A Lightweight Integrity Authentication Scheme based on Reversible Watermark for Wireless Body Area Networks

Xiyao Liu[1,2], Yu Ge[2], Yuesheng Zhu[1] and Dajun Wu[2]
[1] The Communication and Information Security Lab,
Institute of Big Data Technologies, Shenzhen Graduate School
Peking University, Shenzhen, China
[lxyzoewx@163.com; zhuys@pkusz.edu.cn]
[2] Institute for Infocomm Research, Singapore
[geyu@i2r.a-star.edu.sg; djwu@i2r.a-star.edu.sg]
*Corresponding author: Y.S. Zhu[1]

Received April 28, 2014; revised July 17, 2014; accepted October 22, 2014; published December 31, 2014

Abstract

Integrity authentication of biometric data in Wireless Body Area Network (WBAN) is a critical issue because the sensitive data transmitted over broadcast wireless channels could be attacked easily. However, traditional cryptograph-based integrity authentication schemes are not suitable for WBAN as they consume much computational resource on the sensor nodes with limited memory, computational capability and power. To address this problem, a novel lightweight integrity authentication scheme based on reversible watermark is proposed for WBAN and implemented on a TinyOS-based WBAN test bed in this paper. In the proposed scheme, the data is divided into groups with a fixed size to improve grouping efficiency; the histogram shifting technique is adopted to avoid possible underflow or overflow; local maps are generated to restore the shifted data; and the watermarks are generated and embedded in a chaining way for integrity authentication. Our analytic and experimental results demonstrate that the integrity of biometric data can be reliably authenticated with low cost, and the data can be entirely recovered for healthcare applications by using our proposed scheme.

Keywords: Wireless Body Area Network, integrity authentication, reversible watermark, histogram shifting, lightweight

http://dx.doi.org/10.3837/tiis.2014.12.023

1. Introduction

In general, a Wireless Body Area Network (WBAN) consists of several small wearable or implantable sensors on/near/in a human body and a data aggregator to collect and process the sensor data. By collecting a person's biometric data, the continuous, real-time and ubiquitous health monitoring system with a WBAN can improve the quality of healthcare services [1-5]. However, the sensitive biometric data transmitted over the broadcast wireless channels in WBAN could be attacked easily, therefore how to ensure the integrity of data is one of the major security issues in WBAN [6-7]. Cryptography-based integrity authentication schemes [8-11], although ensuring the data integrity, are not suitable for the resource-constrained WBAN, due to their high demanding of memory, computational capability and power [12, 16-17].

By contrast, watermark-based schemes are lightweight solutions for integrity authentication [13-17]. In these schemes, the streaming data is divided into different groups; the watermarks are subsequently generated from the data groups and finally embedded into them in a chaining way. These watermarks will be damaged if there is any modification, insertion or deletion in the streaming data, thus the integrity of the streaming data can be verified by checking the watermarks. Most of the watermark-based schemes [13-16] introduce certain irreversible modifications to the streaming data. These slight modifications do not significantly affect the information expression and might be acceptable for non-critical applications. However, they cannot be applied directly to healthcare applications in WBAN because the biometric data is extremely sensitive that any inaccuracies can lead to incorrect medical assessments and potential serious consequences.

Recently, an integrity authentication scheme based on reversible fragile watermarking method for Wireless Sensor Network (WSN) was proposed by Shi et al [17]. This heuristic scheme is designed to avoid the modifications introduced by the watermark embedding. However, the hash values for each data element need to be calculated individually in its dynamic grouping method with variable group sizes. As a result, its computational complexity is much higher than that of the static grouping method with a fixed group size [16] in which only one hash value of the whole group is calculated. Considering the constraint of limited computational resources on a sensor node, the static grouping method is more suitable for WBAN. More importantly, the underflows and overflows may be caused by expanding of prediction errors during its watermark embedding process, which would cause errors in the recovered data and should be completely prevented.

For image authentication, the histogram shifting was adopted to avoid the risks of underflows and overflows in Tai's reversible watermarking scheme [18]. In this scheme, a local map is generated to restore the pixels changed by the histogram shifting and compressed in a lossless manner by using the run-length coding algorithm. The watermark is then generated by concatenating the compressed local map with the copyright information and embedded by expanding the difference of adjacent pixels. However, when this scheme is directly applied to authenticate the streaming data in WBAN, its watermarking procedure cannot chain the data groups together to check the insertions or deletions of groups; it also introduces additional cost for compressing the local map by using the run-length coding algorithm.

In this paper, a novel integrity authentication scheme based on reversible watermark for WBAN is proposed to solve the aforementioned problems. The key points of the proposed

scheme are as follows:

1) A static grouping method is applied to improve resource efficiency, which is different from Shi's scheme.

2) In our scheme, each data group is considered similar to one image in Tai's scheme [18]: the histogram distribution of the data elements in one group is shifted to avoid possible underflow or overflow; the local maps of different groups are generated to restore the shifted data and embedded as a part of watermark to save the transmission overhead for recording them; and a watermark is embedded by expanding the difference of adjacent data elements in one group for data restoring. Unlike Tai's scheme, the watermark embedded in the current group is generated by concatenating the hash values of the former group and the local map of the current group. In this manner, different groups are chained together, which is crucial to authenticate the integrity of streaming data. In addition, the run-length coding algorithm for compressing the local maps is not used to save the computational resource.

3) Unlike other watermark-based authentication schemes, the proposed scheme is implemented on a TinyOS-based WBAN test bed to validate its feasibility and effectiveness.

Our security analysis and experimental results demonstrate that the proposed scheme can detect any modification, insertion or deletion on biometric data reliably, without any underflow or overflow during the watermarking processes, and the original data can be entirely recovered after the watermark extraction. The computational complexity of our schemes is much lower compared to that of the Shi's scheme [17]. Lastly, the transmission overhead in the proposed scheme is insignificant.

The rest of the paper is organized as follows. The detailed description of our proposed scheme is provided in Section 2. The security analysis and experimental results are discussed in Section 3 and Section 4 respectively. Finally, the conclusions are drawn in Section 5 draws the conclusions.

The notations used in our paper are listed in **Table 1.**

Table 1. Notations

Notations	Definition
S	The size of data element
L	The watermark embedding level
N	The number of data elements in one group
F	The sampling rate of pulse sensor
P	The number of data elements in one packet
SN	The serial number
m	The length of the SN
DF	The group delimiter flag
G_i	The original i th group of streaming data
h_i	The secure hash value of G_i
h_i'	The secure hash value of the attacked G_i
k	The predefined symmetric key
$HASH$	The hash function
$\|\|$	The concatenation operator
W_i	The watermark embedded in G_i
M_i	The local map of G_i
O_i	The length of M_i
EC_i	The watermark embedding capability of G_i,
G_i'	The watermarked i th group of streaming data
W_i'	The watermark extracted from G_i'

M_i'	The local map extracted from G_i'
O_i'	The length of M_i'
Wp_i'	The pure watermark after excluding the local map
EC_i'	The watermark embedding capability of G_i'
EC	The watermark embedding capability
ECp	The pure watermark embedding capability after excluding the local map

2. Proposed Integrity Authentication Scheme

2.1 System Model

The system model of the proposed integrity authentication scheme is shown in **Fig. 1**. In this system, the watermarks are generated from the collected biometric data and embedded into it at the sensor mote, and then the watermarked biometric data is sent to the base station through WBAN. At the base station, the watermarks are extracted for integrity authentication and the original data is restored for further healthcare applications. The detailed procedures of integrity authentication scheme will be described in the following sections.

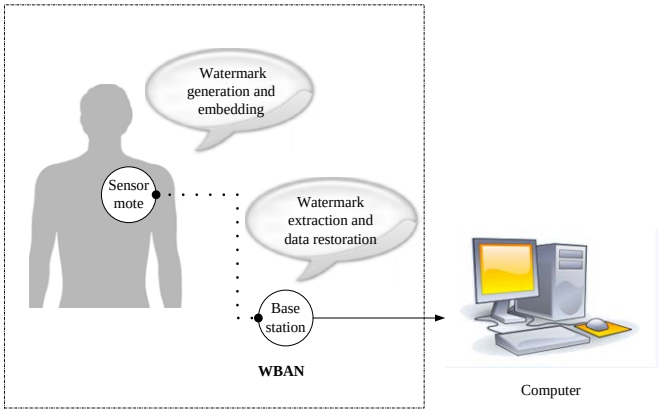

Fig. 1. System model

2.2 Encoding Phase

The encoding phase is executed on the sensor mote, where the collected biometric data is divided to a series of groups. For each group, the histogram distribution of its data elements is shifted and a local map is obtained. The watermark is generated from the hash value of the former group and the local map of current group, and then embedded into the current group by expanding the difference of adjacent data elements. The detailed procedures are shown in **Fig. 2** and described below.

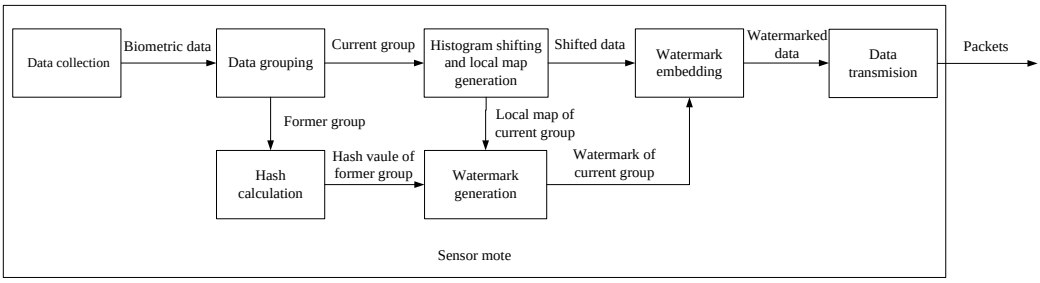

Fig. 2. Block diagram of encoding phase

1) Data grouping

The biometric data is collected continuously in real time by the wearable sensor mote, and then divided into a series of groups with a fixed size in a similar manner to the method in literature [16]. Without loss of generality, it is assumed that each group contains N data elements, and N is usually a large number to ensure the watermark embedding capacity. Due to the limitation of packet payload size in WBAN, each group is needed to be further divided into several packets consisting of several data elements for transmission. To help the receiver to identify which group is attacked and count how many group insertions or deletions occur, a *SN* is generated and attached to the packet. The *SN* is an ascending number from 0 to 2^m-1. Once *SN* reaches to 2^m-1, it is reset to 0 to prevent the indefinitely increasing. In addition, a *DF* is generated for the packet to identify when a group ends. It is set to 1 when a group ends and set to 0 otherwise. Both of the *SN* and *DF* are designed for individual packet rather than for a group in our grouping method, which is different from the method adopted in the literature [16]. The overhead caused by the design of *SN* and *DF* is $(m+1)/8$ bytes per packet. It is expected to be negligible when the packet size is sufficiently large, which will be described in detail in Section 4.2.3. **Fig. 3** shows an example of data grouping.

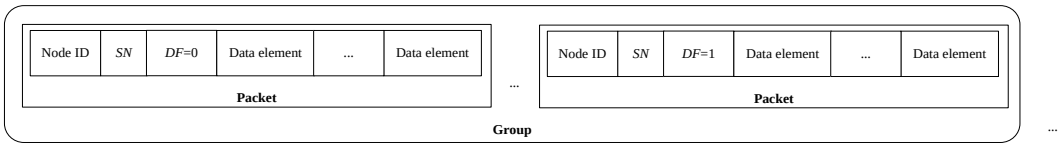

Fig. 3. Example of data grouping

2) Histogram shifting and local map generation

In general, the differences between two adjacent data elements collected by sensor mote are usually small since the sampling rate of sensor mote is quite high for healthcare applications in WBAN. This situation is similar to that of two adjacent pixels in one image. Therefore, a data group in our scheme is considered similar to an image in Tai's scheme [18]. For each group, the histogram distribution of data elements is narrowed down by 2^L units from both sides, and a local map is generated to distinguish the data elements changed by histogram shifting from the unchanged ones according to the following rule: for a data element with the original value in the range $[0, 2^L-1]$ or $[2^S-2^L, 2^S-1]$, 1 is assigned in the local map; for a data element with the original value is in the range $[2^L, 2^{L+1}-1]$ or $[2^S-2^{L+1}, 2^S-2^L-1]$, 0 is assigned in the local map. L is determined by the payload size of the copyright information.

3) Hash calculation

For G_i, h_i is computed for generating the watermark as shown in Eq. (1). If G_i is attacked, the calculated h_i' will be totally different from its original value. In our scheme, SHA-1 is adopted as the hash function to get an output with a fixed length.

$$h_i = HASH(k \| G_i) \tag{1}$$

4) Watermark generation

The W_i is generated as shown in Eq. (2) and the run-length coding algorithm for compressing the local map is not used to save the computational resource, which is different from Tai's scheme [18].

$$W_i = M_i \| h_{i-1} \tag{2}$$

By adopting the watermarks generated in this manner, different groups are chained together as shown in the **Fig. 4**, and no transmission overhead is introduced to record the M_i.

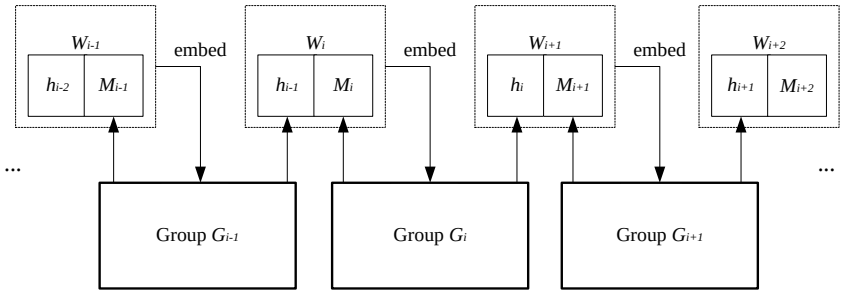

Fig. 4. Example of the chaining groups

5) Watermark embedding

Calculate the differences between the adjacent data elements in G_i as shown in Eq. (3).

$$D_j = \begin{cases} 0, & j = 1 \\ |X_j - X_{j-1}|, & otherwise \end{cases} \tag{3}$$

where X_j denotes the original value of the j th data element in G_i, $X_j \in Z$, $X_j \in [0, 2^S\text{-}1]$, $1 \le j \le N$, and D_j is the original difference between the j th data element and the j-1 th data element in G_i.

Expand these differences to embed W_i as shown in Eq. (4).

$$Y_j = \begin{cases} X_j, & if\ j = 1 \\ X_{j-1} + D_j^{'}, & if\ X_j \ge X_{j-1}\ and\ j \ne 1 \\ X_{j-1} - D_j^{'}, & if\ X_j < X_{j-1}\ and\ j \ne 1 \end{cases} \tag{4}$$

$$where:$$

$$D_j^{'} = \begin{cases} D_j + 2^L, & if\ D_j \ge 2^L \\ 2 \times D_j + w, & if\ D_j < 2^L \end{cases}$$

where Y_j is the value of the j th data element after watermark embedding, w is the embedded bit value of W_i, D_j is the original difference between the j th data element and the j-1 th data element in G_i, and $D_j^{'}$ is the expanded difference between them.

It is noted that the value of a data element is modified by at most 2^L units during the watermark embedding procedure. Therefore, underflow and overflow can be entirely avoided by the histogram shifting described in step 2.

The EC_i is then calculated. If EC_i is larger than the size of W_i, the M_i and the whole h_{i-1} are embedded. Otherwise, the M_i and only the lowest EC_i-O_i bits of h_{i-1} are embedded.

6) Data transmission

After watermark embedding, the $G_i^{'}$ is transmitted packet by packet through the WBAN.

2.3 Decoding Phase

The decoding phase is executed on the base station, where the watermark is extracted and the watermarked data is recovered to its original value when one group of watermarked data has been entirely received; then the watermark extracted from the received group is compared with the hash value of its former group to authenticate the data integrity of the former group. The detailed procedures are shown in **Fig. 5** and described below.

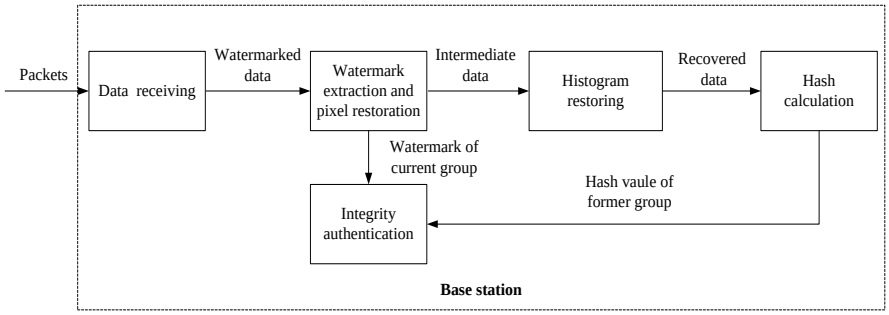

Fig. 5. Block diagram of the decoding phase

Without loss of generality, we assume that the received group is $G_{i+1}^{'}$. The detailed procedures of the decoding phase to authenticate the integrity of G_i are described as follows.

1) Data receiving

Receive the packet and check its *DF*. If *DF* is 0, the received packet is not the end of $G_{i+1}^{'}$, then the packet is stored in a buffer, and the base station continues to receive the next packet; otherwise the packet is the end of $G_{i+1}^{'}$, then the packet is put into the buffer, and watermark extraction is conducted.

2) Watermark extraction

Calculate the difference between the adjacent data elements in $G_{i+1}^{'}$. If $|Y_j\text{-}Z_{j-1}|<2^{L+1}$, extract the $W_{i+1}^{'}$ in Eq. (5).

$$w=\begin{cases}0, & if\ \ |Y_j-Z_{j-1}|\ \ is\ even\\ 1, & if\ \ |Y_j-Z_{j-1}|\ \ is\ odd\end{cases} \tag{5}$$

where Y_j is the value of the j th data element in $G_{i+1}^{'}$, Z_{j-1} is the restored value of Y_{j-1}, and w is the extracted bit value of $W_{i+1}^{'}$.

3) Data difference shrinking

Shrink the difference between two adjacent data elements as shown in Eq. (6).

$$Z_j=\begin{cases}Y_j, & if\ \ j=1\\ Z_{j-1}-D_j, & if\ \ Y_j<Z_{j-1}\ and\ j\neq 1\\ Z_{j-1}+D_j, & if\ \ Y_j\geq Z_{j-1}\ and\ j\neq 1\end{cases}$$

$$where:$$

$$D_j=\begin{cases}\left\lfloor \dfrac{D_j^{'}}{2}\right\rfloor, & if\ \ D_j^{'}<2^{L+1}\ and\ j\neq 1\\ D_j^{'}-2^{L}, & if\ \ D_j^{'}\geq 2^{L+1}\ and\ j\neq 1\end{cases} \tag{6}$$

$$D_j^{'}=|Y_j-Z_{j-1}|,\quad if\ \ j\neq 1$$

where Y_j is the value of the j th data element in $G_{i+1}^{'}$, Z_j is the restored value of Y_j, D_j is the recovered difference between the j th data element and the j-1 th data element in G_{i+1}, and $D_j^{'}$ is the expanded difference between them.

Continue step 2 and step 3 until $W_{i+1}^{'}$ is entirely extracted, and then the $EC_{i+1}^{'}$ is record.

4) Histogram restoring

Count the number of the restored data elements with the value in the range $[2^L, 2^{L+1}\text{-}1]$ or $[2^S\text{-}2^{L+1}, 2^S\text{-}2^L\text{-}1]$ and denote it as $O_{i+1}^{'}$. The first $O_{i+1}^{'}$ bits of $W_{i+1}^{'}$ are obtained as $M_{i+1}^{'}$, which should be the same with M_{i+1} if no attack or packet drop happens. For the data element with the value in the range $[2^L, 2^{L+1}\text{-}1]$ or $[2^S\text{-}2^{L+1}, 2^S\text{-}2^L\text{-}1]$, its assigned bit of $M_{i+1}^{'}$ is checked: if value

1 is assigned, the data element is shifted to its original state by 2^L; otherwise, no change is required. After the histogram restoring, G_{i+1} can be completely recovered.

5) Hash calculation

Obtain the G_i, which is recovered during the previous procedures for authenticating the integrity of G_{i-1}, from the buffer and calculate h_i as shown in Eq. (1).

6) Integrity authentication

Get $Wp_{i+1}^{'}$ by excluding $M_{i+1}^{'}$ from $W_{i+1}^{'}$. If the $EC_{i+1}^{'}$- $O_{i+1}^{'}$ is larger than 160 bits, compare $Wp_{i+1}^{'}$ with h_i, otherwise, compare the $Wp_{i+1}^{'}$ with the lowest $EC_{i+1}^{'}$- $O_{i+1}^{'}$ bits of the h_i. If they are the same, the integrity of the stored G_i is authenticated and G_i is sent into the database; otherwise, G_i will be rejected by the base station.

7) Buffer updating

Finally, G_i and $G_{i+1}^{'}$ are cleared from the buffer, and recovered G_{i+1} is stored in the buffer for further integrity authentication.

In our scheme, a $(S\times N)/8$ bytes buffer is needed on sensor mote to cache the data elements of the current group before watermark embedding, and another $(S\times N)/4$ bytes buffer is needed on base station to record data elements of both the current and the former groups. Delays will also be caused by buffering data elements of the current group before watermark embedding. The delays for different data elements are different and the longest one is $(N-1)/F$ s. The values of the two buffer sizes and the longest delay are given in Section 4.2.2.

3. Security Analysis

The scenarios discussed below generally occur independently. If two or more scenarios occur concurrently, security analysis can be derived from the corresponding individual scenarios.

3.1 Modification

Without loss of generality, the security analysis assumes that only the content of $G_i^{'}$ is modified. The influences and authentication results in different scenarios of modifications are shown in **Table 2**.

Table 2. Influences and authentication results in different scenarios of modifications

Scenarios	Influences	Authentication results
Change only the least bits of the data elements in which the watermark W_i is embedded.	W_i is changed; h_i is not changed.	G_{i-1} will be rejected; G_i will be accepted.
Change other bits of the data elements, but does not affect the watermark extracting result.	W_i is not changed; h_i is changed.	G_{i-1} will be accepted; G_i will be rejected.
Change other bits of the data elements, and affects the watermark extracting result.	W_i is changed; h_i is changed.	G_{i-1} will be rejected; G_i will be rejected.
Change the group *SN*.	W_i is not changed; h_i is changed.	G_{i-1} will be accepted; G_i will be rejected
Change the group *DF* from value 1 to value 0.	$G_i^{'}$ and $G_{i+1}^{'}$ are considered to be one group; W_i, W_{i+1}, h_i and h_{i+1} are all changed.	G_{i-1}, G_i and G_{i+1} will be rejected.
Change the group *DF* from value 0 to value 1.	$G_i^{'}$ is divided into several groups; W_i and h_i are changed.	G_{i-1} will be rejected; the divided groups will be rejected.

3.2 Insertion

The influences and authentication results in different scenarios of insertions are shown in **Table 3**.

Table 3. Influences and authentication results in different scenarios of insertions

Scenarios	Influences	Authentication results
Several data elements, *SNs* or *DFs* are inserted into G_i', and it does not affect the watermark extracting result.	W_i is not changed; h_i is changed.	G_{i-1} will be accepted; G_i will be rejected.
Several data elements, *SNs* or *DFs* are inserted into G_i', and it affects the watermark extracting result.	W_i is changed; h_i is changed	G_{i-1} will be rejected; G_i will be rejected.
Some packets with *DF*=0 are inserted into G_i.	The number of the packets in G_i' is changed; h_i is changed	G_{i-1} will be rejected; G_i will be rejected.
Some packets with *DF*=1 are inserted into G_i'.	G_i' is divided into two or several groups; W_i and h_i are changed.	G_{i-1} will be rejected; the divided groups will be rejected.
Some groups G_{k1}, G_{k2}, G_{k3}, …, G_{kn} are inserted between G_{i-1}' and G_i'.	The W_{k1} cannot match h_{i-1}, the W_i cannot match the h_{kn}, and G_{k1}, G_{k2}, G_{k3}, …, G_{kn} cannot be well chained since the attacker does not have the secret key k; h_i is not changed.	The inserted groups G_{k1}, G_{k2}, G_{k3}, …, G_{kn} and G_{i-1} will be rejected; G_i will be accepted.

3.3 Deletion

The influences and authentication results in different scenarios of deletions are shown in **Table 4**.

Table 4. Influences and authentication results in different scenarios of deletions

Scenarios	Influences	Authentication results
Several data elements are deleted from G_i', but it does not affect the watermark extracting result.	W_i is not changed; h_i is changed.	G_{i-1} will be accepted; G_i will be rejected.
Several data elements are deleted from G_i', and it affects the watermark extracting result.	W_i is changed; h_i is changed.	G_{i-1} will be rejected; G_i will be rejected.
Several *SNs* or *DFs* with value 0 are deleted from G_i'.	W_i is not changed; h_i is changed.	G_{i-1} will be accepted; G_i will be rejected.
The *DF* with value 1 is deleted from G_i'.	G_i' and G_{i+1}' are considered to be one group; W_i, W_{i+1}, h_i and h_{i+1} are all changed.	G_{i-1}, G_i and G_{i+1} will be rejected.
Some packets with *DF*=0 are deleted from G_i'.	The number of the packets in G_i' is changed; h_i is changed.	G_{i-1} will be rejected; G_i will be rejected.
The packet with *DF*=1 is deleted from G_i'.	G_i' and G_{i+1}' are considered to be one group; W_i, W_{i+1}, h_i and h_{i+1} are all changed.	G_{i-1}, G_i and G_{i+1} will be rejected.
Some groups G_i', G_{i+1}',…, G_{i+n}' are deleted.	The W_{i+n+1} cannot match h_{i-1}; h_{i+n+1} is not changed.	G_{i-1} will be rejected; G_{i+n+1} will be accepted.

3.4 Packet Loss

Although packet loss is not an active attack, it still poses challenges in data integrity and may cause incorrect medical assessment in healthcare applications using WBAN. In our scheme, packet loss can be identified as shown in **Table 5**.

Table 5. Influences and authentication results in different scenarios of packet loss

Scenarios	Influences	Authentication results
Some packets with DF=0 in G_i' are lost.	The number of the packets in G_i' will be changed; h_i is changed.	G_{i-1} will be rejected; G_i will be rejected.
The packet with DF=1 in G_i' is lost.	G_i' and G_{i+1}' are considered to be one group; W_i, W_{i+1}, h_i and h_{i+1} are all changed.	G_{i-1}, G_i and G_{i+1} will be rejected.

The above analyses have proved that our proposed scheme is able to robustly and reliably identify the three major types of attacks in WBAN, *i.e.*, modification, insertion and deletion, as well as packet loss situation. It indicates the effectiveness of the proposed scheme.

4. Experimental Results

4.1 Configurations of Experiments

We implemented the proposed scheme on a WBAN test bed based on TinyOS 2.x using the nesC programming language. A MicaZ sensor node, developed by Crossbow Technology [19], is adopted as the sensor mote on an adult body. The sensor node consists of an ATmega128L microcontroller, a 2.4GHz CC2420 RF transceiver, a 4KB RAM and a 128KB ROM. We use a pulse sensor in our experiments, which is developed by Sparkfun [20]. A gateway node is adopted as the base station to receive the data from the sensor node and is connected to a PC for data visualization and analysis. The encoding phase is executed in the on-body sensor node while the decoding phase is carried out at the gateway.

In our implementation, the size of data element is 16-bits, which consists of 10-bits collected raw pulse data and 6-bits heart rate data calculated from the pulse data for further healthcare applications. The pulse data is chosen as the watermark carrier. The watermark is generated from the local map of the pulse data and the hash value of the whole data group. In each packet, there are 16-bits Node ID, 7-bits *SN* and 1-bit *DF*. The structure of one packet is shown as **Fig. 6**.

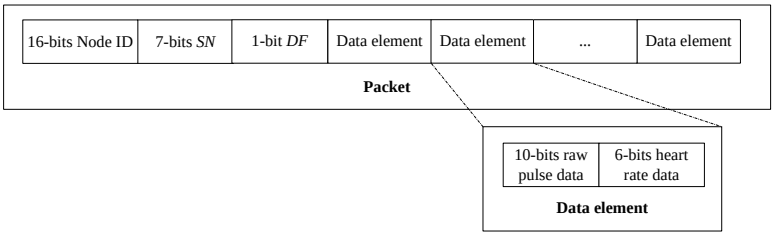

Fig. 6. Structure of one packet

4.2 Experiment A: Selection of Experimental Parameters

In this section, experiments and analyses are carried out to determine the critical parameters, including *L*, *F*, *N* and *P*. The selection criteria include *EC*, resource cost and transmission overhead.

4.2.1 Selection of *L*

L affects the *EC* and the variation of watermarked pulse data. Different from the image applications, we only use the restored pulse data after the watermark extraction for further healthcare applications. Therefore, the variation of watermarked pulse data is not the main issue of consideration. On the contrary, it is preferable that the *ECp* is greater than 160 bits. Otherwise, the possibility of the hash collision will increase since the length of the SHA-1 hashed value is fixed to 160 bits. In such a case, the performances of integrity authentication will be degraded. In our experiment, *EC*, *ECp* and *O* are calculated under different values of *L* in different situations. For each situation, 100 groups of data are collected, and their mean values are calculated. The experiment results are shown in **Fig. 7**.

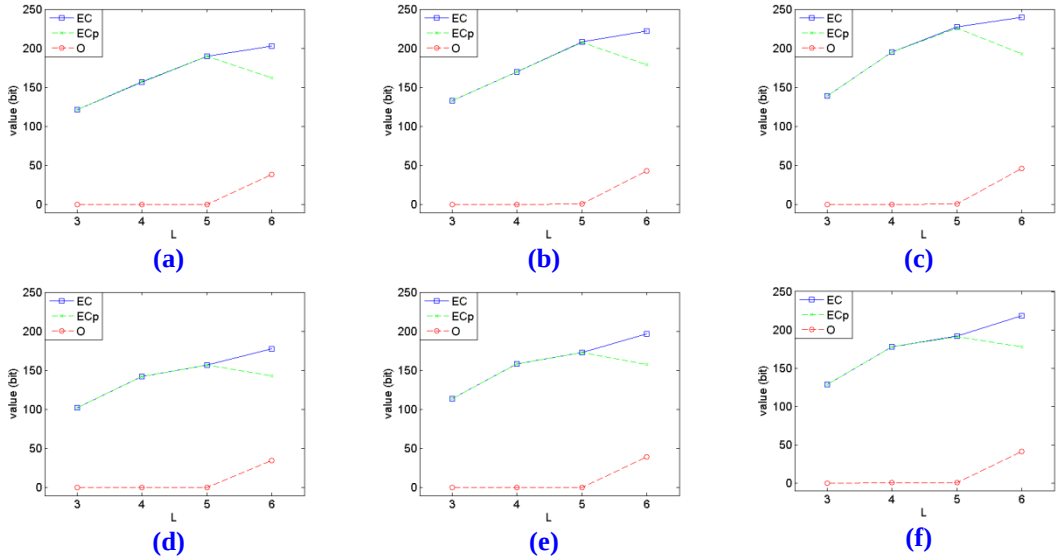

Fig. 7. *EC* under different *L* in different situations:
(a) *F*=100, *N*=250 (b) *F*=100, *N*=275 (c) *F*=100, *N*=300
(d) *F*=50, *N*=250 (e) *F*=50, *N*=275 (f) *F*=50, *N*=300.

It is observed from **Fig. 7** that as *L* increases, the *ECp* increases firstly and decreases thereafter, which can be explained theoretically. Since we only embed a watermark bit when the difference between two adjacent data elements is smaller than 2^L, it is obvious that the *EC* will increase when *L* increases. On the other hand, *O* will also increase when *L* increases since the local map is designed for data elements in the range [0, 2^{L+1}-1] or [1024 -2^{L+1}, 1023]. In addition, since the data elements are rarely near value 0 or 1023, the increment of *O* is very little when *L* is small whilst *O* will grow rapidly when *L* is large.

The *ECp* is given in Eq. (7), as:

$$ECp = EC - O \tag{7}$$

Thus, if *L* is small, the *ECp* will increase when *L* increases due to the increment of *EC;*

otherwise, it will decrease when L increases due to increment of O.

In the latter experiments, L is set to 5 according to **Fig. 7** to maximize the *ECp*.

4.2.2 Selection of *F* and *N*

Since both F and N can affect the *EC* and resource cost, those two parameters should be chosen collectively. When F increases, *EC* will increase but the power consumption will increase; while when N increases, *EC* will increase but the buffer size will increase, resulting in the increase of memory consumption. In this experiment, the mean value of *ECp* of 100 groups of pulse data for different F and N are calculated in **Table 6**.

Table 6. Mean value of *ECp* under different F and N

F (*Hz*)	100	66	50	40
N=225, mean *ECp* (*bit*)	174.1[a]	151.7	142.1	126.4
N=250, mean *ECp* (*bit*)	189.5[b]	168.7[a]	156.6	138.7
N=275, mean *ECp* (*bit*)	207.5[c]	183.2[b]	172.7[a]	152.1
N=300, mean *ECp* (*bit*)	226.1[c]	207.5[c]	191.1[b]	171.3[a]
N=325, mean *ECp* (*bit*)	246.3[c]	229.8[c]	213.9[c]	193.5[b]

Note: a: the category 1, b: the category 2, c: the category 3.

As shown in **Table 6**, there are three categories in which the mean values of *ECp* are greater than 160 bits. On one hand, since the mean values of *ECp* in the category 1 are slightly greater than 160 bits, many individual values of *ECp* in this category will be still smaller than 160 bits. Therefore, we do not consider the category 1. On the other hand, since larger F and N will lead to higher power and memory consumption, the category 3 is not adopted. Due to the above considerations, the category 2 is chosen to ensure sufficient *ECp* as well as save costs. By default, F is set to 50 *Hz*, and N is set to 300 in our latter experiments to achieve a trade-off between *ECp*, power consumption and memory consumption. In this manner, the sizes of the buffers on the sensor mote and the base station are $16\times N/8=600$ bytes and $16\times N/4=1200$ bytes, respectively, which can be provided by the current embedded devices. The longest delay for data elements is $(N\text{-}1)/F$ =5.98s, which can satisfy the real-time requirements of majority of health monitoring systems.

4.2.3 Selection of *P*

The data group is transmitted packet by packet in our scheme, thus N should be divisible by P to maximize the utilization of the payload of packet. In addition, each packet contains the 16-bits Node ID, 7-bits *SN* and 1-bit *DF*, which are also considered as the transmission overhead. To minimize the rate of such overheads, P should be set as large as possible. However, the allowed maximum packet size is limited according to TinyOS specification. Due to overall consideration of the above aspects, P is set to 50 and the packet size is 103 bytes. In this scenario, the rate of transmission overhead of Node ID is 2/103 (1.94%), and the rate of transmission overhead of *SN* and *DF* is 1/103 (0.97%). Both of them are insignificant.

4.3 Experiment B: Performances of Watermark Embedding

Watermark embedding is one core procedure for integrity authentication in the proposed

scheme. As mentioned in Section 4.2.1, it is better that *ECp* is greater than 160 bits. In this experiment, the *EC* of different groups for a single person and the *EC* for different persons are tested.

4.3.1 *EC* of Different Groups

The *EC* of 100 different groups for a single person are shown in **Fig. 8**.

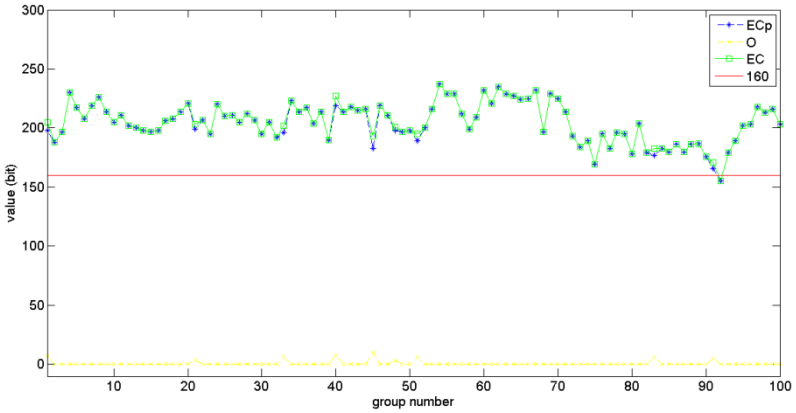

Fig. 8. *EC* of 100 Different Groups.

Fig. 8 shows that the *ECp* of only one group among these 100 groups is smaller than 160 bits. In addition, the *ECp* is 155 bits, which is very close to 160 bits. It demonstrates that the *ECp* in our scheme is sufficient to ensure the reliability of the integrity authentication.

4.3.2 *EC* for Different Persons

All the experiments described above are based on the pulse data collected from the same person. Since the pulse data of different persons are not identical, the *EC* for 12 different persons are obtained to better illustrate the watermark embedding performances of our scheme. For each person, 100 groups of pulse data are collected, and the results are shown in **Table 7**.

Table 7. *EC* for different persons

Person ID	Mean *EC* (bits)	Min. *EC* (bits)	Mean *ECp* (bits)	Min. *ECp* (bits)
1	191.6	155	191.1	155
2	218.3	193	218.3	193
3	207.1	176	206.9	176
4	217.3	179	216.3	179
5	223.4	176	222.7	176
6	191.3	165	190.9	158
7	218.4	185	218.0	185
8	190.3	173	190.3	173
9	210.6	193	209.7	193
10	223.7	195	223.7	195
11	231.8	206	230.9	206
12	235.8	202	235.6	202

Table 7 shows that all the mean *ECp* of the 12 different persons are much greater than

160 bits and the minimum *ECp* are larger than or close to 160 bits, which further demonstrates the sufficient *EC* in our proposed scheme under different situations.

4.4 Experiment C: Performance of Integrity Authentication

In this section, 13 types of different attacks are performed in the WBAN, including randomly modifying one bit of data element, *SN* or *DF*; randomly inserting one data element, *SN*, *DF*, packet or group; randomly deleting one data element, *SN*, *DF*, packet or group. The performances of integrity authentication under the 13 types of attacks as well as the packet loss situation are tested. Similar to Kamel's scheme [16], the integrity authentication is considered a failure if the attack is not detected by the base station in our paper. For each type of attack as well as the packet loss situation, 5000 attacked or packet-lost samples are checked and the testing results are listed in the **Table 8**.

Table 8. Performance of integrity authentication

Type	Number of failure checking	Successful rate
Modification on data element	1.0000	99.980%
Modification on *SN*	0.0000	100.000%
Modification on *DF*	0.0000	100.000%
Insertion of data element	0.0000	100.000%
Insertion of *SN*	0.0000	100.000%
Insertion of *DF*	0.0000	100.000%
Insertion of packet	0.0000	100.000%
Insertion of group	0.0000	100.000%
Deletion of data element	0.0000	100.000%
Deletion of *SN*	0.0000	100.000%
Deletion of *DF*	0.0000	100.000%
Deletion of packet	0.0000	100.000%
Deletion of group	0.0000	100.000%
Packet loss	0.0000	100.000%
Average	0.0714	99.998%

Table 8 shows that the number of failure checking is zero under most situations, and only 1 of 5000 tested samples failed to be checked out under the modification on data element, which is considered to be caused by the possible hash collision. The average successful rate of integrity authentication is as high as 99.998% under all the designed situations. This result is consistent with our analysis given in Section 3 and indicates that integrity of the data can be authenticated robustly and reliably in our proposed scheme.

4.5 Experiment D: Comparative Tests

4.5.1 Comparison of Underflow and Overflow with Shi's Scheme

Since underflow and overflow would cause the errors in watermark extraction and data restoration, they need to be completely prevented. In Shi's scheme [17], they expanded the prediction errors to embed the watermark. However, they did not employ any mechanisms to prevent underflows and overflows caused by expanding of prediction errors. Although the prediction errors in their approach are usually very small, as mentioned in their literature, underflow or overflow are still possible to occur in practical operations. In fact, if a collected pulse data is close to either 0 or 1023, underflow or overflow may happen even when the prediction error is very small. In the experiment, 100 groups of pulse data, *i.e*, 30000 data

elements are used to test the rates of underflow and overflow in the proposed scheme and Shi's scheme. The results are shown in **Table 9**.

Table 9. Comparison of the rates of underflow and overflow

F (*Hz*)	100	66	50	40
Overflow rate in Shi's scheme [17]	0.29%	2.08%	3.52%	6.98%
Underflow rate in Shi's scheme [17]	0.11%	0.15%	0.48%	0.42%
Overflow rate in the proposed scheme	0%	0%	0%	0%
Underflow rate in the proposed scheme	0%	0%	0%	0%

Table 9 shows that though the occurrence of underflow and overflow are low, there still exist some underflows and overflows when Shi's scheme is adopted for the pulse data. In contrast, there is no underflow or overflow when our proposed scheme is adopted. Therefore, the proposed scheme is more robust than Shi's scheme in terms of preventing overflows and underflows.

4.5.2 Comparison of Efficiency with Shi's Scheme

The execution time for the hash calculation is the primary component of the time used in both the encoding and decoding phases. As mentioned above, the static grouping method is adopted in our scheme, thus only the hash value of the whole group needs to be calculated. In contrast, the dynamic grouping is used in Shi's scheme, so that all the hash values of individual data elements have to be calculated. As a result, the proposed scheme is more efficient.

The execution times of encoding and decoding phases in the two schemes are compared in **Table 10**. Without loss of fairness in the comparison, we assume that the group size used in Shi's dynamic grouping method is 300, which is the same as that in the proposed scheme.

Table 10 shows that the execution times for the encoding phase and the decoding phase in Shi's scheme are about 25 times longer than those in the proposed scheme, which supports our aforementioned analysis.

Table 10. Comparison of the execution time

	Proposed scheme	Shi's scheme [17]
Execution times of encoding phase (ms)	106.39	2506.61
Execution times of decoding phase (ms)	97.33	2497.64

4.5.3 Comparison of Memory Consumption with Shi's Scheme

The memory consumptions in the two schemes are compared in **Table 11**. Without loss of fairness in the comparison, we assume that the group size used in Shi's dynamic grouping method is 300.

Table 11. Comparison of the memory consumption

	RAM in our proposed scheme	RAM in Shi's scheme [17]	ROM in our proposed scheme	ROM in Shi's scheme [17]
Sensor mote (bytes)	2257	1581	23980	23308
Base station (bytes)	2519	2541	23852	23870

On the sensor mote, our proposed scheme consumes slightly more RAM space (676 bytes) than Shi's scheme, and the ROM consumptions of the two schemes are almost the same. The difference of memory consumptions between the two schemes is mainly caused by the buffer used in our proposed scheme to cache the data elements of current group. Although the RAM consumption in our proposed scheme is a little more, it is merely around 55% of the RAM capacity. On the base station, since the buffer to cache two groups of data elements is needed in both of the two schemes, their memory consumptions are almost the same, which are around 61% of the RAM capacity and 19% of the ROM capacity. These results indicate the memory consumptions of our implementation can satisfy the memory constraint of WBAN.

4.5.4 Comparison with Other Watermark Based Schemes

A qualitative comparison between the proposed scheme with other watermark based data integrity authentication schemes [13, 16-17] is shown in **Table 12**.

Table 12. Comparison with other watermark based schemes

	Proposed scheme	Guo's scheme [13]	Kamel's scheme [16]	Shi's scheme [17]
Reversible	Yes	No	No	Yes
EC	Sufficient	Sufficient	Sufficient	Sufficient
Underflow/Overflow	None	None	None	Exist
Computational complexity	Low	Median	Low	Median

Table 12 shows that the proposed scheme is reversible compared with Guo's and Kamel's scheme. Furthermore, compared with Shi's scheme, there is no underflow or overflow in the proposed scheme and the computational complexity of the proposed scheme is much lower.

5. Conclusion

In this paper, a novel integrity authentication scheme based on reversible watermark for WBAN is proposed and implemented on a TinyOS-based WBAN. Our analysis and experimental results have demonstrated that our scheme has important advantages over other existing schemes. The integrity authentication of biometric data is robust and reliable under different attacks. The watermarking procedure is reversible, which outperforms Guo's and Kamel's schemes [13, 16]. Underflow and overflow will not occur in the watermark procedure, which outperforms Shi's scheme [17]. The integrity authentication scheme is lightweight and more efficient than Shi's scheme in terms of computational complexity. Lastly, no transmission overhead is introduced by the generated local map, and the transmission overhead in the proposed scheme is also insignificant.

References

[1] H. Cao, V. Leung, C. Chow and H. Chan, "Enabling technologies for wireless body area networks: A survey and outlook," *IEEE Communications Magazine*, vol. 47, no. 12, pp. 84-93, December, 2009. Article (CrossRef Link).

[2] C. Otto, A. Milenkovic, C. Sanders and E. Jovanov, "System architecture of a wireless body area

sensor network for ubiquitous health monitoring," *Journal of Mobile Multimedia*, vol. 1, no. 4, pp.307-326, January, 2006. Article (CrossRef Link).
[3] M. A. Hanson, H. C. Powell Jr, A. T. Barth, K. Ringgenberg, B. H. Calhoun, J. H. Aylor and J. Lach, "Body area sensor networks: Challenges and opportunities," *Computer*, vol. 42, no. 1, pp.58-65, January, 2009. Article (CrossRef Link).
[4] S. Lim and H. Lee, "Factors affecting medical incident care on WBAN," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 7, no. 5, pp. 1058-1076, May, 2013. Article (CrossRef Link).
[5] Y. O. Mohammed, U. A. Baroudi, "Partially observable Markov decision processes (POMDPS) and wireless body area networks (WBAN)," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 7, no. 5, pp. 1036-1057, May, 2013. Article (CrossRef Link).
[6] M. A. Ameen, J. Liu and K. Kwak, "Security and privacy issues in wireless sensor networks for healthcare applications," *Journal of medical systems*, vol. 36, no. 1, pp. 93-101, February, 2012. Article (CrossRef Link).
[7] M. Li, W. Lou and K. Ren, "Data security and privacy in wireless body area networks," *Wireless Communications, IEEE*, vol. 17, no. 1, pp. 51-58, February, 2010. Article (CrossRef Link).
[8] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen and D. E. Culler, "SPINS: Security protocols for sensor networks," *Wireless networks*, vol. 8, no. 5, pp. 521-534, September, 2002. Article (CrossRef Link).
[9] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proc. of the 2nd Int. Conf. on Embedded networked sensor systems, ACM*, pp.162-175, November, 2004. Article (CrossRef Link).
[10] B. Przydatek, D. Song and A. Perrig, "SIA: Secure information aggregation in sensor networks," in *Proc. of the 1st Int. Conf. on Embedded networked sensor systems, ACM*, pp. 255-265, November 5-7, 2003. Article (CrossRef Link).
[11] X. Ren and H. Yu, "Security mechanisms for wireless sensor networks," *IJCSNS International Journal of Computer Science and Network Security*, vol. 6, no. 3, pp. 155-156, March, 2006. Article (CrossRef Link).
[12] S. S. Iyengar, A. Durresi, V. Paruchuri and R. Kannan, "Data integrity protocol for sensor networks," *International Journal of Distributed Sensor Networks*, vol. 1, no. 2, pp. 205-214, 2005. Article (CrossRef Link).
[13] H. Guo, Y. Li and S. Jajodia, "Chaining watermarks for detecting malicious modifications to streaming data," *Information Sciences*, vol. 177, no. 1, pp. 281-298, January, 2007. Article (CrossRef Link).
[14] H. Juma, I. Kamel and L. Kaya, "Watermarking sensor data for protecting the integrity," in *Proc. of IEEE Conf. on Innovations in Information Technology*, pp. 598-602, December 16-18, 2008. Article (CrossRef Link).
[15] I. Kamel and H. Juma, "Simplified watermarking scheme for sensor networks," *International Journal of Internet Protocol Technology*, vol. 5, no. 1, pp. 101-111, April, 2010. Article (CrossRef Link).
[16] I. Kamel and H. Juma, "A lightweight data integrity scheme for sensor networks," *Sensors*, vol. 11, no. 4, pp. 4118-4136, April, 2011. Article (CrossRef Link).
[17] X. Shi and D. Xiao, "A reversible watermarking authentication scheme for wireless sensor networks," *Information Sciences*, vol. 240, no. 10, pp. 173-183, August, 2013. Article (CrossRef Link).
[18] W. L. Tai, C. M. Yeh and C. C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no.6, pp. 906-910, June, 2009. Article (CrossRef Link).
[19] CrossBow, "MICAZ datasheet." 2010. Available: Article (CrossRef Link).
[20] Sparkfun, "Pulse Sensor SEN-11574." Available: Article (CrossRef Link).

Xiyao Liu was born in Hunan Province, in 1987. He received the B.S. degree from School of Electrical Engineering and Computer Sciences, Department of Microelectronics, Peking University in 2008. He is currently pursuing the Ph.D. degree in School of Electrical and Computer Engineering, Department of Micro-electronics and Solid-State Electronics, Peking University. His research interests include digital signal processing, multimedia technology, and information security for sensor networks.

Yu Ge is a scientist in the Institute for Infocomm Research (I^2R), A-Star, Singapore. She received her M.Eng. and Ph.D. degrees from National University of Singapore and Nanyang Technological University respectively, all in wireless communication networks area. She joined I^2R in 2001 and worked in various research areas including VoIP in heterogeneous wireless networks, wireless mesh/ad hoc networks, and wireless sensor networks. She is currently leading a research team in the area of wireless body area networks (WBANs) for human-centric sensing. Her current research interests are transmission and sensing technologies in wireless communication networks for end-to-end human-centric service provisioning.

Yuesheng Zhu received his B.Eng. degree in Radio Engineering, M.Eng. degree in Circuits and Systems and Ph.D. degree in Electronics Engineering in 1982, 1989 and 1996, respectively. He is currently working as a professor at the Lab of Communication and Information Security, Shenzhen Graduate School, Peking University. He is a senior member of IEEE, fellow of China Institute of Electronics, and senior member of China Institute of Communications. His interests include digital signal processing, multimedia technology, communication and information security.

Dajun Wu received the B.S. degree in computer science from Northwest University, Xi'an, China, and the M. Eng. degree in computer engineering from Xi'an Jiaotong University, Xi'an, China, in 1993 and 1998, respectively. From 1998 to 2000, he was a Research Scholar in the School of Computer Engineering, Nanyang Technological University, Singapore. In 2007, he obtained the Ph.D degree in electrical and computer engineering from National University of Singapore. Since 2000, he has been with Institute for Infocomm Research, Singapore. His research field includes multi-media coding/transmission and sensor networks.