ACCOUNTABILITY IN

SMART GRID AND

MEDICAL SENSOR NETWORK

by

JING LIU

YANG XIAO, COMMITTEE CHAIR

SUSAN V. VRBSKY
JINGYUAN ZHANG
XIAOYAN HONG
SHUHUI LI

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2013

ABSTRACT

Although advanced cyber security technology has protected every level of current network infrastructure, vulnerabilities continue to emerge after new functions are added. As a complement, accountability is required to further secure the network in terms of privacy, integrity, and confidentiality. Even if a security issue presents itself, the built-in accountability mechanism will find out who is responsible for it. This dissertation mainly studies existing technologies of accountability and tries to address several important cyber security issues using these techniques.

One specific problem has been raised in smart grids. As we know, power utility company charges customers solely based on readings from their power meters. Considering operating cost, the utility just measures aggregated power supply to a service area. Once a meter is compromised by cyber attacks, the utility can hardly find it out and thus may have economic loss. To make the smart grid more reliable, we proposed accountable metering systems in both home area and neighborhood area networks. Analysis and simulation results show that abnormal meters could be effectively identified under certain reasonable assumptions.

Another case is the medical sensor network (MSN). In this context, patients are deployed with medical sensors and wearable devices and are remotely monitored by professionals. Since it is an economical way to reduce healthcare costs and save medical resources, we expect a robust, reliable, and scalable MSN in the near future. However, the time signal and temporal history in current MSN are vulnerable due to unsecured infrastructure and transmission strategies.

Meanwhile, the MSN may leak patients' identifications or other sensitive information that violates personal privacy. To make sure the correctness of critical time signal, we presented two temporal accountability schemes for the MSN. In the meantime, these schemes also provide privacy-preserving ability.

DEDICATION

This dissertation is dedicated to my son, wife, and parents.

ACKNOWLEDGMENTS

I must give my high, respectful gratitude to my major advisor and committee chair, Dr. Yang Xiao, for his encouraging guidance, exceptional advice in my research, moral and financial support through my PhD studies. His strength and belief in me, allowed me to grow and develop the study to a completion.

I would like to thank all my dissertation committee members, Dr. Susan Vrbsky, Dr. Jingyuan Zhang, Dr. Xiaoyan Hong, and Dr. Shuhui Li for their advice and support of both my dissertation and my academic progress.

I would also like to thank the Department of Computer Science and the Office of Institutional Research & Assessment at UA, which offered me graduate assistant positions and fostered my development in the field of Computer Science. This work was partially supported by the US National Science Foundation (NSF) under grant numbers: CNS-0737325, CNS-0716211, CCF-0829827, and CNS-1059265.

In addition, I would like to express my eternal appreciation towards my parents and family who have always been there for me no matter where I am, for all unconditional supports and patience. Thank you for being ever so understanding and supportive. Also, to my wife, Vivian, thank you for being around, for brings our lovely son Elvin to my life, and for never ending love I have been getting all this while.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

People never stop questioning security issues of a system since it came out in the first place. Albeit a well design could eliminate most threats, vulnerabilities still emerge after new technologies are adopted. Instead of fixing endless security problems, identifying and tracing back misbehavior entities are required to secure a system. This idea is so called accountability. Unlike conventional security attributes, such as confidentiality, integrity, and availability, it is an important but not mature technology that mainly focuses on taking responsibility for what has been done and when it happens.

Generally speaking, in a computer system, accountability may be referred to holding a user accountable for all his/her actions on this computer. Actions could be installing new software or accessing local database. In a computer network, accountability means that the system is recordable and traceable, thus making it liable to those communication principles for its actions. Every change in a local host or network traffic, which may be the most important or most desirable information, can be used as evidence in future judgment. Under such a circumstance, no one can deny their actions, not even the administrators or other users with high privileges. Together with some suitable punishments or laws in the real world, this will prevent a number of attacks. Compared with current cyber security technologies, this concept should be widely accepted in the near future, from system design to application implementation.

In this dissertation, we will study the existing accountability technologies in computer networks. We also explore the vulnerabilities of two modern systems – smart grids and medical sensor networks – and try to address some problems using accountability techniques.

1.1 Motivation

While technology and innovation continue to modernize industry, our electric power system has been maintained in the same way for decades. The increasing load and consumption demands increase electricity complications, such as voltage sags, black outs, and overloads. Meanwhile, the current electrical network contributes greatly to carbon emissions. The United States' power system alone takes up 40% of all nationwide carbon dioxide emissions [46]. Considering both economic and environmental interests, substantial changes must be made to such an unstable and inefficient system. Therefore, many nations (e.g., U.S., EU, Canada, China, Australia, South Africa, etc.) are now modernizing their power grids [42]. They believe that it not only requires reliability, scalability, manageability, and extensibility, but also that it should be secure, interoperable, as well as cost-effective. Such electric infrastructure is referred to as "smart grid." Generally, smart grid is a promising power delivery infrastructure integrated with bi-directional communication technologies that collects and analyzes data captured in near-real-time, including power consumption, distribution, and transmission [2]. According to this data, the smart grid can provide predictive information and relevant recommendations to all stakeholders, including utilities, suppliers, and consumers, regarding the optimizing of their power utilization [2]. By two-way electrical flow, consumers are able to sell their surfeit energy back to utilities [2].

Smart grid is a complex system of systems. Deploying such a system has enormous and far-reaching technical and social benefits. Nevertheless, increased interconnection and integration also introduce cyber-vulnerabilities into the grid. Based on experiences gained from developed IT and telecommunication systems, we know that the envisioned grid will be a potential target for malicious, well-equipped, and well-motivated adversaries [13, 15]. In addition, increased connectivity of the grid will enable personal information collection, which may invade consumer privacy [12, 14, 44, 45]. Failure to address these issues will hinder the modernization of the existing power system. This dissertation will give an overview of relevant cyber security and privacy issues in the smart grid. An important security issue regarding metering system will be discussed. To design an accountable metering system in smart grid is one of our thrusts in this dissertation.

Another motivation is derive from a medical system. Heart disease, also known as cardiovascular disease, continues to be the leading cause of death worldwide and the top killer in the U.S. It is also the single largest cause of mortality in the western world. Specifically, coronary artery disease (CAD), a typical heart disease, kills an estimated 459,000 Americans every year [72]. No country wants to spend money on healthcare delivery as much as the U.S. does, whose overall healthcare expenditures tallied $1.8 trillion (about 45 million uninsured) in 2004. The American Heart Association reports that healthcare will cost over 20% of the U.S. GDP nowadays. At present, more and more of the elderly go to nursing homes. We need a regional (e.g., within a nursing home) and low-cost medical delivery system to monitor the status of patients automatically.

Medical sensor network (MSN) is one of such wireless telemedicine platform. In this context, patients are deployed with certain medical sensors and wearable devices and are

remotely monitored by professionals. Thus, seeing a doctor in person is no longer the only option for those in need of medical care. Since it is also an economical way to reduce healthcare costs and save medical resources, we expect a robust, reliable, and scalable MSN in the near future. However, current MSN are vulnerable due to unsecured infrastructure and transmission strategies. Meanwhile, the MSN may leak patients' identifications or other sensitive information that violates personal privacy. Without solving security and privacy problems, patients would not choose to use this product. Our dissertation will study the information infrastructure of the MSN and try to address a problem on temporal signals using accountability techniques. The result will be used for further secure the MSN and make the temporal signal accountable.

1.2 Research Issues

1.2.1 Issues in Smart Grid

One specific problem in the smart grid is about the bill information. From homeowners' perspective, their primary concern regarding power usage is the monthly power bill sent by their service providers (e.g., power utilities). If possible, homeowners would rather know the details of their power usage than simply a bill with a total consumption. Albeit the real-time, or day-to-day, consumption of electricity could be revealed by the smart meter, we still doubt its reliability: the utility, or the smart meter itself, may alter transmitted data to suit someone's interests or for some other reasons (e.g., due to the fact that they are under attack or malfunctions). As a consequence, a homeowner could have two different electric bills: one from the utility's meter and one from the home meter. Furthermore, in smart grids, prices change with time such that

traditional billing method using a unit price is no longer feasible. Therefore, the exact times when power is used are important and should be made accountable.

From utilities' perspective, they charge customers solely based on the readings from their power meters. In order to get individual power consumption, in the past, the utility would send technicians to manually gather meter readings. At present, by using automatic meter reading (AMR) technology, meter information can be remotely obtained via a private corporate network or the public Internet. Once the meter is compromised or malfunction (i.e., we denote it as a *faulty* meter), the reading may not reflect actual information of power consumption. The utility therefore could have economic loss. This kind of events is usually caused by unauthorized meter modification. A possible solution is to prevent the meter from being altered. For example, if there is an illegal change on the meter, it will be disabled automatically and send a relevant notification to the utility. We could use a circuit design to do this job [5]. However, the hardware approach has the capability of being bypassed by sophisticated cyber-attacks in more complex networks of smart grid. Malicious one may hack the meter via a network system without touching the meter physically. Considering the operating cost and technical difficulty, utility only measures the aggregated power supply (in a substation) to a service area. For each branch of the supply, the utility installs one meter (at the consumer's side) to monitor the power usage. Within such infrastructure, it is really difficult for the utility to find a ***faulty*** meter. They just monitor the aggregated reading and the sum of all branch readings. If there is any difference (within a tolerable range considering normal transmission loss) between them, then the monitored area could be suspicious. By this means, the utility only narrows down the suspicious group but may hardly identify the ***faulty*** one.

To solve the above problems and to make the smart grid reliable are the two objectives of this dissertation. After reviewing metering systems in smart grids, we design two accountable, communication protocols for home area network (HAN) and neighborhood area network (NAN) using a peer review strategy.

1.2.2 Issues in Medical Sensor Network

Although the new platform saves time for patients to see a doctor, problems still exist in the MSN that cannot be ignored. Medical sensors may have different capabilities, such as detecting electrocardiographs (ECG), heart rate, blood pressure, or pulse rate. All these parameters are important to timely detection and classification of abnormal physical statuses. To obtain accurate sensor readings in unreliable channels is always the goal of ongoing research. Nevertheless, it is hard to get the ideal readings because of sensors' limitation. On the one hand, a sensor's wireless communication range is limited (typically < 100 feet, due to the limited power and capacity of the tiny antenna). On the other hand, sensors have deficient usability and poor security, especially the immature patient privacy-preserving technique. Hence, many hospitals and patients are afraid of using current telemedicine systems. A tradeoff between their usability and credibility needs to be achieved [73]. According to the study in [74], we believe that a multi-hop message communication system cannot be well protected only by typical security technologies (i.e., digital signatures and cryptography). As a complement, accountability and anonymity are required to secure the MSN.

Albeit general system accountability can preserve the integrity and confidentiality for data transmissions, the MSN still has no protection against temporal signal spoofing. It is

obvious that the accuracy of an ECG trace depends on the accuracy of temporal signals within each sensor's report. Any change, no matter whether it derives from an attacker's spoofing or comes from a malfunctioned sensor, may lead to quite another result. To locate the problem, we should hold the temporal signal accountable.

For the privacy issue, since sensor's ID on patient's body corresponds to the patient's profile record in a medical database, disclosure of information source during wireless communications can cause a violation of the patient's privacy. Moreover, when such MSN platforms are widely deployed in the national medical sites (such as nursing homes, hospitals, etc.), they could become the potential attacking objects of cyber-terrorists. Considering the confidentiality of sensitive medical data, we definitely need an end-to-end security scheme to protect them. Two crucial MSN components need to be involved: 1) the sensor-to-sensor communication should be secured through low-cost symmetrical ciphers; 2) the medical data should be authenticated and encrypted through extremely light-weight security schemes. Since sensor network security has been studied extensively, we will only focus on how to overcome current privacy problems while preserving temporal accountability in this dissertation.

1.3 Organization

The rest of this dissertation is organized as follows. Chapter 2 gives brief overviews of smart grid and medical sensor network in terms of their features, architectures, and security issues. The chapter 2 also introduces the related work of accountability. Chapters 3 to 6 are system designs using accountability techniques. Specifically, chapter 3 designs an accountable metering system for smart grid in a home area network. Chapter 4 addresses the same issue in a

neighborhood area smart grid using a different accountable protocol. Chapter 5 is to secure the medical sensor network using temporal accountability scheme. Chapter 6 enhances system performance for designs in chapter 5. Chapter 7 summarizes the dissertation.

The work in Chapter 2 is partially from two journal papers [68] and [110]. A longer version of Chapter 3 has been presented at CCNC 2011 [49]. Chapter 4 is partially from a conference paper [111] and a journal paper [112]. Chapters 5 and 6 are the subject of one journal paper [113]. Besides, the work in Chapter 6 is partially from a conference paper [114].

CHAPTER 2

BACKGROUND

2.1 Smart Grid Cyber Security and Privacy

Smart grid is a promising power delivery infrastructure that is integrated with two-way communication and electricity flows. Through advanced sensing technologies and control methods, it can capture and analyze data regarding power usage, delivery, and generation in near real-time [1]. According to the analysis results, the smart grid may provide predictive information and corresponding recommendations to all stakeholders (e.g., utilities, suppliers, and consumers) regarding the optimization of their power utilization [1]. It may also offer services like intelligent appliance control for energy efficiency and better integration of distributed energy resources (DERs) to reduce carbon emissions [2]. Apparently, it is not a simple grid in the sense of our current power grid. It can be regarded as a "system of systems" that involves both information technology (IT) and electricity system operations and governance.

Such a complex system undoubtedly presents many challenges, especially in cyber security and privacy aspects [3]. Based on experiences gained from developed IT and telecommunication systems, we know that the envisioned grid will be a potential target for malicious, well-equipped, and well-motivated adversaries. Specifically, the grid can be subject to physical attacks by a human being, by malicious software that can harm the control system, or by using up the systems' resources to perform the attacker's own tasks. Any of these forms of

disruption occurring to the grid can be highly dangerous. Threats such as fiddling with billing information of particular users can cause a major economical disturbance, if they are not monitored carefully. The power grids, on the other hand, are a major resource to the national defense, and any form of attack on these can cause havoc. Furthermore, increased connectivity of the grid will enable personal information collection, which may invade consumers' privacy. Failure to eliminate these threats will hinder the modernization of the existing power industry. Although contemporary security technologies, such as virtual private networks (VPNs), intrusion detection systems (IDSs), public key infrastructure (PKI), anti-virus software, firewalls, etc., have well protected the IT infrastructure, they still cannot be very effective by directly deploying them without changes in the smart grid due to their inherent differences, as described in Table 2.1. For example, intruders may utilize VPN to hack the power grid. The North American Equipment Council (NREC) reported the effects of a slammer worm on the power utilities used over in North America [48]. In a quoted example they claim: "The worm migrated through a VPN connection to a company's corporate network until it finally reached the critical supervisory control and data acquisition (SCADA) network. It infected a server on the control-center LAN that was running MS-SQL. The worm traffic blocked SCADA traffic."

In fact, we may transplant some IT security techniques into the smart grid to meet its security and privacy requirements. However, while choosing any of the possible security measures, there always exists a tradeoff among security, cost, and performance. Employing firewall or proxy systems may reduce the risk of having a denial of service attack on the servers, but these strategies fail when there is an attack on the application layer, such as planting a Trojan. A Trojan horse referred to here is malicious software that acts as if it performs the intended functionality, but secretly passes credentials and other secure information to the attacker. Power

grids usually are equipped with their own subnets and IP segments. These measures tend to make them a little more secure when compared to general systems built off the Internet, but an attack by gaining physical access to the system can rarely be avoided. An understanding of system components and associated cyber-vulnerabilities is therefore necessary for the smart grid deployments and is the motivation of this section.

Table 2.1. Difference Between IT Networks and Smart Grid

Categories	IT Networks	Smart Gird
Security Objectives	Confidentiality > Integrity > Availability	Availability > Integrity > Confidentiality [3]
Architecture	1) flexible and dynamic topology; 2) center server requires more protection than periphery hosts [30].	1) relatively stable tree-like hierarchy topology; 2) some field devices require the same security level as the central server [30].
Technology	1) diverse operating systems; 2) public networks; 3) IP-based communication protocols.	1) proprietary operating systems; 2) private networks; 3) IEC61850- & DNP-based communication protocols.
Quality of Service	1) transmission delay and occasional failures are tolerated; 2) allow rebooting [30].	1) high restrictions on transmission delay and failures; 2) no rebooting [30].

2.1.1 Overview of Smart Grid

2.1.1.1 Features

In 2007, the U.S. National Energy Technology Laboratory (NETL) [6] identified seven principal characteristics for modern power grid design. Later in 2009, the U.S. Department of Energy (DOE) merged two of them (self-heals and resists attack) and restated the design features and benefits for smart grid as follows [2]:

1) Enabling Informed Participation by Customers: Unlike traditional power systems, customers are better informed by a two-way communication technology. The entire smart grid becomes an active electricity market that allows customers to shift load and to generate and store energy based on near real-time prices and other economic incentives. Through bidirectional electricity flow, customers are also able to sell surfeit stored energy back to the grid when the price is high. Such demand-response mechanisms help the grid balance power supply and demand, thus enhancing the efficiency of power usage.

2) Accommodating All Generation and Storage Options: The smart grid not only accommodates remote centralized power generation, but also adopts diverse and widespread distributed energy resource (DER) (e.g., solar, wind, or geothermal energy) through flexible network architecture and distributed management. This concept is proposed to alleviate peak load, to support back-up energy during emergencies, and to satisfy the grid's developing in accordance with the natural environment, society, and the economy.

3) Enabling New Products, Services, and Markets: New products and services are essential parts of the smart grid that can promote low-cost and green solutions for all power users. By using consumer-oriented "smart appliances" or intelligent electronic devices (IEDs), for instance, customers or authorized service providers can remotely control IEDs' power usage. Markets act as coordinators managing a series of independent grid parameters, such as time, capacity, the capacity rate of change, service quality, etc. When necessary, markets will adjust those variables to balance the power supply and demand of the entire grid.

4) Providing the Power Quality for the Range of Needs: Power quality involves factors like voltage flicker, voltage volume, momentary interruptions, etc. Different consumers may have distinct power quality requirements (e.g., industrial vs. residential users). To satisfy a

particular consumer's power usage, the smart grid must meet a wide range of power quality needs in terms of architectural designs and contract concerns.

5) Optimizing Asset Utilization and Operating Efficiently: The smart grid is a complex system of systems that manages a variety of appliances, facilities, and DERs. Optimizing the utilization of those assets and enabling efficient operation and maintenance will reduce both whole life-cycle and investment costs and power consumption. A reasonable and robust management method should therefore be developed.

6) Operating Resiliently to Disturbances, Attacks, and Natural Disasters: This concept is proposed to ensure the reliability of the power grid. Regardless of the type of physical damages or cyber-attacks, the smart grid can effectively resist these problematic events through local, regional, and national coordination. As a countermeasure, authorized operators can quickly isolate the suspected grid components and readjust nearby DERs to support the affected areas. The smart grid is also able to "self-heal" hidden faults by using technologies such as advanced sensing systems, timely detection, automatic control devices, etc.

2.1.1.2 Architecture

To date, the architectural framework and implementation standards of the smart grid are still under investigation by the academic [7, 8, 16], industrial [1, 17, 18, 30], and government sectors [2, 4, 6]. Although there are various designs for the grid architecture, almost every case follows the common reference model [4] proposed by the U.S. NIST.

As shown in Fig. 2.1, NIST's model consists of seven logical domains [4]. Each one of the above four (Bulk Generation, Transmission, Distribution, and Customers) can generate, store,

and deliver electricity in two-way. The bottom three (Markets, Service Providers, and Operations) mainly manage the movement of electricity and provide relevant information or services to power consumers and utilities. Three types of customers are present in this model: HAN (Home Area Network), BAN (Building Area Network), and IAN (Industrial Area Network). Within those areas, AMI (Advanced Metering Infrastructure) is deployed to monitor all incoming and outgoing electrical and communication flow.

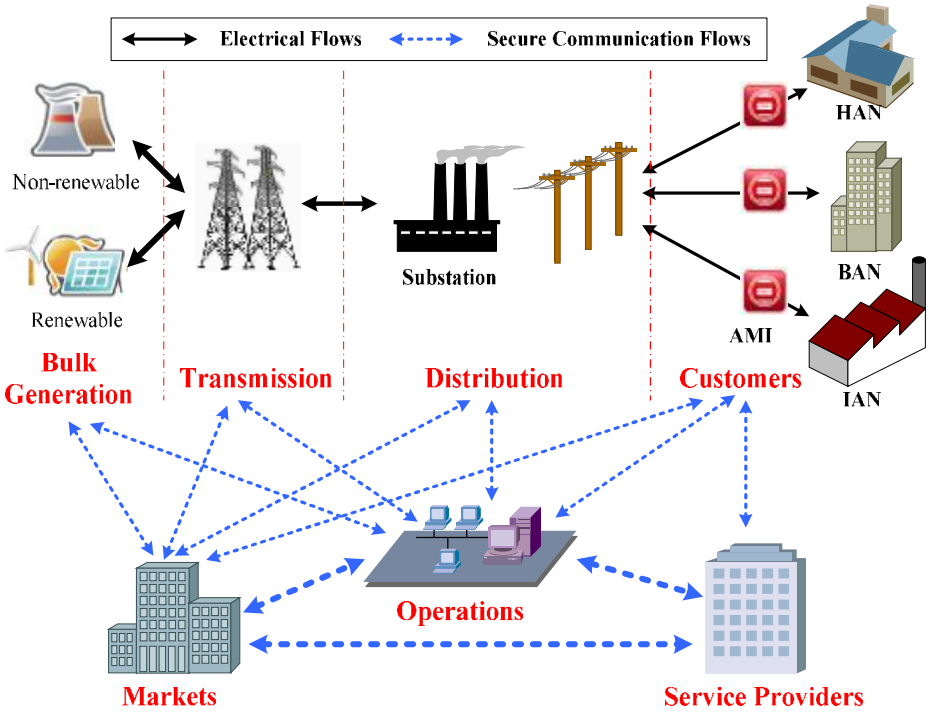


Fig. 2.1. NIST reference model for the smart grid [4, 49].

To interconnect these domains, Cisco [1] argued that the whole system should use an independent "network of networks." It also claimed that the best standard suite of protocols for the smart grid is the Internet Protocol (IP) [1]. Since IP has already achieved great success in the current Internet in terms of flexibility, security, and interoperability, Cisco believes that the

interoperability standards of the smart grid should use IP architecture as reference [1]. In addition, several researchers have proposed their own opinions on how to implement this model. Clark and Pavlovski [7] studied the pros and cons of wireless network applications for the smart grid and then suggested adopting 3G/4G technology for the architectural design. Gadze [8] presented a hierarchical architecture for the operations domain, which is a multi-level decentralized control platform dealing with the potential impacts of emergencies. Wei [16] proposed a peer-to-peer structure for the power delivery system. Basically, every consumer and power generator acts as an interconnected node in a web-like network. Such grids can dynamically balance power supply and demand, but they require more flexible and robust management. The rest of the presented architectures [9-11, 22] are in some way focused on one of four technical issues of the grid: (1) transmitting data over multiple media, (2) collecting and analyzing massive amounts of data rapidly, (3) connecting large numbers of devices and systems, and (4) ensuring reliability and security.

2.1.1.3 Key Components

1) AMI (Advanced Metering Infrastructure):

AMI is an integration of multiple technologies that provides intelligent connections between consumers and system operators [5]. As shown in Fig. 2.2, major applications include smart meters, HAN, meter data management systems (MDMS), and operational gateways [5]. It is designed to help consumers know the near-real-time price of electricity and thus to optimize their power usage accordingly [4, 5]. It also helps the grid obtain valuable information about consumers' power consumption in order to ensure the reliability of the power system [6].

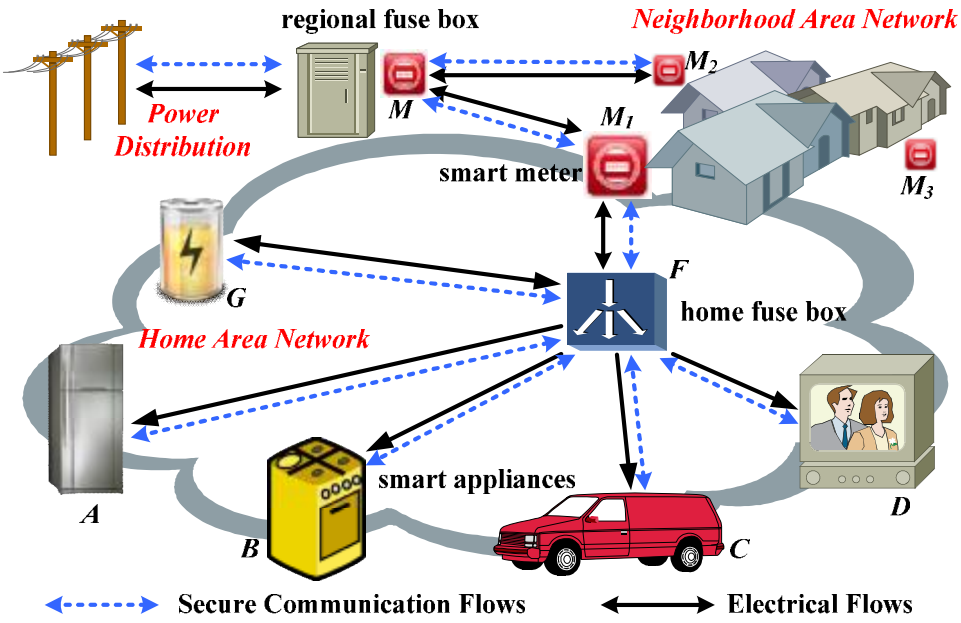


Fig. 2.2. A typical design for the AMI in smart grid.

2) SCADA (Supervisory Control and Data Acquisition):

It is responsible for the real-time monitoring and control of the power delivery network [17, 30]. Through intelligent remote control and distributed automation management at medium voltage substations, it can both help the grid reduce operation and maintenance costs and ensure the reliability of the power supply [17, 22]. Two related subsystems are the energy management system (EMS) and the distribution management system (DMS) [18, 28].

Basically, SCADA systems consist of four parts (as shown in Fig. 2.3) [37]: 1) field data interface devices such as remote terminal units (RTUs) and programmable logic controllers (PLCs), 2) a communication system (e.g., telephone, radio, cable, satellite, etc.), 3) a central master terminal unit (MTU), and 4) human machine interface (HMI) software or systems. By using RTUs and PLCs, most control actions can be performed automatically and remotely [17, 18]. The Idaho National Laboratory (INL)'s report [15] claimed that the current SCADA system

has lots of vulnerabilities (discussed in section 2.1.2.3), but that many of them are proprietary. Creery *et al.* [47] discussed a few realistic situations of attack on physical SCADA systems that caused a major stir in the industry. To secure a SCADA network, a variety of technologies are involved, including user and device authentication, firewalls, IPSec (Internet Protocol Security), VPN, intrusion detection systems (IDSs), etc. [40].

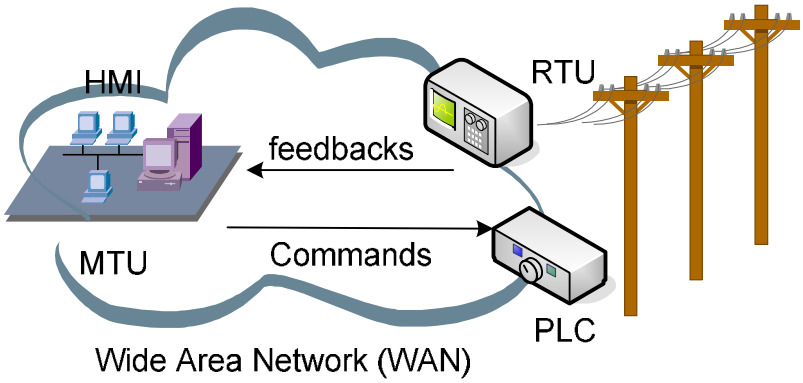


Fig. 2.3. A typical SCADA architecture [37].

3) PHEV (Plug-in Hybrid Electric Vehicle):

Many studies [3, 4, 6, 19-21] have found that PHEVs, in addition to reducing carbon emissions and reliance on fossil fuels, could also provide a means to support DER in the smart grid. Since most PHEV batteries are designed to speed up rapidly for fast discharge, parked PHEVs can supply electric power to the grid [19]. This vehicle-to-grid concept may improve the efficiency and increase the reliability of the power grid [19]. However, it is still under development and the tradeoff between costs and benefits is still uncertain [2].

4) Communication Protocols and Standards:

The communication standards for the power industry were developed by five leading organizations including the IEEE, the IEC (International Electro-technical Commission), and the

DNP3 (Distributed Network Protocol) Users Group [37]. The most prevalent protocols for SCADA communication systems are IEC 60870-5 and DNP3 [37]. The IEC protocol is typically used in Europe for communication between MTU and RTUs in SCADA systems [38, 39]. The DNP3, which is derived from IEC 60870-5 and recognized by the IEEE 1379 standard, is widely used in Asia and North America [38, 39]. IEC 61850 has now been released to support more enhanced capabilities including a peer-to-peer communication mode for field devices [39]. It can be regarded as a successor to the DNP3 [29].

IEC 62351 [41] is a standard that specifies security constraints and concerns of the above communication protocols and standards. It consists of eight parts. The first two parts present an introduction to its background and a glossary of terms. Part-3 specifies the security requirements for TCP/IP profiles in IEC 60870 and IEC 61850. In particular, it describes the TLS (Transport Level Security) configuration for secure interactions [29, 38, 41]. Part-4 addresses MMS (Manufacturing Message Specification, ISO 9506) protocol security in the IEC 61850 standard. Specifically, the MMS will work with the TLS to secure communications [38]. Not all components are required to adopt this secure mechanism [38]. Part-5 focuses on the security of serial communication in IEC 60870 and DNP3. It suggests that the TLS (Transport Layer Security) encryption mechanism can be utilized for serial communication to enable confidentiality and integrity [38]. As for the authentication, the serial version can only address replay, spoofing, modification, and some DoS attacks [38]. It cannot prevent eavesdropping, traffic analysis, or repudiation due to its limited computing capability. However, it could be protected by alternate methods, such as VPNs or “bump-in-the-wire” (a scheme that use an IPSec device as a firewall to filter unwanted packages from the Internet) technologies, depending upon the capabilities of the devices and communications involved [38]. Relevant key

management measures are also described in this part [29]. Part-6 provides security for non-routable peer-to-peer communications. Since the interval of transmitting messages over SCADA networks is limited to 4 milliseconds (according to the IEC 61850-8-1), general encryption or other security methods are not feasible [38]. Authentication therefore becomes the only option that is used for P2P security [38]. Part-7 and Part-8 are still at draft specification and require further study. The objective of Part-7 is to secure the network and system management (NSM) of the information infrastructure. Two existing technologies will be utilized: the simple network management protocol (SNMP) and the ISO common management information protocol (CMIP) [38]. Part-8 is designed to address authorization problems in control centers. One promising mechanism that is mentioned is role-based access control strategy [29].

2.1.2 Cyber Security Issues on Smart Grid

Traditional power delivery system focuses on developing equipment to improve integrity, availability, and confidentiality. Until recently, contemporary communication technologies and equipment were typically regarded as supporting the power industry's reliability. Nevertheless, increased connectivity is becoming more critical to the cyber security of the power system. In a broad sense, the cyber security of the power industry covers all IT and communications issues that affect the operation of power delivery systems and the management of the utilities [3]. Specifically, securing the power grid prevents, prepares for, protects against, mitigates, responds to, and recovers from unexpected cyber events or natural disasters [3].Wei ***et al.*** [30] pointed out that the development of a secure smart grid would encounter the following four challenges:

1) Power delivery system has new communication requirements in terms of protocols, delay, bandwidth, and cost. Avoiding early obsolescence is essential in smart grid development.
2) Many legacy devices have been used in power automation systems for decades. Most of them only focus on a certain functionality and thus lack sufficient memory space or computational capability to deal with security problems. Integrating the existing legacy equipment into the smart grid without weakening their control performance is a challenge.
3) Networking in the current power grid uses heterogeneous technologies and protocols such as ModBus [50], ModBus+ [50], ProfiBus (Process Field Bus) [51], ICCP (Inter-control Center Communication Protocol), DNP3 [37], etc. Nevertheless, most of them were designed for connectivity without cyber security.
4) Current power systems are usually proprietary systems that provide specific performances and functionalities but not security.

Many organizations are currently involved with the development of smart grid security requirements, including NERC CIP (North American Electrical Reliability Corporation – Critical Infrastructure Protection), ISA (International Society of Automation), NIPP (National Infrastructure Protection Plan), IEEE (1402), and NIST. One prominent set of requirements has been reported by the NIST Cyber Security Coordination Task Group (CSCTG) [3]. After reviewing the NIST CSCTG report [3] and existing research [13-15, 17, 21, 22, 24-36, 39-43, 48, 52-67] on cyber security, we have categorized the relevant issues into five groups (as shown in Table 2.2-2.6). Notice that general security problems such as software engineering practices, firewalls, circuit designs, and patch management will not be included in the tables.

2.1.2.1. Device Issues

Table 2.2. Cyber Security Issues on Smart Grid Device

Key Words	***Potential Problems***	***Possible Solutions***
Smart Meter	• Customer tariff varies on individuals, and thus, breaches of the metering database may lead to alternate bills [3]. • Meters may suffer physical attacks: battery change, removal, or modification [3, 30]. • Functions like remote connect/disconnect meters and outage reporting may be used by unwarranted third parties [3, 53].	• Ensure the integrity of meter data. • Secure smart meter maintenance. • Detect illegal changes on meter. • Authorize all accesses to/from AMI networks.
Customer Interface	• Home appliances can interact with service providers or other AMI devices. If manipulated by malicious intruders, they could be unsafe factors in residential areas [3, 14]. • Energy-related information can be revealed on IEDs or on the Internet. Unwarranted data may misguide users' decisions [3].	• Access control to all customer interfaces. • Validate notification. • Improve security of hardware and software upgrades [3].
PHEV	• PHEV can be charged at different locations. Inaccurate billing information or unwarranted service will disrupt market operations [3].	• Establish standards for the electric vehicle [3].

Devices like PLCs (Programmable Logical Controllers), RTUs, and IEDs are widely deployed in power systems to allow administrators to perform maintenance or to dispatch functionalities from a remote location [30]. This function also enables malicious users to manipulate the device and disrupt normal operations of the grid, such as shutting down running devices to disconnect power services or tampering with sensing data to misguide the decisions of the operators [30]. The authors in [53] discussed such a cyber-vulnerability, in which an attacker could switch-off hundreds of millions of smart meters with remote off switches. Although no agreed solutions are proposed in present standards and regulations, some recommended countermeasures in [53] may be considered in further discussions. For the devices, the IEEE 1686-2007 standard has specified security requirements. However, experience shows that typical

IEDs are far from complying with this standard. As described in Table 2.2, potential security problems may present in the applications of smart meter, customer interfaces, and PHEVs.

As for the meter device, a conventional physical meter can be modified by reversing the internal usage counter or be manipulated to control the calculation of the electric flow [14]. Addressing this problem may require hardware support. We will not focus on its solutions in this section. Besides, data aggregation is generally perceived as a main function for the smart meter. Several algorithms [60, 61] have been proposed to prevent the meter data from being compromised. Authors in [61] analyzed the tradeoff between security and efficiency and designed two algorithms for per-hop and end-to-end communication protocol respectively. They used AES-CCM with 128 bit shared key to encrypt the line between the meter and the gateway, which showed their protocol is reliable and energy efficient (based on their experiment results).

As for the customer interfaces and PHEVs, not too many papers are presented to address potential security problems. Ongoing relevant research mainly focuses on issues of malware attacks and fast encryption. Metke and Ekl [27] proposed some suggestions for malware protection on embedded systems and general purpose computer systems. For embedded systems, manufacturers should take full responsibility for securing software development and upgrade procedures. To meet this requirement, three possible approaches are discussed. First, the manufacturer may issue a public key to each device and encrypt all new software with the corresponding private key. The device can then validate the source of the updated patch and thus secure the system. The second method is called the “high assurance boot” (HAB) method. The embedded system will be validated once it boots up. The validation script is safely coded into its hardware by the manufacturer. Since not all devices can be rebooted very often, secure validation software is considered as the third solution. By using a device attestation technique, devices can

be validated while running. When it comes to general purpose computer systems, the authors in [27] argue that current antivirus software cannot prevent the system from suffering malware attacks. Although there is currently no solution, one thing is recommended: all mobile code (e.g., ActiveX, JavaScript, etc.) in the grid should be strictly controlled from suppliers to operators.

As we know, tens of millions of sensors or RTUs are deployed in the grid for distributed automation (DA). These devices have limited bandwidth, power (battery or long sleep cycles), storage, memory, and intermittent connections [3]. Because of these constraints, applications like key management should require less centralization and more persistent connectivity than current approaches; it should also retain a certain level of trust and security for the entire infrastructure [3]. NIST requirements [3] suggest that each device has unique key and credential materials such that, if one has been compromised, others will not be affected. Zhang *et al.* [31] proposed a 256-bit AES-based solution to secure the traffic between two smart grid devices in Ethernet networks. AES algorithms have inherent requirements for the smart grid: it must only require a few memory spaces and be able to be used for wireless sensor networks (WSNs). In their design, all data packets in Ethernet networks consist of four fields: one header and three data fields. Specifically, the header contains the destination IP address. All other nodes except the recipient cannot read the data payload and will simply discard it. The data payload includes 3 fields. Each of them is 16 bytes, since AES will only process 16-byte sized data. To indicate whether a message is encrypted or not, the header adds an extra AES status flag; thus this message may be transmitted through other networks. By using the Altera Cyclone-2 FPGA (Field Programmable Gate Array) based platform, they have successfully implemented their design into the hardware. Experiment results indicate that the data transmission is secure only if no eavesdroppers exist on the Ethernet network and that the throughput (bytes encrypted per second) can be 1,202 bps [31].

2.1.2.2. Networking Issues

Table 2.3. Cyber Security Issues on Smart Grid Networking

Key Words	***Potential Problems***	***Possible Solutions***
Internet	• Certain applications may be built on the Internet. Inherent problems like malicious malware and DoS attacks are threats to the grid operations [1, 3, 17, 30, 32].	• Adopt TCP/IP for smart grid networks [1, 3]. • Use VPN (IPSec), SSH, SSL/ TLS [17, 40, 67]. • Deploy intrusion detection and firewalls [17, 30].
Wireless Network	• In wireless networks, layer 2/3 can be easily attacked via traffic modification and injection. Without routing security, traffic on these layers is not reliable [3, 17].	• Protect routing protocols in layer 2/3 networks [3]. • Adopt security capabilities in 802.11i, 802.16e, and 3GPP LTE [27].
Sensor Network	• Sensor data is critical for the grid. Forging, intercepting, tampering, or misrepresenting sensor data may damage the grid [3, 30].	• Adopt AES encryption [31, 61].

Potential security problems of networking in smart grids mainly focus on issues of the Internet, wireless networks, and sensor networks (as shown in Table 2.3). Just like the Internet, multiple networking technologies can be utilized for the smart grid, including fiber optics, land mobile radio (LMR), 3G/4G (WiMax), RS-232/RS-485 serial links, WiFi, and so on [27]. Which one will be used depends on the requirements of the grid environment and is an open issue in the development of smart grid communication standards.

For wired networks, Sun *et al.* [28] claimed that Ethernet Passive Optical Networks (EPON) would be a promising solution for the smart grid broadband access networks due to the following metrics: 1) backward compatibility, 2) low-cost fiber deployment and maintenance, and 3) minimal protocol overhead. EPON also has been regarded as next-generation Gigabit-Ethernet by IEEE 802.3ah standard.

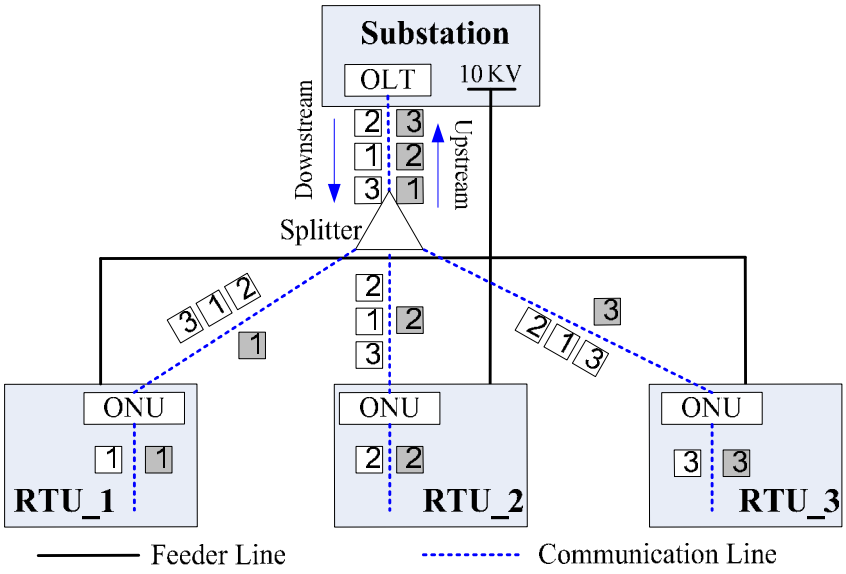


Fig. 2.4. Typical tree-based EPON system for the power grid [28].

As shown in Fig. 2.4, a tree-based EPON will broadcast messages to every ONU (Optical Network Unit), all of which share one common channel over which to deliver data to an OLT (Optical Line Termination). In this case, every ONU is able to capture all downstream traffic from the OLT and will vie with other ONUs for limited upload bandwidth. Therefore, EPON can be easily attacked by methods such as spoofing, DoS, and eavesdropping. By using identity-based cryptography (IBC) and challenge-response technology, the authors then proposed a secure communication protocol for the EPON. Unlike traditional asymmetric cryptographic approaches, the IBC generates a public key by using an arbitrary data string, and the corresponding private key binds this information, which is signed by a trusted key distribution center (KDC) [28]. In their scheme, the OLT and ONUs will periodically perform mutual authentication. First, the OLT challenges an ONU, *i*, with a message, *n*, encrypted with *i*'s public key. After verifying this message, *i* will respond with *n* and a random number *m* encrypted with OLT's public key. Upon getting this response message, the OLT can verify the identity of *i*.

Finally, the OLT will send *m* back to *i*. Thus, *i* is able to verify the authority of the OLT. This mechanism establishes a secure channel between two devices. In fact, it is also adopted by the DNP standard for secure communication [39].

For wireless networks, airborne radio waves would be potential vulnerabilities to adversaries. In particular, such an unprotected physical medium may disclose neighboring energy consumption data and thus cause a privacy invasion. A NIST report [3] claimed that schemes like 802.11i would help to secure smart grid wireless deployment. Moreover, Metke and Ekl [27] argued that wireless smart grids could be further secured with existing standards like 802.16e (Mobile WiMax) and 3GPP LTE. Possible technologies for wireless security include: mutual or server EAP (Extensible Authentication Protocol), 4-way handshake, AES-CCMP (AES-Counter Mode CBC-MAC Protocol), CBCMAC (Cipher Block Chaining Message Authentication Code), 128 group encryption key, 3DES (Triple Data Encryption Standard), PKMv2 (Privacy and Key Management version 2) RSA acknowledgement message, and mutual authentication between UE (User Equipment) and MME (Mobility Management Entity) [27].

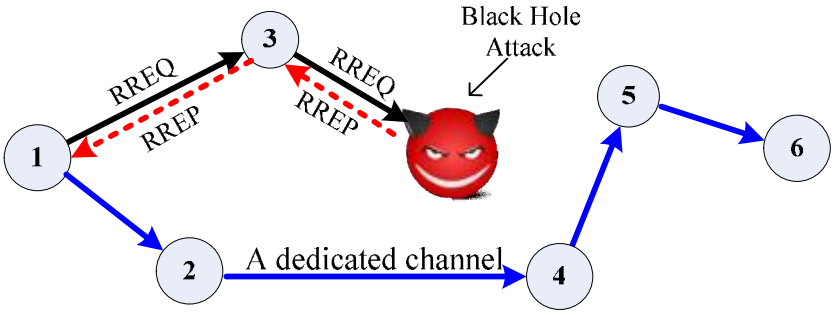


Fig. 2.5. Black hole attack against AODV routing protocol.

For sensor networks, to date, researchers [3, 27, 36] have reached the unanimous consensus that wireless mesh networks should be deployed in the AMI. A primary reason for this

is that mesh networks can overcome bad links by using redundant communication paths [21]. Nevertheless, the IT industry has witnessed a series of attacks against wireless mesh technologies such as cross-layer traffic injection, node impersonation, route injection, message modification, etc. [3]. Most existing routing protocols lack specific strategies to secure the paths and the data mainly because of their inherent distribution features [3]. Without routing security, traffic in the AMI is not reliable. Hence, Zigbee Alliance released a standard to address this problem based on Zigbee Pro and 802.15.4 standards [36]. Bennett and Wicker [36], however, argued that the conventional Zigbee protocol would suffer from severe delays due to the multi-tier feature of the cluster tree based routing strategy. Specifically, if a meter polled 11,250 bytes of data (88 packets) every 15 minutes under 802.15.4 context with a maximum data rate of 250 kbps, then the meter only had 4 milliseconds to deliver a packet, which would become even shorter when sending additional control messages or retransmission. To speed up the transfer rate, the authors suggested adding a new layer between layers 2 and 3 of Zigbee networks. This layer would use a modified multiprotocol label switching (MPLS) layer 2.5 protocol to decrease end-to-end delays. In addition, they suggested that the routing protocol in Zigbee networks should be pure AODV (Ad Hoc On Demand Distance Vector), which could significantly shorten the time required to establish a path. Through an in-depth study of the AODV protocol, the authors also found that this routing mechanism would easily suffer from "black hole" attacks [36] that discard path establishing messages. To address this problem, they proposed the solution of establishing a dedicated path between the two communication principals (as shown in Fig. 2.5). Simulation results in [36] indicated that these recommendations could improve the network throughput enough to meet the meter reading requirements.

2.1.2.3. Dispatching and Management Issues

Table 2.4. Cyber Security Issues on Dispatching and Management

Key Words	***Potential Problems***	***Possible Solutions***
SCADA / EMS / DMS	• Distribution control commands and access logs are critical. Intercepting, tampering, or forging these data damages the grid [3, 30, 39, 54-59]. • Synchronizing time-tagged data in wide areas is essential; without it the safety and reliability of the SCADA system cannot be achieved [3, 20]. • Every decision of SCADA comes from analysis of raw data with reasonable models. Improper models may mislead operator's actions [3]. Different vendors using distinct SCADA models will disrupt the consistency of the grid [3, 17]. • EMS load management provides both active and passive control. Inconsistent agreement on load control may cause unwarranted outages [3]. • DER management includes load forecasts. False forecast may misguide DMS's decisions [3].	• Ensure commands and log files are accurate and secure [3]. • Use a common time reference (GPS time-stamped) for time synchronization [32]. • Sign a contract with the utility company that allows the DER to be used for load support [3]. • Adopt multi-layer intrusion detection [30].
Asset Manage	• When assets need to be replaced, unplanned outages and equipment damages could occurs [3]. • Compatibility problems could emerge while integrating legacy devices into the grid, which may cause the system to fail or malfunction [1, 3].	• Maximize the life-cyle of assets via operators' cooperation [3]. • Back-up data mart [3]. • Enabling backwards compatibility [3].
Cipher Key Manage	• Data encryption and digital signatures are required in sensors to secure communications. Most of existing cryptographic scheme lack of efficiency under limited space and computation [3]. • Access and communication may occur across different domains [43]. To manage their own credential keys in different areas is difficult, especially in a national wide senario [3]. • Device or system may be "locked out" when an emergency occurs [3].	• Use PKI for key management [17, 27]. • Adopt IBE for Encryption [43]. • Design hierarchical, decentralized, and delegated schemes, or their hybridization [3]. • Design a bypass means for emergency while remaining secure in daily operations [3].
Real-time Operation	• Some applications (e.g., real-time process) must meet limited time constraints. Increasing interoperability may cause unbounded and uncontrollable delays of the power system [3, 30].	• Minimize and make predictable timing impacts of security protections [3].

Dispatching and management issues in the smart grid mainly present in its SCADA system (refer to Table 2.4). Smart grid can be regarded as a combination of several micro grids [11]. Each micro grid operates autonomously within its local SCADA system and interacts with others like “Island Functionality” or “Islanding.” Meanwhile, all micro grids will be controlled by a central master SCADA system in which every local SCADA acts as a slave controller providing energy related information to the central controller. This framework ensures reliability of the grid and thus has been approved by the IEEE-1547 standard. Traditionally, those SCADA systems are isolated and controlled by authorized personnel. Most of them lack real-time control and monitoring capabilities [32]. Until recently, GPS time-stamped (in milliseconds) phasor measurement units (PMUs) offered a solution to this problem. To address the clock synchronization problem in the distributed context, the NTP (Network Time Protocol) and the IEEE 1588 standard are adopted [40]. This increased interoperability, however, makes them more accessible to public users, which inevitably increases the risk of the system being compromised as follows:

1) *Take down the server*: If the IP of the SCADA server and the network path are known to the attacker, the server can be easily taken down by the DoS attacks or by simply deleting the system files. These attacks can cause a major danger to future services as well.

2) *Gaining control over the system*: This is achieved by planting a Trojan or by backdoor entry into the system registries. This is the highest scale of security threat, by which a false alarm and manipulated controls can be generated and sent to RTUs causing large scale collapses.

3) *Stealing corporate data*: These problems occur if the enterprise security level is poor and the software architecture used is not highly capable. The corporate data can be stolen from the database for the internal rivalry between the competing service providers.

4) *Fiddling with billing information*: Intruders might access the billing and other financial information from the system to get the details, which can later be misused and can cause major problems to the consumers. A powerful firewall is needed to protect the servers.

5) *Key logger software*: Attackers generally tend to use the logged key strokes of the system keyboard and gain access to the system passwords and usernames.

6) *Gain competitive advantage*: Attackers from one service provider generally tend to access the data of the others to get to know their strategies and thus orient their planning in a way that they would eventually benefit in a competitive environment.

7) *Misuse the SCADA servers*: to attack the other servers in the system and gain access information to the valuable information from the utility companies.

8) *Manipulate mathematical data points*: to off track the utility operators, who then tend to detect a false alarm and tend to shutdown or rescale the system causing unnecessary latencies.

9) *Change user logged data in a distant and remote DBMS*: this can affect the innocent users as well as the utility companies.

For example, an attacker can attack the power grid by attacking the energy management system (EMS) [58] via faking meter data and misleading EMS by the state estimator to make bad decisions. Papers [54-59] studied stealthy false-data attacks against state estimators located in control centers in power systems. The authors in [54] first studied the attack, and authors in [55-57] further extend their work. In [56], quantified security metrics are adopted to measure the difficulty of conducting a stealth attack against measurements. In [55], encryption was used to protect a state estimator from attacks. In [57], a security index was presented based on [56] and a protection scheme was proposed to further encrypt measurements to achieve maximizing their utility in terms of increased system security. The authors in [58] adopted a graph theoretic

approach to detect and localize attacks of state estimations. For those unobservable data attacks (i.e. meter access restriction), a polynomial-time complexity algorithm is presented to find a minimum size of compromised meters that is required. In [59], effects on the electricity market caused by the attack are discussed. From an economic view, the authors point out that such attack can manipulate the nodal price of Ex-Post market, which may bring financial profit to the attacker. However, no relevant countermeasure is presented.

The authors in [52] proposed a multilevel framework for a trust model for smart grids with distributive control systems, and the scheme migrates against widespread failures when control system components themselves are compromised. In order to limit the access right only to authorized personnel, Cheung *et al.* [24] have proposed a smart grid role-based access control (SRAC) strategy. In this strategy, each micro grid is divided into several sub-domains according to its functionalities and energy resources. The local SCADA system acts as a gateway to authorize access privileges to both local and foreign domain users with predefined security policies and role constraints. In the SRAC model, a user may be assigned several roles with different authorities and functions across the grids. A role could share its responsibilities with other roles. The role hierarchy can be organized and stored as a tree structure, in which the "parent" role directly inherits all the privileges of its "children" roles. A corresponding XML (Extensible Markup Language)-based security policy has been developed to simplify SRAC security management. Nevertheless, the authors have not clarified the details of the SRAC administration and authentication procedures in [24]. To achieve this goal, Hamlyn *et al.* [26] proposed the following constructive suggestions for the SRAC implementation: 1) use state-of-the-art digital credential technologies to verify all access requests; 2) check the reliability and trustworthiness of each request before issuing any certificate to the user. For the above two

suggestions, a flexible, robust, and efficient key management strategy is required. Years of research on securing IT communication systems tell us that deploying symmetric keys into a large number of devices can be expensive and unreliable [27]. Contemporary trust management technologies should be customized specifically for the smart grid communication system [27]. Metke and Ekl [27] believe that PKI technology is the best key management solution for the smart grid. In the current power grid, four relevant technical factors are discussed for PKI implementation: 1) PKI standards, 2) automated trust anchor (TA) security, 3) certificate attributes, and 4) smart grid PKI tools [27]. First of all, establishing a set of smart grid PKI standards is critical for device manufacturers and power service providers. Those standards should specify the security policies, PKI practices, and certificate formats. Second, they should ensure that each smart grid device has correct TA information [27]. One possible approach is to use a factory preload certificate. Every time they install a new device, the smart grid operator will authenticate it with a root certificate by using the manufacturer's TA transfer tool. Then the device's TA information will be securely stored in a local policy database. Third, certificate attributes should not involve the participation of security severs because of their unreachable situation. Thus, local policy attributes and local certificate statuses are required. Finally, they must build relevant smart grid PKI tools to ease the management of PKI components [27]. This process can be accomplished by modifying existing PKI operation tools.

A good example to deploy PKI technology into the smart grid is proposed by Hayden ***et al.*** [43]. By using an identity-based cryptograph (IBC) method, they addressed the confidentiality and authenticity issues in an AMI communication network. Based on their implementation results, they argued that this design did not require a complex setup procedure and was scalable in terms of small packet overhead (128 bytes). However, this mechanism

requires a central key-generating server to distribute a private key for a certain device or a user. Thus, key management is still an issue for regional and national wide deployment.

2.1.2.4. Anomaly Detection Issues

Table 2.5. Cyber Security Issues on Anomaly Detection

Key Words	***Potential Problems***	***Possible Solutions***
Temporal Information	• Unsecured time information may be used for replay attacks and revoked access, which has a significant impact on many security protocols [3, 36]. • Timestamps in event logs may be tampered by malicious people [3].	• Use phasor measurement units to ensure accurate time information [3]. • Adopt existing forensic technologies to ensure accurate temporal logs [3].
Data & Service	• RTUs may be damaged in various ways. The accuracy of transmitted data and the quality of services therefore can not be guaranteed [33].	• Utilize fraud detection algorithms and models used in credit card transaction monitoring [3].

Reliable operations of the smart grid require accurate and timely detection for anomalous and outliers events [3] (as shown in Table 2.5). Ways of detecting and coping with errors and faults in the power grid need to be reviewed and studied in a model that includes systematic malicious manipulation [3].

To meet the criteria for automated fault analysis in the smart grid, several studies were undertaken [34, 35], many of which are still on-going. These include 1) a concept for detecting, classifying, and mitigating cascading events based on local and system-wide monitoring data; 2) implementing an optimal fault location algorithm that uses data from substation IEDs, as well as data from the SCADA PI Historian and simulation data from short circuit programs; 3) developing a risk-based asset management methodology for maintenance scheduling that takes into account condition-based data captured by substation IEDs; 4) proposing an intelligent alarm

processor approach to take advantage of enhanced protective relay data in explaining cause-effect relationships between alarms; 5) a neural network based protective relaying scheme that enables simultaneous enhancements in dependability and security of transmission line protection.

Pang *et al.* [33] proposed a multi-agent based fault location algorithm for the smart grid. In their model, every smart distribution unit is regarded as an agent, and all agents construct a multi-agent system. There are three types of agents: node agent, control agent, and database agent. The node agent is typically bounded with an IED that locates at a feeder node of a smart grid. It can collect transient zero voltage and current signals and calculate the transient reactive power in selected frequency bands (SFB). This calculation result will be shared among different neighboring node agents. According to the transient reactive power of this node and the transient reactive power of the neighbor node, the node agent can judge whether the feeder section concluding this node is faulty or not. The control agent is located at the control centre of a smart grid. It can receive the fault information from node agents and send the control command to node agents and trigger the alarm device when a fault occurs. Meanwhile, it can manage fault data in database agent to print, display the fault data, and so on. The database agent is responsible for storing fault data and the control command [33]. When one feeder is fault, every agent obtains transient zero voltage and current and computes transient zero reactive power in special frequency bands. Through communication and collaboration among agents, all fault information is shared. According to comparing amplitude and direction of transient zero reactive power between neighbor nodes, the fault section is located.

In [63], the authors explored the threats model and constraints of the AMI and then analyzed the requirements for host intrusion detection design. They claimed that the best IDS choice for AMI is specification-based detection, which is defined as “identifying deviations from

a correct behavior profile predefined using logical specifications." The paper only gives a guideline for architectural design. More intensive study is required to complete this work.

2.1.2.5. Other Issues

Table 2.6. Other Cyber Security Issues on Smart Grid

Key Words	***Potential Problems***	***Possible Solutions***
Demand Response	• Tampering with information of real time pricing (RTP) may cause financial and legal problems [3, 25, 30, 59]. • Malware may infect the grid, indicating false trend of supply and demand. This causes substantial damage to the power delivery system [27, 32, 48].	• Deploy trusted computing platforms [27].
Protocols & Standards	• Existing protocols may have some inherent security flaws [30, 32].	• Set secure standards for the grid automation and communication [29].

Other cyber security issues on smart grid are shown in Table 2.6, where most researchers focus on vulnerabilities in existing power grid's protocols and standards. In this subsection, we give two examples to show how to improve and secure current protocols and standards.

Virtually all modern data communication protocols adhere to a messaging protocol that is well documented and available in the public domain. The DNP protocol is widely used by electric utilities throughout North America. The DNP protocol specification can be attained for a nominal user fee. Using these documented protocols allows an intruder to do reverse engineering of the data acquisition protocol and exploit the protocol using a "Man-in-the-middle" attack. The adverse effects could include sending misleading data to the field device or control center operator resulting in 1) financial loss if the attack leads to excess generation output; 2) physical

danger if a line is energized while linemen are in the field servicing the line; 3) equipment damage if control commands sent to the field result in overload conditions [30].

Another issue involves information communication standards. IEC 61850 is a popular standard that specifies interoperability technologies and data formats for communication in the domain of power automation [29]. Authors in [67] proposed a prototype multicast system SecureSCL (Secure Substation Configuration Language) to handle publish-subscribe relationships in IEC 61850 power substation networks. It is a cross-layer design that secures the inter-substation communications by using IPsec multicast. Besides, the authors also developed a tool to detect multicast configuration anomalies. Preliminary experiment results show that their work can meet the latency requirement of power substations.

IEC 62351 is a support standard for IEC 61850 that particularly focuses on security and technical requirements of vendors. Fries *et al.* [29] gave an overview of both documents and pointed out that IEC 62351 should be updated due to some new use cases in the smart grid. Those use cases are mainly derived from customer participation and demand response in the grid. According to the results of the IEC 62351, the authors argued that the MMS and XML should be further improved to ensure the integrity of the application layer. As depicted in Fig. 2.6, when a central command is forwarded by an intermediate substation, the current MMS version in IEC 62351 is unable to ensure its integrity in the application layer. To address this problem, the authors proposed a possible solution by adding a "Cryptotoken" to the command packet. First, it establishes a TLS connection on every hop with corresponding session keys on the transport level. Second, it establishes an end-to-end communication channel on the application level and negotiates the session key during the handshake phase. Third, it uses this session key to secure all subsequent traffic. Through these steps, integrity in the application layer is achieved.

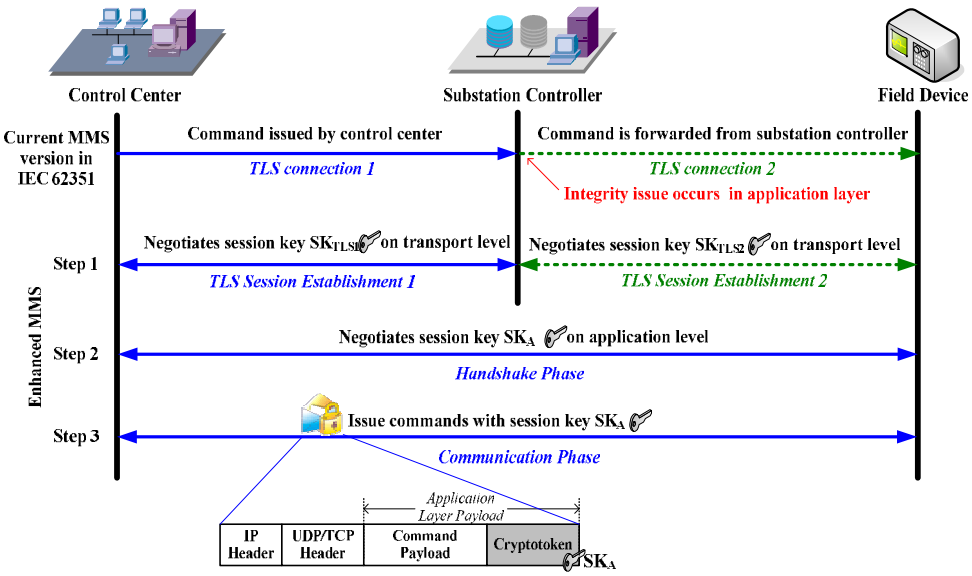


Fig. 2.6. Enhanced MMS protocol in IEC 62351 [29].

2.1.3 Privacy Issues on Smart Grid

Intelligent control and economic management of energy consumption require more interoperability between consumers and service providers. Unprotected energy-related data will cause invasions of privacy in the smart grid. In particular, radio waves in AMI may disclose information about where people were and when and what they were doing [23]. Failure to address privacy issues in the smart grid will not be accepted by regulators and customers. In this section, we will give a brief overview of current studies on privacy issues in the smart grid.

2.1.3.1. Personal Information

Personal information is any recorded information that can identify an individual directly or indirectly [3, 12]. Besides one's name, biographical, and contact information, it may also

involve personal choices, social activities, health problem, or any economic, physical, or mental information derived from the above, and information about other relatives [12]. Considering in the smart grid context, any type of energy use data that links to personal information should be secured and monitored in a proper way. NIST guidelines [3] have provided a list of personal information that may be available through the smart grid as follows:

1) *Name*: responsible for the account.
2) *Address*: location to which service is being taken.
3) *Account Number*: unique identifier for the account.
4) *Meter Reading*: kW, kWh consumption recorded at 15-60 minute intervals.
5) *Current Bill*: current amount due on the account.
6) *Billing History*: past meter readings and bills, including late/failure to pay history, if any.
7) *HAN*: in-home electrical appliances.
8) *Lifestyle*: when the home is occupied and it is unoccupied, when occupants are awake and when they are asleep, how many various appliances are used, etc.
9) *DER*: the presence of on-site generation and/or storage devices, operational status, net supply to or consumption from the grid, usage patterns.
10) *Meter IP*: the IP address for the meter, if applicable.
11) *Service Provider*: identity of the party supplying this account.

2.1.3.2. Privacy Concerns

In smart grids, energy consumption data obtained by a third party may disclose personal information without one's permission. Besides establishing corresponding laws and regulations

to protect personal information in the smart grid, we also require a secure mechanism to prevent privacy violation from breaching local data and remote copies. According to the study of NIST [3], four typical areas of privacy concern in the smart grid are presented as follows:

First of all, fraud should be considered, especially when energy consumption is attributed to a different location (e.g., in PHEVs' case) [3]. The metering system (either physical recording or electronically and remotely metering systems) should not allow any personnel abuse or modify the collected data [3]. In particular, NIST's report [3] has analyzed two relevant privacy use cases in detail. One case is about a landlord with tenants who have PHEVs that require being charged separately. For the purpose of preserving the privacy of the tenants, utility is involved to authenticate communications between the smart meter and PHEVs through a secure line and energy services communication interface (ESCI) provided by the utility and/or vehicle manufacturer. Another case is regarding PHEV general registration and enrollment process. In order to complete initial setup for PHEVs, NIST believes that utilities should offer the following services to customers: 1) enrollment, 2) registration, 3) initial connection, 4) ability to repeatedly re-establish connection between a utility and PHEV, 5) ability to provide a PHEV tariff or charging status information to customer interfaces, and 6) correct bill.

Second, data in the smart meter and HAN could reveal certain activities of home smart appliances [3]. In addition, it is can be used for tracking specific times and locations of energy consumption in specific areas of the home, which may further indicate the appliances used and/or types of activities. For example, appliance vendors may want this kind of data to know both how and why individuals used their products in certain ways. Such information therefore could impact appliance warranties. Meanwhile, other entities may need this data to conduct target marketing. Georgios *et al.* [44] designed a system that utilized a power router and a

rechargeable battery to hide or obscure load signatures in a home area. In this system, they assume that the home will have several energy storage and generation devices in the future. Through a power router, appliance load signature or usage pattern will be moderated and thus cannot be recognized and tracked by a malicious intruder. They have further improved this model in [65] and named it as ElecPrivacy. Besides, a number of privacy measurement approaches are provided for this model in [65].

Third, obtaining near-real-time data regarding energy consumption may infer whether a residence or facility is occupied, where people are in the structure, what they are doing, and so on [3]. Authors in [60] proposed a data aggregation approach for all level meters based on spanning tree topology. By using homomorphic encryption method, data is secured all the way from home meters to the data center. It can well protect the privacy of the individual power usage according to their analysis and evaluation in [66]. In [62], researchers pointed out that customers would possibly deploy a separate measurement device at home to better monitor their power usage. The redundant meter data, if transmitted in an unsecured wireless line, could leak customer's information to an eavesdropper. By compressing the data to a rate below its entropy, the authors in [62] proposed a coding method that well addressed this problem.

Fourth, personal lifestyle information derived from energy use data could be valuable to some vendors or parties [3]. For instance, vendors may use this information for targeted marketing, which could not be welcomed by those targets. The beneficial information may be revealed by new technologies like smart meters, time of use and demand rates, and direct load control of equipment. They could be further sold and used for energy management analysis and peer comparisons. Costas ***et al.*** [45] proposed an escrow-based anonymization scheme to prevent personal information from being tracked by unauthorized third parties. They categorized

metering data into two parts: “high-frequency” and “low-frequency.” Then corresponding setup and communication procedures were designed for each type of data. Those procedures are both regular PKI authentication approaches. Since the anonymity degree of the system depends on the size of the “anonymity set,” to widely deploy such a scheme requires a further investigation.

In addition, two aspects of personal data need to be considered in the review of existing laws and regulatory policies to ensure [3]: 1) granular and available data is on use of individual appliances by time and location; 2) public awareness of contractual agreements about data ownership and what may be revealed about people’s daily activities.

2.1.3.3. Recommendations

NIST has delivered a report [3] on the consumer-to-utility privacy impact assessment (PIA) of the smart grid. Ten potential design principles are proposed to address privacy issues:

1) An organization should ensure that information security and privacy policies and practices exist and are documented and followed. Audit functions should be present to monitor all data accesses and modifications.
2) Before collecting and sharing personal information and energy use data, a clearly-specified notice should be announced.
3) Available choices should be presented to all users. Organizations need to obtain users’ consent or implied consent if it is not feasible, with respect to the collection, use, and disclosure of their personal information.
4) Only personal information that is required to fulfill the stated purpose should be collected. Treatment of the information should conform to these privacy principles.

5) Information should only be used or disclosed for the purpose for which it was collected and should only be divulged to those parties authorized to receive it. Personal information should be aggregated or anonymized wherever possible to limit the potential for computer matching of records. Personal information should only be kept as long as is necessary to fulfill the purposes for which it was collected.
6) Individuals are allowed to check their corresponding personal information and to request the correction of perceived inaccuracies. Personal information data subjects should be notified about parties with whom personal information has been shared.
7) Personal information should be used only for the purposes for which it was collected. Personal information should not be disclosed to any other parties except for those identified in the notice, or with the explicit consent of the service recipient.
8) Personal information in all forms should be protected from unauthorized modification, copying, disclosure, access, use, loss, or theft.
9) Organizations should ensure the data usage information is complete, accurate, and relevant for the purposes identified in the notice.
10) Privacy policies should be made available to service recipients. They can challenge an organization's compliance with their state privacy regulations and organizational privacy policies as well as their actual privacy practices.

Cavoukian ***et al.*** [12] presented the conceptual model "SmartPrivacy" to prevent potential invasions of privacy while ensuring full functionality of the smart grid. In the case of utilities providing personal information to a third party with the express consent of an individual, the following are examples of SmartPrivacy defaults that offer greater protection of privacy:

- The information provided to third parties should be minimized such that it only fulfills the purpose of relevant services.
- When data is transmitted, the risk of interception arises. Appropriate and secure channels of transmission between different communication protocols are required to ensure strong privacy protection in the smart grid.
- Anonymize identity if possible. When sharing data with a third party, consider using a pseudonym that an individual would be permitted to reset at any time.
- Third parties should not request information from the utility about consumers, or consumers must be able to maintain control over the type of information that is disclosed to third parties by the utility.
- Third parties should agree not to correlate data with data obtained from other sources or the individual, without the consent of the individual.

2.2 Medical Sensor Network

Medical sensor network (MSN) is a wireless telemedicine platform where the patient's physical status is delivered and monitored. The telemedicine can be defined as an information technology that enables doctors to perform medical consultations and diagnoses away from patients [68, 79]. That is, doctors can remotely examine patients by viewing and asking symptoms via monitors and sound devices and gather physiological data through telecommunication. Conventional telemedicine systems are designed to be used at particular facilities and can rarely be moved to other places. Since recent wireless sensor network (WSN)

technology has allowed people to carry medical kits to patients' homes and set them up there, current systems, such as MSN, are inevitably more compact and simpler than conventional ones.

In a typical MSN, the patient is equipped with multiple medical sensors and wearable devices. These appliances are used for recording the patient's physical status and delivering that information to the monitor center (or central workstation). At the monitor center, all data (e.g., medication intake, medical records) will eventually reveal patient's real-time situation by using professional software [69]. Once an abnormal signal has been detected, doctors or nurses may take further actions on that particular patient (e.g., remind him/her to take pills immediately via telephone). In practice, ECG (Electrocardiograph) signals play a very important role for the MSN system. Each piece of ECG data may carry significant medical information [69]. Data error or loss is not tolerated. Fortunately, scholars and researchers have already explored a way to deliver a continuous and stable ECG signals in a radio-based wireless network [69].

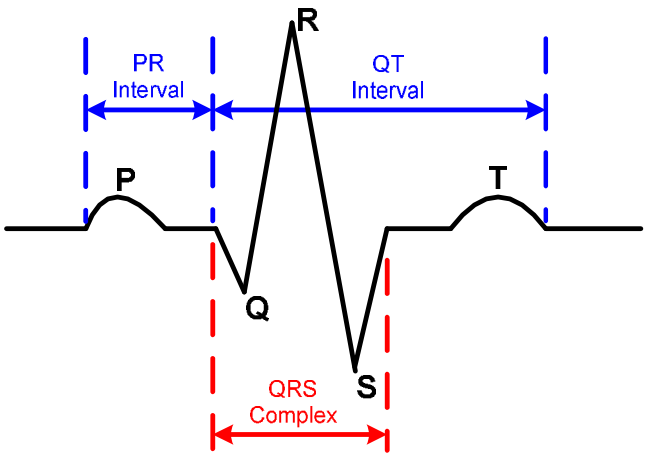


Fig. 2.7. A typical ECG trace [71].

An ECG is used for detecting abnormal heart rhythms, excessive tensing of the heart muscle, and blood and oxygen supplies [68]. Since the heart muscle's movement is initiated by

electricity as an electrically mechanical pump, this electrical activity can be captured by surface sensors (electrodes) connected to an ECG recorder [70]. A commonly used ECG recorder comes with 12-15 leads with electrodes [68]. Those leads are fixed on the patient's body (e.g., put on chest, arms, and right leg) and collect both the cardio rhythms and the heart's electrical impulses over a short period of time [69]. After that, software running on an ECG recorder will amplify these transferred electrical signals and visualize them on the display of the system or on a rolled paper [69]. Fig. 2.7 shows an example of a typical ECG trace, which has three major parts: a P wave, a QRS complex, and a T wave [71]. The P wave corresponds to an electrical signature which causes a trial contraction; the QRS complex represents the current that causes contraction of the ventricles; and the T wave reflects the ventricles' depolarization [71]. The presence or absence of these waves, including the QT interval and the PR interval shown in the figure, are meaningful parameters in screening and diagnosis of cardiovascular diseases [71].

There are several options of the ECG system for MSN to choose. They have different lead placements which range from 3-lead to 12-lead [71]. The 3-lead system is non-diagnostic and is meant for rhythm interpretation, while the 12-lead system is diagnostic [71]. Although the 12-lead system provides a more thorough coverage of ECG functionalities, it is also more costly, both financially and in terms of transport time [71]. Hence, a 3-lead system is the preferred choice for our design.

2.2.1 MSN Architecture

In the context of wireless telemedicine, two portions of the wired connections of the conventional telemedicine systems will be replaced by the wireless network connectivity: One is

a connection between the local base station which will aggregate patients' vital data from several medical peripherals locally, and existing phone lines; the other is a connection between the local base station and medical peripherals that establishes the medical sensor network (MSN) [68].

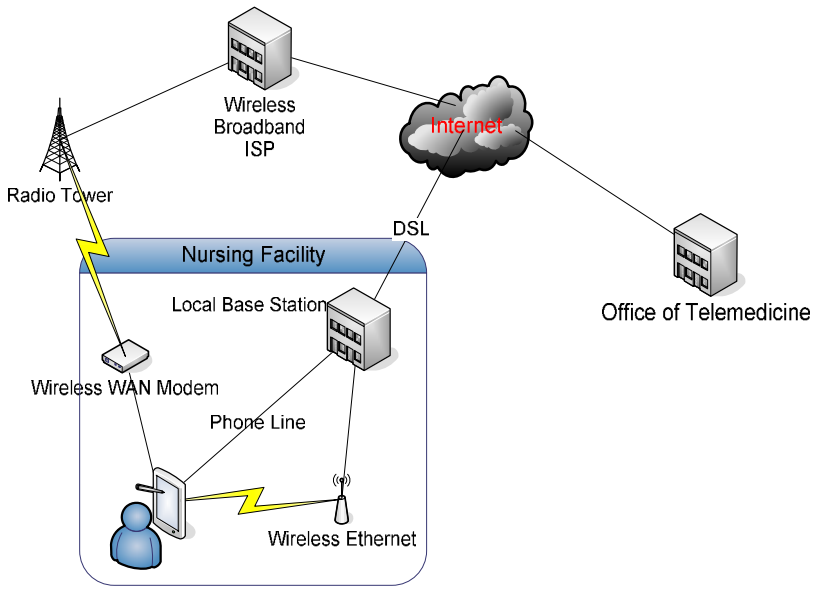


Fig. 2.8. Wireless telemedicine system architecture [68].

By utilizing these two wireless connections, it becomes possible to exploit two types of wireless configurations. The first is to employ the wireless connectivity in only the former part of the connection, i.e., between the local base station of the telemedicine kit and existing phone lines, as shown in Fig. 2.8. In this configuration, the portability of telemedicine kits will be enhanced to allow medical practitioners to carry them to patients' homes and make medical consultations away from the telemedicine facilities. Since this wireless communication will be made up by the wireless local area networks (WLANs), e.g., the IEEE 802.11 standard, the wireless access points are required to be within the range of the radio transmission. Alternatively, this wireless connection is replaced by a wireless wide area network (WWAN). In this case,

extra wireless network units, e.g. a modem or a particular PC card, are required to establish the WWAN connectivity. There also need to be companies to offer the WWAN connectivity.

The other configuration is to allow the wireless telemedicine kits to involve both of the aforementioned wireless connections, that is, between the local base station and medical peripherals as well as a wireless connection between the local base station and existing phone lines, as shown in Fig. 2.9. In this configuration, since patients are comparatively free from the wires connected to the local main station when within range of the radio transmission, they can easily move around without disturbing the data sampling. The MSN connectivity will be established by the IEEE 802.15.1 or IEEE 802.15.4 [68]. The wireless transmission ranges will be shorter than those of the WLAN.

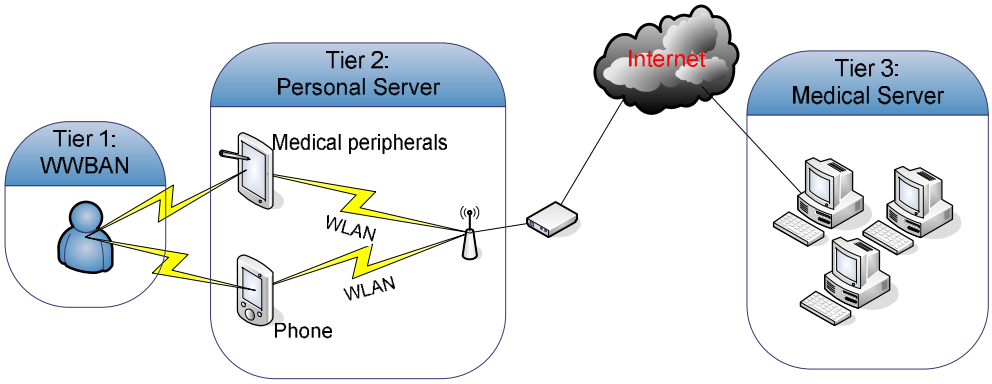


Fig. 2.9. Medical sensor network architecture [68, 77].

The MSN architecture is basically broken down into three tiers, as shown in Fig. 2.9 [68, 77]. It mainly consists of the sensing tier, the Graphical User Interface (GUI) and Data Processing (DP) tier, and the database (DB) tier [78].

1) Tier One: Wearable Physiological Sensors

The first tier is mainly responsible for sampling the patient's vital signals. This tier includes wearable medical sensors, such as a wearable pulse oximeter, ECG sensor, and blood pressure sensor [77]. These vital sensors that are integrated with a mote continuously monitor the patient's physiological signals and transmit them in real time to the local client devices, such as PDA's, which are integrated with the receiving mote. Basically, the local client devices are locally used by the medical practitioners. In addition, these wearable sensors have onboard memory so that one practitioner will temporally store the patient care records (PCR) and another practitioner will load these records when taking over the patient [76].

2) Tier Two: Personal Server Running on Intermediate Terminals

The second tier mainly interfaces with several medical sensors in tier one and the medical personnel. It is also responsible for data communications with the medical server or base station and the database at the hospital. A goal of tier two is to achieve the aggregation of data from the medical sensors, human manual input, flexible user interface configuration, and rule-based user input [78]. Tier two usually deploys a variety of local client devices, such as PDA's and tablet PC's, which run the PS that is designed to achieve the aforementioned prospects and has a user-friendly interface that will transact physiological signals and communicate with the medical server (the base station). Usually the data transmission between physiological sensors and the client devices relies on the short range wireless local network connectivity, such as IEEE 802.15.1 and IEEE 802.15.4, and this connectivity establishes the MSN around patients [68].

The PS (Personal Sever) usually provides the audio and user-friendly GUI that helps manual inputs of clinical data and gives early alerts of patient degradation [68]. In the context of the user-friendly GUI, some applications, such as iRevive, employ the meta-data driven approach that will allow users to set up the GUI layout in running time and the rule-based

approach to show procedures of data collection in particular medical cases and the relationships among sampling data [78]. These topics will be explored in later sections.

3) Tier Three: Medical Server (Database)

Tier three is responsible for aggregating and managing patient care records (PCR) and assigning network channels to the local client devices [77]. For example, in the iRevive architecture, this tier preserves three kinds of data: the PCRs, the meta-data, and the predefined medical rules [78]. The PCR including the patient's physiological data will be stored in the local database. It will also be available to various authenticated people, such as emergency department personnel, incident commanders, and medical specialists, by employing secure web portal technology [76]. The meta-data includes the entry modules for the GUI of the personal server, and the rules are the meta-data defining the procedure of the medical data sampling. Such clinical results can be used for further research in the field, and this medical history can be applied to current clinical operations as well [68]. Moreover, in the iRevive architecture, when updating the PCR to or downloading them from organizational data repositories, the data transfers are subject to the Health Level 7 version 3 (HL7v3) data exchange standard [78].

2.2.2 Wearable Medical Sensors in MSN

Recent progress of sensor technology enables miniature, light weight, low-power, low-cost wireless transceivers, or motes, to be commercially available. The motes have the basic functionality of sensing, processing, transmitting, and receiving data, and typically are used for the vehicle tracking or habitat monitoring of ducks [75]. These miniature transceivers are now ready to be used in medical applications by being integrated into several medical devices.

Wearable medical sensors are built in a wearable patch, bandage, or pair of shoes that integrates a mote and a variety of medical sensors collecting patients' physiological data in real time [68, 79]. They can establish the MSN around patients. The goal of the wearable medical sensors is to accomplish non-obtrusiveness of the medical sensors in patients. So, even though patients wear a couple of medical sensors for an extended time period, they will have little feeling that they are wearing such medical devices. Also, this non-obtrusiveness will gain freedom of mobility of patients during the medical data collections. Fig. 2.10 shows an example of medical treatments in an emergency case [76]. It is designed to support patient monitoring, patient record generation, and remote patient record review for emergency cases and mass casualty disasters.

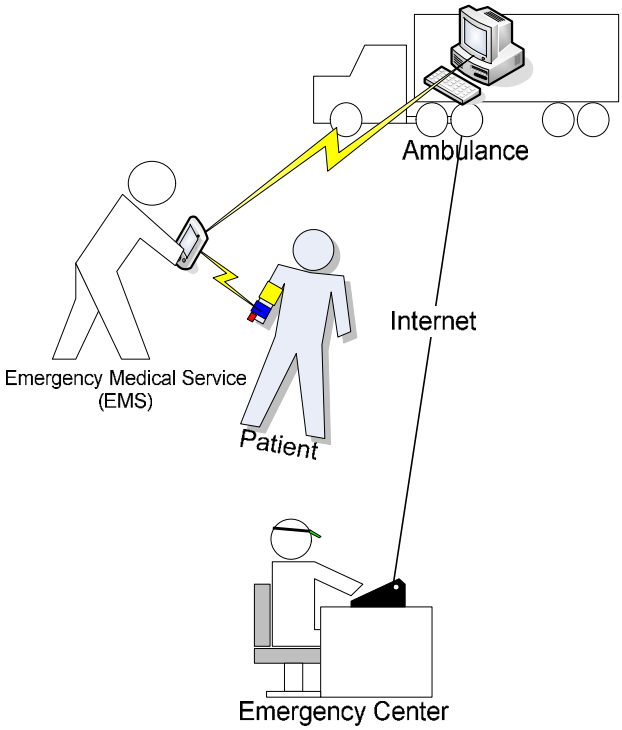


Fig. 2.10. Patient information flow [76].

During emergency cases, EMS (Emergency Medical Service) crews will distribute medical care kits with several wearable medical sensors and a mote with a wrist strap to each patient. A mote will be wrapped about the patient's wrist, and medical sensors will be placed on appropriate parts of the patients. Medical sensors will record the patient's physiological data in real time and automatically transmit them to the local base station, e.g., a laptop PC or PDA, which runs the PS (Personal Server) occupied by the EMS crew [76].

2.2.3 MSN Deployment Scenarios

The MSN architecture can be configured in three ways in terms of the arrangement of wireless network devices in tier two [77]. Fig. 2.11 shows one of three scenarios in which the wearable medical sensors and the PS running on the local client devices can communicate directly through the short-range wireless network connectivity, such as the IEEE 802.15.1 or 802.15.3/4, and the PS is connected to the home server by utilizing the WLAN connectivity, such as the IEEE 802.11 standard. The home server can usually establish communication with the Internet and send physiological data to the base station including some repositories at hospitals. Since the data transmission would occur on a regular basis, this type of wireless telemedicine architecture is suited for homes, the work place, or the hospital healthcare.

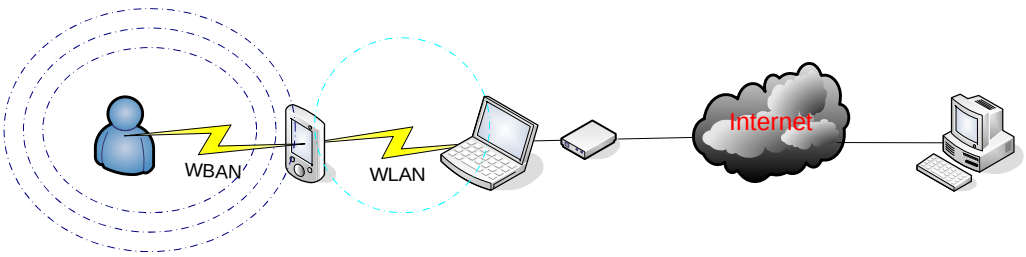


Fig. 2.11. MSN scenario 1 [68].

Fig. 2.12 adds slight modifications to the first scenario. In this type of architecture, the sensor network coordinator is directly integrated into the home server that runs the PS [77]. The patient's physiological data captured by wearable medical sensors are directly transferred to the home server through the short-range wireless network connectivity. Then this data is sent to the base station at hospitals through the Internet. This configuration is thus suited for home healthcare. However, although effectively cutting the cost of the client devices, this model can suffer from a higher energy consumption caused by requiring more RF output power and frequent retransmissions because of the low QoS.

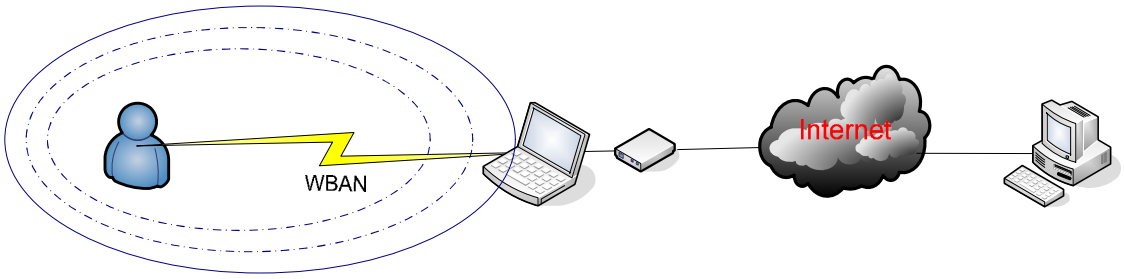


Fig. 2.12. MSN scenario 2 [68].

Fig. 2.13 shows another possible modification of the wireless telemedicine architecture, in which the portable devices running the PS, such as PDA's or 3G cell phones, which can equip a Wireless Wide Area Network (WWAN) interface, can establish communication with the medical server directly [77]. This configuration can allow the patient's condition to be continuously monitored locally, while patients can continue their lives as usual. Thus, doctors will be alerted immediately only when patients fall into critical conditions. This model is therefore suited for in home rehabilitation.

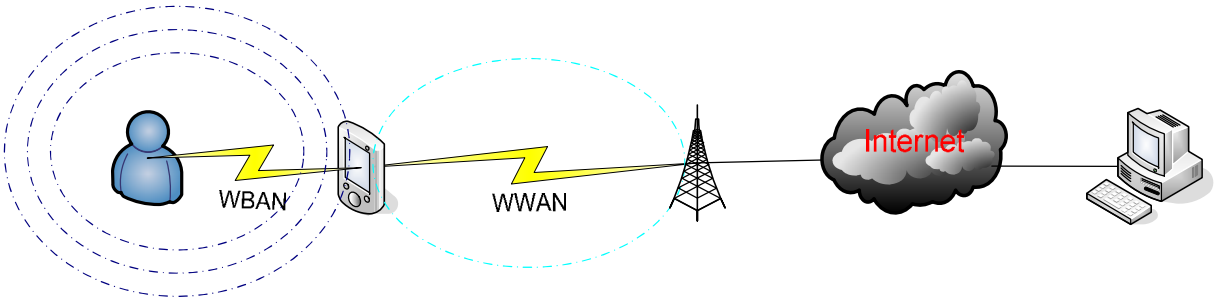


Fig. 2.13. MSN Scenario 3 [68]

2.3 Accountability

Accountability has been defined in several ways. Bhattacharya and Paul in [74] claim that, "*accountability broadly implies that transacting parties in a secure system should be made liable to what they (each, individually) did do, as well as did not do.*" Ferreira *et al*. in [73] state that the main objective of accountability system is "*to provide a means to verify, analyze, and investigate users' actions*," and "*to ensure procedures are correctly followed*." Generally speaking, in a computer system, accountability may be referred to holding a user accountable for all his/her actions on this computer. Actions could be installing new software or accessing local database. In a computer network, accountability means that the system is recordable and traceable, thus making it liable to those communication principles for its actions. Every change in a local host or network traffic, which may be the most important or most desirable information, can be used as evidence in future judgment. Under such a circumstance, no one can deny their actions, not even the administrators or other users with high privileges. Together with some suitable punishments or laws in the real world, this will prevent a number of attacks. In this section, a brief overview of existing accountability technologies is presented.

2.3.1 Theory and Model

Since the very beginning of researching accountability, people have proposed various systems and models without a formal definition. Hence, it becomes a problem to quantify an accountability degree in terms of performance or security level. To date, only a few works [84-89] are presented to address this issue.

In [80], Bella and Paulson adopt an inductive approach [81] to evaluate system accountability based on a formalized model. The key idea is to verify the evidence and fairness of the system. The result can be utilized for evaluating the accuracy of its accountable protocol. Two case studies are given: one is non-repudiation protocol [82] and another is certified email protocol [83]. The authors claim that both cases work well on their model. Just the certified email protocol provides weaker guarantees.

Jagadessan *et al*. [84] present an accountability model based on three major components: discrete timed process algebra, I/O automata, and communicating sequential processes. It is able to question a communication principal who does not follow a pre-defined protocol specification. However, if the principal is dishonest, the model cannot provide a convincing evidence to against it. The same problem applies when the principal uses cryptography to encrypt its messages.

My colleagues, Xiao *et al*. [85, 86], propose P-Accountability model to adapt the requirements of modeling a complex network and assess the degree of accountability in a fine-grained manner. It mainly provides a qualitative framework to define the degree of system accountability from perfect accountability (i.e., $p = 1$) to no-accountability (i.e., $p = 0$). At the same time, Fu *et al*. [87] propose Q-Accountability, an accountable network logging model using

users' accepted overhead (called Q-Accountable Logging by Overhead). The idea is to build the degree of accountability based on the acceptable overhead of the users.

Küsters *et al*. [88] propose a general definition of accountability in both symbolic and computational interpretations. The authors define the accountability on two major aspects: 1) fairness – no correct principals will be suspected; 2) completeness – the principals who misbehaved, or at least some of them, will be suspected. Both symbolic and computational models are interpreted based on the above aspects. This definition can be used as an analysis tool to evaluate a given accountability protocol. Three case studies are analyzed: contract-signing, voting, and auction.

Feigenbaum *et al*. [89] provide a formal accountability model based on event traces and utility functions. This is the first accountability model that considers anonymity issue. The authors argue that in certain scenarios, the correct principal will be in anonymous while misbehaviour one's identity could be exposed. To some extent, this model is more general and potentially more widely applicable.

2.3.2 Internet and Network Accountability

Accountability in the Internet is all about the methods to identify, segregate, and penalize "bad behavior" [90]. Inherent property in the initial plan of the Internet is open belief, which all users and protocols are always relied upon, by means of its interconnected requirement of accountability. Such a representation of open belief allows malicious one to utilize exposures in the set of communication devices to commence a number of cyber attacks whilst benefitting from the relief of not being followed [90].

Specifically, in a IP packet, the source IP address can be easily forged, in spite of firewalls [91], and surplus packets are capable of encroaching a host with simplicity, which is frequently fooled into unpremeditated “collaborator” (e.g., botnets), intensifying and dispersing assaults to a lot of other susceptible hosts [90]. To address this problem, many companies and organizations decide to protect their private networks by using technologies like Virtual Private Networks (VPNs), or firewalls to filter several kinds of packets. Apparently, such countermeasures are quite inadequate in their range or efficiency since intelligent worms or viruses know how to bypass firewalls [91]. Additionally, they are quite unyielding, and every now and then, they crack accessible applications and possibly obstruct the formation and operation of services and applications.

There are a lot of deliberations in the networking organizations concerning how to protect and strengthen the existing Internet while preserving its open structural design and current standards. The Internet does not provide data on the fate of information that is being broadcasted [90]. As a result, when packets lose their way or are deferred, there is no clear method for the exaggerated parties to confine the setback plus repair it if it is confined, inquire for recompense if a service-level accord has been dishonored or still study from it. Inquisitive tools similar to traceroute would help to confine the network breakdown. On the other hand, they portray their terminations derived from the destiny of investigation and not the genuine interchange, which makes them susceptible to exploitation by transfer networks. Additionally, such gears frequently disclose the interior arrangement and direction-finding strategies of ISPs, and this gives the malicious one an opportunity to deliver their networks obscure to snooping [90]. Current researchers have directed their attention to creating a number of forms of accountability to the network service. One suggestion is that an “accountability provider” officially marks every

packet with a signature [90]. The signature knows how to be confirmed by ISPs throughout the length of the trail from the dispatcher to the recipient. Users might not be content with the idea that evidence of their communication is live even in the IP level, and yet, if it will not be exposed apart from the case of wrongdoing. It is known that this information is considered to be extremely susceptible by ISPs [90].

The main reason intruders are able to login is due to the minor flaws that are left undetected in each hosts. It is highly difficult to build a system with high security without any flaws. Even if such a system is built, it will be misused by intruders who are already in the system. The distributed recognition and accountability [92] algorithm is designed for this purpose. It provides a strong accountability for all individual in the network and to find the users who try to login with different user IDs in order to hides their identities. It also groups all the activities done by a single user based upon the network identifier. This algorithm has some basic assumptions, such as: it considers no loss of packets, synchronization with the network, etc. [92]

2.3.3 Accountability in Distributed Systems

Accountability has been a major concern in distributed systems recently. Once a failure occurs in a distributed system, it is important to know what is happened and who should be responsible for this event [82]. Currently, distributed systems face a variety of threats, which requires an innovative countermeasure to make the system dependable. Accountability is such a promising technology that could make it possible to reduce the threats and prevent new vulnerabilities in distributed systems. At the mean time, it also offers a dependable system by detecting and isolating threats [82].

2.3.3.1 Byzantine Faults

Distributed system could face a variety of threats. A classical one is called Byzantine Faults [93]. It can make a protocol or algorithm to arbitrarily deviate from its correct execution. In a typical case, a Byzantine fault occurs when a faulty communication principal damages its logical condition and sends random messages in order to destabilize the system [93]. Avoiding such kind of faults would have extra cost that could make the process unaffordable. A cheaper and comparatively efficient way is to detect Byzantine faults and to remove them from the system. In this case, each communication principal will be equipped with a detector. Once the detector finds the principal is abnormal, it will alert corresponding auditor application for further suitable measures. This has been a conventional fault detection method so far [93].

By introducing the accountability into the fault detection process, a detector should not only detect the fault, but it should also know which communication principal has performed abnormal actions. In a case that a faulty principal is found, this detector should also let the whole system know about such information of that particular principal. This kind of accountable distributed system need to be complete and accurate. The complete means that whenever a faulty principal is found by a correct one, the system should generate legitimate evidence against it. The accurate means that there is no chance for the system to present any evidence against a correct principal. Opting faulty principals in a distributed system instead of masking is mainly because [93]: 1) the detection process requires less replication of principals; 2) the entire process is cheaper; 3) detecting faults facilitates a timely reaction to faults; 4) the presence of detectors reduces the likelihood of certain faults; 5) the system is recoverable due to backup snapshots; 6) detection plays a vital role in a system with multiple administrators.

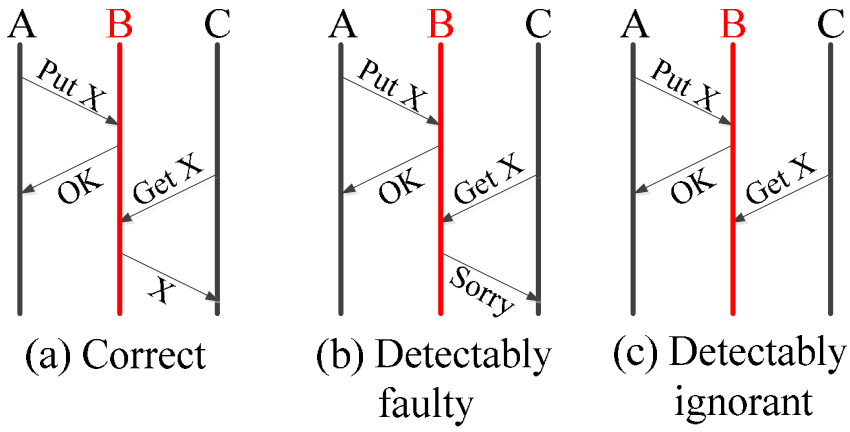


Fig. 2.14. A simple exchange of message [93]

Communication principals in a distributed system can be principally categorized as three types: a) correct, b) detectably faulty, and c) detectably ignorant. Fig. 2.14 explains these categories assuming principals A and C are always correct. The detectably defective and ignorant principals are those needs to be dealt with. If a detector on a correct principal finds an abnormal behavior on the other one, then it will propel a failure indication to all its corresponding local application processes. The indication falls into three forms: *trusted*, *suspected,* and *exposed* [82]. If an *exposed* indication appears, that means there is proof that a particular principal is faulty. If a *suspected* indication arises, it means we have an evidence to support the fact that a principal tries to ignore the requests. If none of the above two hold, it means all principals are *trusted*. In order to check all evidences of a detector, a PeerReview failure detector can be utilized [82].

2.3.3.2 PeerReview

PeerReview [94] system is used for offering accountability in a distributed network. It generates secure log files in each communication principal that records all information traffic

sent and received by a distributed application process. The key idea of the PeerReview is to sustain a protected evidence (e.g., log file) of the behavior by every principal. Such evidence is used for supporting the principal performance diverges from that of a known suggested execution and therefore reveals defective nodes [94]. There are two phases involved to formulate an accountable distributed system with PeerReview. Firstly, it records every communication principal's actions in a secure log file. The second step is to examine the recorded logs and to detect abnormal behaviors. In order to perform these actions, a perfect fault detector should possess ideal completeness and ideal accuracy.

1) Assumptions and Protocols

In order to provide accountability in a distributed system, PeerReview has to make certain reasonable assumptions and accordingly designs several protocols as its components. There are three most important assumptions in the PeerReview system [94]: 1) a message propelled from a single correct principal to another is in the end acknowledged if retransmitted often; 2) the hash function is used by the principals that are image and collision resistant; and 3) every principal uses public key cryptography to do user and/or message authentication. The protocols used by PeerReview are as follows [94]:

- *Commitment Protocol*: It guarantees that the sender of every message acquires provable confirmation that the receiver of the message has logged the transmission.
- *Consistency Protocol*: It guarantees that each principal in a distributed system keeps a secure and linear log with the aim of being reliable with every other authenticator the principal has challenged, or it is uncovered by another single proper observer.

- *Audit Protocol*: It guarantees that for each principal in a distributed system, the principal's actions are dependable with the reference execution of its own state machine, or else, the principal is uncovered by at least one correct witness.
- *Challenge/Response Protocol*: If a principal is failed to acknowledge a message, this protocol guarantees that the particular principal is alleged by at least one correct witness.
- *Evidence Transfer Protocol*: It guarantees that all correct principals in a distributed system give a failure indication for every faulty principal.

2) PeerReview Design

PeerReview is an extended version of FullReview [94]. The main idea of FullReview design is that there is a dependable unit that can consistently and immediately converse with every other communication principal in the system. The membership of the system is static and every principal has the knowledge of the requirement of the complete system [94].

The FullReview performs as follows. Every message is sent via a reliable component, which in turn confirms that all correct principals monitor the identical regulation. In addition, every principal, *i*, maintains a log for every observed principal, *j*. All messages that were sent to or from a principal that observed are recorded in this log. Then, *i* makes sure that the entirety of its logs versus the system requirement. If principal *i* discovers that principal *j* has not up until now propelled the message, *i* must have sent in its most recent experiential state. After that, *i* doubts *j* in anticipation that the message is sent. If *j* has transmitted a message, it should not to have transmitted according to the condition; subsequently, *i* uncovers *j* [94].

There are a number of assumptions on which FullReview is based. We claim the most important among them are: 1) a formal system specification; and 2) a trusted and dependable

medium for communication. PeerReview is a derivative from the FullReview. According to the FullReview's assumptions, it changes its own as follows:

- Only a full copy of the log is maintained by every principal. Other logs are recovered whenever necessary.
- Log consistency is maintained by "tamper-evident" logs and consistence protocol with the set of messages that it has traded with all correct principals.
- Witnesses are those small sets of communication principals that are linked with each other. They assemble proof about the principal, check if it is correct or faulty, and ultimately make the consequences accessible to the remainder of the system.
- A reference execution of the principal is used by the system for checking the logs for defective performance.
- A principal that does not react to a few messages is handled with challenge/response protocol by the system.

In order to put accountability into effect, PeerReview should maintain a protected record of the transmissions of every principal. In addition to this, it must have the ability to sense if that trace has been altered. PeerReview provides a record of such type by the means of a method stimulated by protected logs. A log is append-only, restraining every input and output of an exacting principal's state machine in sequential order. The log as well holds episodic state snapshots in addition to several explanations from the detector module. It is essential to make certain that a node cannot put in an access to its log for a message that it has by no means acknowledged. Furthermore, the system would make sure that a principal's log is complete. That is, it holds a record for every message that is either sent or received. As soon as a principal *i*, transmits a message *m* to principal *j*, *i* has got an obligation to transmit *m*. In addition, *j* should

consign to have acknowledged *m*. They get hold of an authenticator from the additional principal incorporated in the message and its acknowledgment, correspondingly. This authenticator envelops the equivalent log record [82, 94].

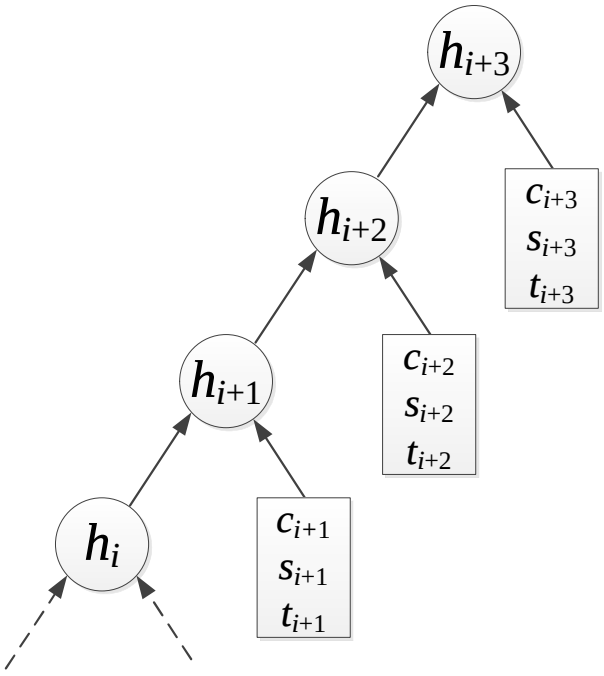


Fig. 2.15. A hash chain with its linear log files [94].

A log record for an acknowledged message should comprise a corresponding authenticator; consequently, a principal cannot create log records for messages on any account it acknowledged. While *i* is getting ready to send *m* to *j*, it generates a log record and sends the product to *j*. Supposing the signature is not legitimate, *j* rejects *m*. Otherwise, *j* produces its own log entry and gives an acknowledgement to *i*. If *i* does not take delivery of a suitable recognition, *i* sends a negative acknowledgement to *j*'s onlookers. A defective node can make an effort to flee from discovery through observing in excess of one log or else a log with numerous undergrowths.

On the way to avoid this assault, the system use the information that a principal is capable of producing a linking log section for all pairs of authenticators that it has continually signed if and only if it preserves a distinct, linear log. If a principal *i* obtains authenticators from a different principal *j*, it should ultimately advance these authenticators to the observer set. As a result, the witnesses attain provable confirmation of the entire message *j* has sent or received. From time to time, every witness decides on the authenticators with the least and the maximum sequence number and dares *j* to go again to visit the entire log records in this range. If *j* is right, these log records figure a hash chain, as shown in Fig. 2.15, which restrains the hash values in all the supplementary authenticators. If they do not, it has acquired provable verification that *j* is defective [94].

2.3.4 Accountability Logic

Logic proof has been widely used in the formal analysis of diversified protocols. It is regarded as an effective way to analyze the accountability of a secure system. To date, there is a wide body of literature [95-102] on accountability logic, most of which is designed for electronic transaction. BAN's logic [95], which is known as the first logic in the analysis of secure protocols, can be used for authenticating and uncovering flaws. Nevertheless, it also generates controversy and confusion under certain conditions. Fortunately, this drawback has been addressed by Abadi and Tuttle in AT's logic [96]. AT's logic is developed from BAN's, but it has more compatible logic and is easier to use than BAN's. In 1993, Syverson [97] mentioned that both BAN's and AT's logic could not capture flaws caused by "casual consistence attack." This is because not every participant holds consistent records of communication history. In order

to logically reveal such flaws, Syverson improved AT's logic by adding temporal formalisms. In practice, however, it is hard to manipulate due to the complexity of the AT basis. In 1995, Stubblebine [98] introduced a notion of recent-secure authentication into the previous logic. His logic involves three temporal properties: *at*, *notbefore*, and *notafter*. These time properties are set as constraints for the participant's authentication. Later, Stubblebine and Wright (SW) [99] extended BAN's logic for a better temporal description based on Syverson's work. Accordingly, the three temporal properties of SW's logic are: 1) at a certain time *t*, 2) at a certain time between t_1 and t_2, and 3) at all times between t_1 and t_2 [99].

In addition, Kailar [100] proposed accountability logic for electronic commerce protocols, such as payment protocols and public key distribution protocols. He defines accountability as a property whereby the association of a unique originator with an object or action can be proved to a third party. Provability has an important role in the analysis of accountability. Since time-critical applications require proofs that guarantee the temporal activities of each principal, Kailar's accountability logic can be extended for use in analyzing such applications [101]. Although the original logic allows some temporal context, such as *During* and *Until* properties, to be added to represent the validation period of security-related information (e.g., a time-critical delegation key), Kudo [101] extended the Kailar's logic so that it could represent the temporal accountability. Based on Kailar's logic, Kudo added 9 new logic constructs (e.g., *timestamp*, *at*, *before*, *after* etc.) and 10 new logic postulates (e.g., *A CanProve x* generated *at t*, *A CanProve x* generated *before t*, etc.). Liang *et al*. [102] claimed that the seventh logic postulate of Kudo' logic had not fully considered. It could not prevent replay attacks. By adding the integrity verification based on timestamps, Liang's logic extended 4 more logic constructs and 2 more logic postulates (e.g., *x At t*, *x Freshbefore t*) on Kudo's. However, without support from a

trusted third party (TTP), Liang's logic will have no difference to Kudo's. Since no TTP is involved in our own designs, we would like to utilize Kudo's logic provability in this dissertation.

CHAPTER 3

SMART GRID: HAN ACCOUNTABILITY

Smart grid is a promising power delivery infrastructure integrated with bi-directional communication technologies that collects and analyzes data captured in near-real-time, including power consumption, distribution, and transmission [2]. According to these data, it can provide predictive information and relevant recommendations to all stakeholders, including utilities, suppliers, and consumers, regarding the optimizing of their power utilization [2]. By two-way electrical flow, consumers are able to sell their surfeit energy back to utilities [2]. In other words, smart grid is a complex system of systems.

Nationwide deployment and popularization of the smart grid require decades of work. Bringing new markets into the grid is encouraged before it can be fully accomplished. It is worth mentioning that the interests of all stakeholders should be considered during development. As such, homeowners must be taken into account. Since enabling consumer participation is a major characteristic of the modern grid, homeowners' considerations are extremely important. As we know, their primary concern regarding power usage is the monthly power bill sent by their service providers (e.g., power utilities). If possible, homeowners would rather know the details of their power usage than simply a bill with a total consumption. Albeit the real-time, or day-today, consumption of electricity could be revealed by the smart meter, we still doubt its reliability: the utility, or the smart meter itself, may alter transmitted data to suit someone's interests or for some other possible reasons (e.g., due to the fact that they are under attack or malfunctions). As a

consequence, a homeowner could have two different electric bills: one from the utility's meter and one from the home meter. Furthermore, in smart grids, prices change with time such that traditional billing method using a unit price is no longer feasible. Therefore, the exact times when power is used are important and should be made accountable. To solve the above problems and to make the smart grid in home areas reliable are the two major motivations of this chapter.

In this chapter, after reviewing home metering system in smart grids, we design an accountable communication protocol for home area network that uses a peer review strategy. Under certain assumptions, the following three major contributions are made in this chapter:

1. A smart meter can prove the correctness of any smart appliance in a home area.
2. A group of smart appliances can prove the correctness of the smart meter.
3. A service provider can prove the correctness of the smart meter.

The rest of this chapter is organized as follows. Section 3.1 discusses how an accountable metering system for a home area smart grid can be designed and deployed. Section 3.2 analyzes and proves the system accountability by accountability logic. Section 3.3 gives our simulation results. Finally, we conclude this chapter in Section 3.4.

3.1 Accountability in Home Area

Although the framework and blueprints of the smart grid have been discussed in recent years [7-11], a specific standard for its implementation is still to be determined. Two steps need to be clarified before designing an accountable system for the AMI (Advanced Metering Infrastructure) in a home area: to build a possible architectural framework for its implementation, and to identify potential security problems.

3.1.1 Architecture

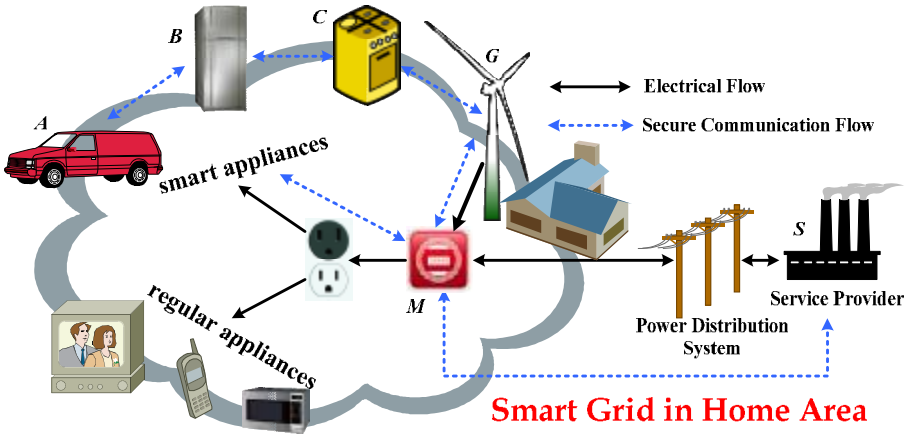


Fig. 3.1. Smart grid in home area.

Based on the smart grid characteristics and system framework, we proposed a reasonable architecture for a home area smart grid, as shown in Fig. 3.1. Note that it works for Building Area Network (BAN) and Industrial Area Network (IAN) as well. As illustrated in Fig 3.1, a smart meter, *M*, acts as a middleman between the service provider, *S*, and home appliances (e.g., *A*, *B*, and *C*). It acts as a gateway, which monitors all incoming and outgoing electricity flow. Meanwhile, it also records power consumption and generation in home areas. We divide electrical appliances into two categories based on their communication capability. One refers to smart appliances and the other to regular appliances. In this specific case, only smart appliances have the ability to exchange information or ***message*** (e.g., market price, trading price, and consumption logs) with others, including the smart meter. They are also capable of recording those *messages*. For those regular appliances that are not interactive, the smart meter simply monitors their activities on corresponding power supply ports. In a modern power grid, most

families would typically equip a power generation and storage device, denoted as *G*. We assume that such equipment is a type of smart appliance. Due to the fact that regular appliances have no communication capabilities, we simply assume that all appliances in future home areas will be smart appliances.

3.1.2 Problem Statement

Conventional metering systems charge electricity consumption according to its reading at the end of each month, as shown in Fig. 3.2. If the meter reading says that *n* kWh has been used within a month, the bill (aka. service amount) without tax will be the product of *n* and a unit average price (denoted as *m* dollars/kWh). Basically, *m* is predefined and published by the service provider. It does not change very often. Therefore, it may be regarded as a constant value.

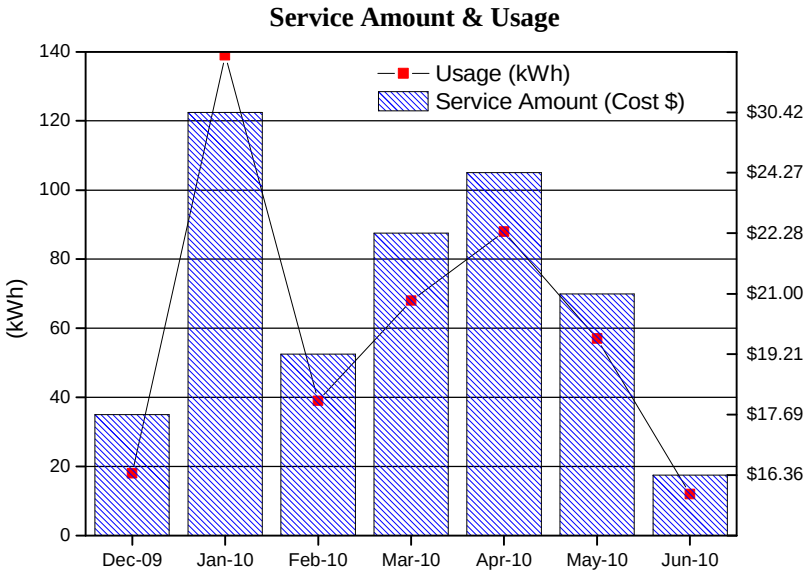


Fig. 3.2. Conventional service amount and usage chart.

Unlike the simple conventional approach, a modern power grid will use smart meters to read electricity usage at a predetermined requested interval (e.g., daily, hourly, or per minute). Those reading data is subsequently stored locally, and then transmitted to the service provider as it would usually. At higher levels, the smart meter will get a real-time unit price (aka. market price) from the service provider or other market via a bidirectional wired or wireless network. Together with the powerful energy management of the AMI, households can not only make economic choices based on dynamic prices, but they can also shift, load, and store or sell surplus energy. Hence, calculating the service amount in such a new power infrastructure is difficult.

Basically, only two key factors affect the bill: 1) the real-time power usage and 2) the market price. The smart meter can obtain both aspects in real-time. However, we cannot simply do a multiplication to get the service amount since the market price is not a constant value and may vary from time to time. For example, the price could remain high during peak hours or high demand periods due to electricity shortage. When outside of peak periods, the price is decreased accordingly. The price may also become affected by local weather conditions. Continuous cloudy or rainy days may reduce the local production of solar energy and then the price might increase. However, if a strong hurricane follows, the price will reasonably fall since it enhances wind power generation at the same time. Hence, it is hard to predict the exact market price at a particular time and a specific location. We instead maintain a record of fore-passed market price. Current solutions, reported by the U.S. DOE [2], take three typical tariff forms: time of use (TOU), critical peak pricing (CPP), and real-time pricing (RTP). TOU pricing is solely based on a peak or off-peak period designation. Prices are set higher during peak hours. Under CPP, prices during peak hours (basically some short periods within a year) are set at a much higher level compared to under normal conditions. RTP pricing is much more flexible, in that hourly prices

are differentiated according to the day-of or day-ahead cost of power to the service provider. Actually, pricing in the smart grid is an interesting and essential open issue that must be addressed. The author in [103] argued that a price response demand mechanism should be introduced in the smart grid. Since pricing is not our primary scope in this paper, we simply assume that the real-time market price may be obtained in a secure and feasible way (via service provider or third party, e.g., markets). Under such conditions, we reasonably suppose that, given any past time *t*, the market price can be determined by a function *M(t)*. As it is a dynamic feature, *M(t)* should be a non-linear and random curve regarding time *t*, as illustrated in Fig. 3.3.

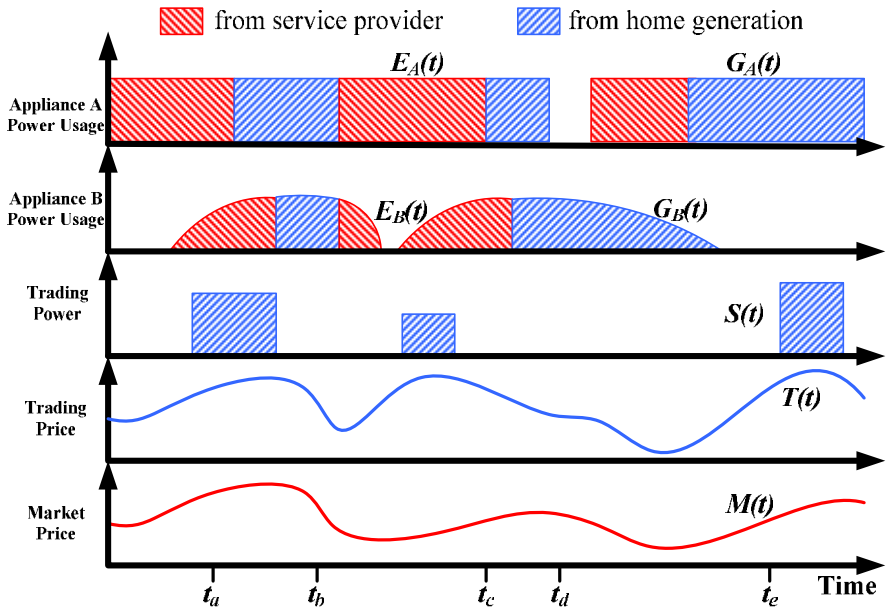


Fig. 3.3. Aggregation information in the smart meter.

Another possible factor affecting the service amount is the presence of a home generated power system (e.g., wind or solar energy). Without consideration of its own consumption, the generated energy may be divided into two parts: those consumed by other electrical appliances at

home and those is sold back to the service provider. Both of them are monitored and recorded by the smart meter, but only the trading portion impacts the service amount. Note that the trading price might be the market price or could possibly even be set by the homeowner. Here we suppose that the trading price is a non-linear function of *t* and denoted as ***T(t)***.

Fig. 3.3 is an example of energy usage in a modern power grid. We denote purchased energy (from a service provider) as ***E(t)***, self-consumed energy (from home generation) as ***G(t)***, and trading power as ***S(t)***. They are all functions with respect to time *t*. If there is no power consumption or sale event during a period, the relevant functions will be zero. Given any time period from t_a to t_b ($t_b > t_a$), the total service amount denoted as $Bill(t_a,t_b)$ should be:

(3.1) $$Bill(t_a,t_b) = \int_{t_a}^{t_b} \left(M(t)\cdot E(t) - T(t)\cdot S(t)\right)dt$$

In equation (3.1), ***E(t)*** can be obtained by attaining the sum of every individual consumption (denoted as $E_i(t)$ where *i* is the name of electrical appliance). For each appliance *i*, the service amount from t_a to t_b ($t_b > t_a$), denoted as $Bill_i(t_a,t_b)$, can be determined by:

(3.2) $$Bill_i(t_a,t_b) = \int_{t_a}^{t_b} M(t)\cdot E_i(t)dt$$

Equation (3.1) can thus be rewritten as:

(3.3) $$Bill(t_a,t_b) = \sum_{i=A,B,\ldots} Bill_i(t_a,t_b) - \int_{t_a}^{t_b} T(t)\cdot S(t)dt$$

According to the former, it is not difficult to see that computing service amounts in a smart grid is indeed a complicated procedure. Many factors in the smart meter may affect the final bill. Any alternation, forgery, delay, or removal of those historical records may lead to a different price. Although we could equip secure smart meters to enhance reliability, it is still possible for homeowners or cyber attackers to manipulate the smart meter for their own interests. In addition, when the service provider brings alternative bills to a homeowner, whom should we

trust? Since most service providers rely on meter readings, ensuring a secure and a reliable smart meter is the primary task.

We consider an entity as ***correct*** only if it strictly follows a given protocol. Otherwise, we regard it as ***faulty***. Here we use smart appliances as witnesses to prove that the smart meter is *correct*. The witness idea was inspired by the PeerReview system [94]. In this case, three new problems should be addressed. First, a smart appliance itself may have errors or be controlled by a malicious person. To make every *faulty* smart appliance detectable is necessary (***Challenge 1***). Second, since appliances have limited capabilities for communication and storage, designing a feasible, observable mechanism for witnesses is also required (***Challenge 2***). Third, home-generated power is managed solely by the smart meter. Other smart appliances do not know where the power load comes from; it may be supplied by the free home generation, or purchased from the service provider. Without supervision, the smart meter may deny that during a certain period an appliance was using power from the service provider (***Challenge 3***). In the following sections, we will describe our design of accountable AMI that addresses these challenges.

3.1.3 Terms and Assumptions

Terms

- $\{A, B, \ldots\}$: a set of communication participants in the smart grid, known as ***principals***. Specifically, *M* stands for the smart meter, *G* represents as the home generation and storage device, and *S* refers to the service provider.
- $\{m, m', n\}$: a set of *messages* or *message* components.
- $\{t_i \mid i = a, b, \ldots\}$: a set of time points.

- $\{K_i, K_i^{-1}\}$: a pair of public/private keys of *principal i*.
- $\{m\}K_i$: *message m* encrypted with the public key of principal *i*.
- $\{m\}K_i^{-1}$: *message m* encrypted or signed with the private key of principal *i*.

Assumptions:

1. Every electrical appliance *i* in the home area is a smart appliance with sufficient storage space and a constant capacity factor P_i (kW).
2. The running state of each appliance (e.g., on or off) is known by the others in real-time.
3. Functions of market price ***M(t)*** and trading price ***T(t)*** are authenticated by the service provider. Every smart appliance shares these functions at the same time.
4. There is a function *w* that maps each appliance to its set of witnesses. Suppose that, for any appliance *i* in a home area, the set $\{i\} \cup w(i)$ contains at least one *correct* appliance.
5. A *message* sent from one *correct* appliance to another will eventually be received.
6. Each involved communication *principal* uses PKI technology to identify itself; they can sign *messages*, but a *faulty principal* cannot forge the signature of a *correct* one.
7. A home generation and storage device *G* must record its own power load ***G(t)*** truthfully. Each appliance *i* will record its power consumption that supplied from *G*, denoted as $G_i(t)$.

Assumption 2 depends on circuit/communication designs which may be achieved by particular sensor units. For simplicity, we suppose that Assumption 2 can be met. More specifically, we suppose that there is a function $R_i(t)$ that records the running state of appliance *i*. When *t* is within the running period of *i*, $R_i(t)$ is granted to 1; otherwise, $R_i(t)$ is set to 0. In Assumption 7, $G_i(t)$ is given by the smart meter. Because appliance does not know where the

supply comes from, smart meter should provide such information. $G_i(t)$ will be signed by smart meter so that it can be further verified with $G(t)$.

3.1.4 Accountable Protocol

Since the power usage of appliance i can be determined by its capacity factor P_i and running state $R_i(t)$, the equation (3.2) for its market service amount can be rewritten as:

$$MPA_i(t_a, t_b) = P_i \cdot \int_{t_a}^{t_b} M(t) \cdot R_i(t) dt \tag{3.4}$$

According to equation (3.4) and Assumption 1 (for flexible P_i, please see discussions in section 3.2.1), if any *principal* j ($j \neq i$) holds P_i, $M(t)$, and $R_i(t)$ at the same time, j is able to determine i's market service amount for any past period. Notice that j still does not know the exact service amount of i, since j has no knowledge of i's power source. If i were using home generated power all the time, i's service amount would be zero. For auditing, i's market service amount can also be specified by:

$$MPS_i(t_a, t_b) = \int_{t_a}^{t_b} M(t) \cdot \left(E_i(t) + G_i(t)\right) dt \tag{3.5}$$

Next, we borrow some ideas from the PeerReview system [94]. Given any period from t_a to t_b, $MPA_i(t_a, t_b)$ should equal to $MPS_i(t_a, t_b)$. Based on this fact, we can design a deterministic mechanism in order to detect *faulty principals* in a home area. Under our proposed architecture, each appliance i has two modules for accountability: a log module L_i and a detector module D_i. L_i generates a complete evidence log of i's power usage. D_i checks other logs to tell whether faults are, or are not, present. Informally, *faulty(j)* is issued when i can prove that j is abnormal; *suspected(j)* is raised when i has not received an expected *message* from j on time; *correct(j)* is released otherwise. Our design therefore follows the following protocols:

- When a new appliance i is plugged in, i will sign P_i with its unique signature K_i^{-1} and broadcast $\{P_i\}K_i^{-1}$ among all ***principals*** in the home area.
- Smart meter will notify each appliance if it currently uses home generated power.
- Every appliance has one copy of its own log, which is ensured by the tamper-evident log mechanism (see section 2.3.3.2) [94]; other logs will be retrieved when required. Appliances exchange just enough ***messages*** to prove themselves.
- Each appliance is mapped to several other appliances. They act as witnesses that collect its log, check its correctness, and report the results to the rest of the system.
- A commitment protocol [94] is adopted in order to ensure that witnesses will retrieve exactly the same log as the target appliance owns. It also guarantees that no one can deny a received ***message***.
- This protocol uses a challenge/response scheme [94] to address the problem that some appliances do not respond or fail to acknowledge that ***messages*** were successfully sent.

Next, we will demonstrate how it works in detail. Initially, every new appliance i will be assigned a set of witnesses w_i by the smart meter. Then, i will sign P_i with its unique signature K_i^{-1} and send $\{P_i\}K_i^{-1}$ to the smart meter and each member of w_i. When i is running, L_i generates a tamper-evident log to record its power usage. Since the smart meter will notify i regarding its power source, the log will record both $E_i(t)$ and $G_i(t)$. In order to check whether i is ***correct*** or not, each witness of w_i will periodically request its most recent log segment. Suppose that the last audit time is t_a and the current time is t_b. In this case, i first requests and records the latest $M(t)$ and $T(t)$ from the smart meter. Then, it sends back each and every one of the log entries since time t_a, together with the corresponding market service amount determined by equation (3.5).

Specifically, the response *message* m_i should be $\{t_a, t_b, E_i(t), G_i(t), MPS_i(t_a, t_b)\}K_i^{-1}$. Note that, m_i could have other information to support certain needs. For instance, adding a sequence number to prevent replay attacks. In this paper, we only focus on accountability part for simplicity. When a witness j ($j \in w_i$) receives m_i (using K_i to verify m_i), D_j will recalculate i's market service amount $MPA_i(t_a, t_b)$ by equation (3.4) according to its own records of P_i, $M(t)$, and $R_i(t)$ (refer to assumptions 1, 2, and 3). If the difference of $MPA_i(t_a, t_b)$ and $MPS_i(t_a, t_b)$ is tolerable (i.e., less than a predefined threshold Δ), D_j will issue *correct(i)*; otherwise, *faulty(i)* is issued (***Challenge 2*** is addressed). Since we use a challenge/response scheme here, every appliance i must respond to the requests from its witnesses, or else *suspected(i)* will be indicated. We also adopt the commitment protocol here, so that all signed *messages* may become evidence against *faulty* appliances. Because there is always a *correct* witness j within w_i (Assumption 4) and all delivered *messages* will be received (Assumption 5), a *faulty* appliance i will eventually be exposed by D_j with its indicators: *suspected(i)* or *faulty(i)* (***Challenge 1*** is addressed). To deal with ***Challenge 3***, we consider all appliances in the home area as witnesses of the smart meter. When suspicious are raised against the smart meter, the third party (e.g., the service provider) will retrieve all evident logs regarding $G_i(t)$ from each home appliance i, together with the self-consumed energy record $G(t)$ from the home generation and storage device G. Since every *principal* uses tamper-evident logs to record its behavior, any mismatch between $\sum(G_i(t)+E_i(t))$ and $G(t)+E(t)$ will prove that the smart meter is not *correct* according to Assumptions 4 and 7.

The protocol described above has addressed three aforementioned challenges. Convinced evidences are able to eliminate the questionable charges on the final bill. As the message latency, throughput, and traffic overhead, paper [94] has shown that this peer review mechanism is scalable in distributed system based on experiments and mathematical analysis.

3.2 HAN Scheme Analysis

3.2.1 Assumption Analysis

The proposed protocol only works under certain assumptions. In Assumption 1, all electrical devices in home areas are determined to be smart appliances. In fact, the smart grid should obtain downward compatibility. Current regular appliances may still work in the modern power grid. For those appliances with no capabilities of communication, finding appropriate witnesses for them subsequently becomes a problem. Considering regular appliance issues, our protocol needs to be modified in future work. In addition, we suppose that every appliance has a constant value for its power capacity factor. In reality, this may be a false assumption. Electric cookers and water heaters are good counterexamples. With a flexible power capacity factor, our protocol is unable to detect *faulty* principals. For Assumption 7, if home power generation device *G* can forge its power load before recording it into the tamper-evident log, the accountability goal **G4** cannot be met. Since only the smart meter monitors its behavior in our architecture, it is hard to convince others that *G* is correct solely based on the proof of the smart meter. Making *G* accountable is required in the next step. Regardless of the architectural design, there is still much research to do before we can build an accountable smart grid.

Considering the issue of flexible power capacity factor, we remove the condition of constant value in Assumption 1. Meanwhile, we require an extra assumption (Assumption 8): *every appliance is able to sample others' power capacity factor $P_i(t)$ at a certain time t.* The sampling job can be done by particular sensors. The witness will record the sensors' reading with its "temper evident log". As depicted in Fig. 3.4, the real power capacity factor (continuous area)

can be estimated by using simple integral knowledge on these recorded discrete factors (blocks). Other protocol remains the same. By this means, ***faulty*** principal still may be found with certain possible false reports. This idea should be further studied in the future work, especially when $P_i(t)$ is not accurately sampled.

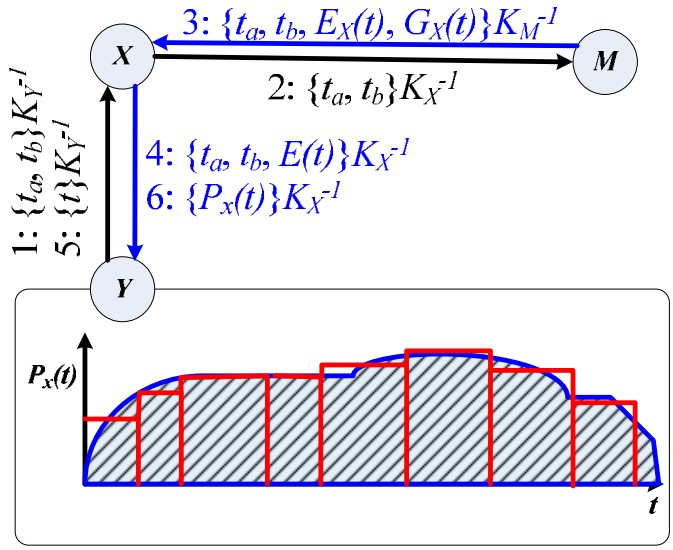


Fig. 3.4. Improved protocol without constant power capacity factor.

3.2.2 Protocol Analysis

Throughout this section, we will analyze the accountability of our HAN protocol by using the same analysis method as in [101]. First, it defines accountability goals. Then it will interpret every *message* into a logical description. After that, the initial assumptions will be restated in a logical way. Based on the logic described in [101], we can eventually prove that our protocol can achieve all accountability goals by using the message interpretation and the initial assumptions.

We present different accountability goals for our proposed protocol based on the definitions and three ***Challenges*** stated in section 3.1.2. Suppose that X is any appliance in the home area and that Y is X's witness. The goals can therefore be described as follows:

G1: M CanProve (*X is faulty or correct*)

G2: X CanProve (*M is faulty or correct*)

G3: Y CanProve (*X is faulty or correct*)

G4: S CanProve (*M is faulty or correct*)

Since an unsigned *message* has no effect on the achievement of goals in accountability logic, we only consider signed ones. The *message* flows can therefore be interpreted as follows:

***Message* 1**: M Receives ($\{P_X\}$ SignedWith K_X^{-1})

***Message* 2**: Y Receives ($\{P_X\}$ SignedWith K_X^{-1})

***Message* 3**: X Receives ($\{t_a, t_b, E_X(t), G_X(t), \{M(t), T(t)\}$SignedWith $K_S^{-1}\}$SignedWith K_M^{-1})

***Message* 4**: Y Receives ($\{t_a, t_b, \{M(t), T(t)\}$ SignedWith $K_S^{-1}\}$ SignedWith K_M^{-1})

***Message* 5**: Y Receives ($\{t_a, t_b, E_X(t), G_X(t), MPS_X(t_a, t_b)\}$SignedWith K_X^{-1})

***Message* 6**: S Receives ($\{\{G_i(t)\}$ SignedWith $K_i^{-1} | i \in$ all appliances$\}$,

$\{G(t)\}$ SignedWith K_G^{-1}, $\{E(t)\}$ SignedWith K_M^{-1})

The initial state assumptions required in the analysis are:

A1: Y Receives ($\{P_X\}$ SignedWith K_X^{-1}) => (Y CanProve (P_X isTrusted))

A2: X Receives ($\{E_X(t), G_X(t)\}$ SignedWith K_M^{-1}) =>

(X CanProve ($E_X(t)$ isTrusted) and ($G_X(t)$ isTrusted))

A3: X Receives ($\{M(t), T(t)\}$ SignedWith K_S^{-1}) =>

(X CanProve ($M(t)$ isTrusted) and ($T(t)$ isTrusted))

A4: Y CanProve ($R_i(t)$ isTrusted)

Using the above formal definitions, our protocol accountability can be proved as follows:

***Message* 1**: When *M* receives *message* 1, *M* knows it was sent by *X* based on its unique signature. Since *M* can monitor *X*'s power usage, P_X can be verified by *M*. If P_X is not true, *M* can claim *X* is *faulty*. Otherwise, *M* can prove the following statement by applying the accountability postulate [100, 101]: *M* CanProve $(X \text{ says } P_X)$ and $(P_X$ isTrusted$)$. When a suspicion is issued against P_X, this statement can be used as an evidence to prove (P_X isTrusted). This is the accountability goal **G1**.

***Message* 2**: *Y* receives *message* 2 at the same time as *M* receives *message* 1. *Y* can prove the following statement by applying the accountability postulate and **A1**: *Y* CanProve (X *says* P_X) and (P_X isTrusted). When a suspicion is issued against P_X, this statement can be used as an evidence to prove (P_X isTrusted). This is the accountability goal **G3**.

***Message* 3**: It is required when Assumption 3 is made. *X* will periodically request *message* 3 from *M*. Since *X* knows its total power consumption $cost_{ab}$ during the period from t_a to t_b, *X* can verify $E_X(t)$ and $G_X(t)$ by comparing their summation with $cost_{ab}$. *Faulty(M)* will be issued if the result is not equal. Although *X* could be compromised, at least we know that there must be a *faulty* node between *X* and *M*. Further investigation is needed here. This is the accountability goal **G2**. Then *X* can prove the following statement by applying the accountability postulate, **A2**, and **A3**: *X* CanProve ($\{E_X(t), G_X(t), M(t), T(t)\}$ isTrusted). When a suspicion is issued against $E_X(t)$, $G_X(t)$, $M(t)$, and $T(t)$, this statement can be used as an evidence to prove ($\{E_X(t), G_X(t), M(t), T(t)\}$ isTrusted).

***Message* 4**: It is similar to *message* 3. By recording *message* 4, *Y* can prove the following statement by applying the accountability postulate and **A3**: *Y* CanProve ($M(t)$ isTrusted) and ($T(t)$

isTrusted). When a suspicion is issued against *M(t)* and *T(t)*, this statement can be used as an evidence to prove that they are both trusted. This is also the accountability goal **G2**.

***Message* 5**: It is a key to achieving accountability goal **G3**. When *Y* receives *message* 5, D_Y will process the auditing of this *message*. Together with the statements from *messages* 2 and 4, *Y* can eventually prove the following statement by applying the accountability postulate and **A4**: *Y* CanProve (*X is faulty or correct*). By combining all such statements from every appliance, the accountability goal **G2** will also be achieved. That is, if no *suspected* signal issued among appliances, the total power consumption of all appliances should equal to the reading of *M*. *Faulty(M)* will be issued if they are not match.

***Message* 6**: Through checking the difference between *G(t)* and the summation of $G_i(t)$ for each appliance *i*, *S* can easily verify whether or not the home power supply is correctly recorded by *M*. Hence, we have the following statement: *S* CanProve (*G(t)* isTrusted). If such checking is failed, *S* can directly issue a *faulty(M)* signal against *M*. Otherwise, *S* will further check *E(t)* with its supply records, if possible. For the most situations, the supply records are solely based on previous readings from *M*. How to determine the *M* is misbehavior varies on different utility's policies. If there is a suspicion, *S* can retrieve all log files of home appliances to see whether or not *E(t)* is correctly recorded. By this means, *S* can prove the following statement by using the *message* 6: *S* CanProve (*M is faulty or correct*). This is the accountability goal **G4**.

3.3 HAN Scheme Simulation

In this section, we simulate our protocol running in HAN. Note that equations (3.4) and (3.5) are time-sensitive functions. Different time periods may cause distinct service amounts. If

the smart meter has not synchronized the witness's local time, the witness's calculation result by equation (3.4) could be different from the service amount in ***message*** 5. Therefore, this witness could issue a false report against a ***correct principal***. One possible solution to address this problem is threshold mechanism. By using a predefined value Δ, witnesses will issue a ***faulty(i)*** only when the difference of two service amounts (one from equation (3.4) and another from equation (3.5)) exceeds Δ. In order to choose a better value for Δ to minimize the number of false reports, we need to evaluate the accuracy of detection for ***faulty principals*** on different Δ. Throughout this section, we also simulate our protocol to evaluate its scalability in terms of average message delay, amount of network traffic, and disk space per witness.

We use discrete event simulation method to simplify our experiment. Specifically, we deploy α (e.g., α = 10, 20, 50, and 100) smart appliances and one smart meter in HAN. Each appliance has β (e.g., β = 3, 4, and 5) witnesses and can communicate with the smart meter directly. We assume that all ***principals*** are in the same communication range. The distance between any two ***principals*** is one hop. No forwarding is necessary in our scenario. This is a reasonable assumption due to the limited space in a home area. The constant capacity factor P_i is randomly selected from 0.1 kW to 1 kW. Each appliance will be turned on or off at short intervals. This process will follow a Poisson distribution, whose mean value is uniformly distributed in an hour for each appliance. In addition, witnesses will challenge each observed appliance every few hours. It is also a Poisson process with a mean interval time γ (e.g., γ = 1, 1.5, 2…) hours. A challenge event will only be processed when both observed ***principal*** and its witness are in running status. If the observed ***principal*** is off, the witness will postpone its scheduled challenge time for an hour. For simplicity reasons, we do not consider the home power supply. Hence, there is no trading part in our simulation. The market price ***M(t)*** is a hourly-based

discrete function. According to the DOE report [104], the peak price is 20 cents per kWh and the minimum price is 7 cents per kWh. We also do not consider the propagation time in our environment. Therefore, the propagation time is assumed to be zero.

For every evaluated situation, we run 100 times, 500 time units (e.g., virtual hours) at a time, and take the average value of the outputs as our results. The simulation platform is Windows 7 64-bit, 2GB RAM, and Intel Core2 6400, 2.13 GHz CPU.

3.3.1 *Threshold Effect*

The threshold directly affects the judgment of a witness. In order to evaluate a threshold, we set different Δ values and measure the number of false reports under different conditions. If the rate of false reports goes to zero, that means we find the right threshold value for Δ.

Different wired or wireless devices may have distinct local times with a certain timer resolution. One may be slower or faster than another. Temporal records therefore may vary from each individual. In order to simulate distinct local times for different ***principals***, we adopt the time-driven simulation method [105]. That is, the drift clock for each ***principal*** is subject to three factors: ***offset***, ***skew***, and ***drift***. If current system time is ***t***, the drift clock ***D(t)*** can be presented as:

$$D(t) = offset + skew \times t + drift \times t^2 \tag{3.6}$$

Therefore, the local time ***L(t)*** can be obtained by equation (3.7):

$$L(t) = t + D(t) = offset + (skew + 1) \times t + drift \times t^2 \tag{3.7}$$

As we can see, the three factors have the capability of being positive or negative. In our simulation, all ***offset*** values are uniformly distributed between -0.2 and 0.2; all ***skew*** values are uniformly distributed between -0.002 and 0.002; all ***drift*** values are uniformly distributed

between -0.0002 and 0.0002. The reference clock is the smart meter's local time. In fact, the difference between two service amounts derives from the time drift of the objects and its witness. Intuitively, the maximum value of the difference is the peak price ($0.2/kWh) times maximum drift time (0.4 hours) times the maximum constant capacity factor (1 kW). Hence, we set the threshold value between 1 cent and 10 cents (around the maximum value of 8 cents).

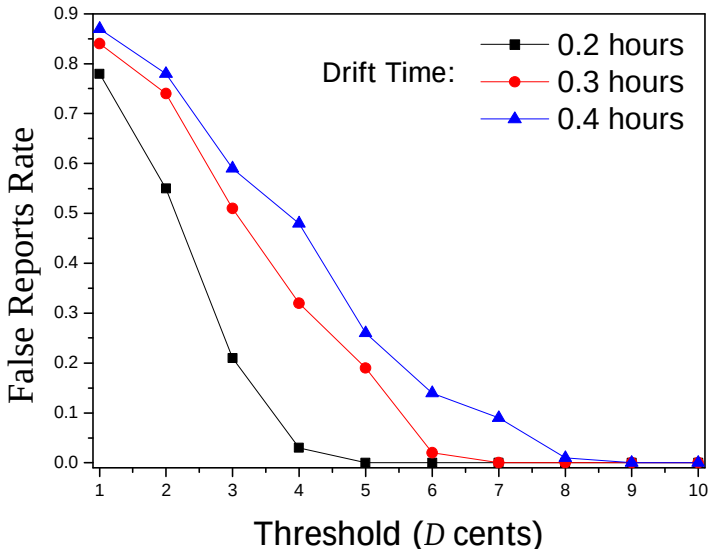


Fig. 3.5. Simulation results on threshold effect.

Suppose all *principals* are *correct*. If there are n packets of *message* 5 in the simulation and m ($m<n$) packets have been reported as *suspected* by witnesses, the rate of false reports is defined as m/n. We set α=10, β=3, and γ=1. As shown in Fig. 3.5, with the increase in value of threshold Δ from 1 cent to 10 cents, the rate of false reports gradually approaches to 0%. When the maximum *drift* time increases, the best value for Δ is increased as well. This is in accordance with what we thought. That is, the threshold value Δ should be bigger than the maximum product value of the drift time, the constant capacity factor, and the peak price.

3.3.2 *Average Message Delay*

Suppose that the average access time for retrieving some data from the log file is 500 milliseconds and the average processing time (e.g., calculate the service amount, send message, etc.) is 50 milliseconds. Messages will be queued in the buffer area when another message is sending. Due to the fact that witnesses will issue challenge messages at hourly-based intervals, the message delay in milliseconds will not impact the performance of the smart appliance. We therefore, only examine the message delay brought by the smart meter. If x messages have been processed by the smart meter and the total delay time is y ms, the average message delay is defined as y/x.

We set β=3, and γ=1. As Fig. 3.6 depicts, with the increase of α from 10 to 100, the average delay is slight; it is in milliseconds and less than a second. Since the number of appliances in most families is less than 100, this result indicates that our protocol is scalable in terms of average message delay.

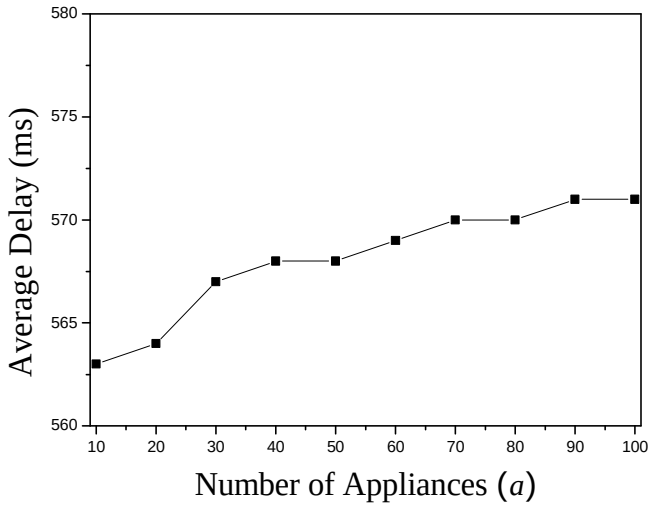
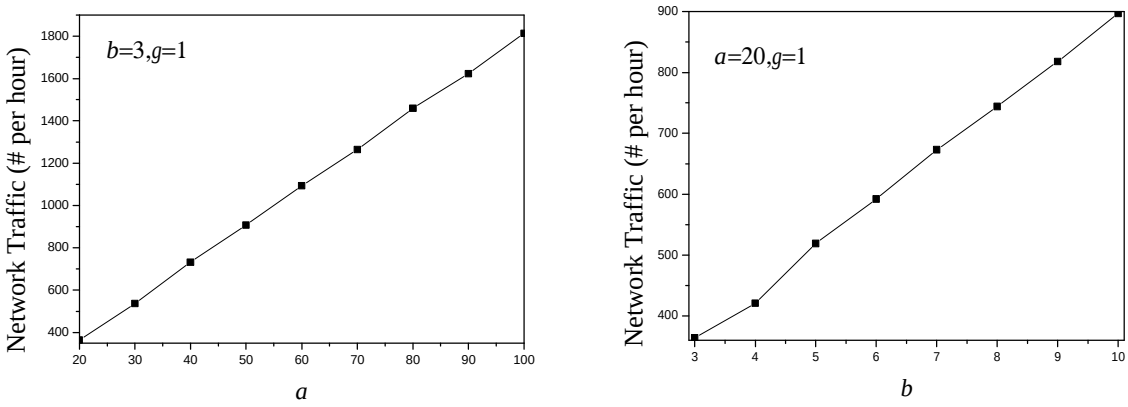
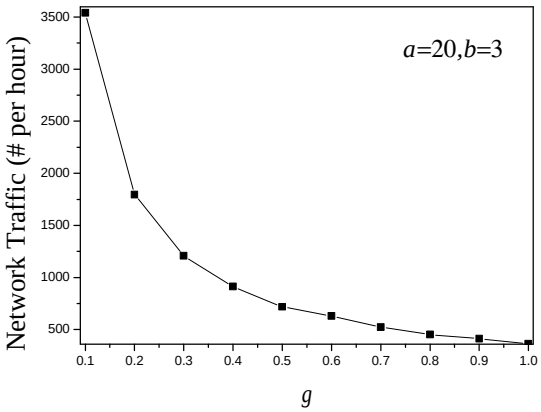


Fig. 3.6. Simulation results on average message delay in smart meter.

3.3.3 *Network Traffic*



(a) α effects (b) β effects



(c) γ effects

Fig. 3.7. Simulation results on network traffic effect.

We measure the network traffic as the average number of ***messages*** that have been sent during one unit time (e.g., a virtual hour). At first, we set α=20, β=3, and γ=1. Then we only adjust one parameter and let the other two remain the same. As Fig. 3.7 (a) and (b) depict, with the increase of α (from 20 to 100) and β (from 3 to 10), the total number of ***messages*** grows linearly. However, the upper bound of the network traffic in each case is just thousands of

messages per hour. It becomes acceptable for communication in a home area. As shown in Fig. 3.7 (c), with the increase of γ (from 0.1 to 1), the total number of *messages* decreased logarithmically. Since γ is the mean interval time for sending challenge *messages*, this curve indicates the larger interval lower the number of challenges in a unit time. Typically, the interval time will be set at least 0.5 hours. Only thousands *messages* occur in an hour. This result indicates that our protocol is scalable in terms of network traffic.

3.3.4 *Disk Space*

Suppose that each log entry will occupy one unit (e.g., 8 kb) of disk space. According to our protocol, log entry could be running state $R_i(t)$, market price $M(t)$, and own power consumption. For each *principal*, the size of the log files for the market price and own power consumption is a fixed value during one time unit (e.g., hourly). Only the log for running state of the observed *principal* will affect the disk space of that witness node. Therefore, we set α=20 and γ=1, then measure the disk space on different value of β.

Fig. 3.8 shows the average logging space (in unit size) in a smart appliance that has been used during one hour. As we can see, the disk space grows linearly with the increase of the number of witnesses. Nevertheless, only several space units are used during one hour. If a unit is 8kb, it will take a week to occupy 1 MB size disk space for logging. In addition, the smart appliance can clear a portion of log files after a period (e.g., monthly). Hence, the usage of disk space is acceptable in HAN. This result indicates that our protocol is scalable in terms of disk space usage.

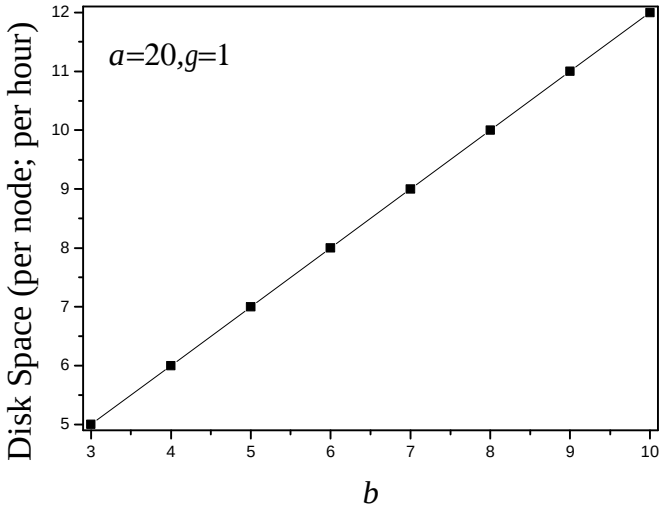


Fig. 3.8. Simulation results on disk space effect.

3.4 Conclusion

A feasible architectural framework for the smart grid in home areas has been presented based on the NIST smart grid interoperability standards (release 1.0). This chapter has designed an accountable communication protocol using the proposed architecture with certain reasonable assumptions. Analysis and simulation results indicate that such a design makes all power loads in home areas accountable.

CHAPTER 4

SMART GRID: NAN ACCOUNTABILITY

We proposed an accountable metering system for smart grid in a home area network (HAN) in chapter 3. Through mutual observations among smart appliances, their power consumptions can be verified that whether they are veritably recorded or not. ***Faulty*** meter, therefore, may be found out if the reading does not match the appliances' records. The records can be served as evidences to against the ***faulty*** meter. It appears that our HAN scheme well solved ***faulty*** meter problem (as described in section 1.2.1) in smart grids. However, it is only reasonable when user claims their power bill is incorrect. Utility company can hardly locate the problem if the ***faulty*** meter holds all the evidences and never sends them out. From the utility's perspective, obtaining all the observed records to find ***faulty*** meters is an unwise choice. On the one hand, it generates too much network traffic that is hard to process and manage. On the other hand, no ***faulty*** meter would like send evidence against itself. In this case, a more efficient and feasible solution is required for the utility. In this chapter, we try to address this problem from the utility's perspective. We therefore proposed an accountable scheme for the smart grid in a neighborhood area network (NAN).

The rest of this chapter is organized as follows. Section 4.1 discusses how an accountable metering system for a neighborhood area smart grid can be designed and deployed. Section 4.2 analyzes and proves the system accountability by accountability logic. Section 4.3 gives simulation results on system performance. Finally, we conclude this chapter in Section 4.4.

4.1 Accountability in Neighborhood Area

4.1.1 Architecture

Despite the fact that there are currently no explicit specifications available for smart grid implementation, we still can reach a consensus that both communication and electric paths are bi-directional. According to the characteristics and blueprint of the smart grid, we can reasonably present a framework for smart grid in a neighborhood area as shown in Fig. 4.1.

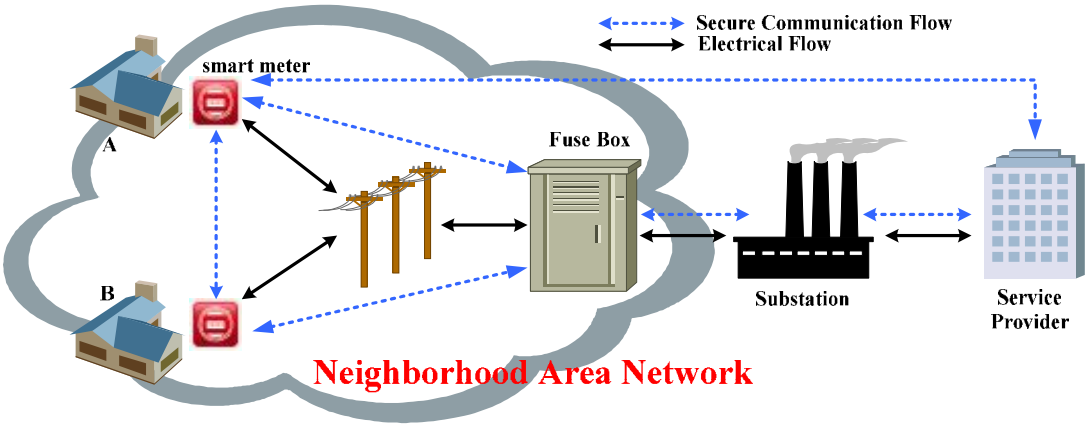


Fig. 4.1. Smart grid in neighborhood area.

In a conventional power distribution system, a community power supply is typically served by the same electric utility company. Within every community, there is a distribution room or a fuse box that delivers power to each customer's home. It is just like a "power router" as described in [44]. This fuse box may equip a meter, denoted as master meter, which measures the aggregated power supply from the service provider but not the power consumption for each end user. For each branch of the supply, utility only installs one meter at the consumer side to

monitor their power usage. Current power grid widely uses automatic meter reading (AMR) technology to remotely collect the meter information. For the sake of saving operating cost, it is more efficient to maintain the same topology of the distribution system. The main difference is that, in the smart grid, all communication and electric flows are bi-directional. A smart meter may directly connect to the service provider via a feasible public communication network (e.g., Internet). It may also connect the service provider through a fuse box and multiple substations using a private corporate network. We do not specify the communication technology preference since we believe the accountability protocol should not rely on that. In addition, all regional smart meters could exchange information with each other. This functionality not only enables power transactions among neighbors, but also helps accountability systems collect convinced evidence. For the case in Fig. 4.1, meters *A* and *B* located in a same community have a common service provider. We refer to the power distribution system in that community a "neighborhood area network."

Considering customers' privacy, a meter would never expose too much information to the others. The communication flows within the NAN can become anonymous by using pseudonym mechanism. The traffic information therefore is not easily associated with its originator. More specific solutions are given in [44] and [45]. Since our goal is to design an accountable NAN scheme, we simply assume that the privacy problem has already been addressed.

4.1.2 Problem Statement

Generally speaking, accountability systems will set a number of witnesses to monitor activities of the observation object. Once an abnormal behavior is detected, those witnesses will

provide relevant observation evidence in order to support their findings. These evidences are typically undeniable and thus, trustworthy. Making power consumption accountable is the primary target of this chapter. In chapter 3, with certain assumptions, we make all power loads in a home area accountable. By this means, a customer can easily verify his/her monthly electricity bill and thus, the ***faulty*** meter will be discovered. In this section, we try to address the problem from the utility's perspective. However, we cannot use the proposed scheme for the smart grid in a NAN directly. Since a meter can only measure one power line at a time, it is very difficult for a household meter to monitor other neighbor's power usage. In other words, if we want to prove the correctness of a smart meter, an additional meter should be installed on the same power line for witness purpose.

As we discussed in section 4.1.1, conventional power grid at most deploys one extra meter (aka. master meter) in the fuse box for one NAN. Once there exists a faulty meter, it is highly possible that the sum of all meter readings in that area does not match the master meter's reading. Notice that such difference may be caused by power loss for normal distribution. For computational simplicity, we regard it as an empirical value that has the capability of being obtained from previous measurements. Based on this value, we may define a threshold Δ so that: if the difference is less than the Δ, the NAN works properly; otherwise, the power usage in that NAN is abnormal. Still, the utility does not know how many ***faulty*** meters exist and where they are located. Instead of sending technicians to inspect every meter in that area, a more efficient way should be considered for the smart grid.

In fact, we may deploy multiple meters in the fuse box for witness purpose. Intuitively, the more witnesses at hand, the fewer steps are required to locate faulty meters. The optimal method is of course one-to-one witness as well as object pairs. However, it is quite impossible to

double the number of meters nationwide. Therefore, a feasible solution would be adopting "intersected grouping" technology to minimize the number of witnesses. As the case shown in Fig. 4.2, there are six household meters in a NAN (e.g., *a*, *b*, *c*, *d*, *e*, and *f*) and three witnesses in a fuse box (e.g., one master meter, *M*, and two additional meters, *A* and *B*). By using witnesses *A* and *B*, six household meters have been divided into 3 groups: Group A (e.g., *a* and *b*), Group A-B (e.g., *c* and *d*), and Group B (e.g., *e* and *f*). According to the witness results, we have the ability to narrow down the searching area for *faulty* meters. For example, if there is only one *faulty* meter in the NAN, but both witnesses *A* and *B* report abnormal activities, we may reasonably infer that the *faulty* one is in Group A-B. Then using witnesses *A* and *B* to monitor *c* and *d* respectively, the *faulty* meter will be found eventually. Nevertheless, if the number of *faulty* meters becomes two or more, things will be more complicated. How to do the grouping and regrouping under different scenarios is therefore, our task in the following sections.

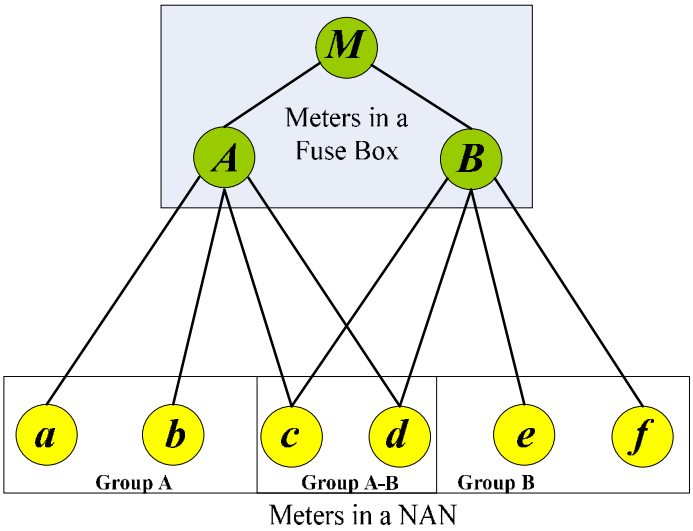


Fig. 4.2. Accountable power distribution system in NAN.

4.1.3 Terms and Assumptions

Terms

Besides the terms defined in section 3.1.3, we have more as follows:

- {*A, B,* …}: a set of meters in the fuse box, known as witnesses. Specifically, *M* stands for the master meter and *S* refers to the service provider.
- {*a, b,* …}: a set of household meters in the NAN, known as observation objects (or objects).
- λ: the number of witnesses in the fuse box except the master meter.
- μ: the number of household meters in the NAN.
- τ: the number of ***faulty*** meters in the NAN; typically, $\tau = 1$ or 2.

Assumptions

1. Smart meter can communicate, and it has sufficient storage space to save log files.
2. The fuse box has at least two witnesses and one master meter inside. All of them are *correct* meters.
3. A witness can choose and change its observation objects at any time.
4. The number of ***faulty*** meters is much less than the total number of meters in the NAN.
5. There exists a function ***w*** that maps each witness to its group of observation objects so that the number of groups in the NAN is maximized.
6. A ***message*** sent from one ***correct*** meter to another will eventually be received.
7. Each involved communication principal uses PKI technology to identify themselves; they may sign ***messages***, but a ***faulty principal*** cannot forge the signature of a ***correct*** one.

Assumption 1 is a fundamental premise of our scheme. Without mutual observation and communication, no one believes a single device who claims itself is *correct*. Assumption 2 may increase the operation cost of the power utility. An economic way to achieve this goal is manually deploy the witness's meter when necessary (e.g., on demand or periodically checking). Assumption 3 depends on circuit/communication designs, which may be achieved by dynamically dispatching power supply to desired branch. This function has already been achieved in power router [44]. The router can switch power supplies (inputs) to different devices (outputs), and may also choose the desired supply from many power sources. It acts like the router in the computer networks. In our assumption, we just adopt its circuit design in the fuse box, and deploy the witness meters at the inputs to monitor those outputs. For simplicity, we suppose that Assumption 3 can be met.

4.1.4 Accountable Scheme

By using the "intersected grouping" method described in section 4.1.2, μ household meters are assigned to λ witnesses according to function w. Therefore, those household meters are divided into several groups. After a fixed time of observation, denoted as t, the witnesses will know which groups are *correct* and which are *suspected* by comparing their readings. Our accountable scheme therefore, may be described as follows:

- When a new household meter i is deployed, M will assign a pseudonym P_i to i with its unique signature K_M^{-1}. P_i will be periodically changed due to privacy consideration.

- Every meter has one copy of its own log, which is ensured by the tamper-evident, log mechanism [94]; other logs will be retrieved when required. Meters exchange just enough messages to prove themselves.
- Each household meter is mapped to several witnesses; the witnesses collect its log, check its correctness by comparing the readings, and report the results to the rest of the system.
- Witnesses will be reassigned observation objects according to function w at set intervals.
- A commitment protocol [94] is adopted to ensure that witnesses will retrieve exactly the same log, as the observation object owns; it also guarantees that no one can deny a received message.
- Using a challenge/response scheme [94] to address the problem that some household meters do not respond or fail to acknowledge that messages were successfully sent.

Next, we will demonstrate how it works in detail. Every household meter i will be assigned a pseudonym P_i and a set of witnesses w_i by M. All the witnesses in w_i will be notified that P_i is their observation object. When meter i is running, it will generate a tamper-evident log to record its power usage, $E_i(t)$. In order to check whether meter i is *correct* or not, each witness of w_i will periodically request its most recent log segment. Such requests are sent by group broadcasting *messages*. Only i in W's observed group will accept the corresponding request, while others simply discard it. Suppose that the last audit time is t_a and the current time is t_b. In this case, meter i will send back all the log entries since time t_a. Specifically, the response *message* m_i should be $\{t_a, t_b, E_i(t)\}K_{Pi}^{-1}$. When a witness W ($W \in w_i$) receives all corresponding *messages* from its observation objects, W will compare its own reading with the sum of all $E_i(t)$ during the same period (from t_a to t_b). If their difference is within a tolerable range (e.g.,

considering the distribution power loss), *W* will claim its observation objects are all *correct*. Otherwise, *W* reports that its observation objects are all *suspected*.

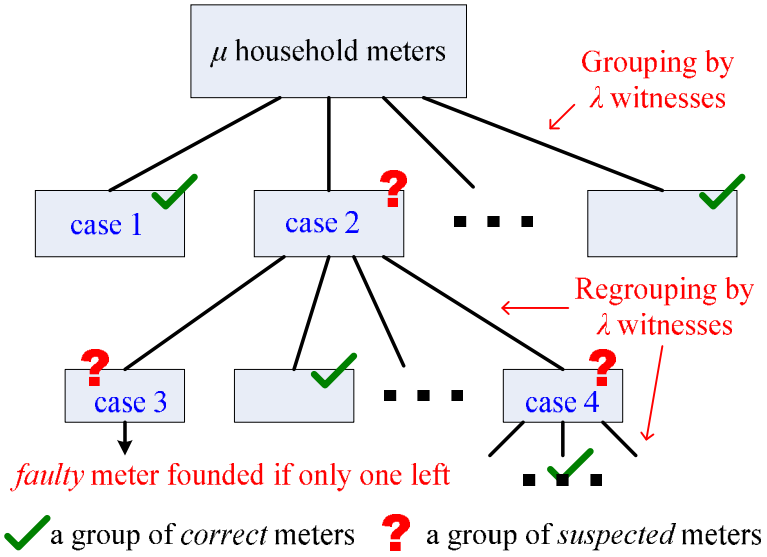


Fig. 4.3. A grouping scheme in accountable NAN.

As an example shown in Fig. 4.3, case 1 is a group of *correct* meters and case 2 is a group of *suspected* meters. For each group that falls into case 2, *M* will regroup it using the same λ witnesses and do further observations in the next time period. This process will be repeated until we reach case 3. Case 3 refers to a *suspected* group with only one meter inside. By then, we may claim that a *faulty* meter is found. As we noticed, not all *suspected* groups have *faulty* meters inside. Taking case 4 as an example, all meters in that group are actually *correct*. The reason this group has been marked *suspected* is because some mutual witnesses find abnormal activities in their observation set. Fortunately, case 4 will be immediately clarified after one-step further observation. Since there are only a few *faulty* meters in the NAN (Assumption 4), case 4 should be an infrequent event.

Apparently, "intersected grouping" is the key of our accountable scheme. There are plenty of ways to do this job. One could group *suspected* meters based on their previous behaviors (e.g., previous *suspected* meters would be grouped together), while others may divide them according to their geographic locations. It is hard to tell which one is better to detect *faulty* meters. They could be anywhere at any time, or they may appear in the same community. The size of NAN is also an important impact factor. Different scenarios may have different result. To use the same grouping strategy for every situation is not a wise choice. We let the utility companies to design the best one for their interests. In this paper, a feasible approach is presented for demonstration purpose.

Table 4.1: Group Algorithm

```
1.   FUNCTION GROUP (witnesses, suspected_meters)
2.   faulty_meters = Φ;
3.   REPEAT UNTIL suspected_meters == Φ
4.     IF |suspected_meters| ↓, THEN    // do regrouping
5.       reset observation sets for all witnesses;
6.       setup enough subgroups to hold suspected_meters;
7.       uniformly assign suspected_meters to subgroups;
8.     ELSE      // means all witnesses report suspected
9.       check suspected_meters with the most witnesses;
10.      check the rest of meters;
11.      regroup suspected_meters again;
12.    END IF
13.    wait for a fixed period of time;
14.    check the readings for each witness;
15.    update suspected_meters and faulty_meters;
16.  END REPEAT
17.  RETURN faulty_meters
```

To call the above function GROUP (shown in Table 4.1), initially set the input parameters to all witnesses and all household meters respectively. It is a straightforward

algorithm, which continuously reduces the number of *suspected* meters by separating the *faulty* meters from *correct* ones. Lines 4 to 7 are normal grouping procedures. In line 6, the word "enough" means each subgroup at least has one *suspected* meter, and any two subgroups must have at least one different witness. Lines 8 through 12 will be called when all witnesses find problems. In this case, we first check those *suspected* meters with the most witnesses. Then check the others (i.e., by the order of number of witnesses) after that. Due to Assumption 4 (in section 4.1.3), it is highly possible to find the *faulty* ones in the groups that have the most witnesses. Lines 13 to 15 may identify some meters through observations. The loop will be end until all *suspected* meters are identified. It may also be terminated after a period of time. Instead, this function may return a group of *suspected* meters.

4.2 NAN Scheme Analysis

As we may see, λ witnesses in the fuse box can at most divide a group of household meters into 2^{λ} subgroups. Since every household meter should have at least one witness, the number of valid subgroups is 2^{λ}-1 (empty subgroup is eliminated). As shown in Fig. 4.3, every regrouping will cause at most 2^{λ}-1 branches for one *suspected* group. The architecture is similar to a classical data structure – B-Tree. Each node (aka. subgroup) in the tree has at most 2^{λ}-1 children. In the first level of the tree, every node has at most $m/(2^{l}-1)$ meters inside. In the second level, the number goes down to $m/(2^{l}-1)^{2}$ for each node. If the tree height is h, we have $m/(2^{l}-1)^{h}=1$. Thus, we may reasonably draw the conclusion that finding one *faulty* meter should cost $O(h)=\mathrm{O}(\log_{2^{l}-1}{}^{m})$ time. In a NAN, the number of household meters μ is typically

less than 10,000. In essence, it only requires less than 6 times regrouping to find a *faulty* meter if λ is equal to 2 or 3. However, if there is more than one *faulty* meter in the NAN, things become much more complicated. The running time for searching the *faulty* meters is depends on variety of factors, such as the number of witnesses (λ), the number of household meters (μ), and the way to do the regrouping. We will analyze it in section 4.3.

Similarly, based on the logic described in [101], we can prove that our protocol can achieve all accountability goals by using the message interpretation and the initial assumptions. Our goal is to find the *faulty* meters in a NAN. Suppose that *x* is any household meter in a NAN and that *W* is *x*'s witness. The goals can therefore be described as follows:

G1: *M* CanProve (*x is faulty* or *correct*)

G2: *W* CanProve (*x is suspected or correct*)

Since an unsigned *message* has no effect on the achievement of goals in accountability logic, we only consider signed ones. The *message* flows can therefore be interpreted as follows:

***Message* 1**: *x* Receives ($\{P_x\}$ SignedWith K_M^{-1})

***Message* 2**: *x* Receives ($\{t_a, t_b\}$ SignedWith K_W^{-1})

***Message* 3**: *W* Receives ($\{t_a, t_b, E_x(t)\}$ SignedWith K_{Px}^{-1})

***Message* 4**: *M* Receives ($\{t_a, t_b, E_W(t), \{t_a, t_b, E_x(t)\}$ SignedWith $K_{Px}^{-1}\}$ SignedWith K_W^{-1}).

The initial state assumptions required in the analysis are:

A1: *M* Receives ($\{P_x$ *is faulty*$\}$ SignedWith K_W^{-1}) => (*M* CanProve (P_x *is faulty*))

A2: *W* Receives ($\{t_1,t_2,E_x(t)\}$ SignedWith K_{Px}^{-1}) => (*W* CanProve (*x is suspected/correct*))

Note that, **A2** will use "intersected grouping" technique to check whether P_x is in a *suspected* group or not. Using the above formal definitions, our protocol accountability can be proved as follows:

***Message* 1**: When *x* receives *message* 1, *x* knows it was sent by *M* based on its unique signature. After that, *x* can use P_x as its pseudonym to communicate with other meters. Since *W* also knows P_x is one of its observation objects, if P_x does not respond *W*'s request/challenge, *W* can claim P_x is *faulty*. It will be sent to *M* for further verification. By applying the accountability postulate [101, 102] and **A1**, we have: *M* CanProve (*W says* P_x *is faulty*) and (P_x *is faulty*). When a *suspected* is issued against P_x, the above statement can be used as evidence to prove (*x is faulty*). This is the accountability goal **G1**.

***Message* 2**: *W* will periodically broadcast *message* 2 to all of its observation objects. Since *x* knows its pseudonym is P_x, it will be received by *x*. Other meters who got this broadcast *message* will discard it. When *W* does not get any response from P_x after a given time, this *message* can be served as an evidence to prove (P_x is *suspected*). This is the accountability goal **G2**.

***Message* 3**: It is a key to achieving accountability goal **G2**. When *W* receives *message* 3 from all its observation objects within a given time, *W* will process the auditing procedure. If there is any one missing (possible delay too much) or whose timestamps (e.g., t_a and t_b) do not match its corresponding challenge *message*, *W* can directly claim the following statement: *W* CanProve (*x is suspected*). Otherwise, the auditing procedure will adopt aforementioned "intersected grouping" technique to filter out *suspected* meters. Given enough time, *W* can eventually prove the following statement by applying the accountability postulate and **A2**: *W* CanProve (*x is suspected or correct*).

***Message* 4**: When *x* is *suspected*, its witness *W* will notify *M* with *message* 4. If there is only one *suspected* household meter, *M* can directly claim the following statement: *M* CanProve (*x is faulty*). For those meters in *correct* groups, we have: *M* CanProve (*x is correct*). Otherwise,

M will reassign all ***suspected*** meters to λ witnesses for further checking. ***Message*** 4 becomes the evidence against ***faulty*** meters. By combining all such statements from every witness, the accountability goal **G1** will be achieved: *M* CanProve (***x is faulty or correct***).

4.3 NAN Scheme Simulation

One goal in this chapter is to achieve accountability in the NAN. As we can see, it relies on witnesses' observations and undeniable log files. Logic proof has been given in section 4.2 for our proposed protocol. The rest part of this section should evaluate performance of witnesses' observations. According to regrouping times and hitting ratio (percentage of ***faulty*** meters in a ***suspected*** group), we analyze the performance of our grouping algorithm in different scenarios.

We use the GROUP function described in Table 4.1 for our experiment. Specifically, deploy λ (e.g., λ = 2, 3, 4, and 5) witnesses and μ (e.g., $\mu = 10^2$, 10^3, 10^4, and 10^5) household meters in a NAN. According to our test cases, manually set τ (e.g., τ = 1, 2, 3, 4, and 5) meters' reading different from their actual consumptions, regarded as ***faulty*** meters. Only one master meter exists in the fuse box that manages all λ witnesses. Each witness is able to communicate with any household meter directly. No matter how they communicate, either via wireless or wired channel, private or public network, ***messages*** will be eventually delivered, safely and in a timely manner.

For simplicity reasons, we assume all meters' local time is synchronized. Because meters barely have constraints on power and computing resources, it may be easily achieved by variety time synchronization methods in computer networks. In addition, we assume there is no power loss during distribution and no prorogation delay. To remove this assumption, one may simply

adopt a predefined threshold and minimize the false report as we did in section 3.3.1. Here we only focus on more important aspects of our protocol.

For every evaluated situation, we run 100 times, 500 time units (e.g., virtual hours) at a time, and take the average value of the outputs as our results. The simulation platform is Windows 7 64-bit, 2GB RAM, and Intel Core2 6400, 2.13 GHz CPU.

4.3.1 *Performance in Unlimited Time*

Given unlimited time, witnesses can eventually find out all *faulty* meters. To evaluate the performance under different scenarios, we use the total times of regrouping as a "criterion." Apparently, less regrouping time means better performance on finding the *faulty* meters.

Typically, the number of household meters in a NAN is ranging from 10^2 to 10^5. We deploy a small number (e.g., 2 to 5) of witnesses in the fuse box to monitor the NAN. As shown in Fig. 4.4, if there are less than 5 *faulty* meters in a NAN, the average regrouping time is no more than 62. It seems unacceptable in reality when we challenge *suspected* meters every one hour for regrouping. The worst case will cost 3 days to locate all *faulty* ones. However, as we can see, about 73.4% (i.e., 47 out of 64) cases can be done within a day, which is somehow tolerable for most situations. Especially, if there is only one *faulty* meter in the NAN, our algorithm is able to find it with average time of 6 hours. When the number of household meters (on each vertical line) increase, the average time of regrouping goes up as well. When the number of *faulty* meters increases by one, the regrouping time could be doubled. Based on the simulation result, we may claim that our algorithm works well when there are only a few *faulty* meters in the NAN.

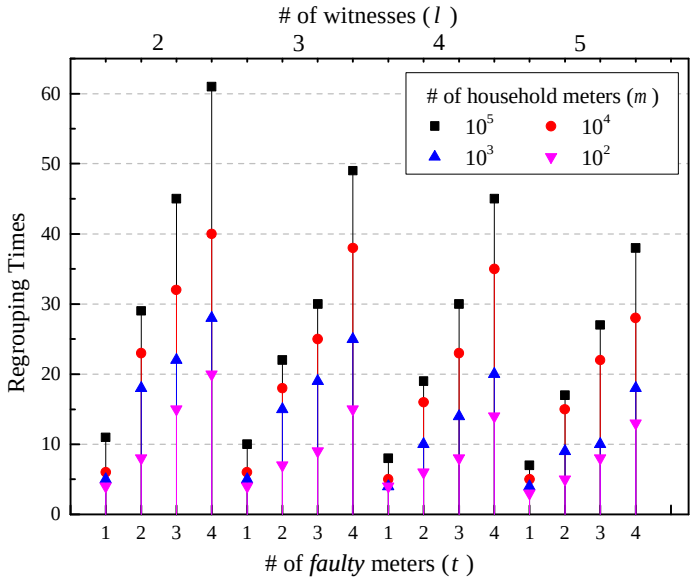


Fig. 4.4. Performance in unlimited time with a little witnesses.

4.3.2 *Performance in Limited Time*

In fact, we do not know how many *faulty* meters out there at the beginning. If there are plenty, our algorithm may cost days to get the result. It is absolutely not acceptable. One solution is to set a timer for the program. Within a given time, witnesses may not find out all *faulty* meters. However, returning a small group of *suspected* meters is tolerable. We could manually check those meters using our traditional way. To evaluate the performance in such case, we use the hitting ratio (i.e., number of *faulty* meters / number of *suspected* meters) as a "criterion." Apparently, higher hitting ratio means better performance on finding the *faulty* ones.

In our simulation, the NAN has ten thousands household meters, which is a very typical case for a community. The timer is set as 100 regrouping times, which could be up to 4 days in reality. When the time is up, we calculate the hitting ratio under different scenarios. As we can see in Fig. 4.5, the ratio drops quickly as the number of *faulty* meters grows up. Adding witness

only gains 10% to 20% hitting ratio at a time. But it is not an economic solution. When the number of *faulty* meters is above 15, the ratio could be less than 60%. Hence, the grouping algorithm is only work for a small number of *faulty* meters. More sophisticated approach should be studied in the future work.

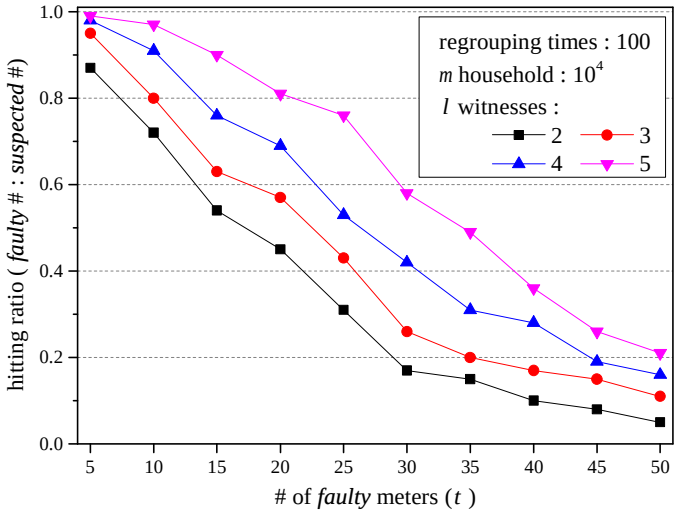


Fig. 4.5. Performance in limited time with different number of witnesses.

4.4 Conclusion

An accountable scheme for the smart grid in a neighborhood area network has been presented based on the NIST smart grid interoperability standards (release 1.0). The main idea is to deploy extra witness meters in the fuse box to monitor all household meters. By intersected grouping technology, we could narrow down the searching area for *faulty* meters. Through a logic analysis and simulations, we argue that our scheme can effectively detect any *faulty* meter within the NAN under some reasonable assumptions.

CHAPTER 5

MSN: TEMPORAL ACCOUNTABILITY

Telemedicine is a technology that uses communications and computing to implement high-quality healthcare regardless of location. Recent technological progresses in wireless communications, micro-electro-mechanical systems, cryptography, and digital electronics have caused the telemedicine system to become more sophisticated. Medical sensor network (MSN) [69] is an example of such promising system. Patients are deployed with multiple medical sensors or wearable devices. These appliances are responsible for recording patients' physical statuses and for transmitting these data to the monitor center via a wireless channel. At the monitor center, the data and corresponding medical records can reveal patients' real-time situations after a series of analysis procedures. Once an undesirable status has been detected, doctors or nurses may take further actions on that particular patient (e.g., remind him/her to take pills immediately via telephone).

Although the new platform saves time for patients to see a doctor, problems still exist in the MSN that cannot be ignored. Medical sensors may have different capabilities, such as detecting electrocardiographs (ECG), heart rate, blood pressure, or pulse rate. All these parameters are important to timely detection and classification of abnormal physical statuses. To obtain accurate sensor readings in unreliable channels is always the goal of ongoing researches. Nevertheless, it is hard to get the ideal readings because of sensors' limitation. On the one hand, a sensor's wireless communication range is limited (typically < 100 feet, due to the limited

power and capacity of the tiny antenna). In order to build a regional and low-cost MSN, we adopt a patient-to-patient (hop-to-hop) transmission relay scheme and "receiver-only" timestamp analysis in our design. The hop-to-hop strategy enlarges the communication range to some extent. The "receiver-only" timestamp analysis saves sensors' power and synchronizes their local clock. On the other hand, sensors have deficient usability and poor security, especially the immature patient privacy-preserving technique. Hence, many hospitals and patients are afraid of using current telemedicine systems. A tradeoff between their usability and credibility needs to be achieved [73]. According to the study in [74], we believe that a multi-hop message communication system cannot be well protected only by typical security technologies (i.e., digital signatures and cryptography). As a complement, accountability and anonymity are required to secure the MSN.

Albeit general system accountability can preserve the integrity and confidentiality for data transmissions, the MSN still has no protection against temporal signal spoofing. It is obvious that the accuracy of an ECG trace depends on the accuracy of temporal signals within each sensor's report. Any change, no matter whether it derives from an attacker's spoofing or comes from a malfunctioned sensor, may lead to quite another result. To locate the problem, we should hold the temporal signal accountable. One important issue is how to synchronize temporal signals among all sensor nodes and wireless devices. It should be an effective approach that is accurate, lightweight, flexible, and comprehensive. Actually, corresponding solutions have been proposed in a similar research area – wireless sensor networks (WSNs) [106]. In order to avoid the large shortening of medical sensor lifetime, a modified DMTS (Delay Measurement Time Synchronization) approach [107] will be adopted in our design. It is so called "receiver-only" local timestamp analysis which has much better energy efficiency in wireless communications.

For the privacy issue, since sensor's ID on patient's body corresponds to the patient's profile record in a medical database, disclosure of information source during wireless communications can cause a violation of the patient's privacy. Moreover, when such MSN platforms are widely deployed in the national medical sites (such as nursing homes, hospitals, etc.), they could become the potential attacking objects of cyber-terrorists. Considering the confidentiality of sensitive medical data, we definitely need an end-to-end security scheme to protect them. It can be achieved through an implementation of the following two crucial MSN components: first, the sensor-to-sensor communication should be secured through low-cost symmetrical ciphers; second, the medical data should also be authenticated and encrypted through extremely light-weight security schemes. Since sensor network security has been studied extensively, we will only focus on how to overcome current privacy problems while preserving temporal accountability in this chapter. To minimize the communication cost and obtain a certain degree of anonymity, we select "Crowds" out of three typical anonymous communication systems [108].

The rest of this chapter is organized as follows: Section 5.1 discusses the problems and challenges; Section 5.2 presents our design and architecture for the new MSN platform; Section 5.3 mainly evaluates our design using logical proof; more insights regarding simulation results are offered in Section 5.4; Section 5.5 is the conclusion of this chapter.

5.1 Problem Statement

In a typical MSN (see Fig. 2.9 and Fig. 5.1), the patient's medical information is normally collected by a wearable wireless device (e.g., PDA). Then it is delivered to nearby

access points (APs) via hop-by-hop wireless communications. Through wired or wireless channels among different APs, the collected information is transferred to a nursing home monitor center. To some extent, this architecture is scalable, manageable, and easily to be deployed. Our work is therefore illustrated based on such architecture.

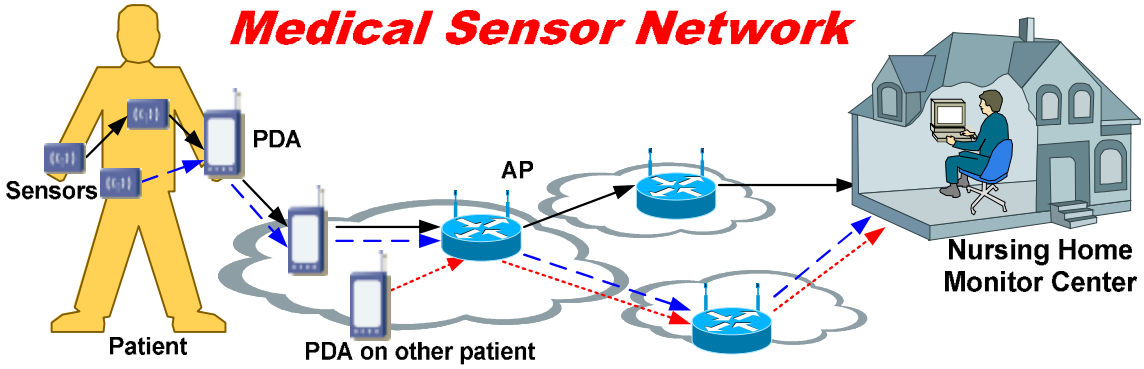


Fig. 5.1. A typical MSN architecture.

MSN can help patients save time and money. It also optimizes medical resources so that every patient is able to receive better treatment than before. Nevertheless, current MSN is not good enough regarding integrity. The integrity not only refers to completeness of transmitting data in a wireless context, but also considers time consistency of delivered data between a sender and monitor center. Unfortunately, most existing work is dedicated to ensuring the data completeness but neglects the time consistency. For instance, ECG anomaly detections depend on the accurate time interval analysis of different ECG signal changes; a simple change of time signals may lead to quite another output. Inheriting generic technologies, such as time synchronization, cryptography, and wireless communication, cannot guarantee the consistency of time signals in the MSN. On the one hand, transmitted data may be delayed, forged, or dropped by an intermediate device along a routing path. We need find out who should be responsible for

this alteration and when it happens. On the other hand, wireless devices may have distinct local times with certain timer resolutions. It is not easy to synchronize all temporal signals with a high resolution, especially in a low-cost wireless sensor network. Based on these two factors, we formalize two challenges in a MSN: ***Challenge* 1**: forgery, alternation, delay, or removal of temporal records may be initiated by the sender, the receiver, or both in conspiracy; ***Challenge* 2**: either the sender is or the receiver's clocks are not trusted, as they may be slow or fast. Another significant concern within a MSN is maintaining trust and confidence between patients and physicians. Maintaining confidentiality of a patient's medical record is of great importance. Nevertheless, it becomes a controversial topic when computerized information systems are considered to handle health data. It is the fear of many medical professionals that the confidentiality of medical and personal data will not be appropriately maintained. Such a fear is not totally unsupported. Anonymous communication technologies can be utilized to address this problem. The temporal accountability, however, is contradictory to anonymity. Evidence of temporal records can be used to reveal the sender's identity. Therefore, our ***Challenge* 3** is: maintain sender's privacy while preserving temporal accountability.

5.2 Communication Protocol

5.2.1 Terms, Definitions, and Assumptions

Terms

- {*A, B,* …}: a set of communication participants, known as *principals*. Especially, *M* stands for the monitor center.

- $\{m, m', n\}$: a set of *messages* or *message* components.
- $\{t_i \mid i = A, B, \ldots\}$: a set of timestamps within the *messages*.
- $\{m(t_i), m'(t_i), n(t_i) \mid i = A, B, \ldots\}$: a set of *messages* with timestamp t_i.
- $\{K_A, K_A^{-1}\}$: a pair of public and private keys of *principal A*.
- $\{m\}K_A$: *m* encrypted with the public key of *principal A*.
- $\{m\}K_A^{-1}$: *m* encrypted or signed with the private key of *principal A*.

Definitions

- ***Temporal Accountability***: for any *message* $m(t_i)$ received by the monitor center *M*, if (t_i is modified by a *principal X*) *at* t_x, *M CanProve* (*X* sees $m(t_i)$, *X* modifies $m(t_i)$, and (*X* says $m(t_i')$) *at* t_x).
- ***Neighbor***: *principle B* is a neighbor of *principle A*, if *B* is in *A*'s communication range.
- ***Temporal Evidence***: system will keep a log file or take similar approaches to record any modification of temporal signals.
- ***Temporal Undeniable***: no one can deny its actions on the temporal evidence.
- ***Preserving/Maintaining Privacy***: sender's identity cannot be disclosed by any user or attacker except the authorized agency (e.g., monitor center).
- ***Synchronize time signals***: all data generated by communication *principals* hold the same (or with a little deviation) time reference clock at the recipient (e.g., monitor center).

Assumptions

1. Monitor center is assumed to be trusted, and its clock is assumed to be accurate.
2. Each *principal* except the monitor center has more than two neighbors.

3. The local time of each *principal* except the monitor center is not trusted.
4. The wireless communication channel is assumed to be unsecured, and all traffic in MSN can be observed by any *principal*.
5. No *message* loss occurs during transmitting in wireless context.
6. Digital signature and *message* encryption algorithm is based on public key cryptography, and no private key can be compromised by intruders.
7. Computing and storage space for the monitor center and APs are assumed to be unlimited.
8. No denial-of-service attack occurs in MSN.
9. All wired communication channels are secured.
10. No IP spoofing attack occurs in MSN.

5.2.2 Temporal Accountability Module

There have only been a few studies [97, 100, 102, 101] conducted on network temporal accountability. Most of them work for electronic transaction in wired network. Unlike the logic of electronic commerce, the "receiver" (monitor center) does not assign a permitted period to each "sender" (medical sensor) in a MSN, since a sender may constantly deliver *messages* with vital signals. However, a medical *message* is time sensitive, which is only valid during a certain period. For example, if sensors detect a heart attack and deliver relevant information immediately, we argue that the medical center should receive this *message* as soon as possible, or it will be useless when the patient is dead after a period. We therefore need a strategy to determine whether a received *message* is fresh or has been modified or postponed for an unacceptable period. Note that either a sender or an intermediate node is not trusted and may

change the time signal for certain reasons. Some may be damaged and others may be manipulated by malicious people; none of these are expected to be received.

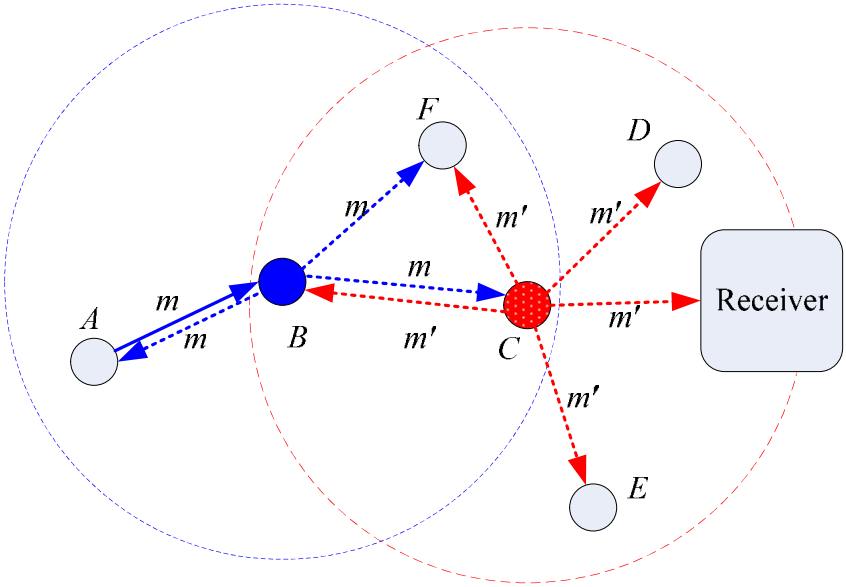


Fig. 5.2. Temporal accountability module.

Based on Assumptions 4, 6, and 9, we just need to consider two transmission scenarios (as illustrated in Fig. 5.1): 1) Sensor-PDA and 2) PDAs-APs. They are both in wireless environment. The major difference is that the latter has more powerful capacities for computation and storage. Since PDAs and APs can be regarded as super sensor nodes in terms of transmitting *messages*, it is reasonable to merge these two scenarios into one abstracted case, as shown in Fig. 5.2. This combined module involves two kinds of components: 1) multiple transferring *principals* and 2) one receiver. As in the Sensor-PDA scenario, the first part represents the sensors while the receiver stands for a PDA. The PDA is regarded as relatively immobile to a patient. It is responsible for receiving and recording all medical data from sensors deployed on the patient and for communicating with other wireless devices to transfer medical

information. The information will eventually be forwarded to the nursing home monitor center via multi-hops among several PDAs or APs. As in the PDAs-APs scenario, multiple transferring *principles* can be regarded as PDAs or APs while the receiver is the monitor center. Hence, if we can achieve temporal accountability in this module, then we address the ***Challenge* 1**.

As we know, to send or forward a *message* in a wireless environment, a *principle* simply broadcasts it within its communication range. Therefore, if a *principle A* sends a *message m* to the monitor center via its neighbor *B*, *A* will receive a broadcast *message m* from *B* when *B* tries to forward it to the destination. Based on this observation, we propose a feasible solution for temporal accountability in the MSN.

Specifically, each *principal* except APs and the monitor center should hold a *memory* to record recently sent, forwarded, or passively received *messages* for further review. As Fig. 5.2 depicts, once a *principle B* wants to send a *message m* to the receiver, *B* will proceed along the following procedures: 1) sets the receiver as the destination; 2) signs *m* with its unique signature key K_B^{-1}; 3) records $\{m\}K_B^{-1}$ in its *memory*; 4) sends $\{m\}K_B^{-1}$ to the next hop *C* via broadcasting. There are three neighbors (*A*, *C*, and *F*) for *principal B* in this case. All of them will receive $\{m\}K_B^{-1}$. Since *A* and *F* are not the next hop of this *message* (we will discuss the routing path in section 5.2.4), they just simply record $\{m\}K_B^{-1}$ in their own *memory*. For *principal C*, it will process and forward this *message* to the receiver. Similarly, *principal F* will receive the forwarded *message* $\{m'\}K_C^{-1}$ from *principal C*. Because K_B and K_C are public, *principal F* can verify the sender's identification of *message* $\{m\}K_B^{-1}$ and $\{m'\}K_C^{-1}$. *F* can also compare the received *message* $\{m'\}K_C^{-1}$ with the *message* $\{m\}K_B^{-1}$ in its *memory*. We denote $\{m\}K_B^{-1}$ is a temporal evidence of $\{m'\}K_C^{-1}$. Once *m* and *m'* are satisfied by a predefined temporal requirement (discuss in the next paragraph), we say *m'* is equal to *m*. In this case, *principal F*

will delete $\{m\}K_B^{-1}$ from its *memory* and record $\{m'\}K_C^{-1}$into its *memory* for further surveillance. Otherwise, we say *m'* is not equal to *m*, and *principal F* will report a suspicious temporal activity to the receiver with relevant temporal evidences (e.g., $\{m\}K_B^{-1}$).

The temporal requirement is not over a threshold value Δ predefined by the monitor center. When *F* receives *m*, *F* will mark the received time as t_1. Similarly, when *F* receives *m'*, *F* gets the corresponding received time as t_2. Note that both t_1 and t_2 are relative to *F*'s local time. After that, *F* will calculate the difference of t_1 and t_2, denoted as Δt. *F* will also calculate the difference between two timestamps within *m* and *m'*, denoted as $\Delta t'$. When $|\Delta t - \Delta t'|$ is no greater than Δ, we say *m* and *m'* are satisfied by this predefined temporal requirement.

Based on the Assumption 2, every *principal* will have at least two observers to monitor its actions. Thus, most malicious modifications on temporal signals will be captured. Then we have achieved temporal undeniable and addressed the ***Challenge* 1** here.

5.2.3 Time Synchronization Module

This section is dedicated to synchronize all temporal records in a MSN. It targets ***Challenge* 2**. In fact, global time synchronization is not necessary required for each communication node in the MSN. Every sensor is an independent node that does not need to know the local time of others. It is only responsible for delivering *message* to the monitor center. The monitor center will address the global time issue. By evaluating different forms of delay in the network, the monitor center can calculate the occurrence time of each received *message*. That means *messages* are "synchronized" at the monitor center. This protocol is so called delay measurement time synchronization (DMTS) [106]. The delay is composed of factors affecting

transmission time from node to node. The synchronization accuracy of this protocol is limited mainly by the precision of the delay measurements along the path. For the sake of ensuring high accuracy and energy efficiency in our design, we need appropriate modifications on DMTS protocol. The modified protocol should also be computationally lightweight. No complex numerical operations are involved. Therefore, we have modified the method proposed in [107].

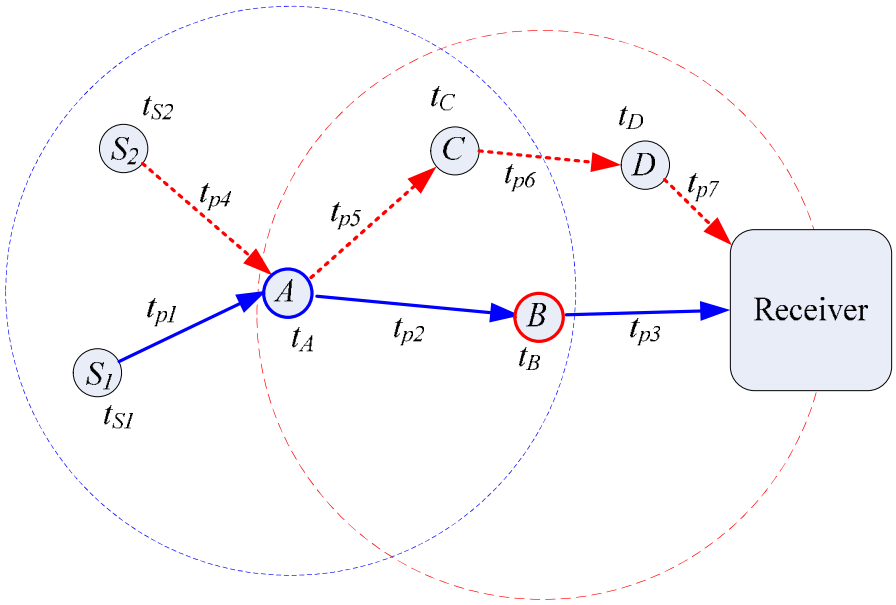


Fig. 5.3. Time synchronization module.

We illustrate our approach through an example. In Fig. 5.3, let t_X be the residence time (including queuing time, processing time, and transmitting time) at node X and let t_{pi} be the propagation delay for the hop i. Then, the residence time of the *message* from S_1 is given by: $T_{S1} = t_{S1} + t_A + t_B + t_{p1} + t_{p2} + t_{p3}$. Note that the propagation delay (of radio waves) incurred over several hundred meters (path distance to sink) is in the order of nanoseconds, we neglect this part. The time spent at a node is generally on the order of milliseconds and cannot be ignored. Under this assumption, T_{S1} can be calculated by summing up the times spent at each node. That is, $T_{S1} =$

$t_{S1} + t_A + t_B$. Similarly, $T_{S2} = t_{S2} + t_A + t_C + t_D$. As S_1's ***message*** reaches the receiver, the receiver notes the time (its own local time) at which it received this packet as τ_{S1}. Hence, the ***message*** must have been generated at $\tau_{S1} - T_{S1}$ (T_{S1} is obtained from the ***message***) in the local time of the receiver. The same procedure is applied for S_2. As we can see, the accumulative residence time can be used for estimating the occurrence time of received ***message***. So, we take such accumulative time as a timestamp stored in the ***message***.

This scheme eliminates many errors that time synchronization schemes have to contend with because we compute residence times close to the device. However, perhaps to a greater extent than those schemes, this scheme is impacted by clock drift. There are two problems brought about by clock drift: 1) timestamp can be significantly skewed if the residence times take so long; 2) clock drift can change the sample clocking, ***i.e.***, sensor samples may not be exactly 10ms apart when sampling at 100 Hz. The second problem might be considered unimportant, as sensor would be sampling the phenomenon correctly (when it happens), just not at the frequency it was supposed to. We return later to discuss the first problem in section 5.4.2.

5.2.4 Anonymity Module

Some laws and regulations are applicable for specifying how electronic patient record (EPR) should be handled, but they are subject to different interpretations. Patients must trust telemedicine systems to protect their private rights. However, EPRs are being used by different medical and administrative personnel; each has different professional and legal responsibilities. The system is risky with respect to privacy and confidentiality. This section is dedicated to implement privacy protection in the MSN, which targets ***Challenge* 3**.

Anonymous communication is an effective mechanism for protecting a user's privacy and also complies with the principle of least information [109]. It aims to hide communication relationship between two parties. Practical anonymity services such as Tor (The Onion Router) have been deployed to protect privacy and deterred censorship for many users. Wireless networks have posed additional challenges to anonymity, such as those stated in reference [109].

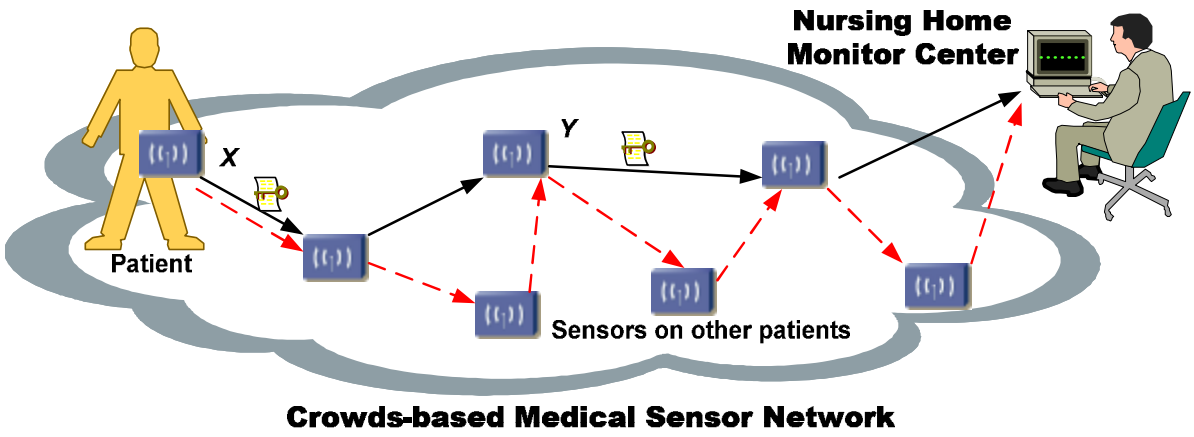


Fig. 5.4. Anonymity module.

There are three typical anonymous communication systems for reference: MIX, Onion Routing, and Crowds. For simplicity and energy efficient, we will adopt a Crowds system [108] in our design to enable anonymous communication. According to forwarding strategy of the Crowds system, *messages* are delivered in a dynamic way (shown in Fig. 5.4). More specifically, when a *message* arrives at an intermediate node between the sender and receiver, the node will replace the sender's address in the *message* with its own address. Similarly, the *message* will arrive at the destination after a series of forwarding procedures in such Crowds system. This strategy therefore guarantees to some extent that the sender's identity cannot be revealed.

5.2.5 System Framework

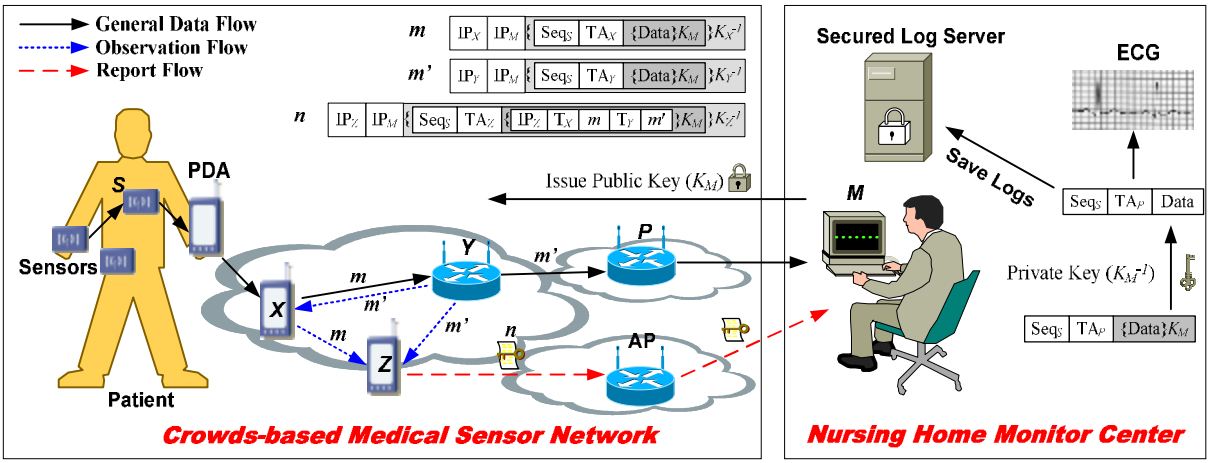


Fig. 5.5. Architecture of accountable MSN.

The system we propose here involves two major parts: Crowds-based MSN and the nursing home monitor center. It has combined three aforementioned modules to build our final system. As we discussed before, for the sake of building a low-cost and reliable MSN system, "receiver-only" local timestamp analysis technology is introduced for time synchronization. In general, Crowds-based MSN is responsible for collecting patients' information and forwarding them to the nursing home monitor center anonymously. The nursing home monitor center mainly processes the received data and stores relevant sensitive information in its secured log server. It is also in charge of public key management and time synchronization. In the following, we use examples to illustrate some implementation details.

1) Data Plane

As shown in Fig. 5.5, a *message* delivered in a MSN has 5 fields: 1) sender's address; 2) destination's address; 3) sequence number; 4) timestamp; and 5) encrypted data. The data contains sensor's ID and relevant medical information. The data is encrypted using a public key

issued by the monitor center, namely K_M. Hence, the data cannot be modified except by the monitor center. Note that every *message* is signed by sender's private key, which can be used for identifying sender's identification. For example, suppose a *principal X* delivers a *message m* to the monitor center, *m* will be formed as: $\{IP_X, IP_M, \{Seq_S, TA_X, \{Data\}K_M\}K_X^{-1}\}$. By using private key K_X^{-1}, no one can forge this *message* except *X*. Since K_X is public, every *principal* is able to verify this *message*. Due to Crowds-based forwarding strategy, the first field of the *message* will be replaced while passing through intermediate nodes (e.g., *principal Y* in Fig. 5.5). Specifically, *m* becomes $\{IP_Y, IP_M, \{Seq_S, TA_Y, \{Data\}K_M\}K_Y^{-1}\}$ after passing through *Y*. The fourth field also has been changed since the accumulative timestamp has been updated. In the same manner, this *message* (denoted as *m'*) is signed by K_Y^{-1} for identification reason. Eventually, this *message* will arrive at the monitor center.

At the monitor center, received *messages* will be extracted for further processing. Fields 3), 4), and 5) are used for medical analysis. For instance, they can be utilized to reconstruct an ECG. Field 5) will be decrypted using the corresponding private key K_M^{-1}. Field 4) will be used for estimating the occurrence time of this *message*. It can be easily obtained by subtracting the value of field 4) from local time of the monitor center. By using this computed value, together with field 3) and the sender's ID extracted from field 5), medical information is able to be restored. Finally, fields 3), 4), and 5) will be stored in a secured log server for future surveillance (e.g., auditing a certain *principal* for its temporal change during a long period).

2) General Flow

Our *message* transmission protocol has three phases: 1) delivery, 2) observation, and 3) report. In delivery phase, a sender delivers a *message* to the monitor center via several intermediate nodes. In observation phase, some neighbors will monitor forwarding *message* to

see if temporal signal has been modified. In report phase, the neighbor will deliver certain temporal evidence to the monitor center when an abnormal modification has been found.

We will explain the meaning of general flow step by step using examples. In the delivery phase, *X* first requests *Y* to forward a ***message m*** to the monitor center *M* (***Message*** 1). When *Y* received *m*, it will forward *m* to the next hop *P* (next hop decided by *Y*). Note that *m* has been changed to *m'* (***Message*** 2) after passing through *P*. Eventually, this ***message*** will be delivered to *M*. In the observation phase, when *X* delivers *m* to *Y*, *X*'s neighbor *Z* will also receive the ***message m*** (***Message*** 3). Then *Z* will record the ***Message*** 3 as {***Message*** 3, T_X}, which means *Z* has received the ***Message*** 3 at *Z*'s local time T_X. Similarly, when *Y* sends *m'* to *P*, *Y*'s neighbor *Z* will receive the *m'* (***Message*** 4) as well. Then *Z* will record the ***Message*** 4 as {***Message*** 4, T_Y}. Of course, *X* can also be an observation node of *Y*. Therefore, X will also receive the ***message m'*** (***Message*** 5) and record it as {***Message*** 5, T_Y'}. Note that, T_Y is the local time of *Z* while T_Y' is the local time of *X*. In the report phase, if *Z* has found that temporal signal of ***Message*** 4 is abnormal relative to ***Message*** 3, *Z* will send a report ***message n*** (***Message*** 6) to the monitor center through an alternative path. The ***message n*** will be formed as {IP_Z, IP_M, {Seq_S, TA_Z, {IP_Z, T_X, *m*, T_Y, *m'*}K_M}K_Z^{-1}}. As we can see, the report ***message n*** has the same structure as a regular ***message m***. The only difference is the data field of *m* has been changed to a temporal evidence {IP_Z, T_X, *m*, T_Y, *m'*} that only can be seen by the monitor center *M*.

5.3 Protocol Analysis

We have already addressed three ***Challenges*** in section 5.1. In the following two sections, we will try to analyze and evaluate our design in terms of temporal accountability, time accuracy,

and scalability. We adopt the same analysis method as Kudo's [101]. Firstly, present accountability goals for transmission protocol according to the definition stated in section 5.2.1.

G1: *M* CanProve (*X sees* $m(t_i)$ *at* t_{x1})

G2: *M* CanProve (*X modifies* $m(t_i)$ to $m(t_i')$)

G3: *M* CanProve (*X says* $m(t_i')$) *at* t_{x2})

Since an unsigned *message* has no effect on the achievement of goals in accountability logic, only the following flows will be interpreted:

***Message* 1**: *Y* Receives ({Seq_S, TA_X, {Data}K_M} SignedWith K_X^{-1})

***Message* 2**: *P* Receives ({Seq_S, TA_Y, {Data}K_M} SignedWith K_Y^{-1})

***Message* 3**: *Z* Receives ({Seq_S, TA_X, {Data}K_M} SignedWith K_X^{-1})

***Message* 4**: *Z* Receives ({Seq_S, TA_Y, {Data}K_M} SignedWith K_Y^{-1})

***Message* 5**: *X* Receives ({Seq_S, TA_Y, {Data}K_M} SignedWith K_Y^{-1})

***Message* 6**: *M* Receives ({Seq_S, TA_Z,

{IP_Z, T_X, *Message* 3, T_Y, *Message* 4}K_M} SignedWith K_Z^{-1})

The initial state assumptions required in the analysis are as follows:

A1: (*X says* ({Seq_S, TA_X, {Data}K_M}) *at* T_{XY})

=> (*X delivers* Data *at* T_{XY} TimestampWith TA_X)

A2: *M* CanProve (*X says* ({Seq_S, TA_X, {Data}K_M}) *at* T_X)

and (*Y says* ({Seq_S, TA_Y, {Data}K_M}) *at* T_Y)

=> (*M* CanProve (*Y modifies* $m(TA_X)$ to $m(TA_Y)$ *at* T_Y))

Using the above formal definitions, our protocol temporal accountability can be proved as follows:

***Message* 1**: When *Y* receives *Message* 1 at T_{XY}, *Y* knows it is sent by *X* based on IP_X field and *X*'s signature. By applying the accountability postulate [101, 102], we have: *Y* CanProve (*X says* ({Seq_S, TA_X, {Data}K_M}) *at* T_{XY}). This statement can be transformed by applying **A1**: *Y* CanProve (*X delivers* Data *at* T_{XY} TimestampWith TA_X). When *M* request the log file of *Y*, this statement can be used as a temporal evidence to prove (*X says m*(TA_X)) *at* T_{XY}). This is the accountability goal **G3**.

***Message* 2**: *Y* forwards the *Message* 1 to *P* as *X* has requested. This *message* will be eventually delivered to *M* through *P*. When *P* receives *Message* 2 at T_{YP}, by applying the accountability postulate and **A1**, we have: *P* CanProve (*Y delivers* Data *at* T_{YP} TimestampWith TA_Y). When *M* request the log file of *P*, this statement can be used as a temporal evidence to prove (*Y says m*(TA_Y)) *at* T_{YP}). This is the accountability goal **G3**.

***Message* 3**: TA field is required when the general Assumption 3 is made. When *X* broadcasts *Message* 1 with *X*'s signature, its neighbor *Z* instantly receives this *message* and records it as {*Message* 3, T_X}. T_X is the local time of *Z* when *Z* detects *Message* 1. Then, by applying the accountability postulate and **A1**, we have: *Z* CanProve (*X delivers* Data *at* T_X TimestampWith TA_X). When *M* request the log file of *Z*, this statement can be used as a temporal evidence to prove (*X* says *m*(TA_X)) *at* T_X) and (*Y sees m*(TA_X) *at* T_X). This is the accountability goal **G1** and **G3**.

***Message* 4**: It is similar with *Message* 3. By recording the *Message* 4 and applying the accountability postulate and **A1**, we have: *Z* CanProve (*Y delivers* Data *at* T_Y TimestampWith TA_Y). T_Y is the local time of *Z* when *Z* detects *Message* 2 broadcasted by *Y*. When *M* request the log file of *Z*, this statement can be used as a temporal evidence to prove (*Y* says *m*(TA_Y)) *at* T_Y) and (*P sees m*(TA_Y) *at* T_Y). This is the accountability goal **G1** and **G3**.

***Message* 5**: It is also similar with *Message* 3. By recording the *Message* 5 and by applying the accountability postulate and **A1**, we have: X CanProve (Y *delivers* Data *at* T_{YX} TimestampWith TA_Y). T_{YX} is the local time of X when X detects *Message* 2 broadcasted by Y. When M request the log file of X, this statement can be used as a temporal evidence to prove (Y says $m(TA_Y)$) *at* T_Y) and (P *sees* $m(TA_Y)$ *at* T_{YX}). This is the accountability goal **G1** and **G3**.

***Message* 6**: Through checking the difference between T_X and T_Y, together with the difference between TA_X and TA_Y, Z can easily verify whether *Message* 3 and *Message* 4 are satisfied with predefined temporal requirement (see section 5.2.2) or not. If they do not meet the requirement, Z will send a *Message* 6 to the monitor center. When *Message* 6 is received by M, M can request temporal evidences from relevant principals to prove the authenticity of Z. Therefore, by using the temporal evidence generated by Z, we have: M CanProve (X *says* ($\{Seq_S$, TA_X, $\{Data\}K_M\}$) *at* T_X) and (Y *says* ($\{Seq_S$, TA_Y, $\{Data\}K_M\}$) *at* T_Y). This statement can be transformed by applying **A2**: M CanProve (Y *modifies* $m(TA_X)$ to $m(TA_Y)$ at T_Y). This is the accountability goal **G2**.

5.4 Evaluation

In this section, we have evaluated our system for its performance in terms of temporal accountability, time accuracy, and scalability by using discrete event simulation method. Only wireless scenarios are modeled. No wired connection in our simulations. Suppose that no *message* will be delivered over five hops to the destination. We distribute all wireless nodes into five consecutive blocks. The destination is connected to the fifth block. In practice, there always have many possible intermediate nodes in the middle of a transmission path, but a few at the

beginning and the end. Therefore, we assign the proportion of the nodes in each block as 1:2:4:2:1 in quantity. For example, if there are ten nodes in a MSN, there will be one node in the first and fifth block respectively, two nodes in the second and fourth block respectively, and four nodes in the third block. Besides, considering more powerful wireless devices (like APs) almost close to the destination, the average service time for each block should be decreased along the transmission path. In a node, the service time refers to duration between start getting a *message* from a queue and complete sending this *message* to an uplink. Take the *message* authentication time into consideration, the average service times for block 1 to block 5 are assigned as: 25ms, 20ms, 15ms, 10ms, and 5ms respectively. For simplicity reasons, we do not consider the propagation time in wireless environment. So the propagation time is assumed to be zero. Moreover, the interval time of *message* arrival (it is generated by itself, not by receiving) at each node is exponentially distributed. The mean interval time for each block is different. We suppose the mean interval time for block 1 to block 5 are: 2s, 4s, 8s, 16s, and 32s respectively.

In order to simulate distinct local times for different wireless devices, we still utilize the time-driven simulation method [105] described in section 3.3.1. The drift clock for each device is subject to three factors: *offset*, *skew*, and *drift*. If current system time is *t*, the drift clock *D(t)* and the local time *L(t)* can be calculated by equation (3.6) and equation (3.7) respectively. As we can see, the three factors could be positive or negative. In our simulation, all *offset* values are uniformly distributed between -0.2 and 0.2; all *skew* values are uniformly distributed between -0.002 and 0.002; all *drift* values are uniformly distributed between -0.0002 and 0.0002.

For every situation, we run 100 times, 1000 seconds at a time, and take the average value of the outputs as our results. The simulation platform is Windows 7 64-bit, 2GB RAM, and Intel Core 2 6400, 2.13GHz CPU.

5.4.1 Temporal Accountability

We have already proved temporal accountability by using accountability logic in section 5.3. Note that the threshold direct affects the judgments of neighbor nodes in surveillance. A good threshold value is the key to make our system temporal accountable. In order to evaluate what is a good threshold, we set it in different values and measure the average accuracy of detection for abnormal temporal signals. We randomly set 10% of ***messages*** as abnormal by manually increasing their timestamp by one second. If there are x such ***messages*** in total and y ($0<y<x$) messages have been detected by surveillance, the accuracy of detection is y/x. For a threshold, we set the value between 1ms (it is mean value of ***skew***) and 100ms (it is less than the modified value 1s). Obviously, the threshold cannot be too high. Otherwise, some undesirable temporal signals with minor changes may be ignored by neighbors' surveillance. The threshold cannot be too small either, or the bias of local clock may be regarded as an abnormal behavior.

As Fig. 5.6 depicts, with the increase in value of threshold from 1ms to 100ms, the accuracy of detection gradually approaches to 100%. The fluctuation of each line can be explained by the effects of topology. Some nodes may only have two neighbors while others have five or more. Intuitively, when malicious nodes have more neighbors, they can be detected by more chances. So the accuracy of detection should be high. Otherwise, they may be conspiracy and cannot be detected by limited number of neighbors. Then the accuracy of detection should be low in this case.

From the analysis and simulation results we can tell that, the threshold value should be bigger than mean value of ***skew***, and less than abnormal modified value. At here, the mean value of ***skew*** is 1ms, and the modified value is 1s. Then the threshold can be select from 1ms to 1s.

But we need avoid the unnecessary fluctuation to obtain a better accuracy of detection. So the threshold is better to be the mean value of the above range. That is 500ms. Hence, a good threshold for a general purpose should be set as the mean value of a range from average *skew* (in microsecond) to a trivial abnormal modified value (in millisecond).

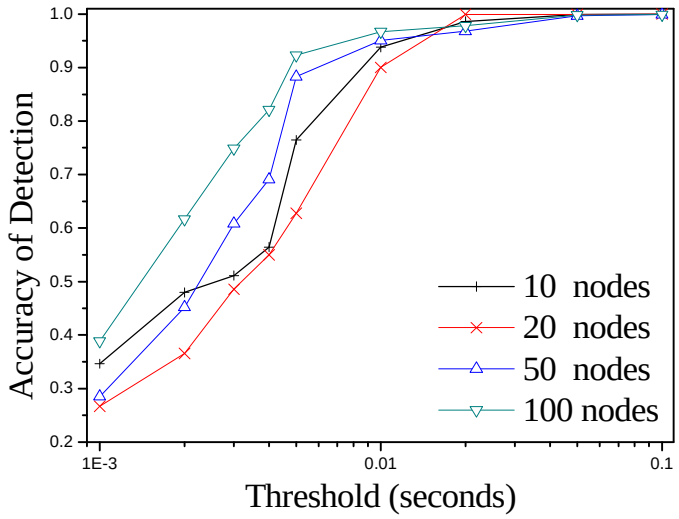


Fig. 5.6. Simulation results on threshold effect.

5.4.2 Time Accuracy

As Fig. 5.7 depicts, with the increase in number of hops, the average error is linear growth in milliseconds. The fluctuation of this line can be explained by the random distribution of local drift time. Since every hop increases the chance of time drifting to the system time, the average error increases along with the number of hops is a normal behavior. Fortunately, in a MSN environment, the number of hops will not reach up to 10 and all local drift time is too trivial to affect medical information. This result indicates that our protocol has average errors between 1.2ms and 2.3ms. It can be accepted in a MSN system.

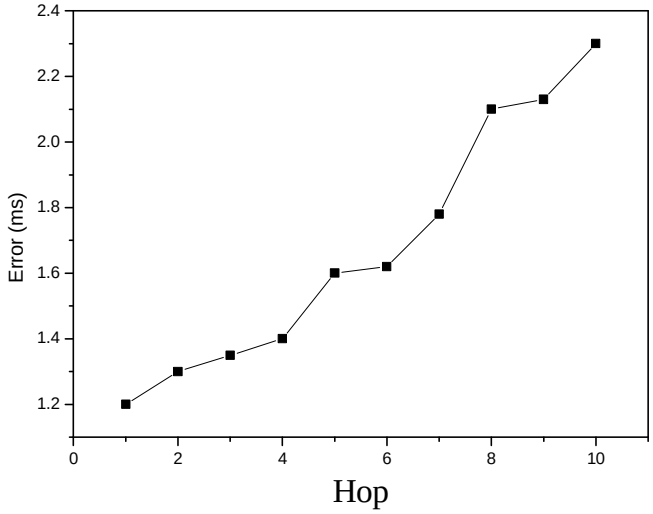


Fig. 5.7. Simulation results on hop effect.

5.4.3 Scalability

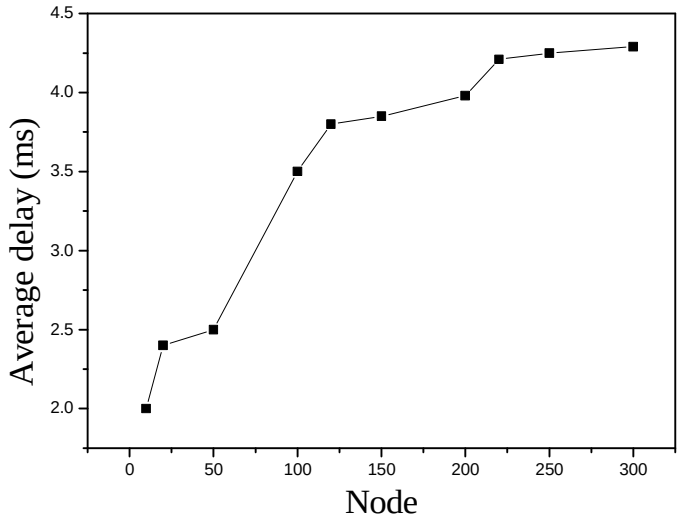


Fig. 5.8. Simulation results on scalability.

As Fig. 5.8 depicts, the average delay grows rapidly at beginning. This is because *messages* are too many to be queued in limited number of intermediate nodes. Later, with the

number of nodes increases, the growth of delay becomes slow. The reason is forwarding paths becomes more while sensors in block 1 remain the same sampling rates. The delay is less than 4.5 ms with 300 nodes. The result indicates our protocol is scalable in a typical MSN system.

5.5 Conclusion

This chapter mainly addressed temporal accountability and anonymity issues in MSN systems. By using mutual surveillance in heterogeneous wireless environment, the system we proposed is capable of detecting abnormal temporal signal and identifying the root cause of such event. By adopting the idea of Crowds anonymous communication system, our design can also well protect sender's identification on each transmitting *message*. Logical analysis and simulation results indicate that, if all wireless communication *principals* have a neighbor for surveillance, our MSN system can make majority of the temporal signal accountable. At the same time, all temporal signals can be synchronized with high precision (millisecond) at the monitor center. The "receiver-only" local timestamp analysis scheme incurs little overhead (a residence time field in every *message*) and can be implemented easily as we discussed. In addition, it is a scalable design that can be deployed into any other wireless communication system for temporal accountability objective.

CHAPTER 6

ENHANCED TEMPORAL ACCOUNTABLE MSN

In Chapter 5, we have presented a temporal accountable communication protocol in a medical sensor network (MSN). Each sensor node acts as both a sender and an observer. Since most wireless devices use broadcast mode to deliver ***messages***, every node within their communication range may capture the ***messages*** even it is not the recipient. Thus, the temporal signal can be exposed to all nodes that near the ***message***'s transmission path. With the help of observations from nearby nodes, almost all temporal signals are accountable and able to be detected if they have been modified by intermediate nodes. Besides, we also considered other two practical problems in the MSN. The first one focuses on time synchronization issue. In order to avoid the large shortening of medical sensor lifetime, a modified DMTS (Delay Measurement Time Synchronization) approach [107] has been adopted in our design. Another one aims to the privacy issue. To minimize the communication cost and obtain a certain degree of anonymity, we select "Crowds" out of three typical anonymous communication systems [108].

However, our previous work has several drawbacks that should be improved. Firstly, the broadcast scheme should be changed for many reasons, such as energy saving, network traffic reduces, conflict prevention, etc. Secondly, adopted public key infrastructure may be replaced by other light-weighted cryptography scheme due to complex re-key and management processes. To address these problems becomes our major thrust of this chapter. Three essential contributions are made: 1) a temporal accountable unicast mode is presented; 2) the communication between a

sensor and the monitor center is secured through a low-cost symmetrical key; 3) the medical data is authenticated through extremely light-weight security schemes.

The rest of this chapter is organized as follows: Section 6.1 discusses the problems; Section 6.2 proposes our improvements on the communication protocol; Section 6.3 mainly evaluates our design using logical testing and security analysis; Section 6.4 is the conclusion.

6.1 Problem Statement

The assumptions stated in section 5.2.1 obviously hinder the deployment of our proposed accountable MSN. On the one hand, the communication protocol relies on a broadcast scheme. That means, each sensor should have two antennas for full duplex communication. It not only costs more money, but also burdens the network bandwidth. More realistic issues present when *messages* are broadcasted in a wireless sensor network: 1) sensor requires much more energy to send, filter, and receive *messages*; 2) transmitting conflicts are highly possible when all sensor nodes broadcast *messages* in a limited area; and 3) communication *principals* in a wired network cannot be witnesses due to our accountable scheme.

On the other hand, sensor needs to encrypt every medical data for identity hidden purpose. The monitor center should issue and manage its public key in the MSN. Encryption itself will shorten the life-cycle of the sensor. The public key scheme makes it even worse. Current asymmetric key encryption algorithms all suffer two problems: 1) encryption and decryption processes are relatively slow compared to symmetric key based scheme; 2) rekey process is very complex and time consuming, especially for a large amount of objects.

Since these drawbacks limit the MSN's performance, addressing or improving these issues is therefore highly demanded. In this chapter, we will try to redesign the communication protocol in the MSN in order to overcome the aforementioned problems.

6.2 Protocol Design

6.2.1 Terms and Assumptions

In addition to the terms described in section 5.2.1, the following ones are required to better describe our design.

- $\{m\}H_X$: the *message* component *m* signed with a unique hash function of *X*.
- F_p: a finite field.
- E: an elliptic curve defined on F_p with a large order.
- G: the group of elliptic curve points on *E*.
- P_X: a point on *E* for *principal X*, $P_X \in G$.
- $h(x)$: a public one-way hash function, it maps *x* onto *G*.
- s: the monitor center's private key.
- ID_X: a pseudonym of *principal X*.

As our requirement changed, the assumptions are modified accordingly. This makes our design more realistic and robust than previous one. The new MSN platform is built upon following assumptions:

1. Monitor center is assumed to be trusted, and its clock is assumed to be accurate.

2. The local time of each *principal* except the monitor center is not trusted.
3. No *message* loss occurs during transmitting in either wireless or wired context.
4. Computing and storage space for the monitor center and access points are assumed to be unlimited. Sensors have enough storage space for logging purpose.

6.2.2 Temporal Accountability Module

In our original design (as described in chapter 5), each communication *principal* broadcasts its *message*s to their neighbors for a surveillance purpose. Once an abnormal time signal is detected, *correct* neighbors will report such event to the monitor center in a timely manner. Broadcast mode obviously brings many performance issues, especially in a wireless sensor network. Instead of broadcast mode, designing another effective approach to achieve the same temporal accountability goal will definitely enhance the performance of the MSN in terms of energy saving, network traffic reduces, conflict prevention, etc.

Actually, in most cases, only a few abnormal *principals* exist in the MSN. Sometimes, it is tolerable if we could find those abnormal ones at a later time. Therefore we could introduce assigned "witnesses" for each *principal* to avoid continues broadcasting. When a new communication *principal*, denoted as a node, joins the MSN, the monitor center will assign at least two active witnesses for this node after verifying its identity. The "active witness" means currently online and *correct* node. The new joined node and its witnesses, at the first time, will exchange their address information for later uses. In the case described in Fig. 6.1, *A* and *F* are *B*'s witnesses, and C's witnesses are *E* and *F*. Note that, it is not necessary for a witness to be in the communication range of an observed object. For instance, *D* also could be a witness of *B*

only if *B* could successfully deliver a *message* to *D* (e.g., via one hop through *F* or *C*). In the following, we will illustrate our enhanced temporal accountability module in details.

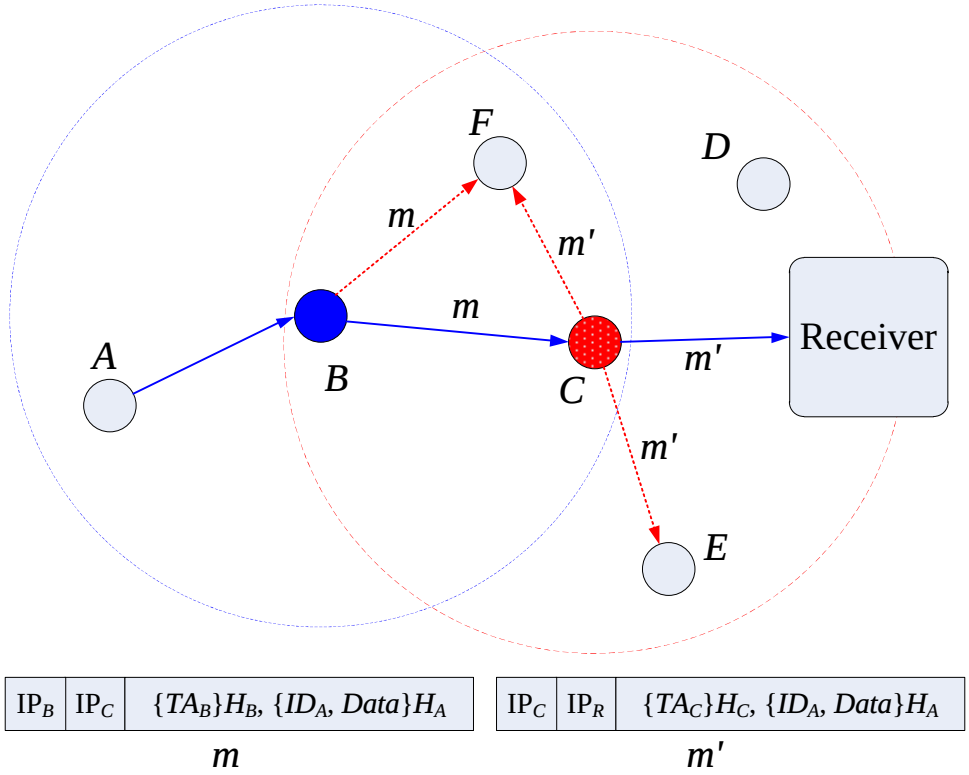


Fig. 6.1. Enhanced temporal accountability module.

In order to distinguish our previous design, we introduce a new term here. Let $\{m\}H_X$ be the *message* component *m* signed with an unique hash function of *X*. The hash function itself is public and will be issued by the monitor center through a secure channel based on a security policy. The unique means each *principal* holds a distinct symmetric key with the monitor center, which can be used for generating a unique hashing value on a same *message*. In the following discussion, we assume the monitor center would handle this key establishment job (discussed in section 6.2.3).

Fig. 6.1 illustrates an example that a *principal A* tries to deliver a *message* to a receiver *R* through *B* and *C*. Note that, *A*, *B*, and *C* could be any communication *principal* in the MSN, and *R* could be a PDA, an AP, or the monitor center *M*. When the *message* goes through *B*, *B* will replace the sender's IP with its own IP address IP_B for anonymity purpose. The *message* is changed to *m* as $\{IP_B, IP_C, \{TA_B\}H_B, \{ID_A, Data\}H_A\}$, which will be recorded in *B*'s local log files. The log file becomes $\{m, IP_A, t_B\}$, where IP_A is *m*'s previous hop IP, and t_B is *B*'s local time when *m* has been processed. In *message m*, TA_B is an accumulated timestamp used for time synchronization purpose (discussed in chapter 5), and ID_A is a pseudonym assigned by *M* for the original sender *A*. Similarly, *principal C* will modify and record the forwarding *message* to *m′* as $\{IP_C, IP_R, \{TA_C\}H_C, \{ID_A, Data\}H_A\}$. The log file of *m′* is $\{m', IP_B, t_C\}$. The *message* will be eventually delivered to the monitor center with a unicast mode.

At a predefined time interval, witnesses will challenge their observed objects for surveillance purpose. In our example case, *A* and *F* will ask *B* for recent processed log files, and *C* will be challenged by *E* and *F* for the same thing. When *E* and *F* receives *C*'s log file, they know *m′* is originally from *principal B* through IP_B. Therefore, *E* and *F* will ask *B* for recent log files regarding $\{ID_A, Data\}H_A$. Note that, this could disclose sender's identity (we will discuss a solution in section 6.2.4). Then *B* sends a copy of *m* to *E* and *F* within a given time period. If *B* fails to response (not because *B* is offline), *E* and *F* are reasonable to assume *B* is an abnormal *principal*. Otherwise, *E* and *F* will check whether the difference of TA_B and TA_C is equal to the difference of the two log times (aka. t_B and t_C). We denote $\{m, t_B\}$ is a temporal evidence of $\{m', t_C\}$. Once the difference is satisfied by our temporal requirement (discuss later), we say *m′* is equal to *m*. In other cases, we say *m'* is not equal to *m*, and *principal E and F* will report a suspicious temporal activity to the receiver with relevant temporal evidence. Since either *B* or *C*

may change their recorded *messages*, *F* cannot tell which one is abnormal solely based on their responses. Fortunately, the monitor center is able to recover the transmission path through back tracking on log files of each intermediate *principal*. Abnormal one can be found out by comparing all temporal records along the path.

Similarly, *principal A* will ask *B* for its log files for surveillance. Since both *A* and *B* knows *m* is from *A* through IP_A, *B*'s response *message* will not include $\{m, IP_A, t_B\}$.

The temporal requirement is not over a threshold value Δ predefined by the monitor center. During surveillance, *F* will calculate the difference of t_B and t_C, denoted as Δt. It also calculates the difference between two timestamps within *m* and *m'*, denoted as $\Delta t'$. When $|\Delta t - \Delta t'|$ is no greater than Δ, we say *m* and *m'* are satisfied by this predefined temporal requirement.

6.2.3 Key Establishment

As we known, key establishment and distribution are the fundamental tasks for entity authentication. We can use either symmetric key cryptography (SKC) or public key cryptography (PKC) for their implementations, but we have to know the pros and the cons of each algorithm. SKC-based schemes suffer the following problems: they require a large memory to store key materials, provide low scalability due to distribution of the keys, add and revoke keys, and require complicated key pre-distribution. On the other hand, PKC-based schemes suffer from high energy consumption and considerable time delay. PKC provides a more flexible and simple interface compared to SKC, which does not require key pre-distribution, pair-wise key sharing, or complicated one-way key chain schemes. In most cases, PKC-based schemes are used for digital signatures. However, it costs too much for sensor nodes on their energy and

computing resources as we discussed. For our situation, it is a wise choice if we adopt a PKC-based solution only for key distribution and at the meantime we also address the aforementioned constraint problems with SKC-based scheme. Based on current research achievements, we believe ECC (Elliptic Curve Cryptography)-based solution is a solid one to be considered for key distribution [114]. The reason we choose the ECC is that, its light-weight cryptographic scheme is better suit for the constrained MSN scenario.

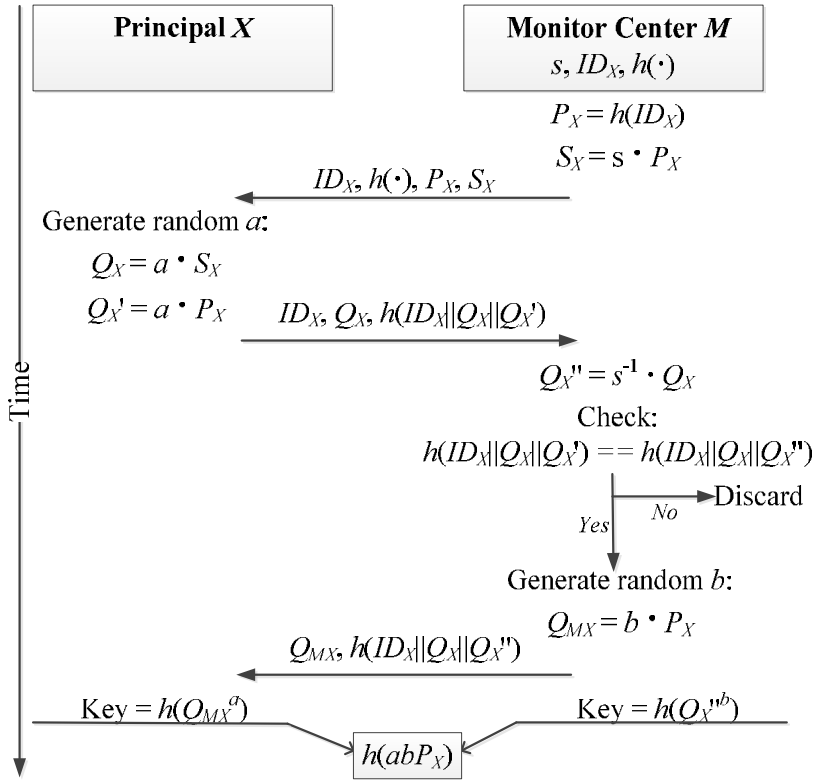


Fig. 6.2. Symmetric key establishment procedures.

In order to establish a symmetric key for a communication *principal* in the MSN, taking a monitor center *M* and a sensor node *X* as an example, only three essential steps are required as shown in Fig. 6.2.

Firstly, the monitor center will assign a pseudonym ID_X for each *principal X*. Then *M* computes a corresponding point P_X on *G* with a public one way hash function $h(ID_X)$. *M* will also generate the private key of *X* as $S_X = s \cdot P_X$ over F_p. Note that, *s* is a secret key of *M* that is assumed to be assigned before the *M* has joined the MSN. Later, *M* will send the first *message* to *X* as $\{ID_X, h(\cdot), P_X, S_X\}$.

Secondly, the *principal X* generates an ephemeral private key *a* and computes $Q_X = a \cdot S_X$ and $Q_X' = a \cdot P_X$ over F_p. Then *X* will send an authentication *message* $\{ID_X, Q_X, h(ID_X||Q_X||Q_X')\}$ to *M*. Once receive the *message*, *M* will compute $Q_X'' = s^{-1} \cdot Q_X$ over F_p and $h(ID_X||Q_X||Q_X'')$ respectively. Then it checks whether $h(ID_X||Q_X||Q_X'')$ is equal to $h(ID_X||Q_X||Q_X')$ or not. If not, authentication fails. Otherwise go to step 3.

The third step is the establishment for the symmetric key of *X* and *M*. Similarly, *M* will choose a random ephemeral key *b* and compute $Q_{MX} = b \cdot P_X$ for the *M-X* pair. The symmetric key will be $h(abP_X)$ based on ECC algorithm. It can be computed by $h(Q_X''^b) = h(abP_X)$. The *principal X* will get the symmetric key from *M*'s response *message*, $\{Q_{MX}, h(ID_X||Q_X||Q_X'')\}$, by computing $h(Q_{MX}^a) = h(abP_X)$.

Once the symmetric key is established, *principal X* is able to communicate with the monitor center directly using $h(abP_X)$. The symmetric key can be further used for rekey purpose, which will save the time and computing energy for authentication process.

6.2.4 Anonymity Module

The original design uses Crowds anonymous communication system to hide sender's identity in a MSN. Each intermediate node in a transmitting path will replace sender's IP field

with its own address. The forwarding path is arbitrary according to Crowds' routing strategy. Although the communication channel is unsecured, and all traffic in a MSN can be observed by any *principal*. No one is able to infer the sender's identification only by an observed *message*, since the sender's IP address may have been replaced many times already. In enhanced communication protocol, such Crowds-based scheme is still adopted. The modification part goes to data field. Previous design encrypts the data field with monitor center's public key. It prevents data's identification information from being disclosed. In fact, only the identification should be hidden. Encrypt other parts of data is not necessary. Therefore, we use a pseudonym to anonymize sender's identity. In this case, sensor is not required to encrypt and decrypt *messages*, which enhances its life-cycle and saves processing time.

As we mentioned in section 6.2.2, witness requests log files from non-observed objects may disclose sender's identity. A malicious one may ask all *principals* to send certain log files for surveillance. With those files, it is easy to find a transmitting path for particular *message*. However, we could prevent this attack by simply verify the challenge requests. Only the next hop's witness and the monitor center are acceptable to get those requests' responses. The verification can be achieved by witness's certificate (along with the challenge request) from the monitor center. Hence, malicious one is unable to get the whole picture of the MSN traffic. In the meantime, the monitor center may prevent the anonymous communication from being abused.

6.3 Analysis

In the following two subsections, we try to analyze our design in terms of temporal accountability and attacks protection. As we can see in the assumptions, the new protocol

eliminates the public key infrastructure and allows the wired scenario in the MSN. This modification makes our new communication protocol more scalable and realistic than previous design.

6.3.1 Protocol Analysis

We adopt the same analysis method as we did in chapter 5. Firstly, present accountability goals for transmission protocol according to the definition stated in section 5.2.1.

G1: *M* CanProve (*X sees* $m(t_i)$ *at* t_x)

G2: *M* CanProve (*X modifies* $m(t_i)$ to $m(t_i')$)

G3: *M* CanProve (*X says* $m(t_i')$) *at* t_x)

Since an unsigned *message* has no effect on the achievement of goals in accountability logic, only the following flows will be interpreted (in a scenario that described in Fig. 6.2):

***Message* 1**: *C* Receives ({TA_B} SignedWith H_B, {ID_A, *Data*} SignedWith H_A})

***Message* 2**: *R* Receives ({TA_C} SignedWith H_C, {ID_A, *Data*} SignedWith H_A)

***Message* 3**: *F* Receives ({*Message* 2, IP_B, t_C} SignedWith H_C)

***Message* 4**: *F* Receives ({*Message* 1, IP_A, t_B} SignedWith H_B)

***Message* 5**: *M* Receives ({IP_F, *Message* 3, *Message* 4} SignedWith H_F)

The initial state assumptions required in the analysis are as follows:

A1: (*X says* ({{TA_X}H_X, {ID_Y, *Data*}H_Y }) *at* t_X)

=> (*X delivers Y*'s *Data at* t_X TimestampWith TA_X)

A2: (*X says* {$m(TA_X)$, IP_Y, t_X})

=> (*X sees m* before t_X) and (*X modifies m* to $m(TA_X)$ *at* t_X) and (*X says* $m(TA_X)$ *at* t_X)

Using the above formal definitions, our protocol temporal accountability can be proved as follows:

***Message* 1**: When *C* receives *Message* 1 at t_B', *C* knows it is sent by *B* based on IP_B field. Note t_B' is the local time of *C* when *B*'s local time is t_B. By applying the accountability postulate [101], we have: *C* CanProve (*B says* $(\{TA_B\}H_B, \{ID_A, Data\}H_A)$ *at* t_B'). This statement can be transformed by applying **A1**: *C* CanProve (*B delivers A*'s *Data at* t_B' TimestampWith TA_B). When *M* request the log file of *C*, this statement can be used as a temporal evidence to prove (*B says* $m(TA_B)$) *at* t_B'). This is the accountability goal **G3**.

***Message* 2**: *C* forwards the *Message* 1 to *R*. This *message* will be eventually delivered to *M* through *R*. When *R* receives *Message* 2 at t_C', by applying the accountability postulate and **A1**, we have: *R* CanProve (*C delivers A*'s Data *at* t_C' TimestampWith TA_C). Note t_C' is the local time of *R* when *C*'s local time is t_C. When *M* request the log file of *R*, this statement can be used as a temporal evidence to prove (*C says* $m(TA_C)$) *at* t_C'). This is the accountability goal **G3**.

***Message* 3**: It is requested when *F* challenges *C* for its recent log files. In this case, *C* will record *Message* 2 when it has been processed at t_C. Then, by applying the accountability postulate and **A2**, we have: *F* CanProve (*C sees m* before t_C) and (*C modifies m* to $m(TA_C)$ at t_C) and (*C says* $m(TA_C)$ *at* t_C). When *M* request the log file of *F*, this statement can be used as a temporal evidence to support the accountability goal **G1**, **G2** and **G3**.

***Message* 4**: It is similar with *Message* 3. By applying the accountability postulate and **A2**, we have: *F* CanProve (*B sees m* before t_B) and (*B modifies m* to $m(TA_B)$ at t_B) and (*B says* $m(TA_B)$ *at* t_B). When *M* request the log file of *F*, this statement can be used as a temporal evidence to support the accountability goal **G1**, **G2** and **G3**.

Message* 5**: Based on *Messages* 3 and 4, through checking the difference between t_B and t_C, together with the difference between TA_B and TA_C, *F* can easily verify whether *Message* 3 and *Message* 4 are satisfied with predefined temporal requirement (see section 6.2.2) or not. If they do not meet the requirement, *F* will send a *Message* 6 to the monitor center. When *Message* 6 is received by *M*, *M* can request temporal evidences from relevant ***principals to prove the correctness of *F*. Hence, by using the temporal evidence generated by *F*, we have: *M* CanProve (*B says* (*Message* 1 *at* t_B) and (*C says* (*Message* 2 *at* t_C). This statement can be transformed as: *M* CanProve (*C modifies* $m(TA_B)$ to $m(TA_C)$ *at* t_C). This is the accountability goal **G2**.

6.3.2 Security Analysis

In this section, we will analyze whether our proposed key establishment protocol is secure or not.

1) Eavesdropping Attack

Each run produces a different symmetric key, and knowledge of past symmetric keys does not allow deduction of future symmetric keys. In our scheme, the symmetric key is calculated by one way hash and random secrets. Only the sensor *X* and monitor center know their symmetric key $h(abP_X)$, which is computed from the random ephemeral key. That is, even if the previous secrets are revealed, the other secrets will remain unknown to the adversary.

2) Man-in-the-middle Attack

Compromising of a long term secret key at some point in the future, does not lead to compromise of communications in the past. Note that in our scheme, even if the adversary compromises the monitor center's secret key *s*, it cannot compromise the previous symmetric key

because the adversary cannot know the ephemeral key a or b such that it cannot compute the symmetric key. Also, our protocols satisfy both partial forward secrecy and perfect forward secrecy since it is hard to compute the symmetric key without knowing the ephemeral key a or b. Hence, no one in the middle can obtain the symmetric key. However, it still requires additional strategy to prevent IP spoofing. Access control and user authentication is required before initiating the key establishment.

3) Key Control Attack

Both communication entities select a random number to generate the symmetric key, which would be discarded after a given time, e.g., the session expired. Neither one can control the outcome of the key by, for example, restricting it to lie in some predetermined small set. In other words, neither entity can force the symmetric key to a pre-selected value. Hence, our proposed protocol can resist any key control attack.

4) Replay Attack

In case a malicious one gained a valid symmetric key or captured network traffic in the MSN, the protocol should resist replay attack by introducing a nonce in every transmitted *message*. However, it is an optional choice that could vary on different applications. Besides, the symmetric key could be used for identification. Therefore, replayed *message* from unidentified person will be discarded.

6.4 Conclusion

This chapter mainly addressed temporal accountability issues in a medical sensor network. It extends and enhances our previous work in Chapter 5, in terms of traffic overhead

and key management. By using a light-weight symmetric key authentication method, sensor does not need encrypt any *message* and thus can save energy and computing resources. An elliptic curve cryptography based symmetric key establishment procedure ensures each *message* is securely authenticated. It also facilitates key management for the monitor center. In our enhanced communication protocol, unicast mode is adopted for *message* transmission, which reduces traffic overhead and transmitting conflict possibility. Assigning witness for each communication principal makes wired network temporal accountable as well. Compared with previous design, the drawback is we cannot detect the abnormal *principal* in a timely manner. A *correct* witness only can point out that a problem exists in two suspicious *principals*. Further inspection should be involved through the monitor center.

CHAPTER 7

CONCLUSION

People never stop questioning security issues of a system since it came out in the first place. Albeit a well design could eliminate most threats, vulnerabilities still emerge after new technologies are adopted. Instead of fixing endless security problems, identifying and tracing back misbehavior entities are alternative ways to secure a system. This technology is so called accountability. Two case studies are presented as follows:

One specific problem has been raised in the power metering system. As we know, the power utility company charges customers solely based on the readings from their power meters. Considering the operating cost and technical difficulty, the utility only measures the aggregated power supply to a service area. In order to get individual power consumption, in the past, utility would send technicians to manually gather meter readings. To date, by using automatic meter reading technology, this information can be remotely obtained via a private corporate network or the public Internet. However, if a meter is compromised or malfunction due to some other reasons, the utility can hardly find it and thus may have economic loss. A possible solution is to prevent the meter from being altered. But this is not enough for more complex networks of the smart grid (*aka.* current power grid integrated with communication and information technologies). In a smart grid, the meter not only measures the incoming power flow, but also can be responsible for calculating the self/home-generated power. The readings for home consumption therefore could be obscured by these two power resources, which make collusion

attack possible. For example, after hacking the meter one could argue that, he never consumes any power from the utility but only use the electricity generated by home solar system. In addition, the digital readings could be accessed online. Theoretically, they could be hacked without the hardware modification as well. Furthermore, prices change along with time in smart grids, traditional billing method therefore is no longer feasible. The exact times when power is used in a single day are important and should be made accountable. To solve the above problems and to make the smart grid reliable, we designed two accountable metering systems by using a peer review strategy and intersected grouping scheme respectively. Through a logic analysis and simulation, we argue that the proposed systems can effectively detect any faulty meter in a home area and neighborhood area networks under some reasonable assumptions.

Another case is the medical sensor network (MSN). In this context, patients are deployed with certain medical sensors and wearable devices and are remotely monitored by professionals. Thus, seeing a doctor in person is no longer the only option for those in need of medical care. Since it is also an economical way to reduce healthcare costs and save medical resources, we expect a robust, reliable, and scalable MSN in the near future. However, the time signal and temporal history in the current MSN are vulnerable due to unsecured infrastructure and transmission strategies. Meanwhile, the MSN may leak patients' identifications or other sensitive information that violates personal privacy. To make sure the critical time signal is accountable, we propose two new communication protocols for MSN that is capable of temporal accountability. In addition, they also provide privacy-preserving ability via a Crowds anonymous system. The analysis and simulation results clearly indicate the advantages of our first design in terms of reliable and scalable features. The second design is the extension of the first one. It enhances the MSN performance in terms of traffic overhead and key management.

REFERENCES

[1] Cisco Systems, Inc., "Internet protocol architecture for the smart grid," ***White Paper***, Jul. 2009, available at: http://www.cisco.com/web/strategy/docs/energy/CISCO_IP_INTEROP_STDS_PPR_TO_NIST_WP.pdf.

[2] U.S. DOE, "Smart grid system report," ***White Paper***, Jul. 2009, available at: http://www.oe.energy.gov/SGSRMain_090707_lowres.pdf.

[3] U.S. NIST, "Guidelines for smart grid cyber security (vol. 1 to 3)," ***NIST IR-7628***, Aug. 2010, available at: http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7628.

[4] U.S. NIST, "NIST framework and roadmap for smart grid interoperability standards, release 1.0," ***NIST Special Publication 1108***, Jan. 2010, available at: http://www.smartgrid.gov/standards/roadmap.

[5] U.S. NETL, "Advanced metering infrastructure," ***White Paper***, Feb. 2008, available at: http://www.smartgrid.gov/white_papers.

[6] U.S. NETL, "A systems view of the modern grid," ***White Paper***, Jan. 2007, available at: http://www.smartgrid.gov/white_papers.

[7] A. Clark and C.J. Pavlovski, "Wireless networks for the smart energy grid: application aware networks," in: ***Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 Vol II (IMECS 2010)***, Hong Kong, Mar. 2010.

[8] J. Gadze, "Control-aware wireless sensor network platform for the smart electric grid," ***IJCSNS International Journal of Computer Science and Network Security***, vol. 9, no. 1, Jan. 2009, pp. 16-26.

[9] D. Dvian and H. Johal, "A smart grid for improving system reliability and asset utilization," ***CES/IEEE 5th International Power Electronics and Motion Control Conference***, Shanghai, China, Aug. 2006, pp. 1-7.

[10] G.N. Srinivasa Prasanna, A. Lakshmi, S. Sumanth, V. Simha, J. Bapat, and G. Koomullil, "Data communication over the smart grid," in: ***IEEE International Symposium on Power Line Communications and Its Applications (ISPLC 2009)***, Dresden, 2009, pp. 273-279.

[11] H. A. Khan, Z. Xu, H. Iu, and V. Sreeram, "Review of technologies and implementation strategies in the area of smart grid," in: *The 10th Postgraduate Electrical Engineering and Computing Symposium*, IEEE WA Section, Perth, Australia, Oct. 2009.

[12] A. Cavoukian, J. Polonetsky, and C. Wolf, "SmartPrivacy for the smart grid: embedding privacy into the design of electricity conservation," *Identity in the Information Society*, Springer Netherlands, ISSN: 1876-0678, Apr. 2010.

[13] S. Spoonamore and R.L. Krutz, "Smart grid and cyber challenges – national security risks and concerns," March 2009, available online: http://www.whitehouse.gov/files/ documents/cyber/Spoonamore-Krutz- Smart Grid Cyber Security Risks and Concerns.pdf.

[14] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," IEEE *Security and Privacy*, vol. 7, no. 3, May/Jun. 2009, pp. 75-77.

[15] Idaho National Laboratory, "Common cyber security vulnerabilities observed in control system assessments by the INL NSTB program," *Idaho National Laboratory Technical Report (INL/EXT-08-13979)*, 2008, available at: http://www.inl.gov/scada/publications.

[16] C. Wei, "A conceptual framework for smart grid," in: *Power and Energy Engineering Conference (APPEEC 2010)*, Chengdu, China, Mar. 2010, pp. 1-4.

[17] A.R. Metke and R.L. Ekl, "Smart grid security technology," in: *Innovative Smart Grid Technologies (ISGT 2010)*, Gaihersburg, MD, Jan. 2010, pp. 1-7.

[18] W.Y. Chu and Dennis J.H. Lin, "Communication strategies in enabling smart grid development," in: *The 8th International Conference on Advances in Power System Control, Operation and Management (APSCOM 2009)*, Hong Kong, China, 2009, pp. 1-6.

[19] W. Shireen and S. Patel, "Plug-in hybrid electric vehicles in the smart grid environment," in: *2010 IEEE PES Transmission and Distribution Conference and Exposition*, New Orleans, LA, Apr. 2010, pp. 1-4.

[20] K. Moslehi and R. Kumar, "A reliability perspective of the smart grid," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, Jun. 2010, pp. 57-64.

[21] W.F. Boyer and S.A. McBride, "Study of security attributes of smart grid systems – current cyber security issues," *Idaho National Laboratory Technical Report (INL/EXT-09-15500)*, Apr. 2009, available at: http://www.inl.gov/scada/publications.

[22] G.N. Ericsson, "Cyber security and power system communication – essential parts of a smart grid infrastructure," *IEEE Transactions on Power Delivery*, vol. 25, no. 3, Jul. 2010, pp. 1501-1507.

[23] Z. Fan, G. Kalogridis, C. Efthymiou, M. Sooriyabandara, M. Serizawa, and J. McGeehan, "The new frontier of communications research: smart grid and smart metering," in:

Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, Passau, Germany, Apr. 2010, pp. 115-118.

[24] H. Cheung, A. Hamlyn, T. Mander, C. Yang, and R. Cheung, ""Strategy and role-based model of security access control for smart grids computer networks," *2007 IEEE Canada Electrical Power Conference (EPC 2007)*, Montreal, Canada, Oct. 2007, pp. 423-428.

[25] P. Vytelingum, S.D. Ramchurn, T.D. Voice, A. Rogers, and N.R. Jennings, "Trading agents for the smart electricity grid," in: *The Ninth International Conference on Autonomous Agents and Multiagent Systems*, Toronto, Canada, May 2010, pp. 897-904.

[26] A. Hamlyn, H. Cheung, T. Mander, L. Wang, C. Yang, and R. Cheung, "Computer network security management and authentication of smart grids operations," *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, Pittsburgh, PA, Jul. 2008, pp.1-7.

[27] A.R. Metke and R.L. Ekl, "Security technology for smart grid networks," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, Jun. 2010, pp. 99-107.

[28] Z. Sun, S. Huo, Y. Ma, and F. Sun, "Security mechanism for smart distribution grid using ethernet passive optical network," in: *The 2nd International Conference on Advanced Computer Control (ICACC 2010)*, vol. 3, Shenyang, China, Mar. 2010, pp. 246-250.

[29] S. Fries, H.J. Hof, and M. Seewald, "Enhancing IEC 62351 to improve security for energy automation in smart grid environments," in: *The 5th International Conference on Internet and Web Applications and Services*, Barcelona, Spain, May 2010, pp. 135-142.

[30] D. Wei, Y. Lu, M. Jafari, P. Skare, and K. Rohde, "An integrated security system of protecting smart grid against cyber attacks," in: *Innovative Smart Grid Technologies (ISGT 2010)*, Gaithersburg, MD, Jan. 2010, pp. 1-7.

[31] P. Zhang, O. Elkeelany, L. McDaniel, "An implementation of secured smart grid ethernet communications using AES," in: *Proceedings of the IEEE SoutheastCon (SoutheastCon 2010)*, Concord, NC, Mar. 2010, pp. 394-397.

[32] V. Li, F.F. Wu, and J. Zhong, "Communication requirements for risk-limiting dispatch in smart grid," in: *IEEE International Conference on Communications Workshops (ICC 2010)*, Capetown, May 2010, pp. 1-5.

[33] Q. Pang, H. Gao, and M. Xiang, "Multi-agent based fault location algorithm for smart distribution grid," in: *The 10th IET International Conference on Development in Power System Protection (DPSP 2010)*, Manchester, Apr. 2010, pp. 1-5.

[34] Y. Cai and M. Chow, "Exploratory analysis of massive data for distribution fault diagnosis in smart grids," in: *IEEE Power & Energy Society General Meeting (PES '09)*, Calgary, AB, Jul. 2009, pp. 1-6.

[35] M. Kezunovic, "Automated fault analysis in a smart grid," in: *IEEE Asia and Pacific Transmission & Distribution Conference & Exposition*, Seoul, Oct. 2009, pp. 1-3.

[36] C. Bennett and S.B. Wicker, "Decreased time delay and security enhancement recommendations for AMI smart meter networks," in: *Innovative Smart Grid Technologies (ISGT 2010)*, Gaithersburg, MD, Jan. 2010, pp. 1-6.

[37] National Communications System, "Supervisory control and data acquisition (SCADA) systems," *Technical Report*, Oct. 2004, available at: http://www.ncs.gov/library /tech_bulletins/2004/tib_04-1.pdf.

[38] S. Ward, J. O'Brien, B. Beresh, G. Benmouyal, D. Holstein, J.T. Tengdin, K. Fodero, M. Simon, M. Carden, M.V.V.S. Yalla, T. Tibbals, V. Skendzic, S. Mix, R. Young, T. Sidhu, S. Klein, J. Weiss, A. Apostolov, D.-P, Bui, S. Sciacca, C. Preuss, S. Hodder, and G. Seifert, "Cyber security issues for protective relays," in: *IEEE Power Enginnering Society General Meeting*, Tampa, FL, Jun. 2007, pp. 1-27.

[39] S. Hong and M. Lee, "Challenges and direction toward secure communication in the SCADA system," in: *Proceedings of the 8th Annual Communication Networks and Services Research Conference*, Montreal, Canada, May 2010, pp. 381-386.

[40] F. Alsiherov and T. Kim, "Secure SCADA network technology and methods," in: *Proceedings of the 12th WSEAS International Conference on Automatic Control, Modelling & Simulation*, Catania, Italy, May 2010, pp. 434-438.

[41] IEC TC57, "Power system control & associated communications – data & communication security," IEC 62351 Part 1 to 8, Technical Specification and Draft, 2010.

[42] A. Faruqui, R. Hledik, and S. Sergici, "Piloting the smart grid," *The Electricity Journal*, vol. 22, issue 7, 2009, pp. 55-69.

[43] H.K.-H. So, S.H.M. Kwok, E.Y. Lam, and King-Shan Lui, "Zero-configuration identity-based signcryption scheme for smart grid," in: *The First IEEE Internation Conference on Smart Grid Communications (SmartGridComm)*, Gaithersburg, MD, 2010, pp. 321-326.

[44] G. Kalogridis, C. Efthymiou, S.Z. Denic, T.A. Lewis, and R. Cepeda, "Privacy for smart meters: towards undetectable appliance load signatures," in: *The First IEEE Internation Conference on Smart Grid Communications (SmartGridComm)*, Gaithersburg, MD, Oct. 2010, pp. 232-237.

[45] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in: *The First IEEE Internation Conference on Smart Grid Communications (SmartGridComm)*, Gaithersburg, MD, Oct. 2010, pp. 238-243.

[46] NaturalGas.org, "Natural gas and the environment," 2010, available at: http://www.naturalgas.org/environment/naturalgas.asp.

[47] A. Creery and E.J. Byres, "Industrial cybersecurity for power system and SCADA networks," in: *Industry Applications Society 52nd Annual Petroleum and Chemical Industry Conference*, Denver, Colorado, Sept. 2005, pp. 303-309.

[48] North American Electric Reliability Council, "SQL slammer worm lessons learned for consideration by the electricity sector," Jun. 2003, available at: http://www.esisac.com/publicdocs/SQL_Slammer_2003.pdf.

[49] J. Liu, Y. Xiao, and J. Gao, "Accountability in smart grids," in: *IEEE Consumer Communications and Networking Conference 2011 (IEEE CCNC 2011), Smart Grid Special Session*, Las Vegas, Nevada, Jan. 2011.

[50] ModBus, "Modbus specifications and implementation guides," *ModBus Protocol Specification v1.1b*.

[51] ProfiBus, "Profibus standard," *Profibus Specifications & Standards*, available at: http://www.profibus.com/downloads/specifications-standards/.

[52] T. M. Overman and R. W. Sackman, "High assurance smart grid: smart grid control systems communications architecture," in: *Proceedings of the 1st IEEE SmartGridComm 2010*, Gaithersburg, MD, Oct. 2010, pp. 19-24.

[53] R. Anderson and S. Fuloria, "Who controls the off switch?" in: *Proceedings of the 1st IEEE SmartGridComm 2010*, Gaithersburg, MD, Oct. 2010, pp. 96-101.

[54] Y. Liu, P. Ning, and M. Reiter, "False data injection attacks against state estimation in electric power grids," in: *Proceedings of the 16th ACM conference on Computer and Communications Security*, Chicago, Illinois, 2009, pp. 21–32.

[55] R. B. Bobba, K. M. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. J. Overbye, "Detecting false data injection attacks on dc state estimation," in: *Preprints of the First Workshop on Secure Control Systems*, CPSWEEK 2010, 2010.

[56] H. Sandberg, A. Teixeira, and K. H. Johansson, "On security indices for state estimators in power networks," in: *Preprints of the First Workshop on Secure Control Systems*, CPSWEEK 2010, 2010.

[57] G. Dan and H. Sandberg, "Stealth attacks and protection schemes for state estimators in power systems," in: *Proceedings of the 1st IEEE SmartGridComm 2010*, Gaithersburg, MD, Oct. 2010, pp. 214-219.

[58] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on smart grid state estimation: attack strategies and countermeasures," in: *Proceedings of the 1st IEEE SmartGridComm 2010*, Gaithersburg, MD, Oct. 2010, pp. 220-225.

[59] L. Xie, Y. Mo, and B. Sinopoli, "False data injection attacks in electricity markets," in: *Proc of the 1st IEEE SmartGridComm 2010*, Gaithersburg, MD, 2010, pp. 226-231.

[60] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in: *Proceedings of the 1st IEEE SmartGridComm 2010*, Gaithersburg, MD, Oct. 2010, pp. 327-332.

[61] A. Bartoli, J. Hernandez-Serrano, M. Soriano, M. Dohler, A. Kountouris, and D. Barthel, "Secure lossless aggregation for smart grid M2M networks," in: *Proceedings of the 1st IEEE SmartGridComm 2010*, Gaithersburg, MD, Oct. 2010, pp. 333-338.

[62] D. P. Varodayan and G. X. Gao, "Redundant metering for integrity with information-theoretic confidentiality," in: *Proceedings of the 1st IEEE SmartGridComm 2010*, Gaithersburg, MD, Oct. 2010, pp. 345-349.

[63] R. Berthier, W. H. Sanders, and H. Khurana, "Intrusion detection for advanced metering infrastructures: requirements and architectural directions," in: *Proceedings of the 1st IEEE SmartGridComm 2010*, Gaithersburg, MD, Oct. 2010, pp. 350-355.

[64] D. Kundur, X. Feng, S. Mashayekh, S. Liu, T. Zourntos, and K.L. Butler-Purry, "Towards modeling the impact of cyber attack on a smart grid," *International Journal of Security and Networks (IJSN), special issue on security and privacy in smart grids*, Vo. 6, No. 1, 2011.

[65] G. Kalogridis, S. Z. Denic, T. Lewis, and R. Cepeda, "Privacy protection system and metrics for hiding electrical events," *International Journal of Security and Networks (IJSN), special issue on security and privacy in smart grids*, Vo. 6, No. 1, 2011.

[66] F. Li, B. Luo, and P. Liu, "Secure and privacy-preserving information aggregation for smart grids," *International Journal of Security and Networks (IJSN), special issue on security and privacy in smart grids*, Vo. 6, No. 1, 2011.

[67] J. Zhang and C. A. Gunter, "Application-aware secure multicast for power grid communications," *International Journal of Security and Networks (IJSN), special issue on security and privacy in smart grids*, Vo. 6, No. 1, 2011.

[68] Y. Xiao, D. Takahashi, J. Liu, and H. Deng, and J. Zhang, "Wireless telemedicine and M-Health: technologies, applications, and research issues," *International Journal of Sensor Networks (IJSNet*), vol. 10, no. 4, 2011, pp. 202-236.

[69] F. Hu, L. Celentano, and Y. Xiao, "Error-resistant RFID-assisted wireless sensor networks for cardiac telehealthcare," *Wireless Communications and Mobile Computing*, vol. 9, issue 1, Feb. 2008, pp. 85-101.

[70] K. J. Liszka, D. W. York, M. A. Mackin, and M. J. Lichter, "Remote monitoring of a heterogeneous sensor network for biomedical research in space," in *Proceedings of the*

International Conference on Pervasive Computing and Communications, June 2004, pp. 829-833.

[71] F. Hu, M. Jiang, and Y. Xiao, "Low-cost wireless sensor networks for remote cardiac patients monitoring applications," ***Wireless Communications and Mobile Computing***, vol. 8, issue 4, Jan. 2007, pp. 513-529.

[72] Wikipedia, "Heart disease," http://en.wikipedia.org/wiki/Heart_ diseases, 2009.

[73] A. Ferreira, S. Shiu, and A. Baldwin, "Towards accountability for electronic patient records," in ***Proceedings of the 16th IEEE Symposium on Computer-Based Medical Systems***, Jun. 2003, pp. 189-194.

[74] S. Bhattacharya and R. Paul, "Accountability issues in multihop message communication," in ***IEEE Symposium on Application-Specific Systems and Software Engineering and Technology*** (***ASSET'99***), Richardson, USA, May 1999, pp. 74-81.

[75] Welsh, M., Myung, D., Gaynor, M., and Moulton, S, "Resuscitation monitoring with a wireless sensor network," ***Supplement to Circulation: Journal of the American Heart Association***, October 28, 2003.

[76] Gao, T., Greenspan, D., Welsh, M., Juang, R.R., and Alm, A, "Vital signs monitoring and patient tracking over a wireless network," ***Proceeding of the 27th Annual International Conference of the IEEE EMBS***, Sept. 2005, Shanghai, China, pp. 102-105.

[77] Milenković, A., Otto, C., and Jovanov, E, "Wireless sensor networks for personal health monitoring: issues and an implementation," ***Computer Communications***, Vol. 29, No. 13/14, pp. 2521-2533.

[78] Gaynor, M., Myung, D., Patel, R., and Moulton, S, "Human computer interaction in the pre-hospital setting," ***Proceedings of the 40th Annual Hawaii international Conference on System Sciences***, Jan. 3-6, 2007, Big Island, Hawaii, U.S.A.

[79] Liszka, K.J, "A sensor network architecture for cardiac health monitoring," ***The*** 4th ***IEEE Consumer Communications and Networking Conference (CCNC 2007)***, Jan. 2007, Las Vegas, NV, U.S.A., pp.737-740.

[80] G. Bella and L. C. Paulson, "Accountability Protocols: Formalized and Verified," ACM Trans. on Information and System Security, Vol. 9, No. 2, pp. 138–161, 2006.

[81] G. Bella, "Inductive Verification of Cryptographic Protocols," Ph.D. thesis, Research Report 493, Computer Laboratory, University of Cambridge. Accepted for publication as LNCS Monograph by Springer, 2000.

[82] J. Zhou and D. Gollman, "A fair non-repudiation protocol," in: 1996 IEEE Symposium on Security and Privacy, 1996. Proceedings, 1996, pp. 55–61.

[83] M. Abadi and N. Glew, “Certified email with a light on-line trusted third party: design and implementation,” in Proceedings of the 11th international conference on World Wide Web, New York, NY, USA, 2002, pp. 387–395.

[84] R. Jagadeesan, A. Jeffrey, C. Pitcher, and J. Riely. “Towards a theory of accountability and audit.” In ESORICS’09, volume 5789 of LNCS, p. 152–167. Springer, 2009.

[85] Z. Xiao and Y. Xiao, "P-Accountable Networked Systems," Proceeding of INFOCOM 2010, Work in Progress (WIP) Track.

[86] Z. Xiao, Y. Xiao, and J. Wu, “A Quantitative Study of Accountability in Wireless Multi-hop Networks,” Proceedings of 2010 39th International Conference on Parallel Processing (ICPP 2010), pp. 198 – 207.

[87] B. Fu and Y. Xiao, "Q-Accountable: A Overhead-based Quantifiable Accountability in Wireless Networks," Proceedings of IEEE Consumer Communications and Networking Conference (IEEE CCNC 2012), pp. 143-147.

[88] R. Küsters, T. Truderung, and A. Vogt, “Accountability: definition and relationship to verifiability,” in Proceedings of the 17th ACM conference on Computer and communications security, New York, NY, USA, 2010, pp. 526–535.

[89] J. Feigenbaum, A. D. Jaggard, and R. N. Wright, “Towards a formal model of accountability,” in Proceedings of the 2011 workshop on New security paradigms workshop, New York, NY, USA, 2011, pp. 45–56.

[90] J. Mirkovic and P. Reiher. “Building Accountability into the Future Internet.” Secure Network Protocols, 2008. NPSec 2008. 4th Workshop on, pages 45–51, 2008.

[91] G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. “Accountable Internet protocol (AIP).” ***SIGCOMM***, Aug 2008.

[92] C. Ko, D. A. Frincke, T. Goan Jr, T. Heberlein, K. Levitt, B. Mukherjee, and C. Wee, “Analysis of an algorithm for distributed recognition and accountability,” ***in Proceedings of the 1st ACM conference on Computer and communications security,*** 1993, pp. 154–164.

[93] A. Haeberlen, P. Kuznetsov, and P. Druschel, “The Case for Byzantine Fault Detection,” ***Second Workshop on Hot Topics in System Dependability,*** Seattle, WA, November 2006.

[94] A. Haeberlen, P. Kouznetsov, and P. Druschel, “PeerReview: Practical Accountability for Distributed Systems,” ***ACM Symposium on Operating Systems Principles (SoSP),*** October 2007.

[95] M. Burrows, M. Abadi, and R. Needham, “A logic of authentication,” ***ACM Transactions on Computer Systems***, vol. 8, no. 1, Feb. 1990, pp. 18-36.

[96] M. Abadi and M. R. Tuttle, "A semantics for a logic of authentication," in *Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing*, Aug. 1991, pp. 201-216.

[97] P. F. Syverson, "Adding time to a logic of authentication," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, Fairfax, Virginia, 1993, pp. 97-101.

[98] S. G. Stubblebine, "Recent-Secure authentication: enforcing revocation in distributed systems," *19th IEEE Symposium on Research in Security and Privacy*, 1995, pp. 224-235.

[99] S. G. Stubblebine and R. N. Wright, "An authentication logic supporting synchronization, revocation, and recency," in *Proceeding of the 3rd ACM Conference on Computer and Communications Security*, New Delhi, India, Mar. 1996, pp. 95-105.

[100] R. Kailar, "Accountability in electronic commerce protocols," *IEEE Transactions on Software Engineering*, vol. 22, no. 5, May 1996, pp. 313-328.

[101] M. Kudo, "Electronic submission protocol based on temporal accountability," in *Proceedings of the 14th Annual Computer Security Applications Conference*, 1998, pp. 353-363.

[102] J. Liang, Q. Ao, and J. You, "Analyzing the temporal accountability of secure protocols," *Chinese Journal of Electronics*, vol. 30, no. 10, Oct. 2002, pp. 1451-1454.

[103] H. Chao, "Price-responsive demand management for a smart grid world," *The Electricity Journal*, vol. 23, issue 1, 2010, pp. 7-20.

[104] U.S. DOE, "Average retail price of electricity to ultimate customers by end-use sector, by State", Nov. 2010, available online: http://www.eia.doe.gov/electricity/epm/table5_6_b.html.

[105] Y. Quan and G. Liu, "Drifting clock model for network simulation in time synchronization," in *Proceedings of the 2008 3rd International Conference on Innovative Computing Information and Control*, Dalian, China, June 2008, pp. 385-389.

[106] Michael Galloway, Yanping Zhang, and Peng Shao, "Time synchronization in sensor networks and underwater sensor networks," *Underwater Acoustic Sensor Networks*, *Auerbach Publications, Taylor & Francis Group*, ISBN-10: 1420067117, ISBN-13: 978-1420067118, 2009.

[107] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, U.S.A., Nov. 2004, pp. 13-24.

[108] M. K. Reiter and A. D. Rubin, “Crowds: anonymity for web transactions,” *ACM Transactions on Information and System Security*, vol. 1, issue 1, 1998, pp. 66-92.

[109] H. Chen, Y. Xiao, X. Hong, F. Hu, and J. Xie, “A survey of anonymity in wireless communication systems,” *Security and Communication Networks*, vol. 2, issue 5, Dec. 2008, pp. 427-444.

[110] J. Liu, Y. Xiao, S. Li, W. Liang, and C.L.P. Chen, "Cyber security and privacy issues in smart grids," *IEEE Communications Surveys & Tutorials*, vol. 14, issue 4, 2012, pp.981-997.

[111] J. Liu and Y. Xiao, "An accountable neighborhood area network in smart grids," in: *Proceedings of 7th FTRA International Conference on Embedded and Multimedia Computing (EMC 2012)*. (accepted)

[112] J. Liu, Y. Xiao, and J. Gao, “Achieving accountability in smart grids,” *IEEE Systems Journal*, accepted, 2013.

[113] J. Liu and Y. Xiao, "Temporal accountability and anonymity in medical sensor networks," *ACM/Springer Mobile Networks and Applications (MONET), Special issue on Mobility of Systems, Users, Data and Computing*, vol. 16, issue 6, Dec. 2011, pp. 695-712.

[114] J. Liu, Y. Xiao, C.L.P. Chen, "Authentication and access control in the Internet of Things," in: *The 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW 2012)*, Macau, China, June 2012, pp.588-592.