CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Preserving context privacy in distributed hash table wireless sensor networks
Authors	Palmieri, Paolo
Publication date	2015-12
Original Citation	Palmieri, P. (2016) 'Preserving Context Privacy in Distributed Hash Table Wireless Sensor Networks', in Qing, S., Okamoto, E., Kim, K. & Liu, D. (eds.) Information and Communications Security: 17th International Conference, ICICS 2015, Beijing, China, December 9-11, 2015, Revised Selected Papers. Cham: Springer International Publishing, pp. 436-444. doi: 10.1007/978-3-319-29814-6_37
Type of publication	Conference item
Link to publisher's version	https://link.springer.com/chapter/10.1007/978-3-319-29814-6_37 - 10.1007/978-3-319-29814-6_37
Rights	© Springer International Publishing Switzerland 2016. The final publication is available at Springer via http://doi.org/10.1007/978-3-319-29814-6_37
Download date	2026-11-03 03:04:46
Item downloaded from	https://hdl.handle.net/10468/4761



UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Preserving Context Privacy in Distributed Hash Table Wireless Sensor Networks

Paolo Palmieri

Department of Computing and Informatics, Bournemouth University,
Fern Barrow, Poole, BH12 5BB, United Kingdom
ppalmieri@bournemouth.ac.uk

Abstract. Wireless Sensor Networks (WSN) are often deployed in hostile or difficult scenarios, such as military battlefields and disaster recovery, where it is crucial for the network to be highly fault tolerant, scalable and decentralized. For this reason, peer-to-peer primitives such as Distributed Hash Table (DHT), which can greatly enhance the scalability and resilience of a network, are increasingly being introduced in the design of WSN's. Securing the communication within the WSN is also imperative in hostile settings. In particular, context information, such as the network topology and the location and identity of base stations (which collect data gathered by the sensors and are a central point of failure) can be protected using traffic encryption and anonymous routing. In this paper, we propose a protocol achieving a modified version of onion routing over wireless sensor networks based on the DHT paradigm. The protocol prevents adversaries from learning the network topology using traffic analysis, and therefore preserves the context privacy of the network. Furthermore, the proposed scheme is designed to minimize the computational burden and power usage of the nodes, through a novel partitioning scheme and route selection algorithm.

Keywords: Wireless sensor networks, context privacy, anonymity, onion routing, distributed hash table.

1 Introduction

A Wireless Sensor Network (WSN) generally consist of a number of small, low-power computing sensors, deployed in an environment where they observe physical phenomena. The sensors, typically battery-powered and highly constrained in their computational capabilities, are nonetheless able to perform sensing, wireless communication and computation tasks. They collect and disseminate data about the supervised phenomena cooperatively, and collaborate in order to perform a common task. WSN's applications include health monitoring, smart agriculture, weather sensing, intrusion detection and industrial control [9,6]. However, in spite of the extensive research, WSN's still face many challenges, including security, privacy and network robustness and scalability. Wireless sensor networks share some of the above challenges with peer-to-peer networks: in particular, both network designs aim at achieving a high level of scalability in a decentralized

manner, and must be able to cope with nodes failing, entering or abandoning the network at any time. For this reason, some of the network constructions used in the peer-to-peer setting have been adopted in the design of WSN [10]. In particular, the Distributed Hash Table (DHT) network topology has found application in a number of WSN designs [5].

Privacy is also an increasing concern for WSN's [8]. The wireless nature of the communication link makes the network inherently vulnerable to eavesdropping. Moreover, nodes of the wireless sensor network are often deployed outdoor, in unsurveilled areas where they can be subject to tampering. An attacker might be able to gain control of one or more nodes. For this reason, privacy must be preserved even against internal adversaries. In particular, both the privacy of the data gathered and the privacy of the context should be protected. In order to preserve data privacy, nodes should not be able to build a more detailed picture of the data than it is required for their functioning (aggregation).With regard to context privacy, instead, the primary aim is to hide the location of sensors and base stations, the network topology and in certain cases the time data was collected. WSN's are in general highly vulnerable to attacks targeted at base stations, which coordinate network operation and gather data: the failure of a base station can in fact disrupt the whole network. The geographic location of base stations and the network topology should therefore be concealed [2]. Common strategies for hiding location and prevent traffic analysis are flooding, transmissions from fake sources [14], and random walks [7]. Random walks for the transmission of the information, in particular, have been adopted in a number of designs. Zhang proposed self-adjusting directed random walks in [15], while GROW (Greedy Random Walk) [13] introduced a two-way random walk, from both source and destination, to reduce the chance of an eavesdropper being able to collect location information. Finally, layers of encryption can be used to protect the information at each hop in the walk [3].

In this paper we propose a modified onion routing protocol for wireless sensor networks that are based on the distributed hash table topology. Onion routing, the anonymity protocol at the base of Tor and other privacy preserving technologies, can prevent tampering and preserve the secrecy of both the communication content and the location of the sender and receiver. Applied to wireless sensor networks, onion routing is an effective strategy to protect context privacy. However, the limited capabilities of sensing nodes, and the particular network infrastructure of a WSN make implementing onion routing a challenging task. For this reason, we aim at minimizing any computational overhead of onion routing. In particular, the processes of route selection and key distribution are fairly complex tasks in a WSN. Route selection is made difficult by the absence of a complete view of the network by each node. A node, in fact, might be able to communicate directly with only a fraction of all the nodes in the WSN, due, for instance, to distance or wireless signal visibility. We solve this issue by proposing a route selection algorithm based on Bloom filters, which takes into account the limited network view without compromising the secrecy of the route. Without a directory of all nodes, implementing the standard key distribution mechanism of

onion routing is also impossible over a WSN. Therefore, we adopt a key distribution strategy designed for the DHT paradigm, making the network resilient to nodes failing, entering and abandoning it without disrupting the onion routing mechanism. The proposed protocol enables, for the first time, the implementation of a full-fledged onion routing mechanism over a wireless sensor network, overcoming the traditional limitations of the typical WSN infrastructure. This, in turn, will allow for better protection of context privacy and the location of nodes in the network, making WSN's more resilient to attacks targeting base stations or specific nodes.

2 Distributed Hash Table

Peer-to-peer networks are large distributed systems designed around three main principles: *decentralization*, *fault tolerance* and *scalability*. As wireless sensor networks are often required to satisfy the same properties, they sometimes integrate solutions first proposed in the peer-to-peer domain [10]. This is the case, in particular, for the family of WSN's designed around one of the most successful peer-to-peer designs, Distributed Hash Table (DHT) [5],used daily by millions of users through the BitTorrent protocol.

A DHT is defined over a *keyspace*, defined as the output range of a hash function. The keyspace is divided among the nodes in the network by using a partitioning scheme. For each DHT, a function defines the *distance* between any two values in the keyspace. The partitioning is achieved by assigning to each node in the network a value in the keyspace called *identifier*: the peer is responsible for the subset of values in the keyspace that have less than a predefined distance from his identifier value. Each node in the network maintains a connection to a number of other nodes, called *neighbors*. Neighbors are generally selected according to a certain structure, known as the network topology, which, together with the hash and distance functions, defines the specific DHT. For the purpose of this paper, we assume any WSN built over a DHT to specify: the keyspace K; the size of the keyspace s; the hash functions h used to map information generated by the sensors to values in the keyspace; the function f_{dist} determining the distance between two values in the keyspace; and the maximum distance d for which a node with identifier A is responsible for keys in the keyspace. We also assume the keyspace to be two-dimensional: given a distance d, a function f_{dist}, and a value v in the keyspace, there are exactly two values v^+ and v^- for which $f_{dist}(v^+) = f_{dist}(v^-) = d$.

3 Onion Routing

Onion routing is a mechanism allowing anonymous and privacy-preserving communication in a network. First proposed in 1997 [12], onion routing has recently seen widespread use thanks to the Tor implementation [4], which counts millions of active users. Onion routing can prevent surveillance and traffic analysis. The identity, location and network activity of each node is protected by concealing

to external observers both content and routing information of the node's traffic. This is achieved by relaying the traffic, including information about its destination, through a virtual *circuit* composed of three intermediary nodes (called *relays*). Each relay in the circuit only learns the preceding and following node, so that the sender remains anonymous to all relays except the first, and the destination remains secret to all relays except the last. In order to protect the actual content of the communication, messages are repeatedly encrypted with a public key encryption scheme in a layered manner, using in inverse order the public keys of all the relays in the circuit. Traffic going back to the source from the destination is similarly encrypted and routed from the last relay to the first one. Implementing onion routing is an effective strategy to protect communication in wireless sensor networks, especially when deployed in hostile environments, subject to both external and internal attacks. In particular, onion routing can prevent traffic analysis and packet eavesdropping, and at the same time prevent malicious nodes in the network from learning sensitive information about the network topology, such as the identity of base stations and data aggregation nodes. This information is generally being referred to as *context data*.

Each node participating in an onion routing protocol has a public and private key pair called *identity key*. This key serves the main purpose of authenticating the node and prove its identity to other nodes. The identity key pair is generated during the node startup, and is maintained for the entire life of the node. However, in order to minimize the impact of a compromised identity key, each node also generates at regular time intervals a second key pair: the *onion key*. The public key of the current onion key pair is signed using the identity key and then distributed to other nodes. The onion key is used during circuit creation: when establishing a circuit, each node being used as relay is challenged to prove knowledge of its (private) onion key. The actual communication over the circuit is encrypted using a symmetric *session key*, agreed on by each couple of consecutive nodes in the circuit. The key is discarded once the communication over the circuit stops, thus achieving forward secrecy. Onion circuits, in fact, are designed to be used only for a limited period of time. Similarly, onion keys are discarded and replaced after they reach the end of their intended lifetime.

4 Onion Routing over DHT Wireless Sensor Networks

Sensor nodes in a WSN are typically constrained devices: at the hardware level, due to the limited computational capabilities and often battery-powered nature, and at the network level, due to the restricted visibility of the network each node possesses. In order to satisfy these constraints, we design the proposed onion routing protocol so that each node in the wireless sensor network is able to operate both as a source of information and as a relay. However, not all nodes will cover both roles at the same time: for energy efficiency, we select a subset of nodes to act as relays for a given time, and rotate to the next subset of nodes after a predetermined amount of time. Therefore, a node will be in one of the following states at any time: *relay state*, during which the sensor node performs

both sensing tasks and relays traffic of other nodes; *sensor state*, during which the node continues to sense information but does not accept traffic from other nodes for relay purposes. In order to be able to act as a relay, each node has an *identity key* pair, generated using a public key encryption scheme. The identity keys can be either loaded onto the device before deployment (to ease the computational burden) or created during bootstrap of the node. Additionally, when the node enters relay state, it also generates a temporary *onion key* pair, which is valid only for the current relay state. Nodes distribute the public keys of both identity and onion pairs following the strategy described in the following. We say that a node *owns* a key if it generated the key, while we say that a node is *responsible* for a key if it stores the key and distributes it to other peers requesting it. Keys are transmitted over the network by using the DHT underlying the WSN.

Identity Key Distribution The subset of nodes in the network that are responsible for the identity key of a specific node are determined by hashing the identifier of the node using the hash function h defined by the DHT. A node X is responsible for the identity key $i_N \in K$ of the node N if:

$$f_{dist}\left(X - h\left(N\right)\right) < d \ . \tag{1}$$

This defines a subset I_N of size $2d$ of the keyspace K. The node owning the key distributes it to the nodes responsible for storing it. Should the value d increase during network operation, the nodes already responsible for the key distribute it to the new nodes in I_N. Should d instead decrease, the nodes that are no longer in I_N discard the key. It is important to note here that nodes in the WSN reply to request for keys both when in relay and sensor state.

Relay State Partitioning Since we want only a subset of the nodes in the WSN to act as onion relays at any given time, we partition the WSN by using the identifier value of the nodes: we consider the first n bits of the identifier value in order to have 2^n different subsets. For instance, all peers with the first n bits of the identifier having value 0 are in the first subset.

Onion Key Distribution The temporary onion keys are generated by the nodes when entering relay state, and are subsequently distributed to the nodes responsible for storing them similarly to the identity keys. However, we design the protocol so that the nodes that are responsible for storing an identity key will not also be responsible for the onion key of the same node. Moreover, a node responsible for the onion key of a node for the current relay state duration, will not be responsible for a key of the same node for a number of following relay states. We achieve this through the distribution mechanism described in the following, which follows that proposed in [11]. For each node N and its identifier $i_N \in K$, we divide the network into $u = \frac{s}{2d}$ partitions of size $2d$, such that one such partition is I_N, as defined by (1). Nodes in each partition except I_N cyclically store the onion key for N that is valid for the current relay state. The partitions are defined by an arbitrary function

$$f_{on}: \ K \rightarrow \{\text{all possible partitions of } K \text{ of size } 2d\}\,, \tag{2}$$

that takes as input an identifier $i \in K$ and outputs $\{O_1, \ldots, O_{u-1}\}$ (the set of partitions), such that $O_1, \ldots, O_{u-1}$ are disjoint subsets of K of equal size and

$$O_1 \cup \ldots \cup O_{u-1} = K \setminus \{I_N\} \quad . \tag{3}$$

For any $u-1$ subsequent relay states, each partition is selected once for storing the onion key of the node, starting from O_1 and moving to the following partition until O_{u-1} is reached (the cycle starts anew after that).

Private Route Selection Each node in a WSN has a limited visibility of the network, and may be able to communicate with only a subset of all the nodes. This has an impact on two basic functionalities of the proposed network: onion circuit creation, and reporting to the base station. In the former case, two successive relays may not be able to communicate with one another. We solve this by introducing the *relay filter*, an additional set of information generated by relays in relay state, and distributed similarly to the onion public key. The relay filter for a relay node r is the subset of the other currently available relays that are visible from r, encoded using a Bloom filter data structure [1]. Bloom filters allow us to keep the information secret to external observers as well as the nodes themselves. The filter is signed by the relay using its onion private key.

In order to preserve context privacy in the WSN, and therefore the identity and network location of the base stations, these particular nodes act similarly to other nodes for the purpose of onion routing. However, base stations own an additional set of keys, the *station key* pair. These keys are generated before the WSN deployment, and are embedded in each node reporting to the relative base station. Station keys are used to sign a filter similar to the relay filter, the *station filter*. The station filter encodes a randomly selected subset of all the relays that are visible from the relays the base station can connect to. Due to the property of Bloom filters, it is possible for the base station to generate the station filter based on the relay filters. The distribution of station filters and its impact on privacy are discussed in Section 4.1.

The mechanism for building an onion circuit between a sensor node and a base station is presented in Protocol 1. The proposed circuit building approach is alternative to that presented in [13], and to Tor hidden services [4].

4.1 Privacy Analysis

According to the distributed trust principle, no single relay in the circuit should learn the identity of both nodes at the two ends of the circuit. We achieve this by letting the nodes select independently the first relay, and by disclosing to each relay only the previous and following step in the circuit. This is possible thanks to the use of the intersection filters explained in step 2 and 3 of the circuit building protocol. Distributed trust provides protection for the context information of the wireless sensor network. We identify two private context information that we aim to preserve: the network location (that is, the position within the DHT) and the identity of the nodes acting as base stations. Use of

Protocol 1: Building an Onion Circuit.

1. The sensor node S identifies the current relays through the partition function, and acquires relay information (current keys and filters) through the DHT, by querying the appropriate set of responsible nodes O_i. Then S selects among its neighbors a potential first relay for the sensor's end of the circuit.
2. S calculates the intersection filter $b_i = b_r \cap b_s$ between the filter of the selected first relay b_r and the station filter b_s. Then, S sends b_i to the relay and instructs it to build a circuit with one of its own neighbors whose identifier satisfies b_i.
3. Similarly to the sensor, and concurrently, the base station node B selects among its neighbors a potential first relay for the station's end of the circuit. Then, it instructs the relay to build a circuit with a neighboring second relay that satisfies a randomly selected subset of the first relay's filter.
4. Once second relays have been selected at both ends of the circuit, the sensor and the base station instruct the relative second relays to look for potential matches. The two relays start querying all neighboring relays for potential circuits, and open a new circuit each time they find open ends.
5. Among all the resulting circuits, S selects the right circuit by querying the node at the other end, and asking it to prove knowledge of the station private key. In case no suitable circuit is found, the node starts again from step 1.

the proposed onion routing effectively conceals the context information to the sensor nodes communicating with the base station, as no information other than the station filter is required for communication. At the same time, relays used in the circuit cannot distinguish a sensor from a base station, as the same steps are taken by both S and B. This is true for as long as the base station does not select as first relay a sensor node that has access to B's station filter: considering that B has access to the list of sensors reporting to itself, we can safely assume this to be the case. For the same reason, station filters are distributed encrypted using the station public key. New station filters are generated at every change of state for the nodes. In order to preserve secrecy of the identity and position of the base station, the station filter distribution can follow two strategies: random walks (where the filter is passed on from node to node randomly, thus making it harder to identify the originating node) or distribution through an onion circuit (where the relay at the end of the circuit will propagate the filter through flooding). Finally, the identity key of a node is verified through a challenge by the node responsible for its distribution, while the onion key is verified by the node building the circuit. Since the node obtains the identity and onion keys from two different set of nodes (Section 4), an adversary would need to corrupt a majority of nodes in both sets in order to perform an attack. This would also have to be repeated at each change of state.

5 Conclusion

In this paper we proposed an onion routing protocol for wireless sensor networks based on the distributed hash table paradigm. The protocol minimizes

the introduced overhead by partitioning the network and rotating relay responsibilities over time, which is particularly effective in battery-powered devices. The proposed construction allows for limited network visibility by the nodes, and is entirely decentralized. Context privacy is preserved by the distributed trust of the onion mechanism, and both sensor nodes and base stations remain anonymous within the network. The network topology is kept secret as a result, opening the way to implementation in WSN deployed in hostile settings.

References

1. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Commun. ACM 13(7), 422–426 (1970)
2. Deng, J., Han, R., Mishra, S.: Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks. In: DSN 2004. pp. 637–646. IEEE (2004)
3. Deng, J., Han, R., Mishra, S.: Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks. Pervasive and Mobile Computing 2(2), 159–186 (2006)
4. Dingledine, R., Mathewson, N., Syverson, P.F.: Tor: The second-generation onion router. In: USENIX 2004. pp. 303–320 (2004)
5. Fersi, G., Louati, W., Jemaa, M.B.: Distributed hash table-based routing and data management in wireless sensor networks: a survey. Wireless Networks 19(2), 219–236 (2013)
6. Gaitan, S., Calderoni, L., Palmieri, P., Veldhuis, M.C.T., Maio, D., Riemsdijk, M.B.V.: From sensing to action: Quick and reliable access to information in cities vulnerable to heavy rain. IEEE Sensors Journal 14(12), 4175–4184 (2014)
7. Kamat, P., Zhang, Y., Trappe, W., Ozturk, C.: Enhancing source-location privacy in sensor network routing. In: ICDCS 2005. pp. 599–608. IEEE (2005)
8. Li, N., Zhang, N., Das, S.K., Thuraisingham, B.M.: Privacy preservation in wireless sensor networks: A state-of-the-art survey. Ad Hoc Networks 7(8), 1501–1514 (2009)
9. Li, Y., Thai, M.T., Wu, W. (eds.): Wireless Sensor Networks and Applications. Signals and Communication Technology, Springer (2008)
10. McGoldrick, C., Clear, M., Carbajo, R.S., Fritsche, K., Huggard, M.: TinyTorrents: Integrating peer-to-peer and wireless sensor networks. In: WONS 2009. pp. 109–116. IEEE (2009)
11. Palmieri, P., Pouwelse, J.A.: Key management for onion routing in a true peer to peer setting. In: IWSEC 2014. LNCS, vol. 8639, pp. 62–71. Springer (2014)
12. Syverson, P.F., Goldschlag, D.M., Reed, M.G.: Anonymous connections and onion routing. In: IEEE Security & Privacy 1997. pp. 44–54. IEEE (1997)
13. Xi, Y., Schwiebert, L., Shi, W.: Preserving source location privacy in monitoring-based wireless sensor networks. In: IPDPS 2006. IEEE (2006)
14. Yang, Y., Shao, M., Zhu, S., Urgaonkar, B., Cao, G.: Towards event source unobservability with minimum network traffic in sensor networks. In: WISEC 2008. pp. 77–88. ACM (2008)
15. Zhang, L.: A self-adjusting directed random walk approach for enhancing source-location privacy in sensor network routing. In: IWCMC 2006. pp. 33–38. ACM (2006)