On the Use of Queuing Networks to Test the Robustness of Security Protocols:

An Analysis of the Security Vulnerabilities of IEC61850 & IEC62351

James G. Wright

Doctor of Philosophy
Royal Holloway, University of London
2019

Declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text.

(*James G. Wright*)

Abstract

This work presents a probabilistic symbolic formal method, based on queuing networks, for checking the robustness of security protocols. The method has been developed to verify the security promises of availability and synchronisation of state between devices, instead of those that are traditional analysed such as confidentiality/secrecy, integrity, authentication, (CIA) and non-repudiation. This research uses a network of $M/M/c/K$ queues to model packets travelling through the state machine of a device, or between networked devices. This methodology allows for the modelling of distributed systems, which are been computationally hard for other methods in this domain. The method relaxes the level of proof required for a symbolic formal method to calculating the likelihood that a promise is violated. The reduction in proof and complexity translates to modelling either the most likely state of the system. However, unlike other formal methods, the granularity of queuing network doesn't encapsulate message content.

This method builds upon on the work of Osorio & Bierlaire[144] by presenting an implementation with additional state space configurations and probability distributions, along with proofs of completeness and correctness for the implementation. The additional state space configurations provide a view of the total number of packets in each queue; the ordering of different types of packets in a queue; and the number of packets in each stage of the queue.

The second part of this thesis present a series of security vulnerabilities discovered within the IEC61850 substation automation standard (SAS), and its supplementary security standard IEC62351, using the queuing network methodology. These smart grid (SG) standards were chosen as a testbed for finding robustness attacks within a protocol because their primary concern is with the safe and efficient operation of the SG, which means that the focus is on quality of service (QoS) promises, that enforce hard real time limits on the data communication across the network, over security. However, some of the decisions made to ensure the QoS are met such as the omission of acknowledgement messages and requests for retransmission, may also undermine the robustness promises. The SG sector has historically been dependent on the relative obscurity of the standards to limit the attack surface of these communication networks, but the introduction of TCP/IP technologies and the increase in complexity of the standards, to allow for two-way communication between devices, has greatly undermined this approach. This increase in attack surface has been demonstrated in recent years with the stuxnet[87] and crash overide[172, 82] attacks.

Using the queuing network method against these standards allowed the author to develop domain specific attacks, instead of searching for attacks usually deployed against traditional internet technologies. The philosophical approach used in this research was to develop models of how the devices reach undesirable states, before describing what level of access and abilities the adversary required to implement the attack. This approach is agnostic to the adversary's methods of entry and techniques used execute the abilities.

During the course of this research project the queuing network was used to show that a restricted adversary can cause a desynchronisation of state between devices during the issuance of IEC61850 control commands and the desynchronisation of a device from a timing source with the correct accuracy. The method was also used to show probability of success of a maliciously injected Generic Object Oriented Substation Events (GOOSE) message to cause a denial of service attack. A context-free grammar was used to demonstrate how a race condition in the IEC61850's association model can be used in a credential intercept attack. These attacks were published across five papers ([207] to [211]).

Acknowledgements

I am filled with turbulent feelings about this thesis. Writing it was one of the most difficult periods in my life. It is a sombre end to what has been an extraordinary journey in my life. I am better person for having undertaken this research, and I am thankful for all those who have helped me complete this. The community that has come together around me are the most wonderful people, I could have ever dreamt of meeting. I look forward to starting a new journey with you all. I am eager to see what I discover next.

I would first like to thank the Royal Holloway's Information Security Group for taking a chance on me. I began this project with just a handful of radical notions about security and no track record of security work. I now leave with a firmer set radical notion, but more importantly the experience of spending an invaluable amount of time around some of the wisest security researchers I could wish to meet. I cannot thank all those that have involved in the process of getting me through this programme enough. You have provided me with an intellectual playground that has allowed me to grow and flourish into the security researcher I am today.

I would not have been able to finish this work without my supervisor Prof. Wolthusen. Whilst I wasn't the researcher he expected, with his mentorship I hope I have become the researcher he wanted. Thank you for always being available to talk, and always being ready to impart your knowledge and wisdom. You are a saint with the gallows humour most befitting of this field.

The past few years have been filled with wonderful memories. I am thankful for all my friends, old and new, who have been there for me over these four years. I wouldn't be here without you. I would like to give special thanks to Judith, Monique, Humarrah, Max, Marianne, Ela, EJ, Josh, Baines, Jules, Coral, Ceres, Shlee, Helena, Bob, Ross, Stu, Kate, Tack, Craig, Alyx, Emily, Duncan, Jess, and Lenka. To list all those that have helped me through the years would double the length of this thesis. You have made sure that my time has been filled with a constant stream of little wondrous things that spill out at me constantly.

I would also like to thank the Manchester theoretical physicists of the common room, the Egham bridge spinners, the London circus community, Castle Spinalot, the various London BDSM communities, both London 2600 chapters, the London alternative community, and all other communities that I have travelled through over the past few years for filling my life with so much colour.

Special thanks is reserved to my family.

And I would finally like to thank all those that have gone on wondrous adventures with me over the years, without whom I would have finished my PhD a lot sooner.

This thesis is dedicated to the memory of Jessica Malcolmson.
Contra Mundum.

Contents

List of Acronyms

CIA - Confidentiality, Integrity, and Authentication
SAS - Substation Automation Standard
SG - Smart Grid
QoS - Quality of Service
GOOSE - Generic Object Oriented Substation Events
DY - Dolev-Yao
ICS - Industrial Control System
IED - Intelligent Electronic Device
NTP - Network Time Protocol
PTP - Precision Time Protocol
CTL - Computational Tree Logics
LTL - Linar Temporal Logics
BAN - Burrow-Abadi-Needham
GNY - Gong-Needham-Yahalom
PCL - Protocol Composition Logic
BNF - Backu-Naur Form
CSP - Communicating Sequential Processes
CCS - Calculus of Communicating Systems
MiTM - Man in the Midle
CPA - Chosen-Plaintext Attacks
CCA - Chosen-Ciphertext Attacks
UC - Universally Composable
IDS - Intrusion Detection Systems
DDoS - Distributed Denial of Service
BAS - Blocking at Service
FIFO - First In First Out
CTMC - Continuous Time Markov Chains
PIM - Protocol Independent Multicast
RPF - Reverse Path Forwarding
RP - Rendezvous Point
SBO - Select Before Operate
GNSS - Global Navigation Satellite System
CA - Certificate Authority
MAC - Message Authentication Codes

Chapter 1

Introduction

This chapter lays out the author's motivation for developing the queuing network based formal method, and why SG communication standards were selected as a testbed. The chapter goes on to present the author's contribution, with citations to publications, before finally presenting the layout for the rest of the thesis.

1.1 Motivation

Formal methods in the security domain have traditionally focused on demonstrating the resilience of a protocol to compromise by an adversary trying to exfiltrate data from the channel. In this section the author lays out why this narrow focus of protocol verification leaves systems vulnerable to other forms adversarial interactions, and lays out the case why verification of protocol robustness should be on a equal footing with resilience. The author then explains why SG standards were selected for exploring the verification of robustness security promises.

1.1.1 Formal Methods Development

Needham & Schroeder stated that "*security protocols are prone to extremely subtle errors that are unlikely to be detected in normal operation*"[141], with this in mind a large body of research has developed various models for verifying a security protocol's resilience against an adversaries attempts to extract or alter the information that is being guarded.Several formal methods have been developed to confirm resilience promises, as well as providing precise and specific set of statements correct of operation that must be met for the protocol to withstand the classes of enacted by an extractive adversary. Using the various formal methods models have been developed to show that an adversary can't deduce the contents of encrypted messages (confidentiality/secrecy); that the adversary can't alter a messages content without the channel's agents knowing (integrity); that honest agents can believe that they know who they are talking to (availability); and that no agent can deny having received a message (non-repudiation). The protocols are verified using either the symbolic model, where a protocol session is arithmetised and proves that it is resilient to an adversary that controls the communications channel between the honest agents, or the computational model, where the adversary queries multiple sessions of the protocol to gather intelligence to deduce the content of the encrypted message. The evolution of the various formal methods, and the security promises of resilience, stem from the analysis of cryptographic protocols where the singular adversarial objective is to learn the contents of messages being passed between honest agents. In reality adversaries can engage in malicious interactions for alternative reasons than trying to extract information. The focus of this research is test protocols against adversaries that seek to disrupt the passing of information between honest agents[5]. The lack of modelling how protocols react to adversarial disruption means that there are no clear statements correct of operation against this class of attack, which leaves those operating protocols without an understanding as to how the protocol will react. This lack of set of statements of operation ultimately means that there is no guarantees secure message will be delivered through an adversarial channel, when the adversary aims to prevent

this.
The aim of the author's research was to develop a formal method that allowed for the modelling of how protocols cope with disruption attacks, which would lead to the defining of the limits of a protocols robustness. The author introduced traditional concepts from the fault identification methods of critical-systems research to the security domain so orthogonal classes of attack could be modelled to those that are traditional found. The objective of verification in the critical systems field is to demonstrate that protocols can't reach an specific undesirable state, or that it has redundant states to deal with it[28], which aren't goals that are traditionally encapsulated in security modelling. In security formal methods the adversary is only trying to gleam information from the honest agent's state machine. This meant that a new formal method had to be built to verify against this new class of attacks, along with the selection of a new set of security promises test against. For the purposes of this research the author selected the promises of availability and synchronisation of state between honest agents in the face of adversarial disruption. To be able to verify that a state machine can handle/avoid certain states, with the aim of confirming the new promises, an abstraction was developed using queuing networks, which allowed for the encapsulation and manipulation of a communication channel's properties which could then be used to show the affects honest agents' state machines.
It could be argued that existing formal method can be used to model disruption attacks, but using these methods would not provide the users of security protocols with any useful limits of correct operation when the adversary is trying to disrupt the robustness of the protocol. Traditional extractive adversaries are given complete control of the communications network between honest agents, by folding the communication channel's properties into the adversary model, which allows them to drop all the packets should they choose to. A verification done using these adversaries would provide insufficient granularity of correct operating conditions for the protocol against disruption attacks. Traditionally the focus of the research community is in increasing the abilities of the Dolev-Yao (DY) adversary[52], as well as their knowledge of the protocol, as a way of demonstrating how much effort an adversary has to go to to extract the critical information from the secured channel. However, the philosophy for modelling of adversaries attacking robustness is the opposite of those demonstrating just how hard it is to extract the information from the session is. A protocol needs to ensure its robustness against the weakest possible adversary to be able to find out what the protocol's true sensitivity to disruption is. In the queueing network methodology the weaker adversaries are created through a more granular manipulation of the properties of the communications channel. With this granular manipulation different avaliability attacks can be modelled within the formal method such as packet flooding or workflow amplification.
Being able to find the sensitivities of correct operation is especially important in distributed protocols security as they are built on the presumption of being able to pass messages between various nodes to complete their session. However, it is the inclusion of multiple honest agents which makes it difficult to verify the limits of the protocol using formal methods as the inclusion drastically increases the state space that needs to be searched to find potential compromises. Another challenge is the encapsulation of the time dependency that the functionalities of these protocols have within a formal method. This inability to verify the security of distributed systems leaves a large amount of contemporary computing infrastructure operating on untested assumptions that may collapse when adversaries interact with them[41]. The author presents their queuing network based formal method as step in the direction of solving these problems.

1.1.2 Security of Smart Grid systems

An example of a technological domain that relies heavily on distributed communications and has stringent timing requirements for its messages are the protocols that are used to control power systems. These requirements need to be met to ensure that an operation or event isn't missed that leads to physical equipment being damaged that could either begin a cascade of events across the network or injure someone near the equipment. Traditionally the monitoring of and control of power systems communications was done using a hierarchical SCADA set up, which meant that the communications could be air gapped from the rest of the internet. This air gap also relied upon the relative obscurity of how protocols operated. However, a greater level of communication and autonomy of devices are being introduced into power systems due to the challenges of incorporating greener power generation technologies into power networks. This has come in the form of the creation of the SG design paradigm which uses various internet technologies to increase the level of connectivity between devices. This new paradigm, and its associated technologies, have degraded the traditional air gap security principle that the power systems has relied upon for the past few decades as they communications network is now reliant on more commonly know protocols as well as being more enmeshed with the internet. One of the principle design goals of any power system communication network is to ensure the safe operation of the physical equipment. Due to the sectors faith in the air gap relatively little effort has been put into the security of the communication protocols, despite the fact that the second order consequence of the security being compromised could be the disruption of the safe operation of the physical equipment. Efforts have occurred in both academia and industry to secure SG systems against the traditional attack vectors of web service, the research into domain specific attacks has been limited. The safe of the physical equipment on the transmission network manifests within the SG communications protocols as strict QoS requirements that mandate how devices should behave and how quickly messages will be delivered. However, these QoS restricts the deployment of traditional security solutions on the communication infrastructure due to the pursuit of the hard real time requirements leaves little room within the IP technologies for the communications network incorporate the security technologies and processes. This choice between safety and security wouldn't exist if purpose built communications solutions were developed.

With the standard's focus pointed squarely at safety bespoke attacks against SG networks have already begun to emerge in the wild. Attacks against industrial control systems (ICS) had been incredibly rare[155] until the Stuxnet virus[87] which disabled and damaged Iranian Uranium enrichment programme. Since then attacks against ICS have grown more sophisticated. Two notable examples in the electrical grid domain are the attacks against Ukraine's electrical distribution network. In 2015 a foreign adversaries infiltrated their IT infrastructure. From this foothold they were able to gain purchase in the SCADA human interface servers which allowed them to switch on circuit breakers and install malicous firmware that hindered the recovery effort[205]. In December 2016 the Ukrainian power grid was affected by the malware 'CRASHOVERRIDE'[82], which was designed to spread across the communication networks that had implemented the IEC60870 & IEC61850 standards. It traveresed the network and purposefully threw circuit breakers and damaged equipment. The delivered payloads to shut off several transmission substations across Ukraine. The interest in this domain of attack surfaces hasn't gone away since these attacks[161]

The IEC61850 protocol was selected for this research because it declared a specific set of QoS parameters and promises that devices communicating using this standard must obey to ensure the interoperability regardless of vendor. This is in contrast to other standards used the SG domain, such as IEC60870 and DNP3, who lack these declarations as they are deployed in other kinds of cyber-physical systems. The QoS promises of the standard are synchronisation of state, access control, and availability of devices, which are the promises that the author wanted to explore using their methodology. It also sets out various hard limits for the arrival times of the different types of messages. These promises and parameters are declared to ensure the safety of the physical devices connected to the communication network. In spite of this, all the communications models within the standard have been designed with the presumption that protocol sessions will always run as intended, despite including synchronisation of state promise to ensure sessions can quickly be resumed. Whether the incorrect operation occurs due to an error or malicious intent the fact that none of the communication models have any means of handling exceptions will lead to unsafe states occurring which could lead to damage or injury. If a malicious adversary were intent on destroying transmission equipment, they could achieve this by attacking an intelligent electronic devices (IED) by either making it lose the state of its session or denying its ability to communicate with other devices. The would lead to a violation to the QoS limits which lead to physical devices being destroyed from overcurrents. Another concern of author was that if IEC61850 didn't have declared ways of handling undesired that there would be no consensus amongst device vendors on how to handle them. This would lead to violations of interoperability of devices being undermined occurring. The standards communication models hadn't been tested for robustness against error and adversarial interactions which meant the communication network could cause dangerous situations to arise. With these

problems in the standard the author felt that they would be able to find omissions in the standard's QoS using the queuing network methodology they were developing for the reasons given in section 1.1.1.
The author also used their methodology to assess IEC62351 ancillary security standard which offers authentication, integrity protection and confidentiality, as well as mechanisms for replay detection to devices using IEC61850.

1.2 Research Questions

In summary the research questions that this project sought to answer were:

1. Is there way of creating models within a formal method to define the limits of security protocol's robustness against adversaries seeking to disrupt a session between honest agents? This could be from an adversary with the objective to either ensure that protocol session isn't completed (availability), or that a session has to reset if the connection is lost (synchronisation of state between agents).
2. To be able to precisely define the limits of correct operation of security protocol in the face of robustness attacks, how can the DY adversary be made weaker?
3. The IEC61850 SAS declares that its QoS promises are synchronisation of state (session resumption) and availability, to ensure that devices from different can interoperate and upkeep the power transmission infrastructure's safety. Using the methodology developed to answer questions 1 & 2 are their omission in the standard's descriptions of the different sessions where an adversary or error can undermine these promises?

1.3 Contributions

Questions 1 & 2 are concerned with the creation of models to see if a security protocol is vulnerable to promises that aren't traditionally tested for with formal methods. For the reasons given in section in section 1.1.1 the author decided to create a new formal to answer these questions. The author's method is based on symbolic model (which is described in section 2.1.1) methods, but provides a lower standard of proof than those in this family of methodologies. Instead of providing a rigorous proof that a security violation is within the standard, the method gives the probability that a violation of availability or synchronisation of state will occur given a set of operating parameters. The author's probabilistic formal method was developed using a network of $M/M/c/K$ queues, that use the block at service congestion discipline. The is based on the work of Osorio & Bierlaire[144] but was expanded upon by the author by introducing:

- Other state space configurations, that provide the ability to analyse a queuing network in different ways.
- Alternative probability distributions of packet acceptance/processing to M/M, that allow for different traffic models through the network.
- The inclusion of being able to model several packet types at once.

The queuing network is set up to to either represent the state machine of a device running the protocol or a network of devices communicating with a specific protocol. The adversary interacts with the network by either introducing new kinds of packets in the network or altering the arrival or processing rates of packets into a queue. These rates can be used to represent features of the scenario being modelled, such as the size of the packet. The effects of these interactions can be seen through the different queuing network state space configurations. The configurations available in the authors method are:

- The number of packets (agnostic of packet type) in every queue in the network.
- The number of packets of each type that are currently being processed, blocked from transmitting, or waiting to be processed, either in an individual queue, as developed by Osorio & Bierlaire[144], or every queue in the network.

- The order of the packets of each type in either an individual queue or every queue in the network.

The results of these state space calculations can be further developed with the use of traditional queuing theory performance metrics. The use of conditional probability allows for the modelling of weaker adversaries that can only infer information about the state of an honest agent or whether packets may have passed along a path in the network. The author's implementation of the formal method is presented with a set off proofs of completeness and correctness for the data structures and algorithms they have used in the design of method.
All these features greatly expand upon the current research into using queuing theory to model attacks against a systems availability.
Using the queuing network formal method the author was able to find several omissions in the IEC61850 communications models that were developed into attacks that compromised the standards QoS/security promises of availability and synchronisation of state. The attacks discovered by the author were:

- A race condition was demonstrated in the standard's association model which allowed for an adversary to be able to intercept legitimate credentials intercept attack. This attack undermined the standard's promise of access control. This attack was presented using context free grammars[209].
- The standard's multicast messaging services was demonstrated to be utilisable by an adversary to create the conditions for a workflow amplification attack. This attack compromised the standard's promise of availability[209].
- It was shown that an adversary can de-synchronise a server issuing control commands to a node during a session, which would force the session to reset. This attack compromised the standard's promise of synchronisation of state[210].
- The likelihood of successful packet injection attack utilising a malicious GOOSE multicast message was calculated. This was shown that it had the potential of creating a DoS attack due to IEC62351's anti replay attack algorithm[211].
- Showed that a IEC61850 node can easily be de-synchronized from the layer of time-synchronisation network that has the correct accuracy for its functionality, due to the state machine having no states for handling exceptions [207].
- A queuing network model for general packet injections was presented.

In the appendix of this work the author presents a qualitative analysis of the omissions in IEC62351's threat modelling.

1.4 Thesis structure

The rest of this thesis is structured as follows. Chapter 2 goes over the current state of the art in security protocol analysis and smart grid security. The first half of the chapter presents the various formal methods that are used to analyse the limits of the correct operation of security protocols. How each method works will be presented, along with what aspects of the protocol they encapsulate, and the level of human intervention that the method requires. This section will go over the deficiencies of the symbolic model approach, as the author's formal method is primarily based from this approach. It also presents how the research community have increased capabilities of the DY adversary through additions to their abilities and knowledge capacity. It also goes over the computational model used to prove the correctness of cryptographic protocols. A section is dedicated to the security research that has been done using queuing theory and context free grammars.
The second part of the chapter goes over state of research into the security of SG technology, along with the unanswered problems. It first lays out what security considerations are made by the IEC61850 and IEC62351 standards. There are sections that discuss the privacy issues and the attempts to apply the encryption protocols to SG standards. An entire section is dedicated

to the SG specific attack vector of injecting false data into the state estimation of the grid. It also looks at the security problems of the network time protocol (NTP) and precision time protocol (PTP) protocols (these are the standards referenced in IEC61850) as time synchronisation is a critical aspect of substation automation.

Chapter 3 presents the author's probabilistic formal method to find the limits of a security protocol in the face of disruption attacks. The first section goes over the standard results of queuing theory so the mathematics of the $M/M/c/K$ queuing network can be described. It then presents how the author's implementation of the formal method perform its calculation. The next section goes over the proofs correctness and completeness of the author's implementation of the formal method. Before finally providing a comparison of the queuing network formal method with the other methods presented in chapter 2.

Chapter 4 presents the attacks against IEC61850 that the author discovered during this research project. The attacks in this chapter are those listed in section 1.3

The final chapter provides a summary of the work presented in the thesis, along with future work that could be added to the formal method and possible research question that could be perused using the method.

The first appendix presents a paper that gives a qualitative description of the limitations of IEC62351-3 public key management[208]. The second appendix is an unpublished manuscript that gives a broader qualitative critique of the threat modelling done in IEC62351.

Chapter 2

Literature Review

This chapter explores the current state of the art in security protocol verification and research into the security of SG systems. In the first section it is shown that the overarching focus of the security protocol verification community has been to show that security protocols are resilient against adversaries seeking to extracting a secret message, with very little attention being paid to other classes of attack. The review explores the two different schools of thought for verifying the promises of security protocols, the symbolic model and the computational model, as well as reviewing the various projects working on automating the process. The various approaches of each school are presented, a long with the general limitations of the schools of though. This review is demonstrates that there is a niche in the protocol verification community for methods that test for other attack vectors, which the author's formal method begins to fill, and lays the groundwork for the comparison of methods presented in section 3.3.

The review goes onto to looking at how queuing theory has been used to model DoS attacks in security research. In this section it is laid out why queuing theory was selected as the formalism for the author's formal method. It also presents a brief look of how finite state machines have been used in security research.

Section 2.2 reviews the research that has been done on the security of SG protocols and systems. It presents the security considerations that are inherent within the different SG protocols, and then looks at what security promises have been used to analyse the strength of these mechanisms. The section goes on to look at possible solutions that have been researched, such as intrusion detection systems, before looking at the specific security risk that arises due to the cyber physical nature of the SG, attacks against the grid's state estimation mathematics.

2.1 Security Protocol Modelling

This section gives an overview of the two philosophical approaches that are used to test the validity of security promises of protocols, before going over the research into automating these processes. The frameworks of protocol analysis are the symbolic model, where the protocol is arithmetized to see if the adversary can extract the secret message from flaws in the protocol's narration, and the computational model, where the adversary tries to compute the secrete message by inferring from quires made to multiple sessions of the protocol.

2.1.1 Symbolic Model for Analysing Security Protocols

The symbolic model is primarily used to analyse the semantic steps of security protocols run to see if there are flaws in the logic that an adversary can exploit. All symbolic methodologies encapsulate aspects of the functional operations required for the completion of protocol run, such as the messages, keys, nonces, and so on, as variables without, whilst excluding other message qualities such as size or content, which then are used to demonstrate how an adversary's interactions with the protocol will create a counterexample to the promises. In these approaches it is usually presumed that the adversary has no crypto-analytical capabilities so they can't infer a messages content. They can only learn the message if they learn its encryption key. This section starts with the a general critique of symbolic model, and the present the different adversary that have been developed by the research community. The first symbolic model, and the basis for all symbolic adversaries, was done by DY who demonstrated that cascade protocols using public key cryptography weren't secure against an active adversary[52]. They did this by developing a simple operator notation of public key cryptography. This methodology encapsulated only what the agents knew from the values of the session that they posed. After exploring the research into symbolic adversaries an overview of the different symbolic model based formal methods is presented.

Undecidability & Complexity

All of the symbolic methods discussed in this section are bounded approximations to reduce the size of the potential state space of variations of protocol that need to be searched for counterexamples. If the methods didn't enforce some form of bounding, finding counterexamples of promises would become an undecidable problem. The initial research into this domain was done whilst trying to define the complexity of proving confidentiality. The first result in this field was completed by Even & Goldreich, who demonstrated that verifying the confidentiality of a ping-pong protocol was NP-Hard problem[59]. Tiplea *et al.* provided an overview of proofs on the decidability of confidentiality problems. In their work they reduced the post correspondence problem to a confidentiality model which they used to show that if either the nonce and/or the message length in a security protocol model is unbounded then proving confidentiality is undecidable. They also provided an upper bound on the complexity of proving confidentiality promises within different boundary conditions. If the model has an unbounded number of sessions then it can be proved in DEXPTIME, and if it is bounded it is NP-HARD[190].
The author did not find any results regarding the decidability and complexity of other security promises.

Symbolic Adversary Strength

The standard adversary used in by the symbolic model research community is the DY model[52]. The model states that the adversary controls the network between the communicating parties, and so can intercept, edit, delay, or drop any message that is passed between parties, but cannot decipher a message unless the discern the key to the message.
There appears to have been little research into altering the strength of the adversary in the academic literature. The cases the authors have found are, Cohn-Gordon *et al.*[43] who model what security promises can still be maintained in authenticated key exchange protocol when the adversary has compromised the communications channel's keys. They define two types of compromise, weak, where the adversary has the use of the keys but doesn't discerns them, and total, where the adversary discerns the key. They prove that a secure session can still be possible in both situations so long as conditions are met. For weak the adversary needs their access revoked before the secure session begins, and for total there needs to have been at least one uncompromised message exchange before the secure session.
Cremers *et al.*[47] build on this work by exploring other forms of compromise. They add the adversary capabilities of having learned the pre-shared keys or Diffie-Hellman values of session. With these new forms of adversary strength they use the Tamarin modelling checking software to prove that draft 21 of the TLS 1.3 is secure against these forms of compromise, as well as protection against downgrade attacks. However, they did find a potential de-synchronisation of state attack vector that could undermine the clients belief of authentication in the draft.
McEvoy & Wolthusen[125] used π-calculus to model an adversary with various agent devices deployed across a limited part of the network between communicating agents to limit the adversaries strength. They also encapsulate the latency between the adversary's agents, as well as modelling that information might be lost between them. They use their model to demonstrate that if the adversary is restrained enough that they will reveal themselves to the communication network before they perform their malicious actions.

CTL* logics

The CTL* logic is the superset consisting of computational tree logics (CTL) and linear temporal logics (LTL) which are used to formally verify the safety promises of protocols. They are a deductivive verification method that uses Kripke structures to test the rules concurrent finite state systems session against a specified set of promises. Examples of promises that can be checked for are deadlocks, no progress in the run is possible, safety, the set of undesirable states is never reached, fairness, and a state does not have to wait infinitely long before being called upon. The methodology views the state machine of a protocol's communication model as a tree of infinitely long paths. Each branch on the tree is a potential trace that a session could go down. The logics encapsulate rules of the state transitions during a protocol session, and then are used to model a potential trace through the state machine. This approach means that a explicit notion of time is not incorporated into the modelling of a system. The logical formulae are used by CTL*'s to describe the properties of a path. The formulae are made up of path quantifiers and temporal operators. The path quantifiers describe the set of paths the logical state will apply to, and the temporal operators, 'next', 'eventually', 'always', 'until', 'release', and 'never', describe the properties path.
Due to the computational complexity required to describe the computational tree, CTLs and LTLs are bounded forms of CTL* which prevent the problem space that needs to be searched from exploding by only proving certain aspects of the tree. LTLs describe events only along a single path of the tree by placing the restriction on CTL* that path formulae may only be preceded by atomic propositions. The time complexity of LTLs is $O(|T|exp^{|\phi|})$, where $|T|$ is the number of traces and $|\phi|$ is the number of path formulae. CTLs still make statements about the whole tree of state transitions but restrict CTL* by demanding that temporal operator must be immediately preceded by path quantifier. The time complexity of the CTL algorithm is $O((|N| + |K|)|\Phi|)$ where $|N|$ is the number of states, $|K|$ is the number of transitions, & $|\Phi|$ is the number of formulae to be satisfied[14].
Research has been done into modifying LTL to model security promises of a protocol. Corin & Saptawijaya[45] modified an LTL with term algebra, which allowed for the encapsulation of agents in a protocol session as well as describing an adversary's abilities. A compiled term algebra is used to describe a session from which traces analysed using an LTL. A security property is confirmed if a logical statement about the adversary's knowledge is disproved by the logic. This method allows them to test the security of protocols compromises of confidentiality, authentication, and forward secrecy as well as DoS attacks that utilise multiple sessions of a communication model to chock a state machine. This is the only symbolic model based formal method that the author found that looks for attacks against availability in a protocol's state machine. The other security LTL in the academic literature was developed by Carbone *et al.*[11] which modifies an LTL with multiset rewriting. It is the same methodology as Corin's, but instead of traces, a labelled transition system of the session diagram is generated and reasoned about. This approach allows the agents to have different strengths of confidentiality by altering the security promise of a communications channel between agents. It can also model agents that time out eventually. This is the only symbolic model that the author found where the properties of the communication channel were decoupled from the adversary model.
Whilst there are various model checking software packages that use these logics, such as LTSmin, SPIN, NuSMV and TAPAs, the author is unaware of any that have been specifically developed for the analysis of security protocols. In these model checkers the system analysed is defined in the tool's modelling language. These are used to define elements of the system such as variables, and data types, as well as the promises that the system is expected to comply with.

Belief Logics

Belief logics are used to verify the belief held by an agent on the status of a sessions security promises matches reality. They are usually used to prove that an agent's belief in the identity of the agent they are corresponding with (authentication) is correct. The processes that the logics use to prove the security promise of a session is by generating a logic description of a protocol run and then annotating the process by hand to encapsulate the actors beliefs about each step.

The first logic of this kind was the Burrow-Abadi-Needham (BAN) logic[32]. They developed a set of logical connectives to describe the steps in a protocol run and the beliefs of the actors involved. The logic works by deriving an idealised form of the protocol, by removing all information that doesn't contribute to the security beliefs of an actor, and then stating the assumptions that each actor will have of each step. Logical formulae are derived for each step of the protocol, from the assumptions, and then checked against the BAN logic's postulates of authentication. If there is a contradiction between the formulae and postulates then the authentication promise of the run has been undermined by the adversary. The logic was found to have several flaws that undermined the verifications that had been made with it, such as the idealisation process stripping out crucial details, and the lack of encapsulation of compromised actors undermining the authentication proof[122].

Gong-Needham-Yahalom (GNY) logic[71] sought to fix and extend BAN logic, by removing the dependence on universal assumptions onf the protocol run. The logic also encapsulates what an actor expects in the message which allows for the separation of the implied and actual content of messages. These extensions were achieved by increasing the range of logical formulae and postulates of the logic, as well as removing the idealised protocol step.

Datta *et al.*[50] extended the concepts of belief logics with their protocol composition logic (PCL). They created a logic that can assess the security of of protocol runs made up of a composition of different security standards that have concurrent processes. It is achieved by first describing the communication model with a bespoke process calculus that describes the threads of the protocol run. Each thread can have several agents acting on it, and the agents pass information between each other via buffer cords. Once the system has been modelled, logical statements are made about the threads using various sets logical formulae created by Datta *et al.*. The formulae make statements on the knowledge and honesty of actors, and the temporal ordering and nature of actions. From these formulae proofs can be generated for the security promises of the system. The security promises that can be encapsulated by the logic are confidentiality and authentication.

Operational Semantics

Cremer's operational semantics were developed to separate the logical description of a protocol from it's dynamic behaviour. It uses Backu-Naur form (BNF) grammars to describe the protocol operations, along the series and order of actions an agent using the protocol can make. Each agent in the run is given a role specification which describes their initial knowledge and their order of operations. The adversary is different as their specification is only their initial knowledge of the run. The adversary's initial can include the knowledge of agents they have compromised. The grammars are used in labelled transition system to describe a potential trace of a protocol run. The transition system allows for any agent to repeat an event to encapsulate multiple protocol sessions happening between agents at the same time, but it is assumed that the runs are independent of each other. The state of the run is described by the adversary's current knowledge and the remaining events of the current runs.

The semantics prove the security promises of a protocol run by having an agent test their perception of the global state of the system from their local knowledge. The promises that can be tested by an agent are, confidentiality, which is done by checking if the message is discernable with the adversary's current knowledge; granular forms authentication that allow the agent to check if the agents they are communicating are deriving the correct information in the session, and are alive in the session; and several forms of synchronisation, such as message agreement, the protocol executes as if there is no adversary (non-injective), and that runs are resistant against replay attacks (injective)[48].

Cremer created the Scyther tool to check protocols using this approach, which he has been proved to prove the security promises of IKEv1 and IKEv2 protocol suites and the IEC 9798 authentication protocol. The tool implements the algorithm layed out in Creamer's work[48], but the user has to set up the grammar and transition system by hand before the tool can search the problem space. This approach was further incorperated into Cremer's more recent tool Tamarin, which is discussed in section 2.1.3.

Process Calculus

Process calculi are a family of approaches used to model the evolutionary behaviour of concurrent systems after certain events have occurred. The two main approaches were independently developed in the 1970s by Hoare, who developed communicating sequential processes (CSP)[79], and Milner, who developed the calculus of communicating systems (CCS)[128]. These two different approaches have different philosophies of modelling, but both approaches use a set of actions that describe how the behaviours of processes evolve, along with including an equivalence relation to compare different constructions[142]. Whilst the two approaches have converged over time they both began trying to solve distinct problems, and they're still some differences[13, 12]. The types of actions that are common across all the calculi are those related to decisions, interactions between concurrent processes, and behaviours over infinite time scale. The aim of theses constructions is to see if a system is equivalent to an idealised construction of the system's function, or meets some promise that has been described in a temporal logic. The usual standard of equivalence that is bisimulation, which compares if the models have the same structure and accept the same input as the idealised form. CCS builds upon systems described by a labelled transition systems, and focuses on identifying behavioural equivalences between constructed models. CSP is a system for modelling the language used to govern concurrent systems. Their primary use in security research is to see if an adversary is able to gain access to a communication channel, and if so what they can achieve in the channel given the abilities they posses.

The CSP calculus has been developed to be able to verifying the traditional security promises of a cryptographic protocol, confidentiality, authentication and non-repudiation[160]. The most seminal use CSP to verify a security protocol was done by Lowe, who used a CSP based model checker to show that the Needham-Schroeder protocol was vulnerable to a man in the middle attack(MiTM) as well as undermining the promise of authentication[111]. Schneider has used the CSP calculus to build upon traditional security promises. He used CSP to develop a method of proving that a protocol's message may appear to an agent that it could have come from any participant in a protocol run with equal likelihood[167], and then used this result to verify that the agents in the dining cryptographer problem are anonymous[38]. Schneider & Evans introduce the concept of event based time in CSP to verify the authentication promises of the wide mouth frog protocol[58]. Most of Schneider's work was completed using Lowe's Casper tool. Casper requires the user to lay out a narration of the protocol they want analysed as well as the primitives used in the protocol, the agents and their initial knowledge. The solver presumes that the intruder is a Dolev-Yao intruder[112].

Most of the security research using process calculi is done with the CCS derivate called π-calculus. π-calculus is an extension of CCS developed by Milner*et al.*[129, 130] to allow for the encapsulation of mobility in concurrent processes, which allows for communicating automata to proliferate, forge new links, and die in the models of concurrent systems. As π-calculus can build models with restricted channels between processes it is straight forward to test the confidentiality promises of protocols, whilst other promises are more difficult using this method. There have been two main approaches for being able to test other promises. The first was to create new actions for the calculus, as done in the Spi-calculus which added encryption and decryption actions so authentication of protocols could be tested[2]. Spi-calculus has been used to encapsulate the knowledge and expectation of agents, as well as packet fragmentation to reduce the ambiguity of protocol narration descriptions[31]. The other approach is applied π-calculus which allows functions to be passed between processes, this means that a wider range of cryptographic functions can be modelled without having to develop a more complex calculus syntax[1]. To the authour's knowledge there has been no automated implementations of either spi calculus, so any verification of protocols using this approaches will have to be undertaken by hand. The applied π-calculus has been used in various security domains such as verifying election properties, interoperability of web services, and key agreement protocols[159]. Cheval & Cortie further extended applied π-calculus by incorporating time functions into the cryptographic functions. This allowed them to model side channel attacks in security protocol sessions[40]. This approach was incorporated into Blanchet's proverif tool which is described in section 2.1.3.

Strand Space

Herzog *et al.* strand space methodology was developed to ensure cryptographic protocols upheld the security promises of confidentiality and authentication in the presence of an adversary. The strand space models a session diagram as a pair of strands between agents whose only ability are to send or receive messages with certain cryptographic attributes. Each agent's strand represents their local view of the session. A security promise is proven to be upheld when the agents' belief of what happened in the session match, and there is no construction of adversarial strands that give them access to the channel and maintain the agents' beliefs. If a bundle of strands can be constructed to show that the adversary can perform a series of actions that produce an output that simulates an agents belief, it proves that the promise can be undermined. In a Strand Space model the adversary is an expanded DY adversary, who can concatenate messages or separate messages into components[60]. Strand spaces were extended by Syverson to formally encapsulate the belief propositions that can be made using BAN logics[185].
There has been a limited number of protocols verified using this strand spaces. Kamil & Lowe used strand spaces to build a model to verify the security promises of TLS 1.0[85]. As far as the author is aware there has been no automation of this approach.

Induction

Paulson's structural induction method allows for the analysis and confirmation of security promises of an unbounded security protocol, by considering a set of potentially infinitely large traces of protocol runs being enacted between unbounded set of actors. The set of actors include the adversary, who can also discern knowledge from the actors they have compromised. The approach also encapsulates multiple sessions occurring between agents at the same time.
Three sets of types are used to describe the protocol's communication model, agents, messages, and events. The agents list the actors in the simulation and messages describe the cryptographic functions that an agent can perform to, whilst events keeps track of all the traces going across the network. Before each simulation the actors initial knowledge is defined. Each agent will have a list of keys they have access to, and the adversary will only know the keys they have from compromised agents. As the protocol run progresses the adversary gains access to the messages passed across the network, as the approach presumes that the adversary is a DY adversary. The approach uses three operators that allow the agents/adversary to interact with the protocol run's traces, partz, which learn messages from the traces, analz, which extracts and decrypts mesages from the traces, and synth, which allows for the generation of fraudulent messages. The security promises of the run are analysed from a global perspective, where if a certain event takes place in the traces the security promise is declared undermined. For example if the adversary learns an agents key and so is able to perform on the analz operator on a message that is imbued with the promise of confidentiality then the session is compromised. The other promises that the induction method is able to verify are integrity, correct key distribution, authentication, unicity, and that the session isn't reused.
Paulson implemented the induction formalism in his Isabella theorem solver[149], and used it to prove the security of the Kerboros protocol and the security of some smart grid functions.

2.1.2 The Computational Model approach for Verifying Security

The next two approaches for security protocol verification are from the computational model school of thought. In this model the adversary is trying to increase it likelihood of successfully decrypting a message through various kinds of cryptanalytical attacks. In the computational model the messages passed between agents as bitstrings, with the cryptographic primitives being functions of the bitstring. In this case the adversary is modelled as a polynomial time Turing machine, with the objective of discerning the decryption key through cryptanalysis. The author is unaware of any specific automated of generating proofs in this domain.

The Random Oracle and Cryptographic Game Proofs

The concept of probabilistic encryption was first put forward by Goldwasser & Micali[69] as way preventing an adversary from discerning the either the whole or part of a cleartext message that had been encrypted using a deterministic trapdoor function. They describe how an adversary can attack the message if the possible message space is sparsely populated. Their solution was the development of the unapproxable trapdoor function, that uses implied intractability, of the quadratic residuosity problem to encrypt the individual bits of the message. To prove their new protocol was secure they described the notion of semantic security, where if it is shown that a passive adversary is bounded to having access to only polynomial time Turing machine cannot discern which of two messages produced by a given cyphertext with better than probability of 0.5 then it is secure. From this work they introduced the notion of the random oracle[70] to further increase their notion of security. A random oracle is a black box function that if queried with a certain input produces a seemingly random output. If a cryptographic function can proven to appear to act in this way it is deemed secure from adversaries trying to discern a cleartext message by repeatedly querying the protocol's oracle.

These notions of security were developed into a method for testing and developing cryptographic protocols. Bellare & Rogaway[21] showed that an implementation of a cryptographic protocol could be secure if the protocol had proven to act like a random oracle. Which meant that a random oracle could be replaced with a pseudorandom function, as true random oracles cannot be physically implemented, and the new protocol would still be able to maintain its security. This was disputed by Goldreich *et al.*[36] who proved that it is possible for a random oracle secure protocol to not have any secure implementations as ensemble used as the source of randomness for the protocol cannot guarantee that there is no tractable way of correlating the encryptions input and output.

Whilst cryptographic games have been used since at least world war 2[88], their contemporary formulations are based on Goldwasser & Micali's semantic security. The idea is to demonstrate that given an adversarial interaction with a protocol across multiple games that the adversary can't discern a defined bit of knowledge about the protocol run. The promises is usually proven by the adversary trying to discernin which of two messages was encrypted by the protocol. The comparison between games is either done comparing the execution of the protocol across games[20] or the probability of an event in a shared probability space across all the games[169]. Whilst the are is an array of games that an adversary can deploy against a protocol , the two principle ones that are tested against are chosen-plaintext attacks (CPA) and chosen-ciphertext attacks (CCA). CPA is when the adversary is trying to discern between two messages, and is given access to protocol a run's encryption function, whereas CCA extends this to allow the adversary access to both the encryption and decryption function[19].

Universally Composable (UC) Security

The UC method for analysing security protocols, developed by Canetti,[34], demonstrates that an observer of a protocol run can't tell the difference between an idealised form of the protocols objective and the protocol itself. The UC method views a protocol as program that is executed across several interacting machines, which allows for the modelling of concurrent protocols, with an adversarial machines. There is also a machine that governs all the machines in the protocol run called the environmental machine. This machine takes and distributes the protocols initial inputs, and performs the final operation of the run of concatenating all the machines local inputs. All the machine are defined by their ID, the set of machines they can communicate with, and the program that they are operating. The communication set is series of pairs that describe if the other machine can provide input, receive output, and, if the machine is the adversarial machine, or a compromised machine. The first model of the protocol that is used in UC verification is the idealised form. The idealised form is used to define the security promises that are being verified, as the adversary can only compromise environmental variables. The UC method has been used to prove the promises of authentication, confidentiality, and synchronisation of machines, as well resilience against byzantine generals[99], side channel, and coercion attacks. The protocol execution model allows the adversary to compromise any machine in the protocol. If the final concatenated global outputs of the two different forms are the same, then the protocol is has

met that security requirement.
A useful feature of the this methodology is that it reduces the amount of effort required to prove a complex protocol is secure. It has been proven that a protocol that is made up of a set of simpler interconnected protocols that have been proven to be UC-secure, then the new protocol is also secure[35].

2.1.3 Security Analyizer Tools and Proof Solvers

There have been many different automated tools that have been used to analyse the security promises of crytpographic protocols[48, 150, 147], this section will go over the contemporary tools that are currently in development.
The Tamarin solver is being developed by Cremers *et al.*. It is based on combination of multiset rewriting systems[37], first order logical formulae, and labelled transition systems. The labelled transition systems keeps track of what the adversary knows, what messages have been passed on the network, the freshness of values in the network, and the protocol's state. The multiset rewriting systems are used to define the agents and protocol, as well as keep track of the facts and transitions of the labelled transition system. The first thing that needs to be defined is the equational theory of the model. This includes cryptographic operators, as well as mathematical operations that an agent can perform. Multisets are used to define the behaviours of the agents, along with notions of value freshness and how the adversary can interact with the labelled transition system. Finally the security properties that are being verified are laid out in a first order logic. The Tamarin solver looks at the traces of facts as the labelled transition system evolves. The solver looks for counter examples of the promises, which in turn could be potential attacks in a protocol run[166].
Tamarin has be used to discover attacks 5G Authentication[18], the TLS 1.3 drafts[47], and V2X Revocation Protocols[204].
Proverif is an applied π-calculus cryptographic protocol verifier implemented in prolog by Blanchet[24]. The cryptographic operations used by the agents, and the adversary's abilities are described as constructor functions that describe what functions are passed along a communications channel. The adversarial model primarily used by proverif is the DY model. The system can model multi-session attacks as it allow for an operation to repeated limitless times so long as the previous operation has been completed at least once. Given the potential for these process to not terminate Blanchet developed an algorithm that reduces the π-calculus into horn clauses. These horn clause statements are then searched to see if an inference rule that state the adversary's objective is true[23].
Proverif has been used to verify the security of smart grid authentication schemes[165].

2.1.4 Queuing Theory in Security research

The principle focus of queuing theory security research has been describing network level DoS attacks. It is a suitable abstraction for this type of attack as the queuing theory abstraction can easily be mapped onto the idea of a device with a buffer of packets to process. The queue represents the server and the objects entering the queue are equivalent to the packets entering the server's buffer, from this the efficiency that a server can process packets can easily be calculated. Different probability distributions for the packet arrival and processing can be used to model how a server can cope with different scenarios it may be faced with[76]. Despite this natural affinity, seemingly little research has been pursued modelling other attacks against availability of a server in queuing theory.
Most DoS attack research using queuing theory uses either a single $M/M/1$ queue, which will be explained in section 3.1.1, or an open Jackson network[72], which is a network of $M/M/1$ queues. Relying on $M/M/1$ puts a boundary on what can be learned from the model, as this type queue has no limit on its buffer. Given this underlying assumption, all that can be discerned from these models is the degradation of a servers performance. It can't answer the question of when a server will cease to meet the promise of availability.
Below is a selection of the more novel queuing theory DoS models.
Wang *et al.*[200] built a two dimensional embedded Markov $M/M/1$ queue to model the number half open connections a server has caused by both regular and adversarial traffic during

a SYN flood attack. Each packet has a different probability distribution for their arrival rate into the server queue. For their work they develop an algorithm that has a slightly smaller time complexity, $O(N^{5.7})$ where N is the number of half open connections, than Gaussian elimination algorithm usually used to solve continuous time Markov chains to find the stationary probabilities of their queuing model. With their model they are able calculate the probability of connection loss and buffer occupancy percentage. With their metrics they demonstrate that reducing the holding connection time or increasing the maximum number of half open connections reduces the probability connection loss, but only reducing the arrival rate of malicious connections reduces the percentage of buffer occupancy.

Akhlaghi *et al.*[4] use the exact same mathematics, experiments, and metrics as Wang to obtain the same results. The difference between the two papers is Akhlaghi *et al.* is looking at DoS attacks voice over IP proxy servers.

Wan *et al.* [198] developed a countermeasure to a SIP flooding attack using two $M/M/1/\frac{K}{2}$ queues, where K is the size of the servers buffer. They split the traffic entering the server into two priorities. The low priority is for packet inviting the server for a connection, and the high priority is for all other packets. This mathematical solution to this model was generated by Ohta [143]. This model prevents timeouts occurring in legitimate server interactions, as all new requests are processed in a separate queue. Using this method they show that the occupancy of high priority queue remains relative stable, and thus keeping the processing time of the legitimate packet under the timeout threshold, whilst the low priority queue bears the brunt of the DoS attack. However, their model doesn't take into account that initialisation of legitimate sessions may still be drowned out in a DoS attack.

Kesdogan [92] uses queuing theory to analyse the MIX anonymity method, and justify changes to the algorithm to prevent attacks against the protocol. The MIX protocol has servers collect a batch of packets before outputting the batch in a different order than it received them. This process can occur several times to reduce the likelihood of successful statistical analysis, but the process only needs to be successful once to provide some level of anonymity. One attack against the protocol is if the adversary manages to get all but one their packets in batch to be theirs. Kesdogan uses an $M/M/1$ to show that if the MIX server chooses an arbitrarily selected time window for which it collects the batches of packets, it greatly reduces the adversary success.

Kammas *et al.* [86] used an open Jackson network to model the propagation of computer viruses across a computer network. They did this to find a computationally easier abstraction as an alternative to non-linear epidemiological models. Each node in the network represents a computer on the network, and so the probability of transmission between nodes encapsulates the topology of the communication network. The state space of the network represents the number of viruses, anti-virus programs, occupying each node and the number of virus/anti-virus annihilations that have occurred at a node. The use of open Jackson networks reduces the complexity of the calculations required, as the open product form of the state space can be used to generate a solution[72].

Xu *et al.*[212] use the birth-death process, a precursor to $M/M/1$ queue model, to model the two stage low rate DoS attack proposed by Macia-Fernandez *et al.*[115]. The adversary in the attack wants to defeat statistical analysis of their traffic, by injecting pulses of malicious traffic at foretasted points. The attack happens in two stages the first stage the adversary matches the regular traffic to occupy spaces in the servers buffer. When the buffer is full the adversary sends a pulse of packets with an increased arrival rate of their malicious traffic which DoS's legitimate traffic, but looks like just a heavy load to any analysis of the traffic. Their analysis of the attack shows that adversary has a high probability of successfully implementing the attack, but only achieves a modest disruption of the server compared to traditional flooding attacks.

An alternative uses of queuing theory in computer security research are as follows Shintre *et al.*[168] use of an open Jackson network to model optimal strategies for side channel leakage in a packet scheduler. The legitimate and malicious packets enter the scheduler according to a Bernoulli distribution, with the malici1ous packets being given a priority. The adversary wishes to discern the legitimate packet arrival pattern, and does so calculating the time difference of probe packets being sent into the scheduler. From this the adversary can develop optimal non-adaptive strategies to maximize the leakage, using the Shannon entropy in the model.

After discovering the workfactor amplification attack, described in section 4.2, the author wanted to explore disruption attacks further. To do this an abstraction was required to be able model different kinds of attacks against of availability. Given the use of queuing theory to model DoS attacks, the author felt that with a little extension that the abstraction could be developed other kinds of availability. In spite of the ease of adaptation to modelling packet level availability attacks, the focus of the research was to explore the attacks at the state machine level. Given how the abstraction was good at modelling this class of attack, a decision was made to see if a device's state machine could be mapped to a queuing network. With the selection of Osorio & Bierlaire[144] queuing network abstraction, the research focus became to develop the methodology and see what aspects of the state machine could be modelled and what kinds of attack vectors could be encapsulated at this level of abstraction.

2.1.5 Finite state machines and Context-Free Grammars in Security Research

Given the first attack that the author published attack, the credential intercept attack described in section 4.1, was developed using finite state machines and context-free grammars, a brief overview of how these techniques have been used in security research is presented here. Finite state machines have been used to validate the general promises of communications protocols for decades[30]; however, they have only recently been applied to security promises. Poll and Rutiter[151] used automata, along with black box fuzzing techniques, to show that session languages are usually poorly defined leading to vulnerabilities. Wood and Harang[206] proposed a framework for using formal language theory to secure protocols, as it is better at defining the data transiting between points of a network.

The use of context-free grammars has been applied to various security problems. Sassaman *et al.*[163] used context-free grammars and pushdown automata to create a framework for a language based intrusion detection systems. Liu *et al.*[103] used probabilistic context-free grammar to prove that an adversary could impersonate authentication server in a Point-to-Point Protocol over Ethernet protocol.

2.2 Smart Grid Security Research

This section presents the security features inherent in IEC61850 and IEC62351, a long with the features in IEC60870 and DNP3, to present the reasons as to why this class of distributed system would be a good testbed for the authors queuing network based formal method. The section then goes on to present an overview of the various avenues of research into the security of SG systems, as well as the security of time synchronisation protocols.

2.2.1 The Inherent security of SG protocols

This section explores the security promises and mechanisms that are presented in the various SG SAS. This lays out the security of IEC61850 to show why it is was selected as a test bed for the author's method. Even though this research focuses on testing IEC61850 standard, with the SG security ancillary IEC62351, to test the queuing network methodology, the security of other SASs used in the SG domain are presented here to further explain IEC61850s selection. As stated in the introduction care needs to be taken when thinking about the security of an SAS, because the without a proper security implementation the safety QoS of the standard can be compromised by an adversary. The standards do attempt to secure the SAS systems, but the designers have failed to do their due diligence as they haven't developed a threat model that covers their domain. Along with this provide no form analysis on how to assessing their countermeasures are with dealing with the threats.

More importantly none of the standard provides a complete set formalised communication models of the various functions of the standards. The formal model is required for those trying to prove the security promises, and in this case the resilience, of the standards. Also by stating the specific threats the standards are designed to prevent analysis can clearly state what is omitted, what communication models need developing to meet the designers intentions, and provide concrete metrics that can be tested against. Without a formal model that describes the acceptable internal states of the devices on the network and the tolerance for unreliable communications channels, anyone performing an analysis is left to infer the security intentions of the authors. The formalising of the desirable states allows people to formally prove that their countermeasures negate threats, and can be used in the calculation of the probability that the system ends up in an undesirable state. The system can end up in up in undesirable states without an interaction with a malicious adversary, and without a formal model it is impossible to assess if the overall system can extricate itself from the undesirable states in a QoS requirements compliant way. The communication models in all the standards makes no comment on how the system should recover from an undesirable state. This omission increases the difficulty of testing a systems recovery time, as it is left to the manufacturer of devices to implement their own recovery functions (if they do at all). The recovery of state is critical to the QoS of SGs, as these systems are designed to run for years without requiring a reboot. Consideration should also be made in the formal models about how to prevent an adversary from exploiting conflicts in security promises or counter measures to achieve their objectives. An example of this is due to the interoperability requirements secured devices must still be able to work with those that haven't, as stated in IEC62351-8. The model would also be useful for those who are developing intrusion detection systems for the communications network to discern the difference between the correct and malicious behaviour.

IEC61850 was developed with the objective to replace the various general SCADA protocols used in the SG domain with one purpose designed interoperable standard. It achieves this by declaring how IED should communicate with each other through explicitly stating the semantics of messages, a device's communication interface, and the logical relations between communication functions. It also lays out very specific QoS promises for message delivery to ensure the safe operations of the electricity distribution infrastructure. With these promises in place it can declare how a devices in a substation will handle the data and commands it receives, instead of just describing the communication channel like the generalised SCADA protocol. The encapsulation of all the functions for transmission and distribution substations into a series of modular logical nodes that can be combined together into a custom IED. The logical nodes describe the protection, monitoring and control operations that can be configured in a device, as well as how they can be automated. There are seventeen different communication models that provide session diagrams for how the logical nodes can communicate.

The standard demands that devices in the communications network always be available and to automatically recover after failure (which can be posited as a form of synchronisation of state), as well as messages between devices maintain their correctness during transmission (integrity). The standard maintains the promise of availability by specifying that there should be a "*dual port redundancy*" which requires each device on the network has access to a redundant communications network, so if the primary network is unavailable to them it can still send and receive messages. The standard sets that maximum grace time that a communication interconnect can suffer is of the order of $10ms$.These promises are to ensure the QoS requirements are met by the nodes on the communication network so they can always be able to send and receive emergency messages, such as tripping a circuit breaker, in $3ms$ in order to prevent any damage to the transmission infrastructure. The dependability of these safety messages is calculated using

$$D = 1 - P_{missing\ commands}, \tag{2.1}$$

with the accepted probability of a safety message failing being 10^{-5}. To complement this the standards' definition of the security of the communications channel is defined as

$$D = 1 - P_{unwanted\ commands}. \tag{2.2}$$

An example of an unwanted command would tripping a circuit breaker when it isn't required. The accepted probability of unwanted trips occurring on the communications network should be 10^{-8}. However the designers of the are assuming that adversarial interactions can be modelled in the same way as faults in the network. The two equations make no consideration for if an adversary sends a sequence of messages. IEC61850 makes limited comments on the security of the communications channel, and provides no consideration of the security of the device. The only resilience promise the standard makes is the deployment of access control to prevent DoS attacks. The standard states it will prevent link layer and association DoS attacks. It accomplishes this by limiting access to systems using the authorisation establishment communication model, which is described by the application association communication model in IEC61850-7-2 [120]. An attack against this communication model is demonstrated in section 4.1. From this instance an association between entities on the network can have one of five different privileges to describe what the accessing party can do to a device. If the the accessing party has no privilege to access a device, the standard dictates that this doesn't automatically mean that the device can respond as if it is being attacked. It transitions to a presumed state of attack after an unspecified number of no privilege attempts have been made.

It is the combination of the QoS promises, large number of communication models, and hierarchy of stringent real time message delivery timings that made this standard an ideal candidate for testing the queuing network method. These promises are declared to make the system robust, not to be able to withstand many classes of adversarial interaction. However its robustness is limited to ensuring the substation continues to operate regardless of operating conditions. This robustness isn't extended to the design of the communication models themselves. The models make concrete statements about the expected timings and orders of messages, but there is no thought given to the devices and functions are meant to recover from any disruptions. If these considerations are left to the implementers of the devices and software there is no guarantee that the their solutions will integrate with each other, which will violate the promise of interoperability of devices.

IEC61850 also declares that the implementer should prevent an "*attacker(s) attempts to make use of the SAS system in a way that the attacker is not authorised*", but provides very little guidance on how this should be achieved.

IEC61850 can be further secured the ancillary security standard IEC62351 which extends the security promises, and threat models, of the communication channels implemented using IEC61850, IEC60870, or DNP3 standards. The standard is written to provide the promises of the CIA triad and non-repudiation. It also presents an algorithm for preventing replay attacks using the IEC61850' GOOSE messaging service. The author describes how the replay attack prevention can be used for packet injection based DoS attacks in section 4.4. IEC62351-1 [180] does acknowledge that it is working in opposition to the QoS requirements of the SG standards it compliments. It also makes clear that despite the need for security in this domain, that secured devices will still have to be to communicate with unsecured ones without compromising security. It provides no description of how these device will exist together. The standard is written to allow for the manufacturer of devices to know what functionality their devices must have to maintain the security of the communications network. The authors have attempted to write the standard so that it "*lists precise design details and leaves little room for interpretation*"[184]. However the standards undermines this aim as it it carries on with IEC61850 focus on the security of the communications channel, but only declares what security functionality the devices should have[184]. Without clear guidance on how the security functionalities should be configured or interact with each other, the standards allow for the likely possibility that the stated promises will be undermined due to the various vendors implementing the required technologies in incompatible ways. However, the devices will still be required to communicate with each other due to the QoS requirement of interoperability, which would produce a network where the lowest common denominator of security is the defacto standard across the network.

IEC62351-3[181] describes how public key cryptography will be deployed on the IEC61850 communication networks. The threat model of this section is to use confidentialiy, integrity, and message level authentication to prevent eavesdropping, MiTM attacks, replay, and spoofing attacks. It states that only TLS 1.2 and subsequent versions should be used to encrypt the network. It also declares which cypher suites can be used. The authors have pointed out the deficiencies of this section in appendices A & A.1.4.

IEC62351-4 [182] describes how TLS will be used to work with the ISO9506 MMS standard that is used by IEC61850 to send messages across the network. The section provides application layer security to the SG infrastructure. It does explicitly allow the use of both the secure and non-secure MMS messages to be used in conjugation with each other.

IEC62351-6 [187] is the section that specifically deals with the securing the features of the IEC61850 standard. It reaffirms the standards defence against MITM and replay attacks, and providing a mechanism for being able to identify tampered messages. This section achieves this by using VLAN technologies and modifying IEC61850's substation configuration description language. However, it provides no description of how these technologies will identify the devices. It is important to note that this standard makes clear that "*applications using GOOSE and IEC 61850-9-2 and requiring 4msec response times, multicast configurations, and low CPU overhead, encryption is not recommended*".

And finally there is IEC62351-7 [184] which describes which parts of the communications infrastructure and devices should be monitored. The section makes no remarks on how these aspects should be monitored.

IEC60870 is a SCADA telecontrol standard which was later extended to be used for electrical transmission. It was designed to control devices either through point to point communications or a multidrop communications bus. This combined with the standards focus on low bandwidth, meant that no two way communication could be deployed across the standard and only the master device could initiate communication. The main advantage of this meant that message collision detection wasn't required for the standard. Only hierarchical network structures are possible in this standard, which means that if a node atthe top of a branch is disabled all subsequent tiers are also lose connectivity. The standard only defined that application level of data transmission between devices, by defining the structure of the messages and what services the standard incorporated. However, it is considered a precursor to IEC601850 as it also drove for interoperability of all devices that used the standard in any domain, so it doesn't lay out any explicit QoS promises.

DNP3 is a communications protocol that describes the communication channel between SCADA devices in any domain. It was originally designed for the use in power systems, but has been deployed in other domains due to its low bandwidth requirements and reliability features. The standard focuses on describing how it achieves its transmission goals by using its three layer OSI model. The protocol was designed for hierarchical network structures, unlike IEC61850. The standard describes many reliability features select before operate and spontaneous reporting, but it doesn't describe the behaviour of the devices when it receives a message. Given its deployment across various domains it has no declared QoS promises that its communications models can be tested against.

There has been research into the security of the protocol. East *et al.*[53] developed a taxonomy of attacks against DNP3. In it they discussed what different strength adversaries could achieve by attack different layers of DNP3's frame. They state the consequence of an adversary modifying each field of a DNP3 message. They concluded that the biggest security threats to the network is the channel no longer being confidential, and the operator losing either situational awareness or control of their network. To address the security concerns a secure authentication functionality was added to the standard. This included a key distribution and an authentication handshake communication model. Cremer *et al.*[46] used their Tamarin protocol analyser to demonstrate that the function upheld their security promise, and allowed them to make further suggestions to improve security.

2.2.2 Protocol Analysis

The focus of this research is to formally analyse the adherence of protocols to their claimed security promises. Other than Cremer's[46] work the author couldn't find any cases of formal methods being used in the SG domain. On top of that there has been limited research into the SG domain specific threats that need to be factored into the development of SG communication protocols. There are plenty of taxonomies of attacks against general smart grid technologies[134, 202, 95, 216], but only since 2010 have there been taxonomies focusing on specific attacks against IEC61850[152, 56, 154]. Most of the theorised attacks against smart grids are either derivatives of computer network exploits, or an infiltration into the smart grid's information network via compromising the affiliated corporate network. Most taxonomies put forward solutions for there proposed attacks based upon their computer network counter parts, without considering if it will conflict with the QoS promises of the protocols.

Despite this, there has been some research directly focusing on attacks using IEC61850's GOOSE multicast messaging service. Most of the attacks using GOOSE are done with the inclusion of, and sometimes caused by, the replay attack protection of IEC62351. Hoyos *et al.*[80] demonstrated a GOOSE spoofing attack where the adversary injects malicious copies of legitimate message, with an incremented *stNum*, the GOOSE variable that counts the number of events that have occured, to force the IED to ignore some of the future messages it will receive. They also flip any Boolean data in the transmitted *DatSet*. The aim of their attack is to get an IED to perform an undesirable action, such as ignoring a command to trip a circuit breaker. Strobel *et al.*[177] qualitatively expands upon Hoyos' attack vector by discussing that if an adversary replayed a GOOSE message within two minutes of a *stNum* rollover, after 2^{32} events the *stNum* counter resets, they can undermine the availability promise of an IED for up to approximately 4.5 years, presuming no one notices. Whilst Kush *et al.*[98] also model an attack against availability by using a near rollover *stNum*, they model two other attacks. In the first attack the adversary floods a GOOSE subscriber with messages that have incrementally higher *stNum* until they exceed the current value, and in the second attack the adversary injects malicious messages, with a marginally higher *stNum*, at a slightly higher rate than the regular messages. Whilst not developing an explicit adversary model El Hariri *et al.*[77] explores an important consideration for the security of the GOOSE protocols whose QoS promises demand that devices from different manufactures be interoperable. They demonstrate that different commercial devices and simulation libraries respond differently when presented with the same undesirable situations. The various implementations had differing responses with messages with old timestamps, and out of order *sTNum*, and with timestamps outside the 2 minute skew of the IEC62351 replay prevention algorithm.

There has also been some research that looks into filling the omissions in IEC62351. Tawde *et al.*[186] propose a bump in the wire key management mechanism to implement IEC62351-5, which is designed to extend the security promises of IEC60870 and DNP3. They propose connecting the bump in the wire devices to the remote terminal units and management terminal units on the network topology. They claim that this would be a practical way of bringing legacy hardware in line with the standard, but make no consideration as to whether the added latency will violate the QoS requirements that the users of IEC60870 or DNP3 need in their implementations. Fries *et al.*[64] identify that MMS messages that use multiple transport layer connections in their traversal of the network undermine the promise of integrity, as the standard assumes that an intermediary in the chain is trusted. They propose introducing security sessions into the MMS protocol to restore this promise. Ruland & Sassmannshausen[158] propose adding the security promises of non-repudiation and traceability to IEC62351-4 to enhance MMS security. They argue that the authentication of communication partner does not guarantee authentication of the origin of data transmitted. They propose encoding XML non-repudiation tokens into the data sent between the client and the server. These tokens will contain proof of authorship and a timestamp. Zhao *et al.*[221] points out that certificate revocation is not a declared part of IEC 62351-3, and propose using a broadcast encryption media key block to secure the communications in a hierarchical device structure which would allow them to revoke a device's certificate. However, they provide no proof as to if their algorithm actually meets the security promises of IEC62351-3. Fuloria *et al.*[65] discuss the various possible encryption choices, and their implementations, given the limited computational resources the communications network will have. They also develop a broader threat model which encryption can defend against. However, their analysis overlooks what could go wrong in key update algorithms. They conclude that encryption may be too great a burden to implement on smart grid network.

2.2.3 Intrusion Detection Systems

There has been a substantial body of work dedicated to the creation of bespoke intrusion detection systems (IDS) for SG communication networks.

Mitchell *et al.*[131] provide a broad, yet thorough, analysis of the current state of IDS for cyber-physical systems research. They identify several gaps in the field that have been ignored by the research community such as a greater need for bespoke performance metrics to be designed specifically for cyber-physical systems, and not enough effort has gone into looking for domain specific attack vectors.

There have been different ways suggested to model and detect malicious behaviour on SG communication networks. Examples of the methods are using stochastic process modelling[6], graph theory to specifically detect aurora type attacks[176], bloom filters[146], packet sniffing looking for forged ARP packets[153], and semantic analysis[102] have all been considered for use in this domain. The drawback of these approaches is the potential for mimicry attacks. Also these methods only look for known attack vectors as they are trained on historical malicious events from the internet domain. Another direction is machine learning, using techniques such as One-Class Support Vector Machines[116], hidden Markov Models, and k-nearest neighbour algorithms[217]. In the training of these methods the implementer would have to make sure that the training sets have no malicious activity in them or that would be allowed through the IDS.

2.2.4 Privacy

There has been a large body of work dedicated to ensuring the privacy of the consumers of electricity provided by an SG enabled transmission networks, as they generate a large amount of data about the user, to increase the efficiency of the system. This has raised questions about how this data could be used in a malicious manner, or entrench power structures. The work below describes some efforts to either define the problem, or develop remedies.

The following papers focus on defining the problem. McDaniel & McLaughlin[124] discuss the potential privacy issues of the customers within the context of current US legal statues. They identify the major source of problematic privacy practises are smart metres in the customers home's, and how that information could be abused by a supplier. They also discuss the potential for abuse by third parties, such as Google. Molina-Markham *et al.*[137] use various statistical techniques on a smart meters data to infer a detailed plan of a households daily routine. They also provide a zero-knowledge method for suppliers to aggregate billing data to prevent such data leaking. Skopik[171] lays out a broad range of potential threats of to a users privacy, whilst laying out philosophical solutions to these problems. Yang *et al.*[215] present technical threats against SG user's privacy, whilst presenting the mathematical basis for their counter measures. The following papers propose solutions to the privacy problem. Mohsenian-Rad *et al.* designed an algorithm using game theory that balances the generators scheduling of the networks loads with the users privacy needs, whilst still incentivising users to use the system.[136]. Lu *et al.*[113] developed a privacy-preserving aggregation scheme for secure SG communications that uses homomorphic Paillier cryptosystem. They also provide proofs of their security promises, but they do not consider the latency that this system will add to the communication of the data. McLaughlin *et al.*[126] developed a procedure to reduce the amount of private information leaked by smart metres through the addition of a battery between the customers residence and the gird which discharges every time an appliance is turned on, so distinct load information isn't leaked. Marmol *et al.*[140] proposed a privacy-aware protocol which hides an individual's measurements from the electricity supplier while allowing it to access the information on the aggregated energy use for their state estimations. Savi *et al.*[164] developed a mathematical framework to assess the privacy guarantee of a specific SG's communication architecture. Their methodology can also be used to assess the tradoff between a users privacy and the precision of aggregated measurements. Borges *et al.*[27] develop two quantum mechanics based protocols for preserving a users privacy, whilst being resistant to quantum computational attacks. They use quantum entanglement and quantum key distribution to uphold their security promises. They also developed the iKUP framework which is a privacy-preserving protocol, which uses DC-Nets and Elliptic Curve Cryptography, that covers power provisioning, consumption, billing, and verification.

2.2.5 Blackouts

The research into the security or QoS promises of SG communication protocols, but little thought has been given as to how the communication protocols can include communication models that negate or help manage blackouts. Bompard *et al.*[26] gives an in depth analysis of the current trends of of blackouts in electrical grids around the world. They also hypothesize about the various challenges faced by SG technology in preventing them in the future. Mahdad & Srairi[117] developed a pattern search algorithm to develop planning strategies during peak consumption periods to prevent blackouts. Chen *et al.*[39] discuss and simulate the ways power generation architecture run by distributed communication networks can minimise the probability of blackouts. Yan *et al.*[214] study the temporal effects on the power load during a blackout, and develop a simulation modelling it. This is in aid of trying to prevent such occurrences happening during a malicious cyber attack. Liu *et al.*[105] demonstrated a cyber-physical attack that harnessed coordinated variable structure switching could cause a blackout.

2.2.6 Encryption

Given the tension between the stringent adherence to the QoS for a SG communication's network and the computational overhead of required in implementing cryptographic systems, research has been done to see if it is possible for cryptographic to be used on SG networks.

Some work has been done on developing specific encryption schemes for IEC61850. Wang *et al.*[199] developed various modes of GCM encryption to protect the integrity of the different message types within the communications protocol, and Yin[218] proposes using identity based encryption.

Robles *et al.*[157] weigh up the pros and cons of asymmetric and symmetric key encryption schemes for SCADA systems that are communicating over the internet. They also note that the latency of sending encrypted instruction is an important factor in the decision of which scheme the implementer should use. Zhang *et al.*[220] do an in depth analysis on a potentially viable implementation of AES encryption. Tsai & Lo[191] develop a mathematical model to show that identity based signatures and encryption can be used to produce an anonymous key distribution scheme for SGs. He *et al.*[78] develop a homomorphic encryption scheme for smart grid communications. Zhao *et al.*[221] analyse the use of media key broadcast encryption scheme for the use key revocation, in spite of the limited computational resources that intelligent electronic devices have. Cairns *et al.*[33] explore their Time-Valid One-Time-Signature authentication protocol. They show that it is resistant to brute force attacks, and that it can go through its verification cycle in under 10 milliseconds. It should be noted that this is still too slow for IEC61850 QoS standards. Dan *et al.*[51] developed a cryptographic system that works with a modified Diffie-Hellman scheme. They prove that the scheme guarantees confidentiality, integrity, and forward secrecy, and that it functions on the order of $\tilde{1}$0ms, which is again too slow. Ward[201] developed a simulator to demonstrate that public key infrastructure "*may be pushed beyond known limits and must go where no PKI has gone before*" within the SG security domain.

2.2.7 False Data Injection Attacks

One domain specific attack vector that has been explored by the research community is the injection of false data into a SG's state estimation calculations to undermine the integrity of the control operations, automated or human, of the network. Gul & Wolthusen present a taxonomy of different objectives and strategies of false data injection attacks[74].

There is a significant body of work dedicate to the use of false data injection attacks against state estimation, where the adversary manipulates a state variable measurement in such a way as to bypass the bad measurement detection and still be able to drive the state estimation to demand an undesirable, and potentially damaging, response. It is a concern as it currently stands the only integrity checks that are done are to see if the data is transmitted correctly, not to see if the data is correct. In Liu *et al.* research they demonstrate that an adversary can approximate a grid's state through power flow analysis or injection measurements, and they also provide simulation results to demonstrate that false data injection attacks are viable attack vector[109]. Kosut *et al.* classify how false data injection attacks affect the decision making in a smart grid. The false data injection can make the state of the grid unobservable if set of meters are compromised by the adversary. The set of target meters can be discovered by the adversary using a polynomial time algorithm. They also discuss how to detect an adversary if they haven't compromised enough meters to make the state unobservable[96]. Ozay *et al.* simulated how an adversary can use the communication network's topology to effect the outcome of their attack, and showed that if the adversary only has access to a cluster of nodes they can only effect the consensus process[145]. Yu & Chin use principal component analysis to analyse the effectiveness of blind data injection and demonstrate that it can still cause damage to the transmission network[219].

Research into other attacks vectors to compromise a SG's state estimation has also been done. Work has also been done by Gul & Wolthusen to demonstrate that an adversary doesn't necessarily inject malicious data into the network, but can achieve the similar levels of disruption by reordering the measurements sent by a device[73]. Baiocco *et al.* demonstrated that introducing delays and jitter into a state estimators communication channel can force a control system to make undesirable decisions[15].

Another class of false data injection attacks is against the networks topology. Kim & Tong present the criteria required to send an incorrect representation of the network topology to the control system[93]. There has been work to look at how this kind of attack would affect different aspect of the power grid infrastructure. Such as how the communications network has either centralised, or distributed topology, affects the control systems when attacked[175, 174], and the difference of adding nonlinear effects into the attack model[110].

This attack vector would also allow the adversary to instigate an attacks against the energy market, as the state estimator is used to set the price of energy[96, 213].

2.2.8 Other Attack Vectors

Other attack vectors that are unique to the SG domain have been considered by the research community. An adversary could use the networks bad data detection system against the infrastructure, by using it to label legitimate meter sources as erroneous depriving the controller of correct information[94]. Some work has been done to see if switching attacks against the grid can be prevented. This is where the adversary, through either physical or cyber access, can manipulate circuit breakers on the power network to cause blackouts[61, 106, 108]. This attack vector is usually a physical consequence of the use of the false data injection attacks. A potential solution to the switching attack vector is using a game theory based controller system to redistribute resources to stabilise the power system. However this proposed system is dependent on it receiving accurate information from the communications network[62]. Another potential use for the false data injection attack vector is to make the system to alter the load distribution on the power grid to allow the adversary to damage certain parts of the network[135]. Jamming of packets in SG communication networks has unique consequences in this space, due to the stringent latency requirements expected of transmission of data. There has been some effort to model and mitigate such attacks. Wang *et al.*[114] undertook an in depth analysis of the consequences of jamming attacks, as well a proposing a profile detection system. Ghosh *et al.*[68] performed an analysis on the negative consequence of data packets being delayed or dropped and how to mitigate them with physical mechanisms in the power grid.

There has been some working theorising, modelling, and developing ways to mitigate DoS attacks on a smart grid's communications network. Hurst *et al.*[81] developed a mathematical framework to help security practitioners evaluate the scale of damage their communication network will suffer if they are attacked by various types of distributed DoS attacks. Liu *et al.*[107] demonstrated that a DoS attack against load frequency controls can affect the stability of the power grid. Li *et al.*[101] studied the time delay suffered by critical communication packets on an IEC61850 communications network when either the physical or application layer is flooded with malicious messages. One potential mitigation strategy against DoS attacks is to use flock based behavioural transition rules to make sure the packets avoid deigning a node availability to the network[203]. Ansilla *et al.*[10] developed a hardware based algorithm to deal with SYNflooding on SG networks.

2.2.9 Attacks Against Time Synchronisation Protocols

Due to the stringent real time QoS promises every device in the SG's communication network must keep an extremely accurate time, so as part of this project is to explore attacks the synchronisation of clocks between devices. This sections reviews how time synchronisation protocols can be disrupted by a malicious adversary.

There have been various taxonomies on the security vulnerabilities of the NTP/PTP protocols where the adversary can either manipulate or control the network[83, 67, 119, 83, 67]. Each taxonomy has proposed countermeasures, such as introducing the CIA triad into this domain and basing the protocol on the P2P network paradigm. Ullmann & Vögeler[194] performed an analysis on the consequence of a delay attack against both the NTP and PTP protocols. They proved that a delay in a sync message would affect all the client clocks, and a delay request message would only affect the client that sent the message. They propsed implementing SHA on the protocol to mitigate these attacks. Tsang & Beznosov[192] created a qualitative taxonomy of attacks against the PTP protocol. They laid out how an adversary could potentially misuse certain messages in the protocol to drive the protocol into an undesirable state. Mizrahi[132] developed some game theoretic strategies to prevent delay attacks against NTP. Malhotra *et al.* [118] demonstrated that NTP's "kiss-o-death" packet can be used to DoS any client on a NTP network, and denying it the ability to synchronise. There has also been research on how the NTP protocol can be used to generate distributed denial of service (DDoS) attacks. Czyz *et al.* [49] performed an analysis of DDoS on the internet that were achieved using unsecured NTP servers, seeing substantial increases of these kinds of attacks between 2013 and 2014. The increase was due to adversaries realising that NTP's `monlist` diagnostic command could be used as a work factor amplification attack vector.

Moussa *et al.*[139] produced a detailed a detailed analysis of the consequences of a delay attack in a SG substation environment. They also provided a mathematical model to counter delay attacks.

Chapter 3

Queuing Network Based Formal Method

This chapter describes the author's queuing network methodology that is used to test the limits of robustness of security protocols. The first section provides an overview of the foundational queuing theory results which provide the base abstraction for the author's formal method. This section goes onto describe the specific features encapsulated in the author's formal method. The second section goes over the proofs of correctness for the author's implementation of the formal method. Before finally comparing the queuing network methodology with other formal methods described in section 2.1.

3.1 The Queuing Network Methodology

This section provides a foundation of queuing theory, focusing on queuing networks, before going onto describe the author's formal method.

3.1.1 The $M/M/c/K$ Queuing Network

Queuing theory uses stochastic process to model and analyse the nuances of objects being processed by a queue. The abstraction of queue is defined by five exogenous variables, represented in Kendall notation as $A/B/X/Y/Z$[89]. A & B are probability distribution that govern the arrial of objects entering a queue and the rate which they are processed at. X is the is the number of processors each queue has. Y is the maximum number of objects a queue can hold. Z, states discipline that a queue's processor selects the next waiting objected to be processed. A network of queues is a set of these objects that form a graph, with packets being passed along the vertices. The author's formal method builds upon the $M/M/c/K$ queueing network developed by Osorio & Bierlaire, where M/M represent the that the arrival of objects occur according to a Poisson process and processed according to an exponential. Due to each queue in the network having a set truncation limit there has to be a stated blocking discipline for network. The network used in this formal method use a blocking at service (BAS) discipline, which means that all subsequent target queues are full, then the server of that queue are cannot process another object until a queue is able to receive a packet. This discipline was chosen as it allowed for the encompassing of attacks against availability, as it represents a device or node no longer being able to accept any more packets.

This kind of queuing network is the base abstraction of the author's probabilistic formal method that allows for the testing of a device's state machine or a network of devices to find the limit of correction operation when subjected to adversaries who seek to disrupt the completion of a session. Each queue in the network represents either an individual state in a honest agent's state machine, with K represent the maximum number of sessions that a device can have in this state, or a node in the network, with K being the size of the buffer. In this abstraction the processing and arrival rates describe the communication channel between the honest agents, or they can be mapped onto the properties of packets. The adversary in this formal method is

generated by stating which of these rates in the network they can manipulate.
The assumptions of the formal method are:

1. Each queue obeys the first-in-first-out (FIFO) discipline for processing packets (its Z).
2. That the transition between states is memoryless.
3. The network is in the steady state.

Given assumptions 2 & 3 continuous time Markov chains (CTMC) is used to calculate the probability that the queuing network will be in a particular state. This is done by solving the state spaces' global balance equation. The formal method further assumes that the network is in the steady state, an approximation where the network's traffic doesn't change over time, which requires the state transitions to be:

- Independent of time.
- Independent of the initial state vector.

If the CTMC is ergodic then an unique steady state probability vector, $\boldsymbol{\pi}$, exists that is independent of any initial probability vector, which means the global balance for each state space can be described by the conservation of probability flux in and out of each state,

$$\sum_{j \in \mathcal{I}} \pi_j q_{ji} = \pi_j \sum_{j \in \mathcal{I}} q_{ij}, \tag{3.1}$$

where π_j is the probability of being in a state, q_{ij} is the rate of transmission between state i and j, and $\mathcal{I}$ is the state space. From the steady state assumptions equation 3.1 can be rearranged into the matrix form,

$$,\mathbf{0} = \boldsymbol{\pi Q} \tag{3.2}$$

where Q is the transition matrix that encapsulates the probability flux for each state. Every transition rate between any pair of states flowing out of a state is stored in the non-diagonal elements, and the sum of all the transitions flowing into a state are stored in the diagonal elements. The steady state vector is found by solving the systems of linear equations described in 3.2 using the boundary condition

$$\sum_{i \in \mathcal{S}} \pi_i = 1. \tag{3.3}$$

Once the steady state vector is calculated the marginal probabilities of the state space can be discerned. These give the probability of a state having a certain configuration of packets, which is dependent on the state space selected. The various configuration developed for the author's formal method are:

- The number of packets (agnostic of packet type) in every queue in the network.
- The number of packets of each type that are currently being processed, blocked from transmitting, or waiting to be processed, either in an individual queue or every queue in the network.
- The order of the packets of each type in either an individual queue or every queue in the network.

These state spaces are defined in section 3.2. The marginal probability is calculated by summing the probabilities of all the states that have a specific value packets in an element in the state space from the initial state vector.

3.1.2 Using the $M/M/c/K$ Queuing Network

To test a specific state space or network of nodes, a queuing network representing these must be defined. Each queue in the network either represents a state in a honest agent's state machine for a particular protocol session, or a node in a network of devices communicating via specific protocol. Each queue is a vertex in the network, and the edges between each queue are declared in the set of matrices $p^{\tau}_{i,j}$. Where i & j represent queue indices, and τ is the index for the specific packet type. Each element in the matrix gives the likelihood of a queue transferring a packet to another queue, and so for the unitarity of the transfer probabilities to be kept

$$\Sigma_{\tau}\Sigma_{j}p^{\tau}_{i,j} = 1. \tag{3.4}$$

In this formal method it is possible for several different packet types can be passed through the network in a simulation, unlike other queuing theory models developed in security. The level of encapsulation the formal method was designed for is to differentiate between expected packets and malicious ones, and not specific message semantics or content. Whilst it is possible to create a new packet type for each kind of message this would drastically a computations state space, and thus run time.

Each queue in the network has set of declared exogenous variables, that describe how efficiently it would process packets if it were not in a network. These are the variables manipulated by the adversary over many iterations of a specific queuing network to find their limit of correct operation. The exogenous variables are shown in table 3.1

Exogenous variable	Symbol	Interval
Truncation Limit	K_i	$\mathbb{Z}^+$
Number of Processors	C_i	$\mathbb{Z}^+$
External Arrival Rate	γ^{τ}_i	$\mathbb{R} \geq 0$
Processing rate	μ^{τ}_i	$\mathbb{R} \geq 0$

Table 3.1: Exogenous variables that must be declared for each queue in the network.

A queue may have multiple packet types passing through it, so for each type a specific γ & μ will need to be defined. From defining the exogenous variable seven endogenous variables must be calculated to discern how a queue functions within the network given the potential for queues to become blocked. These variables are calculated by solving a system of non-linear equations. The endogenous variables are shown in table 3.1.2.

Exogenous variable	Symbol	Interval
Probability Queue is Full	$P(N_i = K_i)$	$1 \geq \mathbb{R} \geq 0$
Probability Queue is Blocked	$\mathcal{P}_i$	$1 \geq \mathbb{R} \geq 0$
Arrival Rate	λ^{τ}_i	$\mathbb{R} \geq 0$
Effective Arrival Rate	$\lambda^{\mathit{eff}\ \tau}_i$	$\mathbb{R} \geq 0$
Common Acceptance rate	$\mu^{\bar{a}\ \tau}_i$	$\mathbb{R} \geq 0$
Effective service rate	$\mu^{eff\ \tau}_i$	$\mathbb{R} \geq 0$
Traffic Intensity	ρ_i	$1 > \mathbb{R} > 0$

Table 3.2: Endogenous variables that are solved for for each queue in the network.

The traffic intensity for all the queues will be less than 1 to keep it in steady state regime. In the author's implementation the probability of a queue being full or being blocked only have one form, whereas the last five are dependant on which probability distributions selected for a queue's arrival and processing rate. The first two equations are calculated using

$$P(N_i = K_i) = \frac{(1-\rho_i)\rho_i^{K_i}}{1-\rho_i^{K_i+1}} \tag{3.5}$$

$$\mathcal{P}_i = \sum_j \bar{p_{ij}} P(N_j = K_j) \tag{3.6}$$

where the mean probability of transmission matrix $\bar{p_{ij}} = \Sigma_{\tau}p^{\tau}_{ij}$ The three different probability distributions used in the author's implementation are exponential, Poisson, and uniform. These are included so the user of the formal method can model different kinds of packet traffic entering a node/state machine. These inclusion of various probability distributions for the arrival and processing rates was implemented to further push the limits of the range of modelling queuing theory could do in the security domain. Other probability distributions could be added to the methodology to model different packet and adversary behaviour. The forms of the next four variables for each distribution are shown the three tables below.

Variable	Poisson
Arrival Rate	$\lambda_i^{\tau} = \frac{\lambda_i^{e\bar{f}f\ \tau}}{1-P(N_i=K_i)}$
Effective Arrival Rate	$\lambda_i^{eff\ \tau} = \gamma_i(1 - P(N_i = K_i))$ $+\Sigma_j p_{j,i}\lambda_j^{eff\ \tau}$
Effective service rate	$\mu_i^{eff\ \tau} = \mu_i^{\tau} + \mathcal{P}_i\mu_i^{\bar{a}\ \tau}$
Mean Acceptance rate	$\mu_i^{\bar{a}\ \tau} = \Sigma_{\{j\mid p_{i,j}\neq 0\}} \left(\frac{p_{i,j}P(N_j=K_j)}{\mathcal{P}_i}\right)^2$ $\left(\frac{\lambda_i^{eff\ \tau}C_j\mu_j^{\bar{e}ff}}{\lambda_j^{\bar{e}ff}}\right)$

Table 3.3: Equations for the endogenous variables using the Poisson distribution.

Variable	Exponential
Arrival Rate	$\lambda_i = \lambda_i^{\bar{e}ff}(1 - P(N_i = K_i))$
Effective Arrival Rate	$\lambda_i^{eff\ \tau} = (\frac{(1-P(N_i=K_i))}{\gamma_i}$ $+\Sigma_j \frac{p_{j,i}}{\lambda_j^{eff\ \tau}})^{-1}$
Effective service rate	$\mu_i^{eff\ \tau} = \left(\frac{1}{\mu_i^{\tau}} + \frac{\mathcal{P}_i}{\mu_i^{\bar{a}\ \tau}}\right)^{-1}$
Mean Acceptance rate	$\mu_i^{\bar{a}\ \tau} = \Sigma_{\{j\mid p_{i,j}\neq 0\}} \frac{\lambda_j^{\bar{e}ff}}{\lambda_i^{eff\ \tau}C_j\mu_j^{\bar{e}ff}}$

Table 3.4: Equations for the endogenous variables using the exponential distribution.

Variable	Uniform
Arrival Rate	$\lambda_i = \frac{\lambda_i^{\bar{e}ff}}{1-P(N_i=K_i)}$
Effective Arrival Rate	$\lambda_i^{eff\ \tau} = \frac{(a+b)(1-P(N_i=K_i))}{2}$ $+\Sigma_j p_{j,i}\lambda_j^{eff\ \tau}$
Effective service rate	$\mu_i^{eff\ \tau} = \frac{a+b}{2} - \mathcal{P}_i\mu_i^{\bar{a}\ \tau}$
Mean Acceptance rate	$\mu_i^{\bar{a}\ \tau} = \Sigma_{\{j\mid p_{i,j}\neq 0\}} \left(\frac{p_{i,j}P(N_j=K_j)}{\mathcal{P}_i}\right)^2$ $\left(\frac{\lambda_i^{eff\ \tau}C_j\mu_j^{\bar{e}ff}}{\lambda_j^{eff}}\right)$

Table 3.5: Equations for the endogenous variables using the uniform distribution. Where a & b are the uniform distribution parameters.

The mean acceptance rate is an approximation used by Osorio & Bierlaire[144] to reduce the computational complexity of the methodology, which the author carried over to their formal method. The true value would require the calculation for a queue to account for all the subsequent queues that its packets passed through.

The equation for the final variable, traffic intensity of a queue, is dependant on the probability distributions selected for a queues' arrival an processing rates. The traffic intensity is a unitless quantity so restricts the combinations of probability distributions can be selected for a queue. If a queues' λ obeys the Poisson distribution,that has the unit s, then it's μ must either be exponential or uniform, which have the units s^{-1}. With the current selection of distributions included in author's implementation of the formal method the two possible traffic intensity calculations can be,

$$\rho_i = \begin{cases} \frac{\lambda_i^{eff}}{C_i \mu_i^{eff}} \\ \frac{C_i \mu_i^{eff}}{\lambda_i^{eff}} \end{cases} \tag{3.7}$$

with the first case being the queue's arrival rate is specified as an exponential distribution and the processing rate is either a uniform or Poisson distribution, and the second case being the reversed.

3.1.3 Queuing Network Performance Measures

Queuing theory provides a selection of performance measures that are calculated using the marginal probabilities of the state space being used to study the network3.1.3. All are written in the context of the state space that shows the agnostic number of packets in each queue, but some metrics can be rewritten in the context of the other views[25, 72].

Performance Metric	Equation
Traffic Intensity	$\rho_i = \sum_{k=1}^{k} \pi_i(k)$
Throughput	$\lambda_i = \sum_{k=1}^{k} \pi_i(k)\mu_i^{eff}$
Total Throughput	$\lambda = \sum_{i=1}^{N} \lambda_{0i}$
Mean Number of Packets	$\bar{k_i} = \sum_{k=1}^{k} k\pi_i(k)$
Mean Queue Length	$\bar{q_i} = \sum_{k=c_i}^{k} (k - c_i)\pi_i(k)$
Mean Response Time	$\bar{T_i} = \frac{\bar{k_i}}{\lambda_i}$
Mean Wait Time	$\bar{W_i} = \bar{T_i} - \frac{1}{\mu_i^{eff}}$
Mean Number of visits	$e_i = \frac{\lambda_i}{\lambda}$
Relative Utilisation	$x_i = \frac{e_i}{\mu_i^{eff}}$

Table 3.6: Probabalistic performance measures of queuing theory

Since these measures are agnostic to the types of packets that pass through a queue, not all of them can be applied to each of the state space configurations used by the methodology. The table below shows how each of the performance measures are applied to the five state space configurations that are described in section 3.2.

Performance Metric	GK	IO	GIO	IS	GIS
Traffic Intensity ρ_i	✓	✓	✓	✓	✓
Throughput λ_i	✓				
Total Throughput λ	✓				
Mean Number of Packets $\bar{k_i}$	✓	✓	✓	✓	✓
Mean Queue Length $\bar{q_i}$	✓				
Mean Response Time $\bar{T_i}$	✓				
Mean Wait Time $\bar{W_i}$	✓				
Mean Number of visits e_i	✓				
Relative Utilisation x_i	✓				

Traffic intensity and mean number of packets were the only metrics applied to internal order and internal states state spaces as they provided further insight into the behaviours of the specific aspects of the queue. Traffic intensity shows the likelihood of a particular type of packet being in a queue position or being blocked, whilst mean number of packets shows the number of the types of packets. The other performance metrics are designed to give the user an insight into the internal state of the queue without having generating the state space, so applying them to the internal states and orders doesn't provide any further insight.

3.1.4 Algorithms used in the author's implementation

The author's implementation of the queuing network methodology is structured into four main parts. The first part are initial calculations that only need to be done once for a run of the implementation. These are

- Making a list of all the exogenous variables that the user has selected for the adversary to manipulate.
- Making lists of the queues a queue can transfer each type of packet to.
- Making a list of which packets pass through each queue.

The next part of the implementation is solving the set of non-linear equations for the endogenous variables of the queues in the network. This involves setting up the matrix where each row is a stores an iteration of the queuing network that has had an exogenous variable altered by the adversary. Each row has an element for each exo- & endogenous variable for every queue in the network. The creation of the matrix is dependent on whether the user wants one set of exogenous variable variations done one after other or recursively nested within each other. Another factor in the generation of the matrix is whether the queues all share the same probability distributions for their arrival & processing rates (if there is queues with a different probability distribution the implementation uses the central limit theorem for the calculation). Once this is done the indexing variables for each non-linear equation is deduced, before using Matlab's *fsolve* function, which uses the Levenberg-Marquardt algorithm[138], t solve them all.
The next stage of the code is the generation of the state spaces and creation of state transition matrix. The state space is recycled for each Q generation, unless a queuing network variation alters a parameter that affects the state space. These parameters are discussed in section 3.2. The generation of each transition matrix requires inserting the variations of endogenous variables into them. From each Q a steady state vector is calculated. This is using MATLAB's *eig* function[8] to solve the eigenvalue problem described by equation 3.2.
The final part is the calculation of each state spaces marginal probability from those the performance measures.
The correctness of the creation of the topological matrix and the state space are proven in section 3.2.

3.1.5 The calculation and use of conditional probability

Conditional probability is used in the author's implementation to allow for the adversary to infer information across the queuing network. A compromised queue or tapped network edge can be used by an adversary who has access conditional probability abilities to infer what has passed through other queues or edges that they are not directly connected to. They conditional probability calculations use the endogenous variables and marginal probabilities of other queues around the compromised aspect of the network to provide the adversary this information. These probabilistic inferences can be used to alter the adversaries behaviour on the network. There are two parts of the author's implementation where conditional probability is used. The first point where conditional probability can be used is during the calculation of the non-linear simultaneous equations, described in section 3.1.4. During the calculation of the endogenous variables for each variation the implementation can calculate the probability of a selected variables across all their variations so far. The variables can be any of the exogenous and endogenous variables in the network. If the probability crosses a set limit there is an option for the implementation to alter the exogenous variables and generate a new set of experiments. This can be used to demonstrate an adversary's ability to infer aspects of an agents state machine, or infer what packets are passing along communication channel that they don't have a tap on. This switching mechanism can only be activated once per run of the implementation, to prevent the code being stuck in recursive state without any exit.
The second point where conditional probability can be calculated occurs after the computation of the selected performance metrics. This conditional probability calculation occurs across the entire set of network experiments (including those if a conditional probability switch). It is just a one off conditional probability calculation of selected variables. The variables can be

selected from any of the network's exogenous or endogenous variables, as well as any marginals calculated, and selected performance metrics.

3.2 Proofs of Correctness of the Queuing Network Methodology

This section presents proofs that the author's implementation of the queuing network formal method provides the correct results for solving the non-linear equation system of endogenous variables in the network, and creating and solving the state space of a given network. Whilst the algorithms used, and thus their proofs, used in the formal method are agnostic of a particular procedural programming language, they are shaped the data structures used. The author used Matlab for their implementation which primarily stores data in vectors and arrays. The complexity of the methodology could be improved by implementing it using different data structures, which would change the proofs in this section. The intent of this work was to present a proof of concept of the formal method to demonstrate that the robustness of security protocols can be modelled and tested.
The first section goes over the proofs of the creation of the matrix that stores all the experiments over selected exogenous variables of a given network, (described by a set of $p_{i,j}$). The definitional axioms of a given network are laid out, before proving that the loop invariant used in the creation the matrix is correct and complete. Finally the equations used to link the variables in the system of non-linear equations are demonstrated to be correct.
The subsequent three sections lay out the proofs for the generation of each kind of network state space, as well as the solution of their state transition matrix Q. The first section presents the configuration of just the number of packets in a queue. The next section presents the proofs of the ordering of packets in a queue, as well as all queues across the network. And the final section lays out the proofs that were completed during the course of this research for the packet in various queue states for a queue, as well as all the queues in the network. This section also lays out some conjectures on how the remaining proofs for this configuration could be demonstrated. The layout of these sections, and the proofs themselves, follow a similar structure. The first step is define what is encapsulated by the state space and the rules for transitioning between states, so that the rules can be shown to be sound. Next is the proof showing how the state space is constructed. Each of these construction proofs is different as the each state space configuration uses different exogenous variables in their definition. Next the various features of the transition matrix Q are proven, before showing the construction of the grouping rules, that are based on the algorithm of state space generation. These rules reduce the complexity of creating Q and calculating the marginal probability. The final proof of each section shows that the algorithm for creating the Q terminates.
Within this section the author will lay out the complexity of their own implementation of the queuing network formal method.

3.2.1 Proofs of Correctness for a Network's Exogenous Variations Matrix

This section goes through the proofs of the generation exogenous variable problem space of a given network. The base axioms of any network modelled in the formal methods are:

Definition 3.2.1.1. The network must not be a disconnected graph, as the method can only represent one network at a time.

Definition 3.2.1.2. The set of packet transfer probabilities matrices should meet the condition $\mathcal{P}_i = \sum_j \bar{p_{ij}} P(N_j = K_j)$, previously stated in equation 3.7.

Definition 3.2.1.3. The number of processors of a queue must be, and always remain, less than or equal than the truncation limit of a queue, $C_i \leq K_i \forall i$.

In addition to the above definitions the author's implementation of the formal method includes these additional conditions to ensure code termination.

- If a run of the implementation runs across a set of the exogenous variable variations, that the parameters in the set must be reachable using the provided step size provided.
- The initial parameter of an exogenous variable must be greater than its final parameter for a given set parameters that will be computed over in a run.
- All $p_{i,j}^{\tau}$ must be immutable over the course of a run.

These axioms are checked for at the beginning of each run of the implementation. If any of the axioms are violated the code terminate with an error. Given the above axioms required for termination, it is now possible to define the loop invariants required for the generation of the matrix, and then use them to prove the correctness of the problem space. A loop invariant is a property of a programmatic loop that remains true for each iteration of a loop[66].
To generate the problem space for each variation of a queuing network a list of any of the exogenous variables that have been given a set of parameters to compute across must be created. The network's problem space is the set of each set of exogenous variable variations. There are two ways of generating this problem space. The first is to go through each variable set sequentially and create an iteration of the network for each. The second way is to generate the problem space recursively, where each variable set is nested within the subsequent set.
The loop invariant for the creation of the problem space is the cardinality of the set of exogenous and endogenous variables for a given queuing network. Each row of the matrix must have this many elements in it.

Theorem 3.2.1.1. *The set of queuing network variables is correct.*

Proof. For the set of non-linear equations to be solved each exogenous variable of each queue must have an value for as defined in table 3.1. An external arrival rate and initial processing rate for a queue must be declared for all packet types that pass through the network, even if a specific packet type don't pass through a particular queue (in this case the value is set to zero). There must also be a column for each endogenous variable for the queues on the network. For the six variables that are packet type dependent, each one has an extra column for the mean of that variable across the types. This means the matrix must have

$$N_{Col} = N_{queue}(5 + 6(N_{\tau} + 1)) \tag{3.8}$$

columns for each iteration of the network.
In the author's construction the first $N_{queue}(3 + 4(N_{\tau} + 1))$ columns are zero columns to hold the endogenous variable's values. The other $2N_{queue}(2 + N_{\tau})$ store the exogenous variables for each iteration. □

Theorem 3.2.1.2. *The set of queuing network variables is complete.*

Proof. There are two methods for constructing the complete set of network variations. The first is the sequential order of all the exogenous variables that have been parameterised. This is done by looping through all the exogenous variables and seeing if any of the queues are parmeterised in that variable, for external arrival and processing rate there is a sub loop for each packet type. Once the list is constructed then the set containing all the possible variations of the variable is inserted into the matrix, with every other exogenous variable of the network remaining at its initial value.
The other construction of the complete set of queue iterations is recursively. This is when a set of a variable's parameter are embedded within another set of parameters, and so on. Instead of looping through the list of variables and inserting the sets into the matrix, the set of parametrised variables of the last member of the list is inserted between each member of the set of the penultimate variable, and so on until the the first variable in the list. □

Before the non-linear equation system can be solved the column indices of each variable required for a specific endogenous equation must be linked to the specific column of that endogenous variable. The linking algorithm loops through each queue index, and then subloops through each packet type. For this loop system the invariant is equations used to calculate the linked variables in the equation. For each type of equation, presented in tables **??** – 3.1.2, there is an individual correctness proof, but they all share the same completeness proof. All the proofs for the endogenous variables, except for the common acceptance rate, are the same regardless of what probability distribution is used, as each equation uses the same variables regardless of distribution.

Proposition 3.2.1.3. *The linking equations used in the calculation of the probability of a queue being full,$P(N_i = K_i)$, is correct.*

Proof. The calculation of the probability of a queue being full requires a queue's traffic intensity, ρ_i, and their truncation limit, K_i. Given the construction of the invariant given in the proof of theorem 3.2.1.1, the queue's traffic intensity index is $N_{queue}(2 + 4(\tau + 1))$ columns away from the $P(N_i = K_i)$'s index, and the truncation limit is $N_{queue}(3 + 6(N_\tau + 1))$ away. □

Proposition 3.2.1.4. *The linking equations used in the calculation of the arrival rate for a specified packet type, λ_i^τ is correct.*

Proof. The calculation of the arrival rate requires a queue's probability of a queue being full,$P(N_i = K_i)$ and their effective arrival rate for the packet type in question, $\lambda_i^{eff\ \tau}$. Given the construction of the invariant given in the proof of theorem 3.2.1.1, the queue's probability of being full index is $-N_{queue}(1+4(\tau-1))$ columns away from the λ_i^τ's index, and the effective arrival rate is $2N_{queue}$ away. □

Proposition 3.2.1.5. *The linking equations used in the calculation of the effective service rate for a specified packet type, $\mu_i^{eff\ \tau}$ is correct.*

Proof. The calculation of the effective service rate for a packet type requires a queue's service rate,μ_i^τ, the common acceptance for the packet type, $\mu^{\bar{a}\ \tau}{}_i$, and their probability of being blocked,$\mathcal{P}_i$. Given the construction of the invariant given in the proof of theorem 3.2.1.1, the queue's service rate index is $N_{queue}(5N_\tau - 3\tau + 9)$ columns away from the $\mu_i^{eff\ \tau}$'s index, the common acceptance is $2N_{queue}$ indexes away, and the probability of being full is $N_{queue}(4(N_\tau - \tau) + 7)$. □

Proposition 3.2.1.6. *The linking equations used in the calculation of the effective arrival rate for a specified packet type, $\lambda_i^{eff\ \tau}$ is correct.*

Proof. The calculation of the effective arrival rate for a packet type requires the queue's probability of a queue being full,$P(N_i = K_i)$ and their initial arrival rate for the packet type, γ_i^τ. The calculation also requires the effective lambda value of any queue that feeds that packets type into the queue. Given the construction of the invariant given in the proof of theorem 3.2.1.1, the queue's probability of being full index is $-N_{queue}(3+4(\tau-1))$ columns away from the $\lambda_i^{eff\ \tau}$'s index, and the initial arrival rate is $N_{queue}(4N_\tau - 3\tau + 7)$ away. The difference in indices of effective lambda values is given by $j - i$ where j is the index of the feeding queue. □

Proposition 3.2.1.7. *The linking equations used in the calculation of the mean acceptance rate for a specified packet type, $\mu^{\bar{a}\ \tau}{}_i$ is correct.*

Proof. There are two cases in the calculation of the common acceptance rate, depending on which probability distribution is used. In both cases the calculation requires the queue's effective arrival rate, $\lambda_i^{eff\ \tau}$ and the mean arrival rate, $\lambda^{e\overline{ff}\ \tau}{}_j$, mean processing rate,$\mu^{e\overline{ff}\ \tau}{}_j$, and the number of processors, C_j, of the queues that the queue of interest feeds into. Given the construction of the invariant given in the proof of theorem 3.2.1.1, the queue's effective arrival rate index is $-N_{queue}$ columns away from the $\mu^{\bar{a}\ \tau}{}_i$'s index, and subject queue's mean arrival rate is $N_{queue}(4 * (N_\tau + 1 - \tau) - 1) - (j - i)$ away. Also the subject queue's mean processing rate is $N_{queue}(4(N_\tau + 1 - \tau) - 2) - (j - i)$ away, and finally the number of processors is $N_{queue}(6N_\tau - 4\tau + 10)$ away.
Additionally, when either the uniform or Poisson distributions are used, the calculation requires the queue's probability of being blocked, which is $N_queue(4(N_\tau - \tau) + 5)$ columns away, and the subject queue's probability of being full, which is $-4N_{queue}\tau + (j - i)$ away. □

Proposition 3.2.1.8. *The linking equations used in the calculation of the probability of a queue being blocked, $\mathcal{P}_i$, is correct.*

Proof. The calculation of the queue's probability of a queue being blocked,$\mathcal{P}_i$, and the queue's effective arrival rate for the packet type, $\lambda_i^{eff\ \tau}$. Given the construction of the invariant given in the proof of theorem 3.2.1.1, the queue's probability of being full index is $-N_{queue}(1+4(\tau-1))$ columns away from the $\mathcal{P}_i$'s index, and the effective arrival rate is $2N_{queue}$ away. □

Proposition 3.2.1.9. *The linking equations used in the calculation of traffic intensity, ρ_i, is correct.*

Proof. The calculation of the traffic intensity requires the queue's mean effective arrival rate,$\overline{\lambda^{eff}}_i$ and the mean effective processing rate, $\overline{\mu^{eff}}_i$, and the number of processors the queue has, C_i. Given the construction of the invariant given in the proof of theorem 3.2.1.1, the queue's mean effective arrival rate index is $-3N_{queue}$ columns away from the ρ_i's index, the mean effective processing rate is $-4N_{queue}$ indexes away, and the number of processor is $2N_{queue}(N_\tau+1)$. □

Where in the above proofs $\tau \in [1, N_\tau+1]$ to include the mean value of the variable. Before proving the completeness of for the above set of invariants, a discussion for including the central limit theorem when calculating the effective arrival and processing rates must be discussed. If all the queues in the network are using the same probability distributions for their lambda and mu calculations then the central limit theorem isn't required, but if a single queue is using a different distribution then it will be. If is used then two rows are inserted between each parameterised set of variables.

Theorem 3.2.1.10. *The linking equations described in the proofs to proposition* 3.1.2.3 *to* 3.1.2.9 *produce a complete set of equations*

Proof. There are two cases of completion. The first is if the central limit theorem isn't used. In this case the complete set of all the linked equations above is just all the rows in the matrix. If the central limit theorem is used then the indices of the pairs of rows is inserted for calculation of the central limit theorem are withdrawn from the list of all matrix's rows. □

3.2.2 Proofs for the Global K State Space

This section lays out the theorems used in the generation of and use of the state space that shows the agnostic number of packets in every queue in the network (that shall from here be referred to as the global K state space). The axioms of this state space are:

Definition 3.2.2.1. The states space $\mathcal{S}^{GK}$ for the global K is defined as

$$\mathcal{S}^{GK} = \{n_1, ..., n_N | 0 \leq n_i \leq K_i\} \tag{3.9}$$

where K_i is the truncation limit of a queue, defined in table 3.1.

There are three types of state transition in this state space. Transmission into the network, I_i, transmissions out of the network, O_i, and transmissions between queues T_i. These transitions are described below.

Definition 3.2.2.2. The transition rules of the global K state space are:

Label	State Transitions	Rate	Conditions
I_i	$(n_i, ...) \rightarrow (n_i+1, ...)$	$-\overline{\lambda_i}$	$\gamma_i \neq 0$ & $n_i \leq K_i - 1$
O_i	$(..., n_i) \rightarrow (..., n_i - 1)$	$-\overline{\mu_i^{eff}}$	$p_{i,\forall j} = 0$ & $n_i \geq 1$
$T_{i,j}$	$(..., n_i, ..., n_j, ...) \rightarrow (..., n_i - 1, ..., n_j + 1, ...)$	$-\overline{\mu_i^{eff}}$	$p_{i,j} \neq 0$ & $n_i \geq 1$ & $n_j \leq K_j - 1$

where are all the rates and conditions are described from the perspective of transitions flowing out of state i.

In the author's implementation the first step is to generate the chosen state space for the network's exogenous variables.

Proposition 3.2.2.1. *The algorithm that creates the state space for a given queuing network, as defined in section 3.2.1, only generates feasible and reachable states by the transition rules given in definition 3.2.2.2 and a given set of truncation limits $\mathcal{K}$ and a mean probability of transmission matrix $\bar{p_{i,j}}$.*

Proof. For the state space $\mathcal{S}^{GK}$ to contain every feasible state is every possible state of a queue in the a given network must be permuted with every possible state of every other queue in the network. If a queue receives packet exogenously (from here on referred to as a source queue) or there is a path from a source queue to a non-source queue through $\bar{p_{i,j}}$ then the queue's state will be within the interval $n_i \in [0, K_i]$, otherwise its state space is $n_i \in \{0\}$ so long as it doesn't violate definition 3.2.1.1. The set of a queue's possible states is defined as $\mathcal{N}_i$.
To generate evert feasible state for $\mathcal{S}^{GK}$ each queue's $\mathcal{N}_i$ undergoes two duplication operations. The first operation duplicates each member of $\mathcal{N}_i$, becoming $\mathcal{N}_i'$, so each state can be matched with every state in the subsequent queues state space. To equal the cardinality of the subsequent queue's state space a duplication factor must,DF_i, is calculated for every queue in the network. This factor is given by the equation,

$$DF_i = \Pi_{j=i+1}^{N_N}(K_j + 1) \tag{3.10}$$

where $DF_N = 1$ as it has no subsequent sets to be duplicated against.
The second operation duplicates the set $\mathcal{N}_i'$, so that it matches the number of states in the proceeding queue. As the proceeding queue will also have been duplicated, and recursively on until the first queue, the number of times that a $\mathcal{N}_i'$ is repeated is given by SR_i. SR_i is calculated using the total number of states, $|\mathcal{S}^{GK}|$, which is equivalent to the number of states in the set $|\mathcal{N}_1'|$. The second duplication factor is given by the expression

$$SR_i = \frac{|\mathcal{S}^{GK}|}{|\mathcal{N}_i'|}. \tag{3.11}$$

These two duplication processes provides every feasible state in a given queuing network with a defined set of queue truncation limits $\mathcal{K}$.
The state space contains every reachable state as the queuing network can't be a disconnected graph, as stated by definition 3.2.1.1. This means it is possible for packets to pass through every queue in the network given the transition rule $T_{i,j}$ stated in definition 3.2.2.2. Packets will only pass through the network if a source queue is present which utilises rule I_i. If there isn't a source queue, or there are queues upstream from a source, then their individual state spaces $\mathcal{N}_i = \{0\}$ as defined at the beginning of this proof. □

Corollary 3.2.2.2. *The number of states in a global K state space is,*

$$|\mathcal{S}^{GK}| = \Pi_{i=1}^{N_{queues}}(n_i + 1), \tag{3.12}$$

where $n_i \in [0, K_i] \vee 0$, depending on whether packets are passing through the queue i. As stated in proposition 3.2.2.1 every reachable and feasible state is every state of a queue being permuted with every state of every other queue. This calculation uses this statement with multiplication principle of sets to give the result.

For the transition matrix Q to be correct the transition rules it is generated from must be sound. Below are the proofs for the transition rules stated in definition 3.2.2.2.

Proposition 3.2.2.3. *The transition rules of the global K state space, as given in definition 3.2.2.2, will not lead to any state collision.*

Proof. Let $A = \{I_i, O_i, T_{i,j}\}$ be the set of transition rule. If all three transition operations were applied to the same state, the concluding state for each operation would be different because the modification that each transition rule applies to a state is different. □

Proposition 3.2.2.4. *The transition rules of the global K state space, as given in definition 3.2.2.2, do not contain any contradictions.*

Proof. The transition rules cannot be applied to the same state, as each rule has a different activation condition. Also the rules wont lead to a gap in their domain, given that the state space is generated using a deterministic algorithm, described in the proof to proposition 3.2.2.1. □

Generating the transition matrix Q from a set of transition rules and state space with a deterministic structure, lead to the matrix having certain set of properties.

Theorem 3.2.2.5. *That the transition matrix Q is invariant for a given state space as defined in prop 3.2.2.1*

Proof. If $\mathcal{K}$ is fixed then the state space $|\mathcal{S}^{GK}|$ is invariant which leads to the dimensions of Q remaining constant, as $|\mathcal{S}^{GK}| \times |\mathcal{S}^{GK}|$. Each index (i, j) in Q remains either zero or non-zero for a given state space. This is dependant on if there is a transition between states. A state will transition if a queue in the state meets the conditions of either being a source or sink queue, or there is connection within $p_{i,j}$ that allows for a transmission of a packet to another queue. The initial and concluding state of any transition will remain invariant because the position of all the states in the state space is determined by the deterministic algorithm that generates the state space, which is described in proposition 3.2.2.1. So any transition between states, regardless of the values of the exogenous variables, will have the same (i, j) so long as the state space isn't altered, by changing a queue's K_i. This doesn't mean that the rate of a transition can't also be zero. □

Theorem 3.2.2.6. *For a given state space, as defined in prop 3.2.2.1, and set of arrival rates γ and processing rates μ there is a unique transition matrix Q*

Proof. As stated in proof 3.2.2.5 for specified set $\mathcal{K}$ and a $p_{i,j}$ there is a specific layout of a Q where the values of the rate for each transition have continuous values. The layout, and thus the solution, becomes unique for a given set of values of λ & $\mu^{]\{\{}$ across all the queues in the network, because each transition rate can then be calculated. □

To decrease the computational complexity of generating Q grouped forms of the transition rules were used, instead of searching the state space to find the concluding state of a transition.

Theorem 3.2.2.7. *That grouping form of the transition rules will generate the correct transition matrix Q for a given state space and exogenous variables as defined in theorem 3.2.2.6.*

Proof. The grouped form of the transition rules are based upon the deterministic structure of $\mathcal{S}$ generated using the algorithm described in proof 3.2.2.1. The grouped rules are used to calculate the indexes of a set of concluding states that all share a set of initial states when one transition operation is applied to them. As stated in the proof 3.2.2.1 the number of times a specific state in a queue is repeated is given by DF_i and the number of times a whole block of states is repeated is given by SR_i. The starting index of any $\mathcal{N}_i^{\prime}$ block in the state space can be found using

$$SI_i^{BI} = 1 + (BI - 1)DF_i(K_i + 1)); \tag{3.13}$$

where BI is the number of $\mathcal{N}_i^{\prime}$ repetitions. Each transition operation is applied to these BI blocks (In the case of the $T_{i,j}$ transition it is the BI block of the transmitting queue). The start of any specific state repetition can be found using

$$SS = SI_i^{BI} + DF_i n_i \tag{3.14}$$

where n_i is the state index. The grouping form of the transition rules are used for calculating which group of specific n_i indexes pair with each other. For the I_i & O_i transitions a block consists of every state of a certain value. The block is paired off with a the block with either the subsequent or proceeding state. To find the start position of the concluding block of an outward state transition from the starting block the following calculations are used,

$$I_i^{conc} = I_i^{start} + DF_i \tag{3.15}$$

&

$$O_i^{conc} = O_i^{start} - DF_i \tag{3.16}$$

For a $T_{i,j}$ state transition the block of states involved in a particular starts off by containing all the states on the lower index queue, regardless of if it is the transmitting or receiving queue, in the set $\mathcal{ID}$. Then subsets of indexes that match up with the states in the higher index queue that can't be involved with the transmission are removed from the set, as there will be several $BI^{higher\ index}$ blocks for one of the lower index's. This will be the states 0^{th} states if higher index queue is transmitting or the K^{th} states if it is receiving. The number of subsets that are removed can computed using the ratio

$$\frac{DF_{LI}(K_{LI}+1)}{DF_{HI}(K_{HI}+1)} \tag{3.17}$$

The calculation between the starting index the initial and concluding blocks is

$$T_{i,j}^{conc} = T_{i,j}^{start} + DF_j - DF_i \tag{3.18}$$

as it is as an I_j and O_i operation are done one after the other.
The grouped form of the rules cover all reachable states in a specified state space as they have the same conditions of use as the rules in definition 3.2.2.2. For I_i & O_i it is simply when they used, whereas with $T_{i,j}$ the rules are used to generate the initial set grouped states. □

The complexity for the generation of Q is dependent on the operation theorem 3.2.2.7 being applied to each queue. So the complexity for each queue is $O((K_i + 1)SR_i)$.

Theorem 3.2.2.8. *That the generation of the transition matrix Q terminates.*

Proof. As described in this state space configuration's complexity argument the generation of Q is completed using three loops. The nested loops loop over the set of queues in the network, $K_i + 1$ for each queue, and the SR_i factor described in proof 3.2.2.1. These monotonic variables are all calculated before the loops are initiated, and are not altered within the loop. Therefore the loops generating Q will terminate. □

The method of the calculation of the marginal probabilities is given in section 3.1.4.

Corollary 3.2.2.9. *The logic and proof for calculating the marginal probabilities is the same as proof of theorem 3.2.2.7.*

3.2.3 Proofs for the Internal Order & Global Internal Order State Spaces

This section lays out the theorems used in the generation of and use of the state spaces that show the ordering of packets in a single queue and every queue in the network. The structure of the proofs are similar to those described in section 3.2.2. The axioms of the state space for the ordering in a single queue are:

Definition 3.2.3.1. The states space $\mathcal{S}^{IO}$ for internal order is defined as

$$\mathcal{S}^{IO} = \{n_1, ..., n_K | n_i \in \tau_{queue}\} \tag{3.19}$$

where τ_{queue} is the set of packet types passing through the individual queue under inspection as well as $\emptyset$ for empty spaces in the queue. An additional rule of the state space is that there can't be an empty space between two packets.

There are two types of state transition for this state space. Transmission into the queue, I_{queue}, and transmissions out of the queue, O_{queue}.

Definition 3.2.3.2. The transition rules of the internal order state space are:

Label	State Transitions	Rate	Conditions
I_{queue}	$(..., n_i, \emptyset, ...) \rightarrow (...n_i, n_{i+1}, ...)$	$-\lambda_{queue}^{\tau}$	$n_K = \emptyset$
O_{queue}	$(..., n_a ..., n_i) \rightarrow (n_{a+1}, ..., n_i, \emptyset)$	$-\mu_{queue}^{eff\ \tau}$	$n_i \neq \emptyset \mid a \in [1, C_{queue}]$

where are all the rates and conditions are declared from the perspective of transitioning out of a state.

The first step generate the state space of all possible orderings of packets for an individual queue.

Proposition 3.2.3.1. *The algorithm that creates the state space of all possibles orderings in queue, only generates states that are feasible and reachable by the transition rules given in definition 3.2.3.2 for a queue with a truncation limits K_{queue}.*

Proof. If the queue receives no packets, the only feasible state in the state space is all positions in the queue are empty. The feasible state space of non-empty queues are built using a recursive process. A queue with $K_{queue} = 1$ is a column vector of the packet types passing through it and the $\emptyset$. Queues' $K_{queue} > 1$ are a union of the $K_{queue} = 1$ column vector, with each state extended with $K_{queue} - 1$ $\emptyset$, and $K_{queue} = 1$, with each state duplicated, combined with with the non-empty state space of a $K_{queue} - 1$ queue. So the complete feasible state space that includes every packet ordering of a queue is the union of the empty queue set, which is just 0's in all positions of the queue, and the recursive structure of non-empty packet orderings.

For the state space $\mathcal{S}^{IO}$ to be generated as described two operations must be performed. The first is a recursive set generation operation, followed by an repetition operations. The recursive set operation happens for each position in the queue. The first state is the $\emptyset$ state where all the positions empty. The generation of the non-empty states begins with the first position in the queue. The set of packet types that pass through the queue, τ^+, has its members duplicated $|\tau^+|^{a-1}$ times, where $a \in [1, K_{queue}]$. The first position consists of concatenation of these duplicated sets. For each subsequent position in the queue the τ^+ member duplication operation is interwoven with a recursion operation. After the members of set are duplicated, it is repeated $|\tau^+|^{pos-1}$, where *pos* is the column position in the queue, before being concatenated. The duplication operation happens one less times than the previous position in the queue because the new position are starting concatenation point $\Sigma^{pos}|\tau^+|^{pos-1}$ to include the generation of the empty queue spaces.
These two process, as well as the addition of the empty queue state, gives you every packet order state of a queue described in definition 3.2.3.1.

In the generation of the queue's internal packet ordering every feasible state is equal to every reachable state as the state space only contains the packet ordering of an individual queue. □

Corollary 3.2.3.2. *The number of states in the state space is,*

$$|\mathcal{S}^{IO}| = \begin{cases} K_i + 1, & |\tau_i| = 1 \\ \frac{|\tau_{queue}|^{Ki+1} - 1}{|\tau_i| - 1}, & |\tau_i| > 1 \end{cases} \tag{3.20}$$

For the transition matrix Q to be correct the transition rules it is generated from must be sound. Below are the proofs for the transition rules stated in definition 3.2.3.2.

Proposition 3.2.3.3. *The transition rules of the state space, as given in definition 3.2.3.2, will not lead to any state collision.*

Proof. Let $A = \{I_i, O_i\}$ be the set of transition rule. If the two transition operations were applied to the same state, the concluding state for each operation would be different because the modification that each transition rule applies to a state is different. □

Proposition 3.2.3.4. *The transition rules of the state space, as given in definition 3.2.3.2, do not contain any contradictions.*

Proof. The transition rules cannot be applied to the same state, as each rule has a different activation condition. Also the rules wont lead to a gap in their domain, given that the state space is generated in a deterministic way leading. □

Generating the transition matrix Q from a set of transition rules and state space with a deterministic structure, lead to the matrix having certain set of properties.

Theorem 3.2.3.5. *That the transition matrix Q is invariant for a given state space as defined in prop 3.2.3.1.*

Proof. A given state space $|\mathcal{S}^{IO}|$ with a fixed $\mathcal{K}$ and τ_i will have an invariant dimensions $|\mathcal{S}^{IO}| \times |\mathcal{S}^{IO}|$. Each index (i,j) in Q remains either zero or non-zero for a given state space, which is dependant on if there is a transition between states. A state will transition if a position in a queue meets the conditions for receiving a packet or processing a packet. The initial and concluding state of any transition will remain invariant because the position of all the states in the state space is determined by the deterministic algorithm that generates the state space, which is described in proposition 3.2.3.1. So any transition between states, regardless of the values of the exogenous variables, will have the same (i,j) so long as the state space isn't altered, by changing a queue's K_i or τ_i. This doesn't mean that the rate of a transition can't also be zero. □

Theorem 3.2.3.6. *For a given state space, as defined in prop 3.2.3.1, and the queue's arrival λ_i and processing μ_i^{eff} rates there is a unique transition matrix Q*

Proof. As stated in proof 3.2.3.5 for specified K_i and a set τ_i there is a specific layout of a Q where the values of the rate for each transition have continuous values. The layout, and thus the solution, becomes unique for the set of λ_i^{τ} & $\mu_i^{eff\ \tau}$ of the packet types that pass through a queue, because each transition rate can then be calculated. □

To decrease the computational complexity of generating Q grouped forms of the transition rules were used, instead of searching the state space to find the concluding state of a transition.

Theorem 3.2.3.7. *That grouping form of the transition rules will generate the correct transition matrix Q for a given state space and exogenous variables as defined in theorem 3.2.3.6.*

Proof. The grouped form of the transition rules are based upon the deterministic structure of $\mathcal{S}^{IO}$ generated using the algorithm described in proof 3.2.3.1. The grouped rules are used to calculate the indexes of the concluding states of a specific transition that is applied to set of initial states. Each queue position is divided into sets of repeated τ^+ whose members are repeated $|\tau^+|^{a-1}$ times, as described in the proof of proposition 3.2.3.1. The starting index of each of these blocks is given by,

$$SI_a = SI_a + |\tau^+|^a \tag{3.21}$$

where $SI_0 = 1$.
The I_{queue} transition grouping is based on the repeated τ^+ sets. Bar from the last repeated block of the first position, every other state in every other repeated block can receive packet of any type that passes through the queue. So to find the block of indices at the end of a I_{queue} transitions the position of the packet type in the unaltered set τ^+ is required. With its position in the initial τ^+ block the starting index of the concluding block with

$$I_{queue}^{conc} = I_{queue}^{init} + (|\tau^+| - t)|\tau^+|^{a-1} + t|\tau^+|^a \tag{3.22}$$

where t is the position of the packet type in the initial τ^+ set, and a is the duplication number of the initial group. I_{queue}^{init} is given by,

$$I_{queue}^{init} = SI_a + (t-1)|\tau^+|^{a-1}. \tag{3.23}$$

With this position a block of states with all the members of τ^+ are concatenated to this state. O_{queue} transitions happen for all states in the queue except for the empty queue state. Each queue has a stated number of processors C_{queue} positions that can be the starting point of an O_{queue} transition. The way to calculate the initial and concluding groups is done a single repeated block at a time. The calculation for the starting index of concluding block is given by,

$$O_{queue}^{conc} = O_{queue}^{init} - |\tau^+|^{a-2} \tag{3.24}$$

For each state in the initial group they transmit to a different position in the concluding state for each C_{queue} position. The indices of concluding group are divided by $|\tau^+|^{C_{layer}-1}$ and then repeated $|\tau^+|$ times. □

The complexity for this state space configuration is $O(\tau_i C_i)$

Theorem 3.2.3.8. *That the generation of the transition matrix Q terminates*

Proof. As described in this state space configuration's complexity argument the generation of Q is completed using two loops. The nested loops loop over each packet type that passes through the queue and the number of processors C that the queue has. These are monotonic variables are all calculated before the loops are initiated, and are not altered within the loop. Therefore the loops generating Q will terminate. □

The method of the calculation of the marginal probability is given in section 3.1.4.

Corollary 3.2.3.9. *The logic and proof for calculating the marginal probability is the same as the recursive structure described in the proof of proposition 3.2.3.1.*

The proofs and definitions packet ordering state space for a single queues can be built upon to define the state space that looks at packet ordering all the queues in the network. The axioms of this state space are:

Definition 3.2.3.3. The states space $\mathcal{S}^{GIO}$ for the global internal order is defined as

$$\mathcal{S}^{GIO} = \{n_{1,1}, ..., n_{1,K1}, ..., n_{N,1}, ..., n_{N,KN} | n_{i,j} \in [\emptyset, N_\tau]\} \tag{3.25}$$

where $n_{i,j}$ can only be the packet type that pass through the specific queue, $n_{i,j} \in \tau_i$.An additional rule of the state space is that there can't be an empty space between any two packets in a specific queue

Definition 3.2.3.4. The transition rules of the global internal order state space are:

Label	State Transitions	Rate	Conditions
I_i	$(..., n_{i,a}, \emptyset, ...) \rightarrow (..., n_{i,a}, n_{i,a+1}, ...)$	$-\lambda_i^\tau$	$n_{i,a} = \emptyset$
O_i	$(..., n_{i,a} ..., n_{i,b}, ...) \rightarrow (..., n_{i,a+1} ..., n_{i,b}, \emptyset, ...)$	$-\mu_i^{eff\ \tau}$	$n_i \neq \emptyset \mid a \in [1, C_i]$
$T_{i,j}$	$(..., n_{i,a} ..., n_{i,b}, ..., n_{j,a}, \emptyset, ...)$ $\rightarrow (..., n_{i,a+1} ..., n_{i,b}, \emptyset, ..., n_{j,a}, n_{j,a+1}, ...)$	$-\mu_i^{eff\ \tau}$	$n_i \neq \emptyset \mid a \in [1, C_{queue}]$

where are all the rates and conditions are declared from the perspective of flowing out of a state.

The algorithm that generates the global internal order state spaces incorporates the algorithm required to generate an individual queue's ordering.

Proposition 3.2.3.10. *The algorithm that creates the state space only generates states that are feasible and reachable by the transition rules given in definition 3.2.3.4 for a given set of truncation limits $\mathcal{K}$, and the set of $p_{i,j}^\tau \forall \tau$.*

Proof. The generation of the global internal order space is based on using the algorithm described in the proof of proposition 3.2.3.1 to generate the state space for each individual queue. Each of the individual queue's state space is then duplicated using the two duplication operations described in the proof of proposition 3.2.2.1.
In the generation of the queue's internal packet ordering every feasible state is equal to every reachable state as the state space only contains the packet ordering of an individual queue. The state space contains every reachable state as the queuing network can't be a disconnected graph, as stated in definition 3.2.1.1. This means it is possible for packets to pass through every queue in the network given the $T_{i,j,k,l}$ transition rule given in definition 3.2.3.4. Packets will only pass through the network if a source queue is present which can utilise the rule $I_{i,j}$. If there isn't a source queue, or there are queues upstream from a source, then their individual state spaces $\mathcal{N}_i = \{0\}$ as defined at the beginning of proof 3.2.3.1. □

Corollary 3.2.3.11. *The number of states in the global internal order state space is,*

$$|\mathcal{S}^{GIO}| = \Pi_{i=1}^{N} |\mathcal{S}_i^{IO}| \tag{3.26}$$

as the total number of reachable and feasible states is the permutation of every internal order state in a queue permuted with every state of every other queue, as described in the construction of the state space algorithm in proof 3.2.3.10.

Proposition 3.2.3.12. *The transition rules of the state space, as given in definition 3.2.3.4, will not lead to any state collision.*

Proof. Let $A = \{I_{i,j}, O_{i,j}, T_{i,j,k,l}\}$ be the set of transition rule. If all three transition operations were applied to the same state, the concluding state for each operation would be different the modification that each transition rule applies to a state is different. □

Proposition 3.2.3.13. *The transition rules of the state space, as given in definition 3.2.3.4, do not contain any contradictions.*

Proof. The transition rules cannot be applied to the same state, as each rule has a different activation condition. Also the rules wont lead to a gap in their domain, given that the state space is generated in a deterministic way leading. □

The generation of transition matrix Q using the set of transition rules and utilising the deterministic state space structure described in 3.2.3.10 give the matrix a specific set of properties.

Theorem 3.2.3.14. *That the transition matrix Q is invariant for a given state space as defined in prop 3.2.3.10.*

Proof. A given state space $|\mathcal{S}^{GIO}|$ with a fixed $\mathcal{K}$ & the set of $p_{i,j}^{\tau}$ will have an invariant dimensions $|\mathcal{S}^{GIO}| \times |\mathcal{S}^{GIO}|$. Each index (i, j) in Q remains either zero or non-zero for a given state space, which is dependant on if there is a transition between states. A state will transition if a position in a queue within the network meets the conditions for receiving a packet or processing a packet from outside of the network, or transferring a packet between queues. The initial and concluding state of any transition will remain invariant because the position of all the states in the state space is determined by the deterministic algorithm that generates the state space, which is described in proposition 3.2.3.10. So any transition between states, regardless of the values of the exogenous variables, will have the same (i, j) so long as the state space isn't altered, by changing a queue's K_i. This doesn't mean that the rate of a transition can't also be zero. □

Theorem 3.2.3.15. *For a given state space, as defined in prop 3.2.3.10, and set of arrival rates λ and effective processing rates μ^{eff} there is a unique transition matrix Q*

Proof. As stated in proof 3.2.3.14 for specified set $\mathcal{K}$ and a $p_{i,j}^{\tau}$ there is a specific layout of a Q where the values of the rate for each transition have continuous values. The layout, and thus the solution, becomes unique for a given set of values of λ & $\mu^{]\{\{}$ across all the queues in the network, because each transition rate can then be calculated. □

To decrease the computational complexity of generating Q grouped forms of the transition rules have been used, instead of resorting to a search of the state space to find the concluding state.

Theorem 3.2.3.16. *That grouping form of the transition rules will generate the correct transition matrix Q for a given state space and exogenous variables as defined in theorem 3.2.3.15.*

Proof. The grouping rules are merger of those described in proof 3.2.3.7 that are then duplicated by the block jumping operation described in proof 3.2.2.7. □

The complexity for each queue's state transition conclusion is $O(\tau_i C_i S R_i)$, so this calculation would need to be done for each queue.

Theorem 3.2.3.17. *That the generation of the transition matrix Q terminates*

Proof. As described in this state space configuration's complexity argument the generation of Q is completed using four loops. The nested loops loop over each packet type that passes through each queue that, the number of processors that the queue has C,and the SR_i factor described in proof 3.2.3.10. These are monotonic variables are all calculated before the loops are initiated, and are not altered within the loop. Therefore the loops generating Q will terminate. □

The method of the calculation of the marginal probability is given in section 3.1.4.

Corollary 3.2.3.18. *The logic and proof for calculating the marginal probability is the same as the recursive structure described in the proof of proposition 3.2.3.10.*

3.2.4 Proofs for the Internal State & Global Internal State State Spaces

This section goes over the proofs for the generation of the state spaces of internal state of an individual queue and internal state of every queue in the network. The author was unable to devise a complete set of proofs for these state spaces, so this section will go over the completed proofs before laying out proof strategies for the missing proofs. It will also lay out complexity arguments for the current implementation of these proofs. The axioms of the internal state state space are:

Definition 3.2.4.1. The states space $\mathcal{S}^{IS}$ for internal state is defined as

$$\mathcal{S}^{IS} = \{a_1, b_1, w_1, ..., a_\mathcal{T}, b_\mathcal{T}, w_\mathcal{T} | a_i, b_i, w_i \in [0, K_{queue}]\} \tag{3.27}$$

where the number of a_i, b_i, w_i represent the number of packets being processed, blocked, or waiting in a queue, and i is the index of a packet type that passes through queue.

Definition 3.2.4.2. The transition rules of the internal state state space are:

Label	state transition	Rate	Conditions
SP_i	$(..., a_i, b_i, w_i, ...) \rightarrow$ $(..., a_i + 1, b_i, w_i, ...)$	$-\lambda_{queue,i}$	$a_i + b_i + w_i = 0$ $\& \ a + b < C_{queue}$
W_i	$(..., a_i, b_i, w_i, ...) \rightarrow$ $(..., a_i, b_i, w_i + 1, ...)$	$-\lambda_{queue,i}$	$w_i < K_{queue} - C_{queue}$ $a + b = C_{queue}$
PC_i^1	$(..., a_i, b_i, w_i, ...) \rightarrow$ $(..., a_i - 1, b_i, w_i, ...)$	$-a_i\mu_{queue,i}(1 - P_{queue})$	$a_i \geq 1$ $\& \ w_i = 0$
PC_i^2	$(..., a_i, b_i, w_i, ...) \rightarrow$ $(..., a_i - 1, b_i, w_i, ..., a_T + 1, b_T, w_T - 1, ...)$ $T = \{i \vert w_i \neq 0\}$	$-a_i\mu_{queue,i}(1 - P_{queue})$	$a_i \geq 1$ $\& \ w_i = 0$
NP_i^1	$(..., a_i, b_i, w_i, ...) \rightarrow$ $(..., a_i, b_i, w_i - 1, ...)$	$-a_i\mu_{queue,i}(1 - P_{queue})$	$a_i \geq 1$ $w_i \geq 1$
NP_i^2	$(..., a_i, b_i, w_i, ...) \rightarrow$ $(..., a_i - 1, b_i, w_i, ..., a_T + 1, b_T, w_T - 1, ...)$ $T = \{i \vert w_i \neq 0\}$	$-a_i\mu_{queue,i}(1 - P_{queue})$	$a_i \geq 1$ $w_i \geq 1$
PB_i	$(..., a_i, b_i, w_i, ...) \rightarrow$ $(..., a_i - 1, b_i + 1, w_i, ...)$	$-a_i\mu_{queue,i}P_{queue}$	$a_i \geq 1$
BC_i^1	$(..., a_i, b_i, w_i, ...) \rightarrow$ $(..., a_i, b_i - 1, w_i, ...)$	$-\mu_i^a$	$b_i \geq 1$ $w_i = 0$
BC_i^2	$(..., a_i, b_i, w_i, ...) \rightarrow$ $(..., a_i, b_i - 1, w_i, ..., a_T + 1, b_T, w_T - 1, ...)$ $T = \{i \vert w_i \neq 0\}$	$-\mu_i^a$	$b_i \geq 1$ $w_i = 0$
BP_i^1	$(..., a_i, b_i, w_i, ...) \rightarrow$ $(..., a_i + 1, b_i - 1, w_i - 1, ...)$	$-\mu_i^a$	$b_i \geq 1$ $w_i \geq 1$
BP_i^2	$(..., a_i, b_i, w_i, ...) \rightarrow$ $(..., a_i, b_i - 1, w_i, ..., a_T + 1, b_T, w_T - 1, ...)$ $T = \{i \vert w_i \neq 0\}$	$-\mu_i^a$	$b_i \geq 1$ $w_i \geq 1$

where the rates and conditions are declared from the perspective of flowing out of a state.

Proposition 3.2.4.1. *The number of feasible and reachable states in the state space is,*

$$|\mathcal{S}^{IS}| = \sum_{n=0}^{C_i-1} \binom{n+2|\mathcal{T}_i|-1}{2|\mathcal{T}_i|-1} + \binom{C_i+2|\mathcal{T}_i|-1}{2|\mathcal{T}_i|-1} \left[\sum_{n=0}^{K_i-C_i} \binom{n+|\mathcal{T}_i|-1}{|\mathcal{T}_i|-1} \right] \tag{3.28}$$

Proof. The number of reachable and feasible states of the internal state state space maps onto the stars and bars problem[63], which is used to calculate the number of possible ways that n objects can be distribute across k bins. The solution of the stars and bars problem is given by

$$P = \binom{n+k-1}{k-1}. \tag{3.29}$$

The number of feasible states can be split into two stars and bars problems. The first problem is how the packets are distributed across the processors, C_i, of a queue, and the second is if the queue's processors are full what are the permutations of ways packets can be waiting to be processed. Each of these problems maps to the terms in equation 3.28.
Each processor can either be processing a packet or blocked, so for every packet type that passes through a queue there are two bins to represent a & b for each type. This means k in equation 3.29 becomes $2|\mathcal{T}_i|$. n packets are distributed across the bins, so the n in equation 3.29 remains unchanged. To include the number of packets waiting to be processed a third bin needs to be added for each packet type. For all possible values of $c < C_i$ there no packets queuing, as packets go straight to a processor, so the waiting bins remain empty. The number of feasible states for $c < C_i$ is given by

$$\sum_{n=0}^{C_i-1} \binom{n+2|\mathcal{T}_i|-1}{2|\mathcal{T}_i|-1} \tag{3.30}$$

When $c = C_i$ the packets begin queuing. This means that the third bins of each packet type start filling. The number of packets can be calculated using equation 3.30 and setting the the starting and terminating conditions of the sum to C_i. How many ways packets of different types waiting can be permuted by the same process as calculated the number of permutations of packets being processed and blocked. For the waiting permutations the number of bins is given by $|\mathcal{T}_i|$. The number of permutations of packets waiting is given by

$$\sum_{n=0}^{K_i-C_i} \binom{n+|\mathcal{T}_i|-1}{|\mathcal{T}_i|-1} \tag{3.31}$$

So the total number of full processing states with all the potential waiting states is the sum of equation 3.31 and equation 3.30 when it is set up to only calculate $c = C_i$. This gives the second term of equation 3.28.
In the generation of the queue's internal packet ordering every feasible state is equal to every reachable state as the state space only contains the packet ordering of an individual queue. □

Below is the proof that the current algorithm that creates the state space only generates a complete state space. However the current algorithm doesn't create a grouped state space that allows for the development of combinatorial rules that allow for the calculation of a set of concluding transitions indexes.

Proposition 3.2.4.2. *The algorithm that creates the state space only generates states that are feasible and reachable by the transition rules given in definition 3.2.4.2 for a queue's truncation limits K_i and number of processors C_i.*

Proof. The algorithm that generates the state space follows a similar procedure as described in the proof of proposition 3.2.4. The algorithm to create all the permutations of a stars and bars problem proceeds by moving a value across the bin space. It starts with the all n objects in the initial bin, $i = 1$. Each subsequent state moves one value into the bin $i = 2$ until all n are in the $i = 2$ bin. The next state moves one value into the $i = 3$ bin and the remaining

$n-1$ objects in the $i=1$ bin. This shifting operating is repeated across all the bins until all n objects are in the $i=k$.
For each $c < C_i$ the above stars and bars algorithm is applied to generate their c-block space. Once all the blocks are generated a column of zeros is inserted after every two columns of the state space, to represent the empty waiting states. Then a c-block for $c = C_i$ generated. The k-space is generated in the same way as the $c < C_i$. Once the k-space is generated each state in the $c = C_i$ block is duplicated to the number of states in the k-space. Each duplicated state of $c = C_i$ has a column of the k-space inserted after every two of the $c = C_i$ columns. □

An algorithm designed to produce grouped states so to ease the calculation transitions indexes, as in the previous sections, two elements of the proof of proposition 3.2.4.2 would have to be changed. The first would the stars and bars generation would have to become generate all the states in the range of $[0, n]$. This could be done with a similar shifting operation that has already been described, but it the initial state ascends from 0 before moving an object to the next column. The second modification would be the weaving together of the c-blocks and k-blocks. A method for doing this is still being researched.

Proposition 3.2.4.3. *The transition rules of the state space, as given in definition 3.2.4.2, will not lead to any state collision.*

Proof. Let $A = \{SP_i, W_i, PC_i^1, PC_i^2, NP_i^1, NP_i^2, PB_i, BC_i^1, BC_i^2, BP_i^1, BP_i^2\}$ be the set of transition rule. If all eleven transition operations were applied to the same state, the concluding state for each operation would be different as the transition operations for each member of set A is different. □

Proposition 3.2.4.4. *The transition rules of the state space, as given in definition 3.2.4.2, do not contain any contradictions.*

Proof. The transition rules cannot be applied to the same state, as each rule has a different activation condition. Also the rules wont lead to a gap in their domain, given that the state space is generated in a deterministic way leading. □

Theorem 3.2.4.5. *That the transition matrix Q is invariant for a given state space as defined in prop 3.2.3.1.*

Proof. A given state space $|\mathcal{S}^{IS}|$ with a fixed K_i, C_i and τ_i will have an invariant dimensions $|\mathcal{S}^{IS}| \times |\mathcal{S}^{IS}|$. Each index (i, j) in Q remains either zero or non-zero for a given state space, which is dependant on if there is a transition between states. A state will transition if a position in a queue meets the conditions in the rules defined in axiom 3.2.4.2. The initial and concluding state of any transition will remain invariant because the position of all the states in the state space is determined by the deterministic algorithm that generates the state space, which is described in proposition 3.2.4.2. So any transition between states, regardless of the values of the exogenous variables, will have the same (i, j) so long as the state space isn't altered, by changing a queue's K_i, C_i, or τ_i. This doesn't mean that the rate of a transition can't also be zero. □

Theorem 3.2.4.6. *For a given state space, as defined in prop 3.2.4.2, and the queue's arrival λ_i and processing rate μ_i^{eff} rates there is a unique transition matrix Q*

Proof. As stated in proof 3.2.4.5 for specified K_i, C_i and a set τ_i there is a specific layout of a Q where the values of the rate for each transition have continuous values. The layout, and thus the solution, becomes unique for the set of λ_i^τ & $\mu_i^{eff\ \tau}$ of the packet types that pass through a queue, because each transition rate can then be calculated. □

Without the derivation of rules of the structure of state space the generation of Q is completed by searching the state space rather than navigating it. Also the current method for calculating the marginal probability for this state space configuration is done by a search. The complexity of this configuration is currently $O(|\mathcal{S}^{GIS}|^2)$, as the state space isn't structured into a grouped form that can easily be navigated across. Each state's transitions leads to a search of the state space until the concluding state is found.

Theorem 3.2.4.7. *That the generation of the transition matrix Q terminates*

Proof. As described in this state space configuration's complexity argument the generation of Q is completed using two loops. The nested loops loop over each state and then loops over them again whilst searching for the concluding state. As the number of states is calculated before the loops are initiated in the proof 3.2.4, and isn't altered within the loop, then the loops generating Q will terminate. □

Having defined the internal state space for an individual queue, the internal state space of all the queues in the network can be derived. The axioms for this state space are:

Definition 3.2.4.3. The states space $\mathcal{S}^{GIS}$ for global internal state state space is defined as

$$\mathcal{S}^{GIS} = \{a_{1,1}, b_{1,1}, w_{1,1}, ..., a_{1,\tau}, b_{1,\tau}, w_{1,\tau}, ..., a_{N,1}, b_{N,1}, w_{N,1}, ..., a_{N,\tau}, b_{N,\tau}, w_{N,\tau} | a_i, b_i, w_i \in [0, K_{queue}]\} \tag{3.32}$$

where the number of $a_{i,\tau}, b_{i,\tau}, w_{i,\tau}$ represent the number of packets being processed, blocked, or waiting in queue i of type τ. Note that the set of τ's for each queue only contain the packet types that pass through the queue.

Definition 3.2.4.4. The transition rules of the global internal state state space are:

Label	state transition	Rate	Conditions
$PB_{i,\tau}$,	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ...) \rightarrow$ $(..., a_{i,\tau} - 1, b_{i,\tau} + 1, w_{i,\tau}, ...)$	$-a_{i,\tau}\mu_{i,\tau}P_i$	$a_{i,\tau} \geq 1$
$ESP_{i,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ...) \rightarrow$ $(..., a_{i,\tau} + 1, b_{i,\tau}, w_{i,\tau}, ...)$	$-\lambda_{i,\tau}$	$a_{i,\tau} + b_{i,\tau}$ $+w_{i,\tau} = 0$ $a_i + b_i < C$
$EW_{i,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ...) \rightarrow$ $(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau} + 1, ...)$	$-\lambda_{i,\tau}$	$a_i + b_i < C$ $w_i < K_i - C_i$
$EPC^1_{i,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ...) \rightarrow$ $(..., a_{i,\tau} - 1, b_{i,\tau}, w_{i,\tau}, ...)$	$-a_\tau\mu_{i,\tau}(1 - P_i)$	$a_{i,\tau} \geq 1$ $w_{i,\tau} = 0$ $w_i = 0$
$EPC^2_{i,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ...) \rightarrow$ $(..., a_{i,\tau} - 1, b_{i,\tau}, w_{i,\tau}, ...$ $a_{i,T} + 1, b_{i,T}, w_{i,T} - 1, ...)$ $T = \{\tau \mid w_{i,\tau} \neq 0\}$	$-a_\tau\mu_{i,\tau}(1 - P_i)$	$a_{i,\tau} \geq 1$ $w_{i,\tau} = 0$ $w_i > 0$
$EBC^1_{i,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ...) \rightarrow$ $(..., a_{i,\tau}, b_{i,\tau} - 1, w_{i,\tau}, ...)$	$-\mu^a_{i,\tau}$	$b_{i,\tau} \geq 1$ $w_{i,\tau} = 0$ $w_i = 0$
$EBC^2_{i,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ...) \rightarrow$ $(..., a_{i,\tau}, b_{i,\tau} - 1, w_{i,\tau}, ...$ $a_{i,T} + 1, b_{i,T}, w_{i,T} - 1, ...)$ $T = \{\tau \mid w_{i,\tau} \neq 0\}$	$-\mu^a_{i,\tau}$	$b_{i,\tau} \geq 1$ $w_{i,\tau} = 0$ $w_i > 0$
$ENP^1_{i,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ...) \rightarrow$ $(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau} - 1, ...)$	$-a_\tau\mu_{i,\tau}(1 - P_i)$	$a_{i,\tau} \geq 1$ $w_{i,\tau} > 0$
$ENP^2_{i,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{i,T}, b_{i,T}, w_{i,T}, ...) \rightarrow$ $(..., a_{i,\tau} - 1, b_{i,\tau}, w_{i,\tau}, ...a_{i,T} + 1, b_{i,T}, w_{i,T} - 1, ...)$ $T = \{\tau \mid w_{i,\tau} \neq 0\}$	$-a_\tau\mu_{i,\tau}(1 - P_i)$	$a_{i,\tau} \geq 1$ $w_{i,\tau} > 0$
$EBP^1_{i,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ...) \rightarrow$ $(..., a_{i,\tau} + 1, b_{i,\tau} - 1, w_{i,\tau} - 1, ...)$	$-\mu^a_{i,\tau}$	$b_{i,\tau} \geq 1$ $w_{i,\tau} > 0$
$EBP^2_{i,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{i,T}, b_{i,T}, w_{i,T}, ...) \rightarrow$ $(..., a_{i,\tau}, b_{i,\tau} - 1, w_{i,\tau}, ...a_{i,T} + 1, b_{i,T}, w_{i,T} - 1, ...)$ $T = \{\tau \mid w_{i,\tau} \neq 0\}$	$-\mu^a_{i,\tau}$	$b_{i,\tau} \geq 1$ $w_{i,\tau} > 0$

$PC^1_{i,\tau} - SP_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau} - 1, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau} + 1, b_{j,\tau}, w_{j,\tau}, ...)$	$-a_\tau \mu_{i,\tau}(1 - P_i)$	$a_{i,\tau} \geq 1$ $w_{i,\tau} = 0$ $w_i = 0$ $a_{j,\tau} + b_{j,\tau}0$ $+w_{j,\tau} = 0$ $a_j + b_j < C$
$PC^2_{i,\tau} - SP_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau} - 1, b_{i,\tau}, w_{i,\tau}, ..., a_{i,T} + 1, b_{i,T}, w_{i,T} - 1$ $..., a_{j,\tau} + 1, b_{j,\tau}, w_{j,\tau}, ...)$ $T = \{\tau \| w_{i,\tau} \neq 0\}$	$-a_\tau \mu_{i,\tau}(1 - P_i)$	$a_{i,\tau} \geq 1$ $w_{i,\tau} = 0$ $w_i > 0$ $a_{j,\tau} + b_{j,\tau}$ $+w_{j,\tau} = 0$ $a_j + b_j < C$
$BC^1_{i,\tau} - SP_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau}, b_{i,\tau} - 1, w_{i,\tau}, ..., a_{j,\tau} + 1, b_{j,\tau}, w_{j,\tau}, ...)$	$-\mu^a_{i,\tau}$	$b_{i,\tau} \geq 1$ $w_{i,\tau} = 0$ $w_i = 0$ $a_{j,\tau} + b_{j,\tau}$ $+w_{j,\tau} = 0$ $a_j + b_j < C$
$BC^2_{i,\tau} - SP_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau}, b_{i,\tau} - 1, w_{i,\tau}, ...a_{i,T} + 1, b_{i,T}, w_{i,T} - 1$ $..., a_{j,\tau} + 1, b_{j,\tau}, w_{j,\tau}, ...)$ $T = \{\tau \| w_{i,\tau} \neq 0\}$	$-\mu^a_{i,\tau}$	$b_{i,\tau} \geq 1$ $w_{i,\tau} = 0$ $w_i > 0$ $a_{j,\tau} + b_{j,\tau}$ $+w_{j,\tau} = 0$ $a_j + b_j < C$
$NP^1_{i,\tau} - SP_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau} - 1, ..., a_{j,\tau} + 1, b_{j,\tau}, w_{j,\tau}, ...)$	$-a_\tau \mu_{i,\tau}(1 - P_i)$	$a_{i,\tau} \geq 1$ $w_{i,\tau} > 0$ $a_j + b_j < C$ $a_{j,\tau} + b_{j,\tau}$ $+w_{j,\tau} = 0$
$NP^2_{i,\tau} - SP_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau} - 1, b_{i,\tau}, w_{i,\tau}, ..., a_{i,T} + 1, b_{i,T}, w_{i,T} - 1,$ $..., a_{j,\tau} + 1, b_{j,\tau}, w_{j,\tau}, ...)$ $T = \{\tau \| w_{i,\tau} \neq 0\}$	$-a_\tau \mu_{i,\tau}(1 - P_i)$	$a_{i,\tau} \geq 1$ $w_{i,\tau} > 0$ $a_j + b_j < C$ $a_{j,\tau} + b_{j,\tau}$ $+w_{j,\tau} = 0$
$BP^1_{i,\tau} - SP_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau} + 1, b_{i,\tau} - 1, w_{i,\tau} - 1, ..., a_{j,\tau} + 1, b_{j,\tau}, w_{j,\tau}, ...)$	$-\mu^a_{i,\tau}$	$b_{i,\tau} \geq 1$ $w_{i,\tau} > 0$ $a_j + b_j < C$ $a_{j,\tau} + b_{j,\tau}$ $+w_{j,\tau} = 0$
$BP^2_{i,\tau} - SP_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau}, b_{i,\tau} - 1, w_{i,\tau}, ..., a_{i,T} + 1, b_{i,T}, w_{i,T} - 1$ $..., a_{j,\tau} + 1, b_{j,\tau}, w_{j,\tau}, ...)$ $T = \{\tau \| w_{i,\tau} \neq 0\}$	$-\mu^a_{i,\tau}$	$b_{i,\tau} \geq 1$ $w_{i,\tau} > 0$ $a_j + b_j < C$ $a_{j,\tau} + b_{j,\tau}$ $+w_{j,\tau} = 0$
$PC^1_{i,\tau} - W_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau} - 1, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau} + 1, ...)$	$-a_\tau \mu_{i,\tau}(1 - P_i)$	$a_{i,\tau} \geq 1$ $w_{i,\tau} = 0$ $w_i = 0$ $a_j + b_j < C$ $w_j < K_j - C_j$
$PC^2_{i,\tau} - W_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau} - 1, b_{i,\tau}, w_{i,\tau}, ..., a_{i,T} + 1, b_{i,T}, w_{i,T} - 1$ $..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau} + 1, ...)$ $T = \{\tau \| w_{i,\tau} \neq 0\}$	$-a_\tau \mu_{i,\tau}(1 - P_i)$	$a_{i,\tau} \geq 1$ $w_{i,\tau} = 0$ $w_i > 0$ $a_j + b_j < C$ $w_j < K_j - C_j$

$BC^1_{i,\tau} - W_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau}, b_{i,\tau} - 1, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau} + 1, ...)$	$-\mu^a_{i,\tau}$	$b_{i,\tau} \geq 1$ $w_{i,\tau} = 0$ $w_i = 0$ $a_j + b_j < C$ $w_j < K_j - C_j$
$BC^2_{i,\tau} - W_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau}, b_{i,\tau} - 1, w_{i,\tau}, ...a_{i,T} + 1, b_{i,T}, w_{i,T} - 1$ $..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau} + 1, ...)$ $T = \{\tau \| w_{i,\tau} \neq 0\}$	$-\mu^a_{i,\tau}$	$b_{i,\tau} \geq 1$ $w_{i,\tau} = 0$ $w_i > 0$ $a_j + b_j < C$ $w_j < K_j - C_j$
$NP^1_{i,\tau} - W_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau} - 1, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau} + 1, ...)$	$-a_\tau \mu_{i,\tau}(1 - P_i)$	$a_{i,\tau} \geq 1$ $w_{i,\tau} > 0$ $a_j + b_j < C$ $w_j < K_j - C_j$
$NP^2_{i,\tau} - W_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau} - 1, b_{i,\tau}, w_{i,\tau}, ..., a_{i,T} + 1, b_{i,T}, w_{i,T} - 1,$ $..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau} + 1, ...)$ $T = \{\tau \| w_{i,\tau} \neq 0\}$	$-a_\tau \mu_{i,\tau}(1 - P_i)$	$a_{i,\tau} \geq 1$ $w_{i,\tau} > 0$ $a_j + b_j < C$ $w_j < K_j - C_j$
$BP^1_{i,\tau} - W_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau} + 1, b_{i,\tau} - 1, w_{i,\tau} - 1, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau} + 1, ...)$	$-\mu^a_{i,\tau}$	$b_{i,\tau} \geq 1$ $w_{i,\tau} > 0$ $a_j + b_j < C$ $w_j < K_j - C_j$
$BP^2_{i,\tau} - W_{j,\tau}$	$(..., a_{i,\tau}, b_{i,\tau}, w_{i,\tau}, ..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau}, ...)$ $\rightarrow$ $(..., a_{i,\tau}, b_{i,\tau} - 1, w_{i,\tau}, ..., a_{i,T} + 1, b_{i,T}, w_{i,T} - 1$ $..., a_{j,\tau}, b_{j,\tau}, w_{j,\tau} + 1, ...)$ $T = \{\tau \| w_{i,\tau} \neq 0\}$	$-\mu^a_{i,\tau}$	$b_{i,\tau} \geq 1$ $w_{i,\tau} > 0$ $a_j + b_j < C$ $w_j < K_j - C_j$

where $x_i = \Sigma_\tau x_{i,\tau}$ and all the rates and conditions are declared from the perspective of flowing out of a state.

Corollary 3.2.4.8. *The number of states in the state space is,*

$$|\mathcal{S}^{GIS}| = \Pi_{i=1}^{N} |\mathcal{S}_i^{IS}| \tag{3.33}$$

as the total number of reachable and feasible states is the permutation of every internal state of a queue with the internal state of every other queue in the network.

Proposition 3.2.4.9. *The algorithm that creates the state space only generates states that are feasible and reachable by the transition rules given in definition 3.2.4.4 for network with a given set of truncation limits $\mathcal{K}$, set of number of processors $\mathcal{C}$, and the set of $p^\tau_{i,j} \forall \tau$.*

Proof. The generation of the global internal state space is based on the algorithm described in the proof of proposition 3.2.4.2. This algorithm is used to generate the state space for each individual queue. Each of the individual queue's state space is then duplicated using the two duplication operations described in the proof of proposition 3.2.2.1.
In the generation of the queue's internal state every feasible state is equal to every reachable state as the state space only contains the packet ordering of an individual queue. The state space contains every reachable state as the queuing network can't be a disconnected graph, as stated in definition 3.2.1.1. This means it is possible for packets to pass through every queue in the network given the set of transition rules rule given in definition 3.2.4.4. Packets will only pass through the network if a source queue is present which can utilise the rules ESP or EW. If there isn't a source queue, or there are queues upstream from a source of a particular packet type, then that hypothetical a_i, b_i, w_i isn't included in the state space. □

Proposition 3.2.4.10. *The transition rules of the state space, as given in definition 3.2.3.4, will not lead to any state collision.*

Proof. Let A be the set of the 27 transition rule given in 3.2.4.4. If all twenty seven transition operations were applied to the same state, the concluding state for each operation would be different as the transition operations for each member of set A is different. □

Proposition 3.2.4.11. *The transition rules of the state space, as given in definition 3.2.4.4, do not contain any contradictions.*

Proof. The transition rules cannot be applied to the same state, as each rule has a different activation condition. Also the rules wont lead to a gap in their domain, given that the state space is generated in a deterministic way leading. □

Theorem 3.2.4.12. *That the transition matrix Q is invariant for a given state space as defined in prop 3.2.4.9.*

Proof. A given state space $|\mathcal{S}^{GIs}|$ with a fixed $\mathcal{K}$, $\mathcal{C}$, & the set of $p^{\tau}_{i,j}$ will have an invariant dimensions $|\mathcal{S}^{GIS}| \times |\mathcal{S}^{GIS}|$. Each index (i, j) in Q remains either zero or non-zero for a given state space, which is dependant on if there is a transition between states. A state will transition if a position in a queue within the network meets one the conditions for receiving a packet or processing a packet from outside of the network, or transferring a packet between queues given in definition 3.2.4.4. The initial and concluding state of any transition will remain invariant because the position of all the states in the state space is determined by the deterministic algorithm that generates the state space, which is described in proposition 3.2.4.9. So any transition between states, regardless of the values of the exogenous variables, will have the same (i, j) so long as the state space isn't altered, by changing a queue's K_i or C_i. This doesn't mean that the rate of a transition can't also be zero. □

Theorem 3.2.4.13. *For a given state space, as defined in prop 3.2.4.9, and set of arrival rates λ, processing rates μ, mean processing rates μ^a and probabilities of being full P there is a unique transition matrix Q*

Proof. As stated in proof 3.2.4.12 for specified set $\mathcal{K}$, $\mathcal{C}$, and a $p^{\tau}_{i,j}$ there is a specific layout of a Q where the values of the rate for each transition have continuous values. The layout, and thus the solution, becomes unique for a given set of values of λ, μ $\mu^{\dashv}$, & P across all the queues in the network are defined, so each transition rate can then be calculated. □

As proposition 3.2.4.2 doesn't produce a grouped structure, the state space for the global internal state space also doesn't have a grouped structure due to its dependence laid out in proposition 3.2.4.9. This means that the concluding states of any transition in the global internal state space are also searched for instead of computed. The complexity of this configuration is then $O(|\mathcal{S}^{GIS}|^2)$ as each state's transitions leads to a search of the state space until the concluding state is found.

Theorem 3.2.4.14. *That the generation of the transition matrix Q terminates*

Proof. As described in this state space configuration's complexity argument the generation of Q is completed using two loops. The nested loops loop over each state and then loops over them again whilst searching for the concluding state. As the number of states is calculated before the loops are initiated in the proof 3.2.4.8, and isn't altered within the loop, then the loops generating Q will terminate. □

The method of the calculation of the marginal is given in section 3.1.4.

3.3 Comparison to Other Formal Methods

This section compares the author's queuing network formal method with the other formal methods described in chapter 2. It will compare what each method can encapsulate, before looking at what different adversaries can be generated. The final aspect of comparison is the performance of the different methods.

3.3.1 Methodology Comparison

To model a security protocol's robustness a class of security promises that are orthogonal to those that are traditionally tested against must be encapsulated. In the case of the formal method presented in this research the author was able to encapsulate the promises of availability and synchronisation of state. The method is able to test these promises because the description of the communication channel that the honest agents are using is not abstracted away into the adversary model, which is what is traditionally done in the symbolic adversary model. This formal method allows the user to declare which parts of the a session diagram they are allowed access to, and in what ways the malicious interference can happen. This granularity manifests itself in the generation of iterations of a given queuing network with variations in the exogenous variables of specified queues in the network, such as arrival rate, processing rate, truncation limit.

Symbolic methods are traditionally formulated to encapsulate and augment the narrative of a protocol run. They achieve this by analysing a bounded set of potential traces of run(s) within the state space of all potential runs. These traces are generated from the potential paths through an honest agent's state machine. They are searching these traces to find a trace where, through illicit manipulation, the adversary is able to discern the secret being passed between the honest agents. If the search of the state space produces a trace where the adversary learns the secret message the security promises is proven to be broken. In these formal methods the notion of time is the order of messages passed between the set of agents a long a particular traces. Symbolic methods create these narrative with variables to define a message's content, the cryptographic primitives, etc, along with functions for how agents interact with the variables. From these feature a model of a protocol session can be created so set a trace can be searched for violations.

The computational adversary model finds promise violations by seeing if an adversary can compute the content of an encrypted message through deductions directed through queries of protocol instances. If the adversary can deduce the message with a greater than fifty percent chance in polynomial time it is said to have found a violation. The computational model's notion of ordering describes how the adversary accrues knowledge over time. Each query to the honest agent is another step in the ordering. In this model the honest agent is encapsulated as a reactive oracle that has no state machine of their own. The only notion of timing this approach has is that of the time complexity of the algorithm the adversary uses to decrypt the message. However, this approach does have a notion of internal state for the adversary, as their knowledge develops over time with the responses from the oracle. This gradual acquiring of knowledge allows their internal algorithm to break the honest agents cypher.

Unlike both the computational and symbolic methods the goal of the author's queuing network formal method is not to extract a message from a channel, but to see how much effort is required to disrupt a secure channel. The method's set up is similar to the symbolic method as it encapsulates the narrative of a protocol run and shares the same base assumption of perfect cryptography. However, due to the orthogonal goals to most symbolic methodology how the queuing network method is similar to the computational model. Instead of providing the user with a counter example that undermines the security promise, like the symbolic methods, the method calculates the probability that a range of exogenous variables will stop the completion of a protocol run. Instead of looking for counter example traces the queueing network method searches the problem space to across a range of queue exogenous variables. From these sets of exogenous variables the method is able to calculate the most likely state of the network, which may be in violation of the security promises. This allows for the user of a protocol to monitor the device's for when they are approaching abnormal behaviour. This reduction in strength of proof has precedent as computational method uses probabilistic arguments to demonstrate the likelihood of an adversarial Turing machine using a polynomial time algorithm to deduce an encrypted message.

The other major difference between the author's methodology and the other symbolic formal methods is the separation of the structure of the communication channel between honest agents and the adversary. In all symbolic formal methods they use some form of the DY adversary, who has complete control over the channel that a protocol session is running across. This regime of modelling provides little resolution for the minimum power an adversary needs to stop a session from being completed, because it is trivial for the DY adversary to drop every packet. To solve this the author has separated the channel from the adversary, and it is up to the user of the method to decide what the adversary has access to. In the author's implementation the adversary has access to a queue's exogenous variables. By manipulating these variables across a range of values the method can discern what the mimimum access and knowledge they need to disrupt the session, and at what value of exogenous variable the protocol session is no longer available.

Unlike the other symbolic methods the queuing network isn't designed to be able distinguish between each kind of the messages in a protocol run. All the packets passed between the states of an agent, other than those manipulated by the adversary, are presumed to be correct and expected. The method could be set up to model to encapsulate message content by having a packet type for each kind message passed in the state machine, but that would detract from modelling the performance of the agent's state machine. The encapsulation of an honest agent's knowledge is only to the level of calculating the likelihood of state machine passing a packet along a certain edge. This lower resolution of protocol run encapsulation is what allow for the modelling of the timing and size of packets to analyse how these features effect the quality of service requirements of the protocol run. The probabilistic model also allows for this method to overcome one of the common problems faced by symbolic methods. As stated in section 2.1.1 boundaries need to be set for the modelling of a protocol otherwise the decision of which trace to analyses become undecidable. These restrictions are either in the form of an algorithm or human decision. This decision problem is circumnavigated by the queuing network methodology as it is not looking at specific traces, but the overall state of an agent's state machine.

The only symbolic method that the author knows of with the ability to find attacks against availability is the security LTL developed by Corin & Saptawijaya[45]. Temporal logics are used to test the robustness of protocols, by seeing if it is possible for the devices state machine can logically reach an undesirable state, but they aren't designed to provide these guarantees in the presence of an adversary. Given that they usually don't encapsulate adversarial interactions, LTL's don't have a way of encapsulating the rate or size of messages being passed through a state machine, just that they will be passed. These are the principle advantages of the queuing networks when compared to temporal logics. An adversary can manipulate the state machine to increase the likelihood of the state machine reaching the undesirable state.

3.3.2 Adversary Comparison

Adversary modelling is another critical aspect of protocol modelling. The features of a formal method dictate what abilities and knowledge an adversary can have.

The traditional symbolic adversary has the objective of violating the confidentiality and authentication of a protocol session to extract a secret message from the honest agents. In the existing symbolic methodologies the DY adversary is omnipresence and omnipotent over the communication channel, which allows them to read any message they have the key for, edit messages, and replay them. The research into this kind of adversary, as described in section 2.1.1, aims to make the DY adversary as strong as possible. This is achieved by enhancing either the adversary's abilities or knowledge capacity. The goal of strengthening an adversary is to demonstrate how difficult it is for an adversary to extract a message from a security protocol. These enhanced DY adversaries could easily implement an attack against availability by severing the channel between honest agents, or inject a message that can cause a de-synchronisation of state. Given this it is clear that these adversaries are too powerful to define the limits of correct operation of a security protocol against disruption attacks with any useful sensitivity. This kind adversary also has some further presumptions baked in that restrict it from modelling disruption attacks. The assumption of the adversary controlling the network between the communicating parties is often reduced to the channel being modelled as as a single edge between the parties. This simplification means that the model doesn't encapsulate the time it would take for the adversary to synchronise the knowledge it gathers from different intercepts on the network. This lack of timing further reduces the nuance of any models of attacks against availability developed using this adversary as it can't encapsulate race conditions or the rate at which packets are processed or packet ordering.

The computational adversary tries to discern enough information about the cryptographic system so that it can decrypt a message in its polynomial time. It does this by making queries against multiple instances of the protocol to try gather information it can use in its decryption algorithm. It is objective is achieved when the probability of decrypting a message is better than a coin flip.

The queuing network method is primarily derived from the symbolic adversary, but the queuing network allows for the development of weaker adversaries which allows for more nuanced models of a protocol's robustness to be developed. This is done to provide a protocol implementer with the most granular statement of a protocol's sensitivity to disruption. The ability within the methodology to select the individual queues in the network that the adversary can manipulate weakens them in two ways. The first is by only having access to select exogenous variables means they may not be able to cease all traces through a honest agent's state machine. The adversary can manipulate a packet's arrival and processing rates which can be mapped over to models where they are adding or removing content from a message, or trying to simulate a large number of instances like in the computational model. This means that the adversary can be modelled with as little knowledge as required for the attack, to the point of violating principle[91]. Expected but incorrect packets can be modelled by generating a new packet type. The second is it limits the concrete knowledge the adversary has about the network. If the adversary only has taps on certain edges of the network or can only tell if a device has reached a specific state but not others, then they have to develop attack strategies based solely on inferences. These inferences are represented through the implementation's conditional probability functionality.These inferences can be set up the model an adversary reacting to if a packet goes down a certain edge of the network, or if a certain amount of packets are reached.

The methodology's adversary can also can be used to demonstrate some of the timing modelling that the other symbolic methods can't, by being able to model the order of packets arriving into a state and calculating the probability that a specific sequence of malicious packets arrives in the order that the adversary desires.

3.3.3 Performance Comparison

The complexity of the queuing network formal method is dependent on which state space is selected to analyse a given network. The worst complexity in methodology is the one for the global internal space configurations as it is performing a search of the state space for each state, which has a complexity $O(|\mathcal{S}^{GIS}|^2)$, to generate the transition matrix. The other state space configurations have more manageable complexity because the generation of the transition matrix utilises the deterministic structure of the state space to calculate groups of transitions in one go. The grouped nature of states meant blocks of concluding state of the transitions could easily be calculated, which led to a reduction in the number of operations that were needed to be performed. Despite this, the complexity for all the state space configurations are below the upper bound of proving confidentiality in $DEXPTIME$ by remaining in the polynomial time complexity class. This complexity could still be improved upon by researching alternative algorithms and data structures, but optimising the method wasn't the focus of the project.

Most symbolic methods require quite a lot human intervention to operate. Each methodology requires the user to formalise the protocol's rules and the agents initial knowledge and abilities before they can compute the proof for a single trace of the protocol. Each potential trace is tightly coupled with the adversaries knowledge and abilities, so the adversary has o be carefully edited for each run. The induction, applied π-calculus, and operational semantics have ben developed into forms that have a degree of automation to them. Each have associated algorithms that generate a set of traces, for a specified adversary knowledge and ability set, that searches the state space by going through permutations of an adversary's order of operations. Instead of formalising the rules of the run, along with an how an agent interacts with the run, all that user needs to do to create a model in the queue network method is to select the queues in the network the adversary is allowed to interact with, and then decided what range of values the adversary is allowed to input into the exogenous variables. This reduction in effort required for the user to build a model comes from the methods orthogonal objective than the other symbolic methods. The method is only trying to define a protocol's sensitivity to specific disruption instead of searching an entire state space to find a counter example. The turn around time for testing a different adversarial is a lot lower as the user doesn't need to change the definition of the adversary, but just change which queue and range of variables the adversary is manipulating.

Chapter 4

Attack Vectors Discovered in the IEC61850 SAS

This chapter lays out the the attacks developed by the author. Most sections present novel attacks discovered in the IEC61850 SAS. The first two attacks in this chapter did not use the author's formal method in their development. The first attack is a credential intercept attack discovered in the IEC61850 association communication model, and the second attack present an example of IEC61850's generic substation event class model can be used to created a workfactor amplification attack. It was during the course of the second attack's development that the author wanted to a more rigorous abstraction for proving that aspects of the protocol undermined it's own promises during disruption attacks. With a formal method proof security omission and defects could be shown to inherent in all implementations of the standard, regardless of the technical specifications of the devices used. From this requirement the queuing network formal method described in chapter 3 was developed. All the subsequent attacks in the chapter were discovered using the author's formal method. The only attack not directly related to IEC61850 security is the generalised packet injection model presented in section 4.5. This model is an expansion of the model used for the GOOSE packet injection attack presented in section 4.4, which allowed for exploration of the dynamics of packet injection attacks.

4.1 IEC61850: Credential Intercept Attack

This section discusses a credential interception attack against IEC61850's association model[120], which allows an adversary to hijack a session during an initial association between an IED and a control server. The consequence of this attack is the adversary denies access to a legitimate device, and can act in any way they please within the privilege of the legitimate device. This undermines the standard's promises of access control, and possibly synchronisation of state if the adversary chooses to not respond to control commands. The feasibility of this attack is proven using a context free grammar.

An adversary, who has no login credentials on the network, is able to hijack the login credentials of a legitimate user whilst they are logging into a logical node for the first time, or who hasn't already been given a predetermined authentication parameter. This scenario is predicated on the adversary doing some passive surveillance of the communications channel, as the two party association model is only instigated when a new entity is connected to the network. Once this precondition is fulfilled the attacker is able to proceed with the attack. This attack only works against networks that haven't deployed the IEC62351 security standard as it encrypts the message in this communication model. There are examples of IEC61850 implementations being deployed without IEC62351 being attacked. In December 2016 and 2017 Ukraine's power system network, which was running IEC61850 devices, was attacked by foreign adversary[82].

Figure 4.1: The logical nodes affirmative response

Figure 4.2: The logical nodes negative response

4.1.1 The Two Party Association Model

The two party association model describes how a client program can connect and transfer packets with a logical node. The standard procedure for the model is that the client sends an access request, $Acc-Req(SA/AP)$, message to a virtual view on the logical node server, LN. Included within the request are the client's login credentials, which includes an authentication parameter, AP, and the server access point reference, SA.
Once the server has received this request, it then decides how to proceed. If the client's login credentials are correct then the server will reply to the client with an affirmative message, *Acc*–$R^{+}(AID/Re)$, that will include an authentication ID, AID, and the result of the attempt, *Re*. However, if the client's login credentials are invalid then the server will reply in the negative, $Acc-R^{-}(Err)$, with an error message, Err.

Figures 4.1 & 4.2 depict the two possible session diagrams of the two party association model.

4.1.2 The Adversary Model

The adversary for this attack is a weakened form of the DY model[52]. The rules governing the adversary for this attack are:

- The adversary can see all packets passing between the client and the LN server, as the channel isn't encrypted.

- The adversary cannot send any message that they have not already seen.

- The adversary has no buffer of messages they have seen. They have to send the message directly after seeing it. Whilst unrealistic this limitation on the adversary is to demonstrate the ease with which the objective can be achieved.

- The adversary can forward and intercept packets.

4.1.3 The Credential Intercept Attack

The attack happens by combining the two potential responses of the server, depicted in figures 4.1 & 4.2, into one session. It begins with the client sending a legitimate login request to the LN server. The adversary sees the client's packet go through their intercept, and then sends a invalid login attempt to the LN. When the LN responds in the affirmative with the AID to the client, the adversary intercepts this packet. When the LN replies in the negative to the adversary, the adversary forwards the packet with the error message to the client. After this the client cannot use their login credentials. The session diagram for the attack is shown in figure 4.3.

Figure 4.3: Session diagram of the proposed attack.

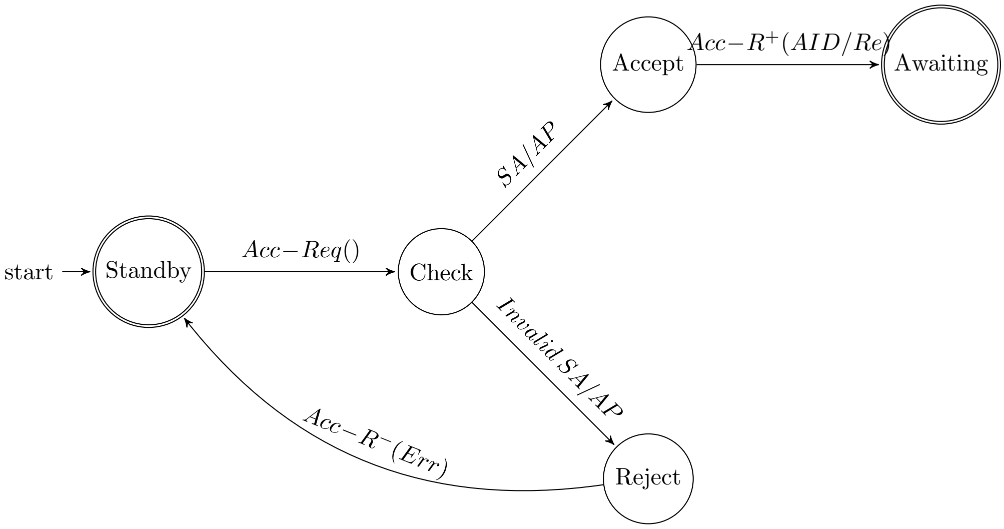

Figure 4.4: The automaton depicting one client logging into a logical node server.

4.1.4 The Automata

Figure 4.4 depicts an automaton that models the process of one client logging in, whilst figure 4.5 shows the union of two one client automatons to show the state machine for two users logging in simultaneously. The two user automaton allows the depiction of the attack described in section 4.1.3. In the two person automaton S represents the standby state, C represents the check state, and A represents the awaiting state.

4.1.5 The Context Free Grammar

The rules that describe a legitimate message that passes through the two person automaton are:-

- A login attempt for one access view must be completed before a second login can be attempted.
- There can only be two successful attempts per run of the automaton.
- An infinite number of failed attempts can be made before the first successful message and between the first and final successful message.

The following rules describe the form of the message that can pass through the two person automaton that leads to an undesired result:

- The adversary can only duplicate a message that has passed their intercept.
- The adversary's duplicate message can only be sent immediately after seeing it. They have no buffer.
- The adversary's '$Acc{-}Req()$' must come before the client's 'AP/SA' is processed by their login view.
- The adversary can only send invalid SA/AP credentials. This is to make sure it ends up in the state they desire (S_1A_2 or A_1S_2)

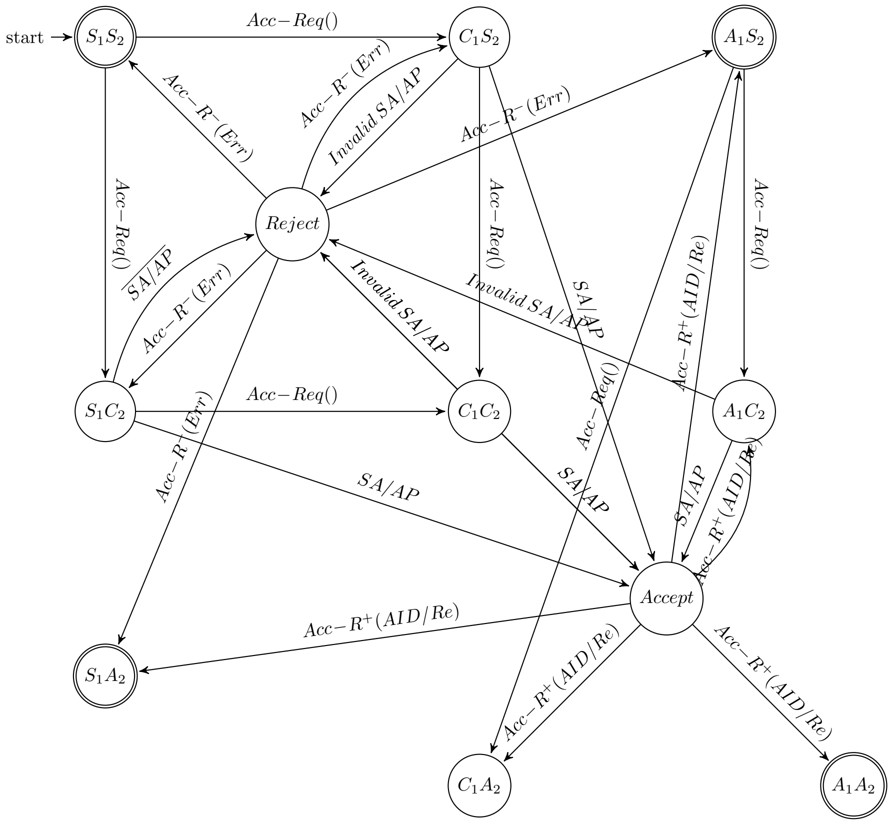

Figure 4.5: The automaton depicting two clients logging into a logical node server simultaneously.

- The legitimate user cannot login after the attacker has intercepted their credentials.

The objective of the adversary is to make sure that the automaton is driven through the C_1C_2 state. This collision state represents the adversary's intercept where they hijack the authentication ID and forward their error message.
The context free grammar that represent the above rules are:

$$\begin{array}{c}
\hline
S \rightarrow TATV|TWTW|TV|TW \\
R \rightarrow Acc - Req() \\
V \rightarrow Invalid\,SA/AP\ Acc - R^{-}(Err) \\
W \rightarrow SA/AP\ Acc - R^{+}(AID/Re) \\
T \rightarrow RU \\
U \rightarrow VT|\epsilon \\
A \rightarrow W|RVW|RWV \\
\hline
\end{array}$$

4.1.6 Mapping to IEC61850-7-2

The above grammar maps to the two party association model by:

- Rule S: Presents the four different message types. From left to right.

  1. Is the comprised attack form. If the attack is not attempted this leads to $n = 0...$ failed attempts, followed by a successful login and then another $n = 0...$ failed logins. However if rule inserts either 'rVW' or 'RWV' instead of 'W', then the undesired form of the message begins. This leads to two '$Acc-Req()$' messages in a row. They can both be seen as undesired as the attacker controls all messages passing through its intercept.

   2. A word with two successful logins with $n = 0...$ failed messages before the first and between the subsequent successful logins.
   3. $n = 0...$ failed logins.
   4. $n = 0...$ failed logins followed by one successful message.

- Rule R: Maps to the request message parameter.

- Rule V: Maps to the incorrect form of 8.3.2.2.2.1, the server access point reference, and 8.3.2.2.2.2, the authentication parameter. Followed by 8.3.2.2.5, response showing the failed attempt error, which “*shall indicate that the service request failed*”.

- Rule W: Maps to the correct form of 8.3.2.2.2.1, the server access point reference, “*which shall identify the server, with which the application association shall be established*”, and 8.3.2.2.2.2, the authentication parameter, “*for this application association to be opened*”. Followed by 8.3.2.2.3, response showing the successful login returning the authentication ID, which “*may be used to differentiate the application associations*”, and request message, which indicates ”*if the establishment of the application association was successful or not*”.

- Rule T: Is the rule that facilitates the $n = 0...$ repeats of the failed login, or it provides an ‘$Acc - Req()$’ packet before terminating the loop.

- Rule U: Provides the terminals to facilitate rule ‘T’.

- Rule A: Is the production rule for the attack. From left to right.

   1. Facilitates the normal success message stuck between to infinite failed attempts.
   2. Two ‘$Acc - Req()$’ packets followed by a failed login attempt and then a successful login.
   3. Like 2, but the error and success messages are reversed.

   2 and 3 are the undesired message forms

The above analysis has shown that the security promise of access control does not hold throughout the IEC61850 standard. The credential intercept attack has been developed and proved using context-free grammar against the two party association model. This scenario would allow them to cause physical damage to the SG, for example they could trip circuit breakers and cause undue stress on the distribution network.

4.2 IEC61850: Workfactor Amplification Attack

This section looks at an attack against the promise availability of the IEC61850 standard. During the investigation it was found that an attack using the generic substation event class model, as described in IEC61850-7-2 section 15[120], could be used to create a DoS attack.
The aim of the adversary of this attack is to degrade the performance of packet transfer between points in the SG communication network to below the acceptable QoS standard. The adversary achieves this by sending messages that connects additional subscribers or topological branches to a LN’s generic substation event subscriber list. This leads to the routers and IEDs on the network having to process extra messages from other devices that they do not expect to be within their broadcast network. The analysis that follows only focuses the calculation of the number of extra bits processed by devices. However, the analysis doesn’t cover the additional latency because that would be dependant on the hardware components of the grid which are beyond the scope of the standard.
This analysis is a proof of concept of this class of attack. It is modelled that the generic substation event communication model has been implemented deployed on a protocol independent multicast (PIM) multicast framework that has been applied to a network substrate that supports it.

4.2.1 The Generic Substation Event Class Model

The generic substation event class model describes the way an IED can broadcast data regarding its current, or changing, status to other devices within the substation that subscribe to its announcements. It is based on a producer/subscriber multicast model, and is implemented as an unidirectional process. There are two types of message in this model. Firstly, the GOOSE message, that is used to broadcast the IED's data, and the second is the GSSE message, which broadcasts any changes in state. When an IED is connected to the network it sends a GOOSE message that announces to devices its current status.

An IED on a generic substation event network will only check to see if the message it has received is a duplicate of a previous message, or if parts of it are missing. The method used to check for this is, again, beyond the scope of IEC61850. In this analysis it is assumed that the IED does not have access to the complete address space of the network and the packets received aren't cryptographically signed.

4.2.2 PIM multicast

The PIM framework is a collection of protocols that describe how devices can multicast packets across an IP network[179]. Depending on the bandwidth and subscriber update constraints of the implementers network topology, PIM offers different protocols for modifying a devices subscriber list. The two protocols that this analysis focuses on are PIM sparse mode and PIM dense mode, where the .

In the PIM sparse mode the subscriber list is updated whenever a device joins the network. The process begins when a new subscriber to the network issues a join request to be added to a publisher's subscriber list, a reverse path forwarding (RPF) check is triggered. A PIM-join message is sent toward rendezvous point (RP), as demonstrated in figure 4.6. The join message is multicast hop by hop upstream to the all the PIM routers until it reaches the RP. The RP router receives the PIM-join message and adds it to the outgoing interface list. The same process is done for when a router wishes to leave the network, but instead sends a PIM-prune message. When a source is added, it multicasts a PIM-register message and sends them by means of unicast to the RP router.

In PIM dense mode, an example is shown in figure 4.7, the principle way a subscriber list is generated is when the publisher periodically sends out a PIM-flood message across the network. This registers all devices on the publisher's subscriber network. If a subscriber no longer wishes to be on the list, it sends a PIM-prune message upstream to the source, which then removes it from the list. If a new receiver wishes to join before the next PIM-flood, they can send a PIM-graft message to the source to be added. However, if these are the main forms of list maintenance then PIM sparse mode is recommended.

4.2.3 The Adversary Model

The adversary model is similar strength to the one described in section 4.1.2. The only modification is this adversary has a buffer so they are not required to send messages they have discerned straight away.

4.2.4 The Attack premise

The attack is predicated on the adversary performing some passive surveillance on a devices' subscriber network. Through their observation they decide which branch or IED they wish to attach to the IED's subscriber list, and which type of PIM network it is. They also discern the IED's specific application ID, which is required for subscribers to receive the GOOSE messages. The adversary sends either a PIM-flood or PIM-graft, for dense PIM networks, or PIM-join or PIM-register for sparse PIM networks. The next time the publisher IED sends out a generic substation event message to the network the IED(s) that have been maliciously subscribed to the network will receive messages they weren't expecting. As they have no access to the address space they cannot tell whether they were meant to receive the message.

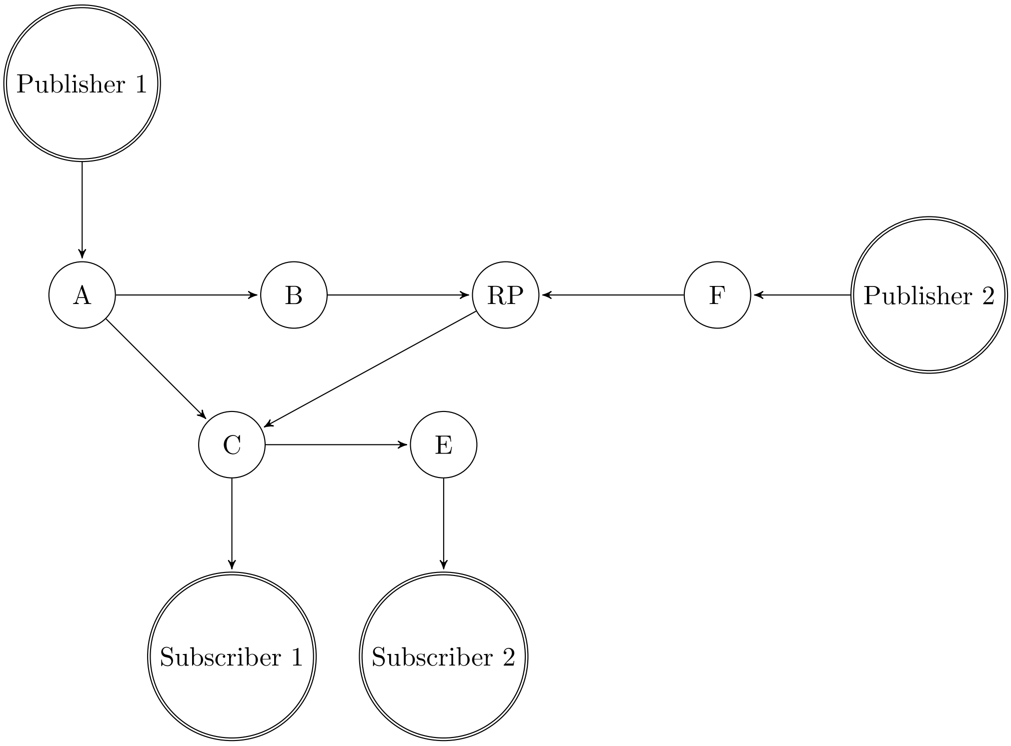

Figure 4.6: An example of the sparse topology PIM multicast system topology[179].

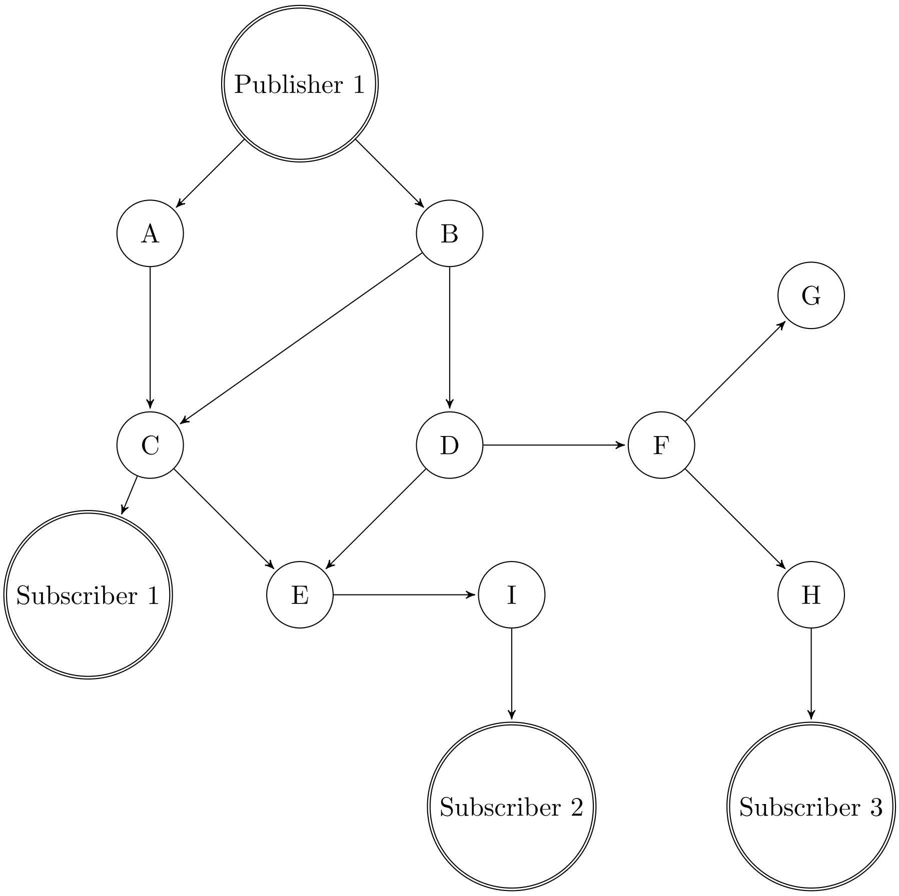

Figure 4.7: An dense topology PIM multicast system topology[179].

4.2.5 The Workfactor Amplification Ratio

The workfactor amplification ratio calculates how many bits of messages are produced to the amount of bits the adversary sent to initiate the attack.

$$\text{Amplification factor} = \frac{\text{Message produced as a consequence of the attack}}{\text{Messages sent by the adversary}} \tag{4.1}$$

The amplification factor for a GOOSE message is,

$$\text{Amplification factor}_{\text{GOOSE}} = \frac{A + \text{length of data set} + \text{data set}}{B + C}, \tag{4.2}$$

where is $A = 187$, $B = 65$, and $C = 4or32$ depending on whether the adversary chooses to edit the PIM message type, or create a new PIM message.

For a GSSE,

$$\text{Amplification factor}_{\text{GSSE}} = \frac{D + (2 * \text{length of data set})}{B + C}, \tag{4.3}$$

where $D = 170$

4.2.6 Examples

This section provides example workfactor amplification ratios for subscriber networks attacked using the generic substation event models of IEC61850. These examples are generated using the PIM topologies laid out in figures 4.6 & 4.7. All of the below examples takes the average number of status logical node variables, which is three, as the length of the data set. As the status variables are usually a boolean variable type, it is assumed for these calculations that they are boolean. For the purpose of these examples the adversary will create a whole new PIM message for their attack.

	AF_{GOOSE}	AF_{GSSE}
Case 1	3.96	3.63
Case 2	23.75	22.14
Case 3	11.87	11.07

Case 1 is set in the depicted dense PIM network. In this case the adversary has chosen to connect router I to the publisher's subscriber list, so to send malicious messages to *subscriber* 3. Case 2 is when the adversary connects a new publisher to the network.

Case 3 is the attack scenario applied to the sparse PIM network example. In this instance the adversary connects publisher 2 to the network to send malicious messages to both *subscriber* 1 and *subscriber* 2. In the case of adding another publisher to the network, the amplification factor would be the same as case 1.

These examples show that an adversary can cause maliciously generate more bits than they use to create the malicious additions to the subscriber list. If an IED is subscribed to too many lists maliciously this could lead to it violating the QoS times to process and act upon emergency messages. This delay in processing would lead to damage to the electrical distribution network and potentially blackouts.

4.3 IEC61850 Control Communication De-synchronisation Attack

This was the first attack discovered using the author's queuing theory method. The attack shows that an adversary can cause the client and server state machines in the IEC61850 control communication model to become de-synchronised during a session. The attack is only valid in the Select Before Operate (SBO) version of the communication model. The adversary achieves this by either increasing or decreasing the rate at which the server receives the $oper - req[TestOK]$ in the SBO control model, described in section 19.2.2 of IEC 61850-7-2 (shown in a truncated form in figure 4.8). The adversary can cause this disruption of state because the standard

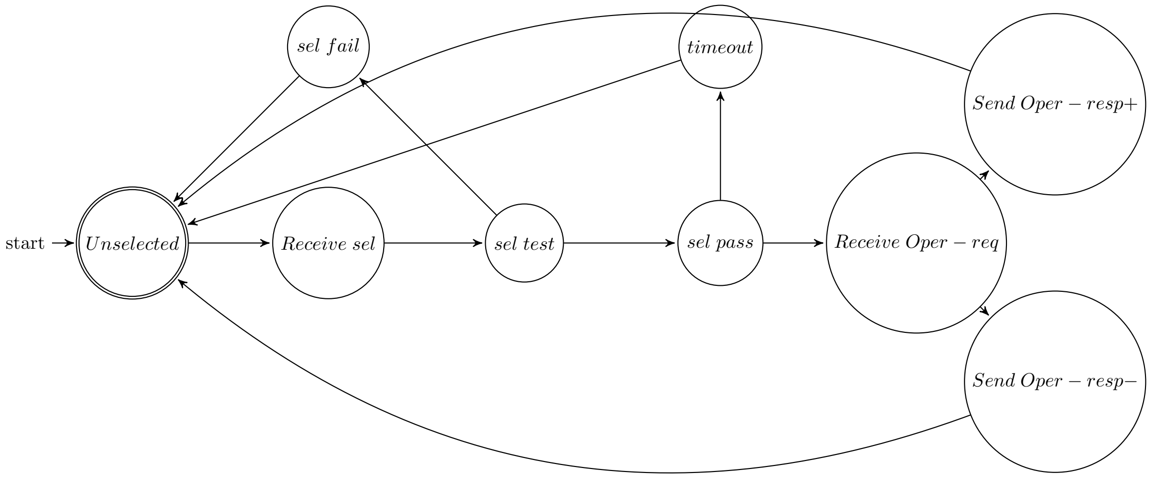

Figure 4.8: A truncated version of the server side of the SBO version of the control communication protocol run described in section 19.2.2 of IEC61850-7-2.

can be interpreted as not requiring the server to send out a *timeout* message to the client if it believes the session has ended before completion. Whilst this is happening the client still thinks the run is following the operation request branch of the protocol run, and will expect an update that will never arrive. This is a legitimate interpretation of the standard that devices will have to be prepared for, due to the QoS promise of interoperability of all devices regardless of manufacturer. This attack vector provides the adversary the ability to create doubt over the state of any logical node that has "*data object instances of a Controllable common data class and whose ctlModel DataAttribute that is not set to "status-only"*"[120]. This includes safety equipment, such as circuit breakers, whose response to an emergency situation have to be completed within $5ms$[121]. If the disruption caused by the de-synchronisation of states causes a control command to violate the QoS requirements, the adversary can cause physical damage to the distribution network.

The adversary in this result is the same the symbolic one described by DY[52]. The adversary in this model is an omnipotent and omnipresent party on the communications network who "*can intercept messages before they reach their intended destination, it can modify and reroute them, possibly with invalid sender fields*".

Figure 4.9 shows that adversary can increase the mean response time of the protocol run receiving the $oper - req[TestOK]$ message by several orders magnitude by altering the processing rate of queue accepting those packets. The dramatic increase in the response time to receiving the packets increases the probability that the protocol run will take the *timeout* path, which would cause the de-synchronisation. In this analysis no other variables were altered. The analysis also used the truncated form of the protocol run depicted in figure 4.8.

4.4 IEC61850: GOOSE Packet Injection

The GOOSE messaging service is one of the two models within IEC61850 to enable multicast messaging between devices in a substation. It is designed as a fast and reliable messaging system to distribute time critical data and commands to the relevant IEDs, such as commands to activate a substation's protection equipment. Due to the hard real time constraints $(3 - 4ms)$ and high level of reliability of all the messages the GOOSE messaging service is designed to be non-routeable, an entire message sent in a single packet, and received without acknowledgement. The GOOSE message is multicast across the communication's network using a publish-subscribe model, so different subscribers can be reached for different calibrated events[97].

The only security promises that the GOOSE messaging service must uphold within IEC61850

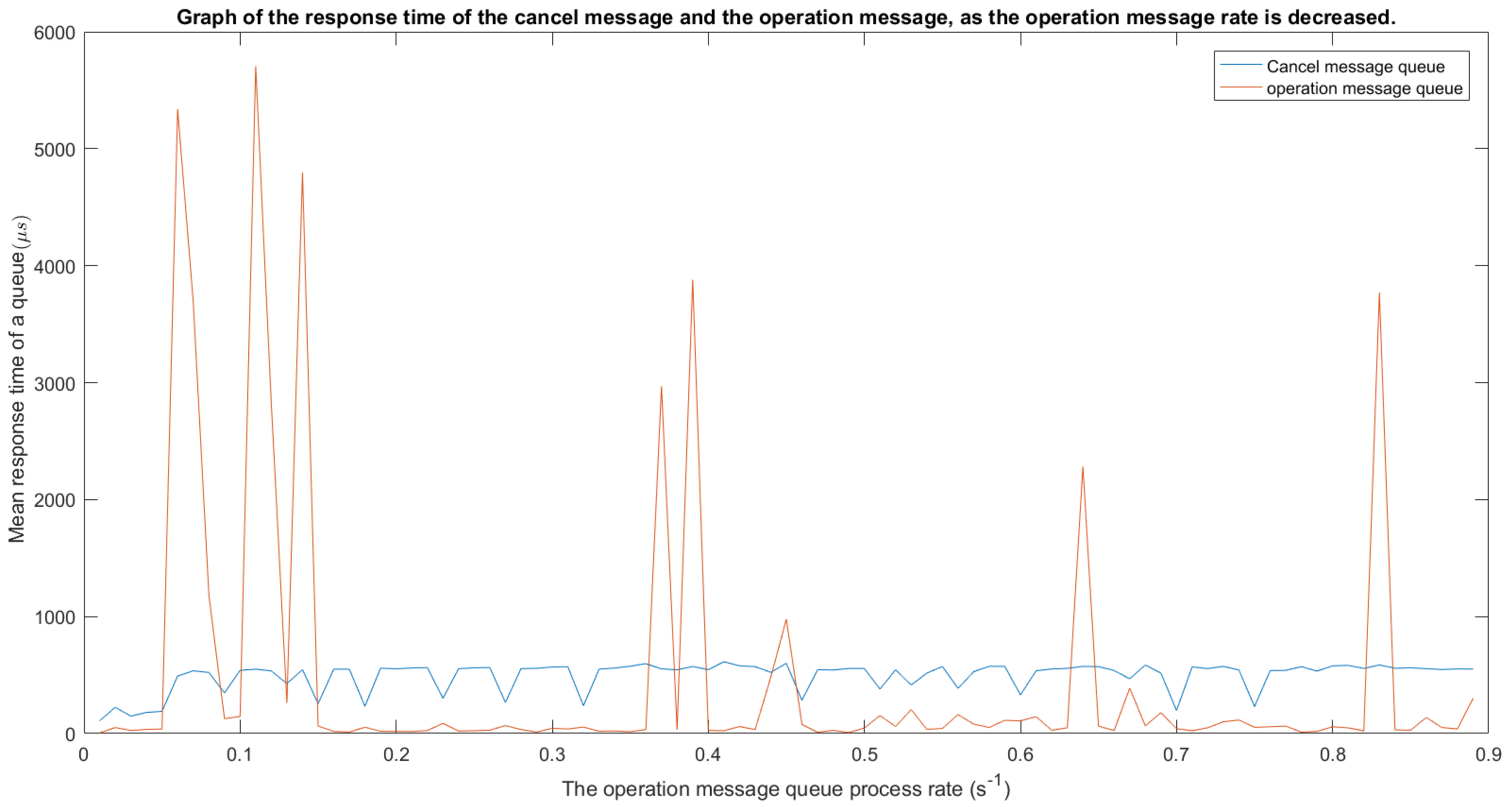

Figure 4.9: The graph shows the mean response time a queue dealing with *timeout* and receiving the $oper - req[TestOK]$ message as the adversary changes the processing rate of the $oper - req[TestOK]$.

are availability and integrity from transmission error. The standard does add that the GOOSE subscribers must be able to detect and dispose of duplicate messages. IEC62351 does integrate traditional security promises into the IEC61850 standard, but it explicitly does not add them to GOOSE as the latency from encryption technologies could violate the message QoS promises[148]. The only security feature that IEC62351 adds to GOOSE messaging service is a replay protection algorithm[**?**]. The IEC61850 and IEC62351 make no considerations for the other specific security problems faced by multicast protocols, which have been discussed by Judge & Ammar [84]. These are access control or group address obfuscation mechanisms. Having a policy on this would hinder an adversary's attempt to used the GOOSE service as an attack vector with which they could map and manipulate the communications network.

The combination of the lack of acknowledgement of message receipt and limited security protection makes the GOOSE service an ideal attack vector against an SG communications network. For this reason GOOSE has been one of the few communications models within IEC61850 that has been probed by the research community for vulnerabilities (an analysis of the various attacks was given in section 2.2). The main focus of GOOSE attack research has been attacks against availability of devices to prevent critical messages from being received. However, there seems to have been no consideration about whether the service is susceptible to injection class attacks.

This section describes the minimum necessary conditions that an adversary must meet for their injection attack to be successful, and remain undetected, given the stringent QoS time constraints of the GOOSE service. The objective of the adversary in this analysis is that they wish to achieve their disruption with as few messages as possible. It then goes on to demonstrate the effectiveness of this attack vector and considers various counter measures.

4.4.1 Message Injection Attack

This work builds off Strobel *et al.*'s[177] DoS attack to calculate the likelihood that the adversary's packet injection will be completed within the window of opportunity before the $stNum$ of a device's GOOSE messages rolls over. Whilst they give the upper timing bound of the injection within two minutes of the rollover to pass IEC62351's replay detection algorithm, they provide no consideration to the rate of injection needed for the attack to deny any messages after the rollover has happened. This analysis picks up from here.

Using this starting point an adversary model can be developed for when they wish to complete their objective using only one message. For this attack to work the adversary will be required to have performed some passive surveillance of the target subscriber IED. This is so they can discover what the current $stNum$ is, and the rate at which they change. Whilst it is not necessary for the adversary to obey Kerckhoffs' principles[91], they must at least know that the GOOSE $stNum$ rollover occurs when it reaches 2^{32}, and that they have to complete their injection within two minutes of the rollover happening. Finally, they must be able to manipulate the rate of transmission of individual messages passing along the communication between devices. First to delay their arrival, before accelerating it so it can be injected at the desired time.

If it is assumed that the subscriber's channel has no buffer of messages it receives, then the chances of adversary's malicious message being the first message increases exponentially with the rate of injection. This is shown in figure 4.10, which shows that the adversary needs to be of order hundred times faster than the normal injection rate to guarantee their success.

4.4.2 Countermeasures: Message Buffer and Rate Limiting

Introducing even a small buffer of messages, to slightly delays the GOOSE message being received by the IED, reduces the probability of the malicious message being the first packet to below $\frac{1}{2}$. Adjusting the adversary model to allow for message duplicates to be injected, does not increase their probability of success greatly if they wish to maintain the objective of not DoSing the buffer with their malicious messages. Another method of limiting the success of the adversary is by limiting the range of injection rates that will be accepted by the subscriber, this means devices can only send messages to the IED with a certain frequency. These two

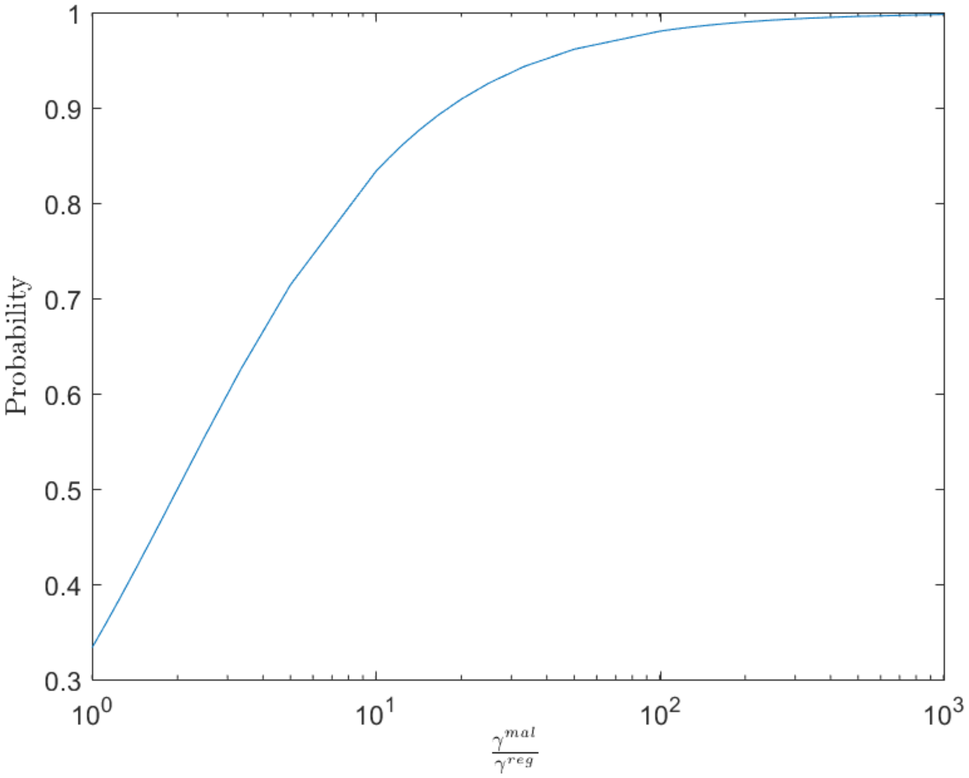

Figure 4.10: The probability of success of an injection, given the ratio of malicious and regular traffic $\left(\frac{\gamma^{mal}}{\gamma^{reg}}\right)$.

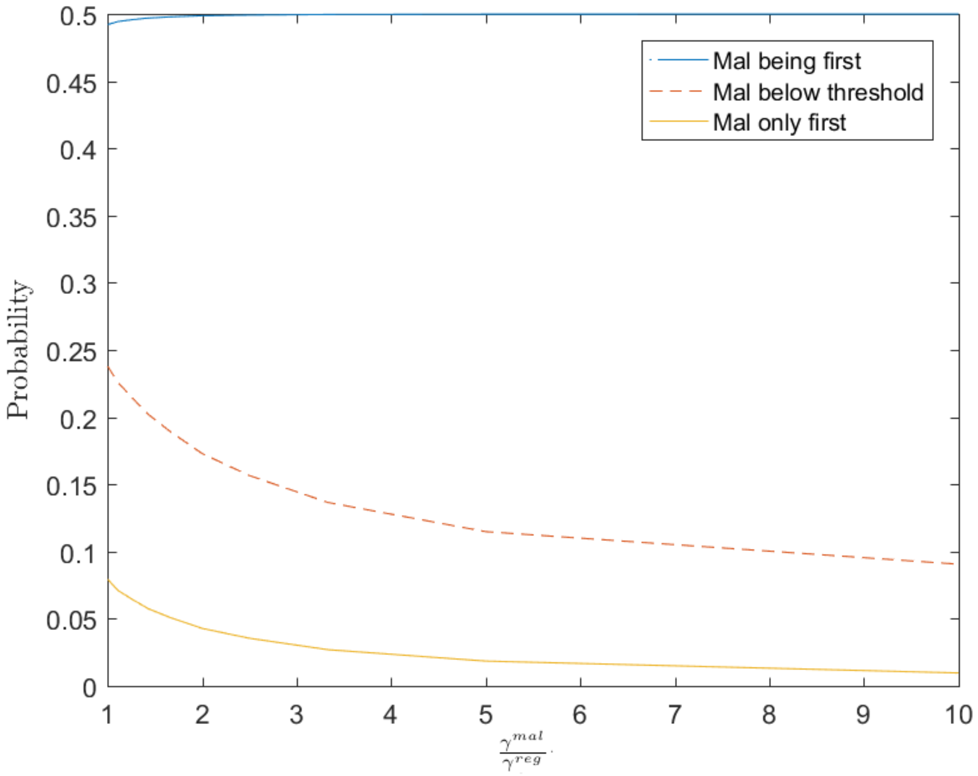

Figure 4.11: The probability of success of an injection, given the ratio of malicious and regular traffic $\left(\frac{\gamma^{mal}}{\gamma^{reg}}\right)$. There is a message buffer, of 5, and an injection rate limit of only ten times faster than regular traffic.

methodologies are demonstrated in figure 4.11, where the buffer is only five messages long, and the threshold of detection is two or more.

4.4.3 Countermeasure: Evasion of buffers & rate limitation and Implementing Inflexible QoS

In the above analysis it is presumed that the malicious message is the same size as the regular traffic. However, if an adversary gains the ability to edit and increase the number of data sets in the GOOSE message it can be shown that they may be able to circumvent the above the countermeasures. In this attack the adversary model is expanded so that they can intercept and edit a GOOSE message. The further knowledge that they require is thee upper limit of size of the *DatSet* that a GOOSE message can carry without tripping the integrity check. As shown in figure 4.12, contrary to the previous section, the optimal rate of injection to evade the discussed countermeasures with the enlarged message is slower than the regular rate of injection. This means for the evasion to be successful the adversary would also be required to change the goal of their passive surveillance. Whilst this attack vector doesn't increase the adversary's odds of remaining undetected, it does increase the chance of their injected message being first. A counter measure to the above attack vector would to have more stringent enforcement on the QoS transmission standard. Currently IEC61850 standard still allows for messages that arrive late to be processed. If the standard tightened this QoS promise, there would be no guarantee that publishers message could get through. An adversary would manipulate this counter measure, using the previously stated adversary model, to invalidate a single GOOSE event in a sequence of GOOSE events. That is to say that a *stNum* is changed without any time for retransmission between the change. An example of this would be in the issuance of tap changer commands[170]. The commands to adjust come in a succession of GOOSE messages,

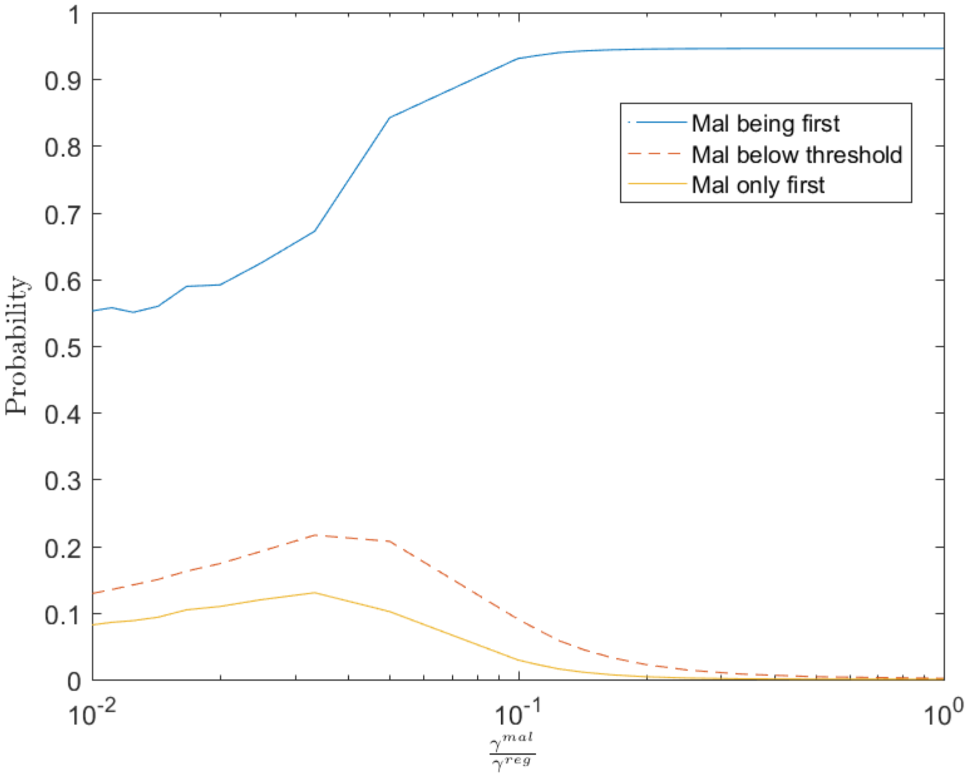

Figure 4.12: The probability of success of an injection, given the ratio of malicious and regular traffic $\left(\frac{\gamma^{mal}}{\gamma^{reg}}\right)$, given a message buffer, when the malicious message is two orders of magnitude larger than the regular message.

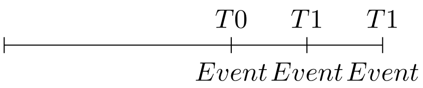

Figure 4.13: The layout of the attack against a tap changer, when QoS is strictly enforced. Where $T0$ & $T1$ are different QoS time requirements defined in IEC61850-5[121]. An 'event' is a new event that causes the GOOSE publisher to start a new $stNum$.

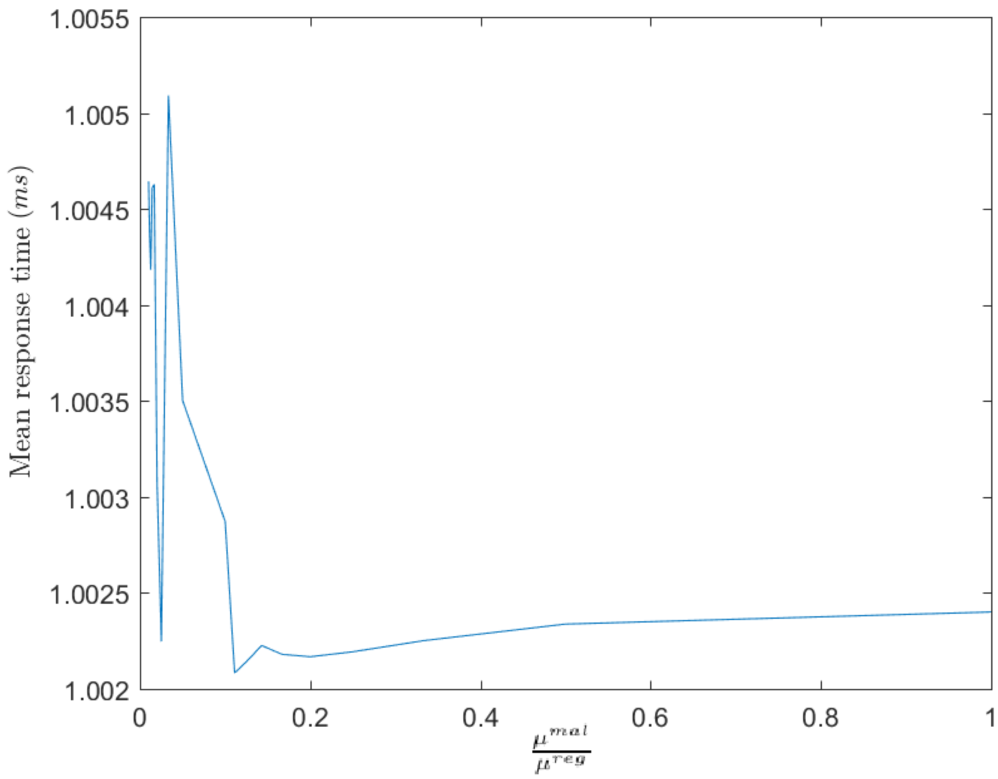

Figure 4.14: A demonstration of the increased processing time when enlarging a malicious message in relation to regular traffic. The x-axis is the ratio of malicious and regular processing times $\left(\frac{\mu^{reg}}{\mu^{mal}}\right)$.

as shown in figure 4.13,and if the tap does not end in the correct position it is likely to damage the physical equipment. Figure 4.14 shows that a less than one order of magnitude increase in the size of the GOOSE message can create a greater delay in the expected processing time of messages. All the variables in figure 4.14 are set to the same, and only μ^{mal} is increased in relation to μ^{norm}. This delay would make subsequent messages miss their QoS permitted window.

This look at the stricter enforcement of QoS presumes that there is no change in the order of magnitude between the translation of a change in the size of the message and the time it takes to process a message. The change in size is demonstrated in the queueing theory model as an increase in time of processing of the malicious message, which has now been uncoupled from the processing rate of regular traffic.

A counter measure that would help prevent the adversary's subversion of the rate limiter, would be to bind the $ConfRev$ variable with the $DatSet$. If the number of data sets changes but doesn't have a corresponding event then the IED disposes of it in the integrity check. This would undermine the adversary being able to increase the size of the GOOSE message.

4.5 Packet Injection Model

This section presents a generalised model of packet injection attacks that that allows for the quantitative analysis of a device's properties to discern the likelihood of an adversary successfully injecting their malicious payload. This model builds upon the GOOSE injection model described in section 4.4. The model makes it possible to study how altering parameter affects the adversary's likelihood of success, along with providing a testbed for developing countermeasures. Having a testbed allows the user to see if the countermeasures hinder the adversary

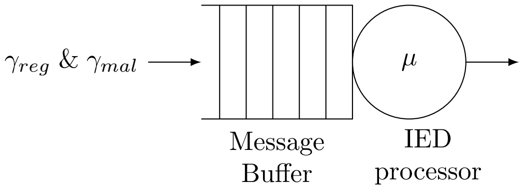

Figure 4.15: A queue representing an IED with a message buffer (K) of 6, with 5 messages in the buffer. It has both regular and malicious traffic coming into it, but they are both processed at the same rate.

whilst still remaining compliant with device's chosen communication protocol.
Further exploration of packet injection attacks was pursued, despite them not necessarily being considered an attack vector for disruption attacks, because the adversary is trying to achieve their goal without creating a noticeable difference to the expected distribution of packets. Creating a model of the weakest possible adversary's injection attack allows for sensitivity to disruption of a standard's QoS to be gauged.

4.5.1 Injection Attack: The Adversary Model

The adversary using a packet injection attack vector wants to their message placed in a server's communication channel with the minimum number of packets. This is so they can minimize the disruption to the expected distribution of messages to avoid detection by the server's IDS. In the model presented the adversary wishes to complete their objective using only one message. The adversary is allowed to do some passive surveillance on the communication channel to discover the rate of packet arriving into the device. The adversary doesn't need the power to generate new packet, just be able to manipulate packet that passes through their intercept. They will be able to change the increasing the packets size (altering their μ) and change their rate of transmission (altering their $\gamma or \lambda$).

4.5.2 Injection attack: Altering the Rate of Packet Arrival

The adversary's objective is not to DoS the queue but to inject their malicious message into the device without triggering its IDS. The focus of the model is to show that a malicious packet can be injected, not the consequences of the injection. As stated in section 3.1.1 the content of the packet isn't encapsulated in the queuing network methodology, and the explicit consequences of the injection is dependent on the communication model being attacked. The adversary achieves their goal by injecting their malicious packet into the front of the device's buffer before any regular packets enter the queue. This attack is modelled using one $M/M/1/K$ queue, as shown in figure 4.15. Figure 4.10 shows that if there is no device buffer that if the adversary is able achieve an arrival rate two orders of magnitude faster than the regular packet's arrival rate, then there is a near certainty that their injection attack will be successful. However, if device incorporates a message buffer then the maximum certainty of a successful injection drops with each successive increase in buffer size, as shown in figure 4.16. Continuing to expand the message buffer indefinitely could be a countermeasure to injection attacks, but it comes at the cost of adding a latency to processing packets in device. This latency is shown in figure 4.17. This added latency may be acceptable for some distributed protocols, but it is not acceptable in the SAS like IEC61850 due to hard real time requirements of the standard. Another countermeasure would be to limit the arrival rate of packets into the device to try and maintain the expected distribution of packet arrival.

4.5.3 Injection attack: Altering the Size of the Malicious Packet

Attempting to DoS a device by flooding it with packets will likely trigger its IDS, leading to the channel being cut, and is also computationally inefficient for the adversary. A more efficient method of DoSing would be for the adversary to inject a packet that takes a lot longer than expected for the device to process, consuming the available space of the buffer for as long as

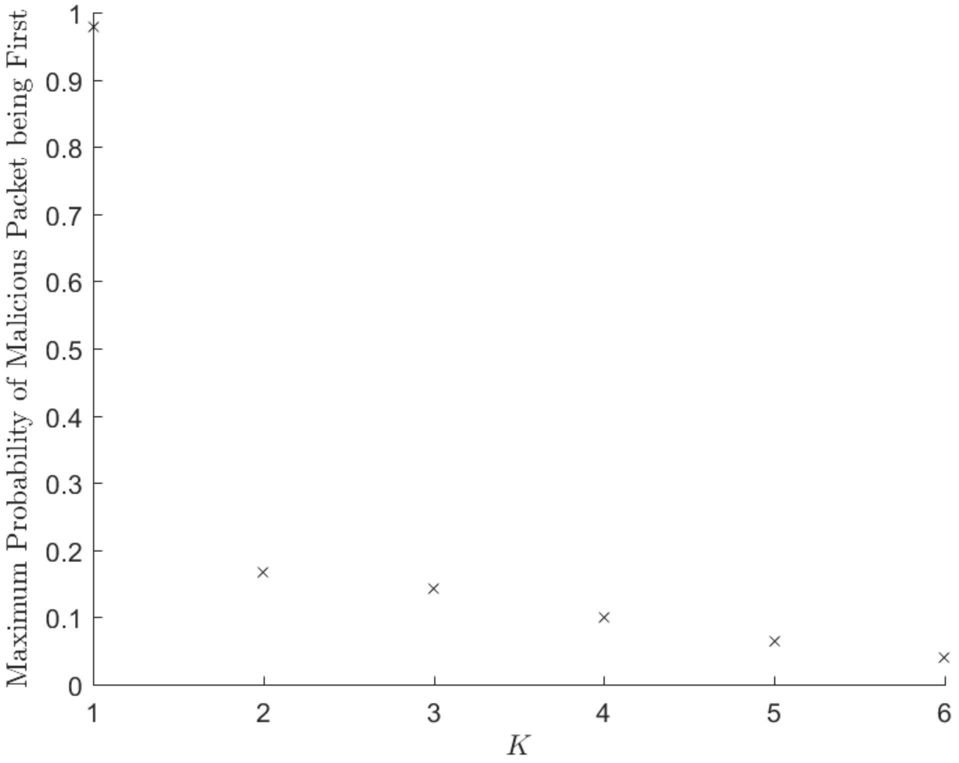

Figure 4.16: The size of the buffer, K, plotted against the maximum injection probability achieved by the adversary.

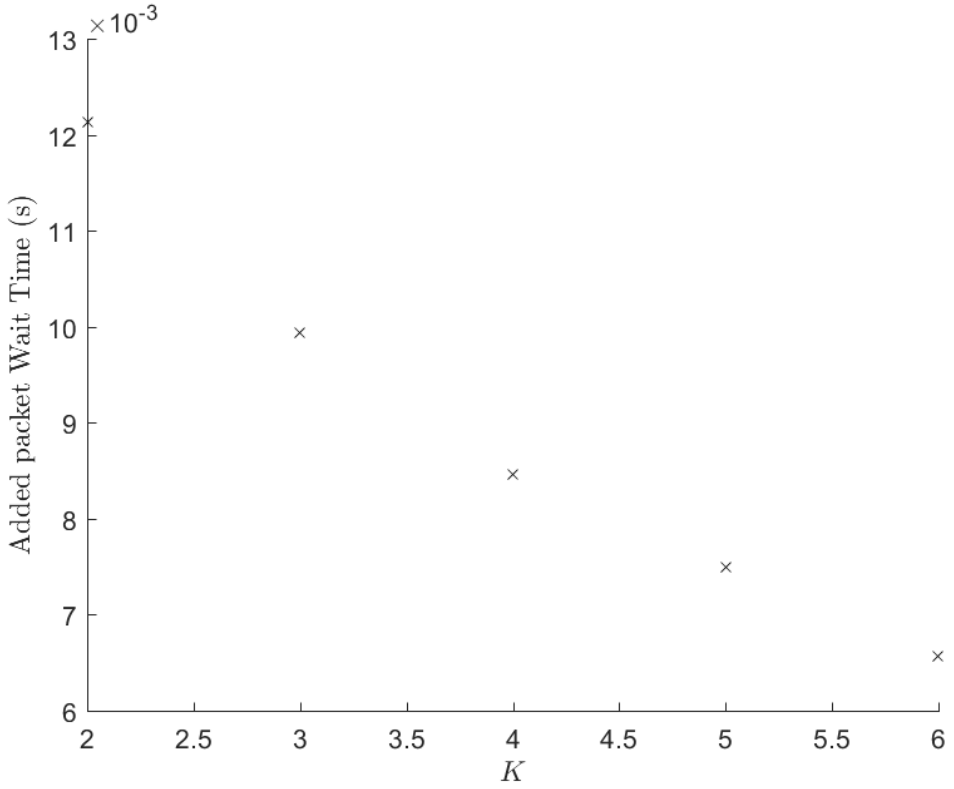

Figure 4.17: The size of the buffer, K, plotted against the added latency of processing of packets.

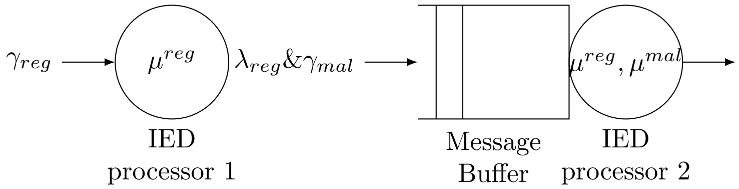

Figure 4.18: A queue representing an IED with a message buffer (K) of 6, with 5 messages in the buffer. It has both regular and malicious traffic coming into it. The malicious packet take longer to process as it takes up 4 spaces to process in the buffer.

possible, which leads to a disruption of the normal distribution of packets. They could do this by increasing the size of the packet. This attack is modelled using two $M/M/1/K$ queues in tandem. The adversary achieves their objective by making sure the first queue can no longer transmit packets, as shown in figure 4.18. Figure 4.19 shows the exponential distribution of this class of attack when $K = 2$. At two orders of magnitude larger than the regular traffic, the node can no longer receive regular traffic.

The same exponential distribution occurs across packet processing rates for larger buffers; countermeasure for this attack would include limiting the maximum size of packets accepted by a device. It should be noted, however, that a queuing network model could also be used to find the correct balance of malicious packet arrival rate and regular packet size to circumvent this. The calibration of a device's IDS would have to have a high sensitivity of correctly identifying malicious packets to reduce the probability of success of this kind of attack.

4.6 IEC61850: Timing De-synchronisation Attack

This section describes a de-synchronisation attack aimed at making the IED's in an SG communication network synchronise with a master server that has the wrong time accuracy for their functionality. An adversary would want to disrupt the time reference of an IED because reliable common time reference for sensors, actuators, and control logic is required for the safe and efficient operation of SGs. The common time reference is required for the monitoring systems to determine causality, for monitoring a network or power system's quality, and to co-ordinate control actions as well as to operate safety mechanisms correctly. Moreover, such synchronisation is also critical for the detection of incidents and attacks in the reconstruction of events. However, these functions require different levels of precision and accuracy.
Both the IEC 61850 [189] and IEC 60870-5-104 [188] standards explicitly make reference to the IETF NTP [123], whose security has been reviewed in section 2.2.9. Whilst the IETF has made efforts to secure NTP with RFC7384 [133], most implementations of the NTP standard remain unsecured. There have been various different attacks that have been theorised for both NTP and PTP. A compromise of the time synchronisation can easily affect the functionality and security of the entire power network as it allows for both attacks and obfuscation of attacks.

This section goes over what both standards say about their time synchronisation communication models, to demonstrate their lack of robustness, before presenting how an adversary could force a desynchronisation in the author's queuing network methodology.

4.6.1 Time synchronisation in IEC61850 & IEC60870-5

Both of the IEC 60870-5 and IEC 61850 standards defines the accuracy of the time source required for an IED to perform its function, and how they should synchronise with the clocks to get the correct time data. Below is a brief description of the time synchronisation communication models in each standard, where the deficiencies are pointed out. Before finally describing the potential subset of implementations that these omissions are likely to be vulnerable to exploitation.

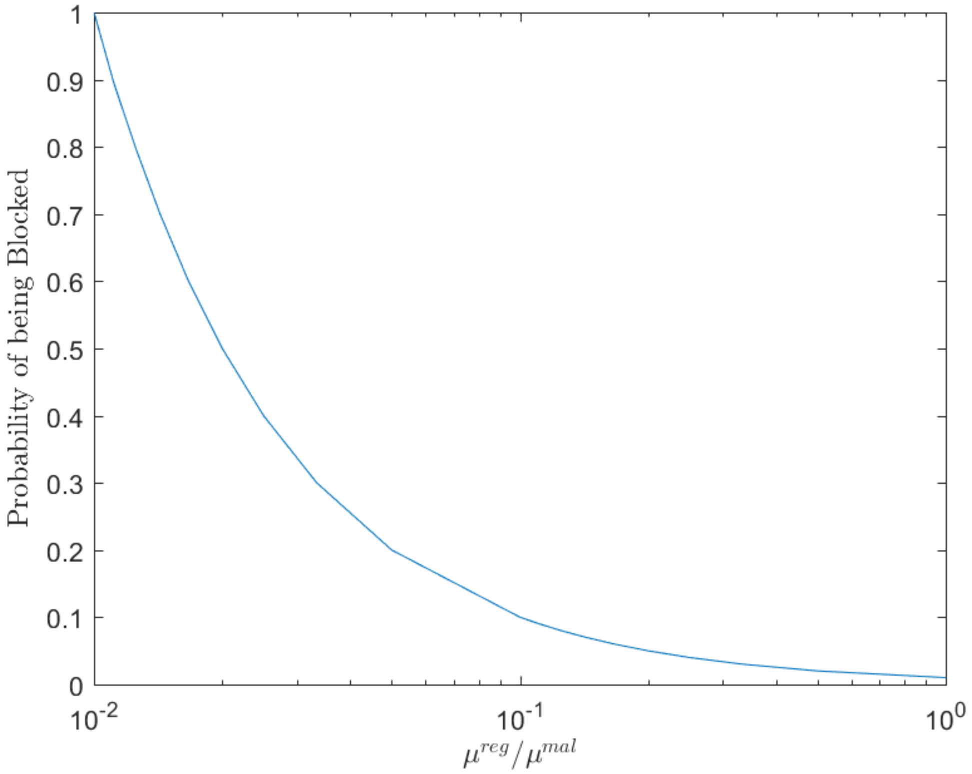

Figure 4.19: The ratio of regular and malicious packet processing rates plotted against the probability that will be blocked from receiving any further packets.

IEC 60870-5

While IEC60870-5 (-101) can rely on implicit time synchronisation based on synchronous communication, its adaptation to wide-area Internet Protocol communication architecture means that IEDs must be able to connect to dedicated time source. The IEC 60870-5-104 standard provides two different time synchronisation models. The first is the synchronisation process between a control server and its client server, and the second is synchronisation based on a calculation and dissemination of the transmission delay between servers. The first process is a simple call-response protocol, with the client device's clock updating when the control server receives an affirmative reply from it. The standard requires clients to be able to reply to the control server for the operation to be valid but does *not* state a preferred time source for the control server to reference. The only integrity check client performs is if the synchronisation process is completed later than the client expects. If this occurs, it flags any operation it performs after completion as potentially inaccurate until its clock is updates again. The standard neglects to state what happens to the flagged messages.

The time delay synchronisation method is another call and response process. During the call and response, the control server calculates the time delay between the two devices as

$$td = \frac{rdt - (sdt + tr)}{2} \tag{4.4}$$

where td is the time delay, rdt is the round trip time, sdt is the time that the client synchronizes to, and tr is the time it takes for the client to reply. Once the control server has completed this operation, it forwards the delay to the client it has synchronised with.

IEC 61850

The time synchronisation communication model described in the IEC61850 standard is minimal in its description. It gives an overview of how time synchronisation occurs between a dedicated time server and a substation and requires a multicast protocol to tell the client IEDs what the new time is, but does not specify how an *authoritative* time is sourced from a wide-area communication network. The standard defers to either GNSS or the simplified NTP protocol as possible time sources, but it does not elaborate on how they interface with the network or how the correct precision is achieved. The only information explicitly called for is that the time server needs to update at a set time interval, specified level of accuracy, and to not exceed a time limit before the next update. Although not explicitly declared, the elapsed time is used to calculate the transmission delay between the time server and its clients. The time server can use multicast to complete the synchronisation operation with its client IEDs.

Exploitable Omissions in the Standards

A single local reference may suffice in trivial cases of deployment, but for larger expanses and distributed systems, multiple time references are going to be required. Here, GNSS receivers are widely used, potentially in conjunction with terrestrial support systems to enhance accuracy. Such systems, however, are not feasible to be deployed indiscriminately for individual IEDs and components since e.g. placement of antennae and line of sight to sufficient constellations are not always optimal. Moreover, it is anticipated that future SG deployment may include heterogeneous systems not on a single time reference. These considerations have not been factored into the development of either time synchronisation communication models. Neither model address what an IED should in the case of delayed or missing timing data, or if they receive data with the incorrect accuracy. This leaves it up to the manufacture of IED's to decide what the appropriate course of action should be, which could lead to either solutions interfering with each other, undermining the interoperability of devices, or a manufacture presuming no solution is required. This omission in could be exploited by an adversary and allow them to instigate events that damage the transmission infrastructure. This could happen through the manipulation of the state estimation of a transmission network, which could lead to the misapplication of protective action [9]; moreover, it should also be noted that such actions also render auditing and monitoring for intrusion detection problematic as time-stamps ordering of

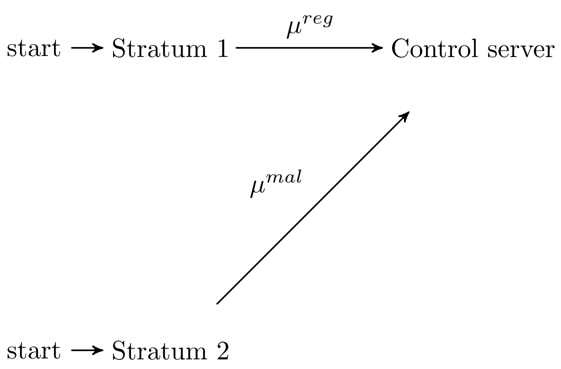

Figure 4.20: An example of network topography used in this class of attack. The NTP servers are on different time accuracy strata, and the adversary manipulates the arrival rate of the packets from the them to ensure that control/master server is synchronised with the inappropriate strata.

events cannot be relied upon.

4.6.2 Adversary Model

The adversary for the attack is a weaker form of DY[52] symbolic model. In the model the adversary can only manipulate the processing rate of packets of different NTP synchronisation server's queues, but can't modify the communication or endpoints. This maps to them being able change the rate of packets travelling across a communication channel. This attack can be carried out even with RFC7384[133] implemented across the network, as they don't manipulate the content of the packet. The adversary doesn't attack the servers themselves. The channels are represented as arrows in figure 4.20. It should be noted that the adversary needs to be able to internally synchronise information and commands between the adversary's taps on various edges of the network that make up communication channel; in this work it is assumed that this is the case, like all implementations of the DY adversary, and do not model this explicitly.

4.6.3 Accuracy Synchronisation Attack

IEC 60870-5-104 and IEC 61850 IEDs require different levels of time accuracy for the different functions that they perform. From power measurements ($\sim 1\mu s$) to logging ($\sim 100ms$)[156], IED and SCADA systems require a time source with the appropriate accuracy for their functions. If the IED is either solely dependant on an NTP network, or is unable to connect to another time source (GNSS satellite, PTP, etc), then it must be able to select a NTP stratum with an appropriate accuracy [127]. We assume that a given power network holds more than one strata of NTP server for reliability reasons. *Neither of the state machines in IEC 60870-5-5 or IEC 61850-7-2 check if the accuracy of the time source they are connecting to is appropriate the function required, nor do they have any correction procedure if an inadequate level of accuracy is chosen.* This oversight provides an adversary with an attack vector that can undermine the functionality of IEDs on the network. This attack vector exists because, as stated in section 4.6.1, neither standards incorporate sub-protocols on how to handle NTP state transitions that will cause the control/master server to violate the QoS requirements. The standards presume that all time sources will always provide the current time with the correct accuracy.

The premise of this attack is that the adversary manipulates the rate of packets arriving from the NTP server from the stratum with correct accuracy (e.g. by interfering with legitimate communication), stratum 1, to increase the probability that the control/master server will synchronise to an NTP server with an inappropriate accuracy, stratum 2. The adversary manipulates the traffic of the communication channels serving the control/master server, which

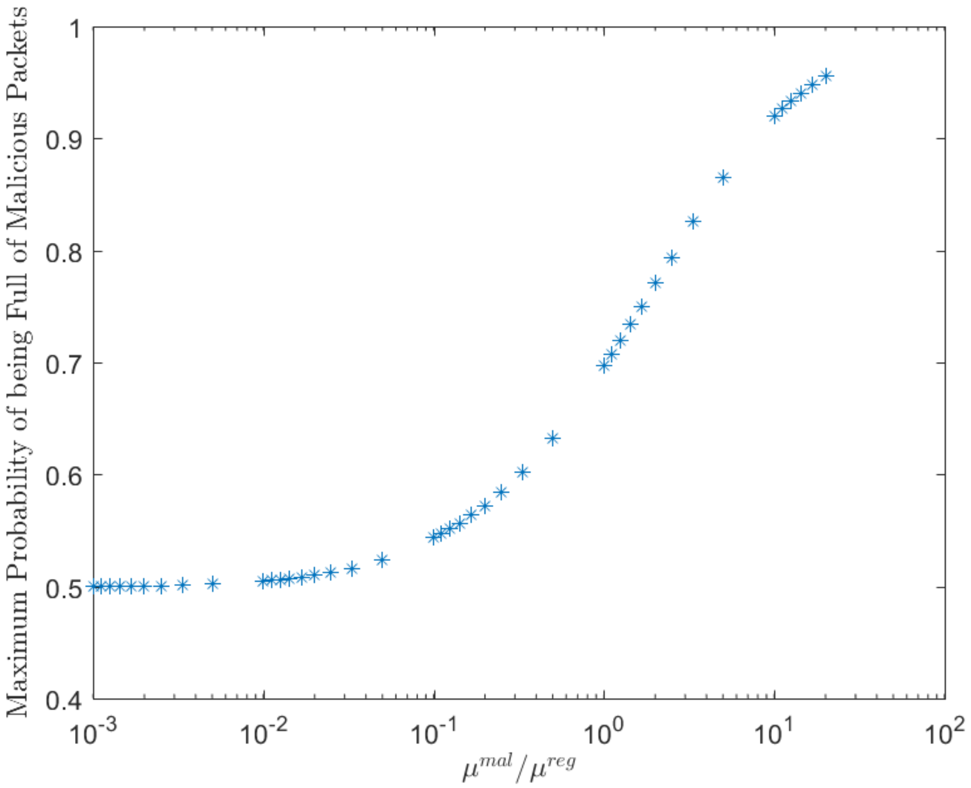

Figure 4.21: Maximum probability of the control server queue being full of malicious packets given the adversary's manipulation of the ratio of arrival rates of packets from the different NTP servers from different strata.

may e.g. be achieved by a MiTM or man on the side attack but which are not elaborated here. The simplified network topology used to demonstrate this attack is shown in figure 4.20. In this model, malicious packets are defined as packets with inappropriate time accuracy. Using the exogenous equations described in section 3.1.1 the effective arrival rate in each queue/NTP server is,

$$\lambda^{eff}_{stratum\ 1,2} = \gamma_{stratum\ 1,2}(1 - P(N_{stratum\ 1,2=K_{stratum\ 1,2}})) \tag{4.5}$$

$$\lambda^{eff}_{control} = \gamma_{stratum\ 1}(1 - P(N_{stratum\ 1=K_{stratum\ 1}})) + \gamma_{stratum\ 2}(1 - P(N_{stratum\ 2=K_{stratum\ 2}})) \tag{4.6}$$

$eff_{control}$ is the sum of the previous two λ^{eff} as all of there packets can only go into the control server. The probability of the queues being blocked are

$$\mathcal{P}_{stratum\ 1,2} = P(N_{control} = K_{control}) \tag{4.7}$$

$$\mathcal{P}_{control} = 0 \tag{4.8}$$

The common acceptance rate for the stratum 1 & 2, queues, equation 6, are

$$\frac{1}{\widetilde{\mu^{a}_{stratum\ 1,2}}} = \frac{\lambda^{eff}_{control}}{\lambda^{eff}_{stratum\ 1,2}\mu_{control}} \tag{4.9}$$

whilst

$$\widetilde{\mu^{a}_{control}} = 0 \tag{4.10}$$

From the above equations the endogenous equations can then be derived. We have conducted a simulation of the attack outlined; this simulation attack shows that adversary only needs to create a ratio of 20 times the difference in processing rates of the two servers for the probability of the success of the attack to approach certainty. The author could not find any second order affects in the simulation. It was hypothesised that with a sufficiently large difference in processing rate, the probability of the stratum 1 being blocked would increase, but no correlation was seen. The author would like to stress that this work attacks a different part of the time-keeping system than the work of Barreto *et al.* [17]. Their attack focuses on disrupting the use of the communication channel between the IED control/master server and the IED slave servers within the distribution network, which are defined in the standards. The work presented here, however, focuses on attacks against the communication channel between the IED control/master server and potential time sources.

Chapter 5

Conclusion

5.1 Conclusion

This research project demonstrated that the alternative security promises of availability and synchronisation of state could be modelled and tested using a symbolic formal method, expanding upon the set of promises that are traditionally tested (CIA and non-repudiation). The work presents these promises as the distinctly new class of attacks as it is argued that the promises needed to be tested against the weakest possible adversary, with the minimum set of abilities and knowledge, whose objective is the disruption of a session, in contrast to other promises that are implemented to repel stronger symbolic attackers whose objectives are to learn the content of messages passed between agents. The weaker adversary is required for these promises to find the sensitivity a protocol session has to adversarial disruption. These arguments culminated in the development of a probabilistic formal method based on queueing networks developed by the author, which used a network of $M/M/c/K$ queues to model packets travelling through an honest agent's state machine or between a network of agents using a specific protocol. The methodology separates the properties of the communication channel, such as their ability to see the entire network topology, from the adversary model, which differentiates itself from the traditional DY adversary. This facilitated the development of weaker symbolic adversaries which made it possible to discern the limits of protocol sessions to be identified. The separation of channel and adversary meant that the adversary's ability to manipulate packets and the knowledge of the environment and distribution of packets could be limited to being only being able to be placed on certain paths of the network, so they are only allowed to survey a subset of packets being passed between honest agents, to limit their knowledge acquisition to the level of inference. It also meant for that more nuanced adversary powers, such as the ability to manipulate the size, arrival rate, and probabilistic distribution of packets received by the honest agent could be modelled. The author's methodology builds on Osorio & Bierlaire's $M/M/c/K$ queuing network. Additional state space configurations were added to their approach, which have been verified with a set of completeness and correctness proofs. The new state spaces allow for the inspection of the number of packets in a queue and the order of packets in the queue, along with Osorio & Bierlaire's packet queue status state space. These varied state spaces were required to be able to put limits on the symbolic adversary.

The author chose the SG domain because it was clear from the literature that the introduction of the TCP/IP technologies into the domain of electrical transmission and distribution had created a tension within its standards between the safety and security requirements of the cyber-physical systems that had remained unresolved. Traditionally the electrical distribution systems had depended upon the airgap of their communications networks and a lack of broad implementation of SCADA technologies to keep the attack surface relatively small. However, implementing the SG technologies onto with traditional internet technologies to increase the amount of system automation and enable two way communication has also increased the number of attack vectors. The development of new SG standard's haven't grappled this new problem space into their design philosophy leaving these critical pieces of infrastructure open to malicious interference. On top of this the absolute rigidity of the SG standard's safety mechanisms leaves little functional space within the domain to implement traditional security technologies. Beyond designing a new class of network communications protocol that are flexible enough to accommodate both safety and security promises[42], it is likely that SG technologies will have security problems for the foreseeable future. The lack of traditional security technologies means that the slack has to be picked up elsewhere within the ecosystem such as monitoring and standard development.

IEC61850 SAS, and its ancillary security standard IEC62351, were chosen from amongst the other SG communications protocols because they had explicitly declared rigid real time standards for data transmission, and a set of QoS promises to ensure the safe operation of the cyber-physical systems that deploy them. Mapping of IEC61850's promises into robustness models allowed for the generation of arguments that adversarial interference of a communication session could drive them to exceed those real time requirements and violate the promises across every implementation of the standard, whereas other standards agnosticism towards defining acceptable limits of operation meant that attacks could only be discovered on an implementation by implementation basis.

Throughout the course of this research project several attacks were discovered within the communication models of IEC61850. The first two attacks were not discovered using the queuing network methodology, but were developed to test the theory that there were attacks within IEC61850 that would manifest in every implementation of the standard. The first of these was a credential intercept attack in the standard's association model was proved using context-free grammar. An adversary could steal legitimate agents credentials due to the standard not enforcing the encryption of this session's communications channel. Without the encryption of the channel it is possible to merge the state machines of two clients to show that there is a race condition that an adversary can exploit to steal legitimate credentials without being detected. The second attack was a demonstration that the lack of restrictions on the multicast communications models, meant that an adversary could create a workfactor amplification attack, which could lead to the violation of availability. The attack was demonstrated on a PIM network topology implementation. These attacks were presented at CRITIS 2016[209].

With the queuing network methodology the author was able to develop the following attacks within the IEC61850 standard:

- Showed that an adversary can de-synchronise a control server from a client node, because the client state machine has no notion that the server has a timeout state. This attack was presented at critis 2017

- Based on the the literature around how the GOOSE anti-replay mechanism would cause more insecurity than it solve, a model of packet injection attacks was developed in the queuing network methodology. With this model the likelihood of these proposed attacks disrupting the QoS promises of IEC61850 were calculated. The model also allowed for various protocol level countermeasures to be tested. This work was presented at ISGT Europe 2018. The packet injection model was further developed into a generalised form.

- Demonstrated that a IEC61850 system can easily be de-synchronized from the layer of the time-synchronisation network that has correct accuracy, due its time synchronisation state machine having no states for handling exceptions. This work was presented at ISGT North America 2019.

5.2 Future Work

This section lays out what work could be done to further build on the author's methodology, and then present possible research that could be done with the methodology.

5.2.1 Improvements of the Methodology

If this project were too be continued there are several additions that could be made to the queuing network formal method. As this is a proof of concept of the approach, the focus hasn't been on developing an implementation with the best computational complexity. To improve the complexity of this approach, research into other data structures, and thus algorithms for their creation and navigation of a network should be done. To ensure that these new implementations are correct the proofs laid out in chapter 3 would have to be redone. Other additions to the methodology could be made, such as:

- The inclusion of other probability distributions for the arrival and processing rates, to model alternative traffic flows.
- Further state transition types, such as packet being dropped, queues dumping their packets, and blocks of queues being either sent at once or broadcast.
- The expansion of the conditional probability functionality switching of arrival and processing rates and the inclusion of priority queues.

5.2.2 Projects Using the Methodology

A project that the queuing network methodology could be used for is developing a formal taxonomy of weaker adversary powers and disruption attacks that can be demonstrated using the queuing network methodology could be created. The first step of this project would be to explore how the adversary's placement on the communication channel affects what they can learn and do on the network. This could be achieved by modelling the adversary and the honest agents at the network level. This avenue would start with the traditional MiTM shown with the DY adversary, before moving on to man on the side, collusion between an honest agent and the adversary, and an honest agent unknowingly communicating directly with an adversary. With the adversary positions modelled the next step would be to model the different ways the DY adversary can be weakened in the queuing network approach. Creating weaker adversaries is important as most of the work on symbolic adversaries has been done to develop stronger to demonstrate a protocols resilience. These are useful when the adversary is trying to obtain a message, but it isn't useful when modelling a protocol's robustness. In robustness models the adversary is trying to stop the completion of a protocol session. Altering the adversary's learning and inference as well as developing a series of capability and timings rules would provide the ground work for the taxonomy. The inference and knowledge models could then be combined with the positioning models to begin to develop a series of attacks. With a taxonomy of attacks a user can input their standard's QoS and be given the likelihood of it being susceptible to that attack. To further develop the adversary in a more theoretical direction, the internal state of the adversary itself could be modelled to show how they respond to certain counter measures. This would move away from the traditional DY adversary that can't react to events made in the protocol sessions. The adversary deploys an ability from their set in order to achieve a goal. Models of the internal state corrupted honest agents could also be developed.

Appendix A

Security Omissions within IEC62351-3

This appendix looks at security promises made by IEC62351-3[181] standard, which is describes how the data packets being sent across a communications network that uses one of IEC60870, IEC61850, or DNP3 SAS will be encrypted using transport layer security TLS 1.2. In addition the security promises made by the standard IEC62351-3 incorporates confidentiality, integrity, and message level authentication to the protocols. The standard does present a list of message types that implementers should omit from encrypting. Those are those that need a real time response, such as SV in IEC61850. IEC62351-6 further elaborabates on the list of types that should be omitted by stating that "*applications using GOOSE and IEC 61850-9-2 and requiring 4ms response times, multicast configurations, and low CPU overhead, encryption is not recommended.*"[183]. IEC62351-3 claims that it will counter MiTM and replay attacks, along with negating eavesdropping. The focus of this appendix is to show that these promises are not be upheld if certain policy omission regarding the certificate authority (CA) trust networks required by the TLS's public key infrastructure aren't rectified.

The violations of the security promises are shown through interrogating IEC62351-3's approach of just declaring the use TLS, message authentication codes(MAC), and CAs whilst it neglecting to comment on CA implementation, certificate validation and revocation policies. As well undermining its security promises, it is shown that these omissions come into clash with the IEC61850-5 quality of service requirement of interoperability of devices[121], because without consistent policy decelerations each vendor can make their own implementations decisions which may clash with each other.

IEC62351-3 overlooks certain key areas that would fulfil its promises. The assumptions made by the specification are that either a CRL or an OCSP algorithms will be deployed alongside the communications network, but both IEC62351-3 and TLS abdicate responsibility on standardising the best practise for the operation of a CA. To uphold its promises IEC62351-3 must specify proper administrative policy on how the CAs are deployed to prevent an attacker being able circumvent the protections provided by TLS.

A.1 Omission in the Public Key Infrastructure Implementation

Shown below are the possible problems that can arise with the major implementations of public key trust architecture, CRL, OCSP, and OCSP stapling. Potential solutions to these problems are suggested. After that, the overview turns to the specific considerations that need to be taken into account for SG public key infrastructure.

A.1.1 Attack vectors of CRL's

Below are potential attack vectors of CRL based trust architecture:

- Public key infrastructure is only as robust as the topology of its trust architecture[193]. The correct choice of trust topology can provide a greater guarantee that a device on the network can authenticate the certificates it receives in a TLS handshake. For example if the trust topology implemented is a hierarchical one then an attacker could mount a denial of service against the promise of authentication, as it would deny a targeted device access to the CA's available on its branch.

- Owners of trust infrastructure need some way of checking that a CA on their network is legitimate, and should have the authority to issue certificates. If the CA cannot be trusted to act honestly when confirming the veracity of a party's certificate, and for them to maintain an accurate CRL, then the promises of both authentication and integrity cannot be upheld.[57]
  Although there haven't been any reported cases where a CA has acted in an intentionally dishonest manner, we can deduce the consequences of not validating their honesty by looking at examples of CA error from recent history[7]. In 2013 Türktrust accidentally marked two customer certificates as CA certificates. Someone then used one these on a network gateway, which allowed them to intercept and decrypt traffic leaving it. In 2012 TrustWave issued an intermediate CA certificate to a customer that then used it to generate end-user certificates, that allowed them to decrypt traffic. In 2011 both Comodo and DigiNotar CAs networks were compromised. In the first instance the attackers issued themselves nine certificates for popular domains, however only one of them was seen deployed in the wild. In the latter incident, the malicious certificates were used to implement a MiTM attack, an attack which IEC62351-3 says it prevents, against Iranian users of Gmail.
  The above cases show that if the CA cannot be trusted to perform its signing function honestly, then it compromises the global trust architecture.

- Certificates used in a trust network should only be accepted at the beginning of a session if it can be shown that it has come from a trusted CA. If a client is unaware of how a CA acts when it receives a certificate signing request, then they are uncertain of the level of trust they can place in any certificates it receives. If anyone can get a certificate for their device without any check that the person has permission to make a certificate signing request, then the promise of authentication cannot be upheld. If there is no validation by the CA of a requester, then an impersonation attack[22] can take place. This is further complicated when the party requesting the CA to sign a certificate is a device rather than an individual. This means that there is no guarantee that a device on the network is sending its data to a legitimate party.

- CRL requests are sent unencrypted across the communications network. This problem could allow an attacker to violate the promises of confidentiality, integrity, and authentication by manipulating or intercepting packets, via a MiTM attack. Due to this weakness in the CRL algorithm, it is possible for an attacker to mount DoS attack against a device. As the CRL response has no boundary on its length[44], an attacker could reply to device's request for an updated CRL with a packet that far exceeds the overhead it can dedicate to the task of processing.

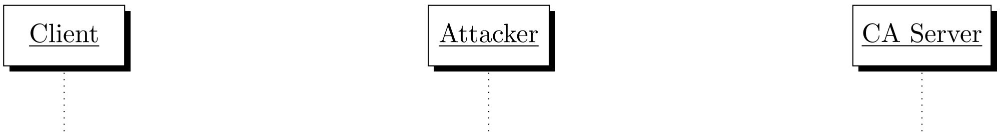

Figure A.1: The attacker sends an array of garbage data for the device to process. Where x and y are the number of bits in the array, and y $\gg$ x

A.1.2 Attack vectors within the OCSP Algorithm

The OCSP algorithm addresses the problem of CRL requests having no limit on packet length, which was stated in section A.1.1. OCSP's have a fixed message complexity, which is indicated by the responses content-length header[162], so an attacker can no longer send a reply that would exceed a devices overhead. However, the OCSP documentation makes no comment on the optimal latency a CA should keep to when replying to a request. Without a policy for what a device should do if it receives no or an invalid certificate a it could be made to accept a revoked or expired certificate by an attacker, which would undermine the promises of integrity and authentication. As the OCSP response is not transmitted over an encrypted channel, like CRL, an attacker can modify the response data packets. If the attacker changes the response to "tryLater" the client doesn't require a signed response, and, depending on the implementation, the algorithm may 'soft fail' and accept the certificate[55].

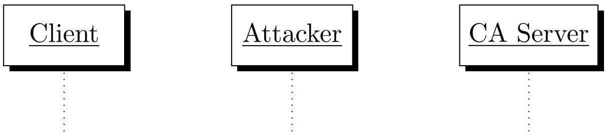

Figure A.2: A session diagram of the 'tryLater' attack.

A.1.3 Attack Vectors within the OCSP Stapling Algorithm

OCSP stapling is designed to allow CA's to reduce the amount of overhead they dedicate to the requests they receive. This is achieved by allowing the certificate holder to query the CA's database itself to get a signed timestamped validation response. The requirement for this process to work is that the binding of the certificate and the CA's validation signature must be immutable, otherwise it then leaves verification in the hands of a potential attacker. This would undermine any promise of authentication. Possible ways of achieving this binding are described in section A.1.4.

Another problem with this validation method is there needs to be a redundancy in the trust architecture if the server is unable to get their certificate signed and timestamp by a CA. If an attacker can prevent the server from making an OCSP request then they have undermined the IEC61850-5 promise of device availability, as no other device should connect with them due to not being able to authenticate their identity.

A.1.4 Potential Solutions and Considerations

To secure the trust architecture of any communications network using IEC62351-3, the protocol would need to go beyond what is written in the referenced public key infrastructure specifications. To improve the security of the CA, the standard would have to declare the best practices expected of any entity issuing certificates on the communication network before allowing access[178].

To prevent the 'soft fail' scenario of the 'tryLater' attack in OCSP, IEC6235-3 would have to make it clear that if a device receives a 'tryLater', revoked, or even no response that it should not continue the session with the other party, as there is no guarantee of authentication.

To make sure that the binding of a certificate and the CA's validation is unalterable in an OCSP stapling trust system, there are two possible was of deploying it. The first method would be to create a dedicated private CA for the grid's communications network. This way an IED can reduce the overhead needed to check that the CA is a valid entity, as if it wasn't the private CA it would just end the session. For this approach to work the private CA would have to develop stringent policies on how it manages its private key(s), as if those are lost then the networks entire trust model would be compromised[90]. It should also be noted that most commercial CA lists used in web browsers are cultivated to be used in as many territories as possible, some of which may decide to use its trust infrastructure to perform MiTM attacks against its own citizens. If one of these states CA's is used on the smart grid trust network, it could be used for similar purposes[173]. The other method is to bind the DNSsec[54] of the communications network with the trust topology. By making a specific tier of the grid's DNS architecture the only CA for the tier immediately below it, a chain of trust can be built between tiers.

It must be noted that only one of the two previous suggestions should be implemented at a time. Using them in combination would allow the private CA to undermine the DNSsec chain of trust as it would allow CA's from any tier be accepted.

A.1.5 Considerations for SG Trust Architecture

There are two challenges faced by any public key infrastructure that is implemented on a SG's communication network. The first is how frequently an IED should receive a copy of the latest CRL[195]. A device is vulnerable to communicating with an adversary when it has a presented an invalid certificate before it receives the new CRL. Most web CAs send out an updated CRL every seven days. This leaves a web browser potentially vulnerable for six days between updates. It is conceivable that the CRL wont be sent until it has reached a significant number of revocations. Having a definitive policy on the time between CRL updates would remove any uncertainty as to when a device will receive the latest CRL, which would reduce the chance that it will accept an invalid certificate.

The second problem that needs to be considered is the overhead an IED can suffer when processing a CRL request. The device must validate the CA's signature, perform databasing tasks every time it receives a CRL, and parse the database to check if the certificate it has received in a handshake is still valid. The device has limited processing power and storage to perform these tasks, and no consideration is made for the latency requirements of the task. Without any QoS requirement imposed on processing a CRL request, the device could be susceptible to an attacker using revoked certificates until the CRL has been processed and parsed. However, IEC62351-3 does make some consideration on the storage of CRL's. It suggests that a CRL stored on an IED should be no larger than 8192 octets. Unfortunately it provides no suggested recourse if a CRL is larger than this. There would be no way of remedying an attack implemented along these vector, without external interference, as the protocol does not allow a checking or inability to access a CRL to end an established session[181].

A.1.6 Downgrade attack

Due to the QoS promise of interoperability of devices taking precedent over security requirements in IEC61850-5, IEC6235-3 cannot guarantee the complete confidentiality of data sent over its network. The specification makes no comment on what happens if there is no agreement on cypher suite during a TLS handshake or CRL response signature algorithm a pair of devices will use in a session. The danger of having no policy for this means that manufacturers could make the judgement for their device to use on no encryption in this instance, which attacker could exploit in an attack. A potential justification for a manufacturer making this decision is it reduces the overhead on the IED's processor, which would make it easier to be compliant with IEC62351. An example of a downgrade type attack was developed against TLS 1.2, as it was shown that a lot of distributions were being implemented with the export grade Diffie-Hellman 512 bit primes as a possible cipher suite. These could be precomputed[3], therefore allowing messages to be compromised in transit. It should be noted that IEC62351-3 wouldn't be susceptible to this attack as 1024 bit key length is the minimum requirement for key exchange.

IEC62351-3 should make explicitly clear what happens to a session if there is no agreement between parties on TLS protocol version or CA response. Whilst this comes into direct clash with the requirements of IEC61850-5, it would guarantee the integrity and confidentiality of the data being transmitted over the SG network. It should also be noted that the standard provides backwards compatibility up to TLS 1.0. It would also strengthen IEC62351-3 security promises to remove this requirement, as these have been shown to be insecure[16].

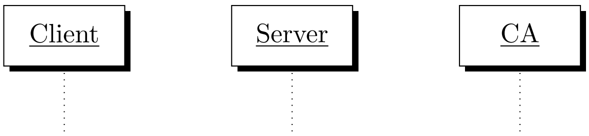

Figure A.3: An example of a session downgrade after the CA server uses an undesirable signing algorithm in its OCSP response.

Appendix B

Threat Modelling Omissions in IEC62351

The following analysis builds upon the work presented in appendix A to further explore the security promises made by IEC61850 SAS standard, and its complementary security standard IEC62351. The analysis shows that the principal focus of the standards is the security of the abstract communications channel, whilst ignoring the rest of the communications infrastructure. The standards declare that the security promises that the messages must maintain throughout their transmission are confidentiality, integrity, message level authentication, availability, and non-repudiation. The key contribution of this appendix is to show that these promises are not upheld if the standards do not comment on the security of the rest of the communication infrastructure. It also points out that the standards fail to provide complete formal model with which any systems operations can be checked against the claimed promises. These omissions arise due to the standard assuming that its threat model is complete, and that the attacks can only occur in isolation. The designers of the standards have used traditional security measures, whilst neglecting to consider the domain specific requirements and the added complexity of distributed systems. The analysis demonstrates that the infrastructure is susceptible to Byzantine faults, a model that describes the resilience of a distributed system when members of the network are compromised, as the standards make no declaration of a lower bound on the number of nodes operating in undesirable states that a network should be able to tolerate. It also looks at attacks against the time synchronisation infrastructure, and communications paths. The analysis includes descriptions of possible solutions to these raised issues.

The authors have chosen to focus on these SAS standards because of their stringent QoS requirements which lends themselves as being a fertile development ground for discovering bespoke threats that only arise in networks with distributed communication topologies. Any solutions to these problems must also meet the QoS standards to ensure the safe operation of the SG cyber-physical system.

B.0.1 Malicious devices on the communication network

As discussed in section 2 one potential attack vector is the injection of false data into the state estimation calculations of an SG systems. Whilst the section presented various models for detecting malicious data, the research usually doesn't comment on how the injected data infiltrates the communications network. One possibility is the adversary compromises a device on the network, or connects their own an unauthorised device. Whilst this threat doesn't undermine the security of a communications channel, it does mean the communications network cannot guarantee that proper authentication and non-repudiation of messages is maintained. Due to the large geographical area that SG communications network will cover, it will be exceeding difficult to visually inspect the network's topology to make sure that there are no unauthorised devices connected. To protect against this threat the only practical solution must come from the communication infrastructure itself.

This attack vector has some similarities to the problems addressed in the research of Byzantine fault tolerance. Research in this field looks at how a distributed communication networks can continue operating effectively when nodes within the system are operating in a misconfigured or malicious manner. These undesirable states do not have to be brought about by malicious interactions, they can happen if a device becomes unsure of its internal state, or desynchronised from the rest of the network. In the SG domain it is important for the communications network be able to resolve these errors promptly, or the undesirable states could lead to damage of the distribution and transmission networks. Byzantine fault tolerance is based on solving the allegorical Byzantine general problem which describes a situation where a group of generals are seiging a city and must come to a consensus about what action to take. It is assumed that a general sends the same message to all other generals, but the process is compromised if a general betrays the group and sends conflicting messages to the others. It has been shown that the group can continue operating so long as two thirds or more of the general are loyal to the groups objective [99]. This model is useful in the SG domain because network operators need a consensus from the nodes so a state estimation of the distribution network can be calculated. There are various algorithms to make sure the distributed networks continue to function in spite of a Byzantine fault. The two common approaches to resolving these kinds of faults are increasing the accountability of nodes on the communications network, or building redundant system for when a node is compromised. The accountability option requires the implementer to keep a database of all the actions made by the nodes on the network. For this method to succeed there would have to be considerations made as to how to make sure the entries in the database weren't modified or deleted, to prevent it becoming another attack vector. An example of this method is the PeerReview system developed by Haeberlen *et al.*[75]. The drawback is the system adds a $O(N^2)$, where N is the number of nodes in the system, to the computational overhead of the network, which may violate the QoS requirements of IEC61850. Attempts have been made to add the promise of accountability to advance metering infrastructure within SG's [104], but there doesn't appear to be any attempts to adapt this system for the rest of SG communications infrastructure.

The other option is building redundant nodes that act as duplicates of nodes in the network topology[196]. When the governing algorithm identifies a faulty node, the network switches to the redundant node to ensure the reliable and accurate operation of the network continue. There doesn't appear to be any attempts to use this method within the SG space.

B.0.2 Clock Synchronisation Attacks

SG's IEDs must have the correct time to ensure reliable operation of their cyber-physical system. Reliable synchronisation of devices and sensors across the network is required so that safety operations can occur in the correct order. IEC61850-7-2 describes how a client server can alert other severs that their time is inaccurate, but doesn't provide an algorithm for how they deduce that. The time synchronisation communication model defers to the SNTP time synchronisation protocol [120] for this functionality. SNTP is a derivative of the NTP protocol, that removes the algorithms that monitor the accuracy of the time servers. This means a network that uses SNTP can only depend on one master clock or collision conditions will be created. However, attacks that have been developed against NTP can also be used against SNTP, alongside unique SNTP vulnerabilities.

Without an accurate time reference the IEDs that monitor and control the distribution network will not function reliably and within the QoS of the standard. An adversary exploiting this attack vector would be able to affect the integrity of any future messages sent from a device and potentially allow them to cause damage to the distribution network. The research into the various attacks against NTP and other time synchronisation standards are described in section 2.2.9.

B.0.3 Communication Path Security

In spite of IEC62351 being written with the manufactures of devices in mind, it makes little comment on the security of the devices themselves. If the physical infrastructure is insecure, then regardless of the channel's security, the QoS and security promises of the standard may not be upheld.

Considerations need to be made for routers, because if an adversary can take control of a router they can manipulate the distances travelled, or delay, or drop messages packets that are sent across the communications network[197]. If an adversary can compromise a router they will be able to undermine the promises of availability, and non-repudiation. The adversary doesn't necessarily need to compromise a router, they may be able to attach their own unauthorised router to the network given the vast geographical area that communications network will have to cover. With a compromised router they could provide false information about potential routes, distance between routers, or create false destinations. These threats could lead to violations of the QoS, as the standards make no comment on how the network should cope with changes in the communications network's topology. There are various different countermeasures against this. Any countermeasure deployed would have to be protocol based, due to the geographical challenges mentioned in section B.0.1. Wan *et al.*[197] propose a reputation based algorithm. The routers check each others messages against a central authority. If a node is sending out disingenuous then its reputation is lowered, until the router is just ignored. This algorithm does rely on having secure communications with an uncorrupted central authority and they don't consider colluding nodes. However, it does have an advantage over similar monitoring algorithms, such as WATCHERS[29], as it looks at the declared settings of a router rather than just its actions. This approach prevents the adversary from trying to get other routers blacklisted so the infrastructure comes to depend on them. An expansion of the reputation method is distribute the job of the central authority amongst the routers on the network[100]. The routers keep a record of the contacts it has made and then distributes them amongst 'witness' routers. These routers compare the properties of those claimed contact with what they have observed, to see if a router is misbehaving.

Bibliography

[1] M. Abadi, B. Blanchet, and C. Fournet. The applied pi calculus: Mobile values, new names, and secure communication. *J. ACM*, 65(1):1:1–1:41, October 2017.

[2] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, CCS '97, pages 36–47, New York, NY, USA, 1997. ACM.

[3] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, A. J. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta, B. VanderSloot, E. Wustrow, S. Zanella-Béguelin, and P. Zimmermann. Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security, CCS 2015*, pages 5–17, New York, NY, USA, 2015. ACM.

[4] A. Akhlaghi, F. Adibnia, and M. H. Shirali-Shahreza. A queue-based analysis for denial of service attacks on voice over ip proxies. In *2008 International Symposium on Telecommunications*, pages 19–24, Aug 2008.

[5] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid. Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions. *Computers & Security*, 74:144 – 166, 2018.

[6] M. Q. Ali and E. Al-Shaer. Probabilistic model checking for AMI intrusion detection. In *IEEE International Conference on Smart Grid Communications, SmartGridComm 2013*, pages 468–473, Oct 2013.

[7] B. Amann, R. Sommer, M. Vallentin, and S. Hall. No Attack Necessary: The Surprising Dynamics of SSL Trust Relationships. In *Proceedings of the 29th Annual Computer Security Applications Conference, ACSAC 2013*, pages 179–188, New York, NY, USA, 2013. ACM.

[8] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. J. Dongarra, J. Du Croz, S. Hammarling, A. Greenbaum, A. McKenney, and D. Sorensen. *LAPACK Users' Guide (Third Ed.).* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.

[9] S. B. Andrade, M. Pignati, G. Dan, M. Paolone, and J. Y. Le Boudec. Undetectable PMU Timing-Attack on Linear State-Estimation by Using Rank-1 Approximation. *IEEE Transactions on Smart Grid*, pages 1–1, 2017.

[10] J. D. Ansilla, N. Vasudevan, J. JayachandraBensam, and J. D. Anunciya. Data security in Smart Grid with hardware implementation against DoS attacks. In *International Conference on Circuits, Power and Computing Technologies, ICCPCT 2015*, pages 1–7, March 2015.

[11] A. Armando, R. Carbone, and L. Compagna. Ltl model checking for security protocols. volume 19, pages 385–396, 07 2007.

[12] J. C. M. Baeten and M. Bravetti. A generic process algebra. *Electronic Notes in Theoretical Computer Science*, 162:65–71, 2006.

[13] J.C.M. Baeten. A brief history of process algebra. *Theoretical Computer Science*, 335(2):131 – 146, 2005. Process Algebra.

[14] C. Baier and J. Katoen. *Principles of Model Checking (Representation and Mind Series)*. The MIT Press, 2008.

[15] A. Baiocco, C. Foglietta, and S. D. Wolthusen. Delay and jitter attacks on hierarchical state estimation. In *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 485–490, Nov 2015.

[16] G. V. Bard. A Challenging but Feasible Blockwise-Adaptive Chosen-Plaintext Attack on SSL. In *Proceedings of the International Conference on Security and Cryptography, SECRYPT 2006, SET'UBAL*, pages 7–10. INSTICC Press, 2006.

[17] S. Barreto, A. Suresh, and J. Y. Le Boudec. Cyber-attack on packet-based time synchronization protocols: The undetectable delay box. In *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pages 1–6, May 2016.

[18] D. A. Basin, J. Dreier, L. Hirschi, S. Radomirovic, R. Sasse, and V. Stettler. Formal analysis of 5g authentication. *CoRR*, abs/1806.10360, 2018.

[19] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology — CRYPTO '98*, pages 26–45, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[20] M. Bellare and P. Rogaway. Code-based game-playing proofs and the security of triple encryption. *Eurocrypt 2006, LNCS*, pages 409–426.

[21] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS '93, pages 62–73, New York, NY, USA, 1993. ACM.

[22] P. Black and R. Layton. Be Careful Who You Trust: Issues with the Public Key Infrastructure. In *Proceedings of the 2014 Fifth Cybercrime and Trustworthy Computing Conference,CTC 2014*, pages 12–21, Washington, DC, USA, 2014. IEEE Computer Society.

[23] B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *Proceedings. 14th IEEE Computer Security Foundations Workshop, 2001.*, pages 82–96, June 2001.

[24] B. Blanchet. Modeling and verifying security protocols with the applied pi calculus and proverif. *Found. Trends Priv. Secur.*, 1(1-2):1–135, October 2016.

[25] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications.* Wiley-Interscience, New York, NY, USA, 1998.

[26] E. Bompard, T. Huang, Y. Wu, and M. Cremenescu. Classification and trend analysis of threats origins to the security of power systems. *International Journal of Electrical Power & Energy Systems*, 50:50 – 64, 2013.

[27] F. Borges, R. A. M. Santos, and F. L. Marquezino. Preserving privacy in a smart grid scenario using quantum mechanics. *Security and Communication Networks*, 8(12):2061–2069, 2015.

[28] J. Bowen and V. Stavridou. Safety-critical systems, formal methods and standards. *Software Engineering Journal*, 8(4):189–209, July 1993.

[29] K. A. Bradley, S. Cheung, N. Puketza, B. Mukherjee, and R. A. Olsson. Detecting disruptive routers: a distributed network monitoring approach. In *Proceedings. 1998 IEEE Symposium on Security and Privacy (Cat. No.98CB36186)*, pages 115–124, May 1998.

[30] D. Brand and P. Zafiropulo. On Communicating Finite-State Machines. *Journal of the ACM*, 30(2):323–342, April 1983.

[31] Sébastien Briais and Uwe Nestmann. A formal semantics for protocol narrations. *Theoretical Computer Science*, 389(3):484 – 511, 2007. Semantic and Logical Foundations of Global Computing.

[32] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Trans. Comput. Syst.*, 8(1):18–36, February 1990.

[33] K. Cairns, C. Hauser, and T. Gamage. Flexible data authentication evaluated for the smart grid. In *IEEE International Conference on Smart Grid Communications, SmartGridComm 2012*, pages 492–497, Oct 2013.

[34] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of the 42Nd IEEE Symposium on Foundations of Computer Science*, FOCS '01, pages 136–, Washington, DC, USA, 2001. IEEE Computer Society.

[35] R. Canetti. Obtaining universally compoable security: Towards the bare bones of trust. In K. Kurosawa, editor, *Advances in Cryptology – ASIACRYPT 2007*, pages 88–112, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[36] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, July 2004.

[37] I. Cervesato, N. A. Durgin, P. D. Lincoln, J. C. Mitchell, and A. Scedrov. A meta-notation for protocol analysis. In *Proceedings of the 12th IEEE Computer Security Foundations Workshop*, pages 55–69, June 1999.

[38] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, Jan 1988.

[39] X. Chen, H. Dinh, and B. Wang. Cascading Failures in Smart Grid - Benefits of Distributed Generation. In *First IEEE International Conference on Smart Grid Communications, SmartGridComm 2010*, pages 73–78, Oct 2010.

[40] V . Cheval and V. Cortier. Timing attacks in security protocols: Symbolic framework and proof techniques. In R. Focardi and A. Myers, editors, *Principles of Security and Trust*, pages 280–299, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[41] S. Chong, J. Guttman, A. Datta, A. C. Myers, B. C. Pierce, P. Schaumont, T. Sherwood, and N. Zeldovich. Report on the NSF workshop on formal methods for security. *CoRR*, abs/1608.00678, 2016.

[42] D. D. Clark. *Designing an Internet*. Information Policy. MIT Press, 2018.

[43] K. Cohn-Gordon, C. Cremers, and L. Garratt. On post-compromise security. In *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, pages 164–178, June 2016.

[44] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. RFC5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Technical report, Internet Engineering Task Force, 2008.

[45] R. Corin and A. Saptawijaya. A logic for constraint-based security protocol analysis. In *2006 IEEE Symposium on Security and Privacy (S P'06)*, pages 14 pp.–168, May 2006.

[46] C. Cremers, M. Dehnel-Wild, and K. Milner. Secure authentication in the grid: A formal analysis of dnp3: Sav5. In S. N. Foley, D. Gollmann, and E. Snekkenes, editors, *ESORICS*, volume 10492, pages 389–407. Springer, 2017.

[47] C. Cremers, M. Horvat, J. Hoyland, S. Scott, and T. van der Merwe. A comprehensive symbolic analysis of tls 1.3. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 1773–1788, New York, NY, USA, 2017. ACM.

[48] C. Cremers and S. Mauw. *Operational Semantics and Verification of Security Protocols.* Information Security and Cryptography. Springer, 2012.

[49] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir. Taming the 800 pound gorilla: The rise and decline of ntp ddos attacks. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, IMC '14, pages 435–448, New York, NY, USA, 2014. ACM.

[50] A. Datta, A. Derek, J. C. Mitchell, and A. Roy. Protocol composition logic (pcl). *Electronic Notes in Theoretical Computer Science*, 172:311 – 358, 2007. Computation, Meaning, and Logic: Articles dedicated to Gordon Plotkin.

[51] G. Dán, K. S. Lui, R. Tabassum, Q. Zhu, and K. Nahrstedt. SELINDA: A secure, scalable and light-weight data collection protocol for smart grids. In *IEEE International Conference on Smart Grid Communication, SmartGridComm 2013*, pages 480–485, Oct 2013.

[52] D. Dolev and A. Yao. On the Security of Public Key Protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, Mar 1983.

[53] S. East, J. Butts, M. Papa, and S. Shenoi. A taxonomy of attacks on the dnp3 protocol. In C. Palmer and S. Shenoi, editors, *Critical Infrastructure Protection III*, pages 67–81, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[54] D. Eastlake. RFC2535 - Domain Name System Security Extensions. Technical report, Internet Engineering Task Force, 1999.

[55] W. El-Hajj. The most recent SSL security attacks: origins, implementation, evaluation, and suggested countermeasures. In *Security and Communication Networks*, volume 5, pages 113–124, 2012.

[56] A. Elgargouri, R. Virrankoski, and M. Elmusrati. IEC 61850 Based Smart Grid Security. In *IEEE International Conference on Industrial Technology, ICIT 2015*, pages 2461–2465, March 2015.

[57] C. Ellison and B. Schneier. Ten risks of PKI: What you're not being told about public key infrastructure. In *Computer Security Journal*, volume 16, pages 1–7, 2000.

[58] N. Evans and S. Schneider. Analysing time dependent security properties in csp using pvs. In F. Cuppens, Y. Deswarte, D. Gollmann, and M. Waidner, editors, *Computer Security - ESORICS 2000*, pages 222–237, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[59] S. Even and O. Goldreich. On the security of multi-party ping-pong protocols. In *24th Annual Symposium on Foundations of Computer Science (sfcs 1983)*, pages 34–39, Nov 1983.

[60] F. J. T. Fabrega, J. C. Herzog, and J. D. Guttman. Strand spaces: why is a security protocol correct? In *Proceedings. 1998 IEEE Symposium on Security and Privacy (Cat. No.98CB36186)*, pages 160–171, May 1998.

[61] A. Farraj, E. Hammad, and D. Kundur. On using distributed control schemes to mitigate switching attacks in smart grids. In *IEEE 28th Canadian Conference on Electrical and Computer Engineering, CCECE 2015*, pages 1578–1582, May 2015.

[62] A. K. Farraj, E. M. Hammad, A. A. Daoud, and D. Kundur. A game-theoretic control approach to mitigate cyber switching attacks in Smart Grid systems. In *IEEE International Conference on Smart Grid Communications, SmartGridComm 2014*, pages 958–963, Nov 2014.

[63] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, January 1968.

[64] S. Fries, H. J. Hof, and M. Seewald. Enhancing IEC 62351 to Improve Security for Energy Automation in Smart Grid Environments. In *Fifth International Conference on Internet and Web Applications and Services, ICIW 2010*, pages 135–142, May 2010.

[65] S. Fuloria, R. Anderson, F. Alvarez, and K. McGrath. Key management for substations: Symmetric keys, public keys or no keys? In *IEEE/PES Power Systems Conference and Exposition, PSCE 2011*, pages 1–6, March 2011.

[66] C. A. Furia, B. Meyer, and S. Velder. Loop invariants: Analysis, classification, and examples. *ACM Comput. Surv.*, 46(3):34:1–34:51, January 2014.

[67] G. Gaderer, A. Treytl, and T. Sauter. Security aspects for IEEE 1588 based clock synchronization protocols. In *IEEE International Workshop on Factory Communication Systems, WFCS 2006, Torino, Italy*, pages 247–250. Citeseer, 2006.

[68] S. Ghosh and M. H. Ali. Minimization of adverse effects of time delay in smart power grid. In *IEEE PES Innovative Smart Grid Technologies Conference, ISGT 2014*, pages 1–5, Feb 2014.

[69] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270 – 299, 1984.

[70] S. Goldwasser, S. Micali, and P. Tong. Why and how to establish a private code on a public network. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 134–144, Nov 1982.

[71] L. Gong, R. Needham, and R. Yahalom. Reasoning about belief in cryptographic protocols. In *Proceedings. 1990 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 234–248, May 1990.

[72] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris. *Fundamentals of Queueing Theory.* Wiley-Interscience, New York, NY, USA, 4th edition, 2008.

[73] A. Gul and S. D. Wolthusen. Measurement re-ordering attacks on power system state estimation. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6, Sep. 2017.

[74] A. Gul and S. D. Wolthusen. *A Review on Attacks and Their Countermeasures in Power System State Estimation*, pages 9–28. Springer International Publishing, Cham, 2018.

[75] A. Haeberlen, P. Kouznetsov, and P. Druschel. PeerReview: Practical Accountability for Distributed Systems. *SIGOPS Oper. Syst. Rev.*, 41(6):175–188, October 2007.

[76] M. Harchol-Balter. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action.* Cambridge University Press, New York, NY, USA, 1st edition, 2013.

[77] M. El Hariri, T. A. Youssef, and O. A. Mohammed. On the implementation of the iec 61850 standard: Will different manufacturer devices behave similarly under identical conditions? *Electronics*, 5(4):201–213, 2016. http://www.mdpi.com/2079-9292/5/4/85.

[78] X. He, M. Pun, and C. C. J. Kuo. Secure and efficient cryptosystem for smart grid using homomorphic encryption. In *IEEE PES Innovative Smart Grid Technologies, ISGT 2012*, pages 1–8, Jan 2012.

[79] C. A. R. Hoare. *Communicating Sequential Processes.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1985.

[80] J. Hoyos, M. Dehus, and T. X. Brown. Exploiting the GOOSE Protocol: A Practical Attack on Cyber-Infrastructure. In *IEEE Globecom Workshops, 2012*, pages 1508–1513, Dec 2012.

[81] W. Hurst, N. Shone, and Q. Monnet. Predicting the Effects of DDoS Attacks on a Network of Critical Infrastructures. In *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, 2015*, pages 1697–1702, Oct 2015.

[82] Dragos Inc. Crashoverride: Analysis of the threat to electric grid operations. 2017.

[83] E. Itkin and A. Wool. A security analysis and revised security extension for the precision time protocol. In *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication, ISPCS 2016*, pages 1–6, Sept 2016.

[84] P. Judge and M. Ammar. Security issues and solutions in multicast content distribution: a survey. *IEEE Network*, 17(1):30–36, Jan 2003.

[85] A. Kamil and G. Lowe. Analysing tls in the strand spaces model. Technical report, 2008.

[86] P. Kammas, T. Komninos, and Y. C. Stamatiou. A queuing theory based model for studying intrusion evolution and elimination in computer networks. In *2008 The Fourth International Conference on Information Assurance and Security*, pages 167–171, Sep. 2008.

[87] S. Karnouskos. Stuxnet worm impact on industrial cyber-physical system security. In *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, pages 4490–4494, Nov 2011.

[88] J. Katz and Y. Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[89] D. G. Kendall. Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain. *Ann. Math. Statist.*, 24(3):338–354, 09 1953.

[90] S. Kent. Evaluating certification authority security. In *Aerospace Conference, 1998 IEEE*, volume 4, pages 319–327 vol.4, Mar 1998.

[91] Auguste Kerckhoffs. La cryptographic militaire. *Journal des sciences militaires*, pages 5–38, 1883.

[92] D. Kesdogan. Evaluation of anonymity providing techniques using queuing theory. In *Proceedings LCN 2001. 26th Annual IEEE Conference on Local Computer Networks*, pages 316–322, Nov 2001.

[93] J. Kim and L. Tong. On Topology Attack of a Smart Grid: Undetectable Attacks and Countermeasures. *IEEE Journal on Selected Areas in Communications*, 31(7):1294–1305, July 2013.

[94] J. Kim, L. Tong, and R. J. Thomas. Data Framing Attack on State Estimation. *IEEE Journal on Selected Areas in Communications*, 32(7):1460–1470, July 2014.

[95] C. Konstantinou, M. Maniatakos, F. Saqib, S. Hu, J. Plusquellic, and Y. Jin. Cyber-Physical Systems: A Security Perspective. In *20th IEEE European Test Symposium, ETS 2015*, pages 1–8, May 2015.

[96] O. Kosut, L. Jia, R. J. Thomas, and L. Tong. Malicious Data Attacks on the Smart Grid. *IEEE Transactions on Smart Grid*, 2(4):645–658, Dec 2011.

[97] C. Kriger, S. Behardien, and J. Retonda. A detailed analysis of the goose message structure in an iec 61850 standard-based substation automation system. 8:708–721, 10 2013.

[98] N. Kush, E. Ahmed, M. Branagan, and E. Foo. Poisoned GOOSE: Exploiting the GOOSE Protocol. In *Proceedings of the Twelfth Australasian Information Security Conference - Volume 149, AISC 2014*, pages 17–22, Darlinghurst, Australia, Australia, 2014. Australian Computer Society, Inc.

[99] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982.

[100] Q. Li and G. Cao. Mitigating routing misbehavior in disruption tolerant networks. *IEEE Transactions on Information Forensics and Security*, 7(2):664–675, April 2012.

[101] Q. Li, C. Ross, J. Yang, J. Di, J. C. Balda, and H. A. Mantooth. The effects of flooding attacks on time-critical communications in the smart grid. In *IEEE PES Innovative Smart Grid Technologies Conference, ISGT 2015*, pages 1–5, Feb 2015.

[102] H. Lin, A. Slagell, Z. Kalbarczyk, P. W. Sauer, and R. K. Iyer. Semantic Security Analysis of SCADA Networks to Detect Malicious Control Commands in Power Grids. In *Proceedings of the First ACM Workshop on Smart Energy Grid Security, SEGS 2013*, pages 29–34, New York, NY, USA, 2013. ACM.

[103] F. Liu, T. Xie, Y. Feng, and D. Feng. On the Security of PPPoE Network. *Security and Communication Networks*, 5(10):1159–1168, Oct 2012.

[104] J. Liu, Y. Xiao, and J. Gao. Accountability in Smart Grids. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pages 1166–1170, Jan 2011.

[105] S. Liu, B. Chen, D. Kundur, T. Zourntos, and K. Butler-Purry. Progressive switching attacks for instigating cascading failures in smart grid. In *IEEE Power Energy Society General Meeting, 2013*, pages 1–5, July 2013.

[106] S. Liu, D. Kundur, T. Zourntos, and K. L. Butler-Purry. Coordinated variable structure switching attack in the presence of model error and state estimation. In *IEEE Third International Conference on Smart Grid Communications, SmartGridComm 2012*, pages 318–323, Nov 2012.

[107] S. Liu, X. P. Liu, and A. E. Saddik. Denial-of-Service (dos) attacks on load frequency control in smart grids. In *IEEE PES Innovative Smart Grid Technologies Conference, ISGT 2013*, pages 1–6, Feb 2013.

[108] S. Liu, S. Mashayekh, D. Kundur, T. Zourntos, and K. Butler-Purry. A Framework for Modeling Cyber-Physical Switching Attacks in Smart Grid. *IEEE Transactions on Emerging Topics in Computing*, 1(2):273–285, Dec 2013.

[109] Y. Liu, P. Ning, and M. K. Reiter. False Data Injection Attacks Against State Estimation in Electric Power Grids. In *Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS 2009*, pages 21–32, New York, NY, USA, 2009. ACM.

[110] J. Liyan, R. J. Thomas, and L. Tong. On the Nonlinearity Effects on Malicious Data Attack on Power System. In *Power and Energy Society general meeting 2012*, July 2012.

[111] G. Lowe. Breaking and fixing the needham-schroeder public-key protocol using fdr. In T. Margaria and B. Steffen, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 147–166, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[112] G. Lowe. Casper: a compiler for the analysis of security protocols. In *Proceedings 10th Computer Security Foundations Workshop*, pages 18–30, June 1997.

[113] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen. EPPA: An Efficient and Privacy-Preserving Aggregation Scheme for Secure Smart Grid Communications. *IEEE Transactions on Parallel and Distributed Systems*, 23(9):1621–1631, Sept 2012.

[114] Z. Lu, W. Wang, and C. Wang. From jammer to gambler: Modeling and detection of jamming attacks against time-critical traffic. In *Proceedings IEEE INFOCOM 2011*, pages 1871–1879, April 2011.

[115] G. Maciá-Fernández, J. E. Díaz-Verdejo, P. García-Teodoro, and F. de Toro-Negro. Lordas: A low-rate dos attack against application servers. In J. Lopez and Bernhard M. B. M. Hämmerli, editors, *Critical Information Infrastructures Security*, pages 197–209, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[116] L. A. Maglaras, J. Jiang, and T. Cruz. Integrated OCSVM mechanism for intrusion detection in SCADA systems. *Electronics Letters*, 50(25):1935–1936, 2014.

[117] B. Mahdad and K. Srairi. Blackout risk prevention in a smart grid based flexible optimal strategy using Grey Wolf-pattern search algorithms. *Energy Conversion and Management*, 98:411 – 429, 2015.

[118] A. Malhotra, I. E. Cohen, E. Brakke, and S. Goldberg. Attacking the network time protocol. In *23rd Annual Network and Distributed System Security Symposium*, 2016.

[119] A. Malhotra and S. Goldberg. Attacking NTP's Authenticated Broadcast Mode. *SIGCOMM Comput. Commun. Rev.*, 46(2):12–17, May 2016.

[120] TC 57 Power Systems Management and Associated Information Exchange. Communication Networks and Systems for Power Utility Automation - Part 7-2: Basic Information and Communication Structure - Abstract Communication Service Interface. IEC standard 61850-7-2. Technical report, International Electrotechnical Commission, 2010.

[121] TC 57 Power Systems Management and Associated Information Exchange. Communication Networks and Systems for Power Utility Automation - Part 5: Communication Requirements for Functions and Device Models. IEC standard 61850-5. Technical report, International Electrotechnical Commission, 2013.

[122] W. Mao and C. Boyd. Towards formal analysis of security protocols. In *[1993] Proceedings Computer Security Foundations Workshop VI*, pages 147–158, June 1993.

[123] J. Martin, J. Burbank, W. Kasch, and D. L. Mills. Network Time Protocol Version 4: Protocol and Algorithms Specification. RFC 5905, June 2010.

[124] P. McDaniel and S. McLaughlin. Security and Privacy Challenges in the Smart Grid. *IEEE Security Privacy*, 7(3):75–77, May 2009.

[125] T. R. McEvoy and S. D. Wolthusen. A formal adversary capability model for scada environments. In *Proceedings of the 5th International Conference on Critical Information Infrastructures Security*, CRITIS'10, pages 93–103, Berlin, Heidelberg, 2011. Springer-Verlag.

[126] S. McLaughlin, P. McDaniel, and W. Aiello. Protecting Consumer Privacy from Electric Load Monitoring. In *18th ACM Conference on Computer and Communications Security, CCS 2011*, pages 87–98, New York, NY, USA, 2011. ACM.

[127] D. L. Mills. *Computer Network Time Synchronization: The Network Time Protocol on Earth and in Space, Second Edition.* CRC Press, Inc., Boca Raton, FL, USA, 2nd edition, 2010.

[128] R. Milner. *A Calculus of Communicating Systems.* Springer-Verlag, Berlin, Heidelberg, 1982.

[129] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, i. *Information and Computation*, 100(1):1 – 40, 1992.

[130] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, ii. *Information and Computation*, 100(1):41 – 77, 1992.

[131] R. Mitchell and I. Chen. A Survey of Intrusion Detection Techniques for Cyber-physical Systems. *ACM Comput. Surv.*, 46(4):55:1–55:29, March 2014.

[132] T. Mizrahi. A game theoretic analysis of delay attacks against time synchronization protocols. In *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, 2012*, pages 1–6, Sept 2012.

[133] Tal Mizrahi. Security Requirements of Time Protocols in Packet Switched Networks. RFC 7384, October 2014.

[134] Y. Mo, T. H. J. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli. Physical Security of a Smart Grid Infrastructure. *Proceedings of the IEEE*, 100(1):195–209, Jan 2012.

[135] A. H. Mohsenian-Rad and A. Leon-Garcia. Distributed Internet-Based Load Altering Attacks Against Smart Power Grids. *IEEE Transactions on Smart Grid*, 2(4):667–674, Dec 2011.

[136] A. H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia. Autonomous Demand-Side Management Based on Game-Theoretic Energy Consumption Scheduling for the Future Smart Grid. *IEEE Transactions on Smart Grid*, 1(3):320–331, Dec 2010.

[137] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin. Private Memoirs of a Smart Meter. In *2Nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building, BuildSys 2010*, pages 61–66, New York, NY, USA, 2010. ACM.

[138] J. J. Moré. The levenberg-marquardt algorithm: Implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, pages 105–116, Berlin, Heidelberg, 1978. Springer Berlin Heidelberg.

[139] B. Moussa, M. Debbabi, and C. Assi. A detection and mitigation model for PTP delay attack in a smart grid substation. In *IEEE International Conference on Smart Grid Communications, SmartGridComm 2015*, pages 497–502, Nov 2015.

[140] F. G. Mármol, C. Sorge, O. Ugus, and G. M. Pérez. Do not snoop my habits: preserving privacy in the smart grid. *IEEE Communications Magazine*, 50(5):166–172, May 2012.

[141] R. M. Needham and M. D. Schroeder. Using encryption for authentication in large networks of computers. *Commun. ACM*, 21(12):993–999, December 1978.

[142] R. De Nicola. A gentle introduction to process algebras ? 2013.

[143] M. Ohta. Overload protection in a sip signaling network. In *International Conference on Internet Surveillance and Protection (ICISP'06)*, pages 11–11, Aug 2006.

[144] C. Osorio and M. Bierlaire. An analytic finite capacity queueing network model capturing the propagation of congestion and blocking. *European Journal of Operational Research*, 196(3):996 – 1007, 2009.

[145] M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor. Sparse Attack Construction and State Estimation in the Smart Grid: Centralized and Distributed Models. *IEEE Journal on Selected Areas in Communications*, 31(7):1306–1318, July 2013.

[146] S. Parthasarathy and D. Kundur. Bloom filter based intrusion detection for smart grid SCADA. In *25th IEEE Canadian Conference on Electrical and Computer Engineering, CCECE 2012*, pages 1–6, April 2012.

[147] R. Patel, B. Borisaniya, A. Patel, D. Patel, M. Rajarajan, and A. Zisman. Comparative analysis of formal model checking tools for security protocol verification. In N. Meghanathan, S. Boumerdassi, N. Chaki, and D. Nagamalai, editors, *Recent Trends in Network Security and Applications*, pages 152–163, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[148] A. K. Pathan. *Securing Cyber-Physical Systems*. CRC Press, Inc., Boca Raton, FL, USA, 2015.

[149] L. C. Paulson. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, 5(3):363–397, Sep 1989.

[150] J. C. L. Pimentel and R. Monroy. Formal support to security protocol development: A survey. *Computación y Sistemas*, 12(1):89–108, 2008.

[151] E. Poll, J. D. Ruiter, and A. Schubert. Protocol State Machines and Session Languages: Specification, implementation, and Security Flaws. In *IEEE Security and Privacy Workshops, SPW 2015*, pages 125–133, May 2015.

[152] U. Premaratne, J. Samarabandu, T. Sidhu, R. Beresh, and J. C. Tan. Security Analysis and Auditing of IEC61850-Based Automated Substations. *IEEE Transactions on Power Delivery*, 25(4):2346–2355, Oct 2010.

[153] U. K. Premaratne, J. Samarabandu, T. S. Sidhu, R. Beresh, and J. C. Tan. An Intrusion Detection System for IEC61850 Automated Substations. *IEEE Transactions on Power Delivery*, 25(4):2376–2383, Oct 2010.

[154] M. T. A. Rashid, S. Yussof, Y. Yusoff, and R. Ismail. A review of Security Attacks on IEC61850 Substation Automation System Network. In *International Conference on Information Technology and Multimedia, ICIMU 2014*, pages 5–10, Nov 2014.

[155] T. Rid. *Cyber War Will Not Take Place.* Oxford University Press, Inc., New York, NY, USA, 2013.

[156] S. Rinaldi, D. D. Giustina, P. Ferrari, A. Flammini, and E. Sisinni. Time synchronization over heterogeneous network for smart grid application: Design and characterization of a real case. *Ad Hoc Networks*, 50:41 – 57, 2016.

[157] R. J. Robles, M. Balitanas, and T. Kim. *Security Encryption Schemes for Internet SCADA: Comparison of the Solutions*, pages 19–27. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[158] K. C. Ruland and J. Sassmannshausen. *Non-repudiation Services for the MMS Protocol of IEC 61850*, pages 70–85. Springer International Publishing, Cham, 2015.

[159] M. D. Ryan and B. Smyth. Applied pi calculus. In *Formal Models and Techniques for Analyzing Security Protocols, chapter 6. IOS*, 2010.

[160] P. Ryan and S. Schneider. *The Modelling and Analysis of Security Protocols: The CSP Approach.* Addison-Wesley Professional, first edition, 2000.

[161] D. E. Sanger and N. Perlroth. *U.S. Escalates Online Attacks on Russia's Power Grid*, 02-10-2019 (accessed 15-06-2019). `https://web.archive.org/web/20191001145309/https://www.nytimes.com/2019/06/15/us/politics/trump-cyber-russia-grid.html`.

[162] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. RFC6960 - X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. Technical report, Internet Engineering Task Force, 2013.

[163] L. Sassaman, M. L. Patterson, S. Bratus, and M. E. Locasto. Security Applications of Formal Language Theory. *IEEE Systems Journal*, 7(3):489–500, Sept 2013.

[164] M. Savi, C. Rottondi, and G. Verticale. Evaluation of the Precision-Privacy Tradeoff of Data Perturbation for Smart Metering. *IEEE Transactions on Smart Grid*, 6(5):2409–2416, Sept 2015.

[165] N. Saxena, B. J. Choi, and R. Lu. Authentication and authorization scheme for various user roles and devices in smart grid. *IEEE Transactions on Information Forensics and Security*, 11(5):907–921, May 2016.

[166] B. Schmidt, S. Meier, C. Cremers, and D. Basin. Automated analysis of diffie-hellman protocols and advanced security properties. In *Proceedings of the 2012 IEEE 25th Computer Security Foundations Symposium*, CSF '12, pages 78–94, Washington, DC, USA, 2012. IEEE Computer Society.

[167] S. Schneider and A. Sidiropoulos. Csp and anonymity. In *Proceedings of the 4th European Symposium on Research in Computer Security: Computer Security*, ESORICS '96, pages 198–218, London, UK, UK, 1996. Springer-Verlag.

[168] S. Shintre, V. Gligor, and J. Barros. Optimal strategies for side-channel leakage in fcfs packet schedulers. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 2515–2519, June 2015.

[169] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004.

[170] N. Sichwart, A. Eltom, and G. Kobet. Transformer load tap changer control using iec 61850 goose messaging. In *2013 IEEE Power Energy Society General Meeting*, pages 1–5, July 2013.

[171] F. Skopik. Security is not enough! on privacy challenges in smart grids. *Int. J. Smart Grid Clean Energy*, 1(1):7–14, 2012.

[172] I. Softley. Hackers. United Kingdom: Universal Studios., 1995.

[173] C. Soghoian and S. Stamm. *Certified Lies: Detecting and Defeating Government Interception Attacks against SSL (Short Paper)*, pages 250–259. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[174] S. Sridhar and G. Manimaran. Data integrity attacks and their impacts on SCADA control system. In *IEEE PES General Meeting*, pages 1–6, July 2010.

[175] S. Sridhar and G. Manimaran. Data integrity attack and its impacts on voltage control loop in power grid. In *IEEE Power and Energy Society General Meeting 2011*, pages 1–6, July 2011.

[176] A. Srivastava, T. Morris, T. Ernster, C. Vellaithurai, S. Pan, and U. Adhikari. Modeling Cyber-Physical Vulnerability of the Smart Grid With Incomplete Information. *IEEE Transactions on Smart Grid*, 4(1):235–244, March 2013.

[177] M. Strobel, N. Wiedermann, and C. Eckert. Novel weaknesses in iec 62351 protected smart grid control systems. In *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 266–270, Nov 2016.

[178] Symantec. Prioritizing Trust: Certificate Authority Best Practices. `https://www.symantec.com/content/en/us/enterprise/white_papers/b-prioritizing-trust-ca-best-practices_WP.en-us.pdf`. Published: 2012, Accessed: 20-07-2016.

[179] Cisco systems. Ip multicast technology overview. `https://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/ip_multicast/White_papers/mcst_ovr.html`. Published: 29-09-2001, Accessed:09-06-2016.

[180] TC 57 Power systems management and associated information exchange. Power systems management and associated information exchange, data and communication security - part 1: Communication network and system security - introduction to security issues. IEC standard 62351-1. Technical report, International Electrotechnical Commission, 2007.

[181] TC 57 Power systems management and associated information exchange. Power systems management and associated information exchange, data and communication security - Part 3: Communication network and system security - Profiles including TCP/IP. IEC standard 62351-3. Technical report, International Electrotechnical Commission, 2007.

[182] TC 57 Power systems management and associated information exchange. Power systems management and associated information exchange, data and communication security - part 4: Profiles including MMS. IEC standard 62351-4. Technical report, International Electrotechnical Commission, 2007.

[183] TC 57 Power systems management and associated information exchange. Power systems management and associated information exchange, data and communication security - Part 6: Security for IEC 61850 . IEC standard 62351-6. Technical report, International Electrotechnical Commission, 2007.

[184] TC 57 Power systems management and associated information exchange. Power systems management and associated information exchange, data and communication security - part 7: Network and system management(NSM) data object models. IEC standard 62351-7. Technical report, International Electrotechnical Commission, 2017.

[185] P. Syverson. Towards a strand semantics for authentication logic. *Electronic Notes in Theoretical Computer Science*, 20:143 – 157, 1999. MFPS XV, Mathematical Foundations of Progamming Semantics, Fifteenth Conference.

[186] R. Tawde, A. Nivangune, and M. Sankhe. Cyber security in smart grid SCADA automation systems. In *International Conference on Innovations in Information, Embedded and Communication Systems, ICIIECS 2015*, pages 1–5, March 2015.

[187] TC 57 - Power Systems Management and Associated Information Exchange. Power Systems Management and Associated Information Exchange, Data and Communication Security - Part 6: Security for IEC 61850. IEC standard 62351-6. Technical report, International Electrotechnical Commission, 2007.

[188] TC 57: Power Systems Management and Associated Information Exchange. Telecontrol equipment and systems - Part 5: Transmission protocols - Section 5: Basic application functions. Technical report, International Electrotechnical Commission, 1995.

[189] TC 57: Power Systems Management and Associated Information Exchange. Communication Networks and Systems for Power Utility Automation - Part 7-2: Basic Information and Communication Structure. Technical report, International Electrotechnical Commission, 2010.

[190] F. L. Tiplea, C. Enea, and C. V. Bîrjoveanu. Decidability and complexity results for security protocols.

[191] J. L. Tsai and N. W. Lo. Secure Anonymous Key Distribution Scheme for Smart Grid. *IEEE Transactions on Smart Grid*, 7(2):906–914, March 2016.

[192] J. Tsang and K. Beznosov. *A Security Analysis of the Precise Time Protocol (Short Paper)*, pages 50–59. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[193] Z. E. Uahhabi and H. E. Bakkali. A Comparative Study of PKI Trust Models. In *2014 International Conference on Next Generation Networks and Services, NGNS*, pages 255–261, May 2014.

[194] M. Ullmann and M. Vögeler. Delay attacks - Implication on NTP and PTP time synchronization. In *International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, 2009*, pages 1–6, Oct 2009.

[195] S. Vandeven. Digital certificate revocation. `https://www.sans.org/reading-room/whitepapers/certificates/digital-certificate-revocation-35292`. Published:15-07-2014, Accessed: 16-07-2016.

[196] G. S. Veronese, M. Correia, A. N. Bessani, L. C. Lung, and P. Verissimo. Efficient Byzantine Fault-Tolerance. *IEEE Transactions on Computers*, 62(1):16–30, Jan 2013.

[197] T. Wan, E. Kranakis, and P. C. van Oorschot. *S-RIP: A Secure Distance Vector Routing Protocol*, pages 103–119. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[198] X. Wan, Zhang Li, and Z. Fan. A sip dos flooding attack defense mechanism based on priority class queue. In *2010 IEEE International Conference on Wireless Communications, Networking and Information Security*, pages 428–431, June 2010.

[199] B. Wang, M. Wang, and S. Zhang. *Research on Secure Message Transmission of Smart Substation Based on GCM Algorithm*, pages 533–538. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[200] Z. Wang and S. S. Tseng. Impact evaluation of ddos attacks on dns cache server using queuing model. *KSII Transactions on Internet and Information Systems*, 7:895–909, 04 2013.

[201] T. L. Ward. Grid Cryptographic Simulation: A Simulator to Evaluate the Scalability of the X.509 Standard in the Smart Grid. Technical Report TR2013-742, Dartmouth College, Computer Science, Hanover, NH, September 2013.

[202] D. Wei, Y. Lu, M. Jafari, P. M. Skare, and K. Rohde. Protecting Smart Grid Automation Systems Against Cyberattacks. *IEEE Transactions on Smart Grid*, 2(4):782–795, Dec 2011.

[203] J. Wei and D. Kundur. A flocking-based model for dos-resilient communication routing in smart grid. In *IEEE Global Communications Conference, GLOBECOM 2012*, pages 3519–3524, Dec 2012.

[204] Jorden Whitefield, L. Chen, F. Kargl, A. Paverd, S. Schneider, H. Treharne, and S. Wesemeyer. Formal analysis of v2x revocation protocols. In G. Livraga and C. Mitchell, editors, *Security and Trust Management*, pages 147–163, Cham, 2017. Springer International Publishing.

[205] D. E. Whitehead, K. Owens, D. Gammel, and J. Smith. Ukraine cyber-induced power outage: Analysis and practical mitigation strategies. In *2017 70th Annual Conference for Protective Relay Engineers (CPRE)*, pages 1–8, April 2017.

[206] D. K. N. Wood and D. R. E. Harang. Grammatical Inference and Language Frameworks for LANGSEC. In *IEEE Security and Privacy Workshops, SPW 2015*, pages 88–98, May 2015.

[207] J. G. Wright and S. Wolthusen. Time accuracy de-synchronisation attacks against iec 60870-5-104 and iec 61850 protocols. In *2019 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5, Feb 2019.

[208] J. G. Wright and S. D. Wolthusen. Limitations of iec62351-3's public key management. In *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, pages 1–6, Nov 2016.

[209] J. G. Wright and S. D. Wolthusen. Access control and availability vulnerabilities in the iso/iec 61850 substation automation protocol. In G. Havarneanu, R. Setola, H. Nassopoulos, and S. Wolthusen, editors, *Critical Information Infrastructures Security*, pages 239–251, Cham, 2017. Springer International Publishing.

[210] J. G. Wright and S. D. Wolthusen. De-synchronisation attack modelling in real-time protocols using queue networks: Attacking the iso/iec 61850 substation automation protocol. In G. D'Agostino and A. Scala, editors, *Critical Information Infrastructures Security*, pages 131–143, Cham, 2018. Springer International Publishing.

[211] J. G. Wright and S. D. Wolthusen. Stealthy injection attacks against iec61850's goose messaging service. In *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6, Oct 2018.

[212] Xiaodong Xu, Xiao Guo, and Shirui Zhu. A queuing analysis for low-rate dos attacks against application servers. In *2010 IEEE International Conference on Wireless Communications, Networking and Information Security*, pages 500–504, June 2010.

[213] L. Xie, Y. Mo, and B. Sinopoli. False Data Injection Attacks in Electricity Markets. In *First IEEE International Conference on Smart Grid Communications, SmartGridComm 2010*, pages 226–231, Oct 2010.

[214] J. Yan, Y. Zhu, H. He, and Y. Sun. Revealing temporal features of attacks against smart grid. In *IEEE PES Innovative Smart Grid Technologies ISGT 2013*, pages 1–6, Feb 2013.

[215] L. Yang, H. Xue, and F. Li. Privacy-preserving data sharing in Smart Grid systems. In *IEEE International Conference on Smart Grid Communications, SmartGridComm 2014*, pages 878–883, Nov 2014.

[216] Y. Yang, T. Littler, S. Sezer, K. McLaughlin, and H. F. Wang. Impact of Cyber-Security Issues on Smart Grid. In *2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies, ISGT Europe 2011*, pages 1–7, Dec 2011.

[217] S. L. P. Yasakethu, J. Jiang, and A. Graziano. Intelligent risk detection and analysis tools for critical infrastructure protection. In *Eurocon 2013*, pages 52–59, July 2013.

[218] S. P. Yin. Research on the Application of IBE in IEC61850 Substation Automation System. In *Mechanical and Electronics Engineering III*, volume 130 of *Applied Mechanics and Materials*, pages 2805–2808. Trans Tech Publications, 1 2012.

[219] Z. Yu and W. Chin. Blind false data injection attack using pca approximation method in smart grid. *IEEE Transactions on Smart Grid*, 6(3):1219–1226, May 2015.

[220] P. Zhang, O. Elkeelany, and L. McDaniel. An implementation of secured Smart Grid Ethernet communications using AES. In *Proceedings of the IEEE SoutheastCon, SoutheastCon 2010*, pages 394–397, March 2010.

[221] F. Zhao, Y. Hanatani, Y. Komano, B. Smyth, S. Ito, and T. Kambayashi. Secure authenticated key exchange with revocation for smart grid. In *IEEE PES Innovative Smart Grid Technologies, ISGT 2012*, pages 1–8, Jan 2012.