Mobile Netw Appl (2011) 16:695–712
DOI 10.1007/s11036-010-0254-6

Temporal Accountability and Anonymity in Medical Sensor Networks

Jing Liu · Yang Xiao

Published online: 2 July 2010

Abstract The increasing number of elderly patients in the world has lead to various new appliances and technologies in the modern tele-healthcare platform. One such application is the medical sensor network (MSN). In this application, 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 that the critical time signal is accountable, we propose a new architecture for the MSN that is capable of temporal accountability. In addition, it also preserves privacy ability via a Crowds anonymous system. The analysis results clearly indicate the advantages of being our proposed methods in terms of low-cost and reliable and having scalable features.

Keywords temporal accountability · medical sensor networks

1 Introduction

Heart disease, also known as cardiovascular disease, continues to be the leading cause of death worldwide and the top killer in the U.S. and the western world. A WHO (World Health Organization) report indicates that heart disease accounts for 30% of deaths globally. Specifically, coronary artery disease (CAD), a typical heart disease, kills an estimated 459,000 Americans every year [1]. As of 2007, in the U.S. alone, a person dies of heart disease every 34 sec [1]. No country spends as much 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. In 2006, the American Heart Association estimated that healthcare would cost Americans over $258 billion. It also predicted that healthcare will consume 20% of the U.S. GDP by 2010. At present, more and more of the elderly go to nursing homes. Most of them have heart diseases. Therefore, we need a regional (e.g., within a nursing home) and low-cost medical delivery system to monitor the status of patients automatically.

Tele-healthcare is a technology that uses communications and computing to implement high-quality healthcare regardless of location. Recent technological progress in wireless communication, micro-electro-mechanical systems (MEMS), cryptography, and digital electronics have caused the tele-healthcare system to become more sophisticated. One promising wireless tele-healthcare application is the medical sensor network (MSN) [2]. In this system, patients are monitored by 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 will reveal patients' real-time situations after a series of analysis procedures. Once an undesirable status has been detected, doctors or nurses may take further action on that particular patient (e.g., remind him/her to take pills immediately via telephone). Although the new platform saves time for

J. Liu · Y. Xiao (✉)
Department of Computer Science, The University of Alabama, 101 Houser Hall, Box 870290, Tuscaloosa, AL 35487-0290, USA
e-mail: yangxiao@ieee.org

patients to see a doctor, problems still exist in the MSN that cannot be ignored. As we know, the medical sensors may have different functions, such as detecting electrocardiographs (ECG), heart rate, blood pressure, or pulse oximetry (SpO_2). All those parameters are important to timely detection and classification of abnormal physical statuses. To obtain accurate values for all those parameters in an unreliable wireless network is the goal of ongoing research. Nevertheless, it is hard to meet such an expectation because of the limitations of medical sensors.

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). Therefore, in order to build a regional and low-cost MSN, we need to adopt a patient-to-patient (hop-to-hop) transmission relay scheme and a "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 the power of each sensor. On the other hand, the sensor has deficient usability and poor security, especially regarding its immature patient privacy preserving technique. Hence, many hospitals and patients are afraid of using current tele-healthcare systems. A balance between their usability and credibility needs to be achieved [3]. According to the study in [4], we believe that a multihop message communication system cannot be well protected by only a typical security technology (i.e., digital signatures and cryptography). As a complement, accountability and anonymity are required to secure the MSN.

Generally speaking, accountability means that the system is recordable and traceable, therefore making it liable to those communication principles for its actions. Together with some suitable punishments or laws in the real world, it will prevent a number of attacks from being mounted. Albeit general system accountability could preserve the integrity and confidentiality of data transmission, the MSN still has no protection against temporal signal spoofing. It is obvious that the accuracy of a retrieved ECG trace depends on the accuracy of the temporal signal within each received packet. Any change, regardless of whether it derives from an attacker's spoofing or from a damaged sensor, may lead to quite another result. To identify the cause, we should make the temporal signal accountable. In our design, each node in the MSN acts as both a sender and an observer. Since most wireless devices use broadcast to deliver messages, every node within their communication range may capture the messages even they are not the destination. Thus, the temporal signal within the message is exposed to all nodes near the transmission path. With the help of observation from these nodes, almost all temporal signals are accountable and able to be detected if they have been modified by malicious intermediate nodes.

For the privacy issue, since the sensor ID on patient's body corresponds to the patient's profile record in the medical database, the disclosure of information sources during wireless communications can cause a violation of the patient's privacy. Moreover, when such MSN platforms are widely deployed in national medical sites (such as nursing homes, hospitals, etc.), they could become the potential targets of cyber-terrorists. Considering the confidentiality of all medical data, we need an end-to-end security scheme to protect them. This can be achieved through the 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 effectively integrate the existing anonymity technology into the MSN and how to overcome current privacy problems in this paper. To minimize the communication cost and obtain a certain degree of anonymity, we select "Crowds" out of three typical anonymous communication systems [5].

The rest of this paper is organized as follows: Section 2 provides some background knowledge on the topics of the MSN, accountability, anonymity, and other work supporting this research. Section 3 presents our design and architecture for the new MSN platform. Section 4 mainly evaluates our design using logical testing. More insights regarding implementation are offered in Section 5. Future challenges and recommendations for some subsequent work are discussed in Section 6. Finally, we conclude this paper in Section 7.

2 Background knowledge

Before going into the details of our design, a brief overview of the MSN, accountability, time synchronization, and anonymous communication technology is presented. In order to verify the temporal accountability of this new system, we also review a couple of accountability logic schemes. More related work can be found in [6–33].

2.1 MSN

An MSN, or medical sensor network, is a tele-healthcare platform in which the patient's physical status is delivered and monitored. In such a system, 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, and ECG signals) will eventually reveal the patient's real-time situation by using professional software [2]. Once an abnormal signal has been

detected, doctors or nurses may take further action on that particular patient (e.g., remind him/her to take pills immediately via telephone). In practice, ECG signals play a very important role in the MSN system. Each piece of ECG data may carry significant medical information [2]. Data error or loss is not tolerated. Fortunately, scholars and researchers have already explored a way to deliver continuous and stable ECG signals in a radio-based wireless network [2]. Since we are mainly concerned with the accountability of the time signal in the ECG data, our design will not focus on an error-resistant approach to the MSN system.

To best understand ECG signals, a brief review is quite necessary. Generally speaking, an ECG is used for detecting abnormal heart rhythms, excessive tensing of the heart muscle, and blood and oxygen supplies [34]. 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 [35]. A commonly used ECG recorder comes with 12–15 leads with electrodes [36]. 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 [2]. 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 [2].

Figure 1 shows an example of a typical ECG trace, which has three major parts: a P wave, a QRS complex, and a T wave [37]. 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' repolarization [37]. The presence or absence of these waves, including the QT interval and the PR interval shown in the figure, are meaningful parameters in the screening and diagnosis of cardiovascular diseases [37].

There are several ECG system options for the MSN to choose from. They have different lead placements which range from 3-lead to 12-lead [37]. The 3-lead system is non-diagnostic and is meant for rhythm interpretation, while the 12-lead system is diagnostic [37]. 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 [37]. Hence, a 3-lead system is the preferred choice for our design.

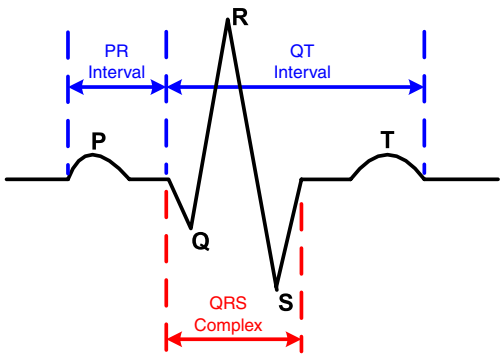


Fig. 1 A typical ECG trace [37]

As shown in Fig. 2, in a typical MSN, the patient's medical information is collected by a wearable wireless device, e.g., PDA. Then it is delivered to nearby access points (AP's) via hop-by-hop wireless communication. Then through wired or wireless channels among different AP's, the collected information is transferred to a nursing home monitor center. To some extent, this architecture is scalable, manageable, and easily deployed. Our work is therefore illustrated based on such an architecture.

2.2 Accountability

Developing the Electronic Patient Record (EPR) will not only speed and simplify access to healthcare information, but will also enhance the quality of this information [3]. Nevertheless, its poor security and deficient usability become the biggest deterrent to its success over a wide range. Security concerns about the EPR in current tele-healthcare systems are widely studied. One important issue is ensuring people will be associated with their actions—that is, accountability [3].

Typically, tele-healthcare platforms have several specific audit methods to help achieve information security and to ensure that processes are correctly followed [3]. In a paper-based system, people often utilize practice experiences or well-understood processes to achieve accountability [3]. However, electronic information has quite different characteristics than paper, and the translation of audit and accountability processes to this new medium is not straightforward [3]. A balance between the abilities of current technology, usability, and credibility needs to be achieved [3]. In a heterogeneous environment, the audit information is collected from different sources. It may be stored in a centralized machine and used only when necessary, or it may be accessed daily by authorized system administrators [3]. In both cases, the information should be accessible and usable while preserving its integrity and confidentiality [3]. In [38], a novel accountable logging was proposed and called Flow-net. In [39–41], quantitative approaches for accountability as proposed and studied.

Accountability has been defined in several ways. Bhattacharya and Paul, in [4], 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 [3], state that the main objective of accountability systems is "to provide a means to verify, analyze, and investigate users' actions"

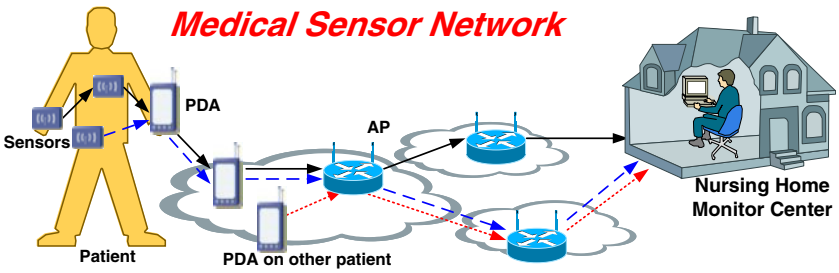


Fig. 2 A typical MSN architecture

and "to ensure procedures are correctly followed." Generally speaking, accountability means a system is recordable and traceable, therefore making it liable to those communication principles for its actions. Every single change in 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 privilege. In [42], an algorithm to achieve true accountable administration was proposed, namely, that an administrator's activities must be accountable. Together with some suitable punishments or laws in the real world, accountability prevents a number of attacks from being mounted.

Since the message gathering from each sensor will eventually deliver data to an MSN monitor center, it is necessary to examine the accountability of the multihop message delivery system. According to the classification in [4], accountability issues in a multihop message communication system involve: (1) non-repudiation of origin (NRO), (2) non-repudiation of receipt (NRR), (3) non-repudiation of submission (NRS), and (4) non-repudiation of delivery (NRD). NRO and NRR measure the accountability at the source and destination nodes, respectively [4]. NRS and NRD focus on all the nodes in the system [4]. For the accountability of an MSN, all four of these properties must hold for each node at any given time. More specifically, the following relevant denials should be resolved in the MSN:

D1: Denial of authorship of a message
D2: Denial of sending a message
D3: Denial of receiving a message
D4: Denial of sending or receiving a message at a given time
D5: Denial of having forwarded a message across an intermediate node

For critical transactions, accountability may include timing aspects as well. For example, in a stock purchase situation, if the buyer cannot prove that he/she sent the message at a particular time, then the motivation for the entire transaction, and refusal thereof (by the receiver), may lose significance. Granting permission during the tele-healthcare process to permit access to certain information or to allow certain actions to be taken implies the need for temporal logic and control, i.e., temporal accountability. There have only been a few studies [43–47] conducted on temporal network accountability. Most of them work for electronic transaction. As we can see, a buyer can compare the price on a receipt and the amount paid later from their bank account to verify a purchase statement from Internet credit-payment systems. However, it is difficult to verify the merchant's system clock's accuracy, which is important if the merchant engages in temporal transactions such as discount selling during a limited period. An order via e-mail can use the corresponding mail server system clock as a trusted clock to verify the correctness of the discount rate on the purchased item. However, the user may connect to an untrusted application server instead of a trusted third party (TTP) so that the clock of neither the application server nor the user's system is trusted. Furthermore, a malicious party may alter or forge temporal records stored in the user's machine or application server. There are two kinds of attacks: 1) changes of the system clock by the user, the server, or both in conspiracy and 2) forgery, alteration, or removal of temporal records by the user, the server, or both in conspiracy. For example, the server can set the clock a little faster to refuse acceptance of an application from a user, giving as a reason that the deadline just passed. An author and a guest editor of a special issue of a journal can participate in the forgery of temporal records about paper submissions: the author submits a paper after the deadline, and the guest editor forges the record of receiving the paper to say that it was received before the deadline. The work in [4] pointed out that traditional cryptographic schemes cannot overcome internal malicious users who send untrustworthy timestamps. The author and the guest editor mount a conspiracy attack that involves forging temporal records so that the paper appears to have arrived before the due date. Countermeasures include periodic adjustment of the system clock, distribution of temporal transactions on multiple machines, use of a TTP for a correct timestamp, and registration of temporal transaction records with another TTP. However, these solutions are not sufficiently secure against attacks because the person with the highest administrative access rights can easily change the system clock at any time. A distributed server group computation may ensure the integrity of temporal transactions but may also suffer from the previously mentioned access-right problem. The

following functions are required: 1) Temporal transaction records must be published outside the server; 2) The temporal transactions of the server should be monitored synchronously by all parties; 3) Registrations provided by a TTP are required; 4) A trusted timestamp server is required. Otherwise, preventing forgery, alteration, or removal of records inside the server cannot be prevented.

2.3 Time synchronization

Since temporal accountability is required in our MSN system, we need a strategy or a mechanism to synchronize the time signals among all sensor nodes and wireless devices. It should be an effective approach that is accurate, lightweight, flexible, and comprehensive. Actually, scholars and researchers have proposed many ways to address the time synchronization problem in current wireless sensor networks (WSN's). In order to select a suitable approach for our design, a brief overview of time synchronization in WSN is presented in this subsection.

Usually, timestamps are utilized to synchronize the clocks on each communication node [48]. All messages transferred in the network have a timestamp with a local system clock. As a matter of fact, instead of an absolute time, the timestamp is more likely to be a boundary of time in which a certain event occurs. Every time a node sends a new message, the latest timestamp is processed according to the local clock and attached to the message. Meanwhile, drift rates between nodes in the network are considered for better approximation of the boundary of the timestamp.

Most WSN's require that their nodes agree on a common global time. Having a global clock will help nodes in the network be more efficient in transmitting data and will let them conserve energy by knowing when to go into a low power state. The drawback for using a global clock to "wake up" nodes is that more complex communications have to be in place to keep the nodes' clocks synchronized within a bounded limit. In this subsection, we will discuss three ways to implement global clock synchronization in a wireless sensor network: the all-node-based method, the cluster-based method, and the fully localized diffusion-based method [48].

2.3.1 All-node-based synchronization

In this approach, we assume that the clock cycle on each node is the same. We also assume that the clock tick cycle is longer than the transmission time. This is an important assumption because a lower clock frequency consumes less energy than a higher clock frequency.

The initial setup finds a cycle that passes through each node that needs to be synchronized in the network at least once. The initiating node then sends a message to each node in the cycle. This message has the first node's current time. When the other nodes in the cycle receive the message, they record their local time and their order in the cycle. When the initial node receives the message back, it sends out another message with the start time (t_s) and end time (t_e) of the previous message cycle. For each node to adjust its local clock (t) to the global clock, the equation $t = t_e - t_i + t_s$ is used, where t_i is the node's local time. After all nodes in the network receive the second message and compute their new time, the network is globally synchronized.

2.3.2 Cluster-based synchronization

The problem with the all-node-based synchronization algorithm is that every node in the network has to participate in the same synchronization session. The cluster-based synchronization protocol attempts to fix this problem.

The cluster-based algorithm works the same way as the all-node-based synchronization, but only with the head cluster nodes. Head nodes synchronize themselves first, and the child nodes in each cluster then synchronize with the head node. This method of synchronization allows some clusters to synchronize independently of other clusters. This also helps conserve energy, as some clusters may have to synchronize more often than others and therefore leave the less frequently updated nodes in an energy saving state.

2.3.3 Fully localized diffusion based synchronization

The above two synchronization methods do not work well in very large scale wireless networks. They have single points of failure, in which case some nodes will never be synchronized again. The local diffusion based protocol is an attempt to fix this problem.

This method works like the first one. Nodes exchange local times with neighboring nodes. This diffusion occurs until all sensor nodes in the network have updated their local clocks. One of the benefits of this time synchronization protocol is that there is no point of failure. If a node wishes to update its local clock, it can use any of its neighbors. No single node initiates a global synchronization of all nodes in the network.

There are two versions of the localized diffusion based algorithm: the synchronous diffusion algorithm and the asynchronous diffusion algorithm [48]. The synchronous diffusion algorithm has to perform the operations to obtain a bounded global clock in a set order. All of the nodes that are finished synchronizing with their neighbor must wait for the other nodes that have not finished. The asynchronous diffusion method does not have the time constraint of waiting for all other nodes in the network to finish their

synchronization before starting another round. This method allows any node in the network to initiate its own time synchronization whenever necessary.

In some cases, WSN's must synchronize their time with the fastest clock. No node will be permitted to decrease its clock time for synchronization. There are many such implementations of the clock synchronization protocol [48]:

Wireless Clock Synchronization Protocols: These protocols are exclusively applied in wireless or ad hoc networks. Instead of adjusting the clock on each node in the network, this protocol adjusts the timestamp. From a source to a destination, the message is passed along each node and has its timestamp changed to match each node's local clock. Error is introduced in the time stamp along the path of the nodes, and more error is introduced as the path gets longer. The 802.11 protocol implements this kind of clock synchronization and has been found to not be scalable to a large number of nodes due to the increased error introduced by each hop.

Receiver-Receiver Synchronization: This type of clock synchronization is done among individual nodes in the network. Any node that needs a clock update can initiate synchronization.

Probabilistic Clock Synchronization: This type of clock synchronization tries to correct unacceptable update methods. One attempt is to interpret the clock of the master node repeatedly until a reply is accepted. This algorithm takes $2/(1-p)$ messages from the master node, where p is the probability of losing the message. This algorithm is repeated n times to reach a probability of synchronization equal to $1-p^k$. This protocol wastes excess amounts of energy by sending large amounts of messages to synchronize the nodes.

Delay measurement time synchronization (DMTS): This protocol trades accuracy for efficiency [48]. It functions by evaluating different forms of delay while updating local clocks in the wireless sensor network. The Berkley motes platform and TinyOS were used in implementing this algorithm. Clock synchronization is dependent on the types of oscillators on the motes. DMTS intends to use low computation complexity and low memory occupation to conserve energy in the nodes. This protocol also aims to be flexible in different network topologies.

To initiate the resynchronization, a leader node is selected as the time master and broadcasts its time. When the message reaches the nodes within the range of the leader, they set their clock to match the leader's clock plus the time delay it took to receive the message. After all nodes receive the message from the leader, their time will be resynchronized bounded by the efficiency of the delay measurements along the path. Delay is composed of the 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. Since only a single message is needed to synchronize all nodes within the leader's transmission range, this method is energy efficient. It is also computationally lightweight, as there are no complex numerical operations involved.

However, single-hop is never enough in a network. Furthermore, in most networks, nodes have no knowledge of their children. DMTS uses the concept of a time-source level to identify the network distance of a node from the master. First, a time master node is chosen. This node is labeled level 0. All of the immediate child nodes of the time master are labeled level 1. Other nodes in the network at level n are labeled level $n+1$. This way each node has a level number and knows how many hops away it is from the time master node. The time master node will occasionally send a message with its local time. Each node that receives a message from the time master node will update its time and broadcast the time message again. These nodes broadcast the time message only once. When the message propagates through the network, other nodes will resynchronize according to their parent node which is closest to the master node. It then sends a message with the time to its neighbors. This happens until all of the nodes in the network have synchronized their time with the time master node. This protocol also produces a minimum amount of traffic by having each node send a time message only once. The traffic load is directionally proportional to the number of nodes in the network.

The time master node can be any node in the network. Any selection algorithm can be applied to choose the master, but in most cases simple voting algorithms are used. The algorithms are run at any time when a master node is not present in the network. The best practice is to select a base station (e.g., the monitor center) as the time master. A reason for this is that the base stations are typically not mobile and are not maintained by batteries. The base station also typically has more processing power than the sensor nodes in the network.

Reference Broadcast Synchronization (RBS) protocol: This protocol is so named because it exploits the broadcast property (transmitting property) of the wireless communication medium. Instead of sending a message with a timestamp, the nodes use the message's time of arrival to compare their local clocks. One of the important constraints of using this protocol is that it requires a physical broadcast channel. It will not work with a WSN that employs direct point to point links. Thus, we do not consider RBS in our design.

In the MSN environment, high wireless communication energy consumption should be avoided. They need to decrease the frequency of sensor-to-sensor message transmissions to exchange keying materials or timestamp control messages. As we discussed, to avoid the large shortening of medical sensor lifetimes, a DMTS approach will be adopted in the MSN. This so called "receiver-only" local timestamp analysis has much better energy efficiency in wireless communications.

2.4 Anonymous communication

A user authentication process is often required when the user connects to a wireless network. A mobile user may roam off its home network and use another network's services. The visiting network often needs the user's credentials to authorize its use of these services. In wireless ad hoc networks, nodes often act as routers which need routing information. In the above scenarios, while availing network services, many applications and services are required to maintain the user's anonymity. To maintain a user's anonymity, two categories of information need to be protected: (1) movements and locations of network users and (2) activities of the network users, i.e., messages sent from or to the user [49]. The former is often referred to as location anonymity (or privacy protection) and the latter as data origin/destination anonymity (privacy protection) [49]. In this paper, we focus on privacy protection.

Anonymity has two meaningful impacts [49]. First, implementing the effective anonymity of a network user reduces security breaches under various attacks. Many attacks are launched by means of impersonation. Keeping a network user's identity anonymous prevents an unintended party from associating its identity with the messages sent to or from the network user or participating in the user's network sessions which the unintended party is not supposed to be in. In other words, it prevents the unintended party from impersonating the network user. Second, implementing the effective anonymity of a network user prevents unintended parties from invading the user's privacy.

Anonymous communication is an effective mechanism for protecting a user's privacy and also complies with the principle of least information [49]. Many studies aim to provide anonymous communication channels and to deter attacks on these channels [49]. Practical anonymity services such as Tor have been deployed and have protected privacy and deterred censorship for many users. The emergence of wireless networks has posed additional challenges to anonymity, such as those stated in reference [49].

2.5 Temporal 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 on accountability logic, most of which is designed for electronic transaction. BAN's logic, known as the first logic in the analysis of secure protocols, can be used for authenticating and uncovering flaws [50]. Nevertheless, it also generates controversy and confusion under certain conditions. Fortunately, this drawback has been addressed by Abadi and Tuttle in AT's logic [51]. 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 [43] mentioned that the logic of both BAN's and AT's logic could not capture flaws caused by a "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 [52] introduced the 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) [53] extended BAN's logic for 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 [53].

Later, Kailar [44] proposed accountability logic for electronic commerce protocols such as payment and public key distribution protocols. He defined 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 [46]. 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, such as a time-critical delegation key, Kudo [46] extended Kailar's logic so that it could represent 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. [45] claimed that the seventh logic postulate of Kudo' logic could not prevent replay attacks. By adding integrity verification based on timestamps, Liang's logic added 4 more logic constructs and 2 more logic postulates (e.g., *x At t*, *x Freshbefore t*) to Kudo's. However, without a TTP support, Liang's logic will be no difference than Kudo's.

Several formal logics were proposed to analyze the accountability of e-commerce protocols, which is related to all kinds of accountability issues. Since we do not use a TTP

in our own design, we would like to utilize Kudo's logic provability in this paper.

3 Design and architecture

The MSN is a promising platform for tele-healthcare systems. It can help patients save time and money. It also optimizes medical resources so that every patient can receive better treatment. However, the current MSN is not sufficient regarding integrity and security. With the widespread use of computing and communication technologies, the issues of integrity and security have become increasingly prominent in a democratic society. In the MSN, however, integrity not only refers to the completeness of transmitting data in the wireless context, but also considers the time consistency of delivered data between the sender and the monitor center. Unfortunately, most existing work is dedicated to ensuring data completeness and neglects 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 (e.g., delay, forge, or denial of delivery) may lead to quite another output. But 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 the path from the sender to the monitor center. We should find out who is responsible for this alteration and at what time it happens. On the other hand, different 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 the MSN as follows:

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's or the receiver's clocks are not trusted, as they may be slower or faster.

Without solving the above temporal challenges, medical results generated by the MSN are untrustworthy. We therefore propose a feasible solution for each of them. As for **Challenge 1**, we will adopt the accountability notion for all temporal records. Any modification of temporal signals should be accountable. Together with certain laws and punishments, temporal attacks (e.g., forgery, alternation, delay, or removal of temporal records) should be prevented. As for **Challenge 2**, we plan to design an appropriate time synchronization method. It should consider the energy consumption and deviation of local times among different wireless sensors and devices.

Another significant concern within the MSN is maintaining trust and confidence between the patient and physician. Maintaining the confidentiality of a patient's medical record is of great importance. Nevertheless, it becomes a controversial topic when computerized information systems are used 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 maintain privacy. Temporal accountability, however, is contradictory to anonymity. Evidence of temporal records can be used to reveal the sender's identity. Therefore, the third challenge that should be addressed in our MSN is:

Challenge 3 Maintain the sender's privacy while preserving temporal accountability.

In our design, we will adopt a Crowds system to enable anonymous communication for **Challenge 3**. According to the forwarding strategy of the Crowds system, messages are delivered in a dynamic way.

Based on the above discussion, this paper should address the three major challenges mentioned. Before describing our design, we should first clarify several terminologies and system assumptions as follows:

3.1 Definitions

Terms

$\{A, B, ...\}$	a set of communication participants, known as principals. Specifically, 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.

Definition

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 the communication range of *A*.
Temporal Evidence	means that the MSN will keep a log file or take similar approaches to record any modification of temporal records.
Temporal Undeniable	means that every communication principal cannot deny its actions shown by the temporal evidence.
Protect Privacy	means that the identity of the sender cannot be disclosed by any user or attacker except the authorized agency (e.g., the monitor center).
Synchronize temporal signal	means that all the data generated by communication principals hold the same (or with a little deviation) time clock as the recipient (e.g., the monitor center).

3.2 Assumptions

Our ideal MSN platform is built upon the following assumptions:

N1: The monitor center is assumed to be trusted, and its clock is assumed to be accurate.
N2: Every two principals have at least one mutual neighbor in their common communication range.
N3: The local time of each principal except the monitor center is not trusted; we do not assume any clock synchronization for them.
N4: The channel for wireless communication is assumed to be unsecured, and all traffic in the MSN can be observed by any principal.
N5: No packet eventual loss occurs during transmitting in wireless context; each packet will eventually arrive at its destination.
N6: The digital signature and message encryption algorithm is based on public key cryptography, and no private key can be compromised by intruders.
N7: The computing and storage resources for the monitor center and AP's are assumed to be unlimited.
N8: No denial-of-service attack occurs in the MSN.
N9: All wired communication channels are secured.
N10: No AP spoof attack occurs in the MSN.

3.3 Design principals

In this paper, we mainly focus on solutions to the aforementioned three challenges. Hence, our design principals are presented accordingly:

DP1: *Achieve temporal accountability in the MSN. This targets* ***Challenge 1***.
DP2: *Synchronize all temporal records in the MSN. This targets* ***Challenge 2***.
DP3: *Implement privacy protection in the MSN. This targets* ***Challenge 3***.
DP4: *Analyze and evaluate our MSN under different kinds of assumptions in terms of temporal accountability and privacy protection.*

3.4 Temporal accountability module

This subsection is dedicated to addressing the temporal accountability issue in **DP1**.

There have only been a few studies [43–47] conducted on network temporal accountability. Most of them work for electronic transactions in wired networks. In an MSN, unlike the temporal accountability logic of electronic commerce, the "receiver" (monitor center) does not assign a permitted period to each "sender" (medical sensor) since they may constantly deliver messages with vital signals. However, a medical message is time sensitive. It is only valid during a certain period. For example, if a sensor detects a heart attack and delivers the relevant information immediately, we argue that the medical center should receive this message as soon as possible, as it will be useless after a patient's death. We therefore still 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 temporal signal for some reasons. Some may be damaged and others may be manipulated by malicious people; none of these are expected to be received. They should be detected and traced back.

Based on assumptions **N4, N6,** and **N9**, we should only consider two transmission scenarios. As illustrated in Fig. 2 (in subsection 2.1), the first is a sensor-PDA scenario and the other is the transmissions between different PDA's and AP's. They are both in the wireless environment. The major difference is that the latter has a more powerful capacity for computation and storage. Since PDA's and AP's can be regarded as super sensor nodes with more powerful computing and storage capabilities, it is reasonable to merge these two scenarios into one, as shown in Fig. 3. This combined module involves two parts: 1) multiple transferring principles and 2) one receiver. As in the sensor-PDA scenario, the first part represents the sensors and the receiver stands for PDA. PDA's are regarded as devices held by patients. They are responsible for receiving and recording all medical messages 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 a multi-hop route through several PDA's or AP's.

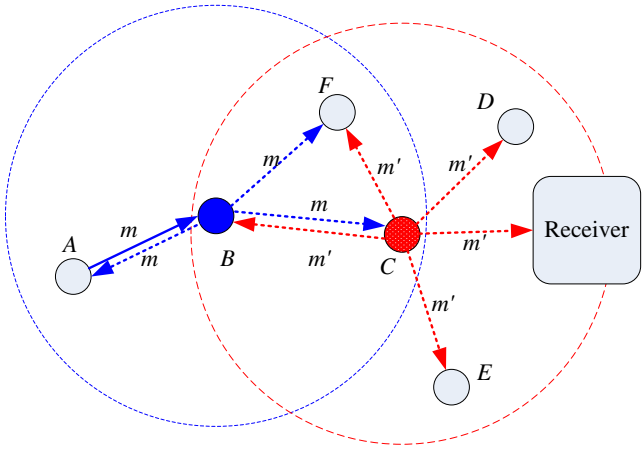


Fig. 3 Temporal accountability module

The records can be utilized for temporal accountability. As in the PDA-AP scenario, multiple transferring principles can be regarded as PDA's or AP's and the receiver is the monitor center. Hence, if we achieve temporal accountability in this module, we address **DP1**.

Generally speaking, to send or forward a message in a wireless environment, a principle simply broadcasts it in its own 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 shortly. Based on this observation, we propose a feasible solution for temporal accountability in the MSN.

Specifically, each principal except AP's and the monitor center should hold a *memory* to record recently processed or received packets for further review. The *memory* should record the packets that have just been sent, forwarded, or passively received. As illustrated in Fig. 3, once a principle B wants to send a message m to the receiver, B will proceed along the following procedures: 1) It sets the receiver as the destination; 2) It signs m with its unique signature key K_B^{-1}; 3) It records $\{m\}K_B^{-1}$ in its *memory*; 4) It sends $\{m\}K_B^{-1}$ to the next hop C via broadcasting. We notice that principal B has 3 neighbors (A, C, and F) for in Fig. 3. All of them will receive $\{m\}K_B^{-1}$. Since A and F are not the next hop of this message, they simply record $\{m\}K_B^{-1}$ into their own *memory*. But for principal C, it will process and forward this message to the receiver.

Based on the assumption **N2**, every two principals will have at least one mutual neighbor for surveillance. Assume that principal F is the mutual neighbor between principal A and principal B. Then, 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 for messages $\{m\}K_B^{-1}$ and $\{m'\}K_C^{-1}$. F can also compare the received message $\{m'\}K_C^{-1}$ with message $\{m\}K_B^{-1}$ in its *memory*. We denote $\{m\}K_B^{-1}$ as temporal evidence of $\{m'\}K_C^{-1}$. Once m and m' are satisfied by the predefined temporal requirements (discussed in subsection 3.7), 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 the relevant temporal evidence (e.g., $\{m\}K_B^{-1}$. Detailed information about the traffic flow will be discussed in subsection 3.7.

Here, the predefined temporal requirement is a threshold vale TH defined by the monitor center. When F receives m, F will mark the received time as t_1. Similarly, when F receives m', F will record the received time as t_2. Notice that both t_1 and t_2 are relative to F's local time. After that, F will calculate the difference between 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 TH, we say m and m' are satisfied by predefined temporal requirement.

Based on the assumption **N2**, every two principals will have at least one mutual neighbor for surveillance. That means all the actions will be monitored by at least two different principals. Thus, almost all malicious modifications on temporal signals will be captured. Then we have achieved temporal undeniable and addressed **DP1** here.

3.5 Time synchronization module

This subsection is dedicated to addressing the time synchronization issue in **DP2**.

As we discussed in subsection 2.3, the synchronization accuracy of DMTS is limited mainly by the precision of the delay measurements along the path. For the sake of ensuring high accuracy and energy efficient in our design, we need appropriate modifications on DMTS protocol. The modified protocol should also be computationally lightweight. No complex numerical operations are involved as well. Therefore, we have modified the method proposed in [54].

We illustrate this approach through an example. In Fig. 4, 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 sample from S_1 is given by: $T_{S1} = t_{S1} + t_A + t_B + t_{p1} + t_{p2} + t_{p3}$.

Noting 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 neglected. 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 this packet reaches the receiver, the receiver notes the time (its own local time) at which it received this packet as τ_{S1}. Hence, the sample must have

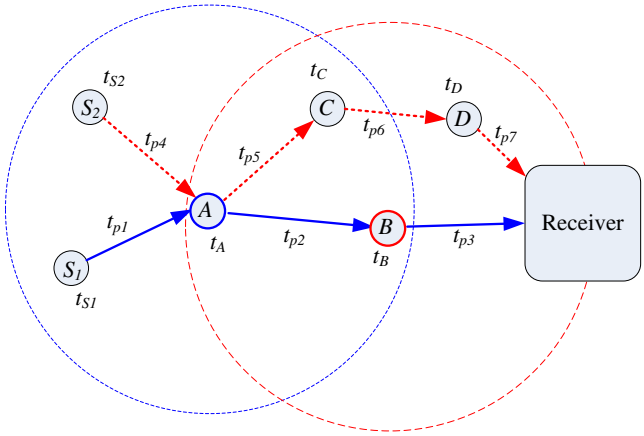


Fig. 4 Time synchronization module

been generated at $\tau_{S1}-T_{S1}$ (T_{S1} is obtained from the packet) in the local time of the sink. The same procedure is applied for sample *S2*.

As we can see, the cumulative residence time can be used for estimating the occurrence time of the message. Therefore, we take the cumulative time as the timestamp stored in the message. When this kind of message arrives at the monitor center, the occurrence time of the message will be revealed by the aforementioned approach.

This scheme eliminates many of the errors that time synchronization schemes have to contend with because we compute residence times close to the device [54]. 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. First, if the residence times are long (as they can be with our compression schemes), then the timestamp can be significantly skewed. Second, clock drift can change the sample clocking, i.e., individual samples may not be exactly 10 ms apart when sampling at 100 Hz. The latter "problem" might be considered unimportant, as the device would be sampling the phenomenon correctly (when it happens), just not at the frequency it was supposed to. We return later to discuss the former problem.

In summary, our "receiver-only" local timestamp analysis scheme incurs little overhead (a residence time field in every packet) and can be implemented easily as we now discuss.

3.6 Anonymity module

Laws and regulations, such as privacy and freedom of information acts, have been introduced to assure and safeguard individual rights. However, the meaning and actual requirements of information and data security for protecting these rights have not been clearly understood. Information security concerns in tele-healthcare systems are particularly sensitive.

The EPR is subject to many potential abuses. It should be used only for the intended healthcare purpose. There are organizations and individuals, including medical support personnel, health related researchers, insurance agents, medical administrators, and patients' relatives, who have certain legitimate needs to access relevant information from the EPR. Conditions under which they are permitted to access information are not clearly understood, and issues related to these conditions are often emotional, controversial, and ambiguous. To have a clear understanding of these conditions is an extremely difficult task.

Some laws and regulations are applicable for specifying how EPR should be handled, but they are subject to different interpretations. Patients must trust tele-healthcare systems to protect their privacy rights. However, EPR's are being used by various medical and administrative personnel, with each having has different professional and legal responsibilities. Tele-healthcare systems are "risky systems" with respect to privacy and confidentiality. This subsection is dedicated to addressing the anonymity issue in **DP3**.

There are three typical anonymous communication systems for reference: MIX, Onion Routing, and Crowds. For simplicity and energy efficiency, we will adopt a Crowds system [5] in our design to enable anonymous communication. According to the forwarding strategy of the Crowds system, messages are delivered in a dynamic way (shown in Fig. 5). More specifically, when a message arrives at a Crowds' router between the sender and receiver, the router 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 in Crowds routers. This strategy therefore guarantees to some extent that the sender's identity cannot be revealed.

3.7 System framework

The system that we propose here involves two major parts: a Crowds-based MSN and the nursing home monitor center. Obviously, they are not isolated from each other. Delivering messages and monitoring behaviors are common activities over the network. As we mentioned 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, a 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 distribution and time synchronization.

In this subsection, we will combine the three aforementioned modules to form our final system. As shown in

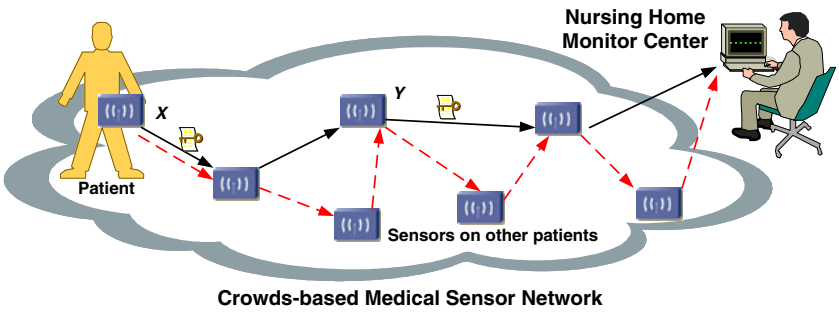


Fig. 5 Anonymity module

Fig. 6, the MSN in a temporal accountability module has the capability of anonymous communication. In order to better understand the functionality of this system, we will use an example to illustrate some implementation details.

As shown in Fig. 6, a packet delivered in the MSN has 5 fields: (1) sender's address; (2) destination's address; (3) sequence number; (4) timestamp; and (5) encrypted data. The data involves the sensor's ID and relevant medical information. The encryption key for data is the public key issued by the monitor center, namely K_M, and is held by every principal. Hence, field (5) cannot be modified except by the monitor center. We notice that every packet is encrypted by the sender's private key. It is only used for signature. The signature function can be used for identifying sender's identification. For example, suppose that a principal X delivers a message m to the monitor center, the package m will be formed like this: $[\mathrm{IP}_X, \mathrm{IP}_M, \{\mathrm{Seq}_S, \mathrm{TA}_X, \{\mathrm{Data}\}K_M\}K_X^{-1}]$. 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 see this message as [IP_X, IP_M, Seq_S, TA_X, $\{\mathrm{Data}\}K_M$]. Due to the Crowds-based forwarding strategy, the first field of the message will be replaced while passing through the intermediate nodes, e.g., principal Y in Fig. 6. Specifically, the package m becomes $[\mathrm{IP}_Y, \mathrm{IP}_M, \{\mathrm{Seq}_S, \mathrm{TA}_Y, \{\mathrm{Data}\}K_M\}K_Y^{-1}]$ after passing through principal Y. The fourth field also has been changed since the accumulative timestamp has been updated. In the same manner, this package (denoted as m') is encrypted by K_Y^{-1} for signature reason. Eventually, this message will arrive at the monitor center.

At the monitor center, received packets will be extracted for further processing. Fields (3), (4), and (5) are used for medical analysis. For instance, they can be utilized to reconstruct ECG pictures. Field (5) will be decrypted by 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 the local time of the monitor center. By using this value, together with the 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 secure log file for further surveillance. The log files can be used for auditing a certain principal for its temporal change during a long period. Such auditing is so called history-aware timestamp statistical analysis which can be a potential future work discussed in Section 6.

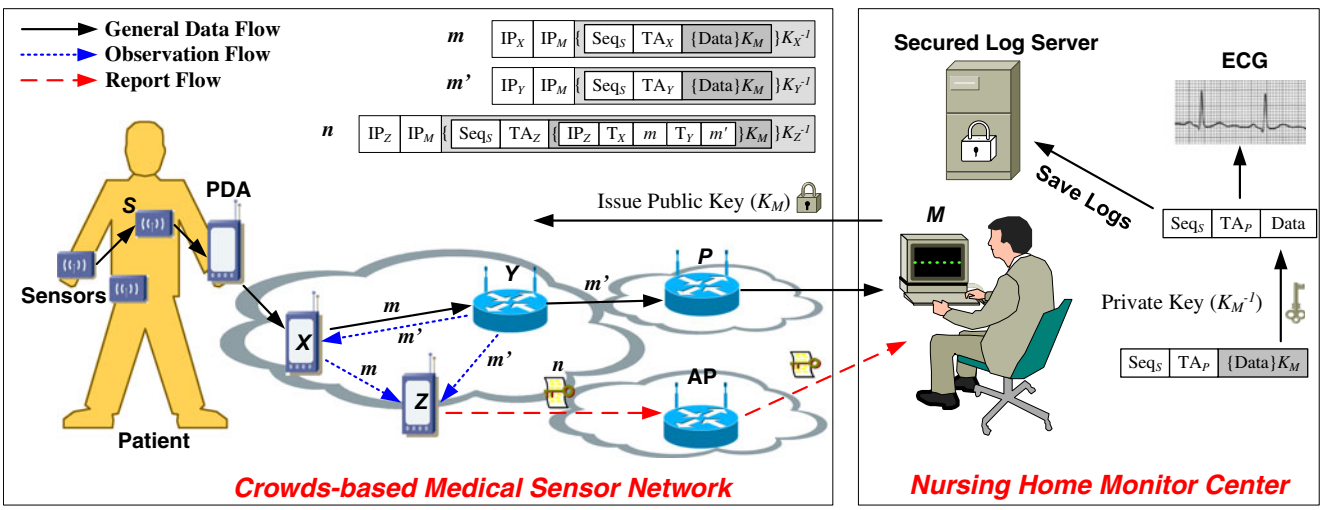


Fig. 6 Architecture of the MSN

The transmission protocol of our design only has three steps: a delivery step, an observation step, and a report step. In the delivery step, a sender asks some intermediate nodes (e.g., *Y* and *P*) to forward a message to the monitor center. In the observation step, some neighbors will monitor the message to see if its temporal signal has been changed. In the report step, the neighbor will deliver certain temporal evidence to the monitor center when it has found an abnormal modification.

We will explain the meaning of general flow step by step. In the delivery step, *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*. Notice that *m* has been changed to *m′* (message 2) after passing through *P*. Eventually, this message will be delivered to *M*. In the observation step, 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 local time T_X. Similarly, when *Y* sends *m′* to *P*, *Y*'s neighbor *Z* will receive the *m′* (message 4). Then *Z* will record the message 4 as [message 4, T_Y]. Of course, *X* can also be the observation node of *Y*. Therefore, X will also receive the message *m′* (message 5) and record it as [message 5, T_Y']. Notice that, T_Y is the local time of *Z* while T_Y' is the local time of *X*. In the report step, if *Z* has found that the 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 $\left[IP_Z, IP_M, \{Seq_S, TA_Z, \{IP_Z, T_X, m, T_Y, m'\}K_M\}K_Z^{-1}\right]$. As we can see, the report message *n* has the same structure as regular message *m*. The only difference is the data field of *m* has been changed to the temporal evidence {IP_Z, T_X, m, T_Y, m′} that only can be seen by the monitor center *M*.

4 Protocol analysis

We have already addressed the first three principals in Section 3. In the following two subsections, we will address principal **DP4**. In this subsection, we adopt the same analysis method as in [44]. It starts with defining temporal accountability goals. Then it will interpret traffic into logical descriptions. After that, the initial assumptions will be restated in a logical way. Based on the logic described in [44], we can eventually prove that our protocol can achieve all temporal accountability goals by using the traffic interpretation and the initial assumptions.

4.1 Temporal accountability goals

We present accountability goals for transmission protocol according to the definition stated in Section 3. Although the following goals are neither necessary nor sufficient, they seem reasonable as general goals.

G1: the monitor center *M* CanProve (*X sees* $m(t_i)$ *at* t_{x1})
G2: the monitor center *M* CanProve (*X modifies* $m(t_i)$ to $m(t_i')$)
G3: the monitor center *M* CanProve (*X says* $m(t_i')$) *at* t_{x2})

4.2 Traffic interpretation

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

1) *Y* Receives ({Seq_S, TA_X, {Data}K_M} SignedWith K_X^{-1})
2) *P* Receives ({Seq_S, TA_Y, {Data}K_M} SignedWith K_Y^{-1})
3) *Z* Receives ({Seq_S, TA_X, {Data}K_M} SignedWith K_X^{-1})
4) *Z* Receives ({Seq_S, TA_Y, {Data}K_M} SignedWith K_Y^{-1})
5) *A* Receives ({Seq_S, TA_Y, {Data}K_M} SignedWith K_Y^{-1})
6) *M* Receives ({Seq_S, TA_Z, {IP_Z, T_X, ({Seq_S, TA_X, {Data}K_M} SignedWith K_X^{-1}), T_Y, ({Seq_S, TA_Y, {Data}K_M} SignedWith K_Y^{-1})}K_M} SignedWith K_Z^{-1})

4.3 Initial assumptions

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))

4.4 Protocol analysis

- *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. *Y* then can prove the following statement by applying the accountability postulate [44].

  $$Y\ \text{CanProve}(X\ says(\{Seq_S, TA_X, \{Data\}K_M\})at\ T_{XY})$$

  This statement can be transformed by applying **A1**.

  $$Y\ \text{CanProve}(X\ delivers\ \text{Data}\ at\ T_{XY}\ \text{Timestamp With}\ TA_X)$$

  When *M* requests 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}, *P* can prove the following statement by applying the accountability postulate and **A1**.

$$P\,\text{CanProve}(Y\ delivers\ \text{Data}\ at\ T_{YP}\ \text{Timestamp With}\ TA_Y)$$

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 **N3** 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 *Z* can prove the following statement by applying the accountability postulate and **A1**.

$$Z\,\text{CanProve}(X\ delivers\ \text{Data}\ at\ T_X\ \text{Timestamp With}\ TA_X)$$

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*:
  Message 4 is similar with Message 3. By recording the Message 4, *Z* can prove the following statement by applying the accountability postulate and **A1.**

$$Z\,\text{CanProve}(Y\ delivers\ \text{Data}\ at\ T_Y\ \text{Timestamp With}\ TA_Y)$$

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*:
  Message 5 is also similar with Message 3. By recording the Message 5, *X* can prove the following statement by applying the accountability postulate and **A1.**

$$X\,\text{CanProve}(Y\ delivers\ \text{Data}\ at\ T_{YX}\ \text{Timestamp With}\ TA_Y)$$

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 subsection 2.3) 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 prove the following statement by using the temporal evidence generated by *Z*. *M* can request temporal evidences from relevant principals to prove the authenticity of *Z*. Therefore, we have:

$$M\,\text{CanProve}(X\ says(\{\text{Seq}_S, TA_X, \{\text{Data}\}K_M\})at\ T_X)$$
$$\text{and}\ (Y\ says(\{\text{Seq}_S, TA_Y, \{\text{Data}\}K_M\})at\ T_Y)$$

This statement can be transformed by applying **A2**.

$$M\,\text{CanProve}(Y\ modifies\ m(TA_X)\text{to}\ m(TA_Y)\text{at}\ T_Y)$$

This is the accountability goal **G2**.

5 Evaluation

In addition to the above analysis, we have also evaluated our system for its performance in temporal accountability, time accuracy, and scalability by using a discrete event simulation in Java environment.

We have only modeled wireless sensor and wireless AP scenarios. No wired connection is deployed. We also assume that no packet will be delivered over five hops to the destination. Therefore, we distribute all nodes into five blocks. The destination is connected to the fifth block. In practice, there are always several intermediate nodes in the middle of the transmission path, but few at the beginning and the last. Due to this reason, we assign the proportion of the nodes in each block as 1:2:4:2:1 in quantity in this simulation. For example, if there are ten nodes in the MSN, there will be one node in the first and fifth blocks, two nodes in the second and fourth blocks, and four nodes in the third block. Considering that more powerful wireless devices (like AP's) are 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 the time period from initially getting a message from the queue and the complete sending of this message to the uplink. Taking the message authentication time into consideration, the average service times for blocks 1 through block 5 are assigned as 25 ms, 20 ms, 15 ms, 10 ms, and 5 ms, respectively.

In order to simulate distinct local times for different wireless devices, we utilize the time-driven simulation method proposed in [55]. That is, the drift clock for each

device is subject to three factors: *offset*, *skew*, and *drift*. If the current system time is t, the drift clock $D(t)$ can be presented as formula (1):

$$D(t) = offset + skew \times t + drift \times t^2 \quad (1)$$

Therefore the local time $L(t)$ can be obtained by formula (2):

$$\begin{aligned} L(t) &= t + D(t) \\ &= offset + (skew + 1) \times t + drift \times t^2 \end{aligned} \quad (2)$$

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, and all *drift* values are uniformly distributed between −0.0002 and 0.0002.

For simplicity, we do not consider the propagation time of the wireless environment. Therefore, the propagation time is assumed to be zero. Moreover, the interval time of message arrival (the message is generated by itself, not by receiving) at each node is exponentially distributed. The mean interval time for each block is different. We suppose that the mean interval time for block 1 to block 5 are: 2 s, 4 s, 8 s, 16 s, and 32 s, respectively.

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 a Windows 7 64-bit, 2 GB RAM, Intel Core 2 6400, 2.13 GHz CPU.

5.1 Temporal accountability

We have already proved temporal accountability by using accountability logic in Section 4. Notice that the threshold directly affects the judgments of neighbor nodes in surveillance. A good threshold value is the key to making our system temporal accountable. In order to evaluate what is a good threshold, we set different threshold values and measure the average accuracy of detection for abnormal temporal signals.

In this simulation, we randomly set 10% of the packets as abnormal messages by manually increasing their time-stamps by 1 s. If there are x such packets in total and y ($0 < y < x$) packets have been detected by surveillance, the accuracy of detection is y/x. For threshold, we set the value between 1 ms (it is mean value of *skew*) and 100 ms (it is less than modified value 1 s). Obviously, the threshold cannot be too high. Otherwise, some undesirable temporal signals with minor changes may be ignored by the neighbors' surveillance. The threshold cannot be too small either, or the bias of the local clock may be regarded as abnormal behavior.

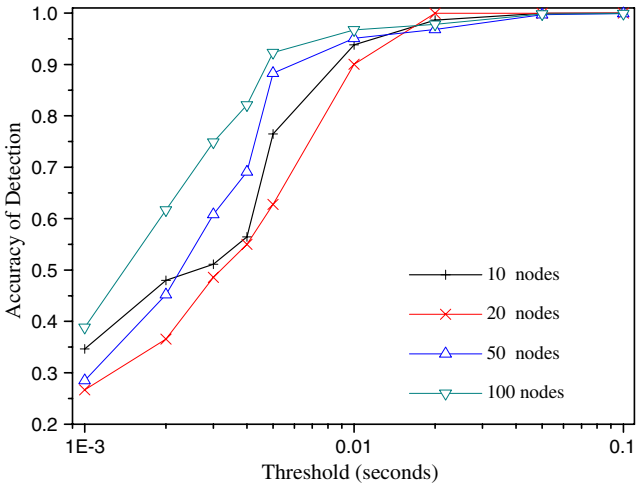


Fig. 7 Simulation results on threshold effect

As Fig. 7 depicts, with the increase in value of threshold from 1 ms to 100 ms, the accuracy of detection gradually approaches 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 are more likely to be detected by more chance. Thus, the accuracy of detection should be high. Otherwise, they may be in conspiracy and not detected by limited neighbors. The accuracy of detection would be low at this time.

From the analysis and simulation results, we can see that the threshold value should be greater than the mean value of *skew* and less than the abnormal modified value. Here, the mean value of *skew* is 1 ms and the modified value is 1 s; then the threshold can be select from 1 ms to 1 s. But we need avoid unnecessary fluctuation to obtain a better accuracy of detection. Therefore, the threshold should be the mean value of the above range, which is 500 ms. Hence, a good threshold for a general purpose should be set as the mean value of the range from average *skew* (in microseconds) to the trivial abnormal modified value (in milliseconds).

5.2 Time accuracy

As illustrated in Fig. 8, with the increase in the number of hops, the average error experiences linear growth in milliseconds. The fluctuation of this line can be explained by the random distribution of local drift times. Since every hop increases the chance of time drifting of the system time, the fact that the average error increases along with the hop number is a normal behavior. Fortunately, in an MSN environment, the hop number will not reach 10 and all local drift times are too trivial to affect medical

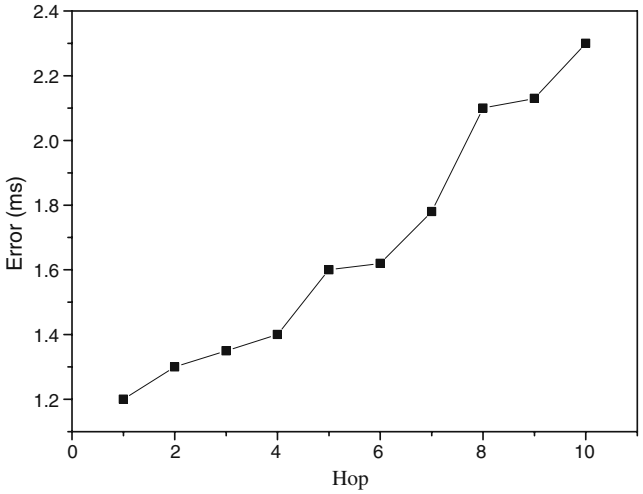


Fig. 8 Simulation results on hop effect

information. This result indicates that our protocol has average errors between 1.2 ms and 2.3 ms. This is acceptable in an MSN system.

5.3 Scalability

As illustrated in Fig. 9, with the increase in the number of nodes, the average delay experiences logarithmic growth in milliseconds which is less than 4.5 ms. This result indicates that our protocol is scalable.

6 Future challenges

One possible future expansion is to research location tracking. In the nursing home context, for instance, although the patient's room number may be revealed via revocable anonymous technology, it is not an appropriate way to seek the dynamic location of a mobile patient. In practice, most of the reveal procedure needs an additional party for surveillance. The nursing home monitor center cannot easily carry out this task alone. Also, it must be a quick action with no complex computing involved. How to decrease medical response time and obtain the accurate location of the patient, especially in emergency situations, is a challenge for future research.

Another major concern is that the monitor center is usually controlled by system administrators who have the authority to access and change the stored audit data [3]. Typically, we use a TTP to supervise the MSN's and nursing homes. Who will be the TTP is an open question for future work. However, using TTP to do surveillance may be impossible in the real world. On the one hand, it is time consuming. On the other hand, it generates more overhead with each packet the system delivers. This is therefore an energy consuming problem. Hence, how to ensure the temporal accountability and anonymity in monitor center without TTP surveillance is a significant issue we should consider.

In addition, the role of history of the temporal records can be used for future analysis. Current work may be good to capture sudden timestamp anomalies (i.e., with large timing changes). However, a smart MSN attacker may keep sending the slowly-change irregular timestamp records for certain periods of time. They will not send obviously abnormal clock information in order to avoid easy detection. For these evolution-style timing attacks, we need to resort to cumulative, history-aware timestamp statistical analysis to detect them.

The patient's PDA may directly connect to the Internet by using 3G or Wi-Fi technologies. It is unnecessary to deliver medical information hop-by-hop among fixed AP's. Nevertheless, our design cannot guarantee temporal accountability in Internet transmission. In order to enable scalability in MSN's, future work should address this issue.

7 Conclusion

This paper mainly addressed temporal accountability and anonymity issues in MSN systems. By using mutual surveillance in a heterogeneous wireless environment, the system that we proposed is capable of detecting abnormal temporal signals and identifying the root causes of such events. By adopting the Crowds anonymous communication system, our design can also well protect the sender's identification of each transmitted message. Logical analysis and simulation results indicate that, if every two wireless communication principals have a

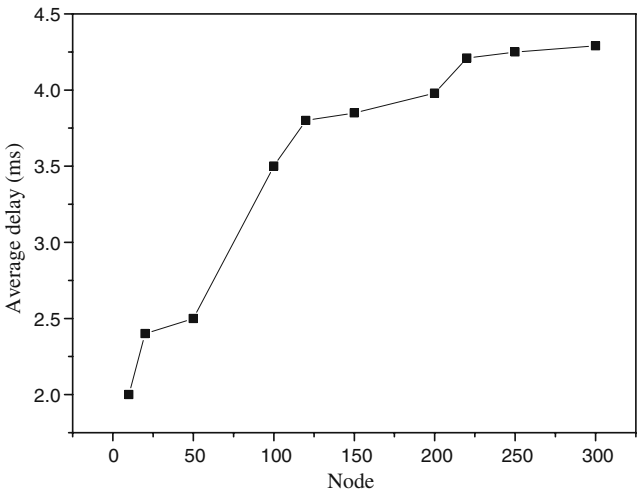


Fig. 9 Simulation results on scalability

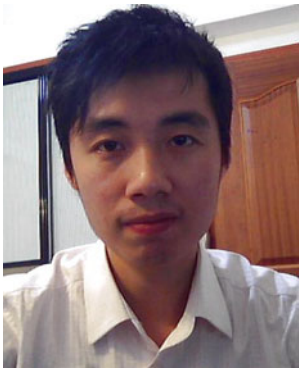
mutual neighbor for surveillance, our MSN system can make the majority of the temporal signal accountable. At the same time, all the temporal signals can be synchronized with high precision (millisecond) at the monitor center. In addition, our design is a scalable approach that can be deployed into any other wireless communication system for temporal accountability objectives.

Acknowledgement This work is supported in part by the US National Science Foundation (NSF) under the grant numbers CNS-0737325, CNS-0716211, and CCF-0829827, as well as RGC (Research Grants Committee) award at The University of Alabama 2008.

References

1. Wikipedia (2009) Heart disease. http://en.wikipedia.org/wiki/Heart_diseases
2. Hu F, Celentano L, Xiao Y (2008) Error-resistant RFID-assisted wireless sensor networks for cardiac telehealthcare. Wireless Comm Mobile Comput 9(1):85–101
3. Ferreira A, Shiu S, Baldwin A (2003) Towards accountability for electronic patient records. In: Proceedings of the 16th IEEE Symposium on Computer-Based Medical Systems, (CBMS'03), New York, USA, pp 189–194, Jun 2003
4. Bhattacharya S, Paul R (1999) Accountability issues in multihop message communication. In: IEEE Symposium on Application-Specific Systems and Software Engineering and Technology (ASSET'99), Richardson, USA, pp 74–81, May 1999
5. Reiter MK, Rubin AD (1998) Crowds: anonymity for web transactions. ACM Trans Inf Syst Secur 1(1):66–92
6. Xiao Y (2006) Editorial. Int J Secur Netw 1(1/2):1
7. Shehab M, Bertino E, Ghafoor A (2006) Workflow authorisation in mediator-free environments. Int J Secur Netw 1(1/2):2–12
8. Jung E, Gouda MG (2006) Vulnerability analysis of certificate graphs. Int J Secur Netw 1(1/2):13–23
9. Kiayias A, Yung M (2006) Secure scalable group signature with dynamic joins and separable authorities. Int J Secur Netw 1(1/2):24–45
10. Franklin M (2006) A survey of key evolving cryptosystems. Int J Secur Netw 1(1/2):46–53
11. Hamadeh I, Kesidis G (2006) A taxonomy of internet traceback. Int J Secur Netw 1(1/2):54–61
12. Jhumka A, Freiling F, Fetzer C, Suri N (2006) An approach to synthesise safe systems. Int J Sec Netw 1(1/2):62–74
13. Evans JB, Wang W, Ewy BJ (2006) Wireless networking security: open issues in trust, management, interoperation and measurement. Int J Secur Netw 1(1/2):84–94
14. Englund H, Johansson T (2006) Three ways to mount distinguishing attacks on irregularly clocked stream ciphers. Int J Secur Netw 1(1/2):95–102
15. Zhu B, Jajodia S, Kankanhalli MS (2006) Building trust in peer-to-peer systems: a review. Int J Secur Netw 1(1/2):103–112
16. Ramkumar M, Memon N (2006) Secure collaborations over message boards. Int J Secur Netw 1(1/2):113–124
17. Xiao Y, Jia X, Sun B, Du X (2006) Editorial: security issues on sensor networks. Int J Secur Netw 1(3/4):125–126
18. Wang H, Sheng B, Li Q (2006) Elliptic curve cryptography-based access control. Int J Secur Netw 1(3/4):127–137
19. Zheng J, Li J, Lee MJ, Anshel M (2006) A lightweight encryption and authentication scheme for wireless sensor networks. Int J Secur Netw 1(3/4):138–146
20. Al-Karaki JN (2006) Analysis of routing security-energy trade-offs in wireless sensor networks. Int J Secur Netw 1(3/4):147–157
21. Araz O, Qi H (2006) Load-balanced key establishment methodologies in wireless sensor networks. Int J Secur Netw 1(3/4):158–166
22. Deng J, Han R, Mishra S (2006) Limiting DoS attacks during multihop data delivery in wireless sensor networks. Int J Secur Netw 1(3/4):167–178
23. Hwu J, Hsu S, Lin Y-B, Chen R-J (2006) End-to-end security mechanisms for SMS. Int J Secur Netw 1(3/4):177–183
24. Wang X (2006) The loop fallacy and deterministic serialisation in tracing intrusion connections through stepping stones. Int J Secur Netw 1(3/4):184–197
25. Jiang Y, Lin C, Shi M, Shen X (2006) A self-encryption authentication protocol for teleconference services. Int J Secur Netw 1(3/4):198–205
26. Owens SF, Levary RR (2006) An adaptive expert system approach for intrusion detection. Int J Secur Netw 1(3/4):206–217
27. Chen Y, Susilo W, Mu Y (2006) Convertible identity-based anonymous designated ring signatures. Int J Secur Netw 1(3/4):218–225
28. Teo J, Tan C, Ng J (2006) Low-power authenticated group key agreement for heterogeneous wireless networks. Int J Secur Netw 1(3/4):226–236
29. Tan C (2006) A new signature scheme without random oracles. Int J Secur Netw 1(3/4):237–242
30. Liu Y, Comaniciu C, Man H (2006) Modelling misbehaviour in ad hoc networks: a game theoretic approach for intrusion detection. Int J Secur Netw 1(3/4):243–254
31. Karyotis V, Papavassiliou S, Grammatikou M, Maglaris V (2006) A novel framework for mobile attack strategy modelling and vulnerability analysis in wireless ad hoc networks. Int J Secur Netw 1(3/4):255–265
32. Haeberlen A, Kouznetsov P, Druschel P (2007) PeerReview: practical accountability for distributed systems. ACM SIGOPS Operating Systems Review 41(6):175–188
33. Ting TC (1999) Privacy and confidentiality in healthcare delivery information system. In Proceedings of the 12th IEEE Symposium on Computer-Based Medical Systems. Stamford, U.S.A., June 1999, pp 2–4
34. Xiao Y, Takahashi D, Hu F (2007) Telemedicine usage and potentials. In: IEEE Wireless Communications and Networking Conference (WCNC'07), Hong Kong, China, Mar 2007, pp 2736–2740
35. Liszka KJ, York DW, Mackin MA, Lichter MJ (2004) 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
36. Shnayder V, Chen B, Lorincz K, Fulford-Jones TRF, Welsh M (2005) Sensor networks for medical care. Technical Report TR-08-05, Division of Engineering and Applied Sciences, Harvard University
37. Hu F, Jiang M, Xiao Y (2007) Low-cost wireless sensor networks for remote cardiac patients monitoring applications. Wireless Comm Mobile Comput 8(4):513–529
38. Xiao Y (2009) Flow-net methodology for accountability in wireless networks. IEEE Netw 23(5):30–37
39. Xiao Z, Xiao Y (2010) PeerReview analysis and re-evaluation for accountability in distributed systems or networks. Proceedings of The 4th International Conference on Information Security and Assurance (ISA2010), CCIS 76, pp 149–162
40. Xiao Z, Xiao Y (2010) P-accountable networked systems. In: Proceeding of INFOCOM 2010, Work in Progress (WIP) Track, accepted

41. Xiao Z, Xiao Y, Wu J (2010) A quantitative study of accountability in wireless multi-hop networks. Proceedings of 2010 39th International Conference on Parallel Processing (ICPP 2010), accepted
42. Xiao Y (2008) Accountability for wireless LANs, ad hoc networks, and wireless mesh networks. IEEE Communication Magazine, special issue on Security in Mobile Ad Hoc and Sensor Networks, 46(4), pp 116–126, Apr
43. Syverson PF (1993) Adding time to a logic of authentication. In: Proceedings of the 1st ACM Conference on Computer and Communications Security. Fairfax, Virginia, U.S.A, pp 97–101
44. Kailar R (1996) Accountability in electronic commerce protocols. IEEE Trans Softw Eng 22(5):313–328
45. Liang J, Ao Q, You J (2002) Analyzing the temporal accountability of secure protocols. Chin J Electron (Acta Electronica Sinica) 30(10):1451–1454
46. Kudo M (1998) Electronic submission protocol based on temporal accountability. In Proceedings of the 14th Annual Computer Security Applications Conference, 1998, pp 353–363
47. Meng B, Zhang H (2005) Research on accountability in electronic transaction. In Proceedings of the 9th International Conference on Computer Supported Cooperative Work in Design, 2005, pp 745–749
48. Galloway M, Zhang Y, Xiao Y, Shao P (2010) Time synchronization in sensor networks and underwater sensor networks. Underwater Acoustic Sensor Networks, CRC Press, ISBN-10: 1420067117, ISBN-13: 978-1420067118, 2010, Chapter 6, pp 143–175
49. Chen H, Xiao Y, Hong X, Hu F, Xie J (2008) A survey of anonymity in wireless communication systems. Secur Comm Network 2(5):427–444
50. Burrows M, Abadi M, Needham R (1990) A logic of authentication. ACM Trans Comput Syst 8(1):18–36
51. Abadi M, Tuttle MR (1991) A semantics for a logic of authentication. In: Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing. Aug, pp 201–216
52. Stubblebine SG (1995) Recent-Secure authentication: enforcing revocation in distributed systems. 19th IEEE Symposium on Research in Security and Privacy, pp 224–235
53. Stubblebine SG, Wright RN (1996) 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
54. Xu N, Rangwala S, Chintalapudi KK, Ganesan D, Broad A, Govindan R, Estrin D (2004) 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
55. Quan Y, Liu G (2008) 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



Jing Liu is a PhD student in the Department of Computer Science at The University of Alabama. He is an active researcher in the area of network security, bio-inspired network and telemedicine, including botnet issues, visual attention, anonymous communication and accountability in telemedicine. He received his B.Sc and M.Sc degrees from the Hunan University (China) in 2005 and 2008, respectively.

Dr. Yang Xiao worked in industry as a MAC (Medium Access Control) architect involving the IEEE 802.11 standard enhancement work before he joined Dept. of Computer Science at The Univ. of Memphis in 2002. He is currently with Dept. of Computer Science (with tenure) at The Univ. of Alabama. He was a voting member of IEEE 802.11 Working Group from 2001 to 2004. He is an IEEE Senior Member. He serves as a panelist for the US National Science Foundation (NSF), Canada Foundation for Innovation (CFI)'s Telecommunications expert committee, and the American Institute of Biological Sciences (AIBS), as well as a referee/reviewer for many national and international funding agencies. His research areas are security, communications/networks, robotics, and telemedicine. He has published more than 160 refereed journal papers (including over 45 IEEE transactions papers) and over 200 refereed conference papers and book chapters related to these research areas. Dr. Xiao's research has been supported by the US National Science Foundation (NSF), U.S. Army Research, The Global Environment for Network Innovations (GENI), Fleet Industrial Supply Center-San Diego (FISCSD), and The University of Alabama's Research Grants Committee. He currently serves as Editor-in-Chief for International Journal of Security and Networks (IJSN) and International Journal of Sensor Networks (IJSNet). He was the founding Editor-in-Chief for International Journal of Telemedicine and Applications (IJTA) (2007–2009).