# Public-Key Authentication for Cloud-based WBANs

Thaier Hayajneh*
School of Engineering and Computing Sciences
New York Institute of Technology
New York, USA
thayajne@nyit.edu

Athanasios V. Vasilakos
Computer Science Department
Kuwait University
Kuwait
vasilako@cs.ku.edu.kw

Ghada Almashaqbeh
Dept. of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN, USA
galmasha@nd.edu

Bassam J. Mohd
Computer Engineering Dept.
The Hashemite University
Zarqa, Jordan
Bassam@hu.edu.jo

Muhammad A. Imran
Center for Communication Systems Research
University of Surrey
UK
m.imran@surrey.ac.uk

Muhammad Z Shakir and Khalid A. Qaraqe
Dept. of Electrical and Computer Engineering
Texas A&M University at Qatar
Qatar
Muhammad.shakir@qatar.tamu.edu
Khalid.qaraqe@qatar.tamu.edu

## ABSTRACT

Merging WBAN systems with cloud computing is an efficient solution to overcome limitations inherent in WBAN, especially in critical human-related applications such as remote health monitoring. In cloud-based WBAN, the nodes are classified into WBAN sensors that report measurements about the human body and WBAN actuators that receive commands from the medical staff and perform actions. Authenticating these commands is a critical security issue as any alteration may lead to serious consequences. This paper presents a light-weight public-key authentication protocol for cloud-based WBAN systems. The proposed protocol is based on the modified Rabin authentication algorithm which is customized in this paper by making some of its components run in parallel. To prove the efficiency of the modified Rabin we implemented the algorithm with different hardware settings using Tmote Sky motes. The Rabin algorithm with and without the parallel settings is also programmed on FPGA to evaluate its design and performance. The results show that secure, direct, instant, and authenticated commands can be delivered from the medical staff located at the cloud side to the WBAN nodes located in/on the human body. Compared to other public-key protocols implemented on the motes, Rabin algorithm achieved extremely faster verification and reasonable signature generation speed. Moreover, the suggested parallel settings of the Rabin signature generation significantly reduced the delays(by almost 80%) which is a critical issue in WBAN applications.

*Corresponding author.


BODYNETS 2014, September 29-October 01, London, Great Britain

DOI 10.4108/icst.bodynets.2014.257172

## 1. INTRODUCTION

Being outside hospitals have emerged recently to be an appealing option for both the medical staff and the patients. However, excessive network resources, real-time response, and smart monitoring with early notifications about the patients' status are some of the requirements to be supported. The most effective and cost efficient solution to achieve the aforementioned requirements is to deploy Wireless Body Area Networks (WBANs) [14][24].

Typically a WBAN consists of several sensor nodes that are attached in, on, or around a human body to report a variety of important physiological measurements. However, storing and processing the reported data at local medical units limits its accessibility and complicates the system design [13]. Integrating cloud computing with health-related systems come to promote the gained performance by utilizing the abundant resources of data processing and storage offered by the cloud [2][6]. In fact, cloud computing-based mobile health monitoring is claimed to be 10 times more energy-efficient and almost 20 times faster than a standalone mobile health monitoring application[2].

Several challenges are facing this integration including congestion, interference and coexistence issues, fast response, smart processing of the reported health-related data, supporting the maximum possible number of users, in addition to flexibility in operation and most importantly security [11][25][21]. In fact, data security is the largest obstacle that may impede the extensive usage of cloud-based WBANs. Researchers in [6] emphasized on the importance of defining system-wide security mechanisms in human-centered systems to guarantee people's privacy. Moreover, the researchers in [21][26] highlighted that security and privacy are amongst

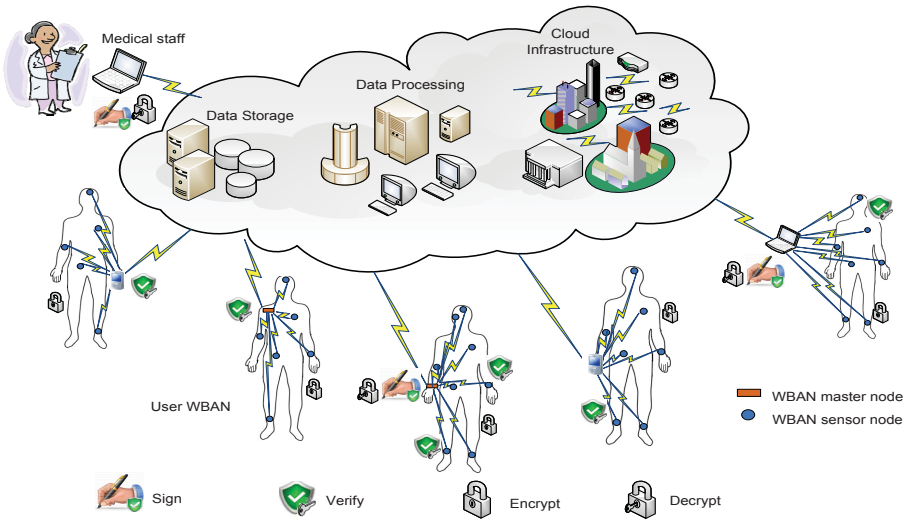


**Figure 1: Cloud-based WBAN Architecture**

the most challenging issues for mobile cloud computing.

In the literature there are some recent research effort that tried to address security in cloud-based WBANs. A scheme to capture data confidentiality in the cloud-assisted WBANs was proposed in [9]. Their goal was to achieve secure data communication between the cloud and WBANs. A secure patient-centric personal health information sharing and access control scheme in cloud computing was presented in [3] and proved to resist various possible attacks and malicious behaviors. [5] introduced a WBAN-cloud architecture to monitor a variety of biomedical conditions and fulfill security goals for various medical services. [12] proposed an attack resistant and lightweight trust management protocol. The proposed protocol was tested in network of TelosB motes and showed to improve the network performance and to protect it from malicious behaviors. Researchers in [28] proposed a practical Lightweight biometric-approach to authenticate messages in WBAN. They also developed an energy efficient key-agreement scheme that allows key sharing between WBAN nodes with low overhead.

Figure 1 shows the architecture of the cloud-based WBAN system that we use to demonstrate the security protocol suggested in this paper. In Figure 1, each patient is represented as a WBAN with tiny sensors or actuators reporting some measurements or performing subtle actions. The WBAN nodes are classified to two main types, the first type are WBAN sensor nodes that report data regularly to the master node about the health vital signs, e.g. heart beat rate, temperature, blood pressure...etc[14]. The second type are WBAN actuators that receive commands from the medical staff to perform actions and handle potential health problems , e.g. insulin pumps in case of diabetes[4].

There are two main security concerns in cloud-based WBAN architectures. The first issue is to guarantee the authenticity and integrity of the commands issued by the medical staff at the cloud side to the WBAN actuators. The commands involve actions that are performed by the WBAN nodes and may have serious impact on the human body. Hence, masquerading a command or creating a fake one is considered a serious threat to the human life. The second issue is to ensure the confidentiality of the reported data from the WBAN sensors to the medical staff.

In this paper, we address the aforementioned security concerns using a light-weight public-key authentication protocol. The proposed protocol is based on the modified Rabin authentication algorithm which has an extremely fast verification process compared to other public-key protocols. In fact, Rabin's scheme was shown to be several hundreds of times faster and lighter than RSA [22][7]. The encryption which is performed by the WBAN sensors is identical to the verification process. This implies that we only require the WBAN nodes to perform the light part of the Rabin algorithm.

On the other hand, the heavy part of the Rabin scheme, i.e. signature generation and data decryption is performed by the medical staff or the master node in some cases. In this paper we modified the Rabin scheme to run some components of the signature generation algorithm in parallel. This should enhance its performance and make the Rabin scheme more suitable for WBAN sensitive applications by reducing the potential response time.

To evaluate the performance of the modified Rabin with the WBAN, we implemented the algorithm with different hardware settings using Tmote Sky motes. Moreover, Rabin algorithm with and without the parallel settings is also implemented on FPGA to evaluate its design and performance. The aim is to prove that a light-weight public-key can achieve the desired real-time response in cloud-based WBANs with high security and minimal power consumption.

The remainder of this paper is structured as follows. Section 2 presents the cloud-based WBAN system and the security models. Section 3 describes the Rabin algorithm. Section 4 illustrates the FPGA implementation where the Testbed implementation is presented in Section 5. Finally, Section 6 concludes the paper.

## 2. SYSTEM AND SECURITY MODELS

In this section, the system and security models that we used in this paper are described. Particularly, the main characteristics, relations, functionalities, and the basic security aspects of the master nodes, the WBAN nodes, and

the transferred data are explored. The basic cloud-based WBAN architecture used in this paper is depicted in Figure 2.

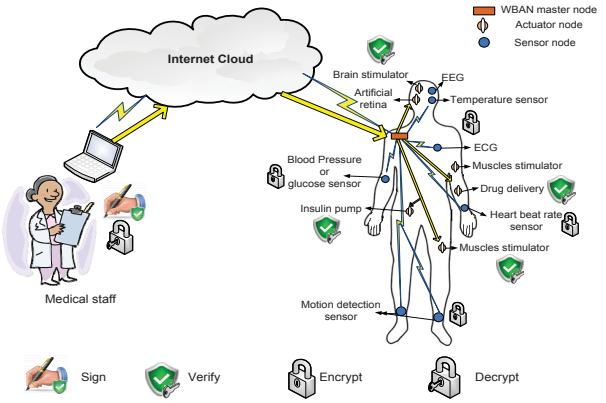


**Figure 2: Cloud WBAN Detailed Model.**

## 2.1 System Model

In our system we assume two types of master nodes: relay and smart master nodes. In the first type the master node collects the reported data from the WBAN nodes, reports them to the medical staff, receives commands from the medical staff, and sends them back to the intended WBAN nodes. In the second type the master node has additional functionality to process the reported data from the WBAN nodes and generate suitable commands to handle a potential health problem faced by the patient. In the first type the master node could be a regular WBAN node, i.e. has limited hardware, software, and power capabilities. While in the second one it could be portable smart digital devices in which the computational and power resources are abundant.

The WBAN nodes used in our system, as shown in Figure 2, are also classified into two types: sensors and actuators. The sensors are able to monitor the body health indicators and generate data packets of the measured data. Examples of WBAN sensors include sensors to measure the body temperature, the heart beat rate, the blood pressure, ECG, EEG...etc [14][23]. On the other hand, the actuators are nodes that have the suitable hardware to perform actions specified by the commands sent by the medical staff at the cloud side or the smart master node. Examples of WBAN actuators are: artificial retina, insulin pump, automatic drug delivery, muscles stimulator...etc [4]. Both WBAN sensors and actuators complement each others work, i.e. based on the reported data by the sensors the needed action is performed by the actuators.

Moreover, as illustrated in Figure 2, the data transmitted through the network is also divided into two types: periodic reported data by the sensors, and commands generated by the medical staff/smart master nodes. The periodic data is sent on regular basis with different packet sizes based on the sensor type. For example, ECG sensors send large data packets and more frequently than temperature sensors [23]. For the commands, their frequency depends on the health status of the patients. Patients with critical situations may receive more commands compared to patients with stable body health signs.

## 2.2 Security Model

As elaborated earlier, one of the most critical security issues in the presented cloud-based WBAN model is to guarantee that the commands issued by the medical staff at the cloud side to the actuators are not altered or fake. Due to the large number of WBAN nodes, their wide distribution, and the possibility for insertion/removal of WBAN nodes, we argue that using public-key cryptography is the most efficient solution to achieve the desired security requirements. The WBAN nodes need only to store the public key(s) of the medical staff that are authorized to issue controlling commands. Although being a less computationally expensive option, using symmetric cryptography imposes new challenges in terms of key management and distribution. However, if a computationally expensive public-key cryptographic system is used to provide all security services then it will burden the WBAN nodes and result in long delays and energy consumption.

Since the main security process that WBAN nodes perform is to verify the authenticity of a signed message we decided to use a public-key scheme that has a fast and efficient signature verification process. We found that Rabin algorithm is an excellent candidate that satisfies our requirements [20]. The medical staff or smart master signs their issued commands using a digital signature with their private key. The WBAN nodes will use the medical staff or smart master public key to verify the integrity and authenticity of the delivered commands.

Another important security service that can also be covered by our proposed security framework is the confidentiality of the reported data from the sensors to the medical staff. In this case, the WBAN nodes can encrypt their reported data using the medical staff or smart master public key. Fortunately, with Rabin scheme this process is identical to the verification process and is considered extremely light compared to the signing process of other public-key algorithms.

# 3. RABIN ALGORITHM

Rabin algorithm was originally proposed by M. K. Rabin in [20], and sometimes it is considered a special case of RSA. However, Rabin's scheme was shown to be several hundreds of times faster and lighter than RSA [22][7]. This makes it an excellent candidate for our cloud-base WBAN model.

## 3.1 Original Rabin

In this section, we present the details of the original Rabin public-key signature scheme[16]. At first each node should perform the following to generate a key pair:

1. Node A chooses two large random strong prime numbers, $p$ and $q$.
2. Compute $n = p.q$.
3. A's public key is $n$, private key is $(p, q)$

Node A signs a message $m \in M$ (where $M$ is the message space) as follows:

1. Compute $\tilde{m} = R(m)$, where R is the redundancy function.
2. Compute $s = \sqrt{\tilde{m}} \mathrm{mod} n$.
3. A's signature for $m$ is $s$.

Node B who receives $s$ can verify the signature as follows:

1. Get A's public key $n$.
2. Compute $\tilde{m} = s^2 \mathrm{mod} n$.

3. Verify that $\tilde{m} \in M_R$ (where $M_R$ is the image of $R$), if not then reject the signature.
4. Recover $m = R^{-1}(\tilde{m})$.

## 3.2 Modified Rabin

To overcome some of the issues with the Rabin scheme a modified version of the Rabin signature is provided in [16]. At first each node do the following to generate a key pair:

1. A selects two random primes $p \equiv 3(\text{mod}8)$, and $q \equiv 7(\text{mod}8)$ and compute $n = pq$.
2. A's public key is $n$, private key is $d = (n-p-q+5)/8$

Node A signs a message $m \in M$ as follows:

1. Compute $\tilde{m} = R(m) = 16m + 6$.
2. Compute the Jacobi symbol $J = \left(\frac{\tilde{m}}{n}\right)$
3. If $J = 1$ then compute $s = \tilde{m}^d \text{mod} n$.
4. If $J = -1$ then compute $s = (\tilde{m}/2)^d \text{mod} n$.
5. A's signature for $m$ is $s$.

For the Jacobi symbol $J = \left(\frac{\tilde{m}}{n}\right)$, we used the following recursive algorithm to compute it:

1. If $\acute{m} = 0$ and $n = 1$, return $J = 1$
2. If $\acute{m} = 0$ and $n = 0$, return $J = 0$
3. If $\acute{m} = 2$ and $n \equiv 1$ or $7(\text{mod } 8)$, return $J = 1$
4. If $\acute{m} = 2$ and $n \equiv 3$ or $5(\text{mod } 8)$, return $J = -1$
5. If $\acute{m} \geq n$, return $\left(\frac{\tilde{m}\%n}{n}\right)$
6. If $\acute{m}\%2 = 0$, return $\left(\frac{2}{n}\right) * \left(\frac{\tilde{m}/2}{n}\right)$
7. If $\acute{m}\%4 = 3$ and $n\%4 = 3$, return $-1 * \left(\frac{n}{\acute{m}}\right)$
8. return $\left(\frac{n}{\acute{m}}\right)$

Node B receives $s$ and can verify the signature as follows:

1. Get A's public key $n$.
2. Compute $\acute{m} = s^2 \text{mod} n$.
3. If $\acute{m} \equiv 6(\text{mod } 8)$, take $\tilde{m} = \acute{m}$.
4. If $\acute{m} \equiv 3(\text{mod } 8)$, take $\tilde{m} = 2\acute{m}$.
5. If $\acute{m} \equiv 7(\text{mod } 8)$, take $\tilde{m} = n - \acute{m}$.
6. If $\acute{m} \equiv 2(\text{mod } 8)$, take $\tilde{m} = 2(n - \acute{m})$.
7. Verify that $\tilde{m} \in M_R$ if not then reject the signature.
8. Recover $m = R^{-1}(\tilde{m}) = (\tilde{m} - 6)/16$

Even with the Jacobi method, Rabin signature generation is not significantly more computationally intensive than RSA[16]. In this paper we used the modified Rabin with the Jacobi algorithm and refer to it as Rabin scheme. We further modify the signature generation of Rabin scheme by running some of its components in parallel to improve its performance in terms of delay. The details of this modification are provided in the coming section. In our testing and evaluation we only considered the signature verification and generation processes as they are identical to the data encryption and decryption processes.

# 4. FPGA IMPLEMENTATION

The Rabin scheme was implemented on Field Programmable Gate Array (FPGA) platform. Compared to other hardware platforms, FPGA implementation provides superior features, including lower cost, faster development time, flexibility, and configurability. The design was implemented in the Verilog Hardware Description Languages (VHDL) and the code was dynamically validated using Modelsim simulations. Next, the Verilog code was compiled and analyzed using Quartus-II software [1]. In the analysis phase, we carefully examined the timing, resource utilization, and power results. Finally, the synthesized code was downloaded on Altera FPGA board to demonstrate the system correctness.

We implemented three different designs with FPGA: the verification process, the Rabin signature, and the proposed parallel Rabin signature. Figure 3 illustrates the Finite State Machine (FSM) used in the verification algorithm at the receiver node. The receiver is initially at the Idle state waiting for commands to arrive. After a message is arrived, the receiver enters the verification state, obtains the public key n of the sender, and run the Rabin verification algorithm. If the message is verified, the required command is performed by the receiver node and then it returns back to the Idle state. On the other hand, if the message is unverified, the receiver drops it and reset to the Idle state.

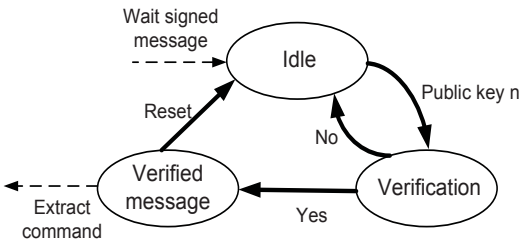


**Figure 3: Rabin verification FSM**

Figure 4 shows the FSM of the Rabin signature design. Again, the sender is initially at the Idle state, when a certain action is needed to be performed by the WBAN nodes a command message is generated. At this time, the sender moves to the next state to start the signature process using its private key. The result of the redundancy function is passed to the Jacobi state to compute the Jacobi symbol $J$. Based on the symbol value the next state is determined, value of 1 transfers the sender to the Full-m state while the value of -1 causes transition to the Half-m state. The two states differ in using either $m$ or $m/2$ value to generate the signature as described earlier. After that, the sender reset and returns back to the Idle state waiting a new command to be signed and sent to the receiver.

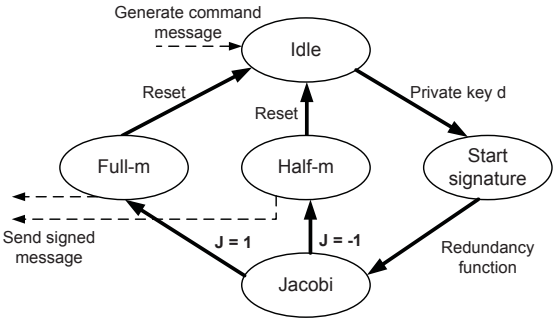


**Figure 4: Rabin signature FSM**

The parallel design of the Rabin is created using multiple modules, as illustrated in Figure 5, that are running at the same time: Jacobi, Full-m, and Half-m modules. As elaborated earlier, the idea is to make the signature generation process as fast as possible to decrease any potential delays, which is important in sensitive WBAN applications. Jacobi algorithm starts from IDLE state and ends up with Jacobi state to compute the Jacobi symbol (as in Figure 4). By the time the Jacobi module is finished, both the Half-m and Full-m states are completed and the one to be used for

Table 1: Resource utilization.

| Algorithm | LEs | LE type | | |
|---|---|---|---|---|
| | | Combinational | Register | Both |
| Rabin verification | 8025 | 7915 | 67 | 43 |
| Rabin signature | 12053 | 11580 | 46 | 427 |
| Parallel Rabin signature | 19781 | 18877 | 47 | 857 |

the signature value is selected (final decision) based on the Jacobi symbol value.

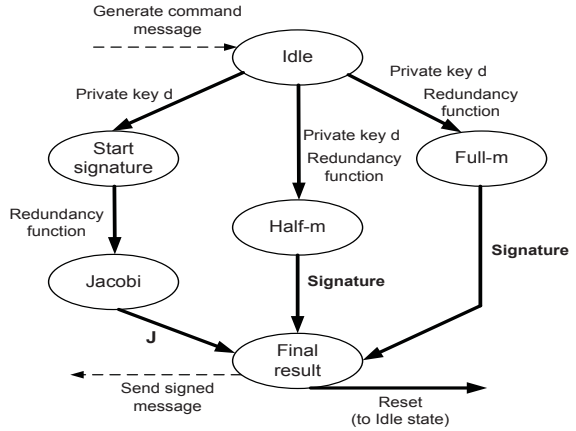


**Figure 5: Parallel Rabin signature FSM**

After that, the sender (signer of the commands) and the receiver (verifier of the commands) were implemented using Verilog. The Verilog code was then verified using Modelsim simulations. Figure 6 shows the waveform for the input and the output signals generated by the sender for the parallel Rabin signature. Clearly, Figure 6 illustrates the correct behavior of the algorithm.

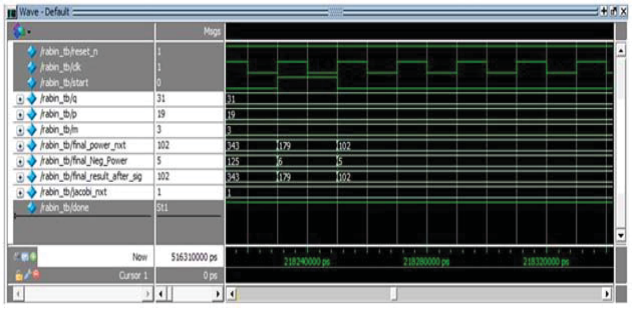

**Figure 6: Modelsim wave diagram at the sender**

In what follows, we discuss the FPGA implementation results in terms of hardware resources, timing, energy and power consumption. Table 1 highlights the resource utilization results of the designs expressed in logical elements (LEs). Logical elements are the smallest unit used in the logic circuits implemented in the Altera board. These elements are configured as combinational circuit, registers, and both. The Parallel Rabin understandably requires 64% extra LEs because of the added parallelism. Furthermore, the verification resources is less than both signature implementations since it is a lighter and less complicated process.

Table 2: Timing analysis (in ns).

| Algorithm | Tsu | Tco | Clk-Clk |
|---|---|---|---|
| Rabin verification | 1.5 | 14.1 | 5 |
| Rabin signature | 10.5 | 297.5 | 372.4 |
| Parallel Rabin signature | 8.7 | 376.8 | 371.2 |

Table 3: Power and energy results.

| Algorithm | Power (mW) | Energy (nJ) |
|---|---|---|
| Modified Rabin verification | 0.4 | 0.052 |
| Modified Rabin signature | 33.9 | 4.41 |
| Parallel Rabin signature | 114.3 | 14.86 |

The timing results for the different implemented designs are summarized in Table 2. The exhibited results include the following categories: propagation delay from primary inputs to register (Tsu), from register to primary outputs (Tco), and from register to register (Clk-Clk). As shown, computationally laden signature algorithm is stretching the timing delays, and lowering the design frequency to just above 50MHz. On the other hand, the verification design does not experience the lengthy timing delays.

Finally, the power and energy results are found in Table 3. Justifiably, the parallel Rabin consumes more power (about three times) and energy (about two times) that of the modified Rabin. This is due to the fact that the added hardware and the increase in switching activity have contributed to the increase in power and energy. Whereas, the verification process in Rabin consumes about 1% of the power consumed by the signature generation process. This justifies our selection for Rabin and requiring the WBAN nodes to only perform the signature verification (or equivalently message encryption) process.

# 5. TESTBED IMPLEMENTATION

In this section, we explore the testbed implementation used to evaluate the performance of the proposed security model. Similar to [10, 18], we used TinyOS and nesC language to develop the algorithms code and to upload them on Moteiv Tmote sky motes. In what follows the main aspects of the used tools, the implementation details, the experiments setup, and the obtained results are discussed.

## 5.1 Implementation Tools and Setup

TinyOS is an open-source operating system designed for wireless embedded sensor networks. It features a component-based architecture which enables rapid innovation and implementation while minimizing the code size. nesC is an extension to C language designed to embody the structuring concepts and execution model of TinyOS. To develop the code of the signature and verification algorithms we translated the big number library from C to nesC. The translated library allows mathematical operations of numbers of size 512 bits.

Tmote sky is an ultra low power wireless module that can be used in rapid application prototyping. Tmote leverages industry standards like USB and IEEE 802.15.4 to interoperate seamlessly with other devices. It uses an 8MHz Texas Instruments MSP430 microcontroller with 10k RAM and 48k Flash. Although Tmote is one of the advanced customizable sensor nodes, its computational capabilities remain very lim-

ited, and this constraint must be considered while building the intended WBANs.

The packet format used in TinyOS is the same one used on the 802.15.4. The default Data field has a maximum size of 29 bytes. We created our own structure to use the data field of the packet as follows:

| 1 | 1 | 1 | 4 | 20 |
|---|---|---|---|---|
| **src** | **pID** | **offset** | **message** | **signature** |

The **src** field is the source address of the sending mote, **pID** is the ID of the current packet (used for fragmentation), **Offset** is the offset from the initial packet (used for fragmentation), **message** is the message to be sent, and finally **signature** is the signature generated by the medical staff/smart master node. As shown, the first three fields are of size 1 byte, while the message is of size 4 bytes and the signature field has a size of 20 bytes. In our experiments, the commands signature is of size 64 byte, hence we need 4 packets for each signature to be sent. To enable us to view the packets sent by the motes, we ran the Serial Forwarder on the ComPort of the mote and the Listener tool available in TinyOS.

## 5.2 Experiment Setup

We built a testbed that contains three motes: master node, WBAN node, and an attacker node. The testbed has been configured to depict two different scenarios. In the first one, shown in Figure 7, the relay master node is tested. However, in the second scenario the same mote is configured to mimic the smart mote functionality that generates signed commands, see Figure 8.

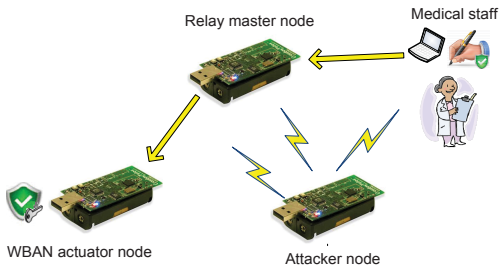


**Figure 7: First testbed scenario.**

The WBAN node is the receiver mote that runs the verification algorithm. Two of the LEDs found on the Tmote sky are used to indicate whether the verification is successful or not as follows: Green LED turns ON if the signature is verified, and Red LED turns ON if the verification of the signature fails.

The relay master node is a mote running TOSBase which is an application in TinyOS that makes the mote act as a bridge between the serial device (a PC) and the radio link (our wireless network). This application includes queues in both directions to guarantee that once a message enters the queue it eventually leaves on the other interface. On the mote, this can be visually observed for every packet received and successfully bridged by the toggling of the Green LED.

The smart master mote uses its own public/private key pair and runs the signing algorithm to generate the signature. We programmed the smart master mote to send the signature in four fragments where one fragment is sent every 400 ms. The receiver mote collects these fragments in

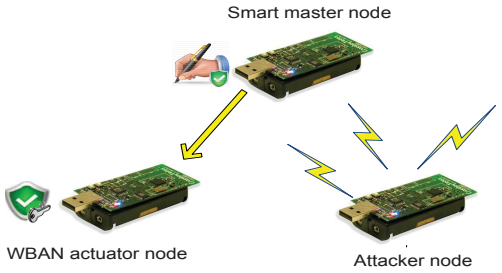


**Figure 8: Second testbed scenario.**

a global variable putting each piece in its correct position. Once it receives all four fragments it runs the verification algorithm, and indicates the result on the LEDs as mentioned before. Finally, we programmed the attack mote to impersonate the master node and sends a bogus signature to the receiver mote.

## 5.3 Results and Comparisons

As mentioned earlier, we used big numbers of size 512 bits for all variables used by the signature and all intermediate calculation. Generally, public-key authentication incorporates complex computations which results in slow performance and high power consumption. Therefore, we evaluated the proposed security model in terms of the needed operations and time to sign/verify the commands.

In terms of computational requirements, the Rabin algorithm has only one expensive operation on the receiver mote which is squaring the received signature modulo the public key. Note that this operation is much cheaper than the two expensive operations that are performed during the generation of the signature, i.e. computing the Jacobi symbol (which is a recursive function that uses modulo), and the modular exponentiation. However, this is a concern for the smart master node that is a WBAN node (limited hardware resources). If the medical staff is performing the commands signing process, the signing complexity is not an issue where high power devices are used for processing. However, in this case achieving a rapid response with minimal delay is still desirable.

For the second scenario with the smart master node, we obtained a signature generation time average of 22 sec. The verification time on the other hand took less than 1 sec. This result is very significant because it reduces the computational requirements on the receiver to verify an incoming signature. It also reduces the effect of a DoS attack in case of a malicious node is sending bogus signatures or packets. On the other hand, using the parallel settings proposed in the previous section the average signature generation time was reduced to 5 sec.

We note that these signing and verification times are much lower than what can be achieved using other public-key authentication protocols. In [15] researchers showed that computing 1024-bit RSA digital signature on 8-bit sensor node requires on the order of 90 seconds, and 10 seconds for signature verification. Moreover, [27] used signature based Elliptic Curve Cryptography on 8-bit sensor node generating a 160-bit signature requires on the order of 20 seconds and around 40 seconds for the verification.

As for the energy consumption, in [8] they showed that it is possible to design public key encryption architectures with power consumption of less that $20\mu W$. They compared

two architectures, Rabin Scheme and NtruEncrypth and the results showed that Rabin scheme has no significant disadvantages compared to NtruEncrypth.

However, SNEP and $\mu$TESLA [19] used only symmetric keys techniques to provide security. The main problem is that they require each node to be time synchronized with the base station and require key management functions and ample storage. This also causes a delay in the authentication process and might not be practical for real-time sensitive WBAN applications. Further, Merkle-Winternitz signature [17] used efficient one-time signature constructions that are fast to be computed on sensor networks. There problem is that they require high communication overhead on the order of 100-200 bytes per signature.

# 6. CONCLUSIONS

Security is a critical issue for WBAN due to the sensitivity of their applications. Medical staff located at the cloud side usually sends important commands to the actuator WBAN nodes to perform critical actions. The authenticity and integrity of these commands is the most critical security issue. In this paper, we deployed a light-weight public-key authentication scheme for cloud-based WBAN systems. To prove its efficiency the modified Rabin was implemented with different hardware settings using Tmote Sky motes. The modified algorithm with and without the parallel settings are also implemented on FPGA to evaluate its design and performance. The implementation results show that secure, direct, instant, and authenticated commands can be delivered from the medical staff located at the cloud side to the WBAN nodes located in/on the human body. Moreover, the suggested parallel setting of the modified Rabin signature generation significantly reduced the delays(by almost 80%), which is a critical issue in WBAN applications.

# 7. REFERENCES


[1] Altera inc., quartus ii introduction using verilog designs. ftp://ftp.altera.com/up/pub/Altera-Material /9.1 /Tutorials /Verilog /Quartus-II-Introduction.pdf.

[2] J. H. Ahnn and M. Potkonjak. mhealthmon: Toward energy-efficient and distributed mobile health monitoring using parallel offloading. *Journal of medical systems*, 37(5):1–11, 2013.

[3] M. Barua, R. Lu, and X. Shen. Sps: Secure personal health information sharing with patient-centric access control in cloud computing. In *Proc. of GLOBECOM*, pages 647–652, Dec 2013.

[4] P. Brandão. *Abstracting information on body area networks*. PhD thesis, University of Cambridge, 2012.

[5] K. Divi and H. Liu. Modeling of wban and cloud integration for secure and reliable healthcare. In *Proc. of BodyNets*, pages 128–131, 2013.

[6] G. Fortino, G. Di Fatta, M. Pathan, and A. Vasilakos. Cloud-assisted body area networks: state-of-the-art and future challenges. *Wireless Networks*, pages 1–14, 2014.

[7] J.-P. K. G. Gaubatz and B. Sunar. Public key cryptography in sensor networks - revisited. In *Proc. of ESAS*, pages 2–18, Springer, Heidelberg, 2004. Lecture Notes in Computer Science.

[8] G. Gaubatz, J. Kaps, and B. Sunar. Public key cryptography in sensor networks - revisited. In *Proc. of ESAS*, pages 2–18, Springer, Heidelberg, 2004. Lecture Notes in Computer Science.

[9] N. D. Han, L. Han, D. M. Tuan, H. P. In, and M. Jo. A scheme for data confidentiality in cloud-assisted wireless body area networks. *Information Sciences*, 2014.

[10] J.-H. Hauer, V. Handziski, and A. Wolisz. Experimental study of the impact of wlan interference on ieee 802.15.4 body area networks. In *Wireless Sensor Networks*, volume 5432 of *Lecture Notes in Computer Science*, pages 17–32. Springer Berlin Heidelberg, 2009.

[11] T. Hayajneh, G. Almashaqbeh, S. Ullah, and A. Vasilakos. A survey of wireless technologies coexistence in wban: analysis and open research issues. *Wireless Networks*, pages 1–35, 2014.

[12] D. He, C. Chen, S. Chan, J. Bu, and A. Vasilakos. Retrust: Attack-resistant and lightweight trust management for medical sensor networks. *Information Technology in Biomedicine, IEEE Transactions on*, 16(4):623–632, 2012.

[13] N. A. Jacob, V. Pillai, S. Nair, D. T. Harrell, R. Delhommer, B. Chen, I. Sanchez, V. Almstrum, and S. Gopalan. Low-cost remote patient monitoring system based on reduced platform computer technology. *Telemedicine and e-Health*, 17(7):536–545, 2011.

[14] B. Latré, B. Braem, I. Moerman, C. Blondia, and P. Demeester. A survey on wireless body area networks. *Wireless Networks*, 17(1):1–18, 2011.

[15] D. J. Malan, M. Welsh, and M. D. Smith. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In *Proc. of SECON*, 2004.

[16] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Waterloo, Canada, 2001.

[17] R. Merkle. A certified digital signature. In *Proc. of CRYPTO*, pages 218–238. Springer New York, 1989.

[18] T. O'Donovan, J. O'Donoghue, C. Sreenan, D. Sammon, P. O'Reilly, and K. O'Connor. A context aware wireless body area network (ban). In *Proc. of PervasiveHealth*, pages 1–8, April 2009.

[19] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. E. Culler. Spins: Security protocols for sensor networks. *Wireless networks*, 8(5):521–534, 2002.

[20] M. Rabin. *Digitalized signatures and public key functions as intractable as factorization*. Massachusetts Institute of Technology, Reading, Massachusetts, 1979.

[21] M. Rahimi, J. Ren, C. Liu, A. Vasilakos, and N. Venkatasubramanian. Mobile cloud computing: A survey, state of art and future directions. *Mobile Networks and Applications*, 19(2):133–143, 2014.

[22] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[23] F. Touati and R. Tabish. U-healthcare system: State-of-the-art review and challenges. *Journal of medical systems*, 37(3):1–20, 2013.

[24] S. Ullah, H. Higgins, B. Braem, B. Latre, C. Blondia, I. Moerman, S. Saleem, Z. Rahman, and K. S. Kwak. A comprehensive survey of wireless body area networks. *Journal of Medical Systems*, 36(3):1065–1094, 2012.

[25] J. Wan, C. Zou, S. Ullah, C.-F. Lai, M. Zhou, and X. Wang. Cloud-enabled wireless body area networks for pervasive healthcare. *Network, IEEE*, 27(5):56–61, September 2013.

[26] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, and A. V. Vasilakos. Security and privacy for storage and computation in cloud computing. *Information Sciences*, 258(0):371 – 386, 2014.

[27] T. Wollinger, J. Pelzl, V. Wittelsberger, C. Paar, G. Saldamli, and C. Koc. Elliptic and hyperelliptic curves on embedded up. In *Proc. of TECS*, 2004.

[28] Z. Zhang, H. Wang, A. Vasilakos, and H. Fang. Ecg-cryptography and authentication in body area networks. *Information Technology in Biomedicine, IEEE Transactions on*, 16(6):1070–1078, 2012.