ESP-JETA
ESP Journal of Engineering & Technology Advancements
ISSN: 2583-2646 / Volume 2 Issue 1, January 2022 / Page No: 1-4
Paper Id: JETA-V2I1P101 / Doi: 10.56472/25832646/JETA-V2I1P101

*Original Article*

# An Effective Malware Detection Algorithm for WSN

**Shamsuddin Ali Akbari A J[1], Mohammed Ali Hussian[2]**
[1]*Research scholar, Department of ECE, Ain Shams University. Egypt*
[2]*Senior Lecturer, Department of ECE, Ain Shams University. Egypt*

Received Date: 16 November 2021 Revised Date: 19 December 2021 Accepted Date: 06 January 2022

***Abstract:*** *Wireless sensor network used as a main element in today world. The sensor network transfers sensed data wireless to the other nodes in the presence of attackers. Attackers in the node affect network performance by dropping and vanishing the packets. In this work, an effective malfunction node detection algorithm is used with malware prevention. The proposed algorithm is implemented using ns2 toll and compared in terms of delay, packet and energy.*

***Keywords:*** *Malware, WSN.*

## INTRODUCTION

Wireless sensor networks (WSNs for short) are wireless networks constituted by several (in some cases thousands) smart devices called sensors capable of computation, communication and sensing. These networks have a wide range: from military applications to industrial applications through environmental, healthcare, multimedia or daily life applications (Fahmy, 2016). Specifically, they may be vulnerable to proximity malware infection since these networks usually lack of the adequate security countermeasures. In addition, a wireless connection requires a greater security configuration. The fight against malware can be performed both implementing efficient malware detection and removing tools, as well as designing software tools to simulate the behavior of malware propagation in WSNs.

This work deals with the second strategy. Each simulation software tool is based on the computational implementation of a mathematical (theoretical) and in the last few years there have appeared several proposals in the scientific literature introducing mathematical models for malware propagation in WSNs. Note that this network administrators with a solution to learn about the effect of a malware attack to learn about the effect of a malware attack on the network. In this sense, data protection is essential and the most important asset for any company, and for which many attackers are lurking.

### *Description of WSN*

A WSN is a wireless network defined by a group of a large number of smart minitature sensor nodes (motes) that are able to sensing, communication and computation (Zhao and Guibas, 2004). This technology is a low-cost solution to a great variety of problems in diverse research areas, and its main function is to collect all type of data (temperature, Sound, vibrations, etc.) Through specialized sensors, process and store this information, and finally, to forward it to a base station. Usually the low-power protocol ZigBee (based on the standard 802.15.4) is used in the WSNs for wireless communications. It uses the PHY and MAC layers. The first system with all characteristics of sensor networks was the Surveillance Sound System (SOSUS), which was designed and deployed – by means of sunken buoys – by the United States to detect and track soviet submarines during the Cold War. Nevertheless, the first applications of WSNs were originated around the 80's under the investigations of the Defense Advance Research Project Agency (DARPA, USA), in a project known as Distributed Sensor Networks (RDS).

The WSNs based on ZigBee standard are mainly composed of four basic elements (see Figure 1): 1. Sensor nodes: its main function is to collect data directly from the environment and convert it into electrical signals. They are also called Zigbee end devices or reduced function devices (RFD). These nodes are usually grouped in clusters. 2. Cluster-head nodes: These nodes are configured to work as middle point between the clusters and the rest of the network. They are also known as Zigbee routers or full function devices (FFD). 3. Sink nodes: they allow the interconnection between the wireless sensor network and a data network (TCP/ IP). They are also called Zigbee coordinators. 4. Base Stations: they can be a computer or a server, and its main function is to collect the data

## THREATS TO WSN SECURITY

Considering that there is a large number of nodes in WSNs and, in many cases, they are usually deployed in hostile unattended environments without human supervision, they become a principal target for malicious attacks. Wireless sensor networks, as well as computer networks, can suffer a great number of attacks, which, for the sake of simplicity, can be categorized in Denial of Service (DOS) (DorcaJosa and Serra-Ruiz, 2014), eavesdropping (Dos Santos, Hennebert and Lauradoux, 2015), spoofing and replay.

Furthermore, its security poses different challenges than traditional network security, mainly due to the limited hardware resources, and the impossibility of using some security tools whose analytical functions require high hardware consumption. WSN attacks were tested in (DorcaJosa and Serra-Ruiz, 2014), where it was determined that there are several possible attacks to the wireless sensor networks according to the layer of the Zigbee protocol. For example, PHY layer attacks can be performed as jamming and data tampering. In MAC layer, attacks can be collision attacks, channel noise and frame length (Mohammadi, Atani and Jadidoleslamy, 2011). Network layer attacks are homming attacks, erroneous or selective routing, black holes, wormholes and Sybil (Raghu Vamsi and Kant, 2016). Moreover, flooding and desynchronization attacks are related to the application layer. Wireless sensor networks are accessible by anyone mainly due to the low cost of the equipment hardware. In addition, there are several open source tools for hacking WSNs, and consequently it is becoming more dangerous to implement wireless networks with low security devices (Oreku and Pazynyuk, 2016). Initially, the nodes of the WSNs had reduced functionalities and capabilities and, as a consequence, they were immune to malware. Nevertheless, today the sensor devices are endowed with more complex operating systems and resources and in some cases they are similar to smart devices (Akyildiz, Melodia, and Chowdhury, 2007) This new scenario allows malware propagation, and specifically the spreading of proximity malware (Zema et al. 2014, Wang et al. 2013, Wang et al. 2014). For this reason, many techniques and tools have been developed for malware detection and elimination, ranging from antivirus, antimalware, antispyware, and others. In recent years, it has been of great interest to develop mathematical models for the malware propagation in WSNs (see Queiruga et al. 2016, and references therein).

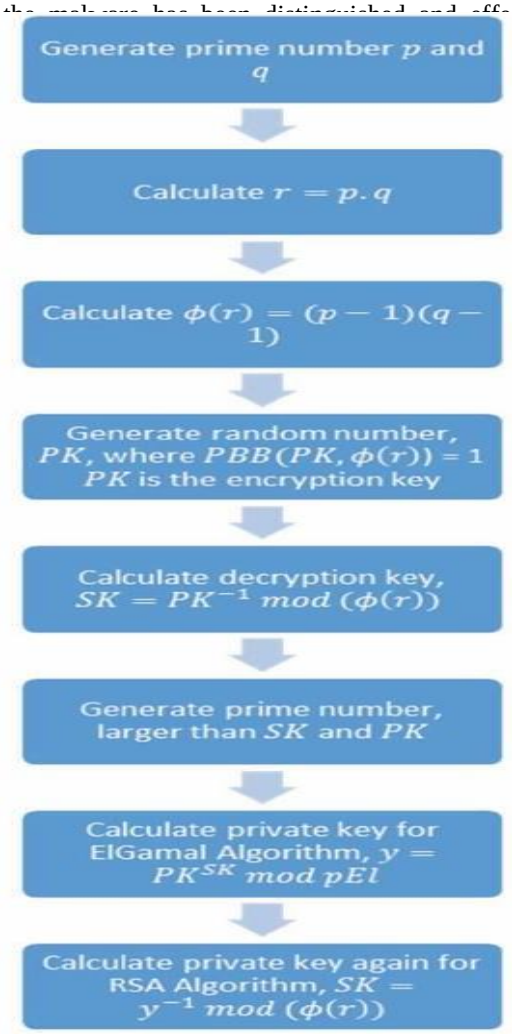
## PROPOSED SYSTEM

RSA is public key cryptographic algorithm by calculatin thr prime varibles. The troubles in the other systems can be improved by using prime number calculations. The proposed model uses elgamal calculation for public key encryption and decryption with Montgomery remainder calculation

The pair of generated key is transfer to the client with encryption and decryption operations. The concept of elgamal adds additional security to the receiver nodes with minimal computations cost. The individual numbers are calculated based on prime numbers for key generations.

. The security of this calculation deceits in the care of computing the discrete logarithm.

In the course of the most recent twenty years, there have been a few models created to mimic malware engendering in various situations: PC organizations, versatile organizations, remote organizations, and so on (see Karyotis et al., 2016; Peng et al., 2014, and references in that). The incredible greater part are compartmental models, that is to say, the number of inhabitants in gadgets is partitioned into various classes as indicated by the malware status and attributes (defenseless, idle, irresistible, recuperated, isolated, secluded, and so on) The point of these models is to concentrate on the dynamic of these compartments into which the populace is separated. Powerless (or sound) gadgets are those gadgets that poor person been tainted by the malware; inactive gadgets are those gadgets that have been reached by the malware however it can neither to play out its malevolent payload nor to engender to another host (the noxious code isn't dynamic); irresistible gadgets are those contaminated gadgets with the end goal that the malware is dynamic; recuperated gadgets are those gadgets where the malware has been distinguished and effectively elimin



Therefore, the dynamic of the model is straightforward (see Figure 3): a powerless gadget becomes tainted (inert or irresistible) when the malware arrives at it; the gadget stays in idle status as long as the malware will be dormant with the goal that when the idle time frame completes the gadget becomes irresistible; if the malware is identified and taken out the host becomes recuperated, any other way it very well may be segregated or isolated. At last, a recuperated gadget becomes helpless again in the event that the recuperation or inoculation processes don't present super durable insusceptibility. Clearly the compartments considered and the dynamic between them rely upon the computerized climate (attributes of the gadgets and the organization) and the principle highlights (engendering designs, payload, and so forth) of the malware. As an outcome, and considering this multitude of contemplations, there are a few classes of compartmental models: SI (Susceptible-Infectious), SIS (Susceptible-Infectious-Susceptible), SLI (Susceptible-Latent-Infectious), SIR (Susceptible-Infectious-Recovered), and so forth.

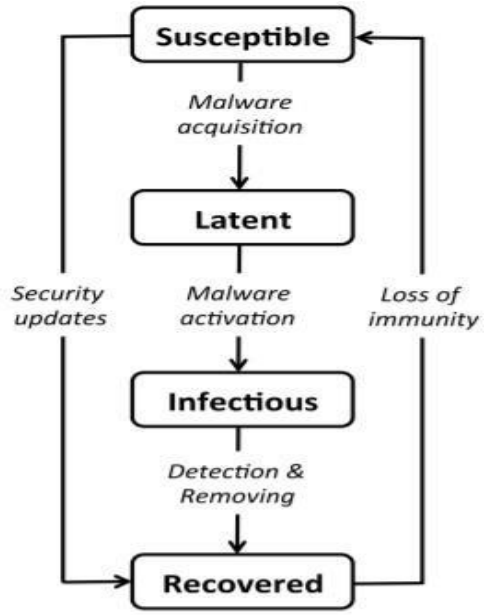



**Fig. 1 Flow diagram**

The flow diagram is representing the dynamic of a SLIRS mathematical model for malware propagation considering security updates

## CONCLUSION

This work proposed a new RSA based architecture in order to improve the network security. The presence of malware in a network affects the network performance by malfunctioning the network activity. The proposed model identify the attackers by performing encryption and decryption operations

## FUTURE WORK

ANN(Artificial neural organization) malware recognition will be utilized.

## REFERENCES

[1] C. C. Zou, W. Gong, and D. Towsley, "Code red worm propagation modeling and analysis," in *Proceedings of the 9th ACM conference on Computer and communications security*. ACM, 2002, pp. 138–147.

[2] Y. Wang, S. Wen, Y. Xiang, and W. Zhou, "Modeling the propagation of worms in networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, 2014. pp. 942–960.

[3] K. Yang, K. Zhang, J. Ren, and X. Shen, "Security and privacy in mobile crowdsourcing networks: challenges and opportunities," *IEEE Communications Magazine*, vol. 53, no. 8, 2015. pp. 75–81.

[4] M. Roberts and J. Heesterbeek, "Mathematical models in epidemiology," *Encyclopedia of Life Support Systems (EOLSS)*, 2003.

[5] R. Pastor-Satorras and A. Vespignani, "Epidemic spreading in scale-free networks," *Physical review letters*, vol. 86, no. 14, 2001. p. 3200.

[6] Y. Moreno, J. B. G´omez, and A. F. Pacheco, "Epidemic incidence in correlated complex networks," *Physical Review E*, vol. 68, no. 3, p.035103, 2003.

[7] C. C. Zou, D. Towsley, and W. Gong, "Modeling and simulation study of the propagation and defense of internet e-mail worms," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 2, 2007, pp. 105–118.

[8] Z. Chen and C. Ji, "Importance-scanning worm using vulnerable-host distribution." in *Proceedings of IEEE GLOBECOM 2005*, 2005, p. 6.

[9] Z. Chen and C. Chen, "A closed-form expression for static worm-scanning strategies," in *Proceedings of IEEE ICC 2008*. IEEE, 2008, pp. 1573–1577.

[10] Q. Wang, Z. Chen, C. Chen, and N. Pissinou, "On the robustness of the botnet topology formed by worm infection," in *Proceedings of IEEE GLOBECOM 2010*. IEEE, 2010, pp. 1–6.

[11] W. Yu, X. Wang, P. Calyam, D. Xuan, and W. Zhao, "Modeling and detection of camouflaging worm," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 3, 2011, pp. 377–390.

[12] M. Nekovee, "Worm epidemics in wireless ad hoc networks," *New Journal of Physics*, vol. 9, no. 6, 2007, p. 189.

[13] ——, "Modeling the spread of worm epidemics in vehicular ad hoc networks," in *Proceedings of IEEE VTC 2006-Spring*, vol. 2. IEEE, 2006, pp. 841–845.

[14] B. K. Mishra, S. K. Srivastava, and B. K. Mishra, "A quarantine model on the spreading behavior of worms in wireless sensor network," *Transaction on IoT and Cloud Computing*, vol. 2, no. 1, 2014, pp. 1–13.

[15] P. De, Y. Liu, and S. K. Das, "An epidemic theoretic framework for vul-nerability analysis of broadcast protocols in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 3, 209, pp. 413–425.

[16] G. Yan and S. Eidenbenz, "Modeling propagation dynamics of Bluetooth worms (extended version)," *IEEE Transactions on Mobile Computing*, vol. 8, no. 3, 2009, pp. 353–368.

[17] B. Liu, W. Zhou, L. Gao, S. Wen, and T. H. Luan, "Mobility increases the risk of malware propagations in wireless networks," in *Proceedings of IEEE TrustCom* 2015.

[18] R. Hekmat, *Ad-hoc networks: fundamental properties and network topologies*. Springer Science & Business Media, 2006.

[19] Z. J. Haas, J. Deng, B. Liang, P. Papadimitratos, and S. Sajama, "Wireless ad hoc networks," *Encyclopedia of Telecommunications*, 2002.

*[20]* C. Wang, J. C. Knight, and M. C. Elder, "On computer viral infection and the effect of immunization," in *Proceedings of Computer Security Applications 2000*. IEEE, 2000, pp. 246–256.

[21] G. Kesidis, I. Hamadeh, Y. Jin, S. Jiwasurat, and M. Vojnovi´c, "A model of the spread of randomly scanning internet worms that saturate access links," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 18, no. 2, 2008, p. 6.

[22] J. Su, K. K. Chan, A. G. Miklas, K. Po, A. Akhavan, S. Saroiu, E. de Lara, and A. Goel, "A preliminary investigation of worm infections in a bluetooth environment," in *Proceedings of the 4th ACM workshop on Recurring malcode*. ACM, 2006, pp. 9–16.