Trustworthy Data Collection From Implantable Medical Devices Via High-Speed Security Implementation Based on IEEE 1363

Fei Hu, *Member, IEEE*, Qi Hao, *Member, IEEE*, Marcin Lukowiak, *Member, IEEE*, Qingquan Sun, Kyle Wilhelm, Stanisław Radziszowski, and Yao Wu

***Abstract*—Implantable medical devices (IMDs) have played an important role in many medical fields. Any failure in IMDs operations could cause serious consequences and it is important to protect the IMDs access from unauthenticated access. This study investigates secure IMD data collection within a telehealthcare [mobile health (m-health)] network. We use medical sensors carried by patients to securely access IMD data and perform secure sensor-to-sensor communications between patients to relay the IMD data to a remote doctor's server. To meet the requirements on low computational complexity, we choose N-th degree truncated polynomial ring (NTRU)-based encryption/decryption to secure IMD–sensor and sensor–sensor communications. An extended matryoshkas model is developed to estimate direct/indirect trust relationship among sensors. An NTRU hardware implementation in very large integrated circuit hardware description language is studied based on industry Standard IEEE 1363 to increase the speed of key generation. The performance analysis results demonstrate the security robustness of the proposed IMD data access trust model.**

***Index Terms*—Implantable medical devices (IMDs), industry Standard IEEE 1363, medical security, NTRU, trust model.**

I. Introduction

HEALTHCARE cost is a large budget percentage in many countries. For example, the U.S. healthcare spending was about $7421 per resident in 2007 and accounted for 16.2% of the national gross domestic product [1]. One of the most efficient ways to reduce healthcare labor cost is to use medical sensors to build a patient monitoring platform, which is called a telehealthcare system [2]. In addition to medical sensors, implantable medical devices (IMDs) have become an important approach to monitor and treat physiological conditions in patients' organs. Many different types of IMDs such as pacemakers, insulin pump, and brain neurostimulators can be used for a series of critical medical purposes including cardiac arrhythmia, diabetes treatment. It was estimated that U.S. citizens used over 25 million IMDs already for life-critical functions [3].

It is important to guarantee the data access security via low-complexity schemes for the IMDs because of the following.

1) IMDs are implanted in patients' organs. Unlike regular medical sensors, those IMDs are so close to organs that any small change in their control parameters could threaten the patient's life. For instance, a pacemaker cannot be stopped in order to activate heartbeats regularly.
2) IMD security is a governmental rule in many countries. For example, U.S. Department of Health and Human Services issued patient privacy protections as part of the Health Insurance Portability and Accountability Act of 1996. Most health insurers, pharmacies, doctors, and others are required to comply with these federal standards [4].

While there exist several secure, well documented, asymmetric algorithms, most of them [such as Rivest, Shamir, and Adleman (RSA)] require large amounts of memory and significant computation time. We propose to use a very efficient, low overhead, public key encryption algorithm to support a high level of security. Such an algorithm is NTRU [5], [6]. In addition, there is a need for building a robust trust model and computing quantitative trust relationships among sensors and IMDs.

Our contributions reported in this paper include the following.

1) Hardware-oriented NTRU design and NTRU speed optimization in medical signal transmission. Real-time sensor data authentication and intrusion detection are expected with low complexity and energy consumption in medical sensor network system, where the stream decryption time cannot go beyond 100 μs [7]. This paper presents a series of optimizations in the NTRU circuit design to achieve a high operation speed with low power dissipation.
2) Integration of NTRU with an indirect/direct trust model. Our initial study on the possibility of using NTRU-based algorithms to achieve medical security has generated some preliminary results [8]–[11]. In this study, we significantly extend our previous research by closely integrating IMD–sensor indirect/direct trust model with NTRU hardware implementation to achieve comprehensive m-health IMD data collections anywhere and anytime.
3) Comprehensive and quantitative performance analysis on NTRU industry standard implementation and trust-based IMD/sensor security. We have evaluated our NTRU hardware design performance under the industry Standard

Manuscript received January 29, 2010; accepted January 29. 2010. Date of publication April 26, 2010; date of current version November 5, 2010.

F. Hu, Q. Hao, Q. Sun, and Y. Wu are with the Electrical and Computer Engineering, The University of Alabama, Tuscaloosa, AL 35487-0286 USA (e-mail: fei@eng.ua.edu; qh@eng.ua.edu; qsun3@bama.ua.edu).

M. Lukowiak and K. Wilhelm are with the Computer Engineering, Rochester Institute of Technology, Rochester, NY 14623-5603 USA (e-mail: mxleec@rit.edu).

S. Radziszowski is with the Department of Computer Science, Rochester Institute of Technology, Rochester, NY 14623-5603 USA (e-mail: spr@cs.rit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITB.2010.2049204

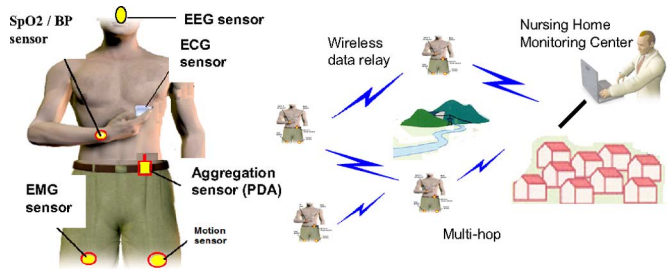


Fig. 1. Wireless sensor networks for m-health.

(a) (b) (c)

Fig. 2. (a) RF mote (radio chip). (b) ECG sensor. (c) EEG sensor.

IEEE 1363. Especially, we have tested the energy efficiency and storage overhead performance for both trust models and NTRU security schemes.

Rest of the paper is organized as follows. Section II describes the system architecture of IMD/sensor network and its security issues. Section III gives the sensor/IMD trust establishment model. Section IV presents the NTRU-based chip design procedure in Hardware Description Language (HDL) based on industry Standard IEEE 1363. The performance analysis of the IMD/sensor security system is given in Section V. Section VI concludes this paper and gives an outline of future research.

II. IMD Access Security in Telehealthcare Networks

A. Mobile-Health (m-Health) Architecture

We have built a medical sensor network (MSN) that consists of a body area network and a WAN [12]. A patient can carry medical sensors such as ECG, electromyogram, EEG, and SpO2 sensors that monitor heart, muscle, and brain activities, and oxygen saturation level. An aggregation sensor (supersensor), as shown in Fig. 1, is used for data integration. The MSN can be applied in large nursing homes due to its scalable routing schemes.

The developed MSN hardware components include the following.

1) *Radio chip (RF motes):* Ember CPU-RF chips were used to build RF motes driven by AA batteries, as shown in Fig. 2(a). The heart of the RF board is the micro central unit/ZigBee transceiver unit.
2) *ECG sensor:* A three-lead ECG sensor was built with accurate heartbeat pattern capture. Fig. 2(b) shows the ECG sensor. Its size is a little larger than 2 AA batteries.
3) *EEG sensor:* A low-cost, portable, wireless EEG sensor platform was developed that includes two primary elements: the analog board (EEG sensor), as shown in Fig. 2(c), and the digital board (i.e., RF board). Two electrodes collect the EEG channel data from the brain and one electrode goes to the right leg. A electrostatic discharge circuit is used for chip protection and user safety.

B. IMD Data Access Security

Since each IMD has an extremely tiny antenna and a low-power RF transceiver, it cannot send out data wirelessly for a long distance (say, 100 m, a typical RF range of medical sensors). Actually, the typical IMD-to-reader communication distance in a hospital is less than 2 m. Therefore, we propose to use a medical sensor carried by the patient to read the IMD data 1–2 m of distance. An IMD (with batteries, an RF antenna, and a microcontroller) can use the one-hop wireless communication to reach a body sensor. The sensor also includes a RF chip for communications with other sensors. However, the communication frequency in sensor-to-sensor links (typically 2.4 GHz) is different from IMD-to-sensor ones (833 MHz or other unlicensed short-distance RF frequency) to avoid RF interference. The sensor also has a capability of collecting medical parameters (such as heartbeat pattern) of the patient.

III. Trust Model Across IMDs and Sensors

A. Trust Model

Trust measures the security level of a wireless communication entity (here is medical sensor). It is important to establish an accurate trust model for the IMD-to-sensor/sensor-to-sensor communication chain because of the following.

1) An adversary can fake to be a legal sensor. If a faked sensor knows the IMD access channel (RF frequency) and modulation scheme, it can easily read the IMD data.
2) An adversary's sensor can misroute the IMD data to make it unable to reach the doctor's server.
3) Trust level cannot be simply represented by a Boolean ("1" for "trustworthy" and "0" for "untrustworthy"), since sometimes a sensor may have mechanical failure or just circuit/RF noise. We cannot simply assign a "0" to those temporarily failed sensors. Therefore, a real number within a range (such as [0, 1]) is needed to describe the trust level of a sensor.

The establishment of a trust relationship depends upon both intrinsic properties and contextual properties of the system. Intrinsic properties, such as the propensity to take risks, the benefits of engaging in a trust relationship, and the personal cost of breaking trust, are defined as the factors that are internal to the trustor and trustee. Generally the trust calculation should consider the following.

1) Direct trust, which is obtained from the direct connection between the source and the target. The sensor that wants to compute its trust value to another one is defined as a source. On the other hand, the sensor that source wants to interact with is defined as a target.
2) Indirect trust that is based on the information provided by other sensors that had experiences in transactions with the target in the past.

In the IMD/sensor communication scenario, the trust model should fit in a multiring network topology: an IMD needs to

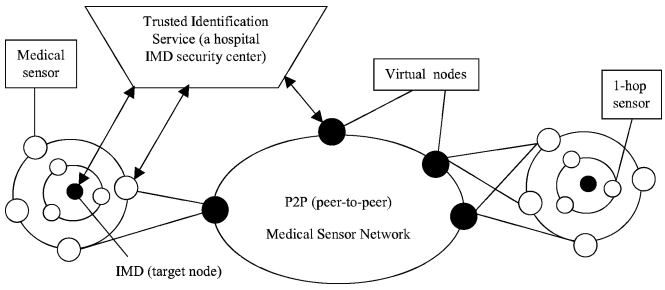


Fig. 3. Matryoshka-based trust model (in IMD/sensor scenario).

evaluate the direct trust with all of its body sensors, which form a one-hop ring, since any of them could collect IMD data. Between patients, there is a trust relationship across different rings. When a sensor selects a next sensor to relay its data, it needs to select a sensor with the largest trust value that is the sum of direct trust and indirect trust values.

B. Matryoshka-Based Trust Model

Conventional trust models cannot reflect such a ring-to-ring trust relationship. An exception is the matryoshka-based trust establishment model that has been proposed in [13] to solve the social network security issues, where complex hierarchical human relationship exists. It can describe the trust relationship between different social levels (such as employers versus employees). The matryoshka approach uses the multihop routing among cooperative sensors. However, the matryoshka approach has a major drawback: it cannot distinguish between direct trust (from one-hop communication entities) and indirect trust (from other remote neighbor's trust recommendations).

This study significantly extends the matryoshka trust scheme by introducing quantitative direct/indirect trust models. We first assume that each medical sensor is uniquely identified by a pseudonym and a sensor identifier. As shown in Fig. 3, the IMD/sensor trust relationship consists of three components that correspond to the entities of a matryoshka system in different hierarchical levels:

1) *Matryoshkas:* The matryoshkas is defined as a hierarchical structure of relationship between a core node and other trusted nodes on concentric rings. The innermost ring consists of a set of nodes, which are trusted by the owner of the matryoshkas (an IMD or a medical sensor). The second ring consists of a set of nodes, which are trusted by the nodes in the first ring. Other rings are established according to the same rule. It is not necessary for nodes on the same ring to trust each other except for the first ring. The messages go through the concentric rings from an innermost node to an outermost node. Each node establishes its matryoshka and keeps updating it. For instance, a new patient's IMD/sensors may join the network. The privacy is preserved based on the hop-by-hop trust relationships.
2) *Peer-to-peer network:* According to user identifiers, the peer-to-peer network, as shown in Fig. 3(center), provides the global access to its data. Each node in the peer-to-peer substrate is arranged in a distributed hash table (DHT) associating with the distributed system. The pseudonym of each node is used to identify its location in the DHT according to the DHT protocol. Thus, the location data include the pointers to nodes on the outermost ring of the requested user's matryoshka. The node in peer-to-peer substrate works as an entry to access the information of the target node. In our case, a medical network may use some servers deployed in different places to manage patients' sensors in the neighborhood. Those servers form the peer-to-peer network.
3) *Trusted identification service:* It is provided by a medical security control center. Such a center manages sensor/IMD network IDs, and stores their trust levels. Each node gets a unique pseudonym, a unique node identifier and two certificates for the authentication of each type of identifiers from the trusted identification service. The pseudonym is used as an identifier in the peer-to-peer system, and the node identifier is used to identify a member of the network. Such a mechanism leads to the protection of Sybil attacks, impersonation attacks, and attacks on the DHT overlay.

C. Implementation of Matryoshka-Based Trust Model

Matryoshkas uses straightforward public key cryptography (a NTRU-based scheme in this study) in order to realize the privacy preservation. Each node has a set of properties (forming a set N) such as the pseudonym and the node identifier. It generates two key pairs: I and P . The identification service certifies the authenticity of I and P to encrypt the pseudonym P and the node identifier, respectively. The relationship among I and P cannot be inferred, expect for the trusted nodes of a user. The matryoshkas trust model provides the following operations in order to realize the security service in IMD/sensor networks.

First, a trust-oriented network account is created by an invitation initiated from a user (patient) u to a different user v; u already exists in the system and v wants to take part in the network system. There are four phases for account creation.

1) Identity creation.
   a) v creates the two key pairs I and P.
   b) v sends a request to u for obtaining pseudonym, node identifier and certificates.
   c) u relays this request to the trusted identification service based on the DHT.
   d) The trusted identification service derives node's pseudonym $P_v = h_1(N)$ and its node identifier $v = h_2(N)$ from node's properties *N*, where (h_1, h_2) are two cryptographic hash functions. It also grants two certificates $\{I^+, v\}_{S_{TTP}}$ and $\{P^+, P_v\}_{S_{TTP}}$, where S_{TTP} is the signature of the identification service.
   e) Once u receives the response from the identification, it will relay the response to v.
2) Joining the P-to-P network.
   a) According to the received certifications, v joins the P-to-P substrate using u as a bootstrapping host and P_v as its pseudonym.

3) Creation of the profile.
   a) v can independently generate its profile consisting of several attributes for each entry, and generates public key pairs, which it signs with I^+, for each attribute in order to share it with preferred users.
   b) Each attribute is encrypted with its respective private key. The *friend list* is an important attribute. v retrieves the name attribute from its contacts like u in their encrypted form and lists these as the friend list, finally encrypted with its own respective key.
   c) A user is able to access the profile only if it is admitted by the nodes in the chain from the outmost to the innermost ring in the matryoshka.
4) Matryoshka creation.
   a) Initially v only knows u.
   b) v stores its encrypted profile in u.
   c) v sends a request to register to DHT and a time-to-live counter ttl to u, that is, $E_{P_u}\{M_{vu}, \text{ttl}\}$ with $M_{vu} = \{k, v, P_u, \{I_v^+, v\}_{S_{TTP}}\}_{S_{Iv}}$, where k is the lookup key for the DHT.
   d) Once u receives the message from v, it selects a node from its contact list arbitrarily, for example, w and encapsulates M_{vu}, then sends it together with the decreased counter ttl$'$ to w, that is, $E_{P_w}\{M_{vw}, \text{ttl}'\}$ with $M_{vw} = \{k, P_u, P_w, \{P_u^+, P_u\}_{S_{TTP}}, M_{vu}\}_{S_{Pu}}$.
   e) Repeat d) recursively until the ttl expires, where ttl is set according to the requirement on the number of the rings in a matryoshka.
   f) Once the message reaches the outermost ring, the node will register the key and authenticate it according to the chain of encapsulated signatures.

Second, we extend the matryoshka model by considering the calculation of direct trust and indirect trust. Indirect trust is reflected by the recommendation that is obtained from the other node's transactions with the recommended node, where a node could be an IMD or a medical sensor. A source can query other nodes to get the recommendation of the target IMD according to the network characteristics. A typical direct trust could be given by [14]

$$\text{Dr}_{ij} = \frac{\text{Suc}_{ij}}{N_{ij}} \tag{1}$$

where Suc_{ij} denotes the number of successful transactions between node i and node j, N_{ij} denotes the total number of data exchanges (communication transactions). Then, the value of recommendation from node i to node j is given by

$$T_{ij} = \frac{\text{Suc}_{ij} - (N_{ij} - \text{Suc}_{ij})}{\sum_k \text{Suc}_{ik}}. \tag{2}$$

The indirect trust is given by

$$\text{Ir}_{ij} = \frac{\sum_{i=1}^{k} R_{ij}(T_{ij} + \lambda)}{k} \tag{3}$$

where R_{ij} denotes reputation [14] from node i to node j. It decreases exponentially due to the punishment for the false recommendation and increases linearly as the recommendation is accurate. λ denotes the adjustment coefficient to ensure that the network is stable. Thus, the overall trust is given by

$$\text{Trust}_{ij} = \alpha \text{Dr}_{ij} + (1 - \alpha)\text{Ir}_{ij}. \tag{4}$$

D. Improved Direct Trust and Indirect Trust

However, the model in [15] is simple and just considers a few factors that affect trustworthiness. For instance, it does not consider the recommender's reputation in indirect trust calculation, which gives an adversary the opportunity to use man-in-the-middle attack. Therefore, we make the following improvement based on a comprehensive indirect model used in the social network [14].

The improved estimation of direct trust is given by

$$\begin{aligned}\text{Dr}_{ij} = {} & \text{dr}_{t-1} t + M(i,j)L(i,j)w(L(i,j))F(j) \\ & + K(j)f(x)P(x)F(y)\end{aligned} \tag{5}$$

where dr_{t-1} is the accumulative direct trust values of node j until $t-1$ times. $M(i,j)$ is the satisfaction degree from node i to node j. $L(i,j)$ is the shortest length between node i and node j. $F(j)$ is the risk that node j takes in the transaction. $F(y)$ is the transaction risk of node y which is recommended by node j. If node j is the member of the clique, $K(j)$ equals to 1, otherwise 0. If the recommendation is false, $f(x)$ equals to -0.5, otherwise 0. $P(x)$ is the recommendation punishment. The trust decay time is given by

$$t = 1 - L(i,j)M(i,j). \tag{6}$$

The weight of the link is given by

$$w(L(i,j)) = \exp(-L(i,j)). \tag{7}$$

The recommendation punishment is given by

$$P(x) = \frac{1}{[1 + \exp(-n)]} \tag{8}$$

where n is the number of false recommendation.

The improved estimation of indirect trust is given by

$$\begin{aligned}\text{Ir}_{xj} = {} & \text{Ir}_{t-1} t \\ & + \frac{1}{N}\sum_{i=1}^{N} \frac{C_{xi}}{C_{xi} + R_{xi}} L(x,j)w(L(x,j))M(x,j)\end{aligned} \tag{9}$$

where N is the number of recommendation nodes. $C_{xi}/(C_{xi} + R_{xi})$ is the trust degree of recommendation nodes. The expression of the overall trust is the same as (4).

The aforementioned model can recognize malicious behaviors such as boast and cheating in networking. Based on the small world theory, they can efficiently handle relationships such as establishing new connections and terminating old connections.

IV. Light-Weight, High-Speed Cipher Implementation

A. NTRU Cipher and Industry Standard: IEEE 1363.1

The IEEE 1363.1 draft standard for public-key cryptographic techniques provides a central reference for public-key techniques when applying NTRU algorithms [5]. In order to support

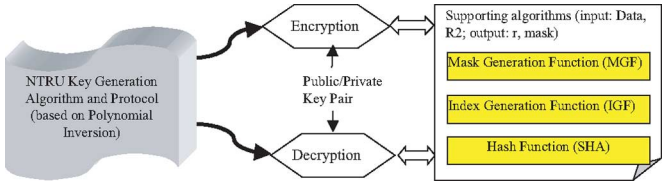


Fig. 4. IEEE 1363.1 standard components.

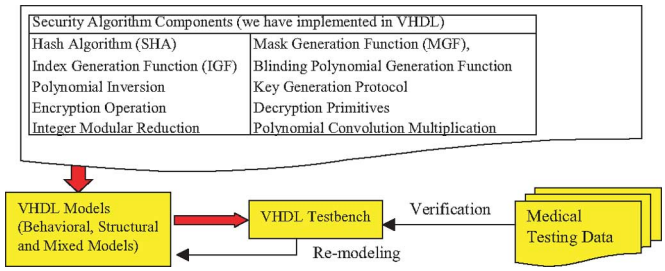


Fig. 5. VHDL model of IEEE 1363.1 implementation.

the functions of key generation, encryption, and decryption, IEEE 1363.1 defines a mask generation function (MGF), index generation function (IGF), and a blinding polynomial generation method (BPGM), which are based on an underlying harsh function. The MGF is used to ensure a reasonably random distribution of bits after encryption and to ensure that a single bit of the output is dependent on multiple input bits. From a security standpoint, the encrypted text will require a much larger search space to discover and should be less vulnerable to attack if it has been sufficiently randomized. An IGF, similar to the MGF except that it is state aware and can, therefore, be called multiple times, is used to provide a source of reasonably random indexes. The output indexes from IGF are then used by BPGM to create the blinding polynomial based on the message being encrypted [5]. Our hardware implementation of NTRU algorithm will consider those IEEE 1361.1 components, as shown in Fig. 4.

B. NTRU Hardware Implementation

We have conducted a hardware implementation of NTRU algorithms and an IEEE 1363.1 system through a hybrid behavioral and structural very large integrated circuit hardware description language (VLSI) model as shown in Fig. 5. Components that are easily translatable to hardware are implemented using structural models, while some of the more complex components are written using behavioral style code.

The implementation of encryption algorithm follows the principle shown in Fig. 6(a). The encryption process is to form a new message M based on original message m. During VHDL testing, the encryption operation module retrieves the public key from the key generation primitive via the test bench. The test bench also provides the input message, from testing data, to the encryption operation and drives the strtenc signal to enable encryption. The message construct M and IGF seed data sData are formed and the seed data is sent directly to the IGF. The BPGM module is then activated, after which the output blinding polynomial r is multiplied by the public key h, using the convolution multiplication module. The output modulo q, R, and modulo two $R2$ are received from the convolution multiplication module and $R2$ is run through the MGF and masked with the message construct. Finally, the encryption operation forms the ciphertext $e = R + i \cdot \bmod q$.

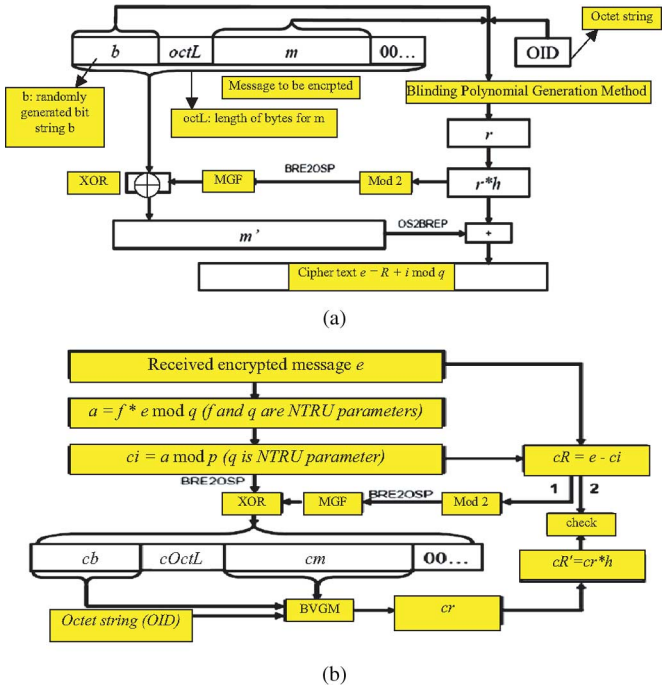


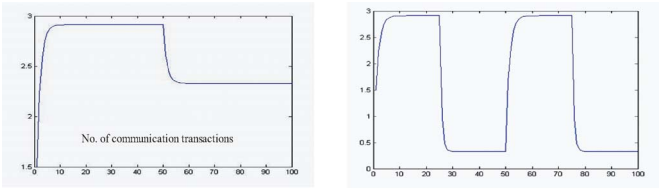
Fig. 6. NTRU implementation in VHDL. (a) Encryption. (b) Decryption.

The decryption operation module is set up in the test bench to accept the ciphertext e from the encryption operation module and the private key f from the key generation primitive module. The decryption module uses internal states to execute the decryption primitive to recover the candidate decrypted polynomial ci. The candidate value for $cr * h$ is recovered by $cR = e - ci$ and taken modulo two to be used in the MGF. The resultant mask is XOR'ed with cI and the candidate message construct cM is retrieved and held as an output from the decryption operation module. Although the candidate message construct was verified manually using the testing data, it should be noted that the BPGM module could easily be integrated with the decryption operation module for automated verification of the data as shown in Fig. 6(b).

V. Performance Analysis

A. Validity of Proposed Trust Model

As discussed in Section III, our trust model can successfully capture direct and indirect trust values between sensors in different distances (one-hop ring or two-hop ring-to-ring). Our trust calculation has extended basic matryoshka model to general telehealthcare communication network scenarios as shown in Fig. 1. To calculate trust value (the sum of direct and indirect trust values), we first make two neighboring nodes A and B (such as an IMD and a sensor) to exchange many rounds of



(a) (b)

Fig. 7. Trust level of sensor B. (a) Without attack. (b) With attack.

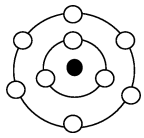


Fig. 8. Matryoshka structure of nine nodes.

TABLE I
FINAL TRUST FOR THE MATRYOSHKAS STRUCTURE OF FIG. 8

core	1^{st} **ring**			2^{nd} **ring**					
#1	**#7**	**#8**	**#3**	**#6**	**#9**	**#2**	**#5**	**#10**	**#4**
#2	1	9	6	7	8	4	3	5	10
#3	2	5	8	1	9	7	10	4	6
#4	1	3	5	7	8	2	6	9	10
#5	7	8	9	6	3	2	1	4	10
#6	7	3	4	9	10	2	5	1	8
#7	6	9	3	4	5	1	2	8	10
#8	2	9	5	1	6	4	3	7	10
#9	1	4	2	7	8	3	5	6	10
#10	1	2	5	7	8	9	6	3	4

medical data. Each round of data exchange is called a transaction. In each transaction, A will calculate B's trust level based on the successful received data events (each time B gets a network packet, it will forward to the medical server. Then, the server will tell A on B's packet receiving behaviors). We also ask other sensors around B to report their recommendations on B's trust level to A.

Case 1: No network attack, however with natural failure events. We first test natural communication with some occasional system failure events (however, no network attacks exist). As shown in our results [see Fig. 7(a)], according to node A, the reputation of node B declines due to RF interference or other reasons that cause B to drop some packets sent out from A. However, because node B does transactions well with other nodes in the neighborhood, the trust recommendations from those nodes on node B are good to node A, and its reputation does not decline to 0. Instead, it finally stabilizes at a certain value.

Case 2: With network attack. Suppose that B is compromised by an adversary. It can intentionally drop packets from time to time. Fig. 7(b) shows that the reputation of node B fluctuates if it behaves maliciously, i.e., boast or belittle in the recommendation periodically. Sometimes B may work normally (when no attack occurs), that is why its reputation does not decline to 0. To verify the efficiency of ring-based trust model in the matryoshkas scheme, we have generated the trust table for ten-node network by using the values from the trust computation we have described in Section III. In order to make the result look straightforward, we do not use floating values. Instead, we generate the trust table by using the random numbers between 0 and 10. Trust value to a node itself is defined as 0. This table shows the initial trust values for each of the nodes without any direct/indirect trust considerations.

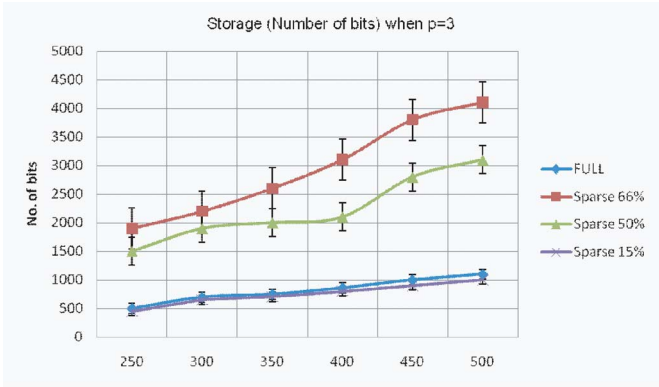
Now, let us assume the structure of matryoshkas is as Fig. 8 and further investigate its new trust values based on the models of direct trust from a node's observations and indirect trust from others' recommendations (see Section III). Here, the matryoshkas structure consists of one core and two rings. The innermost ring has three sensors. The second ring has six sensors. The updated trust for ten nodes is reevaluated based on (4)–(9). The final trust is shown in Table I.

B. NTRU Performance Analysis

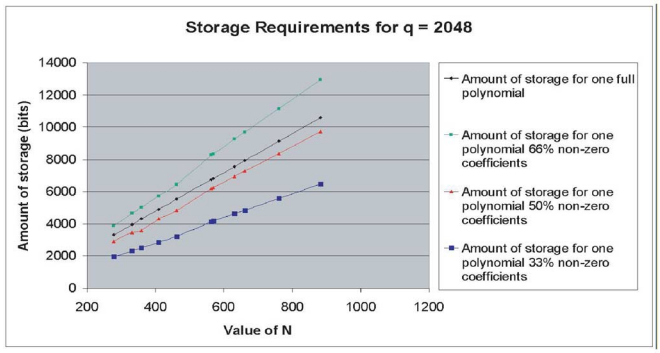
Next, we investigate the security performance of NTRU-based public/private keying schemes. Especially, we concentrate on three aspects as follows: 1) validity of the NTRU implementation, that is, its encryption and decryption effects; 2) the energy consumption performance. Since NTRU will be implemented in IMDs and sensors, its calculation operations cannot consume much energy; and 3) the memory overhead. The storage spaces in body medical devices are very limited (typically $<$ 300 KB). Such a small memory size needs to run an operating system, wireless network protocols, sensor data collection and processing, security algorithms, and other operations. Therefore, we should investigate the storage overhead of the NTRU scheme to optimize the NTRU algorithm and save more storage overhead.

The storage constraint is one of the largest concerns in IMDs. Here, we investigate two methods to store a polynomial: *full storage* and *sparse storage*. The former simply stores each coefficient of a polynomial in a linear array. Thus, we need $N\log(q)$ bits of storage for a polynomial with modulo q. The latter only stores the nonzero coefficients and the corresponding degree for each nonzero coefficient. Assuming there are num_{nz} percentage of nonzero coefficients, this method would require $N[\log(q)+\log(N)]\text{num}_{\text{nz}}$ or $N[\log(p)+\log(N)]\text{num}_{\text{nz}}$ bits of storage per polynomial. To see the relationship between the storage and other parameters such as N, p (or q), and num_{nz}, we first set up one parameter as constant (say, $p=3$), then we change other parameters (say N and num_{nz}). Fig. 9(a) shows the storage results. The data points used to generate the storage graphs were taken from parameter sets found in [16].

It can be seen from Fig. 9(a), when there are only 15% of nonzero coefficients, *full storage* and *sparse storage* almost have the same storage overhead. However, when there are more than 50% nonzero coefficients, *sparse storage* has lost its advantage. While the data amount of the full storage method increases linearly with N, that of the sparse storage method increases approximately $\mathcal{O}(N\log(N))$. As a consequence, with increasing N, the two methods will diverge, and the full storage method will become significantly more efficient.



(a)



(b)

Fig. 9. (a) Storage overhead when $p = 3$. (b) Polynomial storage for $q = 2048$.

TABLE II
ILLUSTRATIVE EXAMPLES OF TWO STORAGE METHODS

num_{nz}	33%	50%	66%
Intersection Point	$\approx 2\log(q)$	$\log(q)$	$\approx 0.5\log(q)$
N (q=2048)	$\approx$ 4,194,304	$\approx$ 2048	$\approx$ 2

Of clear significance, a comparison of the two methodologies for $q = 2048$ is shown in Fig. 9(b). The sparse storage method achieves slightly better results than the full storage method. Indeed, the intersection of the two methodologies can be found by setting the storage requirements equal to one another as in the following equations: $N\log(q) = N[\log(q) + \log(N)]\text{num}_{\text{nz}}$ and $\log(N) = \log(q) + 1/\text{num}_{\text{nz}} - 1$. Table II shows some examples based on the aforementioned equations.

It can be seen that one storage method could be significantly better than the other one either when $q \ll N$ or $N \ll q$. Therefore, it would be better to dynamically choose the storage on whether the modulus that the polynomial is reduced by is much larger than or much smaller than the degree of the polynomial. When those two parameters are close to each other, either way could be used. It would be interesting to see the tradeoff between increasing N as compared to increasing q. In Fig. 10, we compared two cases: 1) hold q constant, N varies and 2) N is constant and q increases. Both have a starting point of $(N, q) = (251, 197)$. As we can see from Fig. 10, the case of increasing q can lead to much less storage cost than that of increasing N.

In the IEEE 1363.1 standard, two equations are used to determine the effect of changes on N and q in relation to the lattice

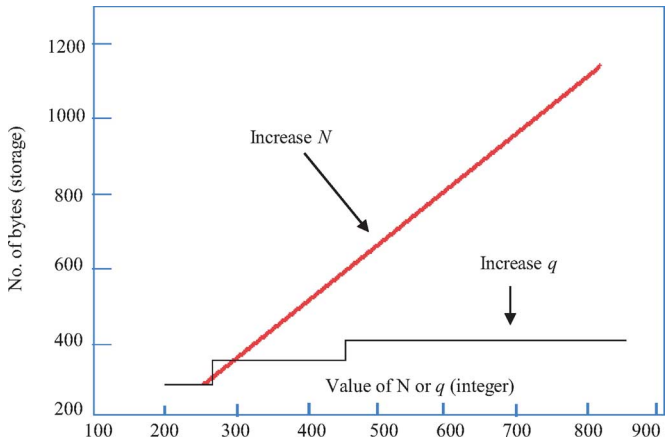


Fig. 10. Tradeoff between N and q.

security [5]: $c = \rho\sqrt{2N}$. In order to increase lattice breaking times, it is suggested to hold $a = N/q$ constant while increasing c. It is also stated that holding c constant and increasing a causes a slight decrease in lattice breaking times. It is also found that holding a and c constant while increasing N yields increases in lattice breaking times, as well.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we have investigated the secure IMD data access via IMD–sensor and sensor–sensor authenticated communications. Due to the low-complexity computation requirements in medical sensor/IMD security, we chose NTRU and Industrial Standard 1363 to build the public/private key pair in IMDs/sensors. To speed up the authentication process, we developed NTRU in hardware chips with detailed VHDL design principles. Because the sensors could be compromised by network attackers, we further developed indirect/direct trust models to determine the trust level of each sensor. We used a ring-based architecture to calculate the trust relationship among sensors. The performance results have shown the energy efficiency and security validity of the trust models. We have also analyzed the memory storage performance of the NTRU algorithms.

REFERENCES

[1] C. for Medicare, M. Services, O. of the Actuary, and N. H. S. Group. (2009, Mar.). *2007 National health care expenditures data* [Online]. Available: http://www.kaiseredu.org

[2] R. Istepanian, S. Laxminarayan, and C. S. Pattichis, *M-Health: Emerging Mobile Health Systems*, 1st ed. New York: Springer-Verlag, 2005.

[3] K. E. Hanna, F. J. Manning, P. Bouxsein, and A. Pope, *Innovation and Invention in Medical, Devices: Workshop, Summary*. Washington, DC: National Academies Press, 2001.

[4] O. for Civil Rights. (2009). *HIPAA medical privacy-national standards to protect the privacy of personal health information*. [Online]. Available: http://www.hhs.gov/ocr/hipaa/

[5] *The NTRU public key cryptosystem: A tutorial*. (2010). [Online]. Available: http://www.ntru.com/cryptolab/

[6] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in *Proc. ANTS III*, 1998, pp. 167–288.

[7] D. R. Stinson, *Cryptography: Theory and Practice*, 3rd ed. Boca Raton, FL: CRC Press, 2006.

[8] F. Hu, X. Cao, K. Wilhelm, M. Lukwiak, and S. Radziszowski, "NTRU-based confidential data transmission in telemedicine sensor networks," in

Security in Ad Hoc and Sensor Networks. Singapore: World Scientific, 2010, pp. 159–192.

[9] F. Hu, M. Jiang, M. Wagner, and D. Dong, "Privacy-preserving telecardiology sensor networks: Towards a low-cost, portable wireless hardware/software co-design," *IEEE Trans. Inf. Technol. Biomed.*, vol. 11, no. 6, pp. 617–627, Sep. 2007.

[10] F. Hu, M. Lukwiak, and Y. Xiao, "$NTRU$-based sensor network security: A low-power hardware implementation perspective," *Int. J. Security Commun. Netw.*, vol. 2, no. 1, pp. 71–81, Sep. 2008.

[11] K. Wilhelm, "Aspects of hardware methodologies for the ntru public-key cryptosystem," Master's thesis, Rochester Inst. Technol., Rochester, NY, Feb. 2008.

[12] F. Hu and Q. H. Y. Xiao, "Congestion-aware, loss-resilient bio-monitoring sensor networking," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 4, pp. 1–20, May 2009.

[13] L. A. Cutillo, R. Molva, and T. Strufe, "Privacy preserving social networking through decentralization," in *Proc. 6th Int. Conf. Wireless On-Demand Netw. Syst. Serv.*, Feb. 2009, pp. 145–152.

[14] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *Proc. IEEE 24th Int. Conf. Data Eng.*, Apr. 2008, pp. 506–515.

[15] A. Mathes. (2004). *Folksonomies: Cooperative classification and communication through shared metadata* [Online]. Available: http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html

[16] J. Hoffstein, N. Howgrave-Graham, J. Pipher, J. H. Silverman, and W. Whyte, "Hybrid lattice reduction and meet in the middle resistant parameter selection for ntruencrypt," NTRU Cryptosystems, Inc., Acton, MA, Tech. Rep., 2007.

Fei Hu (M'02) received the Ph.D. degree in signal processing from Tongji University, Shanghai, China, in 1999, and the Ph.D. degree in electrical and computer engineering from Clarkson University, Potsdam, NY, in 2002.

He is currently an Associate Professor in the Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa. His research interests include cognitive sensor networks, wireless security and their applications in biomedicine. His research has been supported by the National Science Foundation, Cisco, Sprint, BBN Inc., and other sources.

Qi Hao (M'06) received the B.E. and M.E. degrees from Shanghai Jiao Tong University, Shanghai, China, in 1994 and 1997, respectively, and the Ph.D. degree from Duke University, Durham, NC, in 2006, all in electrical and computer engineering.

He was a Postdoctoral Research Fellow in the Center for Visualization and Virtual Environment, The University of Kentucky, where his research was focused on 3-D computer vision for human tracking and identification. Currently, he is an Assistant Professor in the Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa. His research interests include compressive wireless sensors, intelligent wireless sensor networks, and biomedical signal processing.

Marcin Lukowiak (M'03) received the Ph.D. degree in microelectronics from the Poznan University of Technology, Poznan, Poland, in 2001.

He is currently an Assistant Professor in the Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY. His research interests include very large scale integration design (VLSI) design, digital systems, and cryptography algorithms.

Qingquan Sun received the Master of Science degree from the China Academy of Sciences, Beijing, China, in 2008. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa.

His research interests include sensor networks and machine learning.

Kyle Wilhelm is working toward the Graduate degree in the Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY.

His research interest is hardware design in very-high-speed integrated circuit hardware description language (VHDL).

Stanisław Radziszowski received the Ph.D. degree from the Institute of Informatics, University of Warsaw, Warsaw, Poland.

He has been a Professor in the Department of Computer Science, Rochester Institute of Technology, Rochester, NY, since 1995. His current research interests include combinatorial computing—solving classical problems in combinatorics, graph theory, and design theory, usually with the help of massive computations.

Yao Wu is currently working toward the Graduate degree in the Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa.

His research interests include wireless networks.