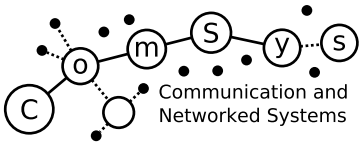
FACULTY OF
COMPUTER SCIENCE



Communication and Networked Systems

Master Thesis

Digital Representation for Web of Trust in Internet of Things

Poorvi Mandyam Bhoolokam

Matr.

Supervisor: Prof. Dr. rer. nat. Mesut Güneş
Assisting Supervisor: MSc. Frank Engelhardt

Institute for Intelligent Cooperating Systems, Otto-von-Guericke-University Magdeburg

27. November 2020

Abstract

Abstract

With Internet of Things (IoT) gaining dominance in the field of technology, devices are getting smarter day by day. The usage of smart devices poses one major challenge - Security. Security of devices and their services are of top priority which has been the key research area in the field of IoT. Security puts forth the concept of trust networks which forms the basis for several smart home security networks. The main task of this research is to model a trust network and to compute the trust value. Many kinds of research use centralized Public Key Infrastructure (PKI) to establish trust between peers. The disadvantage of centralized PKI is that the certificate exchange is dependent on a Central Authority (CA) which in some cases may prove to be malicious. This research employs the concept of the web of trust where peers authenticate themselves and there is no CA involved, in the sense that it is a decentralized PKI. The peers are required to authenticate each other by exchanging their certificates and validating them. In the web of trust, peers exchange certificates and validate them through a digital signature. The creation of a digital signature involves encryption algorithms. The most widely used algorithms are Digital Signature Algorithm (DSA), Rivest–Shamir–Adleman (RSA) and Elliptic Curve Digital Signature Algorithm (ECDSA). Among these, ECDSA proves to be efficient. The major advantages of ECDSA over DSA and RSA are smaller key size implying lesser memory consumption and better security. The trust model is first designed as a network model for a smart home scenario. This is further represented as a bi-directional graph with vertices V, edges E and a weight function w: k_x, k_y, t, A. The proposed approach is to convert the bi-directional graph into an electrical network of resistors to calculate the trust value. The peers exchange their public keys to validate each other using ECDSA following the concept of the web of trust. In the process of key exchange and authentication, some unlikely smart home problems might occur, for instance, a new peer enters the network or an existing peer leaves the network. Such scenarios are considered for trust value computation. The trust value is calculated as conductance G which is the reciprocal of the total resistance R in a network. The translation of the graph to the electrical network is done based on the series and parallel trust properties which can be easily mapped to the series and parallel connection of resistors in the network. It is proven that trust degree along a series path is lesser than that along parallel paths. This can be compared to the conductance in an electrical network of resistors where the conductance across the series connection of resistances is lesser than that across the parallel connection of resistors. The trust value is returned as conductance from the electrical network. The JavaScript Object Notation (JSON) description of the

entire trust model is scripted considering all the key factors. The trust model is evaluated in three parts. Part 1 is the overhead analysis in which overhead between peers during the key exchange is computed. Data overhead, message overhead and database overhead are computed with reference to ECDSA. The overhead is also computed based on smart home problems. Part 2 is the attack tree analysis where all the possible attacks that could occur so as to compromise the system are described as attack scenarios. The attack tree is then analyzed based on the attacks that are prevented by the proposed approach. Part 3 is the validation of the proposed electrical network. The conductance value is calculated for two different examples considering the smart home problems that could occur within a communication network. The trust value calculated as conductance value is verified and evaluated.

Contents

List of Figures

List of Tables

Acronyms

CA Central Authority. iii, 6

DSA Digital Signature Algorithm. iii, v, 11–13, 23, 24, 30, 32

ECC Elliptic Curve Cryptography. 2, 11, 12

ECDSA Elliptic Curve Digital Signature Algorithm. iii–v, vii, 2, 11–13, 23, 24, 30–32, 34, 42–44

FOAF Friend-Of-A-Friend. 7

IoT Internet of Things. iii, v, 1, 5, 9, 12

JSON JavaScript Object Notation. iii, 2, 3

PGP Pretty Good Privacy. 11

PKI Public Key Infrastructure. iii, 2, 6, 7

RSA Rivest–Shamir–Adleman. iii, 11, 23, 24, 32

CHAPTER 1

Introduction

The main focus of the thesis is to model trust networks for IoT in order to ensure security in smart IoT devices. Security is one of the major challenges in the present era's growing usage of smart application devices. Therefore, establishing trust among peers in a communication network becomes necessary to ensure security and to prevent threats. This research involves creating a trust model and computing trust value between any two peers in the network.

1.1 Motivation

IoT is one of the most booming technologies which connects almost everything together. The definition of IoT- Internet of Things is that everything is connected through the internet. In the present world, everything around us is connected through the internet directly or indirectly. This implies that the Internet of things is evolving over the years and in the next 10-15 years we can imagine a whole new revolution in the field of science and technology. IoT finds its main application in Smart devices like Smart Homes, Smart cities, Smart Vehicles, and so on. Smart devices are making people's life comfortable, luxurious, and time-efficient. For example, when Person A enters his house, the light automatically switches on and when he leaves the room it automatically turns off. This is just one smart home scenario. There are several advantages that can be drawn from this scenario like energy efficiency and time efficiency. In this era and for years to come, devices will only become smarter and prove to be efficient. Smart devices use the concept of IoT where devices are connected to each other and people through the internet. With the several advantages of IoT and smart devices comes one major challenge- Security.

Security plays a major role in an IoT device or in any communication where the exchange of data is involved. Any intrusion in the network can result in a security breach like unauthorized access to sensitive and personal data. This is a real problem in the present time since almost all information about a person or organization is stored on the internet. For instance, if two organizations are communicating with each other by exchanging some sensitive data and an intruder enters the network, he can acquire this data which puts the organizations' reputation at stake. This is an example of a very simple attack that can be caused due to a security breach. In Smart devices, this becomes a more serious issue

leading to several other attacks that can cause malicious behaviour of the device itself. This challenge of security requires the concept of trust networks. Trust is a very important aspect of communication and data exchange. When two peers communicating with each other can establish trust amongst each other, security is assured automatically and there is no chance of any security breach.

Trust is very subjective in nature because it is an abstract feeling about another entity. The degree at which you trust one person differs from others and in addition the degree at which you trust your friend may not be the same at which he trusts you. It becomes very necessary to formalize this concept of trust in communication networks. Many types of research perform trust network analysis by using trust metrics which represent trust as percentage values or in binary, either 0 or 1. The trust metrics are ambiguous since percentage values depend on the human perspective, that is 90% trust degree might be considered as high for one person whereas the other person might consider it as a low value.

To overcome this ambiguity of trust metrics, this research focuses on a new electrical network approach for computing trust values. Establishing trust between two entities also involves the exchange of certificates and encryption algorithms for communication and data exchange. In this regard, the concept of the web of trust is applied rather than the typical hierarchical PKI since the web of trust is decentralized. Among the encryption algorithms ECDSA – an extension of the Digital Signature algorithm is implemented. uses the concept of Elliptic Curve Cryptography (ECC) because of which it is more secure. These properties are ECDSA and encryption algorithms are described in the further part of the research. The trust network model proposed combines the concept of the web of trust,ECDSA and the electrical network approach for IoT to achieve enhanced efficiency.

The main contributions of the above research are:

- Modelling a smart home scenario in the form of a network model
- Employing ECDSA as an encryption algorithm along with the web of trust concept for key exchange.
- A new proposed electrical network model for trust value calculation: The graph network of the network model is translated into an electrical model to compute the trust value. The trust value is calculated as a conductance value G which is the reciprocal of the equivalent resistance of a network.
- JSON description of the proposed trust model- The entire trust model is described in JSON in separate file structure as per the flow mentioned in Section 3.2.
- Evaluation of the electrical model is done using examples of electrical networks by illustrating the computation of trust value between peers. In addition to this, the problems of smart homes are also illustrated with examples.
- A security analysis is evaluated through an attack tree that describes all possible attacks that can occur in the trust model to compromise the entire system.
- An overhead analysis is performed to illustrate the overhead due to the key exchange between peers for trust establishment

1.2 Thesis Structure

Chapter 2 explains the existing approaches of trust networks, the concept of the web of trust, and some typical trust metrics that were designed. Chapter 3 describes the idea of trust network as a network model, graph formalization of the model, the encryption methodologies, the electrical network approach, and the JSON representation of the trust model that is designed. Chapter 4 briefly explains the outcome of the research by an attack tree analysis and the electrical network validation. Chapter 5 summarizes the thesis and describes the open tasks for future work.

CHAPTER 2

Related Work

Trust is a characteristic which describes honesty, reliability and transparency between two entities. As defined by [1] “Trust is the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible.” Now the question arises as to why one person needs to trust the other and in what circumstances. Trust comes into picture when there are some threats that need to be tackled. Consider a peer P_1 communicates with peer P_2. If P_1 does not trust P_2, it may lead to an argument or declare what 1 has told to be false though the content is true. This is just a general scenario of trust. In present-day scenarios where people and organizations have to constantly interact with each other without the knowledge of the authenticity of the corresponding organization, analyzing trust between two people without considering any other events is not efficient. It becomes necessary to consider trust between two communicating peers based on roles and experiences. The consideration of these events ensures that trust analysis to be more accurate and easier to model. [1] proposes "Trust Network Analysis with Subjective Logic" which analyses the trust model as series and parallel graphs considering the events such as trust-based on roles and trust based on experiences. For this analysis, this research uses canonical expressions for notation. Trust can be categorized as [2]
1.Direct trust: Two entities have knowledge about each other’s nature, so they trust each other without any information from a trusted third party or intermediate entity.
2. Indirect trust: Two entities are completely unknown to each other, instead trust is established via an intermediate entity or trusted third party.
In pervasive computing networks, the behaviour of entities continuously changes. This property of the dynamic change of behaviour needs to be addressed by trust models. This problem is solved by [2] by incorporating decentralized trust management and taking into consideration properties like direct trust, indirect trust, and changing behaviour of communicating entities. Entities take recommendations into account for trust analysis. In networking, trust plays a very important role to secure information that is being transmitted among several people or organizations. Trust networks form the basis for security systems in most applications. Security systems these days are of major use in IoT applications like Smart Homes. Some of the major attacks recently faced in smart homes are listed below [3]

1. Mirai DDoS botnet attack- disrupted internet.
2. Nest thermostat- wrong information conveyed which left the users feeling cold.
3. A baby monitor was hacked.
4. IoT malware and ransomware attacks.

Trust Networks involve the concept of PKI. PKI is a set of roles and policies which are needed for the creation, usage, and storage of digital certificates. It also manages public-key encryption. The main purpose of PKI is the ability to secure information transfer over a range of several network activities like e-commerce, internet banking and so on. This implies trust networks can be designed using the centralized hierarchical PKI approach or the Web of trust- that is the decentralized PKI approach. Web of trust is a decentralized trust model concept in PGP which deals with authentication of a user by binding the public key and the owner. Pretty good privacy is an encryption method that is used for signing, encrypting and decrypting texts and emails. The centralized PKI approach consists of a CA or a small hierarchy which provides certificates to all its peers. The peers can respectively authenticate themselves based on these certificates. But the main problem arises when several entities having their own certificate hierarchy want to collaborate and communicate with each other[3]. This is when the web of trust has a huge advantage over centralized PKI. The web of trust model assumes that a user can trust many users, can validate the certificates from other users or can trust third party users to verify their certificates. There are two main entities: An introducer is someone who can sign someone else's public key and a meta-introducer is someone who can sign keys, also can choose who is an introducer. This implies that any user can be a central authority [4]. In [5], addresses the problem of typical hierarchical PKI which uses a Central Authority for certificate exchange. The research proposes the concept of Web of Trust in which the network is represented by a directed graph with trust values represented as weights of the edges. A certain non-probable heuristics for calculating the trust values is used which takes into consideration some parameters like worst path, best path and all independent paths. Germano Caronni describes trust heuristics by considering serial and parallel trust paths. In a serial trust path, the overall trust is much lesser compared to the trust calculated in the parallel trust path. This is because of the fact that, in parallel trust paths, any one of the paths with maximum probabilities is enough to calculate the trust. In serial trust paths, the trust decreases as the intermediate nodes increase. [6] states the problem of employing the centralized PKI which does not work well with ad hoc networks and certainly does not solve the certificate chain discovery problem. The certificate chain discovery problem involves finding a certificate chain path to verify a public key, in other words, a certificate chain to authenticate users. The research describes the usage of the web of trust in ad hoc networks where the centralized PKI fails to work. The study is based on an ad hoc network that uses the concept of a web of trust where each node has the ability to provide certificates to others in an organized manner. The proposed method uses a weighted directed graph with notations such as V->set of nodes, E-sent of edges. The approach is divided into two main steps. The first step is the certificate searching phase in which a source node broadcasts a search packet to all the directly trusted nodes. When a particular node receives the packet it adds its own certificate and further broadcasts it to it directly trusted nodes. This continues until the same node receives two packets. The second step is the certificate collection phase in which the destination node upon receiving all certificates from the nodes adds its own certificate and sends it back to the source node. [7] states the problem of trust from a human perspective such as honesty and reliability and the necessity to

describe the importance of trust in semantic web applications. The research describes the Friend-Of-A-Friend (FOAF) project model for semantic web networks with some of their self-tailored parameters to calculate trust metrics. FOAF is a project that can enable users to create and link information about who they know. Information like name, email id and homepage can the stored in the RFD vocabulary of the FOAF schema. This research deals with FOAF extended with some additional properties. Since FOAF is the base of the research, users are identified by their email addresses. The properties indicate trust levels. Like others, this research also uses a weighted graph to represent the network. The trust metrics used are: Maximum and Minimum capacity paths – indicates the trust capacity of paths, Maximum and minimum length paths – indicates the hop count, Weighted average- indicates recommended trust level.

When the concept of trust arises, we need some quantity to measure this trust. This quantity is trust metric. [8] defines trust metric as "Trust Metrics (TM) is a new term and is defined in this paper as the information of an entity that is required and used to evaluate the trustworthiness of the entity". Maurer considers probabilities as the trust metric which is also referred to as confidence levels. This trust metric is very useful since there is the flexibility of converting values from one trust metric to another in the range [0,1], confidence parameters may be based on past experiences which enables to identify risks and many a time the parameters can be assigned automatically so that the end-user need not do it. [9] solves the problem of a typical PKI architecture where trust is represented in binary, 0 or 1. The revocation of the public key is not considered in most of the researches. Thereby this research employs a vector model to derive the trust metric. The model defines trust as a vector of various parameters, thereby contributing to the over trust in different percentages. The main challenge of trust networks is handling recurring intrusion of peers and changing behaviour of entities. This problem requires trust metrics that take into consideration some major points such as alternating malicious behaviour and sudden change in a person's behaviour. These two points play a major role in designing a trust metric as per [10]. There are also some basic requirements of trust dynamics to model this trust metric like

1. Sensitivity to new experiences
2. Sensitivity should not depend on overall experiences
3. Long term behaviour must be recorded.

The dynamic trust metric consists of three main factors

1. short term trust st
2. long term trust lt
3. Penalty factor pt - for the alternating behaviour of a peer

Other researchers Jonker and Treur specify some trust dynamics such as blindly positive, blindly negative, slow positive- fast negative, balanced slow, balanced fast, and slow negative - fast positive that solve the problem as stated by [11].

The Table 2.1 illustrates the advantages of each of the researches as parameters described in this chapter. The main parameters chosen are oscillatory behaviour, flexibility in terms of solving the stated problem, decentralization of trust model and the usage of graph theory for trust analysis. These parameters are considered since they are the major challenges that are being faced in trust networks that need to be solved. The represents that the corresponding concept is not addressed in the research and x indicates that the concept has been addressed in the research.

Trust Network Research Comparison				
Author	Oscillating behaviour	Flexibility	Decentralization	Graph network
Caronni2000	-	-	X	X
Mohri2007	-	X	X	X
Golbeck2003	-	-	-	-
Bicakci2005	-	X	X	X
Duma2005	X	X	X	X

Table 2.1: Research Comparison

CHAPTER 3

Thesis Contribution

3.1 Concept Formalisation

3.1.1 Trust models for IoT networks based on smart home examples

The demand for smart IoT devices is increasing since everyone wants an easy and comfortable life. The usage of these devices comes with risks and threats. In this context, the concept of trust plays a major role. It is important for the user of the network to trust the people who access the device to avoid intruders from causing unnecessary damage to the network. This poses major threats like man-in-the-middle attack and other problems with regard to authenticity and reliability. For this reason, a trust model design is essential. Generally, a trust model for smart homes is represented by a network model. An example for a designed network model for one of the smart home scenarios, that is Smart Lighting System is illustrated in Fig. 3.1. The network model consists of a user who needs to be authenticated before he/she enters the network. This is indicated by a router that acts as a Central Authority for just the user. Once the user enters the network and connected to the Smart device, the can authenticate his peers. This follows the web of trust model wherein peers exchange certificates and authenticates themselves. In our case, there is an extra addition to this. In the process of exchanging certificates, peers also agree upon a common label which is further used to grant access rights based on these labels. With respect to the above description, an axiom is followed in the remainder of the research.

A1. Label agreement: Two peers exchange certificates and the label has to be agreed upon by both the peers.

The typical method of analysing a trust model is to convert the designed network model into a graph to enable simpler assessing of the model. The graph is further analysed for the establishment of trust. In the research of [6], trust is established if there is an edge between two nodes. The weight function across the edge represents the trust degree in percentages. This percentage indicates how one node trusts the other. This percentage value becomes ambiguous since trust is subjective. Based on the human perception the percentage value can indicate different things for different people. In [5], a trust along a serial path is

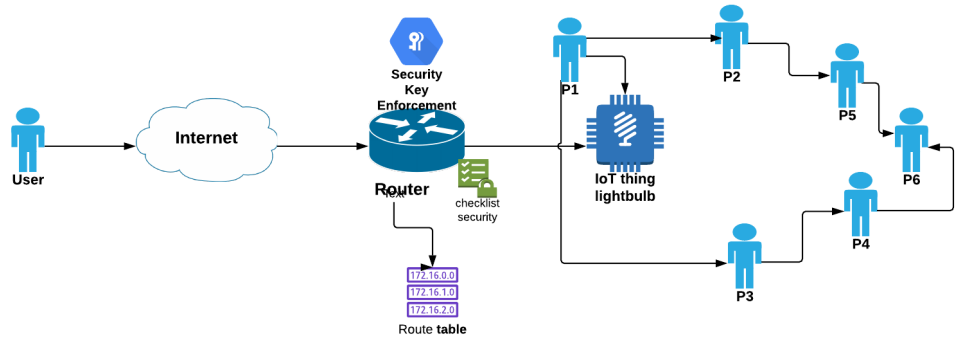


Figure 3.1: Network model for Smart Home Scenario

considered to have lower trust than trust along parallel paths. This is due to the fact that in a series path, the trust degree decreases with the increase in intermediate nodes. In contrast, when multiple parallel paths, exist, trust degree is said to increase because there might be a path in the graph where the number of intermediate nodes is possibly lesser than along the other paths. These references can be used to analyse trust establishment better by firstly, converting the network model to a graph which is the typical approach. Secondly, this graph representation can be translated into an electrical network of series and parallel resistances. In electrical engineering, the conductance G which is the reciprocal of resistance is more for parallel connection of resistances and less for series resistances. This fact maps to the series and parallel path concept of trust in graphs. In the remainder of the thesis, these aspects are formalised and explained in detail with examples. With respect to the above example, the network model is converted into a bi-directional weighted graph to design the trust network. The network model is translated to the graph as in 3.2.

Consider a bi-directed weighted graph $G = (V, E, w)$ with, $V = (P_1, P_2, ...P_n)$ where

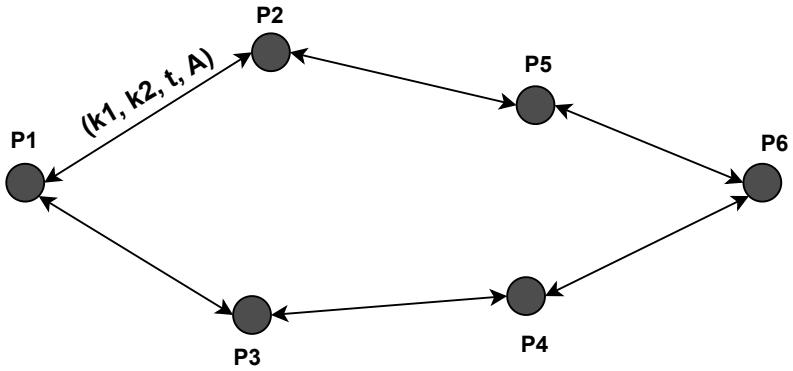


Figure 3.2: Network Model Graph

$n \in N$ representing the nodes. w is the edge weight function

$$w : E \to K \times K \times A; (V_1, V_2) \mapsto (k_1, k_2, a),$$

that maps an edge $E(V_1, V_2)$ to a tuple representing the public keys k_1, k_2 of the corresponding nodes and an agreed label a from a set of labels A. t is the trust degree function $t(V_1, V_2)$ describing is the amount of trust between the nodes.
The metric involves the conductance value G which is computed as the reciprocal of the resistance value. This computation is introduced in Section 3.1.5. The above translation of the network model to a bidirectional graph is important for the analysis of the trust model. The analysis of trust models deals with an important aspect of key exchange and Encryption Algorithms. The Section 3.1.2 describes this concept in detail.

3.1.2 Key exchange and encryption standards

The analysis of the trust model involves authentication of peers which is done using key exchange and certain encryption methodologies. Pretty Good Privacy (PGP) is a security standard which is used for the secure transmission of messages as well as the establishment of trust. PGP as a standard supports certain encryption algorithms like DSA, RSA and ECDSA. These encryption algorithms are used to establish trust and authenticate peers by exchanging keys. The peers generate a pair of a public key and private key. They exchange the public keys and a message amongst each other. A signature is generated using the encrypted message and the sending peer's private key which is created using a hash function. The signature is further verified using the receiving peer's public key. This key exchange uses certain encryption algorithms like DSA, RSA and ECDSA. In Section 3.1.2 the operations of DSA and ECDSA are described in detail [12].

DSA

DSA is described on the basis of modular expressions and discrete logarithm problem. The method involves generating a signature of the message M using the private key. This generated signature is then verified by using the public key. The fact that a signature can be only created using the private key ensures that forging a signature is almost impossible. In addition, since DSA is based on one modular expression and discrete logarithm problem, it is not easy for an intruder to compute these number-theory problems. The hash functions used for generating the private key is also considered secure. The entire DSA schema can be summarized into four steps:

- Key Generation
- Key Distribution
- Signing
- Signature verification

ECDSA

ECDSA is an extension of DSA which is based on ECC. When compared to other public-key cryptography methodologies, ECC has an advantage of reduced key sizes. There is a significantly large difference between these key sizes which are not suitable in cryptography, thereby affecting the speed of operations.

An elliptic curve is basically of the $y^2 = x^3 + ax + b$, where y, x, a and b are parameters of the finite field. The kind of curve that is created is based on the values of a and b. ECC considers a special point infinity in addition to these points on the curve. The private key is just a random number and is not a point on the curve. The public key is generated by multiplying the private key with a point on the curve called generator$g$. This is a point on the curve. An elliptic curve is defined by the following cryptographic parameters -F_p, F_{2^m}, p, a, b and g. In order to create custom curve, these parameters are used, but in most cases, the named curves are used. Named curves are the curves that already exist as packages. Similar to DSA, a signature for message m is generated by using the private key which is then verified by the public key. In this case, the private key is generated using a named elliptic curve and a hash function, usually SHA256.

Fig. 3.3 describes the process of communication between two peers by exchanging certifi-

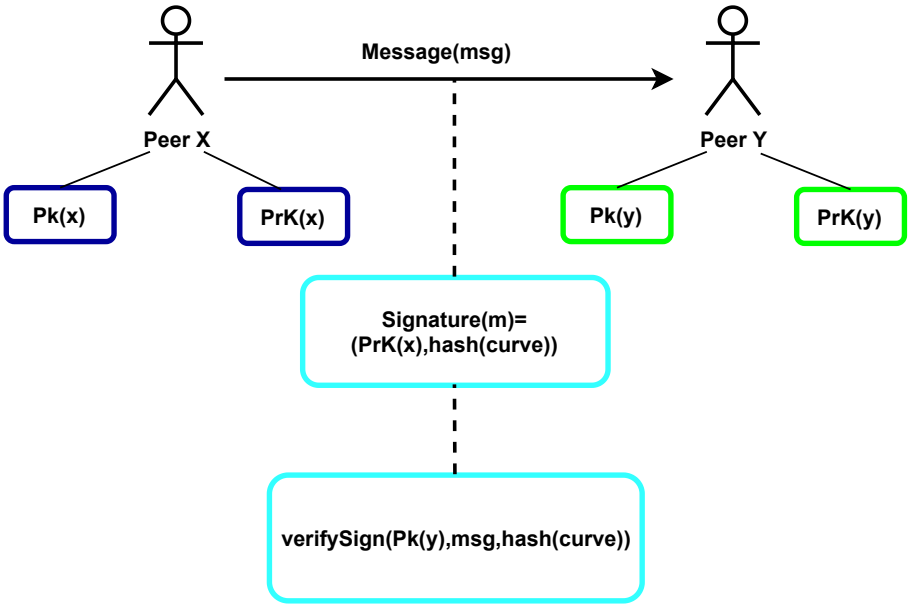


Figure 3.3: ECDSA Algorithm-Operation

cates using ECDSA. Peer X and Peer Y each have a public key and a private key pair. Peer X and Peer Y exchange their public keys. The information send by Peer X is encrypted by Peer Y's public key and can be decrypted only by Peer Y's private key. Both the peers digitally sign this encrypted information with their private key which is generated using a hash function and a named elliptic curve. This signature is then verified by Peer Y using its public key. The successful verification of the signature and decryption of the message ensures that Peer Y is trustworthy.

3.1.3 ECDSA vs DSA

ECDSA has several advantages over the traditional DSA cryptographic method, which is why it is largely applied in most IoT systems. The advantages of ECDSA over DSA are categorized as mentioned below:

- Key Size: The key size in ECDSA is very small compared to that of DSA.As described by[], for a key size of 15 360 bits in DSA, the ECDSA size would be merely 512 bits, which is a huge difference. Key size needs to be minimum for faster processing of the algorithm.
- Processing speed: The processing speed in ECDSA is very high in contrast to DSA. This is because ECDSA uses Elliptic curves to generate keys whereas DSA uses the traditional Hash function.
- Space required: The space required is ECDSA is less than that of DSA because of the size of the keys.
- Time for signature processing: The time for signature processing greatly depends on the key size. The smaller the key size, lesser time is consumed for processing the signature. This above description indicates that ECDSA consumes less time over DSA

The difference between typical hierarchical and web of trust is illustrated implying that Web of trust is more advantageous. In addition, two of the frequently used encryption algorithms that are DSA and ECDSA are explained in detail and the pros and cons of each algorithm are described. As stated ECDSA has more advantages in terms of speed, lesser key size and more security. Therefore, this research employs ECDSA as an encryption algorithm along-with the web of trust concept for peer authorisation. In the process of establishing trust, there can be various problems or unlikely events that can occur within the network. These problems are addressed in Section 3.1.4.

3.1.4 The Problems of Smart Homes

In a smart home scenario, as described in the network model Fig. 3.1, there are few scenarios that could occur during communication within the network. With reference to the above graph description, peers represented as nodes communicate with each other by exchanging their public keys and mutually agree upon a label. The frequently occurring events during the communication are,

- A new peer enters the network: When a new peer enters the network, the peer exchanges certificates with two trust peers existing in the network. This implies that an edge is created between the two nodes. The purpose of exchanging certificate with precisely two nodes is because this forms a parallel path which in turn increases the trust value with respect to that node. The trust value is then computed based on this updated network of the new peer.
- An existing peer leaves the network: When an existing peer leaves the network, the certificate of the peer deleted from the peers that are associated with it. This therefore results in the deletion of the edges between the peers with respect to the leaving peer. The trust value is however calculated from the updated network of the leaving peer.

The events stated above are illustrated with example with reference to Fig. 3.4 as,

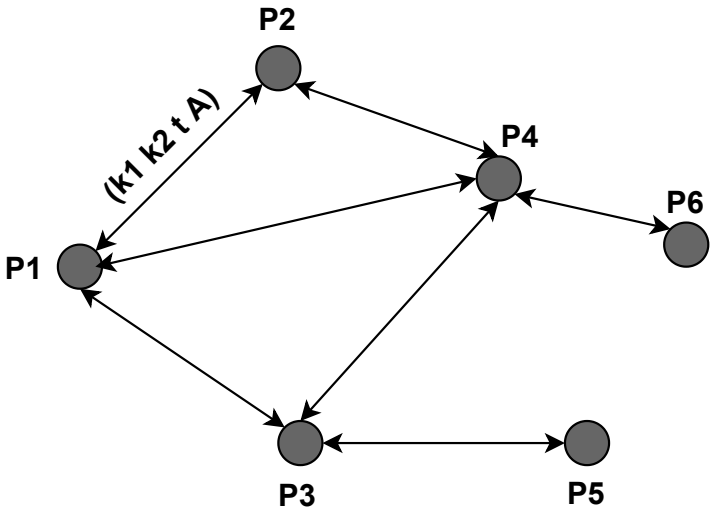


Figure 3.4: Graph example 1

Scenario 1: A new peer enters the network

When a new peer, P_n enters the network, P_n should be able to exchange certificates with two peers to complete the network. This indicates that P_n is supposed to have two edges connecting two other trusted peers.
With reference to the above graph, when a new peer Pn enters the network, the shares its certificates with two other trust peers in this case P_1 and P_2.In addition edges E(P_1,P_n) and E(P_n, P_2) are created. This is represented in Fig. 3.5.

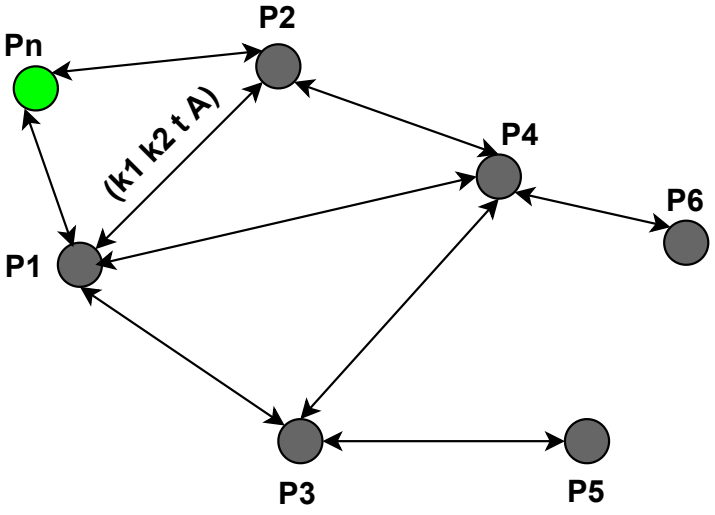


Figure 3.5: Example 1:New node enters the network

Scenario 2: Peer leaves the network

When a peer, P_y leaves the network, the certificates of P_y are deleted from the two associated trusted peers. In addition, the edges connecting Py will also be deleted.
Considering the above graph, when peer P_2 leaves the network, the certificates of P_2 with the corresponding trusted nodes P_1 and P_4 are deleted, indicating that the edges E(P_1,P_2) and E(P_2,P_4) are also deleted. This is illustrated in the Fig. 3.6.

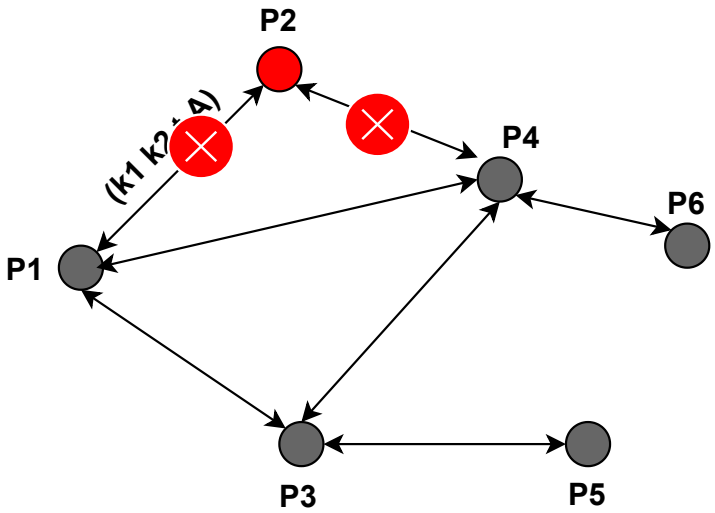


Figure 3.6: Example 1:Existing network node the network

Consider the following network examples as illustrated below. In the Fig. 3.7, two scenarios can occur.

Example 2

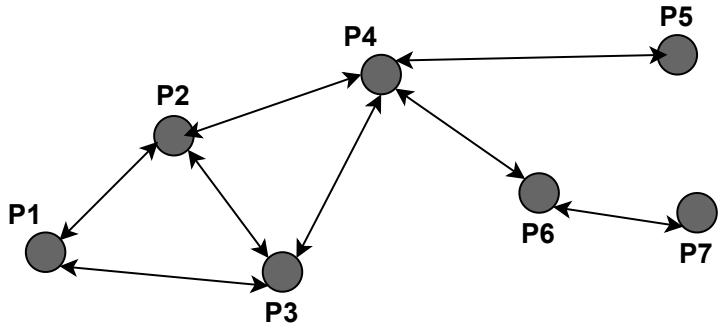


Figure 3.7: Graph example 2

Scenario 1

With reference to Fig. 3.8, consider a peer, P_n enters the network. The peer has to exchange certificates with any two trusted peers. This connection with the trusted peers is random. A peer is said to be trusted if it exchanges its certificate with two other peers and they agree upon a common label. P_n can then connect with such trusted peers, P_4 and P_5. Once the new network is formed, the trust degree is to be re-computed.

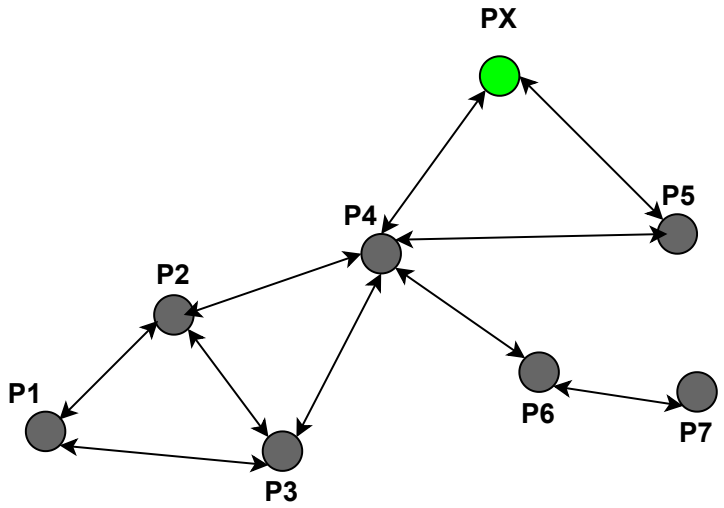


Figure 3.8: Example 2:New node enters the network

Scenario 2

Consider the scenario, where an existing peer P_3 leaves the network. In that case, the certificates of P_3 are deleted with respect to its trusted peers, P_1, P_3 and P_4. A new network is formed with the deletion of the edges corresponding to P_3. The trust value is re-computed for the new network,Fig. 3.9.

Considering Example 2 for the two scenarios as described with reference to Fig. 3.10,

Example 3

Scenario 1

Consider a peer, P_y enters the network. The peer has to exchange certificates with any two trusted peers P_4 and P_6. The edges $E(P_y,P_4)$ and $E(P_y,P_6)$ are created. Once the new network is formed, the trust degree is re-computed as depicted in Fig. 3.11.

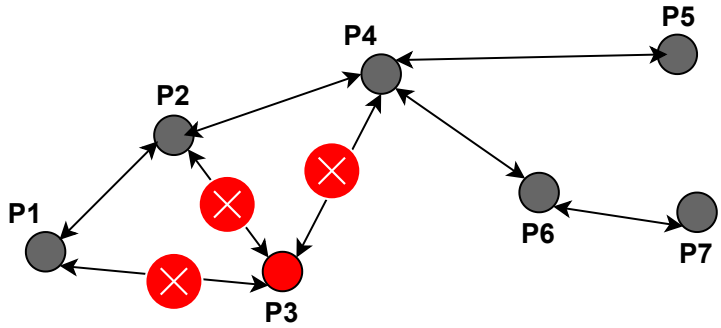


Figure 3.9: Example 2:Existing network node the network

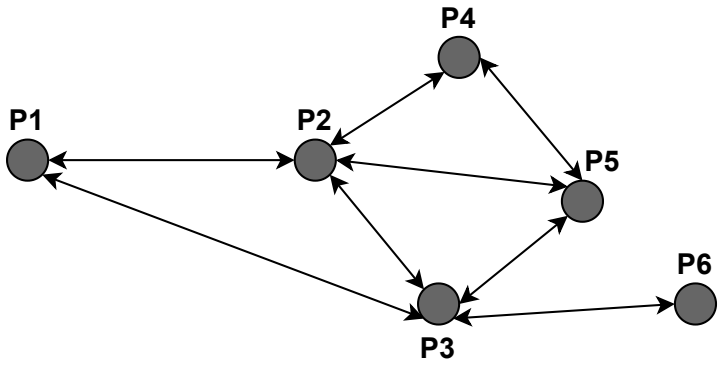


Figure 3.10: Graph example 3

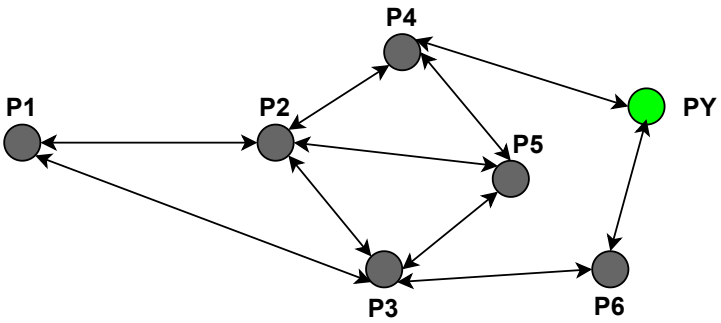


Figure 3.11: Example 3:New node enters the network

Scenario 2

Consider the scenario, where an existing peer P_1 leaves the network. In that case, the certificates of P_3 are deleted with respect to its trusted peers, P_2 and P_3. A new network is formed with the deletion of the edges corresponding to P_1. The trust value is recomputed for the new network, Fig. 3.1

The unlikely or sudden events that occur in the network while establishing trust are addressed and illustrated using examples as described. Section 3.1.5 deals with the translating

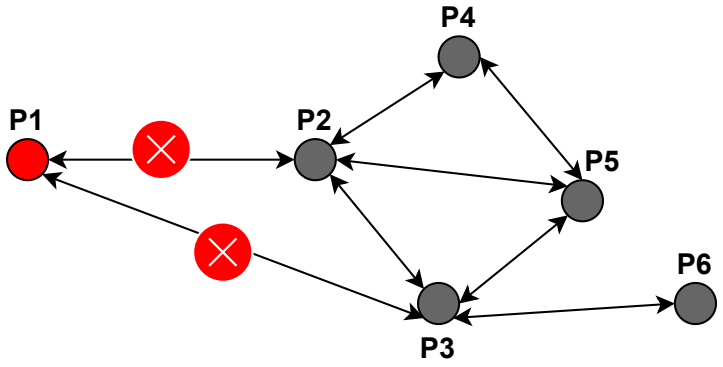


Figure 3.12: Example 3:Existing network node the network

the graph model to the proposed electrical network.

3.1.5 Presenting the electrical conductivity model as a new approach to calculate trust

In this section, the computation of the trust value t is illustrated using anew approach of translating the bi-directional graph as described in Section 3.1.1. The trust value defines how much a source peer trusts the target peer. This is essential to establish for the communication of important sensitive information between peers. To compute this trust value, the bi-directional graph can be converted into an electrical network consisting of resistors between the edges. This means that for every edge between two peers, indicating that the peers trust each other, a resistor is inserted. The complete translated bi-directional graph can be viewed as an electrical network with resistors. This can be easily analysed as series and parallel connection of resistors. Considering the above graph we can deduce some basic properties such as:[ref]

Property 1. Series node paths have a low degree of trust

Property 2. In comparison to Property 1, parallel node paths have a greater degree of trust

These two properties are used as the basis for the remainder of the work. To compute the trust degree as a value of conductance we consider, the graph in Fig. 3.2 that is transformed into a network with resistors between two nodes that have an edge. This network is illustrated in Fig. 3.13.

The possible topology is a series, parallel, delta and Y network.

In electrical engineering, the possible topology is a series, parallel, delta and Y network. A network of resistors connected in series has the same current flowing through all the resistors. This is because the current(I) flows through a single path. The total resistance of n resistors, R = R_1, R_2, R_3,....R_n is calculated using

$$R_{\text{total}} = R_1 + R_2 + R_3 + ...R_n \tag{3.1}$$

In a network with parallel connected resistors the current(I) is distributed along the parallel paths. In the case of parallel connected resistor network, total resistance of n resistors,

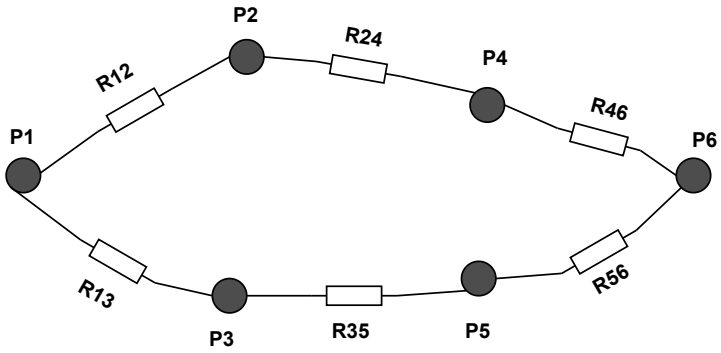


Figure 3.13: Equivalent Electrical

R = R_1, R_2, R_3,....R_n is calculated using

$$R_{\text{total}} = (R_{\text{x}} * R_{\text{y}})/(R_{\text{x}} + R_{\text{y}}) \tag{3.2}$$

The concept of conductance is important with regard to the flow of current. Conductance is a measure of flow of electric charge along a path and its unit is mho. It is the reciprocal of resistance. The conductance for series and parallel network of resistors is given in equation 3.3 and 3.4.

$$G_{\text{series}} = 1/R_{\text{total}} \tag{3.3}$$

$$G_{\text{parallel}} = R_{\text{total}} \tag{3.4}$$

In case of a Delta topology, the network has to be first transformed into a Y network and thereafter the equivalent resistance should be calculated. The transformation from Delta to Y network as illustrated in Fig. 3.14 is given by

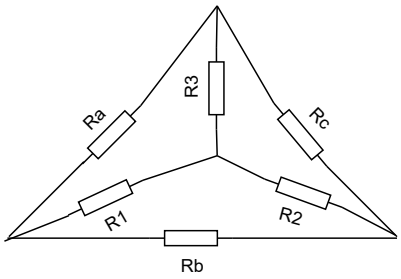


Figure 3.14: Delta-Y Transformation

$$R1 = R_{\text{a}} * R_{\text{b}}/R_{\text{a}} + R_{\text{b}} + R_{\text{c}}$$

$$R2 = R_{\text{b}} * R_{\text{c}}/R_{\text{a}} + R_{\text{b}} + R_{\text{c}}$$

$$R3 = R_{\text{a}} * R_{\text{c}}/R_{\text{a}} + R_{\text{b}} + R_{\text{c}}$$

The purpose of Delta-Y transformation is to reduce the complex network into a simple equivalent network of resistors [13]. The transformed Y-network is then simplified using

the series- parallel resistance calculation operations.
This value of conductance can be very well compared to the trust degree between nodes.Here communication between two peers or trust between two peers is equivalent to the flow of electric charge; current. It can be concluded that the trust degree across a series path is less when compared to that across a parallel path.
The above mentioned properties can be proved by an example :
Consider all resistors of 1ohm,
Considering Fig. 3.15, the conductance G between P1 and P4 can be computed as,

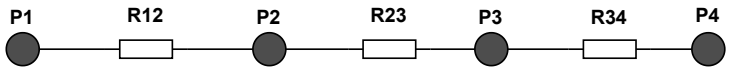


Figure 3.15: Series Graph

$$R_{\mathrm{eq_s}} = R_{12} + R_{23} + R_{34} = 3$$

$$G_{\mathrm{eq_s}} = 1/R_{\mathrm{eq_s}} = 1/3 = 0.33$$

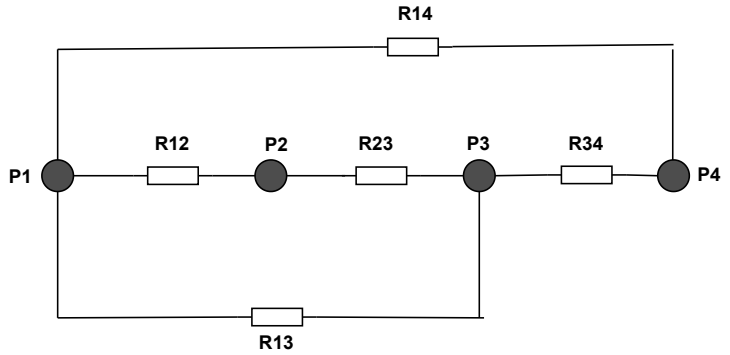


Figure 3.16: Parallel graph

Consider Fig. 3.16, conductance G between P1 and P4 as,

$$R_{\mathrm{eq_{s(1\text{-}3)}}} = R_{12} + R_{23} = 2$$

$$R_{\mathrm{eq_{p(1)}}} = R_{\mathrm{eq_{s(1\text{-}3)}}} * R_{13}/R_{\mathrm{eq_{s(1\text{-}3)}}} + R_{13} = 2/3$$

$$R_{\mathrm{eq_{s(2)}}} = R_{\mathrm{eq_{p(1)}}} + R_{34} = 5/2$$

$$R_{\mathrm{eq_{p(2)}}} = R_{\mathrm{eq_{s(2)}}} * R_{14}/R_{\mathrm{eq_{s(2)}}} + R_{14} = 5/7$$

$$G_{\mathrm{eq}} = 1/R_{\mathrm{eq_{p(2)}}} = 7/5 = 1.40$$

From the above calculations, certain aspects can be deduced. Firstly, it indicates that the conductance across the serial path is lesser than that of the network in parallel paths. There is a significant difference when there is more number of paths between two peers. In the

first network, Fig. 3.15 there is only one serial path for P_1-P_4. In contrast, with respect to Fig. 3.16, there are three possible paths from P_1 to P_4. These observations illustrate that a series network just has a series of resistors so the conductance is calculated across only one path, whereas in a parallel network, the conductance between two nodes will have several possible paths across which it can be calculated. In addition to this, the number of resistors also impacts the conductance value. More number of resistors decreases the conductance and vice versa. This is proved in Section 4.1.3. The conductance value, therefore, depends on both parallel paths and the number of resistors. The conductance in a parallel path is more than that of a serial path. This value of conductance can be translated as the trust degree.
The scenarios stated in Section 3.1.4 can be illustrated below as, Considering Fig. 3.15 for the scenarios, the conductance can be calculated as,
Scenario 1: A new peer, Pn enters the network.
P_n connects itself randomly to two existing peers, P_1 and P_3 in the network. The conductance of this new network is then computed,

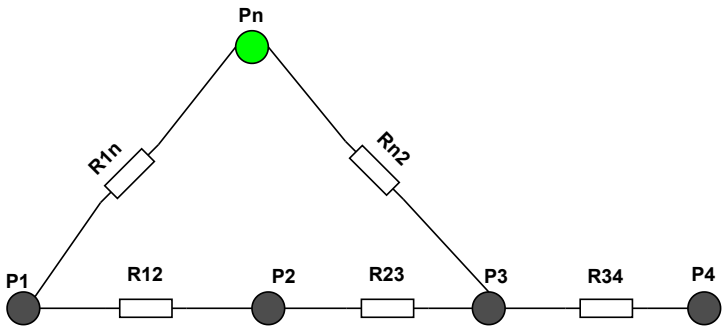


Figure 3.17: New peer enters the network

$$R_{\mathrm{eq_{s(1\text{-}3)}}} = R_{12} + R_{23} = 2$$

Convert delta to star as illustrated in Fig. 3.18, with
1. $R_{1n} = R_3$
2. $R_{n2} = R_2$
3. $R_{eq(p1\text{-}p3)} = R_1$

$$R1 = R_{1n} * R_{\mathrm{eq(p1\text{-}p3)}}/R_{1n} + R_{\mathrm{eq(p1\text{-}p3)}} + \mathrm{Rn2} = 2/3$$

$$R_2 = R_{2n} * R_{\mathrm{eq(p1\text{-}p3)}}/R_{1n} + R_{\mathrm{eq(p1\text{-}p3)}} + \mathrm{Rn2} = 2/3$$

$$R_3 = R_{1n} * R_{n2}/R_{1n} + R_{\mathrm{eq(p1\text{-}p3)}} + \mathrm{Rn2} = 2/4 = 1/3$$

$$R_{\mathrm{eq_{s1}}} = R_1 + R_3 = 2/3 + 1/3 = 1$$

$$R_{\mathrm{eq_{p1}}} = R_{\mathrm{eq_{s1}}} * R_2/R_{\mathrm{eq_{s1}}} + R_2 = 2/5$$

$$R_{\mathrm{eq}} = R_{\mathrm{eq_{p1}}} + R_{34} = 2/5 + 1 = 7/5$$

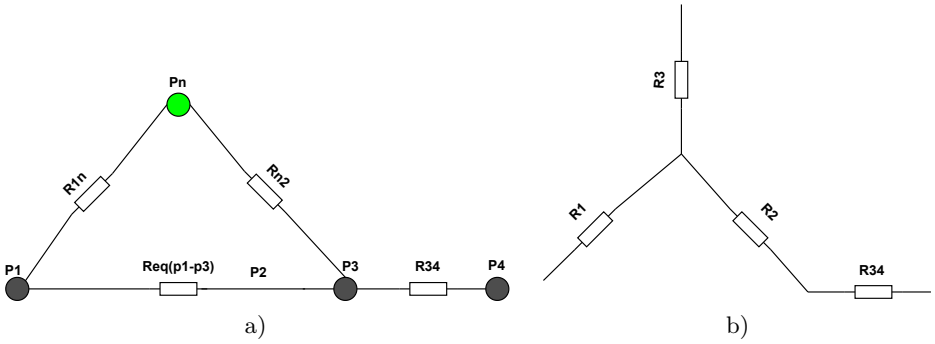


Figure 3.18: Delta-Y transformation

$$G_{\text{eq}} = R_{\text{eq}_{\text{p1}}} + R_{34} = 2/5 + 1 = 5/7$$

Scenario 2: Existing peer leaves the network
When P_4 leaves the network, the all edges of peers(P_3) connected to P_4 are deleted. In this case the resistor R_{34} is removed. The conductance of this new network as in Fig. 3.19 is calculated.

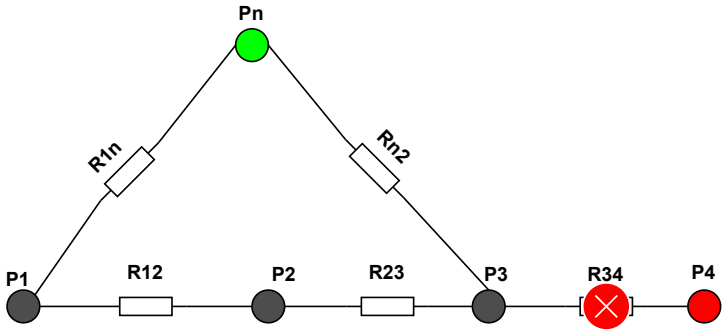


Figure 3.19: Existing peer leaves the network

$$R_{\text{eq}_{\text{s(1-3)}}} = R_{12} + R_{23} = 2$$

Convert delta to star as illustrated in Fig. 3.20, with
1. $R_{1n} = R_3$
2. $R_{n2} = R_2$
3. $R_{\text{eq(p1-p3)}} = R_1$

$$R_1 = R_{1n} * R_{\text{eq(p1-p3)}}/R_{1n} + R_{\text{eq(p1-p3)}} + R_{n2} = 2/3$$
$$R_2 = R_{2n} * R_{\text{eq(p1-p3)}}/R_{1n} + R_{\text{eq(p1-p3)}} + R_{n2} = 2/3$$
$$R_3 = R_{1n} * R_{n2}/R_{1n} + R_{\text{eq(p1-p3)}} + R_{n2} = 2/4 = 1/3$$

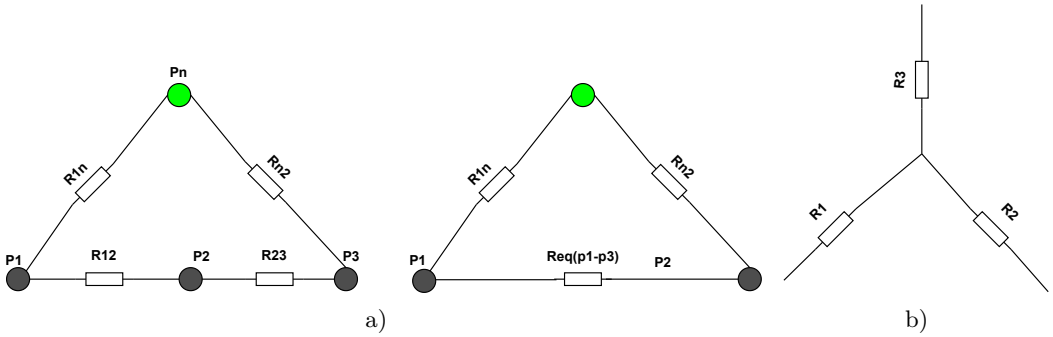


Figure 3.20: Delta-Y transformation

$$R_{\text{eq}_{s1}} = R_1 + R_3 = 2/3 + 1/3 = 1$$
$$R_{\text{eq}} = R_{\text{eq}_{s1}} * R_2/R_{\text{eq}_{s1}} + R_2 = 2/5$$
$$G_{\text{eq}} = 1/R_{\text{eq}} = 5/2$$

The above calculations illustrate the computation of conductance values in the possible two scenarios.

3.1.6 Trust model implementation

The entire research is based on trust model implementation which contains the following blocks as in Fig. 3.21. The goal is to compute trust degree between any two peers in a network. The first block involves the designing the network model. A smart home scenario is considered in which peers authenticate themselves. The establishment of trust follows the Web of Trust where one peer can authenticate other peers and other trusted peers can authenticate more peers forming a web. Each peer possesses a private key- public key pair. Peers establish trust amongst each other by exchanging their public keys. The sender sends a message encrypted by his private key termed as signature. If the receiver can decrypt the encrypted message, then the sender trusts the receiver. This kind of authentication can be applied using several cryptographic algorithms like DSA, RSA and ECDSA.

The next block involves translating the network model into a bi-directional graph. This step describes the entire network model in a simple manner with regard to the peers establishing trust among each other. The graph description involves peers as nodes, edges as trust between peers, k as the public key and t as the trust degree between the two peers. With respect to the above graph description, two possible events can occur in the network.

- Event 1: A new peer can enter the network.
- Solution(Event 1): The peer connects to two random existing trusted peers forming an edge between those two peers, therefore a network is created.
- Event 2: An existing peer can leave the network.
- Solution(Event 1): The corresponding peers delete the certificates of the leaving, resulting in the edges being removed.

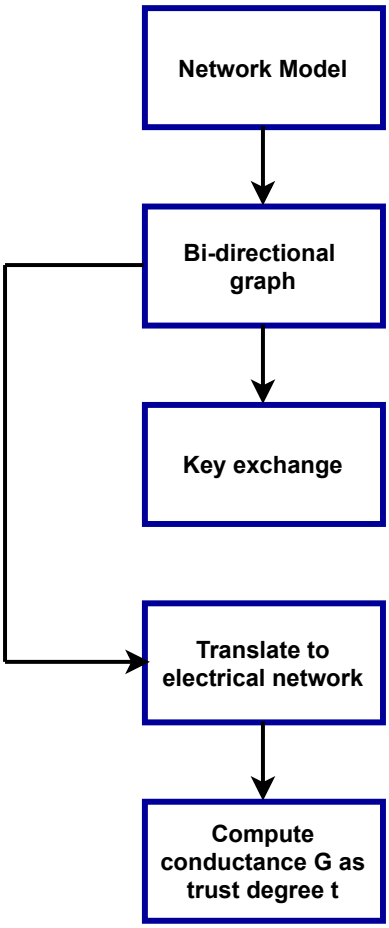


Figure 3.21: Trust Model Implementation

The bi-directional graph is converted into an electrical network of resistors. This translation helps to calculate the trust degree. As stated by (paper author), "A single serial path has a lower trust degree in contrast to one or more parallel paths which have a higher degree of trust". This property of the trust in a graph network resembles series and parallel connection of resistors. The edges of the graph are inserted with resistors of 1ohm and nodes are the same as that of the graph. The resistor values are considered as 1ohm for easy computation process. Based on the network topology, the equivalent resistance is calculated.
The next block is the key exchange block which involves authentication of peers. The peers need to exchange certificates with one another to establish trust and authenticity. This certificate exchange follows cryptographic algorithms like DSA, RSA and ECDSA. RSA performs message encryption and verification faster when compared to DSA. In contrary, DSA is faster in decrypting and signing operations , which is slower in RSA. ECDSA- an algorithm which uses Elliptic Curve Cryptography is an extension of DSA which yields higher performance compared to DSA and RSA. Similar to DSA and RSA it uses hash function for signature generation but additionally uses elliptic curves. This usage of curves consumes less time for key generation in addition to lesser key sizes compared to the other cryptographic algorithms. The algorithm first generates a private key for a peer using the hash function(SHA256) and a named curve(NISTxxxx). A signature is generated using the

peer's private key and a message which the peer uses to communicate with the target peer; peer with whom the sending peer wants to establish trust. The target peer's public using the private key which is then used to verify the signature. In other words, if the public key decrypts the encrypted message, then the signature is valid. Thereby, this exchange of certificates enables the peers to verify the authenticity among each other.
The last block is the calculation of trust degree. The computation of trust degree is on the basis of the electrical network in the third block. The equivalent resistance computed for the electrical network is used to calculate the conductance G. Conductance is defined as the amount of current flowing through the network. This can be directly compared to the trust that exists between two peers based on the electrical network topologies like Delta-Y transformation, Series and parallel connections of resistors. The conductance value G is computed using the equation,

$$G_{(\mathrm{P_x},\mathrm{P_y})} = 1/R_{\mathrm{eq}_{(\mathrm{P_x},\mathrm{P_y})}}$$

Therefore, the conductance value G is determined as the trust value between any two peers. This trust value greatly depends on the number of parallel paths $N(P)$ between the nodes and the number of intermediate nodes($\mathrm{N(I_n)}$ between the two peers, where $\mathrm{I_n}$ is the notation for Intermediate nodes; in case of electrical network intermediate nodes translates to resistors. The trust value t is more when there are more parallel paths and less intermediate nodes. Hence we can deduce that,

$$t \propto N(p)$$

$$t \propto 1/N(I_{\mathrm{n}})$$

The trust model implementation as described above is illustrated as steps using an example.

Step 1: Creation of network model: The network model is designed for a smart home scenario, in this case, the user needs to connect to the smart lighting system of the house.
As illustrated in Fig. 3.22, User 1 enters the network by connecting to the Smart bulb.

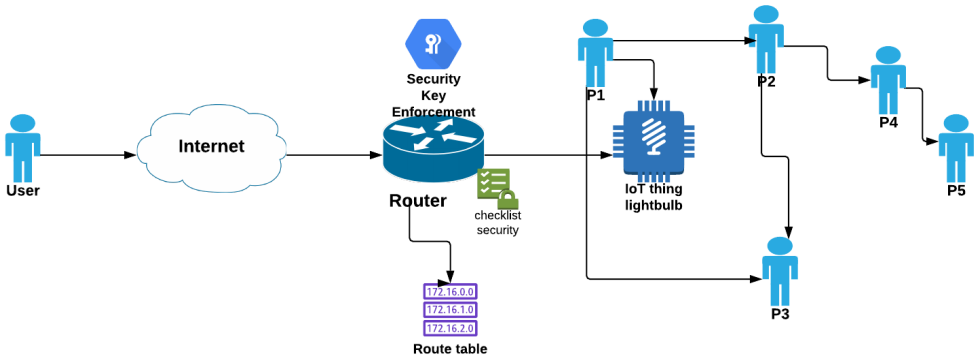


Figure 3.22: Network Model Example

The User enters the network by a security key. The user can authenticate $\mathrm{P_1}$ and $\mathrm{P_2}$. In-turn $\mathrm{P_1}$ can authenticate $\mathrm{P_3}$ and so on forming a web. This is the web of trust concept. The user is the introducer who can authenticate his immediate neighbours

as well as other peers. Once P_1 is authenticated, P_1 can become the introducer to his immediate neighbours. This is the same for all peers. In general, any authenticated peer can authenticate any other peer.

Step 2: Graph description of above trust model
The above network model is converted into a bidirectional graph in order to analyse the trust model effectively. This makes the analysis easy and readable. The graph is bidirectional because both the nodes constantly interact with each other. Every graph has certain parameters which describe it; like nodes, edges and weights. With regard to the example illustrated in Fig. 3.23, the following notations hold true as described in Section 3.1.1.

- V: Nodes of the graph
- V = P_1, P_2, P_3, P_4, P_5
- E: Edges between nodes
- E = (P_1.P_2), (P_1,P_3), (P_2,P_3),(P_2,P_4), (P_4,P_5)
- w: Weight across the edges
- w = $k_{peer(x)}$, $k_{peer(y)}$, A, t

where ($k_{peer(x)}$,$k_{peer(y)}$): public keys of interacting peers. Thereby, the above graph description parameters illustrate the network model entirely.

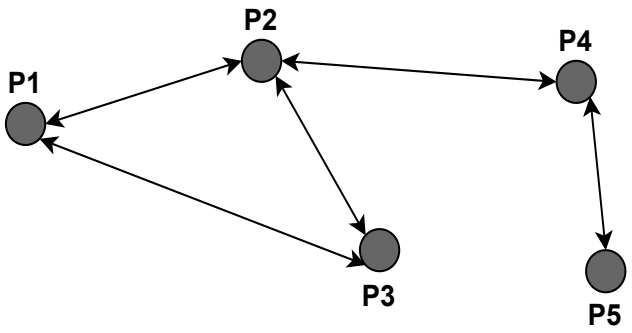


Figure 3.23: Graph Representation-Example

Step 3: Equivalent electrical network
The next step after the conversion of the network model to a bidirectional graph is the translating this bidirectional graph to an equivalent electrical network of resistors. This translation is relevant due to the following facts as stated in [5];

- Trust degree is low along a serial path.
- Trust degree is high in case of multiple parallel paths. This is because the peer can choose a particular path among several parallel paths where the number of intermediate nodes are few in number
- The number of of intermediate peers decides the trust degree. More number of intermediate nodes implies low trust degree, less number of intermediate path

indicates high trust degree.

Considering these properties of the trust model, bidirectional graph to electrical network transformation is important for trust model analysis. This can be proved with an example as illustrated in Fig. 3.24 The equivalent resistance of the network needs

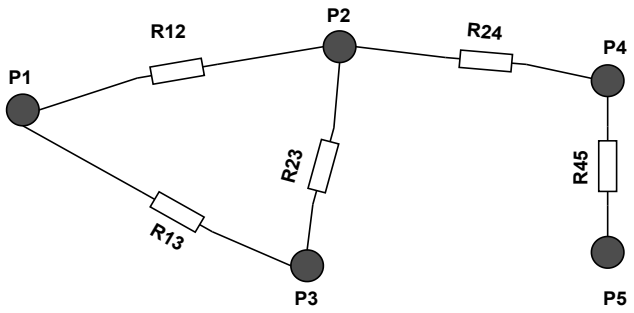


Figure 3.24: Electrical network - Example

to be calculated. Since the network has a delta network, there is a need of Delta-Y transformation for simplification of the computation. The Delta-Y transformation is illustrated below with some basic description of parameters;

- R_a : R_{13}
- R_b : R_{12}
- R_c : R_{23}

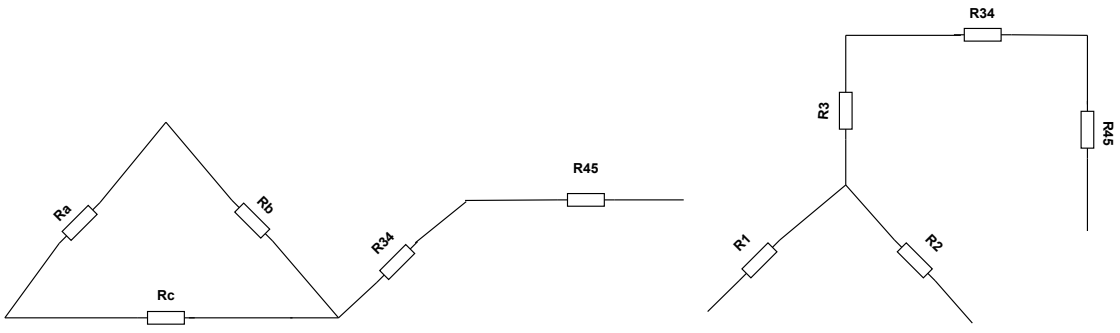


Figure 3.25: Delta-Y transformation

The graph can be slightly modified to highlight the delta network and the illustrate the Delta-Y transformation. The computation is as follows,

$$R_1 = R_a * R_c / R_a + R_b + R_c = 1/3$$

$$R_2 = (R_b * R_c) / R_a + R_b + R_c = 1/3$$

$$R_3 = R_a * R_b / R_a + R_b + R_c = 1/3$$

$$R_{\mathrm{eq_{s1}}} = R_{34} + R_{45} = 2$$

$$R_{\mathrm{eq_{p1}}} = R_{\mathrm{eq_{s1}}} * R_3 / R_{\mathrm{eq_{s1}}} + R_3 = 2/7$$

$$R_{\mathrm{eq_{s2}}} = R_{\mathrm{eq_{p1}}} + R_1 = 13/27$$

$$R_{\mathrm{eq}} = R_{\mathrm{eq_{s2}}} * R_2 / R_{\mathrm{eq_{s2}}} + R_2 = 22/27$$

Step 4: Conductance value calculation

$$G_{\mathrm{eq}} = 1/R_{\mathrm{eq}} = 27/22$$

The entire trust model from the network model to the calculation of the trust value as the conductance G is illustrated step by step. As stated the main concepts involve designing a network model for a smart home scenario and formalising the network model into a graphical representation. Further, peer authentication follows ECDSA encryption algorithm in parallel with web of trust concept.and trigger events that might occur during peer authentication and trust establishment are addressed. Finally the graph model is converted into an electrical network to compute the trust value, which is the major contribution of this research.

3.1.7 Key exchange protocol

JSON

JSON is a data serialisation language, also known as description language which is used to describe network models. When compared to other description languages like XML.
JSON is compact, ensures faster processing and is easily readable.
The above trust model id described using JSON. Firstly, the graph parameters like nodes, edges and weights are briefly explained. The trust between two parameters is also indicated based on the edge between the two peers(E(P_1, P_2)). The weight across the edges consists of the public key of the two trusted peers, the trust degree and the labels agreed upon by both the peers.

3.1.8 Algorithm for the trust model

The algorithm illustrates the trust model designed in this research with input parameters as two nodes in the graph and the output is the trust value represented as the conductance value G. The algorithm follows the steps from graph formalisation to the representation of the graph as an electrical network. Further on, the electrical network is simplified based on the rules of electrical engineering like Delta-Y transformation, series and parallel topologies of resistors. Finally, the conductance is calculated as the reciprocal of the equivalent resistance. This conductance is represented as the trust value between any two nodes in the network.

Algorithm 1 Trust Model
Input : Parameters of Bi-directional graph - (V_x,V_y)
Output : Conductance value G as trust degree t

G : {V, E, W}
V : {P_1, P_2, P_3,..... P_n}
W : {K_{n_x},K_{n_y}, t, A}
Convert corresponding model to electrical network E_N
$\mathbf{E_N}$: {N, E, W_{E_N}} where
$\mathbf{W_{E_N}}$: {R} → *Resistor*
E_N *contains nodes in the orientation* $(P_1P_2 - P_2P_3 - P_3P_1) \implies D_N$ - *Delta network*
Ra, Rb, Rc → *resistors in the delta network* D_N
E_N *contains single series path in the orientation* → $(P_1 - P_2 - P_3 - P_4) \implies \text{Series}_N$
E_N *contains multiple parallel paths in the orientation* → $(P_1 - P_2 - P_3 - P_4, P_1 - P_5 - P_4) \implies Parllel_N$
for E_N contains D_N **do**
    Perform Delta - Y transformation
    $R_1 = R_a * R_b / R_a + R_b + R_c$
    $R_2 = R_b * R_c / R_a + R_b + R_c$
    $R_3 = R_a * R_c / R_a + R_b + R_c$
    $Ra, Rb, Rc, R1, R2, R3 \in R$
    if E_N contains series resistors; Series_N **then**
    $R_{eq} = R_1 + R_2 + R_3 + ... R_n$; where $n \in N$
    $\mathbf{G = 1 / R_{total}}$;
    else if E_N contains parallel resistors Parallel_N **then**
    $R_{eq} = (R_x * R_y)/(R_x + R_y)$; where $R_x, R_y \in R$
    $\mathbf{G = 1 / R_{total}}$;
    break;
    end if
end for
return G;
if E_N contains series resistors; Series_N **then**
$R_{eq} = R_1 + R_2 + R_3 + ... R_n$; where $n \in N$
$\mathbf{G = 1 / R_{total}}$; .
else if E_N contains parallel resistors Parallel_N **then**
$R_{eq} = (R_x * R_y)/(R_x + R_y)$; where $R_x, R_y \in R$
$\mathbf{G = 1 / R_{total}}$;
end if
return G;
Update **W** : {K_{n_x}, K_{n_y}, G, A}

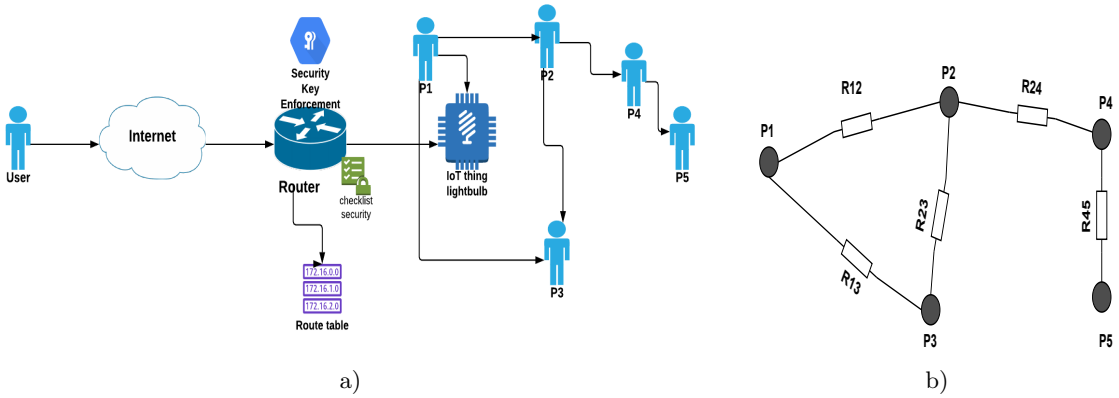


Figure 3.26: Mapping trust model to proposed electrical network

3.2 Trust Model Summary- Map of trust model to Electrical Network

The network model is mapped to the electrical network to make trust value calculation simple. The network model illustrated in Fig. 3.22 describes a scenario where a user enters a network by using a key such as a password to connect to the device. This serves as the authentication for the user. The user can further authenticate other peers following the web of trust concept. As described, Web of Trust involves the process where every authenticated peer can authenticate every other peer within the network. This authentication is done using key exchange and encryption algorithms like DSA and ECDSA.
The network model can be easily mapped to the proposed approach of Electrical network for the trust model. In the network model the peers communicate with each other by exchanging messages and public keys and establish trust by agreeing upon a common label. In the existing approaches, trust is defined by percentage values. This value does not have any specific meaning since it differs based on perspective. For example, one author might consider 98% has higher trust value whereas another author might consider it as a lower trust value. In these approaches the value of trust is ambiguous. To overcome this ambiguity, the value of trust must be computed using a new approach. The new approach is an electrical network which is proposed as a trust model to compute the trust values. In the electrical network, the peers are represented as nodes and the trust value is computed as the conductance value. In this approach, trust is defined as the amount of current flowing through a pair of nodes.
Considering the above two diagrams, in the network model the peers exchange certificates to establish trust. This is depicted in the electrical network where the peers exchange their public keys along with the trust value as the conductance G and further agree upon a common label.

CHAPTER 4

Thesis Outcome

4.1 Evaluation

4.1.1 Overhead Evaluation

Overhead is one of the major factors that need to be considered in communication networks. It is an extra resource that is used to perform a task like time, energy, cost or memory. This plays a very important role in analyzing and evaluating the performance of a system. The trust model proposed in this research is evaluated for the overhead that exists in the network. The overhead is calculated during key exchange among the peers. The overhead for the data is considered to be the public keys of both peers since the two peers exchange their respective public keys. In addition to the public keys, they also exchange their trust values and a common label. This implies that the overhead for data exchange would be the public key size, the trust value and the label. This is the weight of the edge in the graph formalisation as described in Section 3.1.1 which is given as w: k_x, k_y, t, A. This is the overhead in the general case where two peers in the network are communicating with each other to establish trust. The public key size of ECDSA is 128 bit, t and A can be considered as a 8 bit value each. So the overall overhead would be (128, 128, 8, 8)bits.
The digital signature is also considered as an overhead because the two peers need to sign the certificate and verify it. The digital signature also contains the message encrypted, so the signature size can be considered as message overhead.In ECDSA the length of the signature is 96 bits. So the overhead, in this case, is 96 bits.
When a new peer enters the network, the peer exchanges certificates with two other peers. In this case, the new peer sends his public key, a proposal of trust value and a label. The existing peer then sends his public key along-with his trust value and a label. If the new peer decodes the message then trust is established and an acknowledgement of 1 byte is sent to the new peer. This is the overhead due to a new entering the network. New peer P_n sends a message with overhead of (128, 8, 8) bits and the peer with whom this peer exchanges certificate again sends a message with overhead (128, 8, 8) bits in addition to 1 byte of acknowledgement that the trust has been established.
On the other hand, when an existing peer leaves the network, the public key of the leaving node is removed, so overall there is no overhead in this process because nothing is being

exchanged between any of the peers, rather only the certificate of leaving node is deleted. In this case, no message of the form (k_x, k_y,t , A) is sent. So the overhead is 0.
Based on the algorithm, the graph parameters that are given as input need to be stored in a database. The nodes, edges and weight need to be stored for the algorithm to execute. This storage of data in the database and acquiring the data back from the database is considered as overhead since it results in storage costs and computational costs. However, the detailed description of this is not the scope of this research.
As implemented in Python 3, DSA generates a key of size 1024 whereas the same key in ECDSA is just 128 bits. The private key generated in ECDSA is 64bits whereas in DSA it is 1024 bits. This significant difference impacts the performance of the system. In addition to this, the processing time of DSA and ECDSA is mentioned which stands as evidence that ECDSA performs faster when compared to DSA. Even in terms of security, ECDSA is more trustworthy than DSA or RSA which is majorly dependent on the key size. Even though DSA has a larger key size, factorizing large numbers has become convenient with time and technology, thereby infringement of DSA is possible. This is not the case in breaking an ECDSA key since ECDSA requires solving the Elliptic Curve Discrete Logarithmic Problem which to date has no precise solution. Hence, hacking a key of DSA or RSA is more probable than a key generated using ECDSA.

4.1.2 Attack Tree Analysis

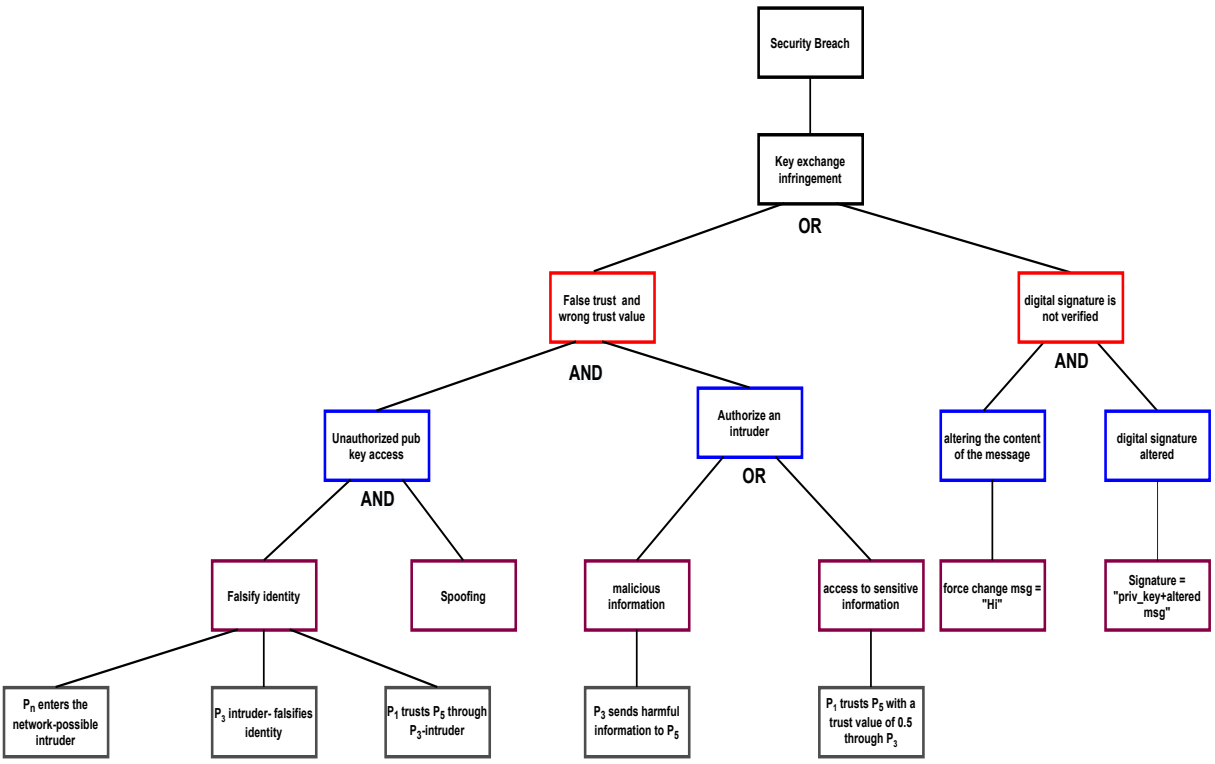


Figure 4.1: Attack Tree

Attack trees are tree structures designed to analyze the possible attacks that can occur so as to compromise a system. This plays a very important role in threat analysis which poses a major challenge in Internet security. It is necessary to analyze all possible threats that can affect the system in a way that the analysis prevents impairment of the functionality as a whole. In the proposed model, the attack tree illustrated below considers all the possible threats that can occur that every point in the trust model.
The attack is structured in a way that the leaf nodes are the events that trigger the attacks mentioned in the corresponding parent nodes based on examples from Fig. 4.6 and Fig. 4.4 In other words, for a particular attack to occur the events mentioned in the leaf should take place.
In this research, a security breach is the major attack that can compromise the entire system, thereby this is the root node. The attack tree illustrated in Fig. 4.1 is described as attack scenarios where the contributors to the respect attack are briefly explained.
Attack Scenario 1: A new peer P_n, enters the network. The peer exchanges certificates with two trusted peers in the network. This new peer can be an intruder who can falsify his identity by claiming to be a trusted peer. Falsifying identity by claiming to be trustworthy enables the intruder to get access to the trusted peer's certificate. This results in a false establishment of trust in-turn resulting in the miscalculation of trust values between the trusted peers and the intruder. This is one attack which causes a security breach, thereby compromising the trust model.
Attack Scenario 2: An existing peer in the network can be an intruder by falsifying his identity, thereby contributing to a similar attack as in Attack scenario 1.
Attack Scenario 3: A man-in-the-middle attack can occur when an existing peer in the middle of two trusted peers is an intruder. In this regard, the intruder can portray a trusted peer as untrustworthy or claim an untrustworthy peer to be a trusted peer. In both cases, the problem of miscalculating trust value and establishing false trust arises, hence compromising the system. Attack scenario 4: An intruder P_3 in the network can send malicious information to another peer which is a part of the above attack scenarios. This contributes to the overall security breach. Hence the tree has an AND node between unauthorized key access and authorizes an intruder since they together contribute to false trust establishment.
Attack Scenario 5: In this scenario, P_1 trusts P_5 with a trust degree of 0.5. P_3 might be an intruder who has acquired a false trust value of 0.5 whereas another peer P_4 is trusted by P_1 with a trust degree of 1. In this case, P_3 who is an intruder has access to certain sensitive information. Instead, P_4 who is trustworthy and not identified as an intruder should have access to the information. This attack scenario also combines with the top three attack scenarios mentioned to compromise the system as a whole.
Attack Scenario 6: An intruder alters the message which is encrypted to form the digital signature. In this case, the digital signature is altered and cannot be verified. This leads to another security breach which in this research is prevented since the message is generated randomly and no peer can manually alter it.
The main attack is the security breach which occurs due to key exchange infringement. This implies that security is mainly compromised during the exchange of keys between peers. The attack caused during key exchange maybe because of either false trust establishment resulting in wrong trust value computation or due to repudiation of the digital signature. At this point in the attack tree, "OR" node is used to depict that either one of the attacks needs to occur for parent attack, that is key exchange infringement to occur. The false

establishment of trust is triggered by unauthorized public key access and authenticating an intruder. Since these two events together contribute to the false trust establishment, "AND" node is necessary to represent that the combination of the leaf nodes contributes to the parent attack. Falsifying identity or claiming to be an authorized peer and spoofing leads to unauthorized key access. When a peer claims to be someone he/she is not, in that case, he/she gains access to the public key of a trusted peer. In addition, he/she spoofs certain information like exchanging his/her trust value and label to gain trust leading to unauthenticated key access. An "AND" node is used in this case similar to the previous one to indicate that both events are required to trigger unauthorized key access. The leaf nodes depict the possible ways the attack can occur through an example from Fig. 4.4. Falsifying identity can occur when a new peer P_n enters the network who might be an intruder and claims himself to be trustworthy. Another way to falsify identity could be that a peer P_3 in the network could be an intruder claiming that another trustworthy peer is non-trustworthy. This mostly does not occur since the model checks for this example scenario during key exchange. Authorizing an intruder which contributes to the key infringement can occur due to either transmitting malicious information between peers P_3 and P_5 in the above example. If P_3 sends P_5 some malicious information, it implies that transmitting any information is possible only when an intruder is authorized. Access to sensitive information can also occur when an intruder is proved to be trustworthy. In this regard if P_1 trusts P_5 with a degree of 0.5 through P_3 who might be an intruder, there is a possibility that P_1 gives P_5 the access to some sensitive information because P_3 has recommended P_5 to be trustworthy. In this case, P_5 might also be an intruder, so such an event occurs when P_3 who is an intruder is authorized. Thereby one of these two examples can result in authorizing an intruder that is either by sending malicious information or by accessing sensitive data. An "OR" node is used since one of the events is enough to trigger the authorization of an intruder. The second attack that can lead to key infringement is the repudiation of the digital signature. Altering the message by force manually and thereby altering the digital signature together can lead to repudiation of the digital signature. This implies that an "AND" node is required since both these events together trigger the "digital signature not verified" attack. This attack is prevented in the current model since ECDSA handles it by default.

4.1.3 Validity of proposed electrical network model

The proposed electrical network approach for computing trust value proves to be efficient since it decreases the computational effort and time. In addition, it gives more meaning to the trust value as a metric. The evaluation of the electrical model is performed by illustrating examples of different network scenarios as electrical networks. The trust values are computed for each scenario, in addition to this trust computation, the trust value for changes in the network when a new peer enters the network or when an existing peer leaves the network is computed. The examples serve as a proof of the above research and satisfy the properties of trust and electrical network mapped together - series and parallel trust to series and parallel connection of resistors. The below examples are used to illustrate the validity of this approach.

As described earlier, the trust degree between two peers is expressed by the conductance

value G which the inverse of the total resistance R_{eq}. In a network, the trust between two peers mainly depends on two factors,

- Number of intermediate nodes
- Presence of parallel connection of resistors

Considering the examples, the number of intermediate nodes, which is number of resistors in this case, impacts the trust value. If the number of resistors in the network between two peers is more, then the trust value is low. This is true because, as the number of intermediate peers between two peers, P_x and P_y increases, the concept of referential trust arises which decreases the trust that P_x places in P_y. Referential trust means that one trusted peer refers another peer he trusts. Referential trust results in lower trust degree than direct trust [1]. For example, A trusts B and B trusts C . So B refers C to A so that A can establish trust with C. In this case, the trust that A places in C will be lower than that which A places in B.
The presence of parallel connection of resistors represents that there are multiple parallel paths of intermediate peers between the two peers who want to establish trust. This thereby agrees with the property stated above that a single serial path yields lower trust value while parallel paths leads in higher trust value [5].
For the validation of the model, two examples are considered. With regard to Example 1 as in Fig. 4.2, trust between peers P_1 and P_3 is computed.

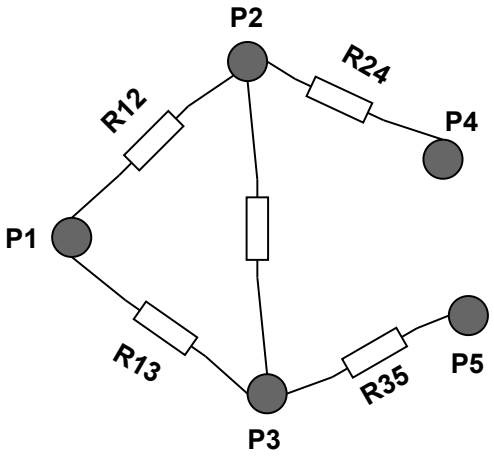


Figure 4.2: Electrical Network Example 1

P_1 and P_3

Considering only the part of the network that covers all possible peers through P_1 to P_3, the trust of the network is computed for the network in Fig. 4.3 as follows, where

Consider the delta network enclosed by nodes P_1, P_2 and P_3
Apply Delta-Y transformation,

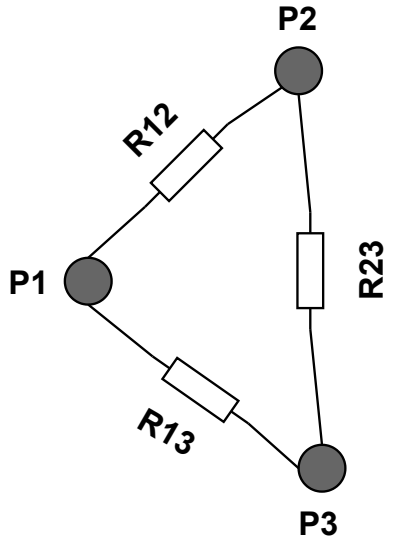


Figure 4.3: Electrical Network Example 1 (P1-P3)

$$R_1 = (R_{13} * R_{23})/(R_{12} + R_{13} + R_{23}) = 1/3$$
$$R_2 = (R_{12} * R_{23})/(R_{12} + R_{13} + R_{23}) = 1/3$$
$$R_3 = (R_{12} * R_{13})/(R_{12} + R_{13} + R_{23}) = 1/3$$
$$R_{\text{series}_1} = R_3 + R_2 = 2/3$$
$$R_{\text{total}_{\text{P1-P3}}} = (R_{\text{series}_1} * R_{\text{series}_2})/(R_{\text{series}_1} + R_{\text{series}_2}) = 2/9$$
$$G_{\text{total}_{\text{P1-P3}}} = 1/R_{\text{total}_{\text{P1-P3}}} = 9/2 = 4.5$$

Possible intermediate peers that occur between P_1 and P_3: 2 Paths
Number of resistors = 3
Path$_1$: P_1-P_3

$$R_{\text{total}} = R_{13} = 1$$
$$G_{\text{total}} = 1/R_{\text{total}} = 1$$

Path$_2$: P_1-P_2-P_3

$$R_{\text{total}} = R_{13} + R_{36} = 1 + 1 = 2$$
$$G_{\text{total}} = 1/R_{\text{total}} = 1/2 = 0.5$$

In the above example where P_1 trusts P_3, there are two possible paths. The overall trust from P_1 to P_3 is 4.5. In a network of 3 peers, for P_1 to trust P_3 there are two possible paths, one is direct trust and the other is referential trust.Therefore this is the case where maximum trust can be achieved. The first path exhibits direct trust whereas the second path has one intermediate node, exhibiting referential trust. Therefore the trust value of Path_1 - 1 is lesser than that of Path_2 - 0.5

P_1 and P_5

Considering only the part of the network that covers all possible intermediate peers from P_1 to P_5, the trust of the network as illustrated Fig. 4.4 is computed as follows: Consider

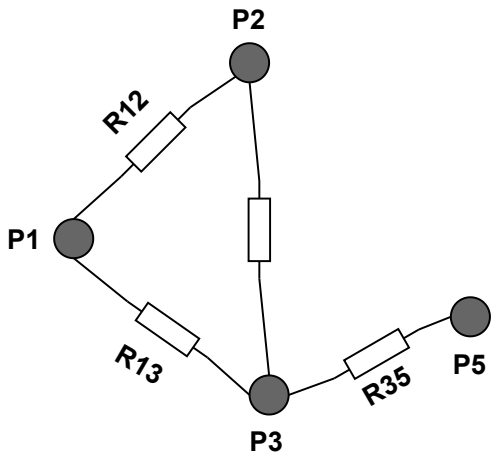


Figure 4.4: Electrical Network Example 1 (P1-P5)

the delta network enclosed by nodes P_1, P_2 and P_3
Apply Delta-Y transformation,

$$R_1 = (R_{13} * R_{23})/(R_{12} + R_{13} + R_{23}) = 1/3$$

$$R_2 = (R_{12} * R_{23})/(R_{12} + R_{13} + R_{23}) = 1/3$$

$$R_3 = (R_{12} * R_{13})/(R_{12} + R_{13} + R_{23}) = 1/3$$

$$R_{\text{series}_1} = R_3 + R_2 = 2/3$$

$$R_{\text{parallel}_1} = (R_{\text{series}_1} * R_{\text{series}_2})/(R_{\text{series}_1} + R_{\text{series}_2}) = 2/9$$

$$R_{\text{total}_{\text{P}_1\text{-P}_3}} = R_{\text{parallel}_1} + R_{35} = 2/9 + 1 = 11/9$$

$$G_{\text{total}_{\text{P}_1\text{-P}_3}} = 1/R_{\text{total}_{\text{P}_1\text{-P}_3}} = 9/11 = 0.818$$

Possible intermediate peers that occur between P_1 and P_5: 2 Paths
Number of resistors = 4
Path1: P_1-P_3-P_5

$$R_{\text{total}} = R_{13} + R_{35} = 2$$

$$G_{\text{total}} = 1/R_{\text{total}} = 1/2 = 0.5$$

Path2: P_1-P_2-P_3-P_5

$$R_{\text{total}} = R_{12} + R_{23} + R_{35} = 1 + 1 + 1 = 3$$

$$G_{\text{total}} = 1/R_{\text{total}} = 1/3 = 0.33$$

In the example where P_1 trusts P_5, there are only two possible paths. The overall trust is from P_1 to P_5 is 0.818. The first path has one intermediate node, so the trust value is 0.5. The second path has three intermediate peers, so the trust value is 0.25. Since there are 4 peers and two possible paths the trust value is less than the trust value between P_1 and P_3 Considering Example 2 as illustrated Fig. 4.5

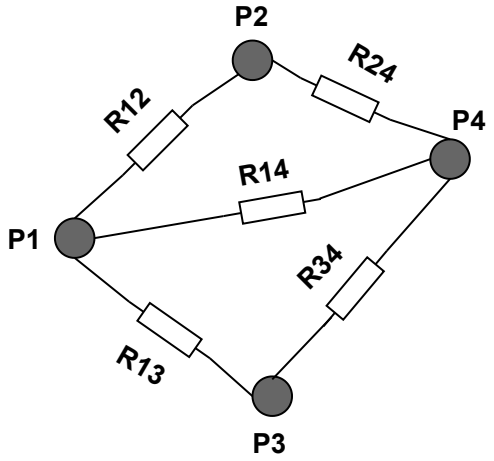


Figure 4.5: Electrical Network Example 2

P_1 and P_4

Considering only the part of the network that covers all possible paths from P_1 to P_4, the trust degree of the network as illustrated Fig. 4.6 is computed as follows: Consider the delta

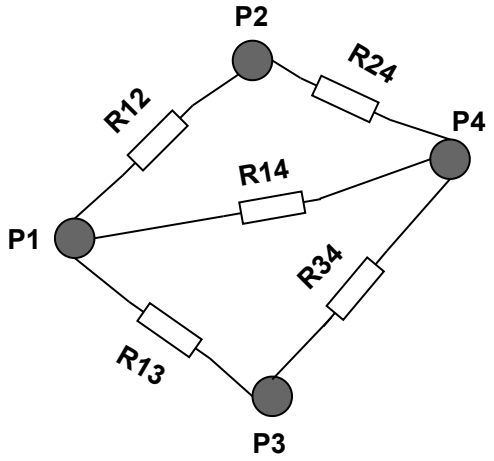


Figure 4.6: Electrical Network Example 2 (P1-P4)

network enclosed by nodes P_1, P_2 and P_4

Apply Delta-Y transformation,

$$R_1 = (R_{12} * R_{14})/(R_{12} + R_{24} + R_{14}) = 1/3$$
$$R_2 = (R_{14} * R_{24})/(R_{12} + R_{24} + R_{14}) = 1/3$$
$$R_3 = (R_{12} * R_{24})/(R_{12} + R_{24} + R_{14}) = 1/3$$
$$R_{\text{series}_1} = R_1 + R_{13} = 1 + 1/3 = 4/3$$
$$R_{\text{series}_2} = R_2 + R_{34} = 1 + 1/3 = 4/3$$
$$R_{\text{parallel}_1} = (R_{\text{series}_1} * R_{\text{series}_2})/(R_{\text{series}_1} + R_{\text{series}_2}) = 16/24$$
$$R_{\text{total}_{\text{P}_1\text{-P}_4}} = R_{\text{parallel}_1} + R_3 = 16/24 + 1/3 = 1$$
$$G_{\text{total}_{\text{P}_1\text{-P}_4}} = 1/R_{\text{total}_{\text{P}_1\text{-P}_4}} = 1$$

Possible paths for computing trust between P_1 and P_4: 3
Number of resistors = 5
Pth1: P_1-P_2-P_4

$$R_{\text{total}} = R_{12} + R_{24} = 1 + 1 = 2$$
$$G_{\text{s}} = 1/R_{\text{total}} = 1/2 = 0.5$$

Pth2: P_1-P_4

$$R_{\text{total}} = R_{14} = 1$$
$$G_{\text{s}} = 1/R_{\text{total}} = 1/1 = 1$$
$$R_{\text{total}} = R_{13} + R_{34} = 1 + 1 = 2$$
$$G_{\text{s}} = 1/R_{\text{total}} = 1/2 = 0.5$$

In the example where P_1 trusts P_4, there are three possible paths where one path is redundant. In other words Path_3 is same as Path_1. The overall trust between P_1 and P_4 is 1. The number of intermediate nodes in Path_3 is 1 so the trust value is 0.5 whereas there are no intermediate values in Path2 therefore the trust value is 1 which is greater than that of P_1. Here P_1 and P_3 are referential trusts since there is only one peer between P_1 and P_4 and P_2 is direct trust.

With the illustration of the above examples, the statements about referential trust and direct trust can be validated. From Fig. 4.3, trust value between P_1 and P_3 is is computed. In this case maximum trust that can be achieved between two peers in a network since there are only 3 peers and there are two possible paths to establish trust between P_1 and P_3. This is the best case scenario that can occur in a network. There is a combination of direct trust and referential trust with only 1 intermediate node between P_1 and P_3. From Fig. 4.6, P_1 needs to establish trust with P_4. This can be done in 3 possible ways. One is through direct trust and the remaining two is using referential trust. The trust value for direct trust is 1 whereas that of referential trust is 0.5. The number of intermediate nodes plays an important role as well as illustrated in Fig. 4.4 in which the trust between nodes P_1 and P_5 through P_2 and P_3 is 0.33 whereas the trust between P_1 and P_5 through P_3 is

0.5. This is because, in the first case there are two intermediate peers and in the second case there is only 1 intermediate peer. The examples depict that the series and parallel resistances can be compared to the series and parallel properties of trust from the graph for an efficient trust model.

Validation based of possible events that might occur

The validation of the proposed electrical network is done for the possible events that might occur in communication networks for trust establishment. The trust between P_1 and P_4 in the normal scenario is calculated as ,

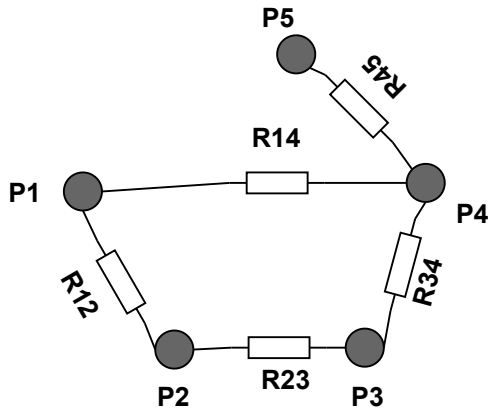


Figure 4.7: Electrical Network Example: Scenarios

$$R_{\text{series}} = R_{12} + R_{23} + R_{23} = 1 + 1 + 1 = 3$$
$$R_{\text{total}_{\text{P}_1\text{-P}_4}} = (R_{\text{series}} * R_{14})/(R_{\text{series}} + R_{14}) = 3/4$$
$$G_{\text{total}_{\text{P}_1\text{-P}_4}} = 1/R_{\text{total}_{\text{P}_1\text{-P}_4}} = 4/3 = 1.33$$

Event 1: When a new peer enters P_n the network, he/she exchanges certificates with two other trusted peers creating an edge between those two peers. The trust value for the updated network as illustrated Fig. 4.8 after the entry of the node is calculated as below.

$$R_{\text{series}_1} = R_{12} + R_{23} + R_{23} = 1 + 1 + 1 = 3$$
$$R_{\text{series}_2} = R_{1\text{n}} + R_{\text{n}5} + R_{45} = 1 + 1 + 1 = 3$$
$$R_{\text{parallel}} = (R_{\text{series}_1} * R_{14})/(R_{\text{series}} + R_{14}) = 3/4$$
$$R_{\text{total}_{\text{P}_1\text{-P}_4}} = (R_{\text{parallel}} * R_{\text{series}_2})/(R_{\text{series}} + R_{\text{series}_2}) = 9/15$$
$$G_{\text{total}_{\text{P}_1\text{-P}_4}} = 1/R_{\text{total}_{\text{P}_1\text{-P}_4}} = 15/9 = 1.66$$

The trust value for the updated network after a new node P_n enters the network is computed. Since there is an extra parallel path, the trust value has increased to 1.66 from 1.33 as in the normal scenario. Event 2: When an existing peer P_4 leaves the network, the certificates

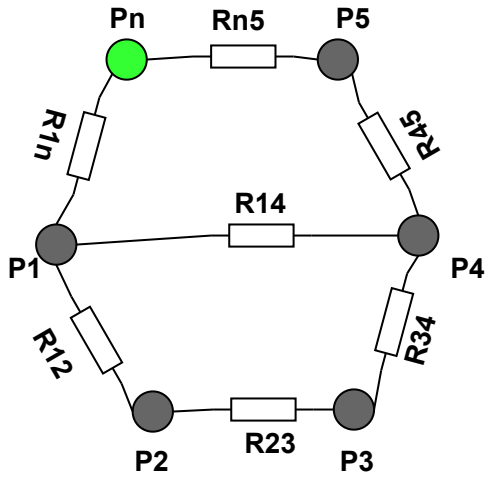


Figure 4.8: Electrical Network Example: New node enters the network

of the leaving peer is removed, thereby the edges with respect to this peer are deleted. In addition to this the resistors associated with this peer are removed. The trust value of the updated network as illustrated Fig. 4.9 after the existing peer leaves the network is computed as,

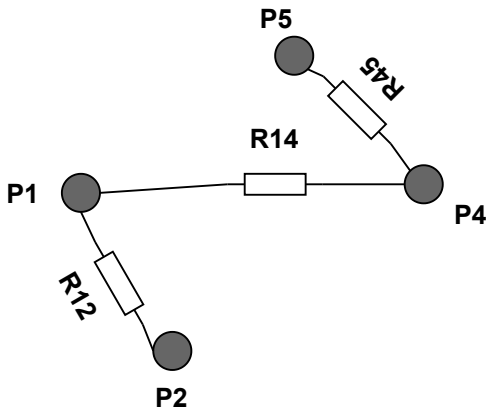


Figure 4.9: Electrical Network Example:Existing node leaves the network

$$R_{\text{total}_{\text{P}_1\text{-P}_4}} = R_{14} = 1$$

$$G_{\text{total}_{\text{P}_1\text{-P}_4}} = 1/R_{\text{total}_{\text{P}_1\text{-P}_4}} = 1$$

When an existing node leaves the network, multiple parallel paths get omitted , thereby decreasing the trust value. In the trust value has decreased to 1 from 1.33 with regard to the normal scenario.
The evaluation of this research mainly consists of three parts. Part 1 is the Overhead evaluation which deals with the overhead introduced during trust establishment. In the trust model, three kinds of overhead are introduced. First, the data overhead which is the weight across the edge of the graph, second the message overhead which is the length of the digital

signature and third is the database overhead which deals with the storage and computation costs of graph parameters stored in the database. All the overhead is illustrated based on ECDSA.

Part 2 is the attack tree analysis which is the security analysis of the thesis. The attack tree describes all the possible attacks that can occur in the model so as to compromise the system. The possible attacks are falsifying identity of a peer and repudiation of digital signature. These two attacks together contribute to key exchange infringement. However the repudiation of digital signature is prevented in the proposed model due to the usage of . The leaf nodes of the tree are represent example scenarios of the attacks illustrated in the attack tree.

Part 3 is the validation of the proposed electrical network. Two examples are illustrated to compute the trust value of randomly created electrical networks to prove the validity of mapping the series and parallel trust properties into the electrical model of series and parallel connection of resistors. Therefore to compute the trust value, the equivalent resistance if calculated first based on the electrical network topologies like series, parallel and Delta networks. Finally, the conductance value is calculated as the trust value which is the reciprocal of the equivalent resistance value.

CHAPTER 5

Conclusion

In this section, the major goal of the thesis is addressed. The existing methods are mentioned and the corresponding drawbacks are analysed. The new proposed method is explained briefly and the flow of the entire process is described as the thesis contribution. Finally, the evaluation techniques and the associated results are summarized as the expected results.

5.1 Summary

The major task of the research is to model a trust network and to find an approach to calculate the trust value between two communicating entities. The existing researches follow the typical graph analysis model and uses heuristics to calculate the trust value. The typical representation of trust values is represented as percentage values or as integer values between the interval [0,1]. These trust metrics also consider some factors like oscillating behaviour of peers and dynamic changes in the network. The computation of trust value based on these factors as percentages or as integers between [0,1] creates ambiguity since it is not clear what these values mean in reality. To overcome these drawbacks, the goal of this research is to model a trust network and the research follows the flow as mentioned below.

A network model is first created for an example smart home scenario. This model is then formalized into a graph network with nodes as peers and edges connecting the nodes. The weight across the edges is a combination of the public keys of the communicating peers, the trust value, and a unique label. The edge between the nodes depict that the peers communicate with each other by exchanging public keys are finally agree upon a common label. The key exchange follows the concept of a web of trust where peers can authenticate each other. For instance, a peer enters a network and authenticates his immediate neighbours, further the authenticated peers can exchange certificates and authenticate his neighbours forming a web. The major advantage is that there is no central authority involved in the certificate exchange. The encryption algorithm used for digital signature is ECDSA which has advantages like low key size, low processing time, and intrusion resistant. The graph network is then translated into an electrical network where each edge is inserted with re-

sistors. The resistance value is set to 1 ohm for easy computation. The degree of trust between two peers is compared to the conductance between two nodes which in electrical engineering is the amount of current that flows through them. This comparison is valid from the perspective of series and parallel trust. As cited, trust along a single series path is lesser when compared to that of multiple parallel paths and trust degree decreases with the increase in the number of intermediate nodes. Based on these properties of trust, translating the graph to an electrical network to calculate trust is efficient.

The validation of the electrical network proves as evidence for the calculation of trust as efficient in this research. Each of the examples illustrates the properties of trust and the topologies of the electrical network. The trust value is calculated in each of the cases by calculating the equivalent resistance between two peers. For example if P_1wants to establish trust with P_2, then the equivalent resistance, R_{eq} between P_1 and P_2 is calculated. After the computation of equivalent resistance, the conductance value G which is the reciprocal of the equivalent resistance between two nodes is calculated. This is the trust value between any two peers. This trust value is calculated in the examples to prove that the series and parallel properties of trust mapped to series and parallel connection of resistances. This validation of examples gives the expected results which are that a single series of resistances yields lesser trust values than a parallel connection of resistances. In addition to this, the number of intermediate peers also play an important role in trust value, the more the number of intermediate nodes, the lesser the trust value and vice versa.

An attack tree analysis is performed to identify the possible attacks that can compromise the system. The analysis is described as attack scenarios that define each attack from the leaf node – an example of an attack to the root node- the main threat. The "AND" and "OR" nodes depicts whether all child node events need to occur for the attack at the parent node to take place or any one of the child node events is enough for the parent node attack. The attack in this research focuses mainly on the security breach which is caused due to key exchange infringement. This is the main attack that can compromise the entire system. The key exchange infringement is further caused by either Falsifying identity or due to the repudiation of a digital signature. The leaf nodes of the attack tree illustrate the example cases that cause these attacks based on the graph examples
An overhead evaluation illustrates the overhead that is introduced between two peers while establishing trust. The overhead in the case of the trust model is the weight function w given by

$$w : k_x, k_y, t, A$$

The public key size in case of ECDSA is 128 bits and the trust and label are considered to be 8 bits each. The digital signature size is considered as message overhead which is 96 bits in case of ECDSA.
The signature size is considered as message overhead because the signature consists of the message that is concatenated to the private key of the ECDSA. The storage of graph parameters into a database and access to these parameters from the database is also considered as an overhead due to storage and computation costs. However, this aspect of overhead is not covered in the scope of this research.

5.2 Future Work

The open tasks after the research on this thesis would be to consider the overhead caused due to storage and acquisition of graph parameters from the database. The storage cost and the computation cost needs to be calculated. This aspect needs to be addressed since database overhead causes a huge impact on the overall performance of the system. Another task would be to provide a set of access rights based on the labels as represented in the weight function. This is important in the field of Smart Homes where different people have a different set of access rights to the device. Te next possible area of research would be to consider certain parameters like oscillating behaviour of the peers through the electrical network approach.

Bibliography

[1] Audun Jøsang, Elizabeth Gray, and Michael Kinateder. Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems*, 4(2):139–161, 2006.

[2] Florina Almenarez, Andres Marin, Celeste Campo, and Carlos Garcia. PTM : A Pervasive Trust Management Model for Dynamic Open Environments. *First Workshop on Pervasive Security, Privacy and Trust, PSPT2004 in conjuntion with Mobiquitous*, 2004.

[3] Eric Zeng, Shrirang Mare, Franziska Roesner, and Paul G Allen. End User Security and Privacy Concerns with Smart Homes End User Security & Privacy Concerns with Smart Homes. *Proceedings of the Thirteenth Symposium on Usable Privacy and Security*, (Soups):255–272, 2017.

[4] Tyrone Grandison and Morri Sloman. A survey of trust in internet applications. *IEEE Communications Surveys & Tutorials*, 3(4):2–16, 2009.

[5] G. Caronni. Walking the Web of trust. *Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE*, 2000-Janua:153–158, 2000.

[6] Hisashi Mohri, Ikuya Yasuda, Yoshiaki Takata, and Hiroyuki Seki. Certificate chain discovery in Web of trust for ad hoc networks. *Proceedings - 21st International Conference on Advanced Information Networking and Applications Workshops/Symposia, AINAW'07*, 1:479–485, 2007.

[7] Jennifer Golbeck, Bijan Parsia, and James Hendler. Trust networks on the Semantic Web. *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 2782:238–249, 2003.

[8] Zainab M. Aljazzaf, Miriam A.M. Capretz, and Mark Perry. Trust bootstrapping services and service providers. *2011 9th Annual International Conference on Privacy, Security and Trust, PST 2011*, (January):7–15, 2011.

[9] Kemal Bicakci, Bruno Crispo, and Andrew S. Tanenbaum. How to incorporate revocation status information into the trust metrics for public-key certification. *Proceedings of the ACM Symposium on Applied Computing*, 2:1594–1598, 2005.

[10] Claudiu Duma, Nahid Shahmehri, and Germano Caronni. Dynamic trust metrics for peer-to-peer systems. *Proceedings - International Workshop on Database and Expert Systems Applications, DEXA*, 2006:776–781, 2005.

[11] Catholijn M. Jonker and Jan Treur. Formal analysis of models for the dynamics of trust

based on experiences. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1647:221–231, 1999.

[12] Ramzi Haddaji, Samia Bouaziz, Raouf Ouni, and Abdellatif Mtibaa. Comparison of Digital Signature Algorithm and Authentication Schemes for H.264 Compressed Video. *International Journal of Advanced Computer Science and Applications*, 7(9), 2016.

[13] V. C. Prasad. Simplification of signal flow graphs. *Circuits, Systems, and Signal Processing*, 30(3):673–682, 2011.

I herewith assure that I wrote the present thesis titled *Digital Representation for Web of Trust in Internet of Things* independently, that the thesis has not been partially or fully submitted as graded academic work and that I have used no other means than the ones indicated. I have indicated all parts of the work in which sources are used according to their wording or to their meaning.

I am aware of the fact that violations of copyright can lead to injunctive relief and claims for damages of the author as well as a penalty by the law enforcement agency.

Magdeburg, December 1, 2020

(Poorvi Mandyam Bhoolokam)