QUANTIFYING THE EFFECTS OF UNCERTAINTY TO MANAGE CYBER-SECURITY RISK AND ENABLE ADAPTIVITY IN POWER GRID WIDE AREA MONITORING AND CONTROL APPLICATIONS

By

YUJUE WANG

A dissertation submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

WASHINGTON STATE UNIVERSITY
School of Electrical Engineering and Computer Science

MAY 2016

To the Faculty of Washington State University:

The members of the Committee appointed to examine the dissertation of YUJUE WANG find it satisfactory and recommend that it be accepted.

Carl Hauser, Ph.D., Chair

Dave Bakken, Ph.D.

Ananth Kalyanaraman, Ph.D.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere appreciation to my adviser, Dr. Carl Hauser. His superb expertise, enduring support, and diligent mentoring throughout my PhD study have been inculcating me and will inspire me along the road ahead. I always feel lucky to be his student. Without his guidance or help, I could not even imagine to accomplish this long journey.

I would like to thank Dr. Dave Bakken and Dr. Thoshitha Gamage for their generous help and support during my study in WSU. I highly appreciate their enlightening ideas. I am grateful to all other GridStat members, especially Dave Anderson, Kelsey Carins, and Chin-Wei Chang. I really miss the days I spent with them.

It is my great honor to have Dr. Ananth Kalyanaraman in my Ph.D. committee. I appreciate his time and effort before and during my dissertation work.

Eventually I would like to offer my most heartfelt appreciations to my parents for their love throughout my life. Without their unconditional support, I could not chase my dream so far away!

My research has been supported by Department of Energy under Award DE-OE0000097 (TCIPG).

QUANTIFYING THE EFFECTS OF UNCERTAINTY TO MANAGE CYBER-SECURITY RISK AND ENABLE ADAPTIVITY IN POWER GRID WIDE AREA MONITORING AND CONTROL APPLICATIONS

Abstract

by Yujue Wang, Ph.D.
Washington State University
May 2016

Chair: Carl Hauser

The smooth operation of the power grid is based on the effective Wide Area Monitoring and Control systems, which is supposed to provide reliable and secure communication of data. Due to the complexity of the system and inaccuracy of modeling, uncertainty is unavoidable in such systems. So it is of great interest to characterize and quantify the uncertainty properly, which is significant to the functionality of power grid.

Trust, as a subjective and expressive concept connoting one party's (the trustor's) reliance on and belief in the performance of another party (the trustee), is modeled to help administrators (trustors) of WAMC systems evaluate the trustworthiness of data sources

(trustees), which is essentially a measurement of uncertainty of this system. Both evidence based methods and data based methods are developed to evaluate trustworthiness and describe uncertainty respectively.

By modeling both aleatory and epistemic uncertainty with subjective logic and probability distributions respectively, a framework quantifying uncertainty is proposed. Quantification of the uncertainties can greatly help the system administrators to select the most fitting security implementation to achieve both security and QoS with a certain confidence. Based on the quantification framework, an adaptive security mechanism is prototyped, which can adjust the security scheme online according to dynamic requirements and environmental changes, to make the best ongoing trade-off between security assurance and QoS.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

Wide-Area Monitoring and Control (WAMC) systems based on synchrophasor data streams are becoming more and more significant to the operation of the smart power grid. Reliable and secure communication, and higher Quality of Service (QoS) (very low latency, high availability, etc.) of data are critical to the success of WAMC systems. Uncertainty about randomness and inaccuracy of modeling, which is inherent to these systems, should be explicitly quantified and appropriately taken into account when making decisions pertaining to the reliability and security of the power grid instead of ignoring the impact of uncertainty.

In order to perceive an instance of uncertainty in WAMC system, we examine the communication between synchrophasors (PMUs) and phasor data concentrators (PDCs) to analyze potential security vulnerabilities present at the transport layer, and investigate the advantages and disadvantages of both the TCP and UDP protocols respectively with an emphasis on security issues. Demonstrations of attacks related to these security vulnerabilities are shown in lab environment; and underlying mechanisms are analyzed to determine the capabilities attackers to succeed with them. Through this, we set up a typical example of uncertainties, which could affect the quality of data, in practical power grid data communication.

One way to characterize the uncertainty of the system comprehensively is *trust assessment*. Trust, a *subjective and expressive concept* connoting one party's (the trustor's) reliance on and belief in the performance of another party (the trustee), is modeled to help

administrators (trustors) of WAMC systems evaluate the trustworthiness of data sources (trustees). Bayesian methodology is a proper method to infer trust degree as an unknown parameter based on uncertain evidence. More specifically, trustworthiness of data sources can be evaluated by calculating the discrepancy between trustor's expectation and behavior of the trustees.

Intrinsic uncertainty makes it difficult for system administrators to decide upon a particular security scheme to secure communications while satisfying the QoS requirements in various WAMC systems. We model different types of uncertainties, epistemic and aleatory, with subjective logic and probability distributions respectively. Quantification of the uncertainties can greatly help the system administrators to select the most fitting security implementation to achieve both security and QoS with a certain confidence. Moreover, an adaptive security mechanism that can adjust the security scheme online, according to dynamic requirements and environmental changes, to make the best ongoing trade-off between security assurance and QoS.

Particularly, contributions can be listed as follows:

- Security implications of two common transport layer protocols (TCP and UDP) applied to synchrophasor data communication between PMUs and PDCs is comprehensively analyzed. It is shown both with theoretical analysis and experiments that transport Layer Security, on top of TCP protecting the payload data, can mitigate false data injection attacks but still leaves open the possibility of Denial of Service attacks if packets can be sniffed or sequence numbers can be inferred. This conveys an illustration

that uncertainty is deep-rooted in power grid communication systems such as WAMC systems and we need to measure and handle uncertainties appropriately.

- A systematic way is established to deal with uncertainty related to trust and proposed an initial theoretical framework in critical infrastructure control systems. Based on Bayesian decision theory, uncertainty can be incorporated into trust related decisions.

- A definition of trust is proposed which is suitable for applications in critical infrastructures. Five significant factors of trust are extracted: *uncertainty*, *belief*, *consistency*, *context* and *subjectivity*. Based on these factors, a mathematical framework is built, which is capable of adapting to situations with different trust policies and prior information.

- A framework is provided to investigate QoS and security of power grid communication system in an integral way by quantifying different types of uncertainties associate with them. Probability distributions are used to capture the characteristics of delay for security overhead and apply subjective logic to describe the stakeholders' opinions on security coverage of security schemes. Monte Carlo method is employed to issue a unified view of the overall uncertainty with these two aspects.This framework can help WAMC system administrators to choose the most fitting security scheme based on the characteristics of their systems.

- We propose adaptive cyber security scheme taking QoS requirements of WAMC applications into account. Compared with traditional fixed cyber security countermeasure,

adaptive cyber security scheme can dynamically optimize security coverage and reduce the probability of QoS violation due to the cyber security countermeasure.

CHAPTER 2. SECURITY IMPLICATIONS OF TRANSPORT LAYER PROTOCOLS IN POWER GRID SYNCHROPHASOR DATA COMMUNICATION

In practice, uncertainty is inherent to complex systems such as smart power grid. In this chapter, we develop some techniques to explore security vulnerabilities and show pragmatic examples of uncertainty related to transport layer protocols in power grid synchrophasor data communication. These uncertainties could affect the security of synchrophasor data transfer, which is the primary target of our exploration.

Emerging synchronized phasor measurement is expected to greatly improve the wide-area situational awareness of power grid operators. For years, the lack of adequate wide-area situational awareness has plagued operators when it comes to devising appropriate responses to impending grid conditions. Synchrophasors, typically installed at the substation level, capture time-synchronized (through a common time reference such as GPS) current and voltage phasor measurements from electrical buses and lines with sample rates up to 60Hz [53]. Thus, measurements from a large geographical area, when aggregated at one location with low network latency and high data availability, can facilitate near real-time Wide-Area Monitoring and Control (WAMC) of the power system.

A typical WAMC architecture, similar to the one depicted in Fig. 2.1, is made up of four major components [12]: synchrophasors (PMUs); phasor data concentrators (PDCs); a communication network; and power applications. The components together implement four

layers of the system: data acquisition, data management, data services, and applications. The focus of our research is the data transfer between the data acquisition and data management layers, specifically the effect of transport layer protocol choice on the security of the critical communications between these layers.

WAMC systems are real-time systems [13]. Decisions related to power grid operations need to be made based on accurate real-time data within a small time window. Slight manipulation of timestamps, added delay or partial loss of data, or modest data tampering all could lead to errors or delays in decision-making thereby affecting reliable and efficient operations of the power grid.

IEEE C37.118.2-2011 [1] is the up-to-date standard for power system synchrophasor data transfer between PMUs and PDCs. However, this standard neither offers any restrictions on the choice of transport layer protocols nor provides a precise description of security mechanisms that should be used. This implementation flexibility has two effects:

- Functionally, the choice of transport layer protocol can greatly influence the data transfer performance as different transport layer protocols usually have very divergent mechanisms and characteristics, and are fit for different use cases;
- Security frameworks and implementations must be appropriate for the chosen transport layer protocol. Different security frameworks provide different levels of security for synchrophasor data transfer.

Thus, the selection of a transport layer protocol not only affects the performance of data transfer but also has implications for security. The scope of our work is to analyze the

potential *security* vulnerabilities of *transport layer protocols* for the communication between synchrophasors and phasor data concentrators.

2.1 Related Work

The integrity and the availability of WAMC data is of great concern. False data injection attacks against WAMC data that are undetectable by state estimator bad-data detectors based on residue checks were proposed by Liu in [50], [51]. Extensive research has been conducted on choosing false data to have certain effects on power state estimators for both SCADA data [19], [75], [61] and synchrophasor data [45], [11]. However, these papers described attackers as modifying the values reported by meters without addressing specific mechanisms that might be used to make such changes. The work reported here is motivated by the observation that modifying the data reported by meters while in transit is a potential approach to carrying out false data injection attacks and would be attractive from an attacker's perspective because it obviates the need to physically access or modify meters.

Since much of the data communication for the smart grid is predicated on the use of Internet protocols, the transport layer protocols of interest are primarily TCP and UDP. The current work builds on previous work on attacking these in the general Internet setting. For most of the published attacks on TCP, inferring or guessing the correct sequence number is the essential first step to launching a successful attack. Morris proposed an off-path attack by guessing TCP sequence numbers in [58]. A desynchronization attack against TCP was

proposed in [38]. Harris in [32] offered a quite comprehensive analysis of attacks against TCP. Qian designed two techniques, which can make use of side-channel information revealed by firewalls [65] or Linux host packet counters [66] to infer the sequence number of an existing TCP connection very swiftly. General security related research on transport layer protocols, not specifically related to synchrophasor data streams, mainly addresses TCP.

Transport layer protocols performance characteristics in synchrophasor applications were addressed in [72]. An analysis of cyber security for smart grid communications can be found in [83]. Current literature, however, lacks a comprehensive analysis of the security implications of the two main transport layer protocols (TCP and UDP) with respect to WAMC applications where data communication patterns are far more predictable, in timing and quantity, than for general Internet applications.

As we observed, there are two gaps in current research related to cyber security of synchrophasor data transfer in WAMC systems. 1) There is much theoretical work discussing false data injection attacks at the system level. However, practical cyber aspect of mounting these attacks is usually ignored. 2) Cyber security analysis usually focuses on the mechanisms at the application level. Here we analyze the security concerns raised by the underlying data transport layer since attackers may be able to exploit vulnerabilities of those mechanisms to conduct coordinated attacks to inject false data or cause denial-of-service (DoS) [18]. The goal is to fill these gaps with a thorough and rigorous analysis of the security implications of transport layer protocols used in synchrophasor data transfer.

2.2 Preliminaries

In the Open Systems Interconnection (OSI) model of data communications, the *Transport layer* provides communication services for application processes. The transport layer is under the session layer and above the network layer. The transport layer facilitates services such as connection-oriented communication, reliability, flow control, and multiplexing to the underlying network layer.

The connection-oriented Transmission Control Protocol (TCP) and the connectionless User Datagram Protocol (UDP) are the two most commonly used transport layer protocols in the Internet protocol family suite. With respect to synchrophasor data transfer, each has advantages and disadvantages.

2.2.1 TCP for Synchrophasor Data Transfer

The Transmission Control Protocol (TCP) provides reliable, ordered, and error-checked delivery of a stream of octets between *processes*. *Flow control* and *congestion control* mechanisms in TCP protect end hosts and the network from overloads by data senders. The ordered delivery and reliability of TCP are achieved by stateful connections, sequence numbers, acknowledgments, and retransmissions. However, ordered delivery and retransmissions for reliability affect the latency with which individual messages are delivered [28]. Furthermore, TCP's 3-way handshake, used to establish the initial connection state, and the use of acknowledgments also limit the achievable performance of the protocol. Sequence numbers

also introduce a potential point of vulnerability to the system, which is one of the main security issues investigated here.

In the context of synchrophasor communication, most of the network performance related parameters could be chosen to reduce the need for TCP flow control and congestion control mechanisms, relative to what happens in general-purpose networks. This observation leads to the consideration of UDP as a transport layer protocol.

2.2.2 UDP for Synchrophasor Data Transfer

The user datagram protocol (UDP) has a simple transport service model requiring only minimal mechanism as part of the protocol. UDP is datagram oriented and allows datagrams to be dropped or reordered. There are no retransmission, flow, or congestion control mechanisms. UDP's simple checksums protect against data integrity errors due to network errors, but are insufficient to protect against deliberate data manipulations.

UDP has several advantages in deterministic network environments. Since there isn't any connection establishment, time-out retransmission, or data acknowledgment, UDP may often transfer data with lower latency than TCP. The lack of connection management results in a smaller per-datagram overhead, which makes UDP more efficient than TCP. UDP can also be used for multicast and broadcast communications, which are not possible using TCP.

2.3 Security Implications of Transport Layer Protocols in Synchrophasor Data Transfer

While general security concerns about TCP and UDP are widely investigated, the synchrophasor data transfer environment reveals some new vulnerabilities and makes it easier to exploit some previously known vulnerabilities.

We focus on security concerns that could give rise to DoS attacks or false data injection attacks against PDCs. A malicious attacker aiming to launch DoS or false data injection attacks might have zero or more of the following capabilities:

- **A0**: The attacker is able to commit IP spoofing.
- **A1**: The attacker knows the PDC's IP address and port number and a PMU's IP address and port number. Then, the attacker can inject packets with the same four tuple of (*source IP address, source port number, destination IP address, destination port number*) as legitimate PMU data packets. We use the term *header-tuple* to explicitly denote these four tuples in the remaining of this chapter.
- **A2**: The attacker can sniff traffic on the connection between a PMU and a PDC.
- **A3**: For a TCP connection, the attacker can infer current sequence numbers with the available side channel information [65], [66].
- **A4**: The attacker can obtain time references as accurate as time references used by synchrophasors.

These capabilities are not mutually exclusive. For example, if an attacker can sniff a segment in the communication, he can also easily derive the header-tuple.

In a given environment, attackers might launch different attacks and cause different levels of disturbances to the decision making procedure with different combinations of these capabilities available to them. Reasonable attackers would choose different attack strategies to maximize the possible damage.

2.3.1 Security Concerns in Plain TCP

TCP provides stateful and connection-oriented communication. In order to correctly order arriving data and to identify duplicate segments, TCP uses sequence numbers and acknowledgement sequence numbers to maintain the state of a connection. It has long been known that using random Initial Sequence Numbers (ISNs) is essential. A thorough introduction to TCP sequence number and ACK mechanisms is in RFC 793 [36]. Over the years that TCP has been in use, too-predictable (i.e., not sufficiently random) ISNs have been a critical vulnerability that enabled many successful attacks.

With respect to applications of synchrophasor data transfer, several observations are applicable to our security analysis: TCP connections for these applications are long-lived; inferring the IP address and port number of the PDC is not very difficult and they are even public knowledge to stakeholders of the WAMC system; data packets following the IEEE C37.118-2011 Standard[1] have a fixed format, so segment lengths for data transfer in a specific connection are invariant; furthermore, the PMUs are tightly synchronized with

external time references (usually GPS) of high precision, so that the time intervals between any two sequential data segments sent from the PMU within a TCP connection are precisely uniform if there is no network congestion. This makes the guessing or inference of TCP sequence numbers easier since invariant packet length and uniform time intervals force the sequence numbers to increase at a constant rate.

These observations combined with general security vulnerabilities of TCP bring forth some specific security concerns.

Connection Reset Attack

The first attack considered is a blind attack scenario with the goal of issuing a successful connection reset, in which the attacker is assumed to have the header-tuple for a connection but no capability beyond random guessing to acquire the current sequence number. Are synchrophasor data streams, given their predictable nature, more susceptible to this kind of attack than ordinary TCP connections?

We denote the constant length of synchrophasor payloads of any TCP segments by l, the maximum sequence number by w, and the set of all possible sequence numbers by $W = \{0, 1, 2, \ldots, w-1\}$. The PMU sends one data packet every t seconds. The probability that an attacker can successfully send a TCP reset segment with the header-tuple and a randomly guessed sequence number that makes this segment acceptable to the PDC is $p_{general} = \frac{1}{w}$. In the standard TCP header, $w = 2^{32}$, so $p_{general}$ is very small and it won't increase no matter how many random trials the attacker conducts.

In our specific scenario, however, an intelligent attacker can guarantee the success of a

connection reset by exploiting the evenly incremental sequence number if the connection can live long enough, although he doesn't know at which step he attains his goal. The attacker's strategy can be described as follows:

1. At time 0, the attacker makes a random guess $a \in W$, but he doesn't know whether he has succeeded in resetting the connection;

2. At time t, the attacker makes another random guess $b \in (W - \{(a + l)\%w\})$;

3. At time $2t$, the attacker makes a random guess $c \in (W - \{(a + 2l)\%w, (b + l)\%w\})$;

4. This procedure goes on and finally the attacker arrives at a search space $|W_{nt}| = 1, W_{nt} = \{\omega\}$. The attacker finally takes ω as the sequence number of a reset packet and finishes this reset attack.

Analysis: The number of packets needed for a session of this attack is w, which usually is $2^{32} - 1 = 4,294,967,295$. In order to avoid detection by routers or a firewall, the attacker should not send out packets at a very high rate. Even if his sending rate is 10,000 packets per second, it will take about 5 days to finish one session of attack. So the probability of success with this type of reset attack is still vanishingly low, even though the sequence number changes at a constant known rate.

Pace tracking TCP Injection

In the previous section we saw that an attacker who was forced to guess the current sequence number, without additional information, faced a daunting task even given the predictable nature of synchrophasor data streams. This relatively optimistic result does not

carry forward to situations where the attacker can gain information about current sequence numbers. General TCP injection techniques are described in [31]. Pace tracking attackers punctiliously follow the data-sending pace of PMUs. The overall idea is to exploit TCP's tolerance of retransmitted segments of TCP to feed forged data segments to the PDC applications. True segments from the PMU will be dropped (by the TCP implementation of operation system) silently if the PDC always receives the forged segments earlier than the corresponding true ones. An attacker with the capabilities covering $\{A0, A2\}$ (on-path attack) or $\{A0, A1, A3\}$ (off-path attack) could quickly succeed in this kind of attack.

Due to the evenly incrementing sequence number over time, if the attacker can sniff a true segment once or make correct inference of the sequence number once, he can keep track of the increment of sequence numbers over time. The only extra requirement necessary to succeed the attack is that the attacker also has an accurate external time reference. The detailed procedure is illustrated in Fig. 2.2:

1. The attacker obtains the correct header-tuple;

2. At time t_0, the attacker succeeds in obtaining the sequence number s_0 of a segment sent by the PMU at a time t_s, (t_s ought to be smaller than but very close to t_0) in the connection;

3. For each time $t_0 + n \times t - \epsilon$, $n = 1, 2, \ldots$, the attacker sends out a fake-headered packet with header-tuple and the sequence numbers as $s_0 + n \times l$. If $\epsilon > t_0 - t_s + (OWD_a -$

OWD_p)[1], the attacker can successfully inject his payload data and the PDC will ACK the falsely injected segment with the victim PMU;

4. Since the PDC considers the true segment as a retransmission, the victim PMU would get a duplicated ACK. It is the only by-product of this attack.

Extensive Attacker

Unlike pace tracking attackers, extensive attackers send out many segments in a short time interval to commit an attack. As Fig. 2.3 shows, in this case, the attacker still has to get the sequence number but may not rely on accurate external time references. Instead, he has a local inaccurate time reference. We adopt our notations from Section 2.3.1. Without external accurate time references, the attacker doesn't have the capability to precisely track the PMU's sending pace. In order to raise the probability of injecting bad data successfully, the attacker sends $2m$ fake segments with aligned sequence numbers simultaneously in one attacking action. Here, $2m$ is the attacking window. The attacking procedure is as follows:

1. The attacker gains the correct header-tuple;

2. At time t_0, the attacker obtains the sequence number s_0 of a segment sent by the PMU at a time t_s, (t_s ought to be smaller than but very close to t_0) in the connection;

3. For each time $t_0 + n \times t - \epsilon$, $n = 1, 2, \ldots$, the attacker sends out $2m$ fake-headered

[1] OWD_a is the one way delay from attacker's access point to the PDC, OWD_p is the one way delay from the PMU to the victim PDC

segments with sequence numbers $s_0 + n \times l - m \times l$, $s_0 + (n+1) \times l - m \times l$, ..., $s_0 + n \times l + m \times l$.

A larger attacking window increases the probability of success, but it is not necessary to make it too large as the window size of acceptable sequence numbers of a TCP connection is limited. Moreover, a larger attack window leads to more duplicate ACKs, which makes the attack more detectable.

2.3.2 Security Concerns in TLS

The Transport Layer Security (TLS) protocol is a modernized version of the older Security Sockets Layer (SSL protocol) [21]. TLS is widely used in the Internet for interacting with web services when security and privacy are important. Would using TLS for synchrophasor communications prevent the attacks described above for plain TCP?

TLS provides communication security between the endpoints of TCP connections. With the help of certificate mechanisms, the endpoints of a connection utilizing TLS employ asymmetric cryptography to authenticate one another, and exchange a symmetric session key for encrypting data. This provides data confidentiality and Message Authentication Codes (MAC) for message integrity. A detailed introduction to TLS can be found in [21].

TLS encrypts the application data (payload of TCP) of TCP communications. Only the payload of TCP is encrypted by TLS. The header of TCP is *not* protected. So TCP connections with TLS are potentially still vulnerable to attacks related to TCP sequence numbers introduced in Section 2.3.1.

TLS does not allow independent decryption of individual records as the integrity check depends on the proceeding data: if record N is not received, then the integrity check on record $N + 1$ will be based on the wrong sequence number and thus will fail.

Moreover, known attacks on TLS such as [3] only lead to plaintext recovery. However, an idea similar to the extensive attack introduced in Section 2.3.1 can cause DoS using TLS. Once the attacker obtains a fresh TCP sequence number by sniffing or inference based on side channel information, he repeats the same procedure introduced in Section 2.3.1 but with a random payload. When any one of the injected segments is accepted by the application of PDC, the TLS connection will break and the PDC will face a data blackout until the TCP connection and TLS session are re-established.

2.3.3 Security Concerns in Plain UDP

Unlike TCP, the datagram-oriented UDP contains no mechanisms that protect against data injection. If the synchrophasor data communication is implemented with plain UDP, without any encryption or authentication procedures, false data injection attacks are trivial. The attacker only has to know the PDC's IP address and the port number used for data communication, and gain an access point to send his UDP packets. The PDC will not be able to distinguish good data sent by the PMU from injected "bad data" based on pure cyber knowledge, though the application could detect the existence of the attack by noting that it was receiving multiple PMU messages conveying the same timestamp.

If the attacker knows the PDC's IP address and the port number used for data com-

munication, it can also perform a buffer flood attack, a kind of well-known denial-of-service (DoS) attack. Here, the attacker sends a large number of UDP packets to the victim PDC's IP address and port number. Since the buffer for a single UDP port is fixed, if the attacker's sending rate is much larger than the PDC's consuming rate, the large number of packets sent by the attacker can overwhelm the PDC's buffer causing the victim PMU's data packets to be discarded.

2.3.4 Security Concerns in Datagram TLS

Datagram Transport Layer Security (DTLS) is a modified version of TLS that runs on top of UDP. DTLS allows datagram-based applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. A more detailed introduction to DTLS can be found in [55].

The overall design philosophy of DTLS is "bang for the buck" [55]. Designers of DTLS made the least possible change to TLS in order to reuse the well analyzed scheme of TLS and its implementations and infrastructures. Unlike for TCP, applications using UDP may observe lost, out-of-order, or duplicate packets. On one hand, DTLS preserves these characteristics of UDP; but on the other hand, in order to adjust to the characteristics of UDP, DTLS makes two changes:

- Different from TLS, DTLS independently encrypts each packet, rather than using a stream cipher, so missing packets will not affect the decryption of subsequent segments;

- TLS records include a MAC which guarantees the record integrity, and the MAC input includes a record sequence number which verifies that no record has been lost, duplicated, or reordered. In TLS, this sequence number is implicit as it increases by one for each segment, but for DTLS, this sequence number is explicit and not encrypted.

To probe the security concerns of DTLS, we need to understand DTLS's packet validation procedure related to the explicit sequence numbers. Sequence number verification is performed using a sliding window. The receiver maintains a bit map indicating received and correctly-validated packets within the sliding window. If a newly received packet falls within the window and is new, or if the packet is to the right of the window, then the receiver proceeds to MAC verification. If the MAC validation fails, the receiver discards the received packet as invalid. The receive window is updated only if the MAC verification succeeds.

If the attacker can sniff packet(s) sent to a PDC, he can easily obtain the sequence number as the explicit sequence number is not part of the encrypted data. Then the attacker can track the pace of the sliding window and send large amount of packets with in-window sequence numbers aiming to overwhelm the buffer, but the validation logic boosts the requirements for the attacker to commit a buffer flood attack since the validation of duplicates and MAC authentication are very fast operations. The OS can remove invalid packets from the buffer at a faster rate. Thus, attackers attempting to overwhelm the buffer of DTLS need to send packets with in-window sequence numbers much more quickly than attackers described in 2.3.3.

2.4 Experiments and Demonstrations

We illustrate these security implications introduced in Section 2.3 with results from several experiments. These experiments deliver a more intuitive view of the security implications analyzed above.

2.4.1 Experimental Setup

The connected PMU used in our experimental setup sends out synchrophasor data at 30Hz following the format specified in C37.118-2011 (listed in Table 2.1).

With the help of Scapy[2], an interactive packet manipulation program, we can forge and decode packets for many protocols, send them on the wire, capture them, and match requests and replies. Particularly, we use Scapy to sniff packets and send arbitrarily forged packets.

We installed Wireshark on the victim PDC. It allows the user to see all traffic visible on that interface and reveals all of the information in the Ethernet header, IP header and UDP/TCP header in a very organized way.

In order to attain generality for our results, we didn't employ OpenPDC or any other specific PDC software. Instead, we developed software that emulates the data consumption behavior of PDCs and exposes a detailed view of the timing that would be observed by

[2]http://www.secdev.org/projects/scapy/

Table 2.1: Synchrophasor Data Frame Format Used in the Experimental Setup

Offset (Bytes)	Position (Byte Offset)	Description
2	0	Preamble
2	2	Size: size of frame (42)
2	4	ID of this PMU
4	6	Second of Century
4	10	Time Flag (1 Byte), Fraction of Second (3 Bytes)
2	14	Status
4	16	Float: Voltage 1 Magnitude
4	20	Float: Voltage 1 Phase Angle
4	24	Float: Current 1 Magnitude
4	28	Float: Current 1 Phase Angle
4	32	Float: Power Frequency
4	36	Float: Delta Power Frequency (Unused)
2	40	CRC 16

PDC software. Primarily, it receives data packets from the PMU, prints out the data and forwards the packets to other computers. Our PMU and PDC are placed within the same LAN. Another computer in the same LAN is configured to mimic the behavior of attackers.

2.4.2 Attacks on Plain UDP

In these experiments, the PMU sends synchrophasor data packets to the PDC using UDP as transport layer protocol. The PDC's processing procedure involves checking the CRC of packets, extracting and printing the 10 variables of each packet and forward them to another entity. The attacker sends UDP packets of the same length as the true packets to the same port of the PDC. Payload data sent by the attacker is set to all 0s. Fig. 2.5 shows the different ratios of lost packets with different attacker's sending rates. When the attacker sends more than 1000 packets/second, packet loss begins to occur. When the sending rate is 40000 packets/second, more than 70% of true packets get dropped because of the buffer flood attack.

Fig. 2.6 presents different ratios of false data packets when the attacker injects false data to the PDC with plain UDP. We flag the PMU ID of the false data as 1 and PMU ID of true packet as 0, so we can easily calculate the ratio of false data. If the attacker injects 400 false data packets per second, more than 90% of the packets obtained by the PDC are false data packets injected by the attacker. The victim PDC can detect this attack but cannot pick out the true data.

2.4.3 Attacks Against TCP

For these experiments, the synchrophasor data communication between the PMU and our PDC is configured to run on top of TCP. The attacker read the computer's system time using library time in Python 2.7.5.

Fig. 2.7 shows the effects of a pace tracking TCP injection attack. The attacker sniffs a true TCP segment transferring from the PMU to the PDC at time 0 and then begins to do the pace tracking attack. The attacker injects TCP segments at the PMU's sending rate, which is 30Hz. We measure the ratio of false data segments among all data segments received in last one second. The victim PDC is spoofed by completely false data for about 1 second. It is expected that if the attacker has access to a more accurate time reference, he can compromise the victim PDC completely by injecting false data for longer time as he can follow the data sending pace of the PMU more closely. For WAMC applications which usually have very high QoS requirements, even missing 1s of data is detrimental [33].

Fig. 2.8 shows the effects of the extensive attack described in Section 2.3.1 with attacking window size of 4, 8 and 12 respectively. The data shows that the larger attacking windows increase the time span of successful complete false data injection.

Fig. 2.9 is a snapshot of Wireshark indicating the success of a sequence number attack against TLS. The synchrophasor data communication between victim PDC and victim PMU is protected by OpenSSL 1.0.1, which is a common implementation of TLS. When the attacker succeeds in injecting a TCP segment with ill-formatted payload data, the true

TCP segment with the same sequence number as the injected false segment sent by victim PMU is considered as a retransmission and then discarded by the operating system. Since the OpenSSL implementation relies on a stream cipher, the lost segment affects all of the following segments. The connection gets reset by the PDC. The communication stalls.

Table 2.2: Types and Requirements of Possible Attacks Against Transport Layer Protocols

	Type	Least Requirements
UDP Buffer Flood	DoS	{A0, A1}
UDP Injection	False Data Injection	{A0, A1}
TCP Pace Tracking	False Data Injection	{A0, A2, A4} or {A0, A1, A3, A4}
TCP Extensive	False Data Injection	{A0, A2} or {A0, A1, A3}
TLS Injection	DoS	{A0, A2} or {A0, A1, A3}

2.5 IPsec

TLS and DTLS operate at the boundary between Transport and Application layers in the layered communication model. Given the concerns investigated above, alternatives for achieving security using other layers should also be investigated. IPsec is a protocol suite for securing communications that operates at the network layer [44]. IPsec is commonly used in creating virtual private networks (VPNs). Compared with TLS/DTLS, IPsec appears to be an attractive choice to protect synchrophasor data communications. Work is under way

to appropriately configure IPsec for use utility control systems: IEEE Standard P2030.102.1 – Standard for Interoperability of Internet Protocol Security (IPsec) Utilized within Utility Control Systems is being developed. Through sound deployment of IPsec, the vulnerabilities brought by transport layer protocols, either UDP or TCP, could be mitigated. While it is not our purpose in this chapter to fully evaluate the tradeoffs in choosing between IPsec and transport-level security mechanisms, there are a number of drawbacks to using IPsec caused by its relative complexity and the complexity of its deployment [76].

- Deployment of IPsec requires special support of routers [24], which increases the cost and brings more possible threats into the system.

- IPsec leaves too much flexibility for implementors. The implementors not only have flexibility in choosing encryption algorithms but also have to determine operation modes (AH vs ESP, Tunnel mode vs Transport mode and IKE vs. manual keys). This causes interoperability problems as implementors may understand the specification differently; given the same options, different implementors may make different choices [73].

- There is no existing way to analyze the security of different configuration options systematically [41]. IPsec is well beyond the level of complexity that can be analyzed or properly implemented with current methodologies [26] and implementors may choose flawed configurations. For example, Encryption-only configurations are fatally insecure [62] and successful attacks against all MAC-then-Encrypt configurations can be launched efficiently in practice[20].

2.6 Discussion

Based on the analysis above, the requirements for committing a successful attack against transport layer protocols in synchrophasor data communication is depicted in Table 2.2.

As discussed in Section 2.3, in a synchrophasor data transfer system running on top of plain TCP, it is almost impossible for an attacker to derive the correct sequence number using only random guessing even though connections are long-lived and operate at a constant rate and predictable rate. However, attacks against plain TCP and TCP with TLS are feasible for attackers who can get a current sequence number. Furthermore, unlike for general applications of TCP, in a synchrophasor data transferring system the attacker can continue injecting false data successfully for a relatively long time based on a single correct sequence number. Thus, operators of synchrophasor data networks using plain TCP or TCP with TLS must prevent attackers from sniffing segments or sequence numbers using side channels. Plain UDP is trivially vulnerable to both DoS and false data injection attacks and should not be used alone. UDP with DTLS provides good protection against both data injection attacks and buffer overflow attacks. The main drawbacks to using DTLS are that it has not been widely used up to now and its implementations have not received the scrutiny that TLS implementations have received.

Providing security at the network layer, using IPsec, is certainly an alternative to the transport layer approaches investigated here. The complexity involved in correctly deploying

IPsec is a potential drawback to this approach that may cause utilities to favor transport-layer security for synchrophasor communications.

Vulnerabilities of the transport layer protocols underneath synchrophasor data transfer are just one type of uncertainty. So modeling and handling uncertainties including the vulnerabilities we explore in this chapter is very significant to the smooth operation of power grid.

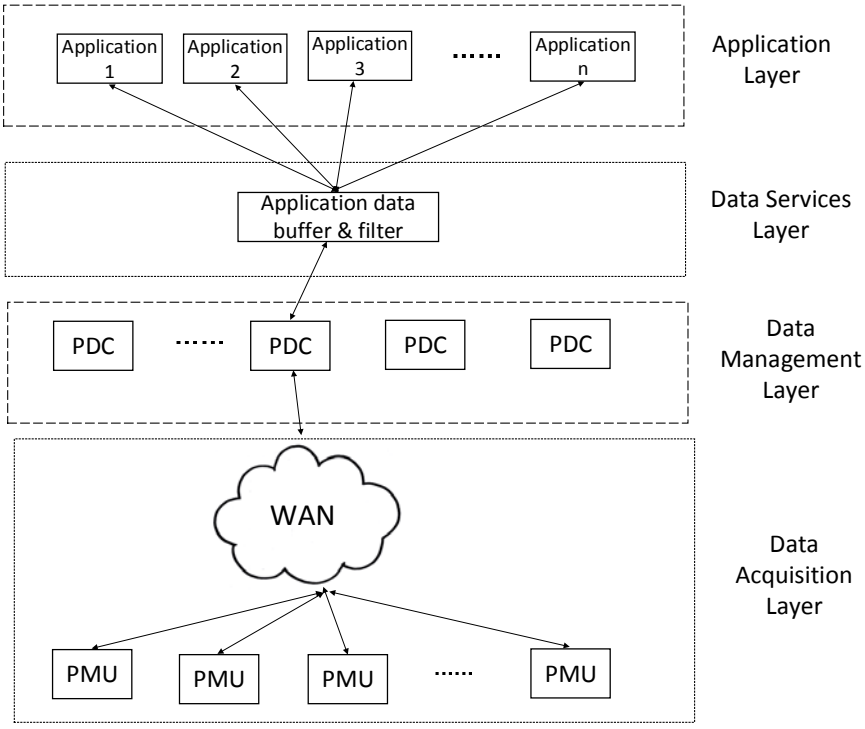

Figure 2.1: The Architecture of Wide-Area Monitoring and Control

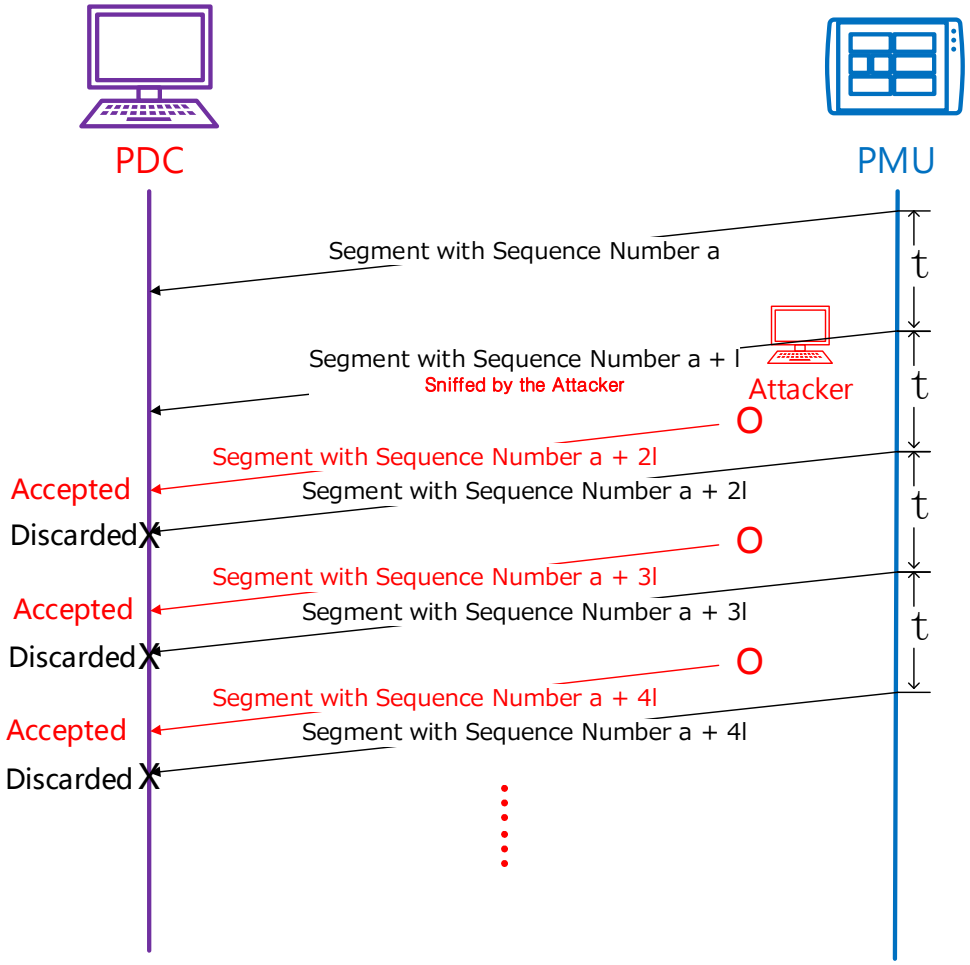

Figure 2.2: TCP Injection with a Pace Tracking Attacker

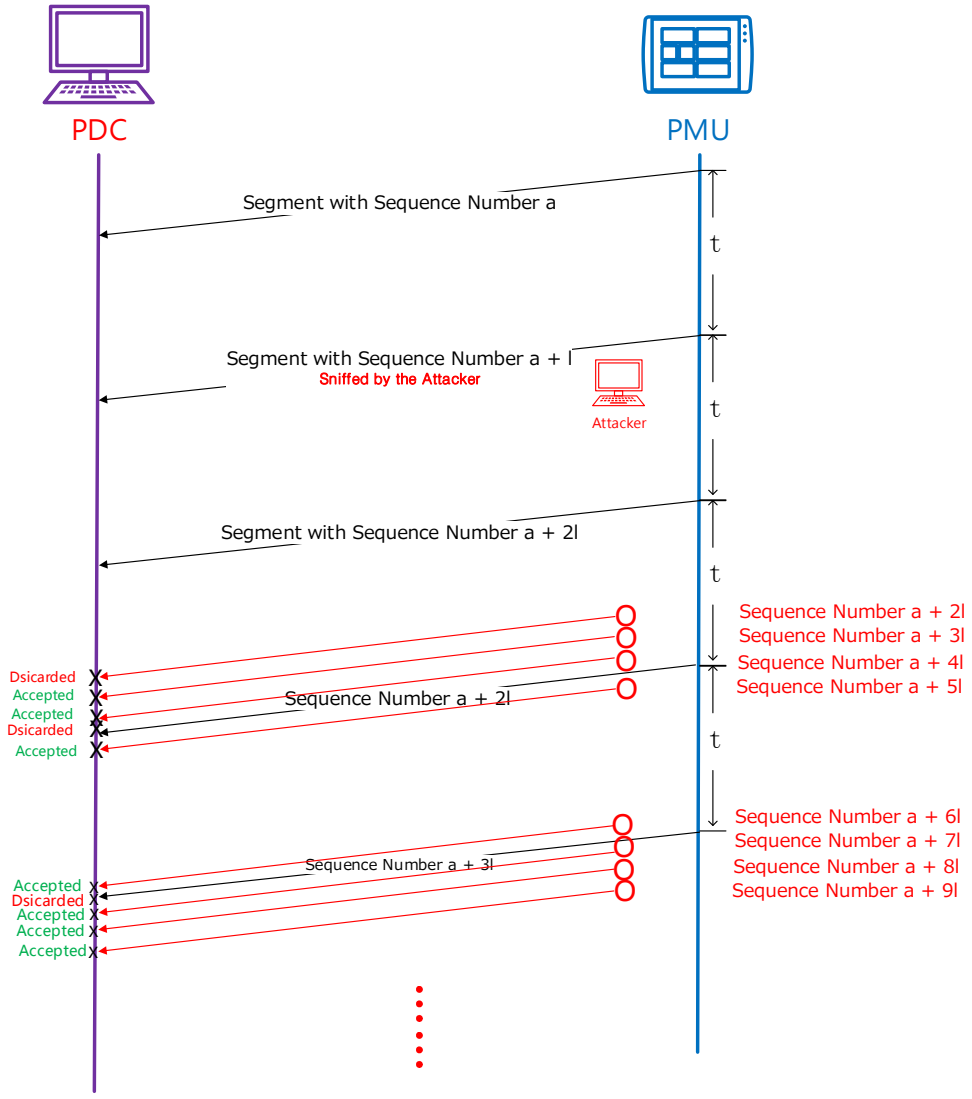

Figure 2.3: TCP Injection with an Extensive Attacker

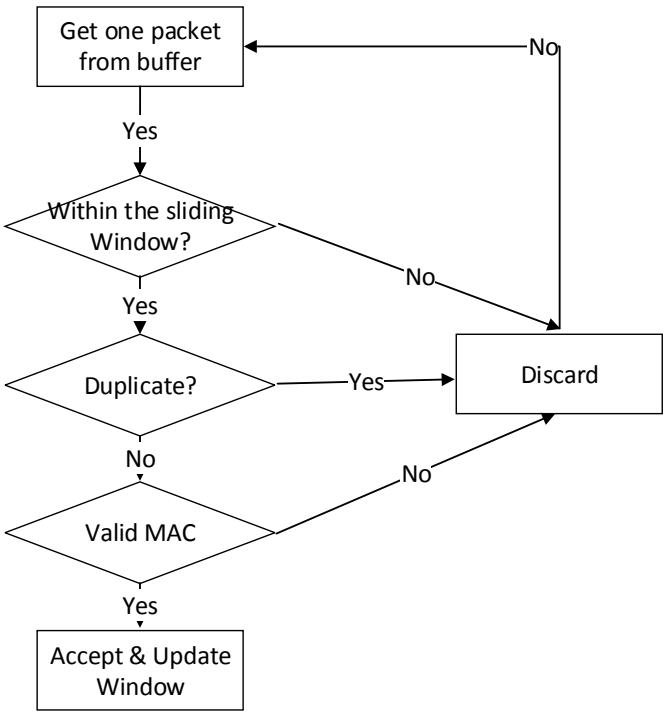

Figure 2.4: Packet Validation Logic of DTLS

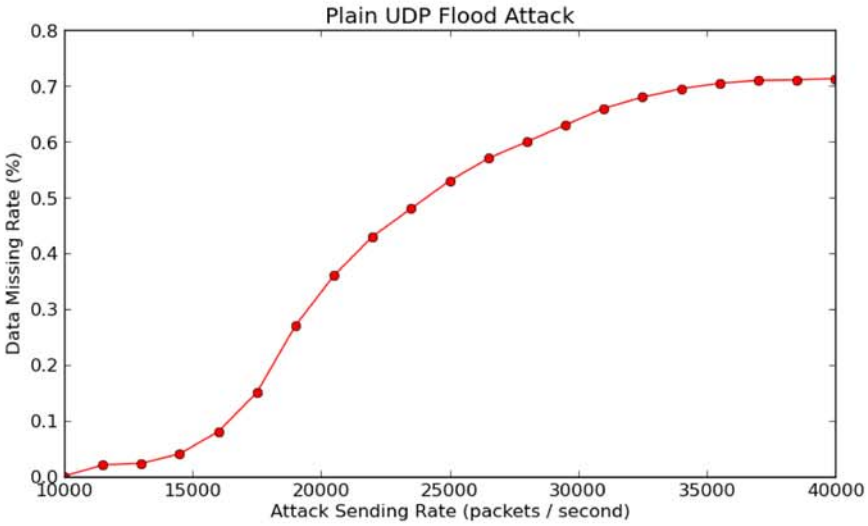

Figure 2.5: Plain UDP Buffer Flood Attack

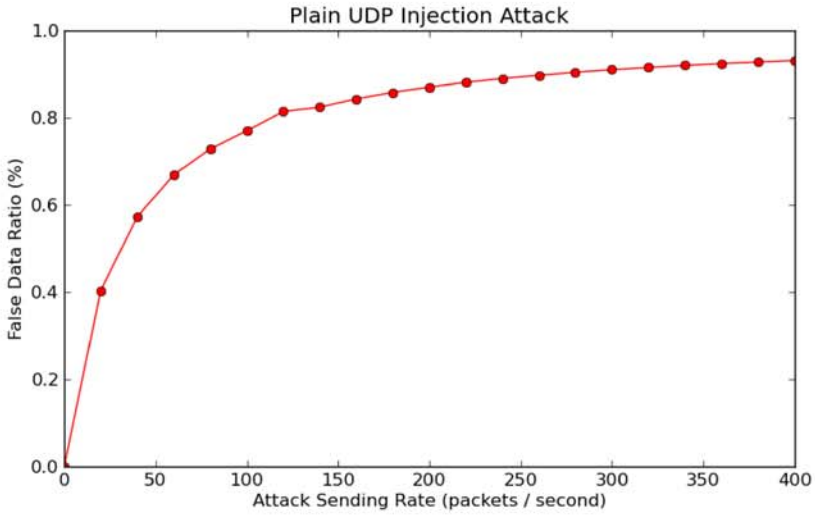

Figure 2.6: Plain UDP Injection Attack

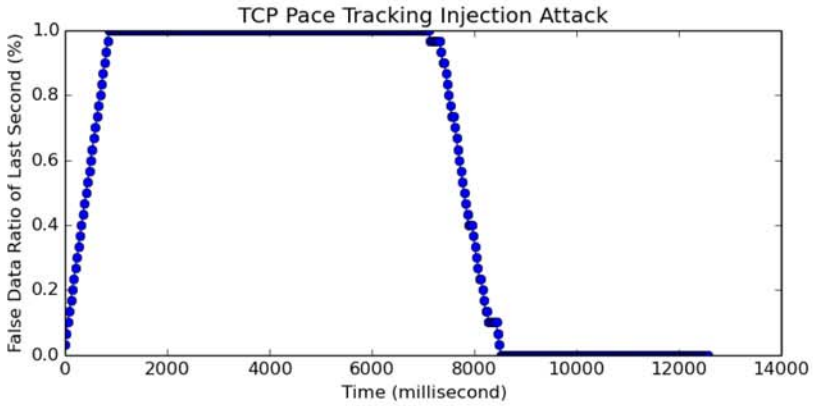

Figure 2.7: TCP Injection with a Pace Tracking Attacker

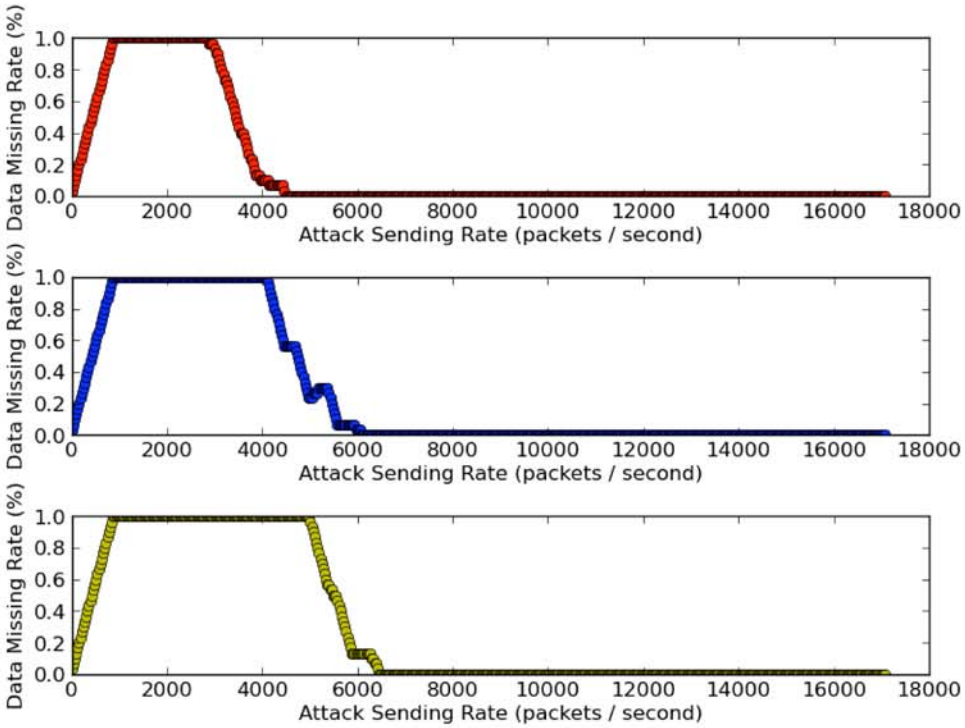

Figure 2.8: TCP Injection with Extensive Attackers with Different Attack Windows

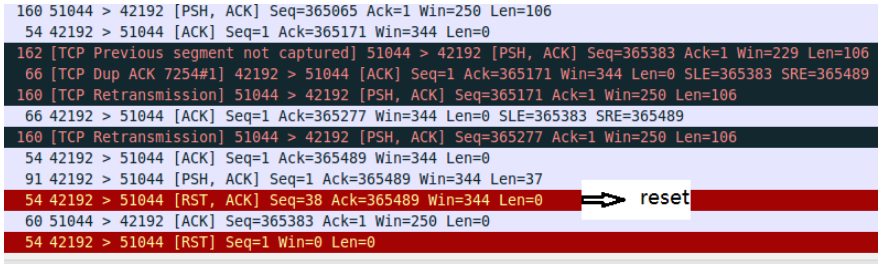

Figure 2.9: Sequence Number Attack against TLS

CHAPTER 3. INTEGRATING UNCERTAINTY INTO DECISION MAKING WITH A BAYESIAN FRAMEWORK

The U.S. power grid is on the cusp of a tremendous expansion in the amount of sensor data that is available to support its operations. For decades the power grid has been operated using supervisory control and data access systems that poll each sensor once every two or four seconds—a situation that some in the industry have characterized as "flying blind". Now, widespread deployment of sensing systems called phasor measurement units (PMUs) that provide accurately time-stamped data 30, 60, or more times each second is near at hand. Data from PMUs and other high-rate sensing devices will be used to support new control schemes in support of reliable and efficient operation of the power grid as larger fractions of electric power demand are met by intermittent sources such as wind and solar, and as controllable loads, such as electric vehicle rechargers, increase.

As power grid operations come to increasingly rely on new control schemes using these data, the security of the data and their delivery, especially availability and integrity, but to some degree confidentiality as well, is of great concern. Good security practices and technologies such as those required by the NERC CIP standards will be even more essential to reliable grid operations than they are today. However, uncertainty is inherent to the large-scale system such as power grid due to the measurement error (e.g. sensor reading error) or the stochasticity in physical processes (e.g. weather condition). What's more, the widely deployed PMUs brings even more uncertainty into the system through the following

factors: (1) PMUs are more widely deployed under the control of various entities throughout the transmission and distribution systems which may have different management policies and configurations; (2) there may be various kinds of malicious cyber attacks on PMUs and the system; (3) the amount of operational data depicting the dynamic system explodes. All these factors make the system not only uncertain but even hard to describe and predict. For example, when authentication is performed using a public-key infrastructure, the reliability of the authentication is ultimately limited by the uncertainty of the binding between a particular public key and the authenticated entity. While one might wish that there were no uncertainty about this, it is in fact quite likely in a large-scale system that some of the bindings are incorrectly known at least some of the time by some entities, whether due to mistake or malicious manipulation.

If the uncertainty is unavoidable, the reliability of the system will either come down to blind faith —*we know the security is uncertain but we have to trust in it because it is all we have*—or to decision processes that explicitly and appropriately take into account the uncertainties associated with security. **Our research proposes a way to integrate uncertainties into the decision making procedure.**

Since the power grid must be controlled in real time in an ever-changing security threat environment, we are interested in decision models that can be fully automated rather than ones that rely on insights of humans. Computational decision making in the face of uncertainty leads to the vast and expanding literature on decision theories that are used in business strategy and operations, military planning, etc. Because Bayesian decision theory fits well

with our desire for a computational solution, our approach uses a Bayesian perspective [69].

The word *trust* is introduced here for its connotations of one party's (the trustor's) reliance on and belief in the performance of another party (the trustee); for example, trust in a PKI certifier to correctly bind a public key to some other entity. The reliance or belief often must occur without certainty or may be in the form of a prediction about the future (itself a source of uncertainty). Trust, however, need not be blind: trustors can use evidence, for example in the form of past experience with a trustee, reputation information, or contracts and laws that impose penalties for non-performance, to form their trust judgments. We believe that if critical infrastructures are to be resilient against attacks it is essential that operational decision making processes *appropriately* take into account evidence about the trustworthiness of their input data. As we will show in the next section, using evidence appropriately means that it is considered in light of the particular decision being made: there is no single approach to judging trust that is universally appropriate.

3.1 A Motivating Analogy

The credit reporting and scoring system for consumer credit (Fig. 3.1) provides an interesting analogy for evidence-based decision making in the presence of risk and illustrates some of the issues. Credit bureaus collect information from various sources and provide credit reports that detail individual consumers' past behavior in borrowing and bill paying. Some companies further analyze the information in credit reports from multiple sources to produce a single, numerical credit score based on statistical analysis of a person's credit reports. The

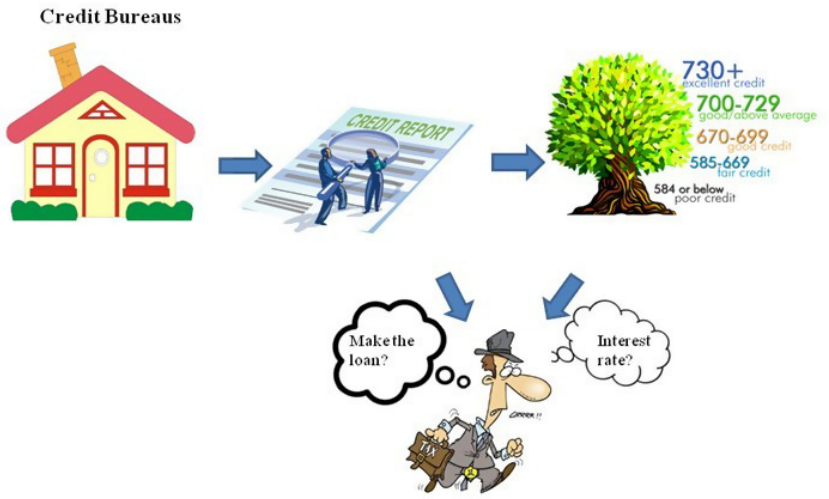

Figure 3.1: Credit Reporting System

credit score is claimed to statistically represent the creditworthiness of an individual.

Now consider the decisions lenders make in analyzing a loan application: they have to decide whether or not to make the loan and on what terms. If the loan is made, a lender stands to make a profit if the borrower pays it back, or a loss if the borrower defaults on payment. A *loss function* describes the lender's payback for various future behaviors of the borrower. While the loss function is known, the future behavior of the borrower is, of course, uncertain at the time the loan is made. The lender thus seeks to make a decision that minimizes expected loss (maximizes expected return) by assessing the probability of different future borrower behaviors. To do this they turn to the credit report or credit score as well as information about employment, income, and stability of residence contained in the loan application.

There are several important things to point out in this analogy.

- First, different lenders will have different loss functions, and a single lender may have different loss functions for different kinds of loans: trust decisions are situational. In the power grid domain, a decision to turn off electric car charging at a time when the the power supply is stressed carries different loss implications than a decision to shed load by turning off power to an entire region.

- Second, different lenders may assess the probability of various borrower behaviors differently based on the same credit report facts: trust decisions are subjective.

- Third, the analogy is imperfect: for lending, risk pooling allows businesses to balance losses from some loans with profits from others, so decisions take into account not only an individual loan but a whole portfolio of loans. Power grid operational decisions' consequences cannot be easily aggregated, so in this domain the decision processes will emphasize analysis of individual decisions.

So there are similarities and differences between the two domains. However, the structure is basically the same: the trustor collects *evidence* about trustees and uses it to probabilistically predict the behavior of the trustee according to a model. The trustor may make decisions that later, based on hindsight, seem wrong, but are nevertheless the best that could be made at the time based on the information available.

3.2 Preliminaries

The distributed control system for a large-scale critical infrastructure (such as the power grid) can be described abstractly as consisting of a collection of controllers, a collection of data sources, and a collection of actuators. Actuators and controllers share the essential characteristics, for our purposes, of dealing with uncertainty of security so we will focus in what follows on controllers and information sources. In the power grid, for example, controllers are things like protective relays, automatic generator controls, remedial action schemes, etc. Data sources include sensors, human operators, and outputs of controllers. Communication channels link data sources to controllers. The essential property of controllers is that they receive inputs from data sources and repeatedly make decisions based on those data, with the decisions ultimately being reflected in an action that changes the physical state of the grid in some way.

Because of noise in sensor outputs, in today's system inputs are assumed probabilistically related to the actual state of the sensed world by considering that each measurement consists of the actual state plus a normally-distributed noise term. Failures in the system can lead to bad inputs (highly improbable in the normally-distributed-noise model) which can often be detected and excluded by bad-data detection algorithms that exploit redundancy present in the inputs. Recent research has addressed ways that input data streams might be intentionally attacked invisibly to the bad data detectors in use today [50].

The approach described is, at a high level, aimed at providing controllers with the abil-

ity to evaluate evidence from a variety of sources regarding the correctness of data received from sensors and the ability of actuators to carry out commanded actions. The uncertainties associated with these aspects as well as with outcomes are modeled probabilistically though with much greater flexibility than afforded by the normally-distributed-noise approach currently used, and with explicit incorporation of uncertain results in the form of loss functions.

3.3 The Bayesian decision model

Decision theory studies the values and uncertainties related to making rational and optimal decisions [27]. Statistical theory has been widely applied to decision theory and is a common tool for decision making problems [63]. Our method is based on the Bayesian statistical paradigm which can quantify the uncertainties of decisions using personal probability. A systematic introduction to Bayesian decision theory can be found in [69].

As previously noted, uncertainty is inherent in complex systems and thus risk, which is a state of uncertainty where some of the possibilities involve a loss, catastrophe, or other undesirable outcome, is unavoidable. In order to reduce risk, every entity in the system should have the ability to incorporate evidence about the trustworthiness of other entities and be inclined to rely on more-trustworthy peers. To begin formalizing this viewpoint, we assume that there are a number of trust-related attributes $E = (E_1, E_2, \ldots, E_p)$ concerning each entity in the system, together forming the trust evidence. Focusing on a single entity $\mathcal{A}$, at a certain time point, it could collect the current evidence about a certain entity $\mathcal{B}$ which can be denoted as $x_i = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_p) \in \mathbb{R}^p$. Over a period of time it will collected a

number of x_is denoted $x = (x_1, x_2, \cdots x_n)$. Based on x, $\mathcal{A}$ will make a decision $d \in \mathcal{D}$ (where $\mathcal{D}$ is the decision space) on $\mathcal{B}$ in light $\mathcal{A}$'s estimate of the value of θ $(0 \leq \theta \leq 1)$ from the parameter space Θ which is called the trustworthiness to be placed on $\mathcal{B}$. Essentially, θ is probability that $\mathcal{B}$ is trustworthy.

In the current model, the decision-making process is considered as a choice of action made by the decision maker among a set of alternatives according to their possible consequences. In the power grid these decisions are made under uncertainty, i.e., the decision maker can neither know the exact consequence of a chosen decision before it occurs nor get accurate values of the evidence due to the complexity and uncertainty of the system. Probabilistic modeling is a natural choice both for interpreting the evidence, E, and evaluating the consequences. The model should not only incorporate the available information in E but also the uncertainty of this information. In the probabilistic model, $x_i \, (1 \leq i \leq n)$ follows a probability distribution f_i, $x_i \sim f_i \left(x_i|\theta, x_1, \cdots, x_{i-1}\right)$ on $\mathbb{R}^p$ where f_i is known but θ is unknown. If x is collected over a short enough period of time, it is reasonable to assume that $x_1, x_2, \cdots, x_n$ are independent repeated trials from identical distributions and the distribution can simply be denoted as

$$x \sim f\left(x|\theta\right)$$

The *likelihood function* l defined as

$$l\left(\theta|x\right) = f\left(x|\theta\right)$$

is equal to f but emphasizes that θ is conditional on x and manifests that θ can be inferred from x. According to our assumptions and the likelihood principle [7], all available

information to make inference of θ is contained in the likelihood function $l(\theta|x)$ and the value of θ can be inferred from x. Decisions can be made based on the inferred value of θ. To combine these processes, when the likelihood function $l(\theta|x)$ is fixed, a function from $\mathcal{X}$ to $\mathcal{D}$ can be obtained as $\delta(x)$ which is called the decision rule as it relates to trust. (Keep in mind that trustworthiness assessment is only one aspect of the overall decision process—decisions are made according to the inferred trustworthiness value, but trustworthiness evaluation is not the end goal).

In the remainer of this section we describe the elements involved in a Bayesian determination of decision rule $\delta(x)$, namely *prior distributions* and *loss functions,* and then state the derived rule.

3.3.1 Modeling prior information

As previously noted, trust decisions are subjective: based on the very same evidence, different trustors may make different decisions. In the Bayesian model, the uncertainty on the trustworthiness value θ of a trustor regarding a trustee *before* receiving evidence is modeled using a probability distribution $\pi(\theta)$ on Θ, called the *prior distribution.* Subjectivity of trust is naturally modelled by different prior distributions.

3.3.2 The loss function

While it is easy to talk about making "good" decisions, the model requires a precise formalization of the notion of goodness.All of the possible choices in a decision should be ordered or quantified.Decision theory uses the *loss function* for this purpose. A loss function is any function $L \geq 0$ from $\Theta \times \mathcal{D}$ to $\mathbb{R}^p$ and represents the penalty $L(\theta, d)$ associated with the decision d when the parameter takes the value θ. In our situation, the penalty $L(\theta, d)$ is the quantified consequence at the time the decision is made when the trustee's trustworthiness value is θ and the trustor chooses decision d. However, it is very hard to measure the trustworthiness value of a trustee in a complex system due to the dynamic and fuzzy nature of trust [9]. So it is important for the model to reflect such uncertainty. A simple way to obtain the loss is to integrate over all of the possible values of θ. What's more, instead of focusing on evaluating one decision, our goal is to assess a decision rule $\delta(x)$ which is the allocation of a decision to each outcome $x \sim f(x|\theta)$, so the loss function $L(\theta, \delta(x))$ should also be integrated on $\mathcal{X}$ which is the whole space of x.

Given the prior distribution, $\pi(\theta)$, and the distribution of x, $f(x|\theta)$, θ should be integrated in proportion to $\pi(\theta)$ and x in proportion to $f(x|\theta)$. So the loss function can be written as:

$$r(\pi, \delta) = \mathbb{E}^{\pi}[R(\theta, \delta)] = \int_{\Theta} \int_{\mathcal{X}} L(\theta, \delta(x)) f(x\theta)\, dx\pi(\theta)\, d\theta$$

where $r(\pi, \delta)$ is called the risk function of δ.

3.3.3 The Bayesian estimator

The goal of the decision-making model is to derive an "optimal" decision rule that provides trustors with rational decisions about trustees based on the observations (evidence), x. Optimality is implemented by minimizing the risk function $r(\pi, \delta)$. The decision maker follows the decision rules that give the smallest risk. However, most of the time, the trustworthiness value θ is unknown, so a problem arises regarding under which situation we minimize the risk function.

A common choice for Bayes paradigm is the minimax rule which chooses the $\widetilde{\delta}$ that satisfies $\underset{\theta}{sup}\, r\left(\theta, \tilde{\delta}\right) = \underset{\delta}{inf} \underset{\theta}{sup}\, r(\theta, \delta)$. Moreover, the minimax rule also fits for our original intention which is to make decisions that reduce the risk of the trustors under uncertainty.

As an implementation of the likelihood principle, the Bayesian paradigm satisfies the decision-related requirements for trust assessment. It not only quantifies uncertainties and minimizes the risk in decision-making, which is a crucial to make rational decisions, but also smoothly incorporates trustors' prior information about the trustees' trustworthiness. This is essential when the decision process is viewed in the context of long term operation of the system: trustors continuously acquire new evidence that must be combined with their prior information when making new decisions.

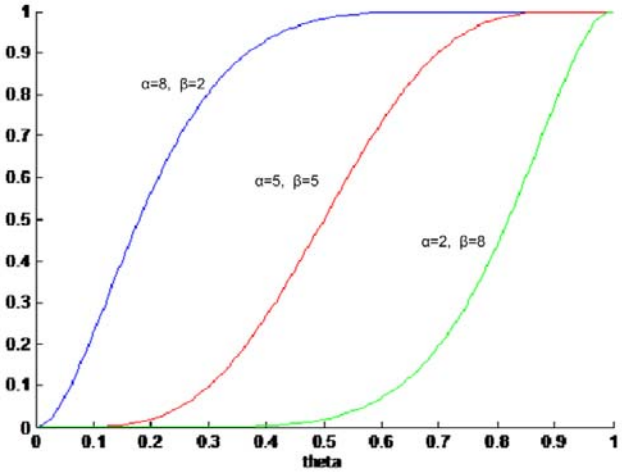

Figure 3.2: CDF of three different prior distributions

3.4 A Simple Example

In this section, we give an example of the decision-making model. We examine the simplified decision-making case with the goal of inferring the trustworthiness value of a trustee based on the observation x: so $\mathcal{D} = \Theta$.

The evidence aggregator of the trustor collects values of the related attributes $E = (E_1, E_2, \ldots, E_p)$ and stores these values in the corresponding vector $x_i = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_p)$. Within a short time, T, this evidence aggregator will collect the n vectors like x_i and form $x = (x_1, x_2, \cdots x_n)$. Since T is short, we assume that $x_1, x_2, \cdots x_n$ are independent repeated trials from identical distributions f. According to the probabilistic modeling, the values of the attributes are conditional on the trustworthiness value θ, so the distribution can be denoted as $f(x|\theta)$.

As we said before, trust is subjective. For example, risk-averse trustors may tend to make negative decisions and risk-preferred trustors may tend to make positive decisions. The differences among trustors could attributed to many factors. For instance, the difference might be attributed to former experience of the trustors: positive experience, which means that the trustor made many correct decisions on trustworthy entities, will make the trustors more risk-preferred. Conversely, negative experience, which means that trustors made wrong decisions and trusted the wrong entities, will make the trustors more cautious. For one-dimensional evidence, this particular kind of subjectivity can be modelled using a Beta-distribution with parameters α and β as the prior distribution of trustors. Let α be the number of past negative experiences and β the number of past positive experiences. The prior information of trustors can be modeled as:

$$\pi(\theta) = Beta(\alpha, \beta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt}$$

where $\pi(\theta)$ is the probability that trustor will decide to trust the trustee. Increasing α makes the trustor more risk-averse and increasing β makes the trustor more risk-preferred.

As Figure 3.2 shows, the decision maker with $\alpha = 8$ and $\beta = 2$ (top line in the diagram) will tend to make negative trust decisions since the probability that it allocates trustworthiness values under 0.5 is high. The decision maker with $\alpha = 2$ and $\beta = 8$ (bottom line in the diagram) is more likely to make positive decisions.

For this simplified example, since we just want to estimate the value of θ, we select a commonly-used simple loss function—the quadratic loss function:

$$L(\theta, \delta) = (\theta - \delta)^2$$

The risk function would be

$$r(\theta, \delta) = \int_{\Theta} \int_{\mathcal{X}} (\theta - \delta)^2 f(x|\theta)\, dx \pi(\theta)\, d\theta$$

for which the computed estimator is

$$\delta(x) = \frac{\int_{\Theta} \theta f(x|\theta)\pi(\theta) d\theta}{\int_{\Theta} f(x|\theta)\pi(\theta) d\theta}$$

Another application of Bayesian estimation would be the calculating of the Top Rated 250 Titles of Internet Movie Database (IMDb www.imdb.com). By the Bayesian estimation process, the IMDb obtains a formula to calculate the weighting rating:

$$W = \frac{Rv + Cm}{v + m}$$

where:

W = Weighted Rating

R = average for the movie as a number from 0 to 10

v = number of votes for the movie

m = minimum votes required to be listed in the Top 250 (currently 3000)

C = the mean vote across te whole report (currently 6.9)

Similar to this, our final goal of this framework is to develop a formula to calculate the trustworthiness value or make decisions relevant to trust based on the evidence collected from the power system.

3.5 Related work

Trust in the information security area is drawing increasing attention. In 1996, Rasmussen and Jansson stated the relationship between security and social control and classified security mechanisms as: *soft security* such as trust and reputation systems and *hard security* like authentication and access control [67]. Actually, typical security mechanisms include some aspects of trust, but they make explicit trust assumptions [15]. In order to overcome some drawbacks of the current security mechanisms such as the inadequacy of authentication [8], a more general concept of "trustworthiness should be managed [2].

Trust management is largely associated with inference or decision making. Related evidence should be collected first and delivered to the trust management system as input for the decision making model. Several trust management systems such as PolicyMaker [8] and REFEREE [16] were designed to collect security credentials and test the compliance of the credential with security policies. Also, some trustworthiness computing models [23] collect trustors' former experience as evidence and make predictions based on this former experience. Some models collect evidence from other entities–these are essentially reputation systems [42]. Generally, however, current trust management systems or trustworthiness computing models set their goal as determining a numerical trustworthiness value for a trustee or making a binary decision about whether a trustee is trustworthy or not. We go beyond this viewpoint by looking at trust decision making as coupled to succeeding decision processes.

CHAPTER 4. A TRUST MODELING FRAMEWORK WITH APPLICATION TO CRITICAL INFRASTRUCTURES

In Chapter 3, we have shown that uncertainty can be integrated into decision making procedure related to trust assessment. Trust, as a decision making procedure, is usually applied as input to other decision making procedures. In this chapter, we take a closer look at the ontology of trust, how to make trust decisions, how this procedure is related to uncertainty measurement in critical infrastructures, and how this framework can be applied by data consumers to select more trustworthy data sources.

4.1 Motivation of Assessing Trust for Critical Infrastructure System

Trust is a relatively new area of research in computer science but has a rich and a mature basis in other research areas. Nearly every aspect of a person's social interaction involves some form of trust [79]. Thus, trust is as almost old as the human race. Trust research first appeared in sociology [52] and then unfolded over other areas including communications [80], economics [85], and political science [37]. In these areas, trust has been intensively studied in situations that require *collaboration* or information sharing under *uncertainty*. Trust is especially becoming much more pronounced in research related to critical infrastructures. The main reason for this is that applications pertaining to critical infrastructures implicitly follows three tendencies.

1. Modern systems are becoming more distributed geographically and are stretching across multiple domains (e.g. cyber-physical systems, social media, cloud computing, etc). Consequently, agents in these systems are more exposed to environments with more uncertainty.

2. Distributed systems in mobile and ubiquitous environments demands much more frequent information sharing as well as smoother collaboration among heterogeneous agents. This again raises important trust issues.

3. The traditional Trusted Third Party (TTP) approach is deemed neither scalable in expanding structures nor resistive to intentional malicious attacks.

Security in large multi-domain systems are also challenged by the lack of expressiveness, and suffer from numerous security implications. Trust, on the other hand, is more expressive than many traditional system properties, including security, due to its *abstract* and *multi-faceted* nature [79]. Moreover, trust models can cover the inherent uncertainties present in these complex distributed systems, which most traditional security models aren't capable of capturing [22].

Uncertainty has a great influence on performance in large complex distributed systems. Unfortunately this is sometahing that is unavoidable because that the information which directly influences the reasoning (cause and effect) and decision making (rules) in these systems aren't always accurate or complete [46]. Such information can come from different sources in different forms, which makes it fundamentally unsound to categorize or classify uncertainty into different types; there is only one kind of uncertainty – *the lack of knowledge*

pertaining to the truth of a proposition [5]. **The overarching goal here is to build a mathematical framework for trust modeling that is suitable to the needs of modern critical infrastructure applications.**

4.2 Definition of Trust

Definition 1 (Definition of Trust) : Trust is the subjective belief in the consistency between trustee's behavior and trustor's expectation under a given context in an environment with uncertainty.

At the conceptual level, trust is very *abstract* and *multifaceted* [79]. The sheer expressiveness of trust makes it possible to interpret and understand it in different ways. This however, has caused the lack of a widely-accepted canonical model of trust, either conceptually or mathematically. Besides, there is no trust model with a critical infrastructure emphasis within which uncertainty is very significant in terms of its protection and security. A definition of trust with implications to critical infrastructure applications has five essential aspects as follows. By taking these aspects in to consideration, a definition of trust for critical infrastructures is provided in Definition 1.

- **Uncertainty**: It only makes sense to talk about trust when facing uncertainty
- **Belief**: Trust is a psychological state i.e. trust is belief
- **Consistency**: Trust is a judgment on the consistency between trustee's behavior and trustor's expectation

- **Context**: A trustor only trusts a trustee under a specific context
- **Subjectivity**: Trust is subjective. Different trustors may make different trust decisions based on the same set of observation under same context

4.3 Trust Assessment and Decision Making

A generalized trust modeling problem can be depicted as follows: assume that there are n agents in a certain problem scope. Given a time window $\mathcal{T}$, timings $t_i,\ t_i \in \mathcal{T}$, and a potential trustee agent α, a trustor agent β can have an observation $x_{t_i}^{\alpha}$ of agent α's behavior on some context Ω. The series of observations $x_{t_i}^{\alpha}$ within $\mathcal{T}$ is denoted as X^{α}. Thus, the trust modeling problem can be defined as follows[3]:

Definition 2 (Trust Modeling Problem) Given a trustee agent α, a trustor agent β, and β's observation X on α within time window $\mathcal{T}$ on context Ω, β has to decide between the hypotheses:

- $\mathcal{H}_0$: trustee α is NOT trustworthy based on X;
- $\mathcal{H}_a$: trustee α is trustworthy based on X.

The aim of the proposed framework is to formulate the rules for the hypotheses testing in Definition 2. From Definition 1, trust modeling is essentially a measurement between the

[3]When the trustee and time are obvious, we write X^{α} as X and $x_{t_1}^{\alpha}$ as x

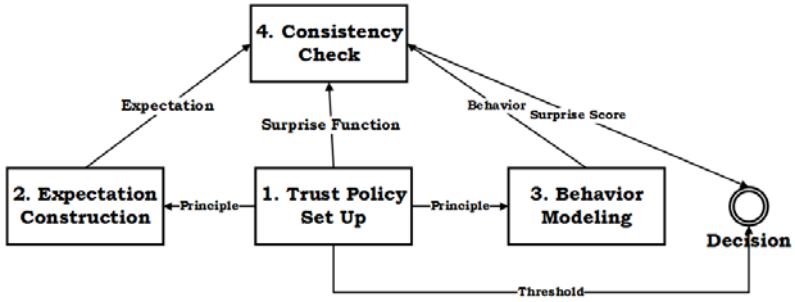

Figure 4.1: The Framework of Trust Modeling for Critical Infrastructures (left) and Trust Policy Setup

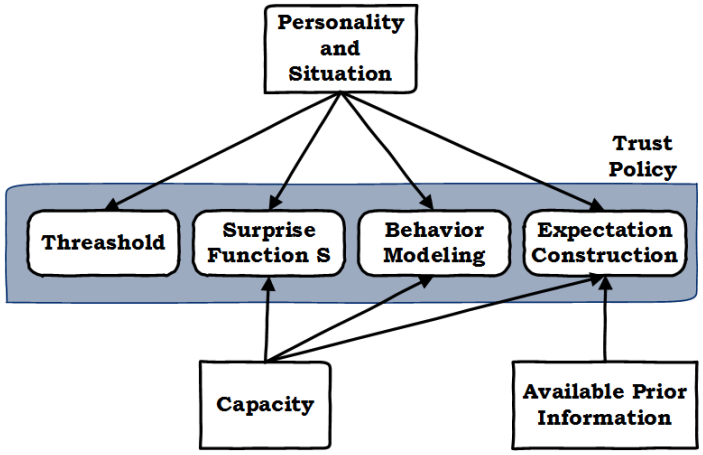

Figure 4.2: The Framework of Trust Modeling for Critical Infrastructures (left) and Trust Policy Setup

difference in trustor's expectation and the trustee's behavior. Specifically, this difference is called the *surprise* and the function to check the difference is termed the *surprise function*. With this, the two hypotheses in Definition 2 can be extended to include surprise as follows:

- $\mathcal{H}_0$: trustee α is NOT trustworthy based on X, if $\mathcal{S}(E, P) > \tau$;

- $\mathcal{H}_a$: trustee α is trustworthy based on X, if $\mathcal{S}(E, P) \leq \tau$

where τ is a threshold determined by the trustor, E is the trustor's expectation, and P is the trustee's behavior which can be inferred from X. Both E and P can be either distributions (functions) or vectors. $\mathcal{S}$ is the surprise function.

The conceptual flow of the rules for this hypotheses testing, i.e., the framework of trust modeling, is illustrated in Figure 4.2. As shown, the whole trust modeling framework is composed of four phases: trust policy set up, expectation construction, behavior modeling, and consistency check. In the *trust policy set up* phase, trustors set up their principles for expectation constructions, behavior modeling and decision makings. In the *expectation construction* phase, trustors form their expectations. In *behavior modeling* phase, trustors model the behavior of trustee agents according to their observations X on these agents. In the final phase, trustors compute the surprise scores indicating the differences between their expectations and trustee agents' behavior models, compare these scores with the thresholds set up in trust policy set up phase, and conclude their final trust decisions.

4.4 Trust Policy Set Up

A trustor sets up his/her trust policy such as the method to construct the expectation and the threshold for the hypotheses testing. These policies should be made according to the trustor's context of *personality* which is his/her internal characteristic and *situation* which is his/her condition in the external environment.

Figure 4.2 shows the trust policy set up phase, which is made up of four factors: a threshold τ for the hypotheses testing, principle to construct expectation E, principle to model trustee's behavior P based on X, and surprise function $\mathcal{S}$. Practically, these factors are usually determined by three trustor attributes: *personality and situation*, *available prior information*, and *capability* to digest the information.

- **Personality** and **Situation**: Personality and situation are inherent to individual trustors, thus, they can influence all aspects of a trustor's trust policies. For example, trustors, who are less vulnerable to untrustworthy trustees, tend to be bold thus, the processes of making a trust decision is relatively easy. Conversely, trustors who are more vulnerable to untrustworthy trustees are usually cautious when making trust decisions. A super agent – an agent with a lot of interactions with a large pool of agents within the same context – maybe picky about trustees since they have more potential options and opinions. At the same time, their trust decisions might have greater impact on the whole system.

- **Available Information**: In addition to the behavior data X, a trustor may have some prior information about a trustee. This prior information determines how the trustor can construct his/her expectation about the trustee. Explicitly, the trustor may have:

    1. Sufficient prior information which means the trustor has very specific expectations on the trustee;

2. Partial prior information signifying that the trustor knows the domain of the expectations for the trustee. The trustor may assign a weight to an expectation indicating the frequency of its occurrence. This forms a distribution of expectations;

3. The trustor may have no prior information to construct his/her expectation(s) except some historical behavior data of trustee. So it would rely completely on the historical behavior data $X_{[1,t_i-1]}$ to construct his/her expectations or adopt expectation from other trusted peers.

- **Capacity**: The selection of function $\mathcal{S}$, trustee's behavior modeling, and the expectation construction are all dependent on trustor's computing capacity and capability to process information and data.

In practice, the process of setting up a trust policy is the most fundamental and difficult task in trust modeling. It has the greatest influence on the performance of the model and reflects trustor's subjectivity and flexibility. Trustors can construct their expectations and complete other trust modeling phases only after the trust policy is set up.

4.4.1 Expectation Construction

In this phase, the trustor should form his/her expectation according to some prior information such as trusted peers' endorsements. There are three cases for the available prior information:

1. The trustor has a concrete expectation within the time window T so that the prior information will be $E = \{e\}$ (e stands for a concrete expectation);

2. The trustor has a set of possible expectations within time window T so that the prior information will be $E = \{e_1, e_2, \ldots, e_n, \ldots\}$ ($e_1, e_2, \ldots, e_n$ are concrete decisions). If the trustor assigns weight to every $e_i \in E$ indicating their frequencies of occurrences, this will form a distribution $f(e)$.

3. The trustor has no expectation and cannot obtain significant prior information about expectation. Thus, $E = \emptyset$. In order to complete the trust decision, the trustor has to construct his/her expectations E' based on historical observations X^- or adopt expectations E^a from other trusted relevant peers.

4.4.2 Behavior Modeling

In this phase, the trust model takes observation data X on the trustee's current behavior as input and outputs a model P describing trustee's behavior. For simplicity we directly use X as the model of trustee's behavior instead of constructing more complicated models.

4.4.3 Consistency Check

In this phase, the trustor's surprise function $\mathcal{S}$ determined in the policy set up phase takes the expectation E formed in the expectation construction phase and the behavior

model P generated in the behavior modeling phase as inputs and outputs a score indicating the difference between trustor's expectation and the trustee's behavior. This value is then compared with the threshold τ which is predefined in the policy set up phase to decide whether trustor should reject $\mathcal{H}_0$.

4.4.4 Trust Modeling Framework

Context is implicitly fixed in certain trust modeling frameworks. A trustor begins with setting up his/her trust policy which can reflect his/her *subjectivity*. Then the modeling of expectation and trustee's behavior takes *uncertainty* into consideration. Consistency check procedure explicitly investigates the *consistency* between trustor's expectation and trustee's behavior. Trustor's *belief* of trust forms after trustor compares the surprise score and the threshold. So generally, our framework closely follows our definition and characteristics of trust.

4.5 Two Concrete Trust Models

In this section, we show two specific models corresponding to three cases for expectation construction. We review the residue check method for bad data detection in power grid state estimation as a trust model complying with concrete expectation $E = e$ case. Then, we propose two concrete algorithms for trust modeling – the trust model with prior expectation (PE model) and the trust model with self-built expectation (SE model) – both of which

closely follow the proposed framework. These two algorithms are corresponding to the case $E = \{e_1, e_2, \ldots, e_n, \ldots\}$ and $E = \emptyset$ respectively.

Power grid operation is heavily relied on the measurement data, which defines the grid's state. These measurements are typically transmitted to a control center, which provides monitoring and control over the power system. State estimation, which is to best estimate the power grid condition according to analysis of measurement data and power system models, is crucial to both this monitoring and control of the power grid [50]. For instance, consider a DC power state estimation based on a linearized AC power flow model [47]:

Specifically, we propose a model that fits cases with a set of known expectations (PE model) and another model which is suitable to applications without foregone expectations available (SE model). Simulation results show: (1) that the trustors have the flexibility to choose a model that best fits the trust decisions they desire; and (2) that both the SE and PE models work well on both data with and without oscillations. Some of the planned future work includes further exploration on trust policy construction and advanced techniques to develop expectations from prior information.

$$\mathbf{z = Hx + a + e}$$

Here, $\mathbf{e}$ is the noise and $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \Sigma_e)$. And $\mathbf{a}$ is the malicious data injected by an adversary ($||\mathbf{a}||_0 \leq k$ means $\mathbf{a}$ is a vector with at most k non-zero entries). $\mathbf{z} \in R^m$ is the vector power flow measurements. $\mathbf{x} \in R^n$ is the system state.

Since the control center obtains the estimated state $\hat{\mathbf{x}}$, its expectation of the measurement $\mathbf{z}$ should be $E = \{\mathbf{H}\hat{\mathbf{x}}\}$. The surprise function $\mathcal{S}$ is the 2-norm of of difference of $\mathbf{z}$ and

$\mathbf{H}\hat{\mathbf{x}}$, which is $\mathbf{r} = ||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}||$. If $\mathbf{r} < \tau$, trustor rejects $\mathcal{H}_0$ and vice versa.

4.5.1 PE Model: Trust Model with Prior Expectation

We can make use of the Kolmogorov-Smirnov (KS) test to define the surprise function $\mathcal{S}$ for cases where the trustor has a distribution of expectations, $E : q = f(e_i), e_i \in E$. In $\mathcal{S}$, we consider the expectation as the reference probability distribution and X as the sample data. Essentially, the KS test is used to decide whether a sample is coming from a population with a specific distribution. Thus, the idea in the PE model is to test whether the behavior data of the trustee (sample data) follows the trustor's expectation (the specific given distribution of expectation). So in this model, trustor can make use of the observation data X on the trustee's behavior directly without modeling it.

KS test is based on the empirical cumulative distribution function (ECDF)[4]. Firstly, we reorganize the data of X in ascending order $X = \{x_1, x_2, \ldots, x_n\}$. If $i < j$, then $x_i \leq x_j$, where $1 \leq i, j \leq n$. Then, by the definition of ECDF in [35], we can obtain the ECDF of X:

$$K_N(x_i) = \frac{n(i)}{N}$$

where $n(i)$ is the number of points less than x_i. The KS test is to measure the maximum distance between the cumulative distribution of expectations and the ECDF of the behavior

[4]ECDF is the distribution function associated with the empirical measure of the sample

data.

The surprise function (Kolmogorov-Smirnov test statistic) is defined as:

$$\mathcal{S} = \sup_{x_i} ||F(x_i) - K_N(x_i)||, \forall x_i \in X$$

where F is the cumulative distribution of the expectation distribution which must be a continuous distribution and fully specified (i.e., the location, scale, and shape parameters cannot be estimated from the data). Thus the hypotheses can be rewritten as:

- $\mathcal{H}_0$: (Trustee is NOT trustworthy) $\mathcal{S}(E, X) = KS(F, K_N) > \tau$. The behavior data doesn't follow the expectation distribution;
- $\mathcal{H}_a$: (Trustee is trustworthy) $\mathcal{S}(E, X) = KS(F, K_N) \leq \tau$. The behavior data follows the expectation distribution.

4.5.2 SE Model: Trust Model with Self-built Expectation

Trustors in critical infrastructures more commonly don't have predefined expectations. Thus, trustors resort to constructing their own expectations based on historical observations. For the expectation established according to historical data, it is further assumed that they are highly correlated to the future behavior. More specifically, a trustor can predict the behavior of the trustee in some sense in accordance to the historical data.

Here, we apply the kernel functions [60] to construct the expectations. A kernel function is a weighting function used in non-parametric techniques that makes use of the data

obtained to estimate the underlying distributions. Let $X_{t-1} = \{x_1, x_2, \ldots, x_n\}$ be the series of observation data within a time window T_{t-1} and assume that they are i.i.d. examples drawn according to a distribution $g(x)$. The Parzen-window estimate [84] of $g(x)$ is based on the n samples in X and is defined as:

$$\hat{g}(x) = \frac{1}{n} \sum_{i=1}^{n} \delta(x - x_i)$$

where $\delta(\cdot)$ is a kernel function with localized support and its exact form depends on n. The Gaussian kernel function is the most common kernel function because Gaussian function is smooth and hence the estimated distribution function $\hat{g}(x)$ also varies smoothly. Thus, $\hat{g}$ can be expressed as a Gaussian kernels with common variance σ^2:

$$\hat{g}(x) = \frac{1}{n(2\pi)^{\frac{d}{2}}\sigma^d} \sum_{i=1}^{n} \exp\{-\frac{||x - x_i||^2}{2\sigma^2}\}$$

where d is the dimensionality of the feature space.

Consider time window T_t right after T_{t-1}, which results in another time series of observation data X_t. Using the same methodology as above, we derive the Parzen-window estimator $\hat{h}(\tilde{x})$ based on X_t.

So for the time instance t, $\hat{g}x$ is considered the expectation E and $\hat{h}(\tilde{x})$ as the behavior of trustee P. Then the question is how can we measure the difference between this P and E which are both distributions?

For the surprise function, we measure the difference between two distributions $\hat{g}(x)$ and $\hat{h}(\tilde{x})$ with relative entropy (Kullback-Leibler (KL) divergence) [48]:

$$\mathcal{S} = KL(\hat{g}(x), \hat{h}(\tilde{x})) = \int_{-\infty}^{\infty} \ln(\frac{\hat{g}(x)}{\hat{h}(\tilde{x})})\hat{g}(x)dx$$

Thus. the hypotheses can be adapted as:

- $\mathcal{H}_0$: (Trustee is NOT trustworthy) $\mathcal{S}(E, X) = KL(\hat{g}(x), \hat{h}(\tilde{x})) > \tau$;
- $\mathcal{H}_a$: (Trustee is trustworthy) $\mathcal{S}(E, X) = KL(\hat{g}(x), \hat{h}(\tilde{x})) \leq \tau$.

4.6 Numerical Results

In this section, we test our models on data sets generated by **GridSim** [4]. GridSim is a power grid simulation software which generates simulated PMU data streams at a rate of 30 samples/sec which includes time, voltage, current, and phase angle measurements. The data is assumed noise free as they are generated by solving highly accurate power system differential equations. For experimental purposes, we manipulate the generated data streams by injecting random vectors to them and then use the original data as data from "trustworthy agents" and manipulated data as data from "untrustworthy agents".

We intercepted 5 minutes of GridSim data and extracted 101 streams of voltage readings between different pairs of buses in Kundur's 2-area system [49]. There were 3 observable oscillations in this 5 minute window. For comparison and performance measurement purposes, we separated the data frames in both trustworthy and untrustworthy data streams into data frames with oscillation and data frames without oscillation and ran our algorithms on both of these kinds of frames.

We tested both of the PE model and SE model on these two types of data frames (with and without oscillation) and measured the *sensitivity* and *specificity* of our algorithms. In statistics, sensitivity and specificity are measures of performance of a binary classification test. Sensitivity measures the proportion of actual positives which are correctly identified as such and specificity measures the proportion of negatives which are correctly identified. In other words, $sensitivity = (1-$ type I error$)$ [5] and $specificity = (1-$ type II error$)$ [6].

4.6.1 Simulations of the PE Model

Figure 4.3 illustrates a ROC curves for the PE model on both data frames with oscillation and data streams without oscillation. The ROC curve is a comprehensive indicator of the performance that is created by plotting the *sensitivity* vs. $(1 - specificity)$ (i.e., type II error) at various thresholds of τ. Since all the data streams are from the same bus system, it is expected that their behaviors are highly related. We arbitrarily pick one of these data streams as the a trustworthy *reference data stream* to construct the expectations for all other streams.

To test the accuracy of the models, we manipulate the other 100 data streams without oscillation data frames and inject random "attack vectors" to emulate "data streams

---

[5] Type I error occurs when the null hypothesis ($\mathcal{H}_0$) is true, but is rejected

[6] Type II error occurs when $\mathcal{H}_0$ is false but is accepted

from untrustworthy PMUs". Then we label the original data streams as "trustworthy data streams" and manipulated ones as "untrustworthy data streams". We perform a KS test on both the trustworthy data streams and the untrustworthy data streams to get the KS statistic values respectively. We make different trust decisions on these data streams as the threshold τ varies between $[0, 1]$ evenly. The same procedure is applied to the 101 data streams with oscillations to obtain its respective ROC curve.

The area under the curve (AUC) of ROC is a number indicating the comprehensive performance of the respective model. Higher AUC number means greater accuracy on trust decisions. ROC in Figure 4.3 without oscillation is 0.8894, which indicates that for the PE model, the probability that a randomly chosen untrustworthy data stream gaining higher KS score than a randomly chosen untrustworthy data stream is 0.8894. AUC of ROC with oscillation is 0.7908.

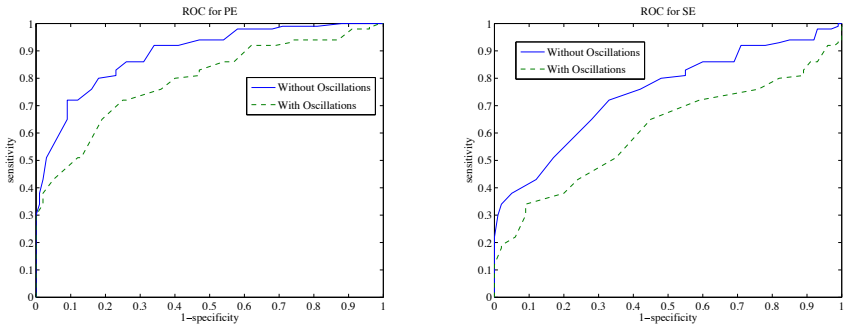

Figure 4.3: ROC Curves of PE Model (left) and SE Model (right)

4.6.2 Simulation on the SE Model

We draw a ROC curve for the SE model by following the same procedure and by injecting a random attack vector as in the PE model above. This is illustrated in Figure 4.3 (left). Here, we use a one second time window and construct distributions according to Parzen-window for every time window. Then we apply the surprise function to measure the relative entropy between every two consecutive time windows and normalize the data. The AUC of ROC for SE model without oscillations is 0.7474 and AUC of ROC with oscillations is 0.6072. What this means is that the probability that a randomly chosen untrustworthy data stream gaining higher score of relative entropy than a randomly chosen untrustworthy data stream is 0.8894 for data streams without oscillation. The same comparison for data streams with oscillations is 0.6072.

As ROC curves in Figure 4.3 (right) show, both PE and SE models work better on data without oscillation than on data with oscillation. Our explanation for this is that both the PE model and SE model are based on some significant assumptions and oscillations break these assumptions. For PE model, it is implicitly assumed that all other data streams follow the data stream that is chosen as the reference expectation. SE model assumes that data points from the same data stream follow the same distribution. When oscillation occurs, the distributions change in unpredictable ways and violate these assumptions.

CHAPTER 5. QUANTIFYING UNCERTAINTIES OF SECURITY AND QOS AND ENABLE ADAPTIVE SECURITY FOR DESIGN OF POWER GRID COMMUNICATIONS SYSTEMS

According to the analysis in Chapter 1, choosing the appropriate security scheme for an application like synchrophasor data communication is an intricate problem. No single security scheme or security protocol solves all security problems, especially when performance and implementation constraints, as well as uncertainties associated with implementation quality and human fallibility are considered. In this chapter, a method is proposed for choosing fitting security schemes in power grid communication systems by means of quantifying related uncertainties.

Archetypal power systems are interconnected systems linking distribution systems, transmission systems, and generation units [56]. Reliable operations of the power grid rely on the communications between and within these sections. So the timely delivery of trustworthy data is critical to the power grid. Therefore, both Qualify of Service and security of the communication system should be taken into account integrally to make best effort to guarantee the service requirements [6] and mitigate malicious attacks [54] [71].

The term "Quality of Service" refers broadly to many aspects of the performance of communication systems. For power system applications, relevant QoS metrics include end-

to-end delay, report rate, packet jitter, and packet loss rate [4]. The QoS requirements of power system applications depend on both the applications algorithms and their purposes (e.g. protection, control, monitoring).

Uncertainty, however, is inherent to both QoS and security, which makes it very difficult to gain an all-inclusive point of view of QoS and security. For instance, randomness of the system could make it very difficult to measure parameters critical to QoS and security of the system very accurately. Methods for quantifying the uncertainty associated with QoS and security schemes and relating the uncertainty to the performance metrics are needed in order to:

- Understand the correspondence or trade-off between QoS and security;
- Understand the overall trustworthiness of the communication to meet the combined QoS and security requirements.

More specifically, quantifying uncertainties of security and QoS can contribute to power grid in three ways:

- It can improve administrators understanding of the trade-offs between security and QoS and the uncertainties pertaining to them;
- It can help the designers of a power systems communication network choose the most appropriate security scheme considering both the QoS requirements and the security requirements;

- It can be informative for the administrators of the system to monitor and evaluate the security schemes in practice.

Uncertainty can be classified into two types: aleatory and epistemic [59]. Different types of uncertainty originate from different sources and have different characteristics. Thus, they should be handled with different techniques. We make use of probabilistic modeling and subjective logic to handle aleatory and epistemic uncertainty separately and then apply the Monte Carlo method to draw the joint distributions. The joint distributions could provide the administrators of the system a comprehensive perspective for making better QoS and security decisions.

5.1 Power System Communication Model and QoS Requirements

QoS requirements depend in part on the communication model being used for a particular application in the power system. Communication models can be generalized from diverse communication scenarios that are encountered. In this section, different communication scenarios are inspected and further classified into different communication models. The characteristics of each communication model are described. QoS requirements can be estimated by analyzing the communication models.

5.1.1 Communication Mode

Communication schemes in power systems can be divided into *Unicast* mode and *Multicast* mode. Unicast means that message is sent to a single destination by its unique address. In power system, for example, communication between Supervisory Control and Data Acquisition (SCADA) systems apply such routing scheme. Each SCADA system polls measurements from one substation at a time. In order to obtain the overall information, it needs to poll every substation one by one. In a similar way, unicast can also be employed to obtain measurements from a Intelligent Electronic Device via Manufacturing Message Specification (MMS). Multicast means same message is addressed to a group of receivers concurrently. In power system, for instance, a Phasor Measurement Units (PMU) may send measurements to a few Phasor Data Concentrators (PDC) using multicast scheme. Similarly, in substation, Generic Object Oriented Substation Event (GOOSE) and Sampled Value (SV) are with multicast as well.

Based on communication initialization, communication modes can be categorized as *polling* and *pushing*. From the point of view of data sources, polling is passive, a data source updates the measurements or status once it receives the request from the data users. Protocols such as IEC60870-5-104, DNP3, and MMS are working with polling mode. However, the other communication mode, pushing, is initialized by the data source. It updates the measurements or status by itself periodically or driven by events occurred or a combination of periodicity and event-driven. For example, PMU sends data based on its configured re-

porting rate of 10-60 packets per second. Similarly, SV also works in periodic pushing mode (in light edition of IEC61850-9-2, it sends 80 packets per power cycle for protection and 256 packets per power cycle for measurement). Taking event-driven pushing mode as an example, an IED can be configured to report via MMS using report mode. In this mode, data receivers do not need to query IEDs to obtain the updated measurements or status. Once the measurement value or status changes, IED automatically sends updated message to the data receivers. The third working mode in pushing mode is combined periodic and even driven mode. As with GOOSE, when there is no event occurred, GOOSE pushes message in a long period of time (i.e. 1 packet per second depending on configuration). When there is event occurred, GOOSE is able to push a large amount of messages in a very short period of time (i.e. up to 1000 packet per second depending on configurations).

The communication routing scheme and communication mode of major power system communication protocols are listed in the following table.

The state diagram and work flow of combined periodical and event driven communication pattern can be illustrated in Fig 5.1 and Fig 5.2.

Fig. 5.3 is drawn to compare different communication patterns. In this figure, an event is introduced into the system at 0.4s.

At top of the Fig. 5.3, the periodical communication pattern is given where the report rate is 60 packets per second. The report rate is constant no matter when an event occurs. In the middle part of the figure corresponds to the event driven communication pattern. The measurement or status is updated only after event occurs. The bottom of the figure

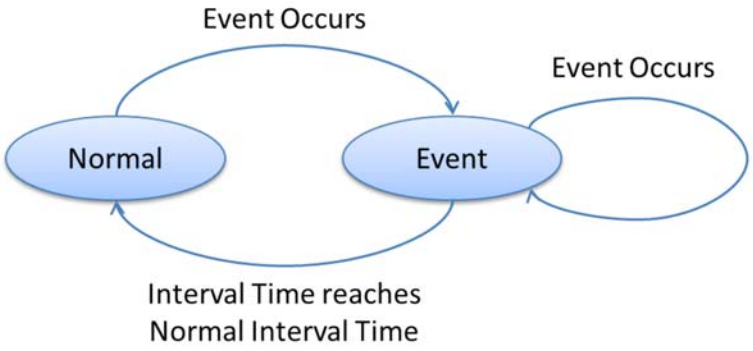

Figure 5.1: State Diagram of combined periodical and event driven communication pattern

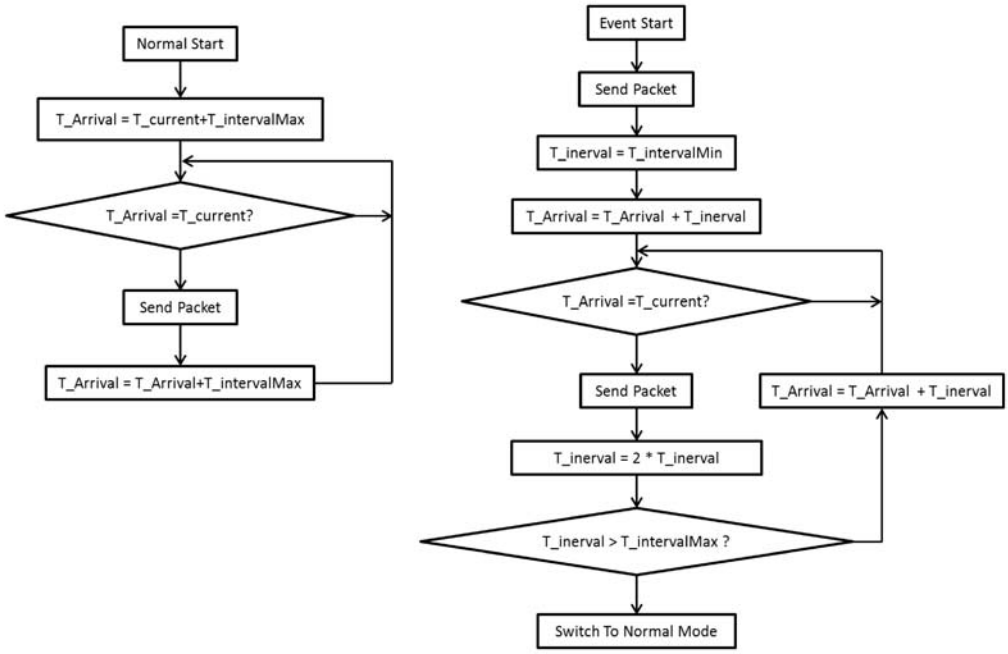

Figure 5.2: Work flow of combined periodical and event driven communication pattern

reveals the combined periodical and event driven communication pattern. The report rate increments exponentially after event occurs and remains as constant when no event occurs.

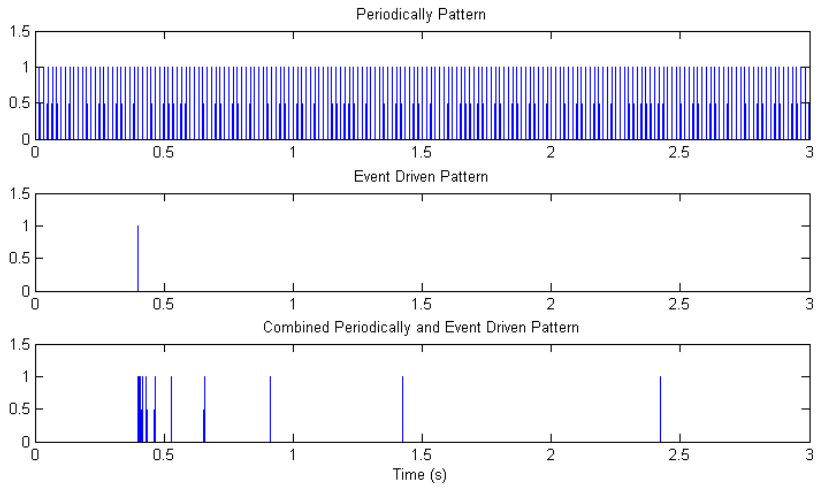

Figure 5.3: Comparison of different communication pattern

5.1.2 Tools for Quantifying Uncertainty

Probability distributions are the most common way to describe uncertainty. A probability distribution associates a probability to each measurable subset of the possible outcomes of a stochastic procedure. Details about probability distribution can be found in [30].

Subjective logic is a relatively new genre of probabilistic logic that specifically takes uncertainty and subjective belief into consideration. In general, subjective logic is suitable for modeling and analyzing situations involving uncertainty and incomplete knowledge [39]. A very typical application of subjective logic is to model trust networks[40].

Opinion is the essential concept of subjective logic, which stands for the belief owner's subjective beliefs on a proposition. A binomial opinion applies to a single proposition, and can be represented as a beta distribution. A multinomial opinion applies to a collection of

propositions, and can be represented as a Dirichlet distribution. Through the correspondence between opinions and Beta/Dirichlet distributions, subjective logic provides an algebra for these functions.

Subjective logic is applied to capture a fundamental uncertainty of human being's inference on a proposition. In addition, whenever the truth of a proposition is expressed, it is always done by an individual, and it can never be considered to represent a general and objective belief. That is the origin and advantage of subjective logic.

Nowadays, the administrators of the system have much flexibility of choosing a specific security scheme. However, the inherent uncertainties often obscure the procedure of choosing a specific scheme. So appropriately quantifying the related uncertainty is key to a secure communication system for power grid with highly guaranteed QoS.

In this chapter, we model two different types of uncertainties with subjective logic and probability distributions respectively. A comprehensive quantification of the uncertainties can greatly sharpen the administrators' understanding on the trade-offs between security and QoS and help them to select the most fitting security scheme and meet the requirements of QoS. A brief study case demonstrates how our method works for two different application scenarios.

5.1.3 Power System Application QoS

In modern power systems, global data is shared for power system protection, control, and operation. Since the data is delivered via Wide Area Network (WAN), it might suffer

unavoidable QoS problems. Such degradation potentially affects the performance of power system applications [64][68].

Researchers have proposed several mechanisms. Firstly, QoS performance can be taken into account of application design. For example techniques such as trajectory extrapolation compensation [77], H_∞ compensation [10][81], Fuzzy algorithms [57], and Phase-lead compensation [14] can be used to compensate latency of the data link. Secondly, QoS performance can be ensured by management of communication infrastructure [6]. Another main factor which can affect QoS performance is cyber security schemes. For example, the computation delay of encryption and authentication are usually not negligible. Depending on which type of security countermeasures is applied, computational delay varies. In order to ensure power system applications work properly, security countermeasures should also be considered incorporating with QoS. Therefore, delay is considered as following equation:

$$T_{\text{latency}} = T_{\text{sec}} + T_{\text{comm}}$$

where T_{sec} is the computational delay introduced by cyber security and T_{comm} is the latency introduced by communication path.

The communication delay is normally considered as following equation:

$$T_{\text{comm}} = T_{\text{prop}} + T_{\text{queue}}$$

where T_{prop} is the propagation delay and T_{queue} comes with routing, buffering process in the communication system. T_{comm} is determined by the application scenario and communication model and T_{prop} can be measured. Therefore, T_{sec} can be inferred straightforwardly.

For QoS requirements related to security, we mostly consider the requirement on delay that can be taken by security schemes and the communication model that the security scheme supports.

5.2 Security Schemes for Cyber Systems of Power Grid

In order to strengthen the reliability of power grid, security schemes are applied to mitigate malicious attacks from external threats [71]. Identical to traditional conceptual features of information security, security objectives of the cyber system of smart grid can also be classified as follows [78]:

- **Availability**: to ensure the timely delivery of proper data;
- **Integrity**: protect authentic data from the malicious manipulations;
- **Confidentiality**: only authorized stakeholders could access and interpret the data.

From the perspective of system reliability, availability, and integrity are the most important security objectives in the Smart Grid.

Within the scope of our work, we don't take Denial of Service Attacks into consideration and we regard availability of the data contained is included in our concept of QoS. Consequently, encryption and integrity are of our major concern.

5.2.1 Security Schemes

Cryptographic algorithms and approaches are primary countermeasures against malicious attacks. Encryption algorithms can mitigate threats pertaining to confidentiality and authentication methods aims to deal with attacks against integrity of data.

- **Encryption**: Encryption is an elementary cryptographic method to achieve secure communication and information protection for any information system. In the Smart Grid, most electronic devices are expected to have at least basic cryptographic capabilities, including the ability to support symmetric ciphers (or public-key cryptography supported by low-cost hardware with embedded cryptography functionality).
  - Asymmetric key cryptography requires more computation resources than symmetric key cryptography for long key size (strong security). Thus, the use of asymmetric key encryption may be limited in embedded computing systems such as RSA.
  - Symmetric key cryptography requires approximately constant computational resources; however, it requires secure exchange and update of secret keys among network nodes, thereby complicating the process of key management.
- **Authentication** is a crucial identification process to eliminate attacks targeting data integrity. Intuitively, design of authentication for the Smart Grid can leverage existing authentication protocols in conventional networks, which have been extensively studied for decades.

5.3 Uncertainties

Classification of uncertainty is of great interest to quantify uncertainty as different uncertainties can be quantified in different ways. Generally, there are two types of uncertainty pertaining to a system: the *aleatory uncertainty* and the *epistemic uncertainty* [59]:

- **Aleatory uncertainty** is also called as variability. It is irreducible and covers uncertainty brought by stochastic procedures of physical system or the randomness of environment under consideration.

- **Epistemic uncertainty** is also termed as reducible uncertainty or subjective uncertainty. Epistemic uncertainty derives from lack of knowledge, or partial information, of the system or the surrounding environment.

Another way to categorize uncertainties depends on the different sources of uncertainties. More specifically, the uncertainty related to security of WAMC originates in several factors:

- **Simplified assumptions**: power systems are very complicated systems. In order to model the system in practice, only significant features are extracted and some conducts of the system are simplified. Modeling abstractions are inaccurate representation of the system and uncertainty arises because of that.

- **Noise of the cyber-physical system**: both cyber and physical components are susceptible to stochastic variations.

- **Human in the loop**: behavior of human beings is much more unpredictable. The

related humans could be operators or malicious attackers. Their actions can bring much variation to the system.

- **Uncertainty in the requirements**: Security is a very comprehensive concept including several factors. Moreover, there is trade-off between security and performance. So system administrators may NOT have a precise objective to construct the security scheme.

According to the definitions above, noise of the cyber-physical system can be considered as aleatory uncertainty and other types of uncertainty is epistemic uncertainty. Thus, we can select different methods to quantify the uncertainties originating in sources.

There are two kinds of uncertainty quantification problems: *forward uncertainty propagation* and *inverse uncertainty quantification* [29]:

- The forward uncertainty propagation problem relies on an observation that the variables' uncertainty would affect the uncertainty of the function depending on these variables. The propagation of uncertainty refers to the problem related to measure how this variables' uncertainty influences the uncertainty of the output of the function.

- The inverse problem primarily focuses on inferring the unknown parameters of the mathematical model based on data observed and estimates the deviation between the experiment and the mathematical model.

What we try to model lies in the area of uncertainty propagation: we model the uncertainty of the input and show the influence to the decision to choose the most fitting security

implementation. Therefore, quantifying or modeling the uncertainty of the input is of vital importance.

5.4 Quantification Framework

In this section, we use a few equations to describe the constraints on the security schemes. We denote all of the possible security schemes under consideration as $\mathcal{D}$. One security scheme can be denoted as $d \in \mathcal{D}$.

Implementation Constraints

Not all of the security schemes are feasible concerning the QoS requirements on communication model. There are some constraints from the implementation perspective. So we use $\mathcal{F}^I$ to denote the implementation constraint function.

$$\mathcal{F}^I(d) \rightarrow \{0, 1\}$$

A decision $d \in \mathcal{D}$ is valid if and only if $\mathcal{F}^I(d) = 1$.

Delay Constraints

As we introduced in Section 5.1, QoS is very significant to operation of the power system and we can determine the delay requirement with the method we introduced. The time consumption of a encryption or authentication method is a very significant concern when choosing a security scheme from the decision space as some of the algorithms may cost too much time and thwart the communication applications demanding real time data.

Security Coverage

As we previously stated, we focus on two aspects of security: *Confidentiality* and *Integrity*. We can estimate the coverage of a decision $d \in \mathcal{D}$ by measuring each of these aspects. So a function can be symbolized as follows:

$$F_x^S(d) = m_x$$

where $d \in \mathcal{D}$, Φ is the set of security aspects $\Phi = \{\text{Confidentiality}, \text{Integrity}\}$, $x \in \Phi$.

So the value of security coverage can be denoted as:

$$\mathcal{S} = \sum_{\forall x \in \Phi} w_x F_x^S(d)$$

where $w \in \mathcal{W}$ and $\mathcal{W}$ is the set of weights.

The problem is formally defined. But when we move to next step to establish the constraint and goal functions specifically, the inherent uncertainty of the system becomes a difficulty.

5.4.1 Quantifying Aleatory Uncertainty

It is straightforward to use probability distributions to quantify the aleatory uncertainty, such as the uncertainties brought by the random procedure of the system (noise).

Probability distributions are natural selections to describe aleatory uncertainty. There are generally two ways to do distribution fitting:

- **Parametric methods** [17]: the parameters of the distribution are calculated from the

data series. Usually, there are three common methods: method of moments, method of L-moments [34] and Maximum likelihood method.

- **Regression method**: using a transformation of the cumulative distribution function so that a linear relation is found between the cumulative probability and the values of the data, which may also need to be transformed, depending on the selected probability distribution. In this method the cumulative probability needs to be estimated by the plotting position.

5.4.2 Quantifying Epistemic Uncertainty

An opinion of the subjective logic is usually denoted as ω_x^A where A is the subject, also called the belief owner, and x is the proposition to which the opinion applies. The proposition x is drawn from a state space denoted as X. The propositions are mutually exclusive and disjoint set in the state space.

Assume that x is a proposition, an opinion can be denoted as a tuple (ordered) as $\omega_x = (b, d, u, a)$, where b is the positive belief that x is true, d is the negative belief that x is not true, u indicates the uncertainty with the beliefs and a is a base rate which is essentially the prior probability if there is no evidence.

The invariant of b, d and u is $b + d + u = 1$ and b, d, u, $a \in [0, 1]$. We can use some corner cases to understand the subjective logic better. $b = 1$ is essentially a binary logic $true$; vice versa $d = 1$ equals to binary logic $false$. $b + d = 1$ is actually a traditional probability

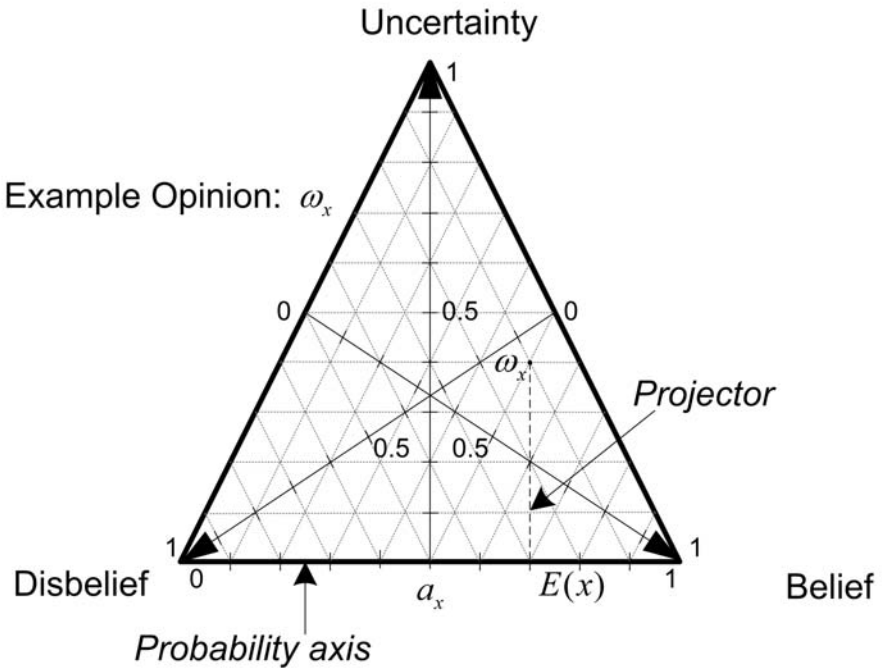

Figure 5.4: The Opinion Triangle in Subjective Logic

without uncertainty considered. $b + d < 1$ indicates that degree of uncertainty exists or is considered. $b + d = 0$ means the belief owner is totally uncertain. The expectation can be defined as $E = b + au$.

Binomial opinions can be represented on an equilateral triangle as shown in Fig 5.4 [39]. Any point within the triangle can be interpreted as the belief tuple of (b, d, u). The b, d, u-axes indicated by the *Belief*, *Disbelief* or *Uncertainty* label dash from one vertex to the opposing edge orthogonally.

Beta distributions are normally denoted as Beta(α, β) where α and β are its two pa-

rameters. The Beta distribution of a binomial opinion $\omega = (b, d, u, a)$ is the function

$$\text{Beta}(p|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}$$

where $\alpha = 2b/u + 2a$ and $\beta = 2d/u + 2(1-a)$.

By this, we can easily convert a subjective logic to a probability distribution.

5.4.3 Eliciting Opinions on Security with Uncertainty from Stakeholders

Security coverage is hard to be measured directly by mathematical models as it is essentially to measure epistemic uncertainty. So eliciting opinions on security coverage from stakeholders might be a natural and practical choice. It is commonly agreed that eliciting user's preferences in terms of complex utility functions is challenging [25].

People are usually not good at expressing opinions with numerical values. However, the verbal categories are more intuitive. In the fuzzy verbal category, opinions are measured with 2-dimensional fuzzy categories: the *likelihood* opinion and *certainty* opinion. Stakeholders can choose their opinions from the verbal category.

However, not all of the opinions in the verbal category make empirical sense. In order to rule out the improper opinions. We overlay the verbal category with a triangle in Fig 5.5. Only opinions lie within the triangle are valid. For example, $1E$ is not a valid opinion in real life as people cannot say something is absolute while completely uncertain about that. So $1E$ stands out of the gray triangle. When the stakeholders choose a value from the fuzzy category, this value can be converted into an subjective opinion if it is valid. Then, this can

Certainty Categories		Likelihood: Not Secure 9	Very Likely Insecure 8	Unlikely Secure 7	Somewhat Insecure 6	Chances about even 5	Somewhat Secure 4	Likely Secure 3	Very Likely Secure 2	Absolutely Secure 1
Completely Uncertain	**E**	9E	8E	7E	6E	5E	4E	3E	2E	1E
Very Uncertain	**D**	9D	8D	7D	6D	5D	4D	3D	2D	1D
Uncertain	**C**	9C	8C	7C	6C	5C	4C	3C	2C	1C
Slightly Uncertain	**B**	9B	8B	7B	6B	5B	4B	3B	2B	1B
Completely Certain	**A**	9A	8A	7A	6A	5A	4A	3A	2A	1A

Figure 5.5: Fuzzy Verbal Categories

be described using a beta distribution.

5.5 A Study Case

A case study has been performed to apply the proposed uncertainties quantification methods to a benchmark power system oscillation model.

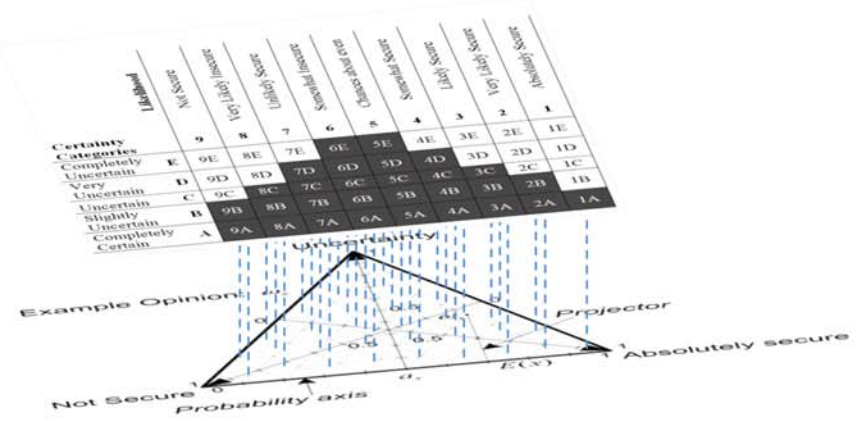

Figure 5.6: Mapping from opinion triangle to fuzzy verbal categories

5.5.1 Power System Description

The model is a two-area four-machine system as shown in the Fig 5.7. The parameters of the system can be found in [49]. In this system, bus 7, 8, and 9 are modeled as substations and called SS_{Bus7}, SS_{Bus8}, and SS_{Bus9} respectively. The study system is used for two types of power system applications uncertainties quantifying study. One type is called protection function. The other type of function in power system is monitoring function, which is out of the scope of this study since monitoring functions have lowest requirement on QoS.

For the protection function, transmission line protection is applied on the lines between SS_{Bus7} and SS_{Bus9}. The current differential line protection is chosen as a protection scheme. According to standard IEC 61850-90-1, current measurement on both SS_{Bus7} and SS_{Bus9} has to be delivered to each other via communication system. At each substation, both remote and local current measurement are compared to identify whether there is fault on line or

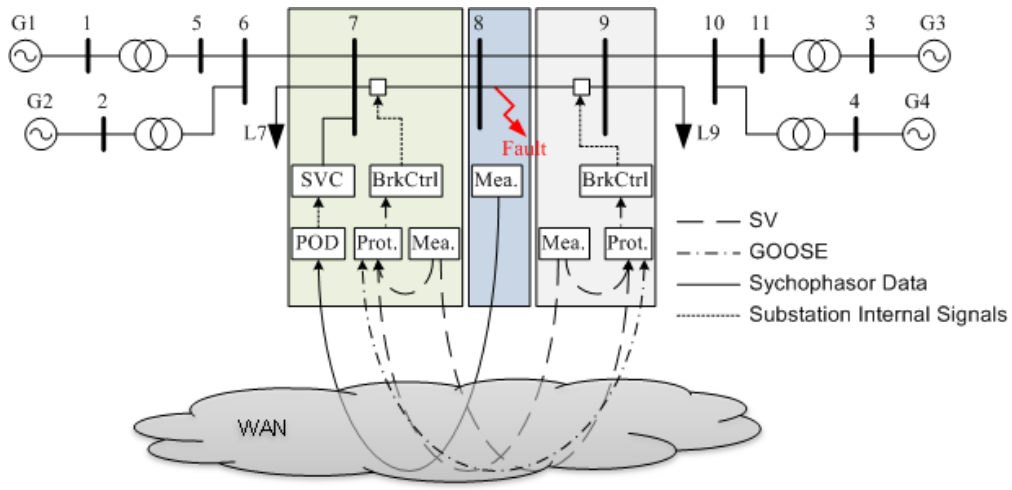

Figure 5.7: Power System used for the case study and involved communication links

not. The result of identification is used to control the breakers to isolate the line if fault occurs. The QoS requirement for this line protection function can be obtain in Table 5.2.

Therefore, the sample value is chosen as protocol for exchange current measurements between SS_{Bus7} and SS_{Bus9}. To ensure the breaker controller works properly, the interlocking function is also applied by using GOOSE message to exchange breaker status between these two substations.

Since this power system model is an inter-area oscillation damping benchmark model, it is unstable when the system runs without power system stabilizer (PSS). In order to keep the system stable, a Static Var Compensator (SVC) is deployed on $Bus7$. Normally, an SVC is used for maintaining voltage level to enable it controlling the system oscillation, a Power Oscillation Damping (POD) controller is used, see more details in [82]. The POD controller requires signal input from SS_{Bus8} in this study case. The control input signal of the POD

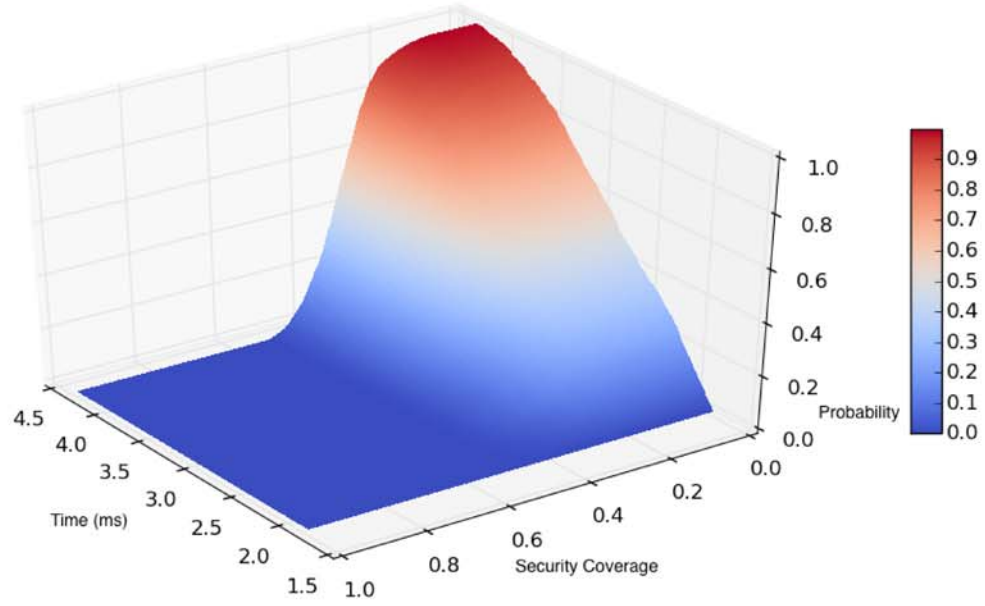

Figure 5.8: Uncertainties with DES

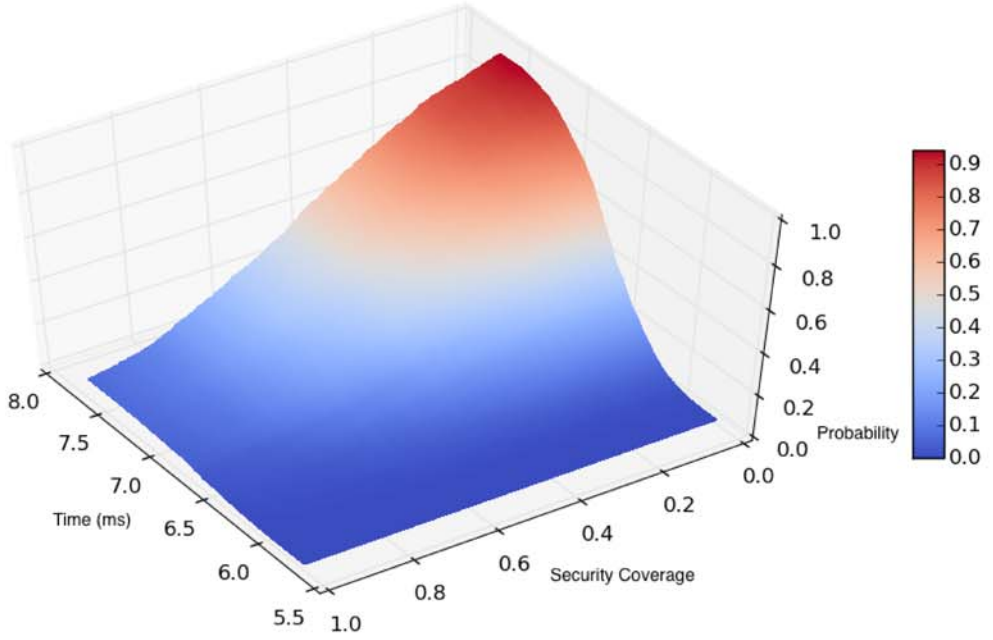

Figure 5.9: Uncertainties with The Triple DES

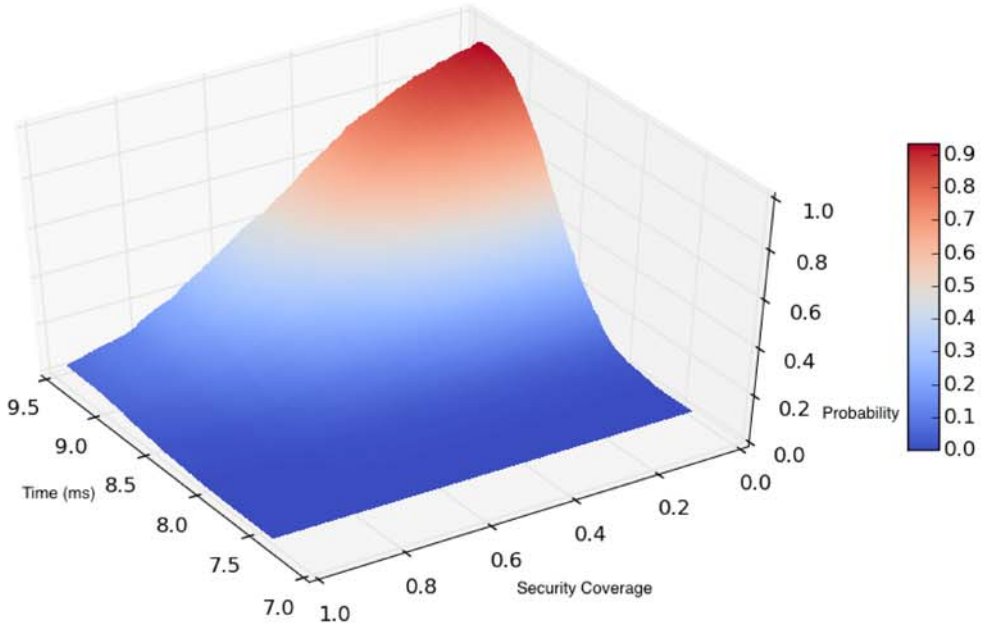

Figure 5.10: Uncertainties with AES

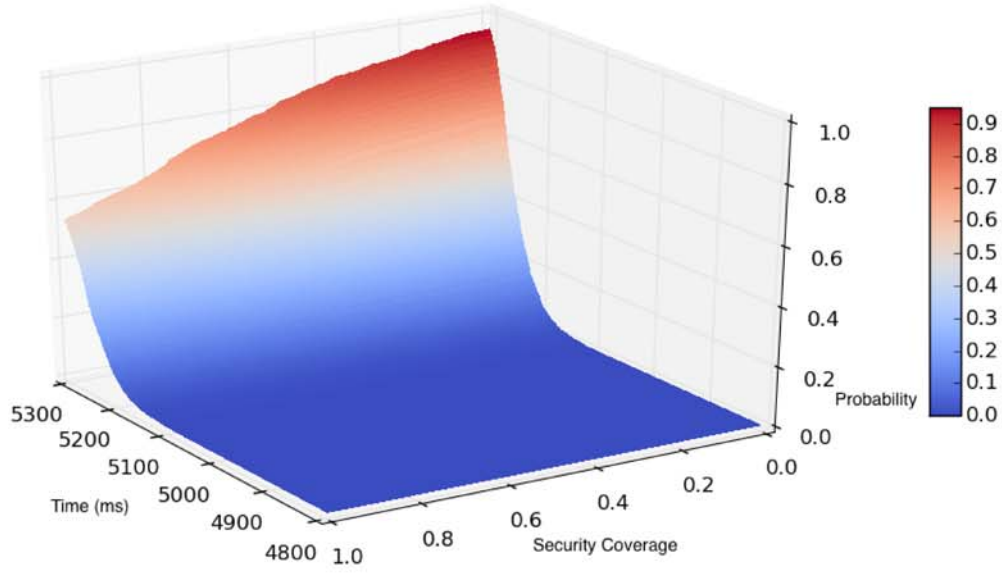

Figure 5.11: Uncertainties with RSA

is assumed as measured from one Phasor Measurement Unit (PMU) in substation SS_{Bus8}. The signal follows synchrophasor data format. From the previous study in [82], the QoS requirement can also be obtained in Table 5.3.

The measurements and status exchange in the study case is also illustrated in the Fig 5.7.

In this simplified study case, we only consider encryption algorithms including: DES, Triple DES, AES, and RSA. Details about these algorithms can be found in [74].

We use a Raspberry Pi Model B to simulate an IED and run Java implementations of these four encryption algorithms it. A set of running time data is obtained by running these algorithms repeatedly. By using the parametric method, we can derive the distributions of running time of encryption algorithms on the Raspberry Pi respectively. Opinions on the security coverage of the encryption algorithms are also drawn.

Table 5.4 lists the distributions of latency and Beta distributions denoting the subjective opinions on the security coverage of the encryption algorithms.

We make use of Monte Carlo method [70] to compute the probabilities of each combination of latency and security coverage. Fig 5.8, Fig 5.9, Fig 5.10, and Fig 5.11 are the generated graph depicting distributions corresponding to the latencies and security coverages.

For our first application scenario, the latency requirement is less than 9ms. RSA is firstly ruled out as it can never meet the latency requirement. For the remaining three algorithms, if the administrator wants the security coverage to be higher than 0.4, then the

probability for DES to satisfy both latency requirement and security requirement is 0.1, Triple DES is 0.6 and AES is 0.5. So the administrator may choose Triple DES as the solution. But if the communication delay is a little bit higher, which make the flexibility for security is less, the administrator may prefer to choose DES.

For the second application scenario (the POD controller case), the latency requirement is less than 200ms. All three symmetric key algorithms can meet the latency requirement. The administrator would favor AES as its security coverage is always better than the other algorithms when latency is not a concern.

5.6 Enabling Adaptive Cyber-security

The integral quantification method can be extended to provide better QoS assurance and cyber security for WAMC applications. We propose a novel *adaptive cyber security scheme*. It provides optimized security coverage based on real time data link QoS performance. Meanwhile, it ensures data link performance fulfilling QoS requirements of WAMC applications.

The proposed adaptive cyber security scheme is based on the architecture proposed in the Stateful Data Delivery Service (SDDS) [82], which is provided by Service Provider (SP). As illustrated in Figure 5.12, there are two SPs. One is WAMC application side SP (SP_{App}) and the other is Data Source side SP (SP_{DS}). WAMC application registers its input data QoS requirements on its SP_{App}. The data link is established between the SP_{App} and the SP_{DS}. The SP_{DS} handles encryption, authentication, and adds time stamping for data

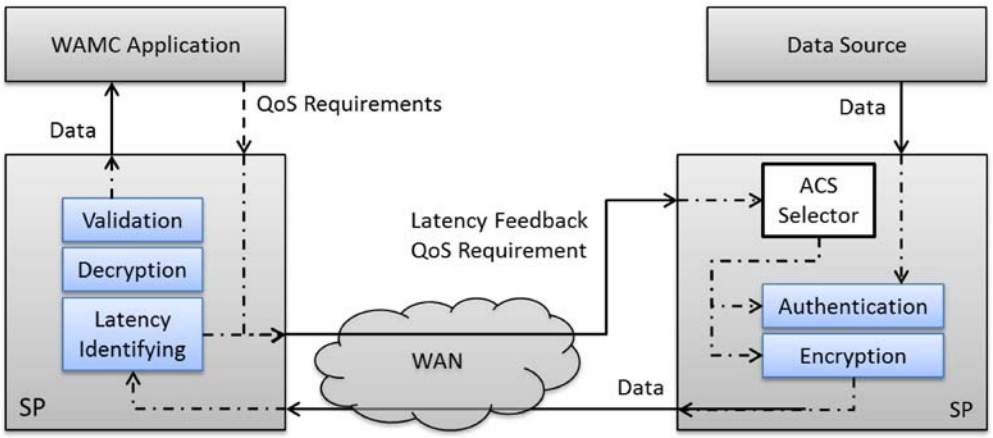

Figure 5.12: The overall architecture.

packets. The data packets are sent through WAN to the SP_{App} where the SP_{App} calculates latency, decrypts received packets, and verifies the authentication.

The measured latency is sent back from SP_{App} to SP_{DS} for cyber security scheme selection (refer to ACS Selector in Figure 5.12). The adaptive cyber security scheme implemented in ACS Selector is described in Section 5.6. Based on such architecture, WAMC applications and data sources do not need to include functions such as authentication, encryption, decryption, verification, and communication latency identification. That simplifies design of the WAMC applications.

For a specific cyber-security countermeasure algorithm running on a specific dedicated device, the maximum security coverage can be obtained once the maximum computational time and minimum probability is decided. The objective of the adaptive cyber-security

algorithm can be formed as a security coverage optimization problem as following:

$$\max C_{m,n} \tag{5.1}$$

where $C_{m,n}$ is security coverage, m represents an encryption algorithm, and n represents an authentication algorithm.

Regarding QoS requirements of a WAMC application, the following constraints can be obtained:

$$\begin{gathered} T^m_{encryp} + T^n_{authen} \leq 1/Q_{RR} \\ T^m_{encryp} + T^n_{authen} + T_{comm} \leq Q_{Latency} \\ 1 - P_{m,n} \leq Q_{PktLoss} \end{gathered} \tag{5.2}$$

where T^m_{encryp} is the time cost of encryption algorithm m, T^n_{authen} is the time cost of authentication algorithm n, T_{comm} is the communication latency which can be calculated as described before, and $P_{m,n}$ is the probability of data fulfilling the computational time and security coverage requirement when using encryption algorithm m and authentication algorithm n.

The security coverage uncertainty of encryption and authentication algorithms are quantified in a combinatorially way. Consequently, computational time for a combination of encryption and authentication is tested. Computational time for encryption algorithm m and authentication algorithm n is denoted as $T^{m,n}_{sec}$. Therefore, the constraints in equation 5.2 can be rewritten as:

$$\begin{gathered} T^{m,n}_{sec} \leq \min(1/Q_{RR}, Q_{Latency} - T_{comm}) \\ P_{m,n} \geq 1 - Q_{PktLoss} \end{gathered} \tag{5.3}$$

To reduce the response time, the optimal cyber-security countermeasure based on communication latency can be pre-processed and stored. When data link is online, an adaptive cyber security algorithm is required to lookup security countermeasure from pre-processed results based on the online communication latency monitoring. The workflow of the pre-process can be illustrated in the Figure 5.13.

The computation time distribution $\mathcal{N}(\mu, \sigma^2)$ is obtained by prior off-line testing performed on intended target platform, e.g. IED, for both data sender and receiver. The expert binomial opinions on each combination of authentication and encryption are based on the fuzzy verbal categories as shown in Figure 5.5. These opinions are regarded as security coverage and can be expressed by beta distribution denoted as $Beta(\alpha, \beta)$ as shown in Figure 5.13. In the online status, the algorithm allows the SPs to choose proper security countermeasure based on the latency monitoring result T_{comm} and lookup the encryption and authentication algorithm from the set obtained from pre-process $Set(T_{comm}, m, n)$.

In addition to selecting a cyber security algorithm, the other two crucial parts are key management and algorithm negotiation [43]. For a single data link there might be several different cyber security algorithms available. Therefore, both sender and receiver keys for all potential cyber security algorithms should be settled before measurement packets are transmitted. Key management is however out of the scope of this scheme. The algorithm negotiation occurs online and depends on communication network performance. Data receiver should be notified from data source which combination of encryption and authentication algorithm is being used to assure data can be decrypted and validated by correct algorithms.

There might be several approaches to achieve such algorithm negotiation. However this is not the main contribution of this research. Therefore, a simple payload size based approach is used here and it is explained as the following. In power system communication protocol, the length of data payload is fixed once the content of the data packet is configured. Therefore, the length of secured data (that has been encrypted and authenticated) also has the same length if the security countermeasure algorithm and key is same. Therefore, the security countermeasure algorithm can be identified by the data receiver based on the length of received data payload. Of course, it might happen that payloads of packets using different algorithms have same length. Such conflict can be identified before data link is established. To avoid such conflict, additional data padding can be added into the data packet. Once the data is received, these data padding will be removed after decryption and validation by the packet receiver.

5.6.1 Simulation

In this section, a benchmark power oscillation model has been studied. Static Var Compensator (SVC) based Power Oscillation Damping (POD) controller has been chosen as a WAMC application. The proposed adaptive cyber security scheme is implemented using different encryption and authentication algorithms.

In this simulation, encryption algorithms include: DES, Triple DES, AES, and RSA. Authentication algorithms are SHA1, SHA256, and MD5. Detail of these algorithms can be found in [74].

Computational time of each encryption and authentication combination is formed as normal distribution as $\mathcal{N}(\mu, \sigma^2)$. Each combination is deployed on a Raspberry Pi Model B (with Processor: ARM1176JZF-S 700MHz; RAM: 512MB; Computational Power: 0.041 GFLOPS) to obtain the computation time by running 10000 times algorithm for authentication, encryption, decryption, and validation. The Raspberry Pi was chosen as the implementation platform since its computational capability reflects real implementation in substation devices such as RTUs, IEDs, PMUs, or Substation Gateways. The mean value and standard deviation of each combination is given in Table 5.5.

The expert binomial opinions on each combination of authentication and encryption are given in Table 5.6 and they are expressed by beta distribution as shown in Table 5.7.

By applying the quantifying method described in our quantification framework, the probability distribution of computation time and security coverage of each algorithm combination can be obtained and the results are shown in Figure 5.14, Figure 5.15, and Figure 5.16.

The system for simulation is a two-area four-machines oscillation model as shown in Figure 5.17. The power system is unstable due to the power oscillation between two areas. The detail parameters of the system can be found in [49]. Instead of power system stabilizer, a SVC based POD controller has been deployed on bus 7. The POD controller uses remote signal measured from bus 8. Design and parameters of the POD controller can be found in [82].

In this system, Phasor Measurement Unit (PMU) data from bus 8 is used for the POD controller located on bus 7. The PMU measurement report rate is 60 packets per second.

The POD control QoS requirements are listed in Table 5.8. The requirement of end-to-end latency is obtained from previous study in [82]. Data source report rate and packet loss due to cyber security countermeasure can be regarded as a trade off for received data rate. The E_{RR} can be obtained by an iterative testing. In this study case,received data rate is obtain as $E_{RR} \geq 10(packets/second)$. Since the data source report rate is $R_{DS} = 60(packets/second)$, the packet loss requirement should be $P_{PktLoss} \leq 83.34$.

Based on the QoS requirements of the WAMC applications and probability distribution of security coverage and computation time, optimized cyber security coverage, combination of encryption and authentication, and probability of data packets fulfilling both computation time and cyber security coverage can be obtained as shown in Figure 5.18.

In the top figure, the security coverage is increased when available computation time is increased and the security coverage is always highest among different countermeasures. In the bottom figure, probability is maintained to ensure the rate of secured packets from data source to data user higher than 16.66% which fulfills the packet loss requirement from the WAMC application. The middle figure shows that the countermeasure uses SHA256 with AES (as number 9) when available computation time is less than 7ms. The countermeasure switches to SHA1 with AES (as number 5) when available computation time is larger than 7ms and less than 80ms. When available computation time is larger than 80ms, SHA256 with RSA is used as countermeasure. Even SHA256 with RSA is evaluated as high security coverage as shown in Table 5.6, it is not chosen by the adaptive cyber security scheme when available computation time is less than 80ms. Because high computation time of SHA256

with RSA might lead to latency of data violating the requirement of the POD application.

The result presented above is restricted to the data sender and receiver with similar capability as justify with similarity to IED compuational capabilities. Sender and receiver with different computation capability might lead to different result. However, the adaptive cyber security scheme presented here is applicable to different hardware by simply replacing test result in Table 5.5 and completing the rest procedure. From the result, RSA requires longer computation time comparing with other encryption algorithm. And its computation time violates report rate and latency requirements in this study case due to the hardware limitation. But by update data sender and receiver hardware, RSA can be a feasible solution to the same data link.

Table 5.1: Power System Communication Protocol Routing Scheme and Communication Mode

Message	Routing Scheme	Communication Mode
IEEE C37.118 SynchroPhasor Data	Multicast	Periodically Push
IEC61850 MMS Server Client	Unicast	Event Driven Poll
IEC61850 MMS Report	Multicast	Measurement Event Driven Push
IEC61850 GOOSE	Multicast	Combined Periodical and Event Driven Push
IEC61850 Sampled Value	Multicast	Periodically Push
IEEE 1815-2012 DNP3	Unicast	Event Driven Poll
IEC60870-5-104	Unicast	Event Driven Poll

Table 5.2: Line Differential Protection QoS Requirement

Item	Requirement
Report Rate	at least 12 data per power cycle
Latency	less than 8ms
BER	less than 10^{-6}s
Time Synchronization	less than 0.1ms

Table 5.3: POD Controller QoS Requirement

Item	Requirement
Report Rate	60 packets per second
Latency	less than 200ms

Table 5.4: Latency Distribution and Opinions on Security Coverage of four Encryption Algorithms

ALgorithms	Latency	Opinions on Security	Distribution
DES	$\mathcal{N}(2.66, 0.19)$	$5D$	$Beta(1, 1.5)$
Triple DES	$\mathcal{N}(6.68, 0.66)$	$4C$	$Beta(4, 4)$
AES	$\mathcal{N}(8.38, 0.65)$	$3B$	$Beta(5, 3)$
RSA	$\mathcal{N}(5324.45, 305.47)$	$2B$	$Beta(16, 2)$

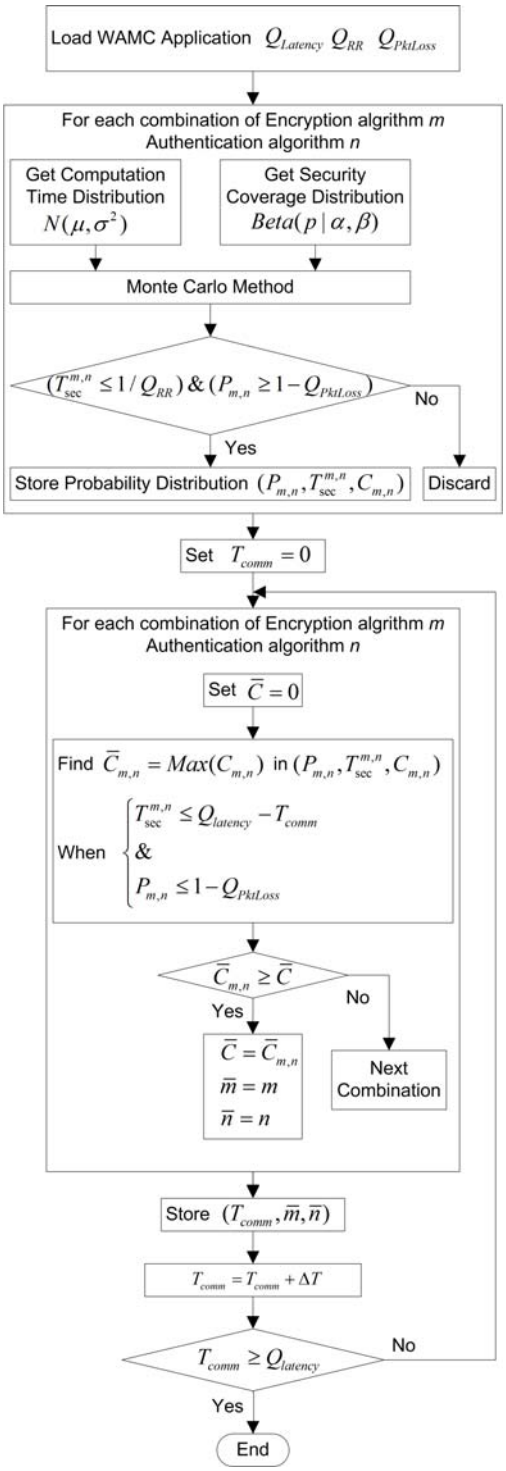

Figure 5.13: Pre-process work flow of adaptive cyber security algorithm.

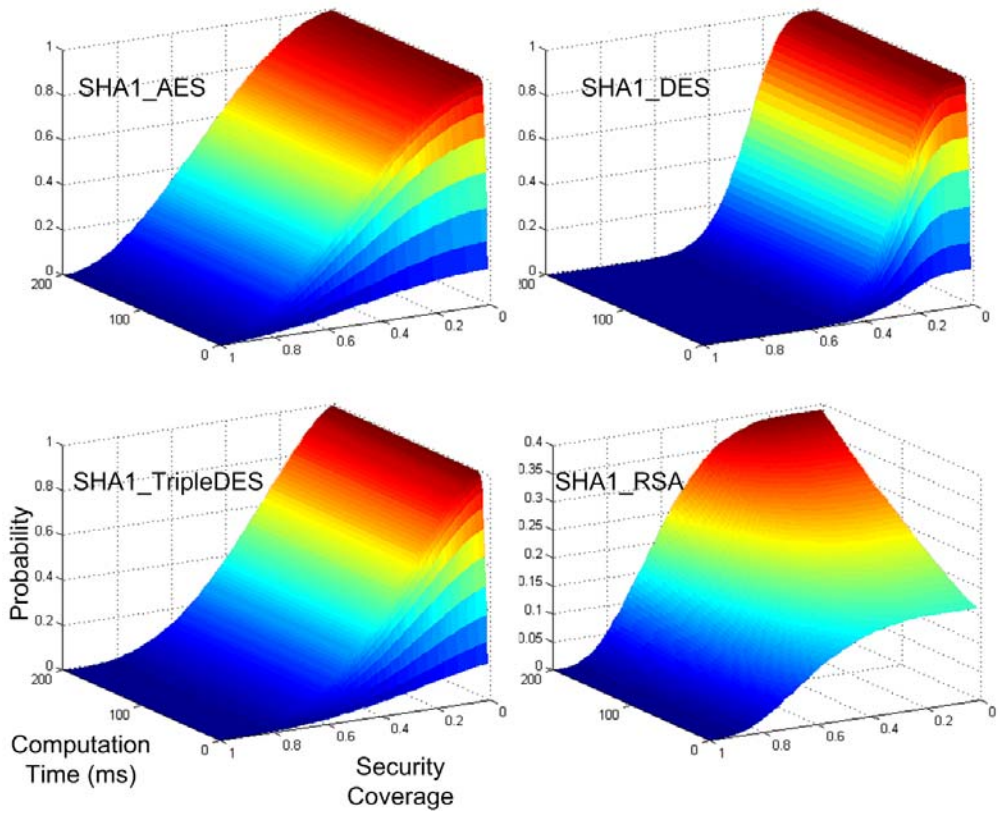

Figure 5.14: Probability Distribution for different encryption algorithms when authentication is SHA1.

Table 5.5: Normal Distribution Parameters for Computation Time (in millisecond)

		AES	**DES**	**TripleDES**	**RSA**
2*__SHA1__	μ	2.44	2.45	2.95	283.11
	σ^2	1.48	1.09	1.16	5.16
2*__SHA256__	μ	2.13	2.35	2.86	282.77
	σ^2	4.41	0.62	0.67	2.22
2*__MD5__	μ	2.09	2.35	2.88	283.00
	σ^2	0.60	0.62	0.67	7.32

Table 5.6: Expert Opinions on Security Coverage

	AES	**DES**	**TripleDES**	**RSA**
SHA1	5C	7B	6C	4B
SHA256	3B	6B	5C	2A
MD5	6C	8B	8C	5C

Table 5.7: Beta Distribution Parameters for Security Coverage

		AES	**DES**	**TripleDES**	**RSA**
2*SHA1	α	2.00	5.00	1.67	5.00
	β	2.00	13.00	3.00	3.00
2*SHA256	α	4.50	4.00	2.00	14.00
	β	3.50	7.30	2.00	4.00
2*MD5	α	1.50	3.00	1.50	2.33
	β	1.75	15.00	6.50	2.33

Table 5.8: POD controller QoS Requirements

QoS Metrics	**QoS Requirement**
End-to-end latency	less than 200 ms
Measurement Report Rate	60 packets/s
Maximum Packet loss	less than 83.34%

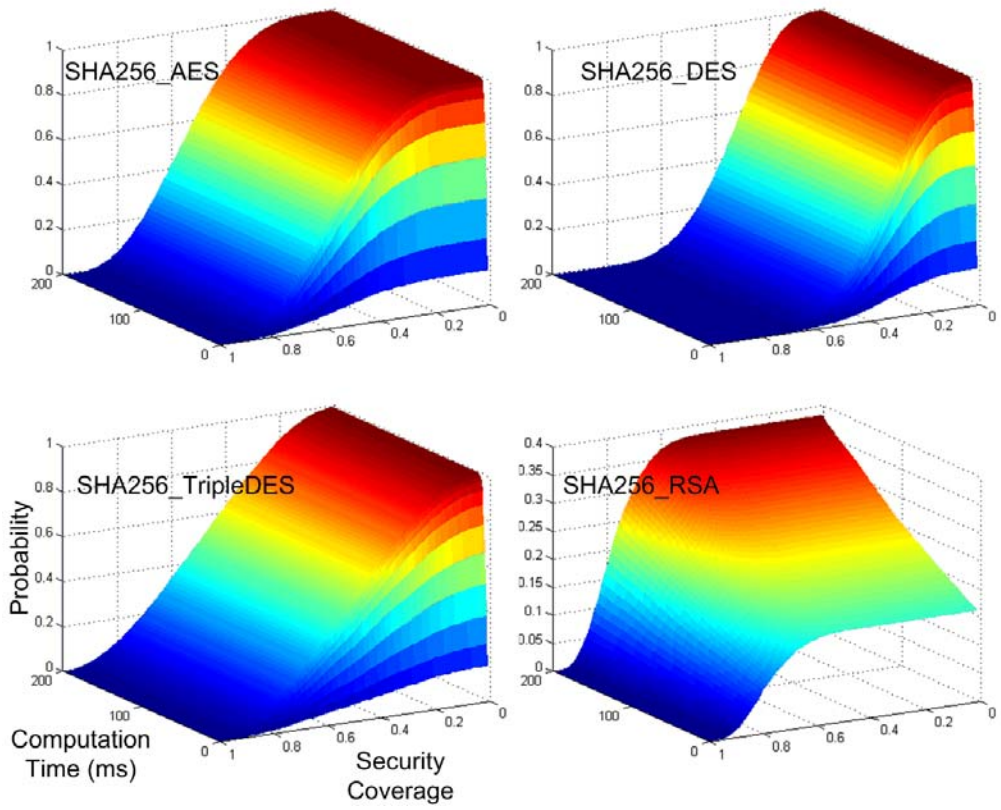

Figure 5.15: Probability Distribution for different encryption algorithms when authentication is SHA256.

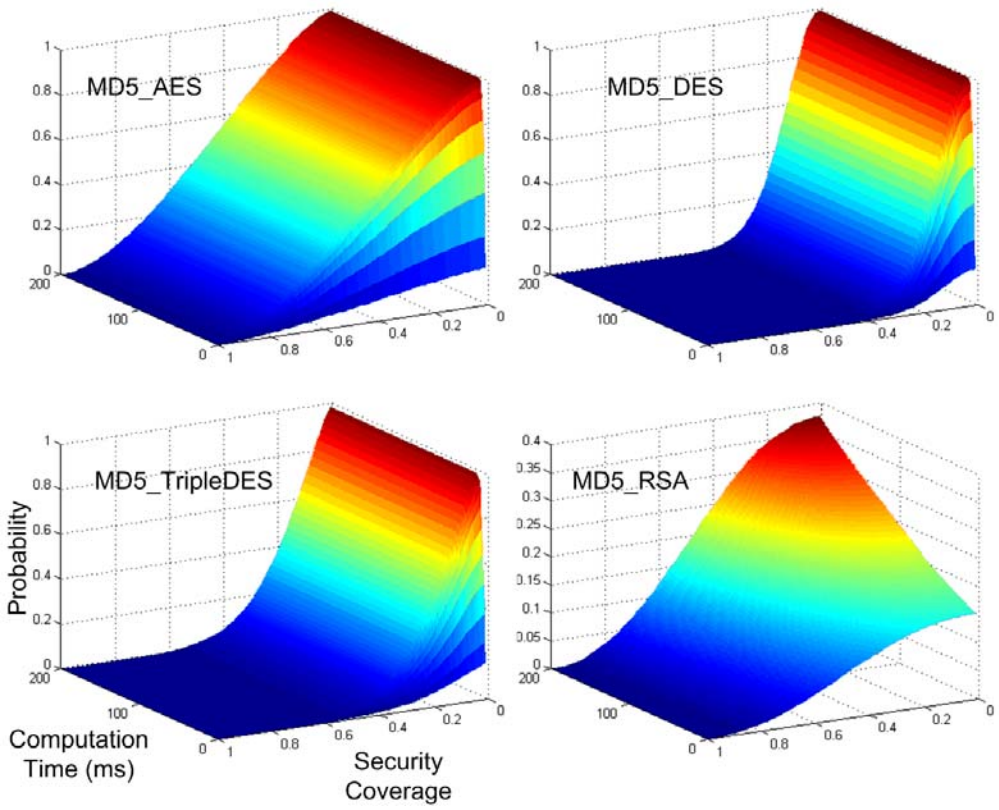

Figure 5.16: Probability Distribution for different encryption algorithms when authentication is MD5.

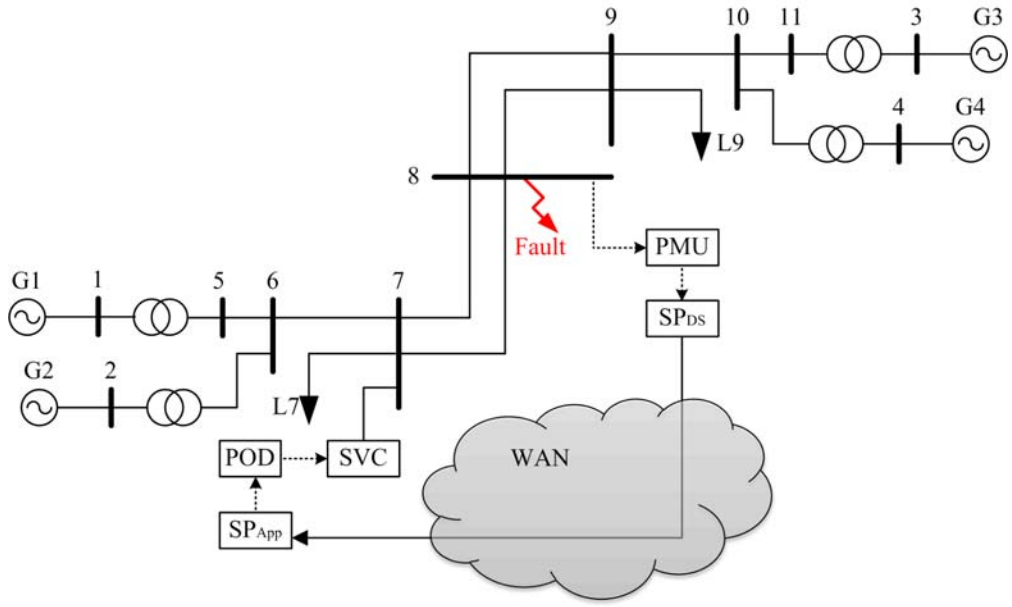

Figure 5.17: Studied power system model.

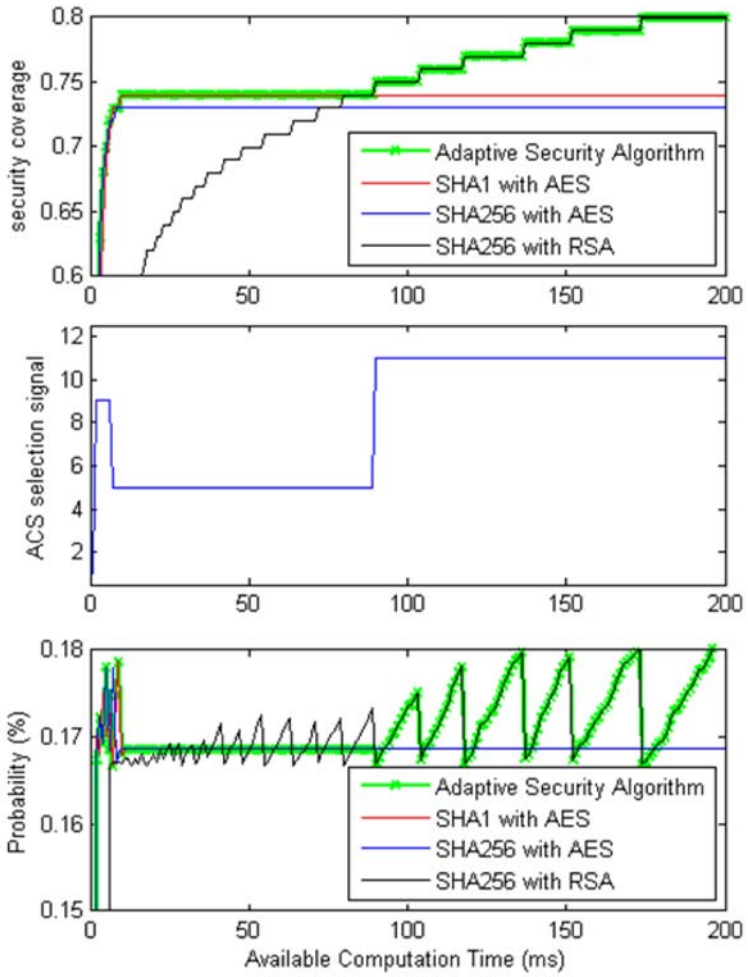

Figure 5.18: Security coverage, countermeasure, and probability vs. available cyber security computation time.

CHAPTER 6. CONCLUSIONS

As a concrete example of uncertainty in pragmatic smart power grid system, we investigate the security implications of two common transport layer protocols (TCP and UDP) applied to synchrophasor data communication between PMUs and PDCs. Transport Layer Security, on top of TCP protecting the payload data, can mitigate false data injection attacks but is still vulnerable to DoS attacks if either packets can be sniffed or sequence numbers can be inferred. On one hand we have shown that data injection attacks on plain TCP and UDP data streams are a practical way (from the perspective of an attacker) to inject false data to applications using those data, without having to physically compromise the measurement units themselves. On the other hand, we have evaluated the protection against such attacks provided by TLS and DTLS, showing that both block false data injection attacks. However, an attacker who can successfully do false data injection on a plain TCP stream can succeed in a DoS attack on a TLS stream. Schemes of incorporation uncertainty with decision making, modeling trust as a meter of uncertainty, measuring uncertainty in choosing most fitting security protocol in power grid system and enabling adaptive security are proposed.

A framework is described for incorporating trust into the decision making processes associated with control of large-scale critical infrastructure systems. Our framework is based on the Bayesian paradigm. The risk function, prior distribution and the distribution of evidence are three components of the Bayesian paradigm. We used the prior distribution to model subjectivity of trustors and showed how it could be combined with newly-acquired

evidence and the derived Bayes risk function to obtain a decision rule by minimizing the risk function.

Then trust as an indicator of uncertainty is studied and modeled. We propose a definition of trust that is suitable for applications in critical infrastructures. Based on the definition, we build a mathematical framework that is capable of adapting to situations with different trust policies and prior information. Specifically, we proposed a model that fits cases with a set of known expectations (PE model) and another model which is suitable to applications without foregone expectations available (SE model).

Through the analysis of security implications of transport layer protocols, we realize that choosing the most fitting security scheme is a great challenge in practical system development and deployment. Appropriately quantifying uncertainty and its effects on security is a key to choosing a sufficiently secure communication system for power grid while meeting QoS requirements. We provide a framework to investigate QoS and security of power grid communication system in an integral way by quantifying different types of uncertainties associated with them. We use probability distributions to capture the characteristics of delay for security overhead and apply subjective logic to describe the stakeholders opinions on security coverage of security schemes. The Monte Carlo method is employed to produce a unified view of the overall uncertainty with these two aspects. Based on the uncertainty quantification framework, we move one step forward to propose a novel adaptive cyber security scheme for WAMC applications. Proposed adaptive cyber security scheme takes both QoS requirements of WAMC applications into account. Compared with traditional fixed cyber

security countermeasure, adaptive cyber security scheme has the following advantages: dynamically optimizing security coverage and reducing the probability that QoS violation due to the cyber security countermeasure. A heuristic cyber security countermeasure algorithm synchronization solution has been illustrated. A simulation is performed to demonstrate the feasibility of proposed adaptive cyber security scheme.

BIBLIOGRAPHY

[1] IEEE Standard for Synchrophasor Measurements for Power Systems. *IEEE Std C37.118.2-2011 (Revision of IEEE Std C37.118-2005)*, pages 1–61, 2011.

[2] A. Abdul-Rahman and S. Hailes. A distributed trust model. In *Proceedings of the 1997 Workshop on New Security Paradigms*, NSPW '97, pages 48–60, New York, NY, USA, 1997. ACM.

[3] N. Al Fardan and K. Paterson. Lucky Thirteen: Breaking the TLS and DTLS Record Protocols. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 526–540, May 2013.

[4] D. Anderson, C. Zhao, C. Hauser, V. Venkatasubramanian, D. Bakken, and A. Bose. A Virtual Smart Grid: Real-Time Simulation for Smart Grid Control and Communications Design. *IEEE Power and Energy Magazine*, 10(1):49–57, 2012.

[5] G. Apostolakis. The Concept of Probability in Safety Assessments of Technological Systems. *Science*, 250(4986):1359–1364, 1990.

[6] D. Bakken, A. Bose, C. Hauser, D. Whitehead, and G. Zweigle. Smart generation and transmission with coherent, real-time data. *Proceedings of the IEEE*, 99(6):928–951, June 2011.

[7] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, 2nd edition edition, 1993.

[8] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. The role of trust management in distributed systems security. In J. Vitek and C. Jensen, editors, *Secure Internet Programming*, volume 1603 of *Lecture Notes in Computer Science*, pages 185–210. Springer Berlin Heidelberg, 1999.

[9] E. Chang, P. Thomson, T. Dillon, and F. Hussain. The fuzzy and dynamic nature of trust. *Trust, Privacy, and Security in Digital Business*, pages 161–174, 2005.

[10] B. Chaudhuri, R. Majumder, and B. Pal. Wide-area measurement-based stabilizing control of power system considering signal transmission delay. *IEEE Transactions on Power Systems*, 19(4):1971–1979, Nov 2004.

[11] P. Chen, S. Cheng, and K. Chen. Smart Attacks in Smart Grid Communication Networks. *Communications Magazine, IEEE*, 50(8):24–29, August 2012.

[12] M. Chenine and L. Nordström. Modeling and Simulation of Wide-Area Communication for Centralized PMU-Based Applications. *IEEE Transactions on Power Delivery*, 26(3):1372–1380, 2011.

[13] M. Chenine, J. Ullberg, L. Nordström, Y. Wu, and G. Ericsson. A Framework for Wide-Area Monitoring and Control Systems Interoperability and Cybersecurity Analysis. *Power Delivery, IEEE Transactions on*, 29(2):633–641, April 2014.

[14] J. H. Chow and S. G. Ghiocel. *An Adaptive Wide-Area Power System Controller using Synchrophasor Data*. Springer, 2012.

[15] B. Christianson and W. S. Harbison. Why isn't trust transitive? *Lecture Notes in Computer Science*, 1189:171–176, 2005.

[16] Y.-H. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick, and M. Strauss. Referee: Trust management for web applications. *World Wide Web J.*, 2(3):127–139, June 1997.

[17] H. Cramr. *Mathematical Methods of Statistics*. Princeton University Press, 1999.

[18] K. R. Davis, K. L. Morrow, R. Bobba, and E. Heine. Power flow cyber attacks and perturbation-based defense. In *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, pages 342–347, Nov 2012.

[19] G. Dan and H. Sandberg. Stealth Attacks and Protection Schemes for State Estimators in Power Systems. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 214–219, Oct 2010.

[20] J. P. Degabriele and K. G. Paterson. On the (in)Security of IPsec in MAC-then-encrypt Configurations. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 493–504, New York, NY, USA, 2010. ACM.

[21] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, RFC Editor, Aug 2008.

[22] I. Dionysiou. *Dynamic and Composable Trust for Indirect Interactions*. PhD thesis, Washington State University, 8 2006.

[23] I. Dionysiou, D. Frincke, D. Bakken, and C. Hauser. An approach to trust management challenges for critical infrastructures. In J. Lopez and B. Hmmerli, editors, *Critical Information Infrastructures Security*, volume 5141 of *Lecture Notes in Computer Science*, pages 173–184. Springer Berlin Heidelberg, 2008.

[24] N. Doraswamy and D. Harkins. *IPSec*. Prentice Hall, 2nd edition edition, 2003.

[25] N. Esfahani, E. Kouroshfar, and S. Malek. Taming uncertainty in self-adaptive software. In *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, pages 234–244, New York, NY, USA, 2011. ACM.

[26] N. Ferguson and B. Schneier. A cryptographic evaluation of IPsec. Technical report, Counterpane Internet Security, Inc, 2000.

[27] S. French. *Decision Theory: An Introduction to the Mathematics of Rationality* . Ellis Horwood, Ltd., 1st edition edition, 1986.

[28] K. Gajrani, K. Sharma, and A. Bhargava. Performance Assessment of Communication Network in WAMS. *International Journal of Distributed and Parallel Systems*, 3(6):127–137, 2012.

[29] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *International Journal for Numerical Methods in Engineering*, 9:15811608, September 2009.

[30] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford University Press, 3rd edition, 2001.

[31] J. T. Hagen and B. E. Mullins. TCP Veto: A Novel Network Attack and its Application to SCADA Protocols. In *IEEE PES Innovative Smart Grid Technologies*, pages 1–6, 2013.

[32] B. Harris and R. Hunt. TCP/IP Security Threats and Attack Methods. *Computer Communications*, 22(10):885 – 897, 1999.

[33] C. Hauser, D. Bakken, and A. Bose. A Failure to Communicate: Next Generation Communication Requirements, Technologies, and Architecture for the Electric Power Grid. *Power and Energy Magazine, IEEE*, 3(2):47–55, Mar 2005.

[34] J. R. M. Hosking. L-moments: Analysis and estimation of distributions using linear combinations of order statistics. *Journal of the Royal Statistical Society. Series B (Methodological)*, 52:105–124, 1990.

[35] R. G. L. Indra Mohan Chakravarti and J. Roy. *Handbook of Methods of Applied Statistics*, volume 1. John Wiley and Sons Inc., 1967.

[36] Information Sciences Institute. Transmission Control Protocol. RFC 793, RFC Editor, Sep 1981.

[37] J. Jack and D. P. Green. Presidential Leadership and the Resurgence of Trust in Government. *British Journal of Political Science*, 16:143–53, 1986.

[38] L. Joncheray. Simple Active Attack Against TCP. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, Salt Lake City, Utah USA, June 1995.

[39] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9:279–311, June 2001.

[40] A. Jøsang, R. Hayward, and S. Pope. Trust network analysis with subjective logic. In *Proceedings of the 29th Australasian Computer Science Conference - Volume 48*, ACSC '06, pages 85–94, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[41] D. Jost. A Constructive Analysis of IPsec, 2014.

[42] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 640–651, New York, NY, USA, 2003. ACM.

[43] C. Kaufman, R. Perlman, and M. Speciner. *Network Security: Private Communication in a Public World, Second Edition*. Prentice Hall Press, Upper Saddle River, NJ, USA, second edition, 2002.

[44] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301, IETF, Dec. 2005.

[45] 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, Jul 2013.

[46] D. Koller and A. Pfeffer. Probabilistic Frame-based Systems. In *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, AAAI '98/IAAI '98, pages 580–587, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.

[47] O. Kosut, L. Jia, R. Thomas, and L. Tong. On Malicious Data Attacks on Power System State Estimation. In *Proceedings of 45th International Universities Power Engineering Conference (UPEC)*, pages 1–6, 2010.

[48] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[49] P. Kundur. *Power System Stability and Control*. McGraw-Hill, the epri power system engineering series edition, 1993.

[50] 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*, pages 21–32, New York, NY, USA, 2009.

[51] Y. Liu, M. K. Reiter, and P. Ning. False Data Injection Attacks Against State Estimation in Electric Power Grids. *ACM Transactions on Information and System Security (TISSEC)*, 14(1):13:1–13:33, June 2011.

[52] N. Luhmann. *Trust and Power.* John Wiley and Sons Inc, 1969.

[53] K. Martin. Synchrophasor Standards Development - IEEE C37.118 & IEC 61850. In *44^{th} Hawaii International Conference on System Sciences*, HICSS '11, pages 1–8, 2011.

[54] Y. Mo, T.-H. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli. Cyber-physical security of a smart grid infrastructure. *Proceedings of the IEEE*, 100(1):195–209, Jan 2012.

[55] N. Modadugu and E. Rescorla. The Design and Implementation of Datagram TLS. In *In Proceedings of Network and Distributed System Security (NDSS) Symposium*, Feb 2004.

[56] S. Mohagheghi, J. Stoupis, and Z. Wang. Communication protocols and networks for power systems-current status and future trends. In *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*, pages 1–9, 2009.

[57] M. Mokhtari, F. Aminifar, D. Nazarpour, and S. Golshannavaz. Wide-area power oscillation damping with a fuzzy controller compensating the continuous communication delays. *IEEE Transactions on Power Systems*, 28(2):1997–2005, May 2013.

[58] R. T. Morris. A Weakness in the 4.2BSD Unix TCP/IP Software, 1985.

[59] W. L. Oberkampfa, J. C. Heltonb, C. A. Joslync, S. F. Wojtkiewiczd, and S. Fersone. Challenge problems: uncertainty in system response given uncertain parameters. *Reliability Engineering and System Safety*, 85(1-3):11–19, July - September 2004.

[60] E. Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[61] F. Pasqualetti, F. Dorfler, and F. Bullo. Cyber-physical Attacks in Power Networks: Models, Fundamental Limitations and Monitor Design. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 2195–2201, Dec 2011.

[62] K. Paterson and A. Yau. Cryptography in Theory and Practice: The Case of Encryption in IPsec. In *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 12–29. Springer Berlin Heidelberg, 2006.

[63] J. Pratt, H. Raiffa, and R. Schlaifer. *Introduction to Statistical Decision Theory*. The MIT Press, 1st edition edition, 1995.

[64] R. Preece, J. Milanovic, A. Almutairi, and O. Marjanovic. Damping of inter-area oscillations in mixed ac/dc networks using wams based supplementary controller. *IEEE Transactions on Power Systems*, 28(2):1160–1169, May 2013.

[65] Z. Qian and Z. Mao. Off-path TCP Sequence Number Inference Attack - How Firewall Middleboxes Reduce Security. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 347–361, May 2012.

[66] Z. Qian, Z. M. Mao, and Y. Xie. Collaborative TCP Sequence Number Inference Attack: How to Crack Sequence Number Under a Second. In *Proceedings of the 2012*

ACM Conference on Computer and Communications Security, CCS '12, pages 593–604, 2012.

[67] L. Rasmusson and S. Jansson. Simulated social control for secure Internet commerce. In *Proceedings of the 1996 workshop on New security paradigms*, NSPW '96, pages 18–25, New York, NY, USA, 1996. ACM.

[68] M. Ritwik, G. B., G. V., and A. M. Closed loop simulation of communication and power network in a zone based system. *Electric Power Systems Research*, 95(0):247 – 256, 2013.

[69] C. Robert. *The Bayesian Choice*. Springer-Verlag New York, 2nd edition, 2007.

[70] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. Wiley-Interscience, 2nd edition edition, 2007.

[71] E. Solum. Achieving over-the-wire configurable confidentiality, integrity, authentication and availability in gridstat's status dissemination, Dec. 2007.

[72] V. Sood, D. Fischer, J. Eklund, and T. Brown. Developing a Communication Infrastructure for the Smart Grid. In *Electrical Power Energy Conference (EPEC), 2009 IEEE*, pages 1–7, Oct 2009.

[73] S. Srivatsan, M. L. Johnson, and S. M. Bellovin. Simple-VPN: Simple IPsec Configuration. 2010.

[74] W. Stallings and L. Brown. *Computer Security: Principles and Practice*. Prentice Hall, 3rd edition edition, 2014.

[75] A. Teixeira, S. Amin, H. Sandberg, K. Johansson, and S. Sastry. Cyber Security Analysis of State Estimators in Electric Power Systems. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5991–5998, Dec 2010.

[76] J. S. Tiller. *A Technical Guide to IPSec Virtual Private Networks*. Auerbach Publications, 1st edition edition, 2000.

[77] S. Wang, W. Gao, J. Wang, and J. Lin. Synchronized sampling technology-based compensation for network effects in wams communication. *IEEE Transactions on Smart Grid*, 3(2):837–845, June 2012.

[78] W. Wang and Z. Lu. Cyber security in the smart grid: Survey and challenges. *Computer Networks*, 57(6):1344–1371, April 2013.

[79] Y. D. Wang and H. H. Emurian. An Overview of Online Trust: Concepts, Elements, and Implications. *Computers in Human Behavior*, 21(1):105–125, 2005.

[80] L. R. Wheeless and J. Grotz. The Measurement of Trust and its Relationship to Self-disclosure. *Human Communication Research*, 3(3):250–257, 1977.

[81] H. Wu, K. Tsakalis, and G. Heydt. Evaluation of time delay effects to wide-area power system stabilizer design. *IEEE Transactions on Power Systems*, 19(4):1935–1941, Nov 2004.

[82] Y. Wu, D. Babazadeh, and L. Nordstrom. Stateful data delivery service for wide area monitoring and control applications. In *Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on*, pages 768–773, June 2014.

[83] Y. Yan, Y. Qian, H. Sharif, and D. Tipper. A Survey on Cyber Security for Smart Grid Communications. *Communications Surveys Tutorials, IEEE*, 14(4):998–1010, 2012.

[84] D.-Y. Yeung and C. Chow. Parzen-Window Network Intrusion Detectors. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 4, pages 385–388, 2002.

[85] L. C. Young and I. F. Wilkinson. The Role of Trust and Co-operation in Marketing Channels: A Preliminary Study. *European Journal of Marketing*, 23(2):109–122, 1989.