PKC-Based DoS Attacks-Resistant Scheme in Wireless Sensor Networks

Daehee Kim, *Student Member, IEEE*, and Sunshin An, *Member, IEEE*

***Abstract*— Public-key cryptography (PKC) operations are heavy to resource-constrained sensor nodes. Therefore, attackers can cripple a sensor node by forcing it to perform a large number of false PKC operations. In this letter, we propose a fully distributed and effective scheme that randomly drops extra PKC request messages beyond its processing capability. Our scheme is not only resistant to PKC-based denial-of-service attacks, but also energy-efficient.**

***Index Terms*— DoS attacks, public-key cryptography, random drop, wireless sensor networks.**

I. Introduction

SINCE the feasibility of public-key cryptography (PKC) was proven [1], PKC has been getting more and more attentions in wireless sensor networks (WSNs) due to the fact that PKC can easily resolve two of the fundamental and difficult problems, authentication and symmetric key distribution, by using digital signatures and Diffie-Hellman key exchange [2], [3]. Although most researches employ elliptic curve cryptography (ECC) that has much smaller overhead than existing PKC algorithms such as RSA, ECC operations are still heavy to resource-constrained sensor nodes. For example, a MICAz mote takes approximately 1.62 seconds and consumes 38.88 mJ for an elliptic curve digital signature algorithm (ECDSA) signature verification [4]. Thus, malicious attackers can overwhelm sensor nodes and deplete their energy by simply sending false messages leading to PKC operations, which are called *PKC-based denial-of-service* (*DoS*) *attacks*. Message-specific puzzles are adapted to mitigate PKC-based DoS attacks [4]. Although this scheme efficiently filters false PKC messages, the sender has large overhead to make a puzzle, and keys for verifying the puzzle must be pre-distributed to all sensor nodes. In this letter, we propose a fully distributed and effective scheme to keep the overall load of PKC operations under the capacity of each sensor node by randomly dropping additional PKC request messages. Furthermore, we adjust an accepting probability according to the residual energy to prolong the lifetime of a sensor node.

II. Proposed Scheme

A. Network Model

First, we call a message causing a PKC operation a *pkc_msg*, which can include an ECDSA signature or lead to elliptic curve Diffie-Hellman (ECDH) key exchange. In this letter, we consider a single sensor node, called a *receiver* of *pkc_msgs*, which has several neighbor nodes, some of which are legitimate nodes and the others are attacker nodes, or simply attackers. We assume that a receiver is able to process C *pkc_msgs*/s and legitimate neighbor nodes send a total of true α *pkc_msgs*/s where α is less than $0.5C$. To disable the receiver, attacker nodes are assumed to be able to send false β $(>C)$ *pkc_msgs*/s. Finally, we do not take packet losses due to the channel condition and collision into account because they can be recovered through low-layer retransmission.

B. PKC-Based DoS Attacks-Resistant Scheme

The main goal of this letter is to keep the processing overhead of incoming *pkc_msg*s under the budget of the receiver. This can be formulated as

$$\alpha + \beta \leq C = k/T_{pkc} \tag{1}$$

where T_{pkc} is the time it takes for a sensor node to perform one PKC operation, and k is the fraction of capacity dedicated to PKC operations, thus $0 < k < 1$. Without attackers, that is, $\beta = 0$, (1) is always satisfied because $\alpha < C$ from the assumption. If attackers exist and inject β *pkc_msgs*/s, we must drop extra *pkc_msgs* beyond C to meet (1). The simple drop-tail mechanism may be a candidate, but it can be unfair under the burst DoS attacks. Instead, we drop extra *pkc_msg*s randomly with a specific probability. In order to compute a probability of accepting *pkc_msgs*, the receiver defines a time window W of an interval T. Now, time is divided into $W(i)$ where $i = 1, 2, 3, \cdots$. Denote the number of incoming *pkc_msgs* during $W(i)$ by $N(i)$. Since the expected number of *pkc_msgs* accepted during each window must not exceed $C \cdot T$, the probability of accepting each *pkc_msg* is calculated as follows.

$$P_{accept} \cdot N(i) \leq C \cdot T \rightarrow P_{accept} \leq \frac{C \cdot T}{N(i)} \tag{2}$$

Since the energy is the most important and scarcest resource in sensor nodes, we further try to aggressively drop the incoming *pkc_msg*s to prolong the lifetime of the receiver as the residual energy decreases only in case that the number of incoming *pkc_msgs* exceeds $C \cdot T$. Note that any incoming *pkc_msg* must not be discarded in case of no attackers. Thus, the receiver accepts each incoming *pkc_msg* with the following probability.

$$P_{accept} = \begin{cases} 1, & if\ N(i) \leq C \cdot T \\ \dfrac{C \cdot T}{N(i)} \cdot \dfrac{E_{res}}{E_{init}}, & if\ N(i) > C \cdot T \end{cases} \tag{3}$$

where E_{init} is the initial energy, and E_{res} is the residual energy of the receiver. To apply this probability to the incoming *pkc_msgs*, we adopt a reservoir sampling [5] which is a memory-efficient algorithm for randomly selecting n samples from a large list S with a memory of n samples only. Similarly, the receiver first chooses $\lfloor C \cdot T \rfloor$ *pkc_msgs* out of $N(i)$ in case of $N(i) > C \cdot T$ with a memory of $\lfloor C \cdot T \rfloor$ *pkc_msgs* only. The receiver then accepts each *pkc_msgs* with a probability

Manuscript received August 27, 2015; revised January 8, 2016; accepted January 14, 2016. Date of publication January 19, 2016; date of current version February 24, 2016. This work was supported by the National Research Foundation of Korea Grant funded through the Korean Government under Grant 2012K1A3A1A09026959. The associate editor coordinating the review of this letter and approving it for publication was Prof. Elena Gaura.

The authors are with Korea University, Seoul 02841, Korea (e-mail: dhkim@dsys.korea.ac.kr; sunshin@dsys.korea.ac.kr).

Digital Object Identifier 10.1109/JSEN.2016.2519539

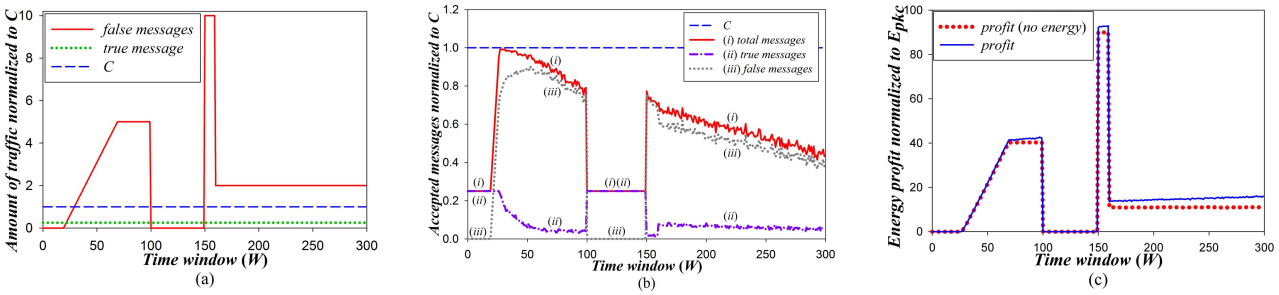

Fig. 1. Simulation results. (a) Input traffic pattern. (b) Normalized accepted messages. (c) Normalized energy profit.

of E_{res}/E_{init} to satisfy (3). Note that the receiver randomly drops true *pkc_msgs* as well as false *pkc_msgs* according to the number of incoming *pkc_msgs* and the residual energy. To justify our scheme in terms of energy consumption, we compare the benefit and the cost due to random drop. The cost of a dropped true *pkc_msg* is the energy required to transmit (by the sender) and receive (by the receiver) again, $E_{tx}+E_{rx}$. On the other hand, the benefit of a dropped false *pkc_msg* is the energy saved by not performing a PKC operation, E_{pkc}. From [6], E_{pkc} is much bigger than $E_{tx}+E_{rx}$ in a sensor node, thus we can express this relation as $E_{pkc} = K \cdot (E_{tx} + E_{rx})$ where $K > 1$. In case of $N(i) > C \cdot T$, the benefit and cost due to random drop is as follows.

$$Benefit = N(i) \cdot (1 - P_{accept}) \cdot \frac{\beta}{\alpha + \beta} \cdot E_{pkc} \quad (4)$$

$$Cost = N(i) \cdot (1 - P_{accept}) \cdot \frac{\alpha}{\alpha + \beta} \cdot (E_{tx} + E_{rx}) \quad (5)$$

By subtracting (5) from (4), and using $E_{pkc} = K \cdot (E_{tx}+E_{rx})$, we can show that the profit, *Benefit* – *Cost*, is always larger than 0 as follows.

$$Profit = \frac{N(i) \cdot (E_{tx} + E_{rx})}{\alpha + \beta} \cdot (1 - P_{accept}) \cdot \{K \cdot \beta - \alpha\} > 0 \quad (6)$$

where $K \cdot \beta > \alpha$ because $K > 1$, $\beta > \alpha$ from the assumption. Therefore, random dropping is always more energy-efficient than not doing it in case of $N(i) > C \cdot T$ which means under attack.

It is important to note how long it takes for a true *pkc_msg* to be accepted in the receiver because our scheme randomly drops true *pkc_msgs* as well as false *pkc_msgs*. Assuming legitimate neighbor nodes can recognize the drop of their true *pkc_msg* through the loss of high-layer acknowledgements and simply resend it after a fixed amount of time T_{out}, the average time it takes for one true *pkc_msg* to be accepted, R, is

$$R = T_{out} \cdot \sum_{j=1}^{\infty} j \cdot P_{accept}(1 - P_{accept})^{j-1} = T_{out}/P_{accept} \quad (7)$$

For lower P_{accept}, which means that there exist noticeable false *pkc_msgs*, R grows larger. To shorten R, we suggest the true sender transmits $K - 1$ copies of a true *pkc_msg* at a time when recognizing the drop of the first *pkc_msg*, which implies that at least one false *pkc_msg* was dropped from the assumption that a total of true α *pkc_msgs*/s is less than $0.5C$. Thus, we can get at least K energy gain from $E_{pkc} = K \cdot (E_{tx} + E_{rx})$. In this case, the probability of accepting at least one out of $K - 1$ *pkc_msgs* is

$$P = 1 - (1 - P_{accept})^{K-1} \quad (8)$$

Suppose P_{accept} is 0.1, implying that the rate of incoming false *pkc_msgs* is at least $9.5C$ without consideration of energy. Then, K required for $P = 0.99$ becomes 45 that is much smaller than K, 139, of MICAz with ECDSA-160 in case of 40 bytes [6]. In other words, a true *pkc_msg* in our scheme can reach the receiver with a probability of 0.99 through one retransmission, consuming less energy than not doing the random drop.

III. Performance Evaluation

In this section, we demonstrate through a simulation during the window size of 300 that our scheme is resistant to PKC-based DoS attacks and energy-efficient under the various traffic patterns which include a fixed true *pkc_msgs* of $0.25C$ and diverse patterns of false *pkc_msgs* such as gradual increase, burst, and steady-state as depicted in Fig. 1 (a). Fig. 1 (b) shows that our scheme always keeps accepted messages under the capacity despite excessive false *pkc_msgs*. Thus, we can say that our scheme is resistant to PKC-based DoS attacks. In addition, the number of accepted messages decreases as time elapses since our scheme adjusts P_{accept} depending on the residual energy. Especially, false *pkc_msgs* are discarded much more than true *pkc_msgs*, which makes our scheme more energy-efficient. Finally, the energy profit is shown in Fig. 1 (c) where our scheme is compared with the one that randomly drops the incoming *pkc_msgs* but does not take the residual energy into account, denoted by '(*no energy*)'. Both schemes have the positive energy profit, which implies that they are more energy-efficient than not doing random drop. However, our scheme can conserve more energy by considering the residual energy.

References

[1] 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. 1st IEEE Int. Conf. Sensor Ad Hoc Commun. Netw.*, Oct. 2004, pp. 71–80.

[2] Y. Liu, J. Li, and M. Guizani, "PKC based broadcast authentication using signature amortization for WSNs," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 2106–2115, Jun. 2012.

[3] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "PAuthKey: A pervasive authentication protocol and key establishment scheme for wireless sensor networks in distributed IoT applications," *Int. J. Distrib. Sensor Netw.*, vol. 2014, Jul. 2014, Art. ID 357430.

[4] P. Ning, A. Liu, and W. Du, "Mitigating DoS attacks against broadcast authentication in wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 4, no. 1, pp. 1–35, 2008.

[5] J. S. Vitter, "Random sampling with a reservoir," *ACM Trans. Math. Softw.*, vol. 11, no. 1, pp. 37–57, 1985.

[6] S. Peter, P. Langendorfer, and K. Piotrowski, "Public key cryptography empowered smart dust is affordable," *Int. J. Sensor Netw.*, vol. 4, nos. 1–2, pp. 130–143, 2008.