A Functional Relationship Based Attestation Scheme for Detecting Compromised Nodes in Large IoT Networks

Yong-Hyuk Moon, Yong-Sung Jeon and Chan-Hyun Youn

Abstract Despite memory traverse is commonly used for attestation, this approach could not feasibly work for an IoT network that requires scalable and sustainable operations. To overcome this limitation, we propose a functional relationship based attestation scheme, which verifies the integrity of battery-powered devices by analyzing the consistency among neighbors, where a consistent edge between two nodes is given if outputs of the same functions at both nodes are equal to each other. Efficiency of the proposed method is demonstrated in terms of attestation termination and detection speed.

Keywords Attestation · Integrity measurement · Device security · Threat detection · Internet of Things

1 Background

Large-scale connectivity with Internet of Things (IoT) provides a power of cooperation for building a contextual and collective intelligence. Although the large number of end-points participate and interact with each other in this environment, distributed systems are built upon the assumption that each element (computing and communication) is legitimate and trustable. This unreasonable belief makes difficulties in establishing and sustaining reliable networking among

Y.-H. Moon(✉) · Y.-S. Jeon
Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea
e-mail: {yhmoon,ysjeon}@etri.re.kr

C.-H. Youn
Korea Advanced Institute of Science (KAIST), Daejeon, Republic of Korea
e-mail: chyoun@kaist.ac.kr

D.-S. Park et al. (eds.), *Advances in Computer Science and Ubiquitous Computing*, Lecture Notes in Electrical Engineering 373,
DOI: 10.1007/978-981-10-0281-6_101

devices, so that devices could be revealed to serious vulnerabilities as evidenced in [1] and [2]. For example, a compromised node could be an active forwarder to propagate malicious codes or perform replay attacks. For this reason, a role of solution that detects exploited devices is crucial. We argue that the existing attestation schemes [3-8] are not feasible for the IoT since battery-powered devices have different requirements (i.e., scalability and sustainability) compared to conventional desktop-class machines.

We briefly review three representative approaches for integrity verification. It is straightforward to use periodic monitors of integrity [3] [4]; however, this technique needs to increase the frequency of monitoring in order to precede unpredictable changes in targets. Continual observation leads to considerable performance overhead and is very weak to transient attacks. To avoid this problem, load-time measurement methods [5] [6] for kernel protection have been studied. One critical drawback of this approach is that a verification routine (e.g., code) can be compromised if an untrusted input is allowed even though the routine is of high integrity. Besides, a process that gets compromised at run time cannot be detected by this technique. In another line of work, temper-resistant hardware based root of trust has been deployed in a device to improve the verification reliability [7] [8]. Due to physical isolation property, these types of hardware generally accommodate a single and static configuration, which takes the role of judgement criterion, so that updates of relevant policy impose additional costs and expose a device to new vulnerabilities.

Particularly, most of existing studies require memory traverse for measuring bytes, resulting in that redundant memory access is unavoidable. Although memory snapshot is somewhat essential in order to verify the security state of target node, the order of memory access could be predictable with a high probability. One more disadvantage is that a memory traverse strongly depends on the memory architecture employed in an IoT device. Despite, the key question of how to effectively attest devices in large-scale networks without memory traverse remains unanswered.

In this paper, we confine our focus to the following two requirements in order to propose a feasible attestation scheme, which does not take a snapshot of memory.

Scalability. Ensuring the integrity of a particular node should be possible by pinpointing malicious or suspicious at least nodes based on strong evidence even in the large-scale device network.

Sustainability. Attestation should not obstruct the management functionalities for an IoT network; and its burden should be restricted to the acceptable notion in terms of service quality.

To satisfy the above requirements, we design a new attestation scheme, which assesses outputs of relevant functions of nodes on a pre-defined path, instead of performing attestation for a single node (e.g., memory fidelity). The proposed scheme also verifies the integrity of suspicious nodes by analyzing the functional relationships (i.e., consistency). The remainder of this paper is organized as follows. Section 2 proposes a functional relationship based attestation scheme.

In Section 3, we discuss related strength and additional issues of the proposed attestation scheme. Simulation results are demonstrated and discussed in terms of termination convergence and detection speed in Section 4. Finally, we conclude the paper and outline issues for future work in Section 5.

2 Proposed Attestation Scheme

This section presents a novel methodology for node attestation based on a concept of integrity verification function (IVF) in a large-scale IoT network. The proposed attestation scheme consists of six phases, such as neighbor discovery, verifier election, path setup, function assessment, path integration and consistency analysis.

2.1 Neighbor Discovery and Verifier Election

IoT devices are connected with many redundant interconnections among network nodes, resulting in that they could form a mesh network over a period of time to relay data through neighbors. Unlike a dedicated verifier based attestation, security status of each node is measured by a random verifier that is elected at every round. This approach maintains sustainability well and is also suitable for scalability.

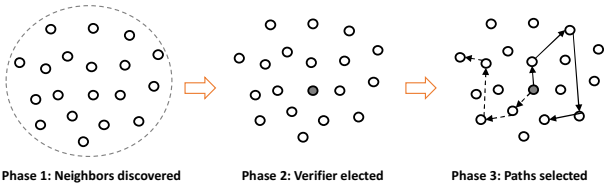

Fig. 1 A schematic view of the proposed attestation scheme based on per-function evaluation using randomly selected subset of neighbors.

Neighbor Discovery. If a node is newly deployed in an IoT network, it initially attempts to discover neighbors by sending a hello message to other nodes in the near distance [9]. After discovery, the node resets or synchronizes its own timer and then establishes a pair-wise key with every neighbors.

Verifier Election. All nodes have an asynchronous timer of detecting a new event for election within a neighbor group. When an interval of time runs out, one node should be elected for taking a role of verifier. To this end, we adopt a similar approach of a cluster-head selection algorithm, such as (LEACH) [10]. Node i whose time is up sends a freeze message to neighbors. Then, neighbors notify that the round for attestation is updated. Node i computes the following threshold value, T_i, and produces a random number, $\ddot{e}$, between 0 and 1. By comparing each other, node i reaches a conclusion on whether to be a verifier. If $\ddot{e} < T_i$, node i becomes a new verifier for attestation in the current round. Next, the verifier broadcasts an advertisement message to the rest of nodes involved in the same neighborhood; thus, neighbors' timers can be unfrozen.

$$T_i = \frac{p}{1 - p\{r \bmod (p-1)\}} \text{ if } i \in D, \tag{1}$$

where i denotes an identification of node, i.e., node i and D stands for a domain of nodes that have not been a verifier in the last $1/p$ rounds. p is a percentage of verifiers (e.g., $p = 0.02$) and r is a current round. If node i does not belong to D, $T_i = 0$.

Once the aforementioned election algorithm is operated at node i, a random interval of time is set. On the other hand, node i broadcasts an election request to the rest of nodes and then each neighbor executes the same task for election.

2.2 Path Establishment and Function Assessment

We suppose that all devices support network-level encryption, so that additional authentication protocol is not required and a message is transmitted through a secure session established between nodes. Since they contain the same group of IVFs, each one is capable of offering an output for a requested function.

Path Setup. In order to construct a subset of neighbors, two attestation paths are established according to the principle of a random walk algorithm [11]. Moreover, the length of both paths are bounded and equal to each other. For example, path 1, $P_1 = \{8, 3, 2, 6, 5, 4, 1, 7\}$, and path 2, $P_2 = \{6, 4, 1, 3, 2, 8, 7, 5\}$, are constructed with eight nodes at round $i + 1$as shown in Fig. 2.

Function Assessment. If attestation paths are given, a function of each node is also determined. In case of P_1, IVFs f_1 to f_8 are used to result in an output in numerical order as a response to a random input generated by a verifier. Likewise, functions f_1 to f_8 are selected for P_2, where $|P_1| = |P_2| = L$. A verifier node v sends an input data ε_v to the first node on a given path, P_1. The first node, 8, produces a result using f_1 with ε_v and then forwards its result to the second node, 3, on P_1. Repeatedly, this process is executed to the last node, 7, on the path. Also, a verifier performs the same task again in order to aggregate immediate results to the final node on P_2. Here, we assume that by comparing individual results sequentially, it is found that there are two functionally inconsistent relationships between node 6 and node 3, and node 1 and node 7 as shown in Fig. 2. The measured functional relationships are then translated to a topology as depicted below.

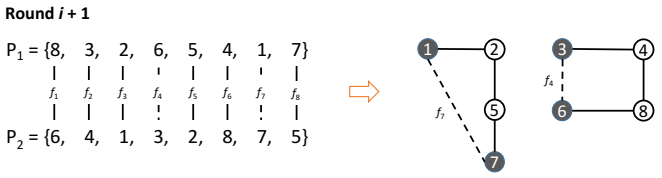

Fig. 2 An example of path setup and intermediate node topology (phase 4). A solid line is an edge with functional consistency between two nodes and a dotted line is for inconsistent relationship.

Remark 1. Although the proposed attestation scheme does not measure and verify all neighbors within a small number of rounds if the size of neighbor group θ is large, all neighbors will be covered by the probabilistic property of random walk over a period of time.

2.3 Path Integration and Consistency Analysis

After phases 3 and 4, a verifier sends topologies as an intermediate result to a gateway. Then, this output is integrated with the last obtained node topology in a gateway for the purpose of topology update. Fig. 3 gives a good example of path integration. Then, a gateway performs a consistency analysis for detecting suspicious nodes.

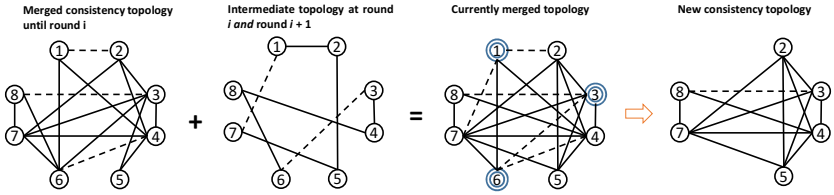

Fig. 3 An illustration of merging a topology with the existing one (phase 5). A circle with two outer lines denotes a node that has more than two inconsistency edges.

Path Integration. As depicted in Fig. 3, a verifier stores a consistency topology integrated until round i and an intermediate topology is built during attestation at round $i + 1$. These two consistency topologies are merged with each other; so, a gateway has a current one as depicted in the third figure of Fig. 3. After this phase, a gateway can point out which node is suspicious according to a predefined inconsistency degree d. For ease of demonstration, we set d to 2 for all nodes; thus, nodes 1, 3, and 6 come under suspicion.

Consistency Analysis. A gateway starts attestation when it is recognized that the following Condition 1 is satisfied in the currently merged graph.

Condition 1. If node i has more than two tolerance degree (i.e., $d_i \geq 2$, $1 \geq i \geq N$), it is said that node i is suspicious. Otherwise, node i turns out to be legitimate. A gateway confirms whether a particular node provides a reliable service or not.

More specifically, the gateway attempts to find functional consistency groups (also known as cliques in graph theory) at every round by using the Bron-Kerbosch (hereinafter referred to as BK) algorithm [12], which compute all cliques in linear time relative to the number of cliques. If the size of IoT network, N, is sufficiently large enough, assuming that the number of legitimate nodes is larger than that of compromised nodes is reasonable. Hence, we establish the second condition in order to verify suspicious nodes.

Condition 2. If node i belongs to a complete sub-group that consists of more than half neighbors, it is said that node i is consistently trustable. Otherwise, node i is treated as a malicious or compromised one, so that node i will be isolated.

In the currently merged topology, node 1 does not form any complete sub-group. Also, there is no clique that includes node 6. In case of node 3, there are five different complete sub-groups, such as {2, 3, 5}, {2, 3, 7}, {3, 5, 7}, {2, 3, 4, 7}, {2, 3, 5, 7}, and {2, 3, 4, 5, 7}. We suppose that a size of complete sub-group, s, is $\theta / 2$, i.e., $s \geq 4$ according to the aforementioned Condition 2. Therefore, nodes 1 and 6 are isolated and node 3 is maintained in the IoT network as shown in the rightmost figure of Fig. 3.

Remark 2. By setting d_i, a gateway pinpoints suspicious nodes, so that the effort of finding complete sub-groups is reduced. In addition to that, L is a fixed constant, so that the BK algorithm can be performed within acceptable time complexity.

3 Analysis and Discussion

So far, we have proposed a functional relationship based attestation scheme for an IoT network. We review the proposed attestation scheme in the three aspects, such as security strength, overhead reduction, and remaining issues.

Strength. In the purely distributed attestation, conventional secret dissemination and its recovery could be vulnerable points against an attacker. However, the proposed attestation scheme does not utilize and share the secret since the same set of functions are loaded on all devices during a manufacturing process. Further, each node has its own interval of time for election (i.e., asynchronous) and its timer is reset with a random interval once the election algorithm is executed. Thus, an attacker is hard to predict the next election time and which node will be elected.

Performance. Our election algorithm allows multiple verifiers; thus, frequent election would occur. However, we can set the number of verifiers that are elected at the similar time zone by adjusting p value in Eq. (1). This means that the proposed scheme decides an attestation event in a probabilistic manner and nodes, which have been selected as a verifier in the last $1 / p$ rounds is excluded for a current election.

Improvement. While the proposed election algorithm works effectively by simply comparing a randomly-generated number with a calculated threshold, if we assume that an elected verifier is not be trustable, an additional scoring factor should be added into Eq. (1) for assessing statistical or historical reputation of election candidate nodes.

4 Performance Evaluation

In this section, we aim to evaluate the effectiveness (i.e., detection convergence) of proposed attestation scheme in terms of two indices: attestation termination, which is measured as attestation rounds spent until the number of remaining neighbors is less than the minimum value of L, L_{min}; and detection speed, which describes the ability of how quickly all compromised nodes are confirmed in order to prevent malfunction or error propagation. To construct an IoT network for simulation study, we use a complete graph that has N vertices and each vertex has N - 1 connections, so that all vertices could be connected to the others when a path is established. In other words, the possible number of edges, M, is equal to $N(N - 1)/2$. To this end, we use a random graph, where the probability δ controls the density of network topology [11].

If the probability of detecting compromised nodes λ is randomly given and the number of compromised nodes μ are known at every round, the expected number of compromised nodes can be calculated as $\lambda \cdot \mu$. In order to avoid unrealistic value of $\lambda \cdot \mu$, we then set its upper and lower bounds as follows:

$$-\log \alpha \leq \lambda \cdot \mu \leq -\log \beta, \tag{2}$$

where α and β are random numbers and vary over rounds ($0 < \alpha, \beta \leq 1$). This implies that we can identify how many infections are occurred among nodes at a particular attestation round.

If $\lambda \cdot \mu$ value is bounded as described in Eq. (2), we randomly decide which function outputs a wrong result. Otherwise, we assume that no function is infected by malicious code. With the probabilistic infection event based on exponential random numbers, we can emulate malicious code injection to a node. Moreover, the BK algorithm is adopted in our simulation to find a complete sub-group of neighbors when the integrated functional relationship is given to a gateway. This simulation is terminated when all nodes are infected by malicious code or when the number of neighbors remaining is less than L_{min}. Furthermore, we use average values that are obtained by conducting the simulation 100 times for five different neighbor groups. Parameters mainly used in the simulation are summarized in Table 1.

Table 1 Parameter configuration used in the simulation

Parameter	Description	Value
N	The number of nodes	100
θ	The size of neighbor group	20
p	Percentage of verifier	0.02
δ	Density of network topology	0.01
L	A length of attestation path	5, 8
d	A tolerance degree	2, 4
S	A size of complete sub-group	Larger than θ / 2

Attestation Termination. As shown in Fig. 4(a), we measure attestation rounds spent until the number of remaining neighbors is less than L_{min} with four different cases of d and L values. If setting a value of d is low, that is 2, it requires performing analysis more frequently, while a gateway find suspicious ones more rapidly than a case of d = 4. Otherwise, IoT network needs to tolerate many suspicious nodes, if a value of d is set too high. Moreover, when L is increased, overhead due to function assessment via communication is possibly expected. On the other hand, a detection rate will be slowly converged to 1 because a gateway can obtain more intermediate topologies for consistency analysis. Due to this reason, two cases of L = 8 have less remaining neighbors at the same round, compared to those of L = 5, regardless of a value of d. Convergence that is measured by attestation termination time occurs at 36, 23, 46, and 48 rounds, respectively for four cases from top to bottom.

Detection Speed. Next, we only change to fix d = 2 and adjust L to floor of $\theta \times 0.4$ under the same simulation configuration in order to investigate different aspect of detection convergence. Fig. 4(b) shows average rounds to find a first suspicious node (5 rounds) as well as to confirm the all compromised nodes (12 rounds) when malicious codes are injected by Eq. (2). This result implies that by adjusting a value of L the proposed scheme can detect all compromised nodes within 7 rounds averagely after a first suspicious node is found. In a real system, the absolute time depends on two factors: an interval of election event and elapsed time for a round.

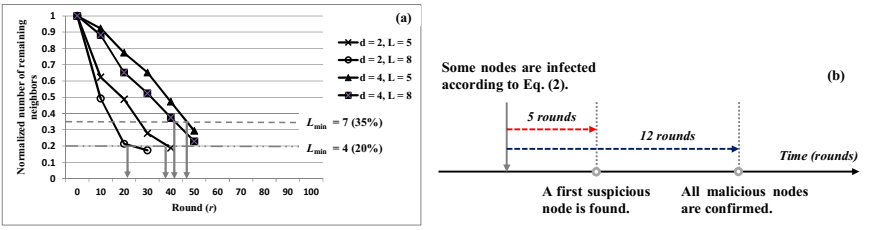

Fig. 4 Detection convergence with adjusting values of d and L.

5 Conclusion and Future Work

We have proposed a functional relationship based attestation scheme, which feeds the same input into multiple neighbor nodes and then compare their outputs in order to create a topology including consistent or inconsistent edges among nodes. The proposed scheme pinpoints suspicious nodes and then verifies their integrity by analyzing consistency without traversing memory. With this approach, attestation is applied to a group of neighbors, so that scalability is guaranteed. Further, our election algorithm only selects one verifier at an interval time and verification is performed by not an end-point node but a gateway. It implies that the sustainability is ensured in the proposed scheme. Our immediate work is to

deploy the proposed scheme in real IoT networks and heterogeneity of IVFs will be considered for more constrained environment. We also will explore probabilistic aspects of the propose scheme in future work.

Acknowledgments This research was supported by Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) [R-20150518-001267, Development of Operating System Security Core Technology for the Smart Lightweight IoT Devices].

References

1. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: A survey. Computer Networks **54**, 2787–2805 (2010). doi:10.1016/j.comnet.2010.05.010
2. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems **29**, 1645–1660 (2013)
3. Loscocco, P.A., Wilson, P.W., Pendergrass, J.A., McDonell, C.D.: Linux kernel integrity measurement using contextual inspection. In: Proceedings of the 2nd ACM Workshop on Scalable Trusted Computing (STC 2007), pp. 21–29. ACM (2007)
4. Petroni Jr., N.L., Hicks, M.: Automated detection of persistent kernel control-flow attacks. In: CCS 2007: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 103–115. ACM (2007)
5. Jaeger, T., Sailer, R., Shankar, U.: PRIMA: policy-reduced integrity measurement architecture. In: Proceeding of the Eleventh ACM Symposium on Access Control Models and Technologies, pp. 19–28 (2006)
6. R. Macdonald, S. Smith, J. Marchesini, and O. Wild. Bear: An open-source virtual secure coprocessor based on TCPA. Technical Report TR2003-471, Department of Computer Science, Dartmouth College, 2003
7. Trusted Computing Group (TCG). TPM Main Specifications. Version 1.2 rev 116, March 1, 2011. http://www.trustedcomputinggroup.org/resources/tpm_main_specification
8. Sailer, R., Zhang, X., Jaeger, T., van Doorn, L.: Design and implementation of a tcg-based integrity measurement architecture. In: Proceedings of the 13th USENIX Security Symposium, August 9–13, 2004, San Diego, CA, USA, pp. 223–238 (2004)
9. Ganesh, A.J., Kermarrec, A.M., Massoulié, L.: Peer-to-peer membership management for gossip-based protocols. IEEE Trans. Comput. **52**(2), 139–149 (2003)
10. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Hawaii International Conference on System Sciences, vol. 8 (2000)
11. Erdős, P., Rényi, A.: On Random Graphs. Publicationes Mathematicae **6**, 290–297 (1959)
12. Bron, C., Kerbosch, J.: Algorithm 457: finding all cliques of an undirected graph. Commun. ACM **16**(9), 575–577 (1973). doi:10.1145/362342.362367