G,\ g * e = e * g = g$. *We say that e is the identity element of G.*
2. $\forall g \in G,\ \exists\ g^{-1} \in G$ *such that* $g * g^{-1} = g^{-1} * g = e$. *We say that g^{-1} is the inverse element of g in G.*
3. $*$ *is associative.*

We say that G is a group under $*$.

If G is an abstract group, then we simply denote the binary operation $*$ defined on G by multiplication. An immediate consequence of the definition of the binary operation on G is that $\forall g, g' \in G,\ gg' \in G$. We say that G is closed under the binary operation. $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$, any ring A, any field K, and the set of $n \times n$ matrices with entries in some field K (denoted $Mat(n, K)$) are all groups under addition. $\mathbb{Q} \setminus \{0\}$, $\mathbb{R} \setminus \{0\}$, $\mathbb{Z}_p \setminus \{0\}$ for some prime integer p, $K \setminus \{0\}$ for some field K, and the set of invertible $n \times n$ matrices with entries in some field K (denoted $GL(n, K)$) are all groups under multiplication. The set of permutations of the set $\{1, 2, \ldots, n\}$ (denoted S_n, the symmetric group), the set of even permutations of the set $\{1, 2, \ldots, n\}$ (denoted A_n, the alternating group), the set of isometries of a regular n-gon (denoted D_n, the dihedral group),

the set of rotations of a regular cube, the set of automorphisms of some field K (denoted $Aut(K)$), and the set of automorphisms of a normal extension field L of K extending the identity map of K (denoted $G(L/K)$, the Galois group of L over K) are all groups under function composition.

Definition 2.9. *Let G be a group. We say the cardinality of G, denoted $|G|$, is the* ***order*** *of G.*

Definition 2.10. *Let G be a group, and $H \subseteq G$. If H is a group, then we say H is a* ***subgroup*** *of G. We write $H \leq G$.*

Definition 2.11. *Let G be a group, and $a \in G$. We say that the set $\langle a \rangle = \{..., a^{-3}, a^{-2}, a^{-1}, a^0 = e, a, a^2, a^3, ...\} = \{a^m : m \in \mathbb{Z}_{\geq 0}\}$ is the* ***subgroup of G generated by a****.*

Definition 2.12. *Let G be a group. If $\exists\ a \in G$ such that $G = \langle a \rangle$, then we say that G is a* ***cyclic group****. We also say that a is a* ***generator*** *of G.*

Proposition 2.13. *Let G be a group, $H \leq G$, and $\sim$: $G \times G$ by $a \sim b$ iff $b^{-1}a \in H$ be a relation of G. Then, $\sim$ is an equivalence relation of G.*

Proof. Let $a, b, c \in G$. $a^{-1}a = e \in H$, by definition, and so $a \sim a$.
Suppose $a \sim b$. Then, $b^{-1}a \in H$. So, $\exists\ h \in H$ such that $b^{-1}a = h$. We have

$$\begin{aligned} b^{-1}a &= h \\ a &= bh \\ ah^{-1} &= b \\ h^{-1} &= a^{-1}b. \end{aligned}$$

Since $h^{-1} \in H$, then $b \sim a$.
Now, suppose $a \sim b$ and $b \sim c$. Then, $b^{-1}a \in H$ and $c^{-1}b \in H$. So, $\exists\ h_1, h_2 \in H$ such that $b^{-1}a = h_1$ and $c^{-1}b = h_2$. We have

$$\begin{aligned} c^{-1}b &= h_2 \\ b &= ch_2 \\ b^{-1} &= h_2^{-1}c^{-1}. \end{aligned}$$

Then,

$$\begin{aligned} b^{-1}a &= h_1 \\ h_2^{-1}c^{-1}a &= h_1 \\ c^{-1}a &= h_2h_1. \end{aligned}$$

Since $h_2h_1 \in H$, then $a \sim c$.
We have shown that $\sim$ is reflexive, symmetric, and transitive, and hence $\sim$ is an equivalence relation. ■

Let G be a group, and $a \in G$. By proposition 2.13, $a \sim b$ iff $b^{-1}a \in H$ is an equivalence relation of G. Consider the equivalence class $[a]$ under $\sim$. Then, $[a] = \{b \in G : a^{-1}b \in H\}$.

Definition 2.14. *Let G be a group, $H \leq G$, and $a \in G$. We say the set $aH = \{ah : h \in H\}$ is a **left coset** of H in G. Similarly, we say the set $Ha = \{ha : h \in H\}$ is a **right coset** of H in G.*

Proposition 2.15. *Let G be a group, $H \leq G$, and $a, b \in G$. Then, $a \in bH$ iff $b^{-1}a \in H$.*

Proof. Suppose $a \in bH$. Then, $\exists\, bh \in H,\ h \in H$, such that $a = bh$. We have

$$\begin{aligned} a &= bh \\ b^{-1}a &= h. \end{aligned}$$

Since $h \in H$, then $b^{-1}a \in H$.
Conversely, suppose $b^{-1}a \in H$. Then, $\exists\, h' \in H$ such that $b^{-1}a = h'$. We have

$$\begin{aligned} b^{-1}a &= h' \\ a &= bh'. \end{aligned}$$

Since $h' \in H$, then $a = bh' \in bH$. ■

Now, we have

$$[a] = \{b \in G : a^{-1}b \in H\}$$

$$[a] = \{b \in G : b \in aH\}, \text{ by proposition 2.15}$$

$$[a] = aH.$$

So, the equivalence classes of $\sim$ are actually the left cosets of H in G.

Lemma 2.16. *Let G be a group, and $H \leq G$. Then, the set $\{aH : a \in G\}$ partitions G.*

Proof. Let $\sim$: $G \times G$ by $a \sim b$ iff $b^{-1}a \in H$ be a relation of G. By proposition 2.13, $\sim$ is an equivalence relation of G. We know that the equivalence classes of G under $\sim$ partition G. We showed that the equivalence classes of G under $\sim$ are the left cosets of H. Hence, $\{aH : a \in G\}$ partitions G. ■

Lemma 2.17. *Let G be a group, and $H \leq G$. Then, $\forall a \in G$, $|aH| = |H|$.*

Proof. Consider some $a \in H$, and define the map $\phi : H \to aH$ by $\phi(h) = ah$. We will show that ϕ is a bijection, and hence $|aH| = |H|$.
Consider $h, h' \in H$ such that $\phi(h) = \phi(h')$. Then, we have

$$\begin{aligned} ah &= ah' \\ h &= h', \text{ by left cancellation.} \end{aligned}$$

So, ϕ is an injection.
Now, consider some $ag \in aH,\ g \in H$. Then, $a^{-1}ag = g \in H$. So, $\phi(g) = ag$, and so ϕ is a surjection.
We have shown that ϕ is a bijection, and so the conclusion follows. ■

Now, we will prove a marvelous result in group theory which is referred to as Lagrange's Theorem.

Definition 2.18. *Let $a, b \in \mathbb{Z}$ such that $a \neq 0$. If $\exists\ m \in \mathbb{Z}$ such that $b = am$, then we say that a divides b. We write $a \mid b$.*

Theorem 2.19. *Let G be a finite group, and $H \leq G$. Then, $|H| \mid |G|$.*

Proof. Let n = $|G|$, m = $|H|$, S = $\{aH : a \in G\}$, and $l = |S|$. By lemma 2.16, S partitions G. We have

$$n = \sum_{i=1}^{l} |a_i H|, \text{ by definition of a set partition.}$$
$$n = \sum_{i=1}^{l} |H|, \text{ by lemma 2.17.}$$
$$n = \sum_{i=1}^{l} m$$
$$n = m \sum_{i=1}^{l} 1$$
$$n = ml.$$

Hence, $m \mid n$. ■

Corollary 2.20. *Let G be a finite group, and $a \in G$. Then, $ord(a) \mid |G|$.*

Proof. By definition, $ord(a) = |\langle a \rangle|$. Since $\langle a \rangle \leq G$, then by Theorem 2.19, $|\langle a \rangle| \mid |G|$, and hence $ord(a) \mid |G|$. ■

Definition 2.21. *Let G be a group and $H \leq G$. If $\forall a \in G$, $aH = Ha$, then we say that H is a **normal subgroup** of H. We write $H \triangleleft G$.*

Definition 2.22. *Let G be a group, and $H \triangleleft G$. Then, we say the set $G/H = \{aH : a \in G\}$ is the **quotient group** of H in G.*

2.3 Introduction to Ring Theory and Field Theory

Ring theory and field theory are two areas of abstract algebra that involve the study of the algebraic structures of rings and fields, respectively. Much of the theory of fields is based on many important results in ring theory. Additionally, field theory includes the study of many other important topics such as field extensions, finite fields, separability, splitting fields, algebraic closures, field isomorphisms, embeddings, and automorphisms, and normal extensions. Field theory sets the theoretical foundation for an extremely important area of abstract algebra called Galois theory. Galois theory proves important results about normal extension fields, and connects field theory to group theory in a very elegant and beautiful way. Ring theory, field theory, and Galois theory have many important applications to public key cryptography. We will begin with a brief overview of the theory of rings

and fields, and then discuss some important results about finite fields that will allow us to construct several significant public key cryptosystems.

Definition 2.23. *A **ring** is a set A with the binary operations of addition and multiplication defined on A, and with the following properties:*

1. $\exists\ 0\ \in\ A$ *such that* $\forall a\ \in\ A,\ a+0=0+a=a$. *We say that 0 is the additive identity of A.*
2. $\exists\ 1\ \in\ A$ *such that* $\forall a\ \in\ A,\ a\cdot 1=1\cdot a=a$. *We say that 1 is the multiplicative identity of A.*
3. $\forall a\ \in\ A,\ \exists\ -a\ \in\ A$ *such that* $a+(-a)=-a+a=0$. *We say that $-a$ is the additive inverse of A.*
4. *Addition and Multiplication are associative.*
5. *Addition is commutative.*
6. *Multiplication is distributive over addition.*

Definition 2.24. *Let A be a ring. If the multiplication is commutative, then we say that A is a **commutative ring**.*

Definition 2.25. *Let A be a commutative ring, and $I\subseteq A$. Then, we say I is an **ideal** of A if I has the following properties:*

1. $0\ \in\ I$.
2. $\forall a,b\ \in\ I,\ a+b\ \in\ I$.
3. $\forall a\ \in\ I, \forall m\ \in\ A,\ ma\ \in\ I$.

Definition 2.26. *Let A be a commutative ring and I be an ideal of A. If I is of the form $(a)=\{ma : m\ \in\ A\}$, for some $a\ \in\ I$, then we say that I is a **principal ideal**.*

Definition 2.27. *Let A be a commutative ring and I be an ideal of A. We say that I is a **maximal ideal** of A if for every ideal J of A such that $I\subseteq J\subseteq A$, then either $I=J$ or $J=A$.*

Definition 2.28. *Let A be a commutative ring, $a\ \in\ A$, and I be an ideal of A. Then, the set $a+I=\{a+r : r\ \in\ I\}$ is a **coset** of I in A.*

Definition 2.29. *Let Let A be a commutative ring and I be an ideal of A. Then, the set $A/I=\{a+I : a\ \in\ A\}$ is the **quotient ring** of I in A.*

It can be easily shown that the quotient ring A/I is in fact a ring.

Definition 2.30. *Let A be a commutative ring and $a,b\ \in\ A$. If $\exists\ m\ \in\ A$ such that $b=am$, then we say that a **divides** b. We write $a|b$.*

Definition 2.31. *Let A be a commutative ring and $a\ \in\ A$. If $a|1$, then we say that a is a **unit**.*

We denote the subset of units of a commutative ring A by A^*. It can be easily shown that A^* is a group under multiplication, and so we call A^* the group of units of A.

Definition 2.32. *Let A be a commutative ring, and $a,b\ \in\ A$. If $a|b$ and $b|a$, then we say that a and b are **associates**.*

Definition 2.33. *Let A be a commutative ring, and $a,b\ \in\ A$ such that $a\neq 0$ and $b\neq 0$. If $ab=0$, then we say that a is a **zero divisor**.*

Definition 2.34. *Let D be a commutative ring. We say D is an* ***integral domain*** *if D has the following properties:*

1. $0 \neq 1$. *That is, the additive identity does not equal the multiplicative identity.*
2. *D has no zero divisors. That is, $\forall a, b \in A$ such that $a \neq 0$ and $b \neq 0$, then $ab \neq 0$.*

Definition 2.35. *Let D be an integral domain. If every ideal of D is a principal ideal, then we say that D is a* ***principal ideal domain*** *(PID).*

Definition 2.36. *Let D be an integral domain and $p \in D$. We say that p is* ***prime*** *if p has the following properties:*

1. *p is neither zero nor a unit.*
2. *$\forall a, b \in D$ such that $p|ab$, then either $p|a$ or $p|b$.*

Definition 2.37. *Let D be an integral domain and $p \in D$. We say that p is* ***irreducible*** *if p has the following properties:*

1. *p is neither zero nor a unit.*
2. *$\forall a, b \in D$ such that $p = ab$, then either a or b is an associate of p, and the other is a unit.*

The following theorem is an extremely important result in ring theory that has a multitude of applications to abstract algebra. We will not prove this result, since the reader is not assumed to possess a sufficient level of mathematical sophistication in order to comprehend the proof.

Theorem 2.38. *Let D be a PID and $p \in D$. Then, TFAE:*

1. *p is prime.*
2. *p is irreducible.*
3. *(p) is a maximal ideal of D.*
4. *$D/(p)$ is a field.*
5. *$D/(p)$ is an integral domain.*

We will soon see an important application of this result when we prove that $\mathbb{Z}_p$ is a finite field for each prime integer p. Next, the concept of ring homomorphisms, isomorphisms, and automorphisms is central to the study of ring theory. However, we will not go into much depth of those topics here since our discussion is tailored towards the theory necessary to understand public key cryptography. So, we will simply present their definitions below.

Definition 2.39. *Let A, B be commutative rings and $\phi : A \to B$ be a mapping from A to B. Then, we say that ϕ is a* ***ring homomorphism*** *if ϕ has the following properties:*

1. *$\forall a_1, a_2 \in A$, $\phi(a_1 + a_2) = \phi(a_1) + \phi(a_2)$.*
2. *$\forall a_1, a_2 \in A$, $\phi(a_1 a_2) = \phi(a_1)\phi(a_2)$.*

We say that ϕ preserves the operations of addition and multiplication.

Definition 2.40. *Let A, B be commutative rings and $\phi : A \to B$ be a ring homomorphism. If ϕ is a bijection, then we say that ϕ is an* ***isomorphism of rings****. We write $A \cong B$.*

Definition 2.41. *Let A be a commutative ring and $\phi : A \to A$ be an isomorphism of rings. Then, we say that ϕ is an **automorphism** of A.*

Now, we will introduce some basic concepts of field theory.

Definition 2.42. *A **field** is a commutative ring K such that $\forall a \in K$ with $a \neq 0$, then $\exists\, a^{-1} \in K$ such that $aa^{-1} = a^{-1}a = 1$. We say that a^{-1} is the multiplicative inverse of a.*

Definition 2.43. *Let K be a field with a finite number of elements. Then, we say that K is a **finite field**.*

Now, we will prove that the commutative ring $\mathbb{Z}_p$, for each prime integer p, is in fact a field. First, we note the following results which we will not prove.

Proposition 2.44. *$\mathbb{Z}$ is a PID.*

Proposition 2.45. *Let $n \in \mathbb{Z}_{\geq 2}$. Then, $\mathbb{Z}/(n) \cong \mathbb{Z}_n$.*

Now that we have those previous results, we will prove that $\mathbb{Z}_p$ is a field.

Proposition 2.46. *Let $p \in \mathbb{Z}$ be prime. Then, $\mathbb{Z}_p$ is a field.*

Proof. By proposition 2.44, $\mathbb{Z}$ is a PID. Since $p \in \mathbb{Z}$ is prime, then by theorem 2.38, $\mathbb{Z}/(p)$ is a field. Since by proposition 2.45, $\mathbb{Z}/(p) \cong \mathbb{Z}_p$, then $\mathbb{Z}_p$ too is a field. ■

It can be shown that every finite field has size p^n, for some positive prime integer p and some positive integer n. Furthermore, every finite field of size p^n is isomorphic to $\mathbb{Z}_p[x]/(x^n + \ldots + 1)$, where $(x^n + \ldots + 1)$ denotes the ideal generated by an irreducible polynomial of degree n over $\mathbb{Z}_p$. Hence, there exists exactly 1 finite field of size p^n isomorphic to $\mathbb{Z}_p[x]/(x^n + \ldots + 1)$. We denote the finite field of size p^n by $\mathbb{F}_p[x]/(x^n + \ldots + 1)$. With much more field theory that is beyond the scope of this discussion, it can be shown that $\mathbb{F}_p[x]/(x^n + \ldots + 1)$ is a finite, separable, splitting, and hence normal (or Galois), extension field of $\mathbb{F}_p$. As such, we also denote $\mathbb{F}_p[x]/(x^n + \ldots + 1)$ by $\mathbb{F}_{p^n}$ or $GF(p^n)$, where GF stands for Galois field.

2.4 Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) is a form of public key cryptography that is based on the algebraic properties of a type of geometric curve called an elliptic curve (EC). ECC has applications to symmetric key agreement and digital signature schemes. We'll see that ECC is popular because it allows for shorter keys than alternative public key cryptosystems for the same level of security.

Definition 2.47. *Let K be a field. An **elliptic curve** is a geometric curve of the form $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, each $a_i \in K$, and a 0-point located at infinity.*

We can define the operation of addition on the points on an elliptic curve over some field K. Let K be a field, $E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ be an E.C. over K, $P = (x_1, y_1), Q =$

$(x_2, y_2) \in E(K)$. We will find $P + Q \in E(K)$. The rule is that the points on any line that intersects E sum to 0.

Case I: $P \neq Q$. The line l through P and Q is determined. Then, l intersects E at a third point $S = (x_3, y_3) \in E(K)$, and so

$$P + Q + S = 0$$
$$P + Q = -S.$$

Then,

$$m = \frac{y_2 - y_1}{x_2 - x_1}.$$

$$l : y - y_1 = m(x - x_1)y = mx + y_1 - mx_1. \tag{2.1}$$

We have

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$
$$(mx + y_1 - mx_1)^2 + a_1x(mx + y_1 - mx_1) + a_3(mx + y_1 - mx_1) = x^3 + a_2x^2 + a_4x + a_6$$
$$m^2x^2 + 2my_1x - 2m^2x_1x + y_1^2 - 2y_1mx_1 + m^2x_1^2 + a_1mx^2 + a_1y_1x - a_1mx_1 + a_3mx + a_3y - a_3mx_1$$
$$= x^3 + a_2x^2 + a_4x + a_6$$

$$x^3 + x^2(-m - a_1m + a_2) + x(-2my_1 + 2m^2x_1 - a_1y_1 - a_3m + a_4) +$$
$$(-y_1^2 + 2y_1mx_1 - m^2x_1^2 + a_1mx_1 - a_3y + a_3y + a_3mx_1 + a_6) = 0 \tag{2.2}$$

Since we know that x_1 and x_2 are roots of (2.2), then we know that $(x - x_1)(x - x_2) = x^2 + x(-x_1 - x_2) + x_1x_2 \mid (2.2)$. Polynomial division shows that

$$\frac{(2.2)}{x^2 + x(-x_1 - x_2) + x_1x_2} = x - m^2 + a_1m + a_2 + x_1 + x_2 = x - (m^2 - a_1m - a_2 - x_1 - x_2).$$

So, we have

$$x_3 = m^2 - a_1 m - a_2 - x_1 - x_2. \tag{2.3}$$

Substituting (2.3) into (2.1), we get

$$y_3 = m(m^2 - a_1 m - a_2 - x_1 - x_2) + y_1 - m x_1$$
$$y_3 = m^3 + m^2(-a_1) + m(-a_2 - 2x_1 - x_2) + y_1. \tag{2.4}$$

Now, we wish to compute $P+Q = -S = -(x_3, y_3)$. By examining E, and applying the quadratic formula to solve for y, we have

$$y = \frac{-(a_1 x + a_3) \pm \sqrt{(a_1 x)^2 - 4(-x^3 - a_2 x^2 - a_4 x_3 - a_6)}}{2}. \tag{2.5}$$

By substituting x_3 into (2.5), we get two values for y. Call these values $y^{'}$ and $y^{''}$. WLOG, let $y^{'} = y_3$. Then, $S = (x_3, y^{'})$, and so $P + Q = -S = (x_3, y^{''})$.

Case II: $P = Q$. Then, we consider the line tangent to E at P. It must intersect E at a third point $T = (x_4, y_4) \in E(K)$. Then, we have

$$P + P + T = 0$$
$$2P = -T.$$

We will find an equation for the line tangent to E at P. Using implicit differentiation, we have

$$E : y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6$$
$$2ydy + a_1 ydx + a_1 xdy + a_3 dy = 3x^2 dx + 2a_2 xdx + a_4 dx$$
$$\frac{dy}{dx} = \frac{3x^2 + 2a_2 x + a_4 - a_1 y}{2y + a_1 x + a_3}.$$
$$m = \frac{dy}{dx}\Big|_{(x_1, y_1)} = \frac{3x_1^2 + 2a_2 x_1 + a_4 - a_1 y_1}{2y_1 + a_1 x_1 + a_3}. \tag{2.6}$$

Then,

$$y - y_1 = x(x - x_1)y = mx - mx_1 + y_1. \tag{2.7}$$

By substituting (2.6) into (2.7), we have the line tangent to E at P. Similarly, by substituting (2.7) into E and dividing by $(x - x_1)(x - x_4) = x^2 + x(-x_1 - x_4) + x_1 x_4$, we get

$$x_4 = -2x_1 - a_2 + m^2 + a_1 m. \tag{2.8}$$

By substituting (2.8) into (2.7), we get

$$y_4 = m^3 + m^2 a_1 + m(-3x_1 - a_2) + y_1. \tag{2.9}$$

So, $T = (x_4, y_4)$. Then, we can use a similar method to case I to find $2P = -T = -(x_4, y_4)$.

In general, let $R \in E(K)$ and $n \in \mathbb{Z}_{>0}$. Then, we can compute $nR = (n-1)R + R$.

We can show that $E(K)$ is a group under point addition, with the 0-point as the group identity, and $-(x, y)$ as the inverse element $\forall (x, y) \in E(K)$. With much more abstract algebra and algebraic geometry, we can show that $E(\mathbb{F}_p) = \langle S \rangle$, for some finite field $\mathbb{F}_p$ where p is a positive prime integer, and some $S \in E(\mathbb{F}_p)$.

We will state the Elliptic Curve Discrete Logarithm Problem (ECDLP) without proof. This result beautifully combines the theory of groups, fields, and elliptic curves, and will allow us to construct several important public key cryptosystems.

Proposition 2.48. *(Elliptic Curve Discrete Logarithm Problem). Let E be an E.C. over $\mathbb{F}_p$, $p \approx 2^{256} \in \mathbb{Z}_{>0}$ is prime, $S \in E(\mathbb{F}_p)$ generates $E(\mathbb{F}_p)$, and $T = nS \in E(\mathbb{F}_p)$, $n \in \mathbb{Z}$. Then, it is impractical to determine n from T.*

2.5 Elliptic Curve Diffie-Hellman

The Elliptic Curve Diffie-Hellman (ECDH) algorithm is a public key cryptosystem used for symmetric key agreement. It is an ECC analogue of the Finite Field Diffie-Hellman algorithm. Alice (A) and Bob (B) wish to agree on a symmetric key for a symmetric key cryptosystem. A chooses the EC E over a field $\mathbb{F}_p$, $p \in \mathbb{Z}_{>=2}$ is prime, $S \in E(\mathbb{F}_p)$ such that $E(\mathbb{F}_p) = \langle S \rangle$, and $a_A \in \mathbb{Z}$ such that $1 < a_A < |E(\mathbb{F}_p)|$. a_A is A's private key. A then computes her public key $a_A S = T \in E(\mathbb{F}_p)$. A sends E, p, S, and T to B. B chooses his private key $a_B \in \mathbb{Z}$ such that $1 < a_B < |E(\mathbb{F}_p)|$, and computes his public key $a_B S = U \in E(\mathbb{F}_p)$. B sends U to A. Then, A computes $a_A U$ and B computes $a_B T$. Since $\mathbb{Z}$ is a commutative ring, then the multiplication is commutative for $\mathbb{Z}$, and so $a_A a_B = a_B a_A$. So, $a_A U = a_A a_B S = a_B a_A S = a_B T$, meaning A and B have now agreed on the symmetric key $a_A a_B S \in E(\mathbb{F}_p)$. If Eve, a third party, intercepts either of the public keys $T = a_A S$ or $U = a_B S$, then by Proposition 2.48, Eve cannot compute either of the private keys a_A or a_B, and hence cannot determine the shared symmetric key $a_A a_B S$.

2.6 Elliptic Curve Digital Signature Algorithm

The Elliptic Curve Digital Signature Algorithm (ECDSA) is a public key cryptosystem used for digital signatures. It is an ECC analogue of the Finite Field Digital Signature Algorithm. First, we

will prove a simple result that is important to the ECDSA.

Proposition 2.49. *Let G be a finite group, $n = |G| \in \mathbb{Z}_{>0}$, $a \in G$, and $l, m \in \mathbb{Z}$. If $l \equiv m (mod\ n)$, then $a^l = a^m$.*

Proof. Since $l \equiv m (\text{mod } n)$, then $n \mid l - m$, and so $l - m = nk$, $k \in \mathbb{Z}$. Then,

$$a^{l-m} = a^{nk}.$$

Let $r = ord(a) \in \mathbb{Z}_{>0}$. By Corollary 2.20, $r \mid n$, and so $n = rs$, $s \in \mathbb{Z}$. We have

$$\begin{aligned} a^{l-m} &= a^{nk} \\ a^{l-m} &= a^{rsk} \\ a^{l-m} &= (a^r)^{sk} \\ a^{l-m} &= e^{sk}, \text{ since } r = ord(a). \\ a^{l-m} &= e \\ a^l a^{-m} &= e \\ a^l a^{-m} a^m &= e a^m \\ a^l &= a^m. \end{aligned}$$

■

Now, we will describe the ECDSA. Alice (A) sends Bob (B) a message, and B wants to be assured of the integrity and authenticity of the message. That is, B wants to verify that the message has not been tampered with by Eve during transmission, and that A was the entity that sent the message. A computes the hash $H \in \mathbb{Z}$ of the message. Then, A chooses the EC E over a field $\mathbb{F}_p$, $p \in \mathbb{Z}_{>=2}$ is prime, $S \in E(\mathbb{F}_p)$ such that $E(\mathbb{F}_p) = \langle S \rangle$, and $a_A \in \mathbb{Z}$ such that $1 < a_A < |E(\mathbb{F}_p)|$. a_A is A's private key. A then computes her public key $a_A S = (x_A, y_A) \in E(\mathbb{F}_p)$. Let $n = |E(\mathbb{F}_p)|$. Next, A chooses a random session key $k \in \mathbb{Z}$ such that $1 << k << n$, and computes $kS = (x_k, y_k)$. A solves the linear modular equation

$$kx \equiv H + a_A x_k (\text{mod } n) \tag{2.10}$$

for x. A sends E, p, S, her public key $a_A S$, and her digital signature $\{kS, x\}$ to B. B computes the hash $H^{'} \in \mathbb{Z}$ of the initial message he received from A. Since $kx \equiv H + a_A x_k (\text{mod } n)$, then by Proposition 2.49, $kxS = (H + a_A x_k)S = HS + a_A x_k S$. Let $HS = (x_H, y_H)$. B computes $H^{'}S = (x_{H'}, y_{H'})$. Since, $kxS = xkS = x(x_k, y_k)$, and $a_A x_k S = x_k a_A S = x_k(x_A, y_A)$, then B computes xx_k and $x_k x_A$, and verifies that $xx_k = x_{H'} + x_k x_A$. If Eve intercepts E, p, S, $a_A S$, or $\{kS, x\}$, then by Proposition 2.48, Eve cannot determine A's private key a_A. So, only A knows A's private key a_A, meaning only A could have solved 2.10 for x and signed the hash of the message. So, B is assured of the authenticity of the message.

2.7 Webs of Trust

A web of trust is a mechanism to enforce key authentication in a decentralized system. There is no centralized authority that issues and signs certificates. Instead, each actor in the system certifies others' certificates, which builds a directed graph of key authentication relationships.

Definition 2.50. *A **web of trust** is a directed graph G such that each vertex $v \in V(G)$ denotes a node in a network, and $\exists$ an edge $(u, w) \in E(G)$ if and only if u certifies the legitimacy of w's public keys.*

Each actor in the web of trust has a document that contains information about the authentication relationships. This document is called a public key ring (PKR). The public key ring for node a_i contains the following fields:

- Identifier for a_i.
- Identity of node with which relationship is established.
- A boolean that represents the direction of the relationship (i.e.: 0 denotes a_i certifying another node, 1 denotes another node certifying a_i).
- A digital signature of the hash value of a certificate belonging to the node that is being certified.
- A trust value on the node being certified for the action of certifying other nodes.

Let W be a web of trust, $V(W)$ be the vertex set of W, $E(W)$ be the edge set of W, and $u, w \in V(W)$. Suppose that u wishes to certify w's public keys. u requests w's certificate. The certificate contains the hash of its contents. u computes the hash of the certificate and verifies that it matches the value published on the certificate. Then, u encrypts the hash of the certificate with a digital signature. u writes the digital signature and its trust value on w to certify other nodes to its PKR. u sends this record in its PKR to w, who adds it to its PKR, modifying the identifier, identity, and boolean fields appropriately. The process of authentication using a web of trust will be discussed in our proposed authentication scheme.

Chapter 3

Introduction to Digital Trust

We begin this discussion of the theory of digital trust with a survey of some basic facts from Govindan [2]. Digital trust has theoretical foundations in probability theory. We cite several results from Feller [1].

3.1 Basic Facts

Definition 3.1. *Let A, B be entities, and act be some fixed action.* ***Digital trust*** *is a function of the probability that A believes B will or will not perform act.*

Definition 3.2. *Let A, B be entities, and act be some fixed action. A* ***trust model*** *is a method of quantifying the trust of A on whether B will or will not perform act.*

Below we define several axioms of digital trust.

Axiom 3.3. *Trust is a relationship established between two entities for a specific action. Let A, B be entities, and act be some fixed action. We denote the trust relationship established from A to B regarding act by* $\{A:\ B,\ act\}$*. We say A is the subject and B is the agent.*

Axiom 3.4. *Trust is a function of uncertainty. If the subject believes the agent will certainly perform the action, then we say the subject fully trusts the agent to perform the action, and there is no uncertainty. If the subject believes the agent will certainly not perform the action, then we say the subject fully distrusts the agent to perform the action, and there is no uncertainty. If the subject does not know whether the agent will or will not perform the action, then the subject has the highest uncertainty.*

Axiom 3.5. *Let A, B be entities, act be some fixed action, and* $T_{A,B} \in \mathbb{R}$ *be the trust value for the relationship* $\{A:\ B,\ act\}$*. We write* $T_{A,B} = \{A:\ B,\ act\}$*.*

Axiom 3.6. *Trust relations are not necessarily symmetric. That is, let A, B be entities, act be some fixed action, and* $T_{A,B},\ T_{B,A} \in \mathbb{R}$ *be the trust values for the relationship* $\{A:\ B,\ act\}$ *and* $\{B:\ A,\ act\}$*, respectively. Then,* $T_{A,B}$ *does not necessarily equal* $T_{B,A}$*.*

Axiom 3.7. *Trust can can be established through direct observation of a node's behavior over time. Such a trust model is called a direct trust model. Alternatively, trust can be established with a recommendation. Let A, B, C are entities, and act is some fixed action. Suppose A wishes to establish trust on C, and B has already established trust on C. Then, A can establish trust on C as a function of B's trust on C and A's trust on B to supply a recommendation.*

3.2 A Direct Trust Model

Josang [3] proposes a trust model based on a probabilistic continuous Beta distribution. Since an assumption of this model is that the subject has observed a finite number of trials of the agent's behavior, we can adapt this model to serve as a direct trust model. Also, we can use the statistical concept of Bayesian inference in order to iteratively update the subject's trust value on the agent after observing more recent data sets of trials.

3.2.1 Bayesian Inference

Bayesian inference is a method of statistical inference that provides a way to combine new evidence with prior history, through the application of Bayes' Theorem. We will see that after observing new evidence, the original posterior probability can be treated as the new prior probability, and the new posterior probability can be computed as a function of the previous data and the new data. So, we can apply this method iteratively in order to repeatedly refine the results of our previously collected evidence.

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of previously observed data, $\tilde{x}$ be a new data point whose distribution is to be predicted, θ be a parameter of the distribution of $\tilde{x}$, and α be the hyperparameter of θ.

Definition 3.8. *The **prior distribution** is the distribution of the parameter before any data is observed.*

The prior distribution of θ describes $P(\theta|\alpha)$. The hyperparameter α is the parameter of the prior distribution.

Definition 3.9. *The **sampling distribution**, or likelihood, is the distribution of the observed data, dependent on its parameter.*

The sampling distribution, or likelihood, of X describes $P(X|\theta)$.

Definition 3.10. *The **marginal likelihood**, or evidence, is the distribution of the observed data marginalized over the parameters.*

The marginal likelihood, or evidence, of X describes $P(X|\alpha)$. We have

$$P(X|\alpha) = \frac{P(X \cap \alpha)}{P(\alpha)}, \text{ by Bayes' Theorem.}$$
$$P(X|\alpha) = \frac{\int_{\theta_1}^{\theta_2} P(X \cap \alpha \cap \theta) d\theta}{P(\alpha)}, \text{ by the Law of Total Probability.}$$
$$P(X|\alpha) = \frac{\int_{\theta_1}^{\theta_2} P(X|\alpha \cap \theta) P(\alpha \cap \theta) d\theta}{P(\alpha)}, \text{ by Bayes' Theorem.}$$
$$P(X|\alpha) = \frac{\int_{\theta_1}^{\theta_2} P(X|\alpha \cap \theta) P(\theta|\alpha) P(\alpha) d\theta}{P(\alpha)}, \text{ by Bayes' Theorem.}$$
$$P(X|\alpha) = \frac{P(\alpha) \int_{\theta_1}^{\theta_2} P(X|\alpha \cap \theta) P(\theta|\alpha) d\theta}{P(\alpha)}$$
$$P(X|\alpha) = \int_{\theta_1}^{\theta_2} P(X|\alpha \cap \theta) P(\theta|\alpha) d\theta$$
$$P(X|\alpha) = \int_{\theta_1}^{\theta_2} P(X|\theta) P(\theta|\alpha) d\theta, \text{ since } \alpha \text{ is assumed to be fixed.}$$

Definition 3.11. *The **posterior distribution** is the distribution of the parameters after observing the data.*

We have

$$P(\theta|X \cap \alpha) = \frac{P(X \cap \alpha|\theta) P(\theta)}{P(X \cap \alpha)}, \text{ by Bayes' Theorem.}$$
$$P(\theta|X \cap \alpha) = \frac{P(X|\theta) P(\alpha|\theta) P(\theta)}{P(X) P(\alpha)}$$
$$P(\theta|X \cap \alpha) = \frac{P(X|\theta) P(\theta|\alpha)}{P(X)}$$
$$P(\theta|X \cap \alpha) = \frac{P(X|\theta) P(\theta|\alpha)}{P(X|\alpha)}$$
$$P(\theta|X \cap \alpha) = \frac{\text{likelihood x prior}}{\text{evidence}}.$$

3.2.2 Beta Distribution

The beta distribution is a family of continuous probability distributions defined on the real interval $[0, 1]$, with parameters $\alpha, \beta \in \mathbb{R}_{>0}$. We say that α and β are the shape parameters of the beta distribution. We will see that the shape of the curve of the probability density function of the beta distribution is determined by α and β. The probability density function (PDF) of the beta distribution is

$$P(X = x) = f_X(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)},$$
$$B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}.$$

The expected value of the beta distribution is

$$E(X) = \frac{\alpha}{\alpha+\beta}.$$

3.2.3 Beta Direct Trust Model

Let the agent's behavior be a random variable $X \in \mathbb{Z}_2$ that follows a Beta distribution such that $X = 1$ if the agent performs the action, and $X = 0$ if the agent does not perform the action. Assume we can make independent successive observations of X. Let $\theta = P(X = 1) \in [0, 1]$, and $1 - \theta = P(X = 0) \in [0, 1]$. Then, we can model X as a sequence of Bernoulli trials with constant single-trial probability of success θ. Indeed, θ follows a continuous Binomial distribution. Suppose we observe n trials of X. Then, $P(X = 1 \text{ k times}) = \binom{n}{k}\theta^k(1-\theta)^{n-k} = P(X = k \text{ Successes } |\theta)$. Then, we have

$$\begin{aligned}
P(X = k \text{ Successes } |\theta) &= \binom{n}{k}\theta^k(1-\theta)^{n-k} \\
&= \frac{n!}{(n-k)!k!}\theta^k(1-\theta)^{n-k} \\
&= \frac{\theta^k(1-\theta)^{n-k}}{\frac{(n-k)!k!}{n!}} \\
&= \frac{\theta^k(1-\theta)^{n-k}}{\frac{\Gamma(n-k+1)\Gamma(k+1)}{\Gamma(n+1)}} \\
&= \frac{\theta^{(k+1)-1}(1-\theta)^{(n-k+1)-1}}{\frac{\Gamma(n-k+1)\Gamma(k+1)}{\Gamma(n+1)}}.
\end{aligned}$$

Let $y = \theta$, $k + 1 = \alpha$, and $n - k + 1 = \beta$. Then, we have

$$\frac{\theta^{(k+1)-1}(1-\theta)^{(n-k+1)-1}}{\frac{\Gamma(n-k+1)\Gamma(k+1)}{\Gamma(n+1)}} = \frac{y^{\alpha-1}(1-y)^{\beta-1}}{B(\alpha,\beta)} = P(\theta = y|\alpha \cap \beta).$$

So, θ is a continuous random variable that follows a Beta distribution. Then, we define the trust value $T \in [0,1]$ as follows:

$$\begin{aligned} T = P(\tilde{x} = 1|X = k \text{ Successes of } n \text{ previous trials}) &= E(\theta) \\ &= \frac{\alpha}{\alpha+\beta} \\ &= \frac{k+1}{k+1+n-k+1} \\ &= \frac{k+1}{n+2}. \end{aligned}$$

Now, we will apply the method of Bayesian inference. We have

$$\begin{aligned} P(\theta = y|X = k \text{ successes of } n \text{ trials } \cap \alpha \cap \beta) &= \frac{\text{likelihood}x\text{prior}}{\text{evidence}} \\ &= \frac{P(X = k \text{ successes of } n \text{ trials } |\theta = y)P(\theta = y|\alpha \cap \beta)}{P(X = k \text{ successes of } n \text{ trials } |\alpha)} \\ &= \frac{(\binom{n}{k}y^k(1-y)^{n-k})(\frac{y^{\alpha-1}(1-y)^{\beta-1}}{B(\alpha,\beta)})}{\int_0^y P(X = k \text{ successes of } n \text{ trials } |\theta = y)P(\theta = y|\alpha \cap \beta)dy} \\ &= \frac{(\binom{n}{k}y^k(1-y)^{n-k})(\frac{y^{\alpha-1}(1-y)^{\beta-1}}{B(\alpha,\beta)})}{\int_0^y (\binom{n}{k}y^k(1-y)^{n-k})(\frac{y^{\alpha-1}(1-y)^{\beta-1}}{B(\alpha,\beta)})dy}. \end{aligned}$$

A computation shows that

$$\frac{(\binom{n}{k}y^k(1-y)^{n-k})(\frac{y^{\alpha-1}(1-y)^{\beta-1}}{B(\alpha,\beta)})}{\int_0^y (\binom{n}{k}y^k(1-y)^{n-k})(\frac{y^{\alpha-1}(1-y)^{\beta-1}}{B(\alpha,\beta)})dy} = \frac{y^{\alpha+k-1}(1-y)^{\beta+n-k-1}}{B(\alpha+k,\beta+n-k)}.$$

So, we have a new beta distribution with parameters $\alpha + k$ and $\beta + n - k$. We can apply this method iteratively by letting the posterior distribution for the previous data become the prior

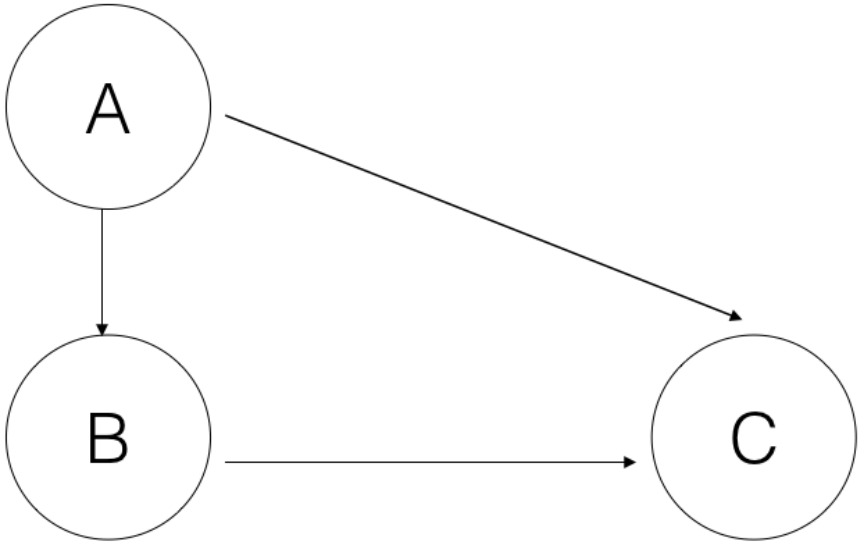

Figure 3.1: Concatenation propagation of trust.

distribution for the new data. That is, by letting the parameters $\alpha_{i+1} = \alpha_i + k$ and $\beta_{i+1} = \beta_i + n - k$. We initialize the parameters of the beta distribution to $\alpha_1 = \beta_1 = 1$, and so our initial trust value $T_1 = E(\theta) = \frac{1}{1+1} = \frac{1}{2}$, which represents complete uncertainty.

3.3 An Indirect Trust Model

Sun [6] provides a probabilistic indirect trust model. Throughout the course of this discussion, the trust value $T_{A,B} = \{A : \ B, \ act\}$ equals the probability $p_{A,B} = P(\text{A believes B will perform } act)$. We will discuss the concatenation propagation and multi-path propagation of trust, and propose a model for the general case of indirect trust establishment through a system of recommendations.

3.3.1 Concatenation Propagation

Consider the following scenario described by Figure 3.1: A wishes to establish trust on C regarding *act*. B has already established trust on C regarding *act*. A requests a recommendation from B regarding C's behavior.

Let $p_{A,B}$ denote P(B providing a good recommendation), $p_{B,C}$ denote P(B believes C will perform *act*), and $p_{A,C}$ denote P(A believes C will perform *act*). Let $p_B^{'}$ denote P(B makes correct recommendation), $p_{C \mid B=1}^{'}$ denote P(C performs *act* if B makes a good recommendation), and $p_{C \mid B=0}^{'}$ denote P(C performs *act* if B makes a bad recommendation). We can represent actions as probabilistic events. Let event X denote C performing act and event Y denote B providing a good recommendation. Clearly, X is dependent on Y. Then,

$$
\begin{aligned}
p_{A,C} &= P(Y \cap X) \cup P(\bar{Y} \cap X) \\
p_{A,C} &= P(Y)P(X|Y) + P(\bar{Y})P(X|\bar{Y}) \\
p_{A,C} &= p_B^{'} p_{C \mid B=1}^{'} + (1 - p_B^{'}) p_{C \mid B=0}^{'} \\
p_{A,C} &= p_{A,B} p_{B,C} + (1 - p_{A,B}) p_{C \mid B=0}^{'}
\end{aligned}
$$

Sun provides the following argument to further simply $p_{A,C}$. In the following discussion, assume that each trust value $T \in [-1, 1]$ such that $T = -1$ corresponds to probability $p = 0$, $T = 0$ corresponds to probability $p = \frac{1}{2}$, and $T = 1$ corresponds to probability $p = 1$.

Axiom 3.12. *Concatenation propagation does not increase trust. That is, let A, B, C be entities and suppose A wishes to establish trust on C regarding act, B has already established trust on C regarding act, and A requests a recommendation from B for C. Let* $T_{A,B}$ *denote* $\{A: B, \textit{make recommendations}\}$, $T_{B,C}$ *denote* $\{B: C, act\}$, *and* $T_{A,C}$ *denote* $\{A: C, act\}$. *Then,* $|T_{A,C}| \leq min\{|T_{A,B}|, |T_{B,C}|\}$.

If $T_{A,B} = 0$, then by Axiom 3.12, we have $|T_{A,C}| \leq min\{0, |T_{B,C}|\}$. Since $|T_{B,C}| \geq 0$, then $min\{0, |T_{B,C}|\} = 0$, and so $|T_{A,C}| \leq 0$. By definition, $|T_{A,C}| \geq 0$. Hence, $|T_{A,C}| = 0$, which corresponds probability $p_{A,C} = \frac{1}{2}$. We have

$$
\begin{aligned}
p_{A,C} &= p_{A,B} p_{B,C} + (1 - p_{A,B}) p_{C \mid B=0}^{'} \\
\frac{1}{2} &= \frac{1}{2} p_{B,C} + (1 - \frac{1}{2}) p_{C \mid B=0}^{'} \\
\frac{1}{2} &= \frac{1}{2} p_{B,C} + (\frac{1}{2}) p_{C \mid B=0}^{'} \\
1 &= p_{B,C} + p_{C \mid B=0}^{'} \\
p_{C \mid B=0}^{'} &= 1 - p_{B,C}
\end{aligned}
$$

Thus,

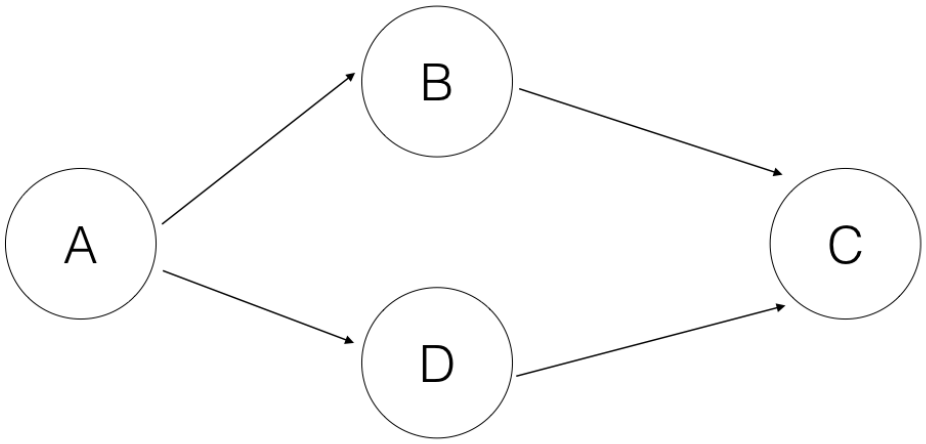

Figure 3.2: Simple multi-path propagation of trust.

$$p_{A,C} = p_{A,B}p_{B,C} + (1 - p_{A,B})(1 - p_{B,C})$$

Now, we return to our original coordinates of trust values (that is, the trust value $T_{A,B} = \{A : B,\ act\}$ equals the probability $p_{A,B} = P(\text{A believes B will perform } act)$). This is okay since it's just a matter of finding a bijection from the coordinates of the probability values to those of the trust values. Sun provides such a bijection.

Proposition 3.13. $p_{A,C} = p_{A,B}p_{B,C} + (1 - p_{A,B})(1 - p_{B,C})$.

3.3.2 Multi-Path Propagation

Consider the scenario in Figure 3.2. A receives independent recommendations from B and D regarding C's behavior.

Sun proposes the following model, based on the data fusion model:

$$\frac{p_{A,C}}{1 - p_{A,C}} = \frac{p_{A,B,C}p_{A,D,C}}{(1 - p_{A,B,C})(1 - p_{A,D,C})}.$$

We generalize this approach as described in Figure 3.3.

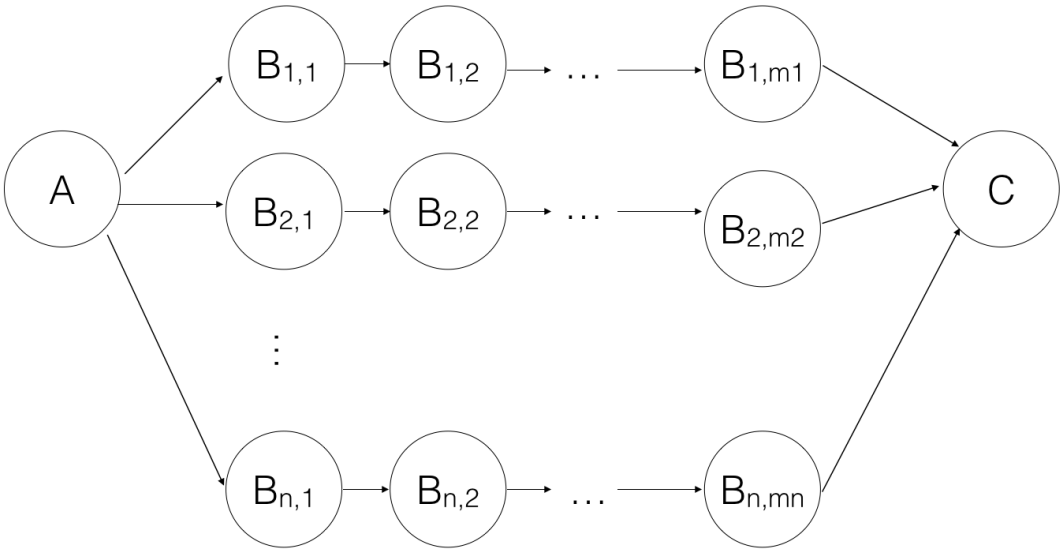

Figure 3.3: General multi-path propagation of trust.

Let each

$$p_{A,\{B_i\}_{i=1}^{m_i},C} = p_{A,B_{i,1}}(\prod_{j=1}^{m_i-1} p_{B_{i,j},B_{i,j+1}})p_{B_{i,m_i},C} + (1-p_{A,B_{i,1}})(\prod_{j=1}^{m_i-1}(1-p_{B_{i,j},B_{i,j+1}}))(1-p_{B_{i,m_i},C}),$$

$\forall i \in \{1,\ldots,n\}$.

Then,

$$\frac{p_{A,C}}{1-p_{A,C}} = \prod_{i=1}^{n} \frac{p_{A,\{B_i\}_{i=1}^{m_i},C}}{(1-p_{A,\{B_i\}_{i=1}^{m_i},C})}.$$

Let $c = \frac{p_{A,C}}{1-p_{A,C}} \in \mathbb{R}$. Then,

$$\begin{aligned} p_{A,C} &= c(1-p_{A,C}) \\ p_{A,C} &= c - cp_{A.C} \\ p_{A,C} + cp_{A.C} &= c \\ p_{A,C}(1+c) &= c \\ p_{A,C} &= \frac{c}{c+1} \end{aligned}$$

Proposition 3.14. $p_{A,C} = \frac{c}{c+1}$, $c \in \mathbb{R}$ *such that*

$$c = \prod_{i=1}^{n} \frac{p_{A,\{B_i\}_{i=1}^{m_i},C}}{(1 - p_{A,\{B_i\}_{i=1}^{m_i},C})}.$$

Chapter 4

An Authentication Scheme for MANETs

Let M be a graph representing a MANET and $V(M) = \{a_1, a_2, \ldots, a_n\}$ be the vertex set of M (that is, the set of nodes in M). Suppose the nodes of M have established a web of trust with the trust value corresponding the the trust relationship $T_{i,j} = \{a_i : a_j$, certifying other nodes' certificates$\}$. We will describe a scheme to perform the following actions:

1. Authenticate a node in the web of trust.
2. Establish the trust relationship $T_{i,j}$.

4.1 Authenticating a Node in the Web of Trust

Let $a_i, a_j \in V(M)$. Suppose that a_i wishes to send a message to a_j. First, a_i must authenticate a_j. If a_j is in a_i's PKR, then a_i trusts the legitimacy of a_j's public keys, and so a_i can send the message to a_j. Otherwise, if a_j is not in a_i's PKR, then a_i performs a depth-first search on the web of trust graph to find all paths to a_j. Consider the set S of all paths from a_i to a_j, and consider some path $P = \{e_{b_1=a_i,b_2}, e_{b_2,b_3}, \ldots, e_{b_{m-1},b_m=a_j}\} \in S$ such that each $b_k \in V(M)$. In order to authenticate a_j, a_i must verify each signature in P, and also compute the trust concatenation propagation by Proposition 3.13. So, a_i performs the following steps:

Procedure 4.1.

1: $T_1 := a_i$'s trust value for b_2 to certify other nodes.
2: $T_2 := 1 - T_1$
3: $PK := a_i$'s public key
4: **for** $k := 2$ **to** $m - 1$ **do**
5: $\quad a_i$ looks up $C := b_k$'s certificate

6: a_i computes H :=HASH(C)

7: a_i requests $R := b_k$'s PKR

8: a_i decrypts b_{k-1}'s signature S in R by computing $V :=$ Verify($S,\ PK$)

9: **if** V does not equal H **then**

10: Mark P as invalid

11: Finished

12: **end if**

13: $T' := b_k$'s trust value for b_{k+1} to certify other nodes in R

14: $T_1 := T_1 T'$

15: $T_2 := T_2(1 - T')$

16: **if** $T_1 + T_2 < T_{th}$ **then**

17: Mark P as invalid

18: Finished

19: **end if**

20: $PK := b_k$'s public key in C

21: **end for**

22: a_i computes the concatenation propagation $T_P = T_1 + T_2$ for P.

After the concatenation propagation has been computed for all paths in S, then a_i computes the multi-path propagation T by Proposition 3.14. If $T > T_{th}$, then a_i trusts the legitimacy of a_j's public keys, and so a_i can send the message to a_j. We use the Advanced Encryption Standard (AES) for a symmetric key cryptosystem, the Elliptic Curve Diffie-Hellman algorithm for symmetric key agreement, the Elliptic Curve Digital Signature Algorithm for digital signatures, and MD5 for a hash algorithm. First, a_i and a_j execute ECDH to agree on a symmetric key. Then, a_i encrypts the message with AES and the symmetric key. Next, a_i computes the hash of the message with MD5, and encrypts the hash with a digital signature using ECDSA. Finally, a_i sends the encrypted message and digital signature to a_j, which then decrypts the message to plaintext, and verifies that the decrypted digital signature matches its computed hash value of the decrypted message. Then, a_j is assured of the integrity and authentication of the message. That is, a_j knows that Eve didn't tamper with the message during transmission, and that the message originated from a_i.

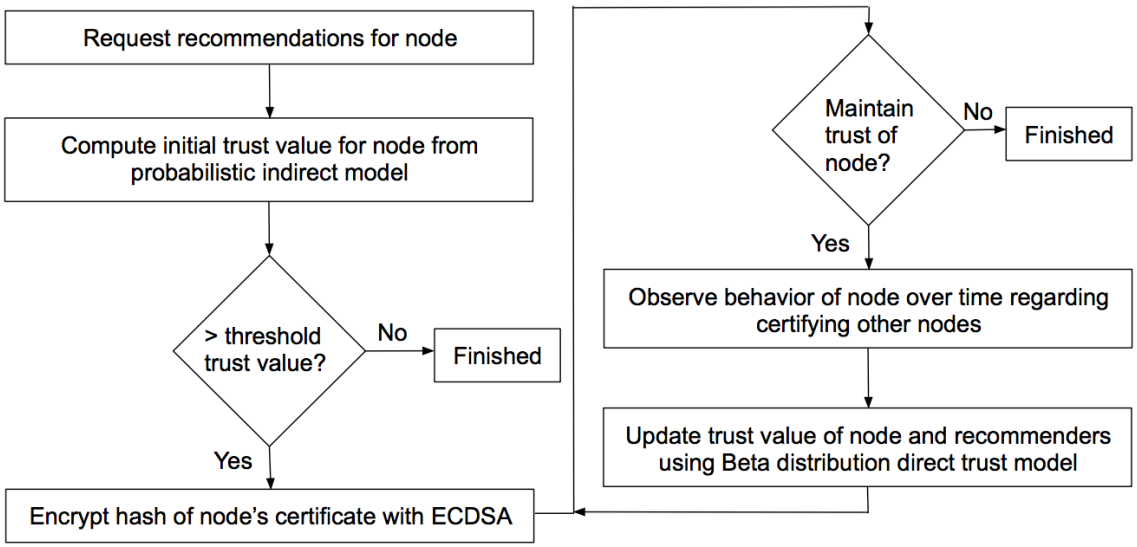

Figure 4.1: Certifying a node's certificate.

4.2 Certifying a Node in the Network

Now, suppose a_i wishes to certify a_j. That is, a_i wishes to add a record for a_j to a_i's PKR. First, a_i authenticates a_j's public keys by Procedure 4.1. Upon successful authentication of a_j's public keys, a_i will compute the trust value for a_j to certify other nodes through direct observation by using the Beta trust model. Then, a_i can sign the hash of a_j's certificate, and add the new record to a_i's PKR. a_i can update this trust value over time after observing further behavior of a_j. This process is described by the flow chart in Figure 4.1.

After establishing trust indirectly on a_j, if a_i wishes to maintain the trust on a_j, then a_i implements the direct observation approach. Initially, a_i's trust on a_j should be a function of both the indirect trust value and the direct trust value. However, over time, a_i will want its trust value on a_j to be weighted more towards that from its direct observation versus recommendations. So, we combine the two by using a remembering factor.

$$T_{a_i,a_j} = \frac{\beta^{t_c - t_r} R_{a_i,a_j} + D_{a_i,a_j}}{\beta^{t_c - t_r} + 1} \in [0,1],$$

where $R_{a_i,a_j}, D_{a_i,a_j} \in [0,1]$ are the trust values established through recommendations direct observation, respectively, $\beta \in (0,1)$ is the remembering factor, t_c is the current time, and t_r is the time at which the recommendation trust value was established. Initially, $t_c - t_r \approx 0$. Over time, $t_c - t_r >> 0$. Since $\beta \in (0,1)$, then

$$\lim_{t_c - t_r \to 0} \beta^{t_c - t_r} = 1,$$

and so

$$\lim_{t_c - t_r \to 0} \frac{\beta^{t_c - t_r} R_{a_i,a_j} + D_{a_i,a_j}}{\beta^{t_c - t_r} + 1} = \frac{R_{a_i,a_j} + D_{a_i,a_j}}{2},$$

which is the arithmetic mean of R_{a_i,a_j} and D_{a_i,a_j}. This makes sense since initially we want R_{a_i,a_j} to have a larger weight than it will later in time. But, we want D_{a_i,a_j} to also have a significant weight since it is computed from a_i's own observations. Additionally,

$$\lim_{t_c - t_r \to \infty} \beta^{t_c - t_r} = 0,$$

and so

$$\lim_{t_c - t_r \to \infty} \frac{\beta^{t_c - t_r} R_{a_i,a_j} + D_{a_i,a_j}}{\beta^{t_c - t_r} + 1} = D_{a_i,a_j}.$$

This makes sense since eventually, the recommendation trust value becomes outdated, and we want a_i's direct observations to have more weight. As described in Figure 4.1, we can use the Beta direct trust model to iteratively update D_{a_i,a_j} over time.

4.3 Attacks on MANETs

We will discuss two prominent attacks on distributed systems. The Sybil attack results when a malicious node forges the identity of another node which shares or takes the blame of the malicious actions from the original node. The newcomer attack results when a malicious node leaves the network and re-registers as a new node, effectively erasing its previous bad history. Both of these attacks can be addressed with an authentication scheme.

The defense against the Sybil attack involves making it more difficult for the malicious node to forge the certificate of a new node. Our authentication scheme involves a web of trust to enforce key authentication of nodes. Also, trust is established in the web of trust using a hybrid trust model. When a node certifies another node, the first node computes an initial trust value on the second node for the action of certifying other nodes. The initial trust values are established indirectly through a system of recommendations, and the trust value is then updated after each subsequent set of observations using the Beta direct trust model. This quantitative approach to trust establishment provides a higher level of granularity of trust management, since the threshold trust values in the system can be adjusted over time depending on how strict or forgiving each node in the network wishes to be. An underlying assumption of webs of trust is that the majority of the nodes are

benevolent. So, the majority of the nodes will honestly certify and provide recommendations for other nodes, making it impractical for a node to forge the identities of other nodes and stay in the network.

The defense against the newcomer attack is a pure authentication problem. For example, the system may enforce a bijective mapping of identities in the network onto hardware addresses. That is, a node should only be assigned exactly 1 identity in the network for its fixed hardware address. Then, a node will not be able to re-register with a new identity after leaving the network.

Chapter 5

Results

We ran some simulations in MATLAB to test our proposed authentication scheme. We chose to model the Sybil attack in which a malicious node attempts to forge the identity of a new node that will perform malicious actions to share the blame with the malicious node. Our simulation included 100 nodes, and we varied the total no. of malicious nodes from 10 to 50 in steps of 10. We modeled the behavior of the Sybil attack by defining the probability that a malicious node performs a bad action as a function of the total number of malicious nodes. If there are more malicious nodes, then each node needs to share less of the blame, and so we have a lower probability of a malicious node performing a bad action. If there are less malicious nodes, then each node needs to share more of the blame, and we so we have a higher probability of a malicious node performing a bad action. Figures 5.1 and 5.2 show plots of the trust computed by some arbitrary node in the simulated network on the malicious node performing the attack and the resulting forged node, respectively, versus the total number of malicious nodes. In both cases, as there are more malicious nodes in the network, then each malicious node needs to share less of the blame, and so each can achieve a higher trust value. However, after applying our authentication and trust scheme, we can see that the trust values for both nodes are < 0.6 when the number of malicious nodes is $< \frac{1}{2}$ of the total nodes in the network. Since an underlying assumption of webs of trust is that the majority of the nodes are benevolent, then these trust values seem reasonable. In this case, a threshold trust value ≥ 0.6 seems appropriate.

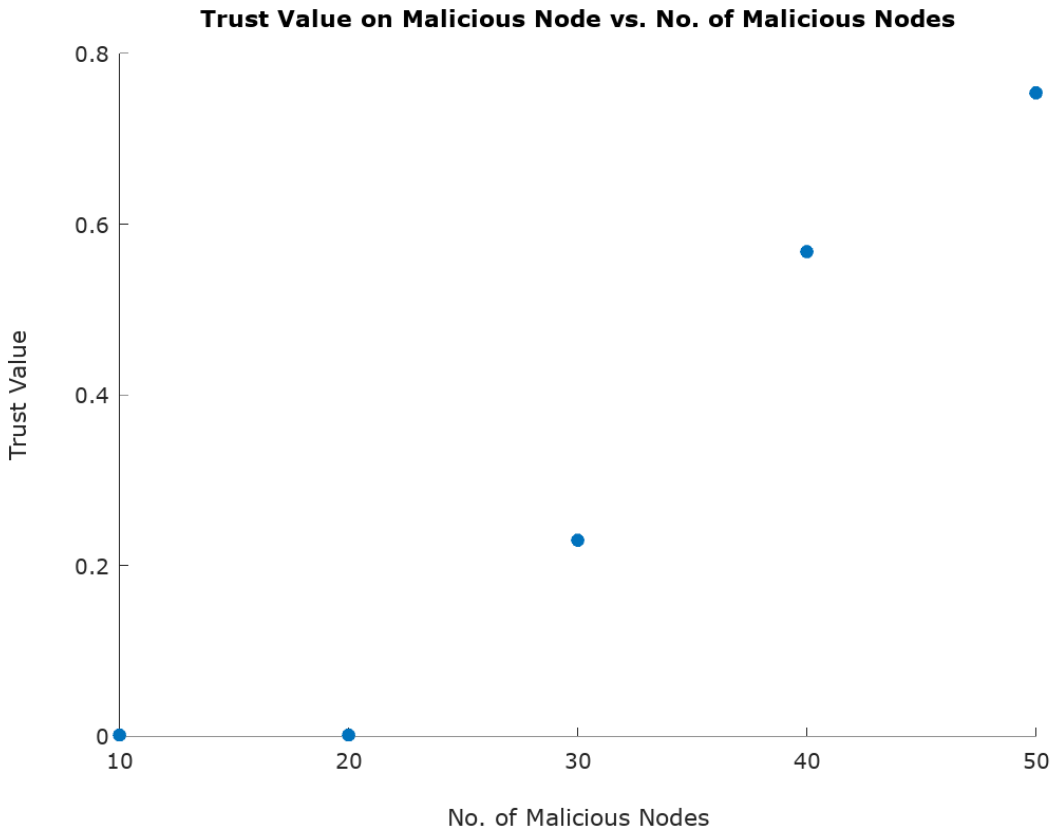

Figure 5.1: Trust on a malicious node vs. total no. of malicious nodes.

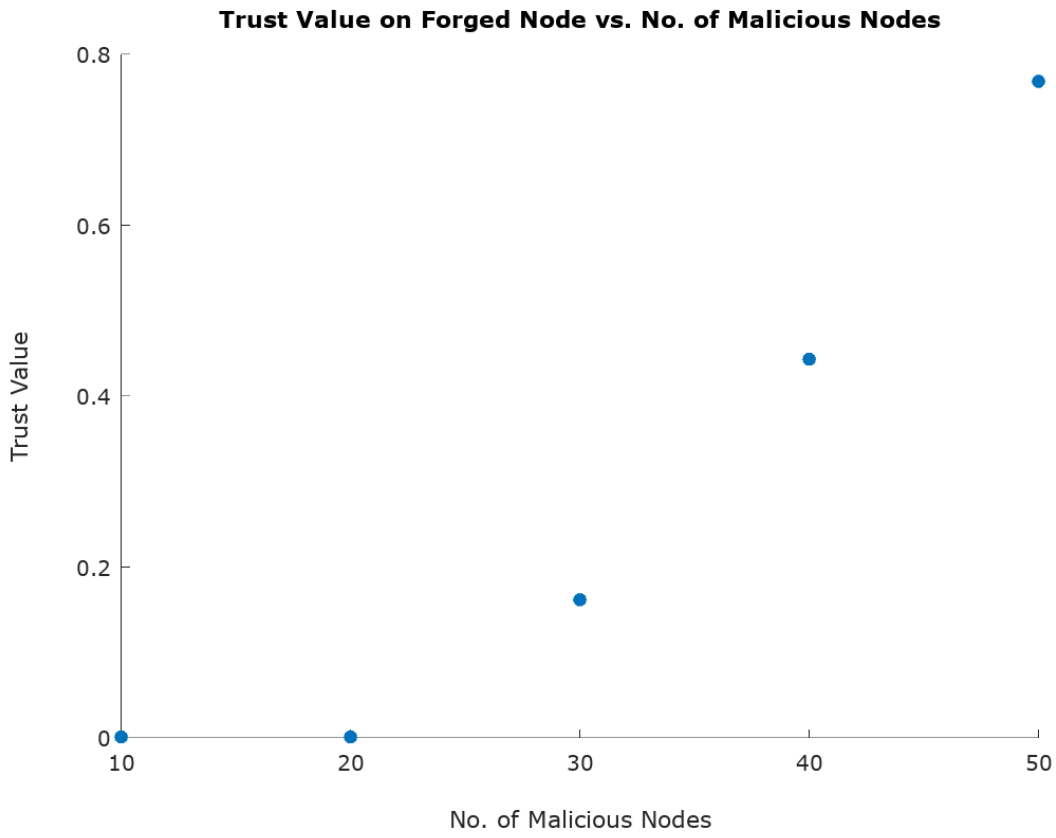

Figure 5.2: Trust on a forged node vs. total no. of malicious nodes.

Chapter 6

Software Design and Architecture

6.1 Requirements

Functional:

- The system will implement a trust model that supplies accurate trust data to an authentication mechanism to authenticate nodes in a distributed mobile ad hoc network.
- The iOS application will be able to discover and view all devices in the network.
- The system will allow the user to select a node in the network with which he or she wishes to communicate.
- Upon successful authentication of that node, the user will be able to view files on that authenticated device.
- The web application will model the trust values and relationships between all nodes in the network which will allow people to analyze it.

Non-functional:

- The system will implement a direct/indirect hybrid trust model that balances accuracy of trust data and performance overhead.
- The system will implement a PGP-style web of trust as an authentication mechanism to authenticate nodes in the distributed network.
- To simulate a network, the system will store a list of simulated devices as well as their trust values in a database. Once a device enters the network, it pulls those values to use for calculation locally.

- The iOS application will support the distribution of text files.
- The iOS application will perform the symmetric key agreement with the Elliptic Curve Diffie-Hellman (ECDH) public key cryptosystem.
- The system will perform the authentication with the Elliptic Curve Digital Signature Algorithm (ECDSA).
- The web aplication will present the trust values and relationships between nodes in a graphical format where the vertices represent devices and edges represent trust values.
- The system will enforce ease-of-use through its user interface.

Design constraints:

- The file-sharing component of the system will will be a mobile iOS application.
- The data visualization component of the system will be a web application.
- The iOS application will be compatible with any version over iOS 10.0.
- The web application will be compatible with Safari and Chrome.

6.2 Use Cases

iOS application

User connects to another device

- Preconditions
  - Application is opened
  - User is connected to WiFi
  - Devices are listed
- Postconditions
  - User will be connected with the target device
- Exceptions
  - User does not have an Internet connection
  - The connection is not secure

User reads files from another device in the network

- Preconditions
  - Device has established a secure connection to another device
- Postconditions
  - User successfully reads files from destination
- Exceptions
  - Low or unsteady wireless connection

Web application

User views the trust graph

- Preconditions
  - Application has been opened
  - User has an Internet connection
- Postconditions
  - User is able to view the trust graph and interact with nodes and edges
- Exceptions
  - No Internet connection

6.3 Technologies Used

- C is the implementation language for our back-end cryptographic implementations, which include the trust model, web of trust, and the AES, ECDH, ECDSA, and MD5 algorithms.
- Cryptlib is a free C cryptography toolkit that provides security services such as data encryption, symmetric key agreement, authentication, message integrity, timestamping, and secret/symmetric and public key management algorithms and protocols.
- Objective-C is our implementation language for the front-end of the iOS mobile application.

- Cocoa Touch is the set of frameworks that compose the iOS software development kit. It mainly provides the user interface graphical drawing services for the front-end component of the application.

- PHP is a widely used scripting language which is suited for server side web development. It will be used for interacting with the database and sending the required information to the iOS application and the web application.

- MySQL is a popular open sourced relational database management system. It will be used in order to make direct calls as well as manage trust values and node data in the system.

- Javascript is a standard multi-paradigm programming language that we will use implement our web front-end component.

- D3.js is a Javascript library for manipulating documents and data visualization. We will use D3.js to present in the front-end the data collected by our web component.

- Git is a version control system that promotes integration and non-linearity and software design. It will enable us to work in parallel while maintaining a continuous master version of our solution.

6.4 Design Rationale

- Objective-C and Cocoa Touch are the standard iOS development technologies.

- We chose C as our back-end implementation language because it is interoperable with Objective-C and because its runtime environment will provide efficient execution of our cryptographic implementations.

- Cryptlib is an extremely popular, efficient, and extensible cryptography and security toolkit that implements AES, ECDH, ECDSA, and MD5.

- We chose PHP because it is a well ingrained standard in server-side development. Since PHP is already installed on the SCU Design Center and our team has prior experience, we are choosing to use it.

- We chose MySQL because it is a popular standard for database management, installed on the SCU Design Center and a technology that our group is already familiar with.

- D3.js is a popular Javascript framework for data visualization which is why we have used it to implement the data-visualization component of the web application.

6.5 Test Plan

For developing the web of trust, we conducted unit testing for each module or function after it has been completed. This made the development process more efficient, and ensured that bugs did not propagate through the entire system.

For the iOS application and the web application we conducted UI testing. We also conducted simple black box testing with expected inputs and outputs to make sure that our iOS application met the functional requirements we had set for it.

6.6 Development Timeline

Our development timeline details our progress for fall, winter and spring quarter, split into roles.

Figure 6.1: *Development Timeline*

6.7 Activity Diagram

Our activity diagram, shown below, consists of two separate modules for our mobile and web applications.

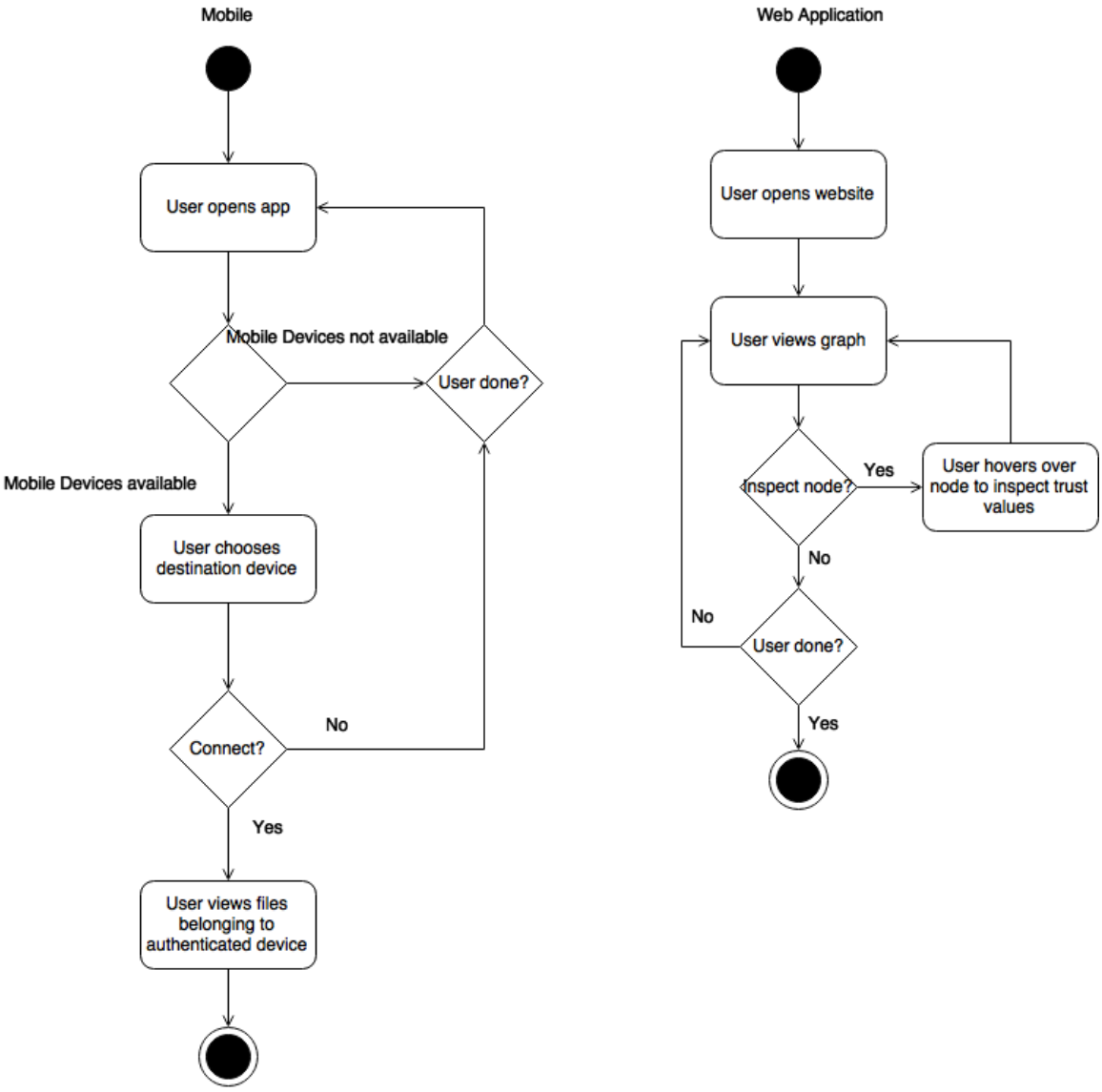

Figure 6.2: *Activity Diagram*

6.8 Architectural Diagrams

The diagrams shown in the below figures describe the architecture of our ad hoc network, iOS application, and web application, respectively.

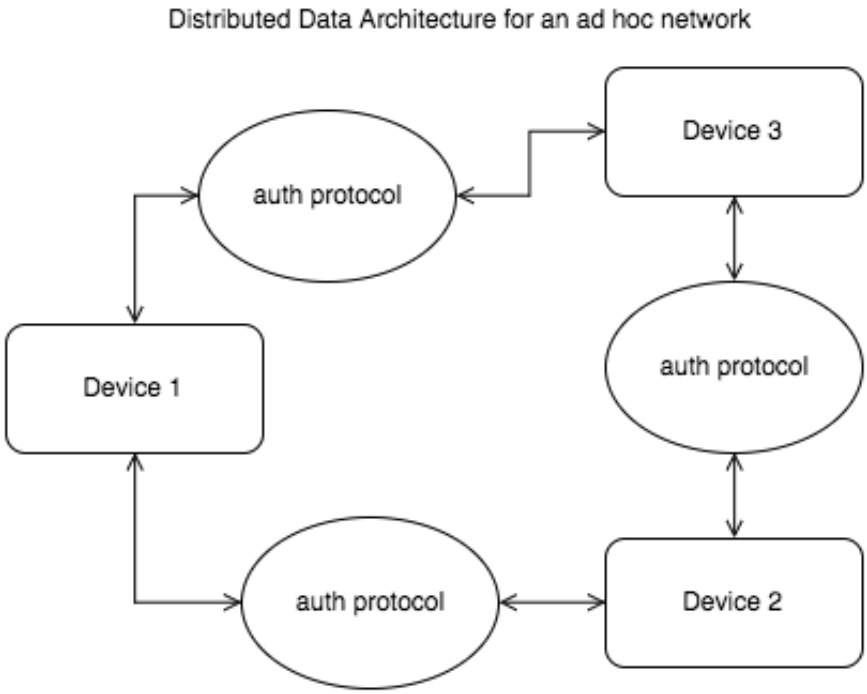

Figure 6.3: *Network Architecture*

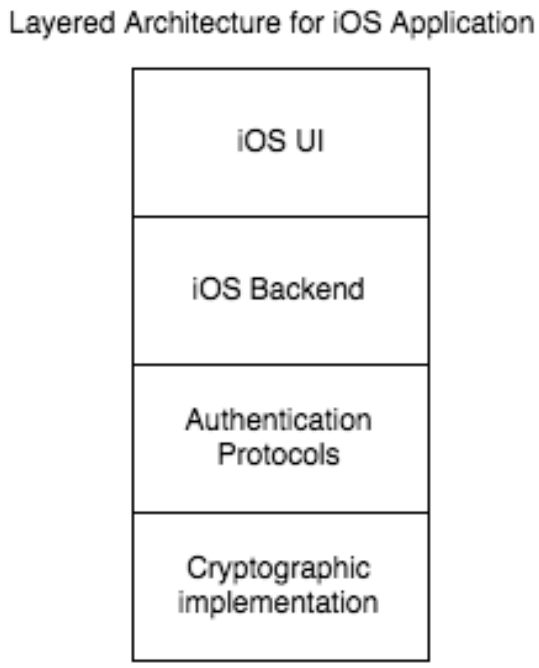

Figure 6.4: *iOS Architecture*

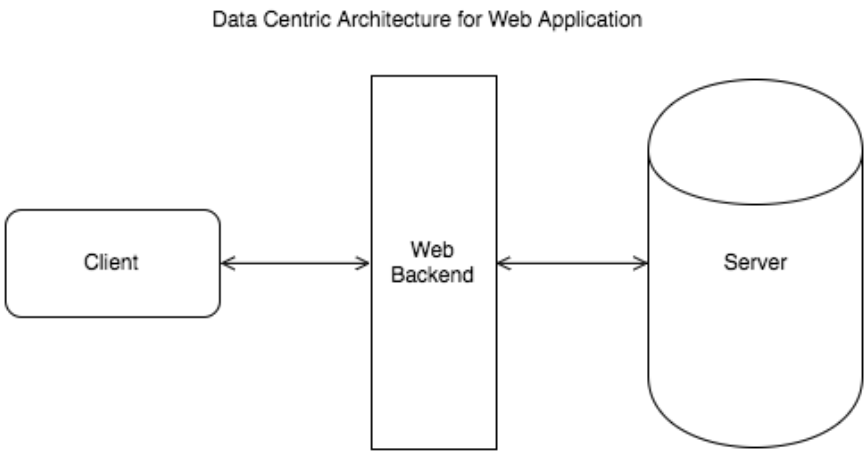

Figure 6.5: *Web Architecture*

6.9 Conceptual Models

The below images are final screen shots of our iOS and web applications, showing potential use cases.

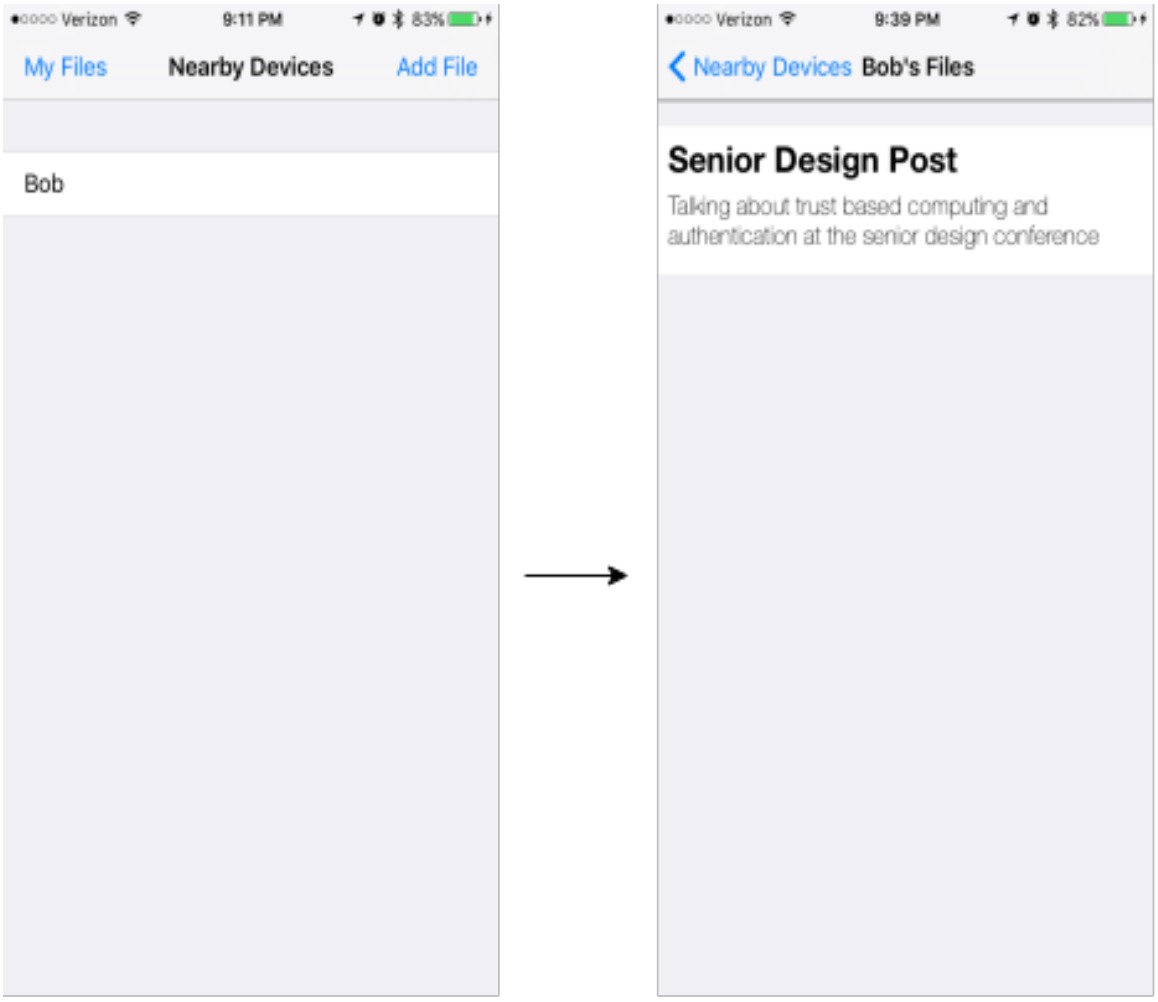

Figure 6.6: *iOS Model*

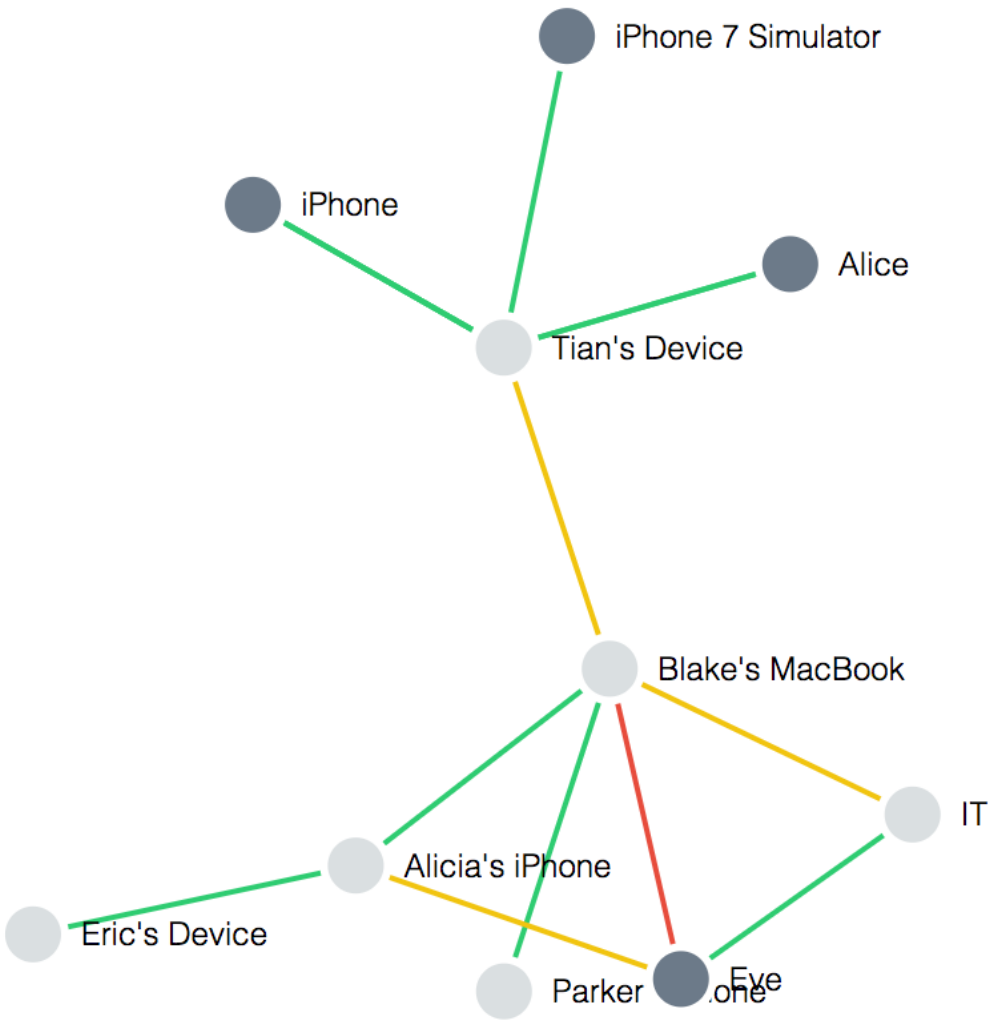

Figure 6.7: *Web Model*

6.10 Risk Analysis

The following figure tabulates the risks we might expect to encounter along our timeline and how we plan to mitigate these risks.

Risk	Consequence	Probability	Severity	Impact	Mitigation Strategies
Illness	Must reassign missed work. Potential to fall behind on timeline.	.9	4	3.6	Ensure all teammates are w Maintain good hygiene by w often and avoiding dirty env
Poor Communication Among Team Members	Damaged team relationship, incomplete or misrepresented product.	.7	6	4.2	Ensure team roles are well l that each team member is w wants to be in terms of load and responsibility.
Data Corruption and/or Loss	Project must be reimplemented.	.1	9	.9	Store local, periodic backups good practice in VCS use.

Figure 6.8: *Potential Risks*

Chapter 7

Societal Implications

With the advent of the smartphone, Internet of Things, and many other modern technologies, we are given many new means of connection and data exchange, but are also exposed to new method of attack. Thus, it is crucial that the authentication schemes in these MANETs be sound and properly implemented. Building these technologies securely can several different societal implications, including:

7.0.1 Ethical

In many cases, sensitive user data is at risk. Breaches of privacy and personal security can occur if authentication fails.

7.0.2 Social

In the same vein, these technologies, particularly IoT devices, allow us to connect with peers in ways that were not possible before. We must be sure that any peer we share data with is properly authenticated.

7.0.3 Economic

Cryptocurrencies, specifically those that are decentralized, have challenged existing currency systems and have become more prominent in world markets. In exchange of cryptocurrency, it is crucial peers are verified beforehand.

Chapter 8

Conclusion

Our project builds on the areas of abstract algebra, elliptic curves, public key cryptography, and digital trust in order to construct an authentication scheme for MANETs based on a hybrid trust model between the direct and indirect approaches. Our authentication scheme addresses the Sybil and newcomer attacks on distributed systems. The MATLAB simulations we ran verify the security of our authentication scheme, and in our example, suggest a threshold trust value of 0.6. Some future work includes performing more sophisticated MATLAB simulations and investigating several implications of the mobility aspect of MANETs on trust management, such as the maximum levels of trust concatenation.

Bibliography

[1] W. Feller, *An Introduction to Probability Theory and Its Applications*. New York: John Wiley & Sons, Inc., 1950.

[2] K. Govindan, and P. Mohapatra, "Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey," *IEEE Comm. Surveys & Tutorials*, vol. 14, no. 2, pp. 279-298, Apr. 2012.

[3] A. Josang and R. Ismail, "The Beta Reputation System," in *15th Bled Electronic Commerce Conference*, Bled, 2002, pp. 1-14.

[4] T. Judson, *Abstract Algebra: Theory and Applications*. Ann Arbor: Orthogonal Publishing L3C, 2015.

[5] C. Long, *Elementary Introduction to Number Theory*. Long Grove: Waveland Press, Inc., 1987.

[6] Y. L. Sun, W. Yu, Z. Han, and K. J. R. Liu, "Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks," *IEEE J. on Sel. Areas in Comm.*, vol. 24, no. 2, pp. 305-317, Feb. 2006.