International Journal of Advanced Smart Convergence Vol.7 No.3 101-109 (2018)
http://dx.doi.org/10.7236/IJASC.2018.7.3.101

IJASC 18-3-12

Adaptive Filtering Scheme for Defense of Energy Consumption Attacks against Wireless Computing Devices

Wan Yeon Lee

Dept. of Computer Science, Dongduk Women's University, Seoul 136-714, South Korea,
wanlee@dongduk.ac.kr

Abstract

In this paper, we propose an adaptive filtering scheme of connection requests for the defense of malicious energy consumption attacks against wireless computing devices with limited energy budget. The energy consumption attack tries to consume the battery energy of a wireless device with repeated connection requests and shut down the wireless device by exhausting its energy budget. The proposed scheme blocks a connection request of the energy consumption attack in the middle, if the same connection request is repeated and its request result is failed continuously. In order to avoid the blocking of innocuous mistakes of normal users, the scheme gives another chance to allow connection request after a fixed blocking time. The scheme changes the blocking time adaptively by comparing the message arriving ate during non-blocking period and that during blocking period. Evaluation shows that the proposed defense scheme saves up to 94% energy consumption compared to the non-defense case.

Keywords: *Energy Consumption, Malicious Attack, Connection Blocking, Filtering Defense, Adaptive Scheme*

1. Introduction

Wireless embedded systems essentially rely on the energy supplied by batteries and sustain their operation until the depletion of available energy. Hence energy management has become a critical issue in wireless embedded systems, compared with traditional computing platforms. For such resource-constraint systems, efficient cryptographic algorithms have been developed recently [1, 2, 3]. Recently developed algorithms considered tradeoff between the strength of security and the energy dissipation rate. However, the previous cryptographic algorithms focused on fast algorithms but did not consider the fast dissipation of the limited energy budget.

In this paper, we introduce a new attack, called *energy consumption attack*, that malicious tries to shut down a wireless device by exhausting its battery energy with repeated connection requests. If the available energy budget of the attached battery is suddenly exhausted, then its function stops during critical operation and may cause a fatal loss. For example, sudden termination of all sensors,

Manuscript Received: August. 10, 2018 / Revised: August. 17, 2018 / Accepted: August. 22, 2018
Corresponding Author: wanlee@dongduk.ac.kr
Tel: +82-2-940-4685, Fax: +82-2-940-4170
Dept. of Computer Science, Dongduk Women's University

monitoring any approach, due to lack of available battery energy may fail to detect an intruder during the night.

Because low energy dissipation of wireless devices is important, recently developed embedded systems are allowed to turn into dormant status in which unused components are powered off except essential components powered on. When wireless devices are idle without requested computation, they change into the dormant status. The energy consumption rate in the dormant status is very low [4]. An attacker can accelerate the energy dissipation of wireless devices by preventing them from changing into the dormant status. It is easily achieved by making them busy with unnecessary computation.

In order to defense the energy consumption attack, we propose an adaptive filtering scheme that blocks the connection request of the energy consumption attack. The scheme checks the result of each connection request, and sends a blocking request message to its gateway device when the number of continuous request failures exceeds a predefined threshold value. Then the gateway device drops the connection request message arrived from the client. On the other hand, innocent users may possibly send wrong request messages continuously. In this case, they are allowed to send request message again with some penalty. The scheme gives the user another chance to send the connection request after a predefined duration, in order to accommodate innocuous mistakes of normal connection request while blocking the energy consumption attack. If the user sends wrong request messages again continuously in the second chance, the blocking duration increases to prevent attackers from utilizing another chance. In this paper, we propose an adaptive formula to determine the blocking duration for suspicious connection requests. The formula analytically calculates the message arrival rate during blocking period and that during non-blocking period. If the message arrival rate during blocking period is much larger than that during non-blocking period, then the blocking duration increases because of its suspicious anomaly. In contrast, if the message arrival rate during blocking period is much smaller than that during non-blocking period, then the blocking duration decreases.

Most of previous energy-efficient designs focused on maximizing the residual lifetime of their battery while satisfying the minimal performance required inside [4, 5, 6]. However, they did not consider the malicious energy consumption attacks arriving outside. A few studies dealt with the defense mechanism against the malicious resource-consuming attacks arriving outside [7, 8]. However, they considered only the attack for *computing power*, but not for *battery energy budget*. The scheme proposed in this paper prevents the energy consumption attack while avoiding the false positive detection induced from innocuous mistakes of normal users. To evaluate the proposed scheme, we implement a prototype system and measure its energy consumption against the energy consumption attack. Evaluation results show that the proposed scheme saves up to 94% energy consumption compared to the non-defense case.

The rest of this paper is organized as follows; Section 2 explains the system model considered in our study. Section 3 describes the proposed scheme in detail. Section 4 shows evaluation results. Section 5 provides concluding remarks.

2. System Model

In the considered system, wireless embedded devices always communicate to outside clients through a gateway. Attackers in outside areas must communicate to wireless embedded devices through the gateway. It is assumed that the gateway is connected to a wired power supplier and thus has sufficient energy budget.

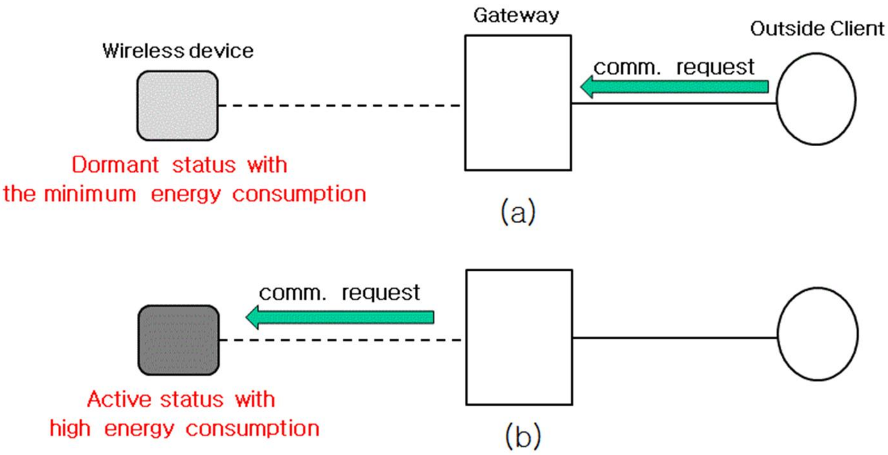


Figure 1. System Model

The wireless devices provide multiple operation modes in which the actively working components among all components are different with different corresponding energy consumption. Recently developed devices support this kind of dynamic operation mode transition, where they can choose one of the energy-aware operation modes and change into another operation mode at any time [9, 10]. Figure 1 shows an example of the considered system model. In this example, there are two operation modes: dormant status and active status. Figure 1(a) shows the dormant status. In the dormant status, a wireless device activates a part of components for essential processing such as periodically checking inbound message arrivals with the minimum energy consumption. Figure 1(b) shows the active status. In the active status, a wireless device activates all components for full processing.

3. Proposed Scheme

In the proposed scheme, communication requests are classified into the normal request and the abnormal request. The normal communication request results in the successful connection. The normal communication request follows a predefined communication protocol and includes exact data for the connection authentication such as client ID and password. The other communication requests except the normal requests belong to the abnormal communication request. The abnormal communication request results in the failure of connection authentication.

When the number of abnormal requests continuously arriving with the same source address exceeds a threshold, the wireless device defines the source address as a deny address and sends it to the gateway. The source address is IP address, MAC address and so on. Figure 2(a) shows the enrollment procedure of a deny address, where a wireless device sends a deny address to the gateway. Then the gateway records all received deny addresses into a list, called *deny address list*. The gateway checks the source address of an outside incoming message and drops the arriving message if its source address matches one of the deny address list, such as shown in Figure 2(b). If the abnormal communication requests are allowed without dropping by the gateway, then the wireless device must keep the active status to check the arriving message. On the other hand, if the abnormal requests are blocked by the gateway, then the wireless device can turn into the dormant status shown in Figure 2(b).

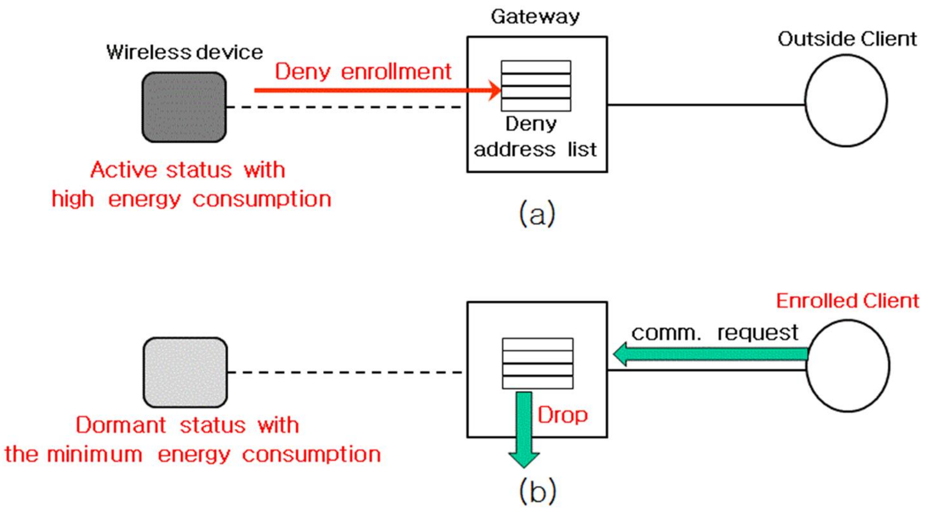


Figure 2. Outline of Proposed Scheme

In some cases, innocent outside clients may send wrong communication requests continuously, for example, authentication requests with forgotten password. In this case, the succeeding communication request is inevitably handled as the energy consumption attack, which is referred to as the false positive detection of energy consumption attack. In order to avoid the false positive detection of energy-consumption attacks, the proposed scheme gives another change with a penalty of waiting time. The gateway allows the passing of each deny source address when a pre-defined time passes after its enrollment time. If the client sends wrong communication requests again continuously in the second chance, its source address is enrolled into the deny address list and its blocking duration increases in order to prevent attackers from utilizing another chance repeatedly. When the number of enrollments as a deny address exceeds a threshold, the source address is permanently blocked. With this adaptively controlled blocking, the scheme accommodates the innocuous mistake of normal requests while defending the energy consumption attack.

The proposed scheme increases or decreases the blocking duration of each deny address by comparing its arriving ratio during blocking period and that during non-blocking period. If the arriving ratio during blocking period is larger than that during non-blocking period, then the next blocking time increases because it implies an alarm of suspicious pattern for the energy consumption attack. On the other hand, if the arriving ratio during blocking period is smaller than that during non-blocking time, then the blocking time decreases because it implies a release of suspicious pattern for the energy consumption attack.

In this paper, we propose an adaptive formula to determine the next blocking time for suspicious connection requests. The next blocking time of each deny address can be formally determined with the following notations:

- Arriving ratio during the non-blocking period: $\boldsymbol{\lambda}$
- Blocking period: $\boldsymbol{D}$
- Passed time after the blocking period starts: $\boldsymbol{T}$
- Passed time without arriving message during the blocking period: Δ

- Number of arriving message during the time $\boldsymbol{T}$: $\boldsymbol{\alpha}$

Then the probability that there is no message arriving during the time $\boldsymbol{T}$ is $e^{-\lambda \boldsymbol{T}}$ according to the Poisson analysis and it is denoted as $\boldsymbol{P}_1$. Also the probability that $\boldsymbol{\alpha}$ messages arrive during the time $\boldsymbol{T}$ is

$$(\lambda \boldsymbol{T})^{\alpha} \cdot e^{-\lambda \boldsymbol{T}} / (\boldsymbol{\alpha}!) \tag{1}$$

Equation (1) is given from the Poisson analysis and it is denoted as $\boldsymbol{P}_2$.

In the proposed scheme, the gateway calculates $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ for each deny address periodically (for example, 5 minutes). When two conditions are satisfied, the gateway decreases the next blocking time. One is that Δ is larger than a given threshold period (for example, one minute), and the other is that $\boldsymbol{P}_1$ is smaller than a given threshold probability (for example, 30%). These two conditions imply that arriving ratio during the blocking period is much smaller than that during the non-blocking period for the given sufficient period. The decreasing amount of the next blocking time is implementation-dependent. A simple approach is that the blocking time is decreased by a fixed ratio (for example, $\boldsymbol{D} = \boldsymbol{D} - \boldsymbol{D} \cdot 20\%$). An elaborate approach utilizing the calculated $\boldsymbol{P}_1$ is that the blocking time is decreased inversely proportional to the probability $\boldsymbol{P}_1$ (for example, $\boldsymbol{D} = \boldsymbol{D} - \boldsymbol{D}/(0.01 \cdot \boldsymbol{P}_1)$).

In contrast, the gateway increases the next blocking time when two conditions are satisfied. One is that $\boldsymbol{\alpha}$ is larger than a given number (for example, 5), and the other is that $\boldsymbol{P}_2$ is smaller than a given threshold probability (for example, 10%). These two conditions imply that arriving ratio during the blocking period is much larger than that during the non-blocking period for sufficient arriving messages. The increasing amount of the blocking is also implementation-dependent. A simple approach is that the blocking time is decreased by a fixed ratio (for example, $\boldsymbol{D} = \boldsymbol{D} + \boldsymbol{D} \cdot 20\%$). An elaborate approach utilizing the calculated $\boldsymbol{P}_2$ is that the blocking time is increased inversely proportional to the probability $\boldsymbol{P}_2$ (for example, $\boldsymbol{D} = \boldsymbol{D} + \boldsymbol{D}/(0.01 \cdot \boldsymbol{P}_2)$).

4. Evaluation

For evaluation of the proposed scheme, we implement a prototype system. Figure 3 shows the configuration of the prototype system. An Arduino Uno board plays a role of the wireless device. An LG Optimus-4 smartphone with Android 7.1 plays a role of the gateway. A Toshiba notebook with CentOS 6.5 Linux plays a role of the energy consumption attack as the outside client.

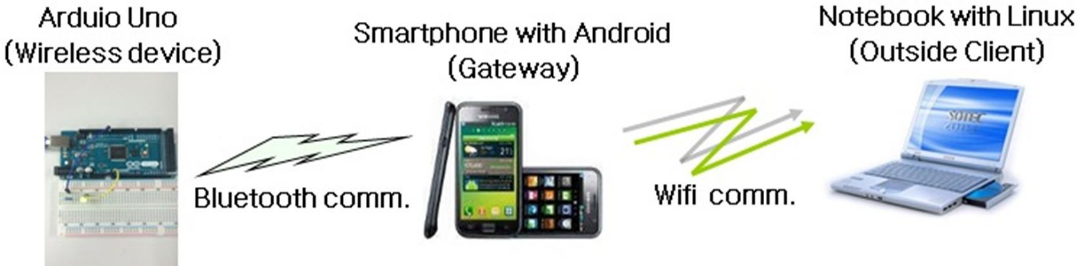


Figure 3. Configuration of Prototype System

The Arduino board relies on the energy of a portable battery. The Arduino board communicates with the smartphone through a Bluetooth wireless channel. The smartphone communicates with the outside clients through a built-in Wifi wireless channel. The notebook sends the connection request message continuously per 500msec to the Arduino board, in order to accelerate the energy consumption of the battery of the

Arduino board.

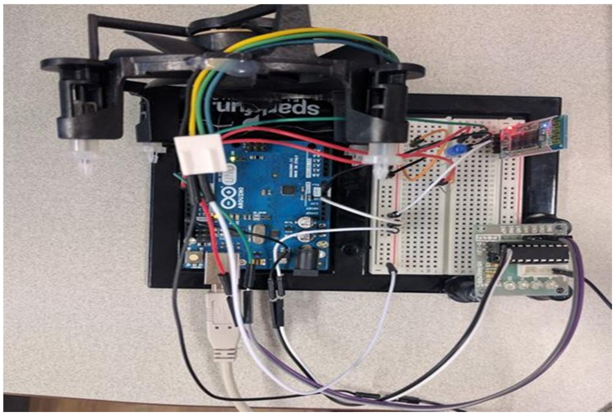
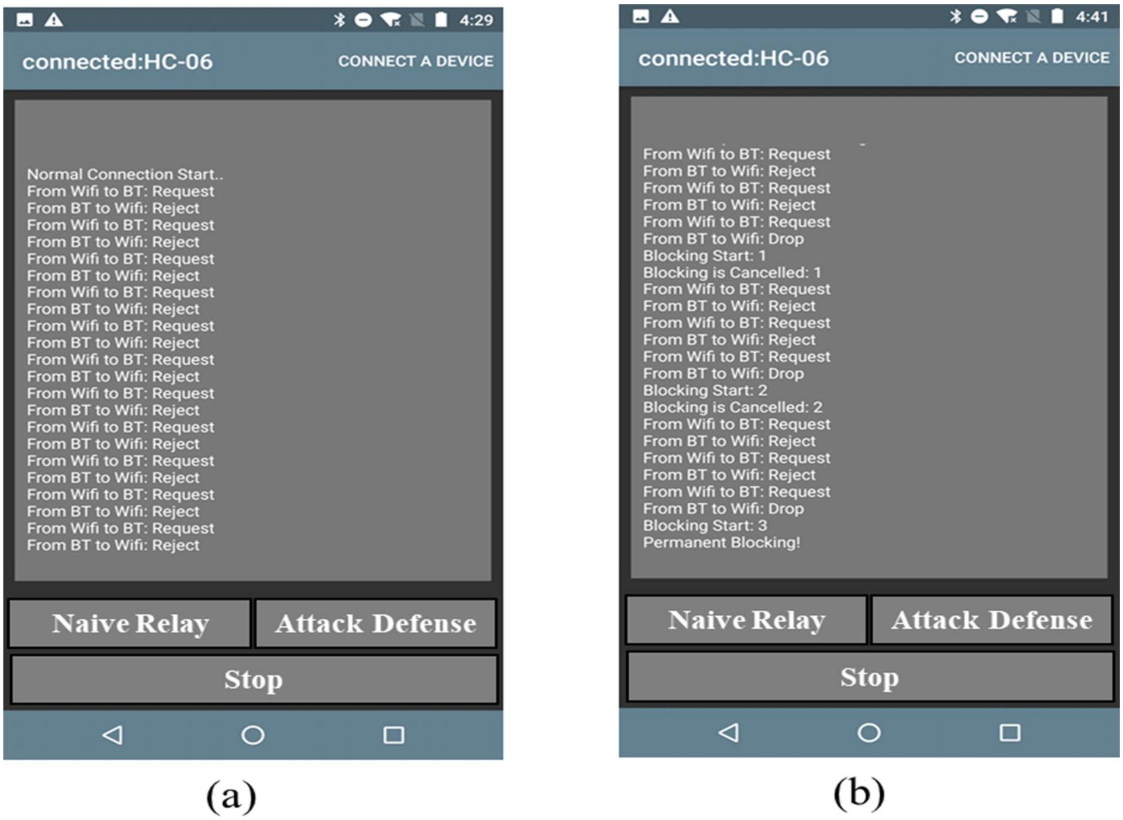


Figure 4. Outline of Proposed Scheme

Figure 4 shows the detail configuration of the wireless device implemented with Arduino Uno board. The board consists of Intel DC-motor fan, Parts & Kits HC-06 Bluetooth chip, and LED. The board is connected to a portable battery, SMODO MS842S model of Myung Sung Inc. In the active status, all components (MCU of Arduino board, DC-motor, Bluetooth chip, and LED) are activated with high energy consumption. In the dormant status, only MCU of Arduino board is activated and the Bluetooth chip is ready with low energy consumption. If three abnormal requests continuously arrive from the same source address, then the board sends the source address to the smartphone with a mark of the deny address, as shown in Figure 2(a).



(a)

```
Normal Connection Start..
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Reject
```

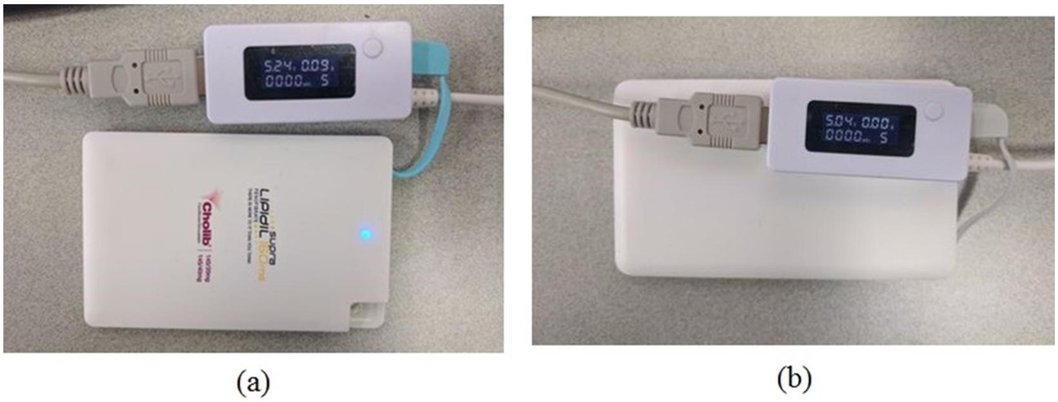
(b)

```
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Drop
Blocking Start: 1
Blocking is Cancelled: 1
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Drop
Blocking Start: 2
Blocking is Cancelled: 2
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Reject
From Wifi to BT: Request
From BT to Wifi: Drop
Blocking Start: 3
Permanent Blocking!
```

Figure 5. Working Display of Gateway Smartphone

Figure 5 shows the working display of the smartphone playing a role of the gateway. Figure 5(a) shows the GUI display of the smartphone when the smartphone does not manage the deny address list, which is referred to as the baseline method in this paper. In this case, the smartphone just relays a message arriving from the notebook to the Arduino board, and vice versa. The GUI display was implemented with an android program upon the Android Studio IDE. Figure 5(b) shows the GUI display when the smartphone manages the deny address list based on the proposed scheme described in Section 3. In this case, the smartphone blocks a message with the deny address list by dropping the message. When the Arduino board sends the source address to the smartphone with a mark of the deny address, the smartphone adds it into the deny address list. Whenever a message arrives from the outside client, the smartphone checks whether or not its source address matches one of the deny address list.

Initially, in Figure 5(b), each deny address remains in the deny address list for 10 seconds. When 10 seconds has passed after its enrollment time, the deny source address is removed one in order to accommodate innocuous mistakes of outside clients. Also its number of enrollment is set to one in order to record the number of enrollments. From that point, the notebook can send a request message to the Arduino board continuously. If the Arduino board sends again the deny address to the smartphone after three abnormal requests, the smartphone adds it again into the deny address list with an increased blocking time of 30 seconds. Similarly, the deny address is removed after 30 seconds and its number of enrollment is set to two. Finally, if the Arduino board sends again the deny address, then the smartphone it again into the deny address list and the blocking time is set to infinite.



(a) (b)

Figure 6. Measurement of Battery Energy Consumption

Figure 6 shows the measurement procedure of energy consumption rate of the battery connected to the Arduino board. The energy consumption rate of the SMODO MS842S battery is measured with the KCX-017 measurement device of Camping & Farming Inc. Figure 6(a) shows the energy consumption rate when the Arduino board is in the active status. In this figure, the voltage is roughly 5.24V and the current is roughly 0.085A (the average of 0.08A and 0.09A). Figure 6(b) shows the energy consumption rate when the Arduino board is in the dormant status. In this figure, the voltage is roughly 5.04V and the current is roughly 0.005A (the average of 0.00A and 0.01A). In the baseline method, the Arduino board is always in the active status. In contrast, in the proposed scheme, the Arduino board is in the active status only when the gateway does not block the arriving message. The gateway allows the message passing for less than 6 seconds (approximately 2 seconds for each passing chance and 6 seconds for three passing chances) even though it gives two more chances in order to accommodate innocuous mistakes of outside clients. For the total running time of ten minutes, the baseline method consumes approximately 5.24V × 0.085A × 600secs = 267.24

Joules. The proposed scheme consumes less than approximately (5.24V × 0.085A × 6secs) + (5.04V × 0.005A × 594secs) = 17.6412 Joules. Hence, the proposed scheme saves about 94% energy consumption of the baseline method, where the energy saving ratio is 100 × (267.24 – 17.6412)/267.24 ≈ 93%. If the running time becomes infinite, the energy saving ratio will is equal to 100 × (energy consumption rate in the active status – energy consumption rate in the dormant status) / (energy consumption rate in the active status), that is, (5.24V × 0.085A – 5.04V × 0.005A)/(5.24V × 0.085A) ≈ 94% in Figure 6.

Table 1. Comparison of Energy Consumption for Different Batteries

Battery Model	SMODO MS842S	CABSTONE Pocket Power	Tronic Powerbank2600	Supra LIPIdIL160mG
Baseline Method	267.24 Joules	259.08 Joules	293.30 Joules	302.14 Joules
Proposed Scheme	17.64 Joules	19.21 Joules	24.43 Joules	27.87 Joules

Table 1 shows the total energy consumption of four different portable battery models for the running time of ten minutes. In the average, the proposed scheme saves approximately 92% energy consumption compared to the baseline method. The proposed scheme saves maximally 93% energy consumption when the SMODO MS842S battery model is used. It is clear that this energy saving benefit of the proposed scheme becomes larger when the total running time increases.

5. Conclusions

In this paper, we introduce a new type of attack called the energy consumption attack. The attack may consume the energy of a wireless device with repeated authentication processing and shut down the wireless device by exhausting its energy budget. We also propose an adaptive filtering scheme that blocks malicious energy consumption attacks targeting wireless computing devices with limited energy budget. The scheme prevents the sudden shutdown failure of the wireless device caused from depletion of its battery energy budget. In order to avoid the false positive detection induced from innocuous mistakes of normal users, the proposed scheme gives another change to go through after a limited blocking time. Finally, we describe an adaptive formula of the blocking time, which changes by comparing the message arriving rate during non-blocking period and that during blocking period. Evaluation shows that the proposed defense scheme saves up to 94% energy consumption compared to the non-defense baseline case.

Acknowledgement

This research was supported by the sabbatical leave grant of the Dongduk Women's University, 2017.

References

[1] Thomas Wollinger, Jorge Guajardo and Christof Paar, "Cryptography in Embedded Systems: An Overview," Proceedings of the Embedded World Exhibition and Conference (Feb. 2003), pp. 735-744.

[2] Wei Jiang, Zhenlin Guo, Yue Ma and Nan Sang, "Research on Cryptographic Algorithms for Embedded Real-time Systems: A Perspective of Measurement-Based Analysis," IEEE 14th International Conference on High Performance Computing and Communications (Jun. 2012), pp. 1495-1501.

[3] Meikang Qiu, Wenzhong Gao, Min Chen, Jian-Wei Niu and Lei Zhang, “Energy Efficient Security Algorithm for Power Grid Wide Area Monitoring System,” IEEE Transactions on Smart Grid (Dec. 2011), vol. 2, no. 4, pp. 715-723.
DOI: http://10.1109/TSG.2011.2160298

[4] Wan Yeon Lee, “Energy-efficient Scheduling of Periodic Real-time Tasks on Lightly Loaded Multicore Processors,” IEEE Transactions on Parallel and Distributed Systems (Mar. 2012), vol. 23, no. 3, pp. 530-537.
DOI: http://10.1109/TPDS.2011.87

[5] Eduardo Cuervo, Aruna Balasubramanian and Dae-ki Cho, “MAUI: Making Smartphones Last Longer with Code Offload,” Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (Jun. 2010), pp. 49-62.

[6] G. Wang, W. Li and X. Hei, “Energy-Aware Real-time Scheduling on Heterogeneous Multi-Processor,” Annual Conference on Information Sciences and Systems (Mar. 2015), pp. 1-7.

[7] R. Smith, C. Estan and S. Jha, “Backtracking Algorithmic Complexity Attack against a NIDS,” The 22nd Annual Computer Security Applications Conference (Dec. 2006), pp. 89-98.

[8] S. Khan and I. Traore, “A Prevention Model for Algorithmic Complexity Attacks,” The 2nd International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (July 2005), pp. 160-173.

[9] Qiang Huang, Hisashi Kobayashi, Bede Liu, Daqing Gu and Jinyun Zhang, “Energy/Security Scalable Mobile Cryptosystem,” TR-2003-79, MERL - A Mitsubishi Electrical Research Laboratory (Feb. 2004), http://www.merl.com.

[10] Wan Yeon Lee, Yun-Seok Choi, "Optimization of ARIA Block-Cipher Algorithm on Embedded Systems with 16-bits Processors," International Journal of Internet, Broadcasting and Communication, vol. 8, no. 1, pp. 88-98, Feb. 2016.
DOI: http://dx.doi.org/10.7236/IJIBC.2016.8.1.42