Design and Implementation of TARF: A Trust-Aware Routing Framework for WSNs

P. Vinod Kumar[1] Narsimha Banothu[2] L. Ravi Kumar[3] G. Charles Babu[4]
[1]M.Tech [2, 4] Asst. Professors [3]Professor & HOD
Department Of Computer Science and Engineering
Holy Mary Institute of Technology and Science Bogaram, Hyderabad

Abstract— The multi-hop routing in wireless sensor networks (WSNs) offers little protection against identity deception through replaying routing information. An adversary can exploit this defect to launch various harmful or even devastating attacks against the routing protocols, including sinkhole attacks, wormhole attacks and Sybil attacks. The situation is further aggravated by mobile and harsh network conditions. Traditional cryptographic techniques or efforts at developing trust-aware routing protocols do not effectively address this severe problem. To secure the WSNs against adversaries misdirecting the multi-hop routing, we have designed and implemented TARF, a robust trust-aware routing framework for dynamic WSNs. Without tight time synchronization or known geographic information, TARF provides trustworthy and energy-efficient route. Most importantly, TARF proves effective against those harmful attacks developed out of identity deception; the resilience of TARF is verified through extensive evaluation with both simulation and empirical experiments on large-scale WSNs under various scenarios including mobile and RF-shielding network conditions. Further, we have implemented a low-overhead TARF module in Tiny OS; as demonstrated, this implementation can be incorporated into existing routing protocols with the least effort. Based on TARF, we also demonstrated a proof-of-concept mobile target detection application that functions well against an anti-detection mechanism.

Keywords: Wireless Sensor Network, Hash Network

I. INTRODUCTION:

The fundamental functionality of wireless sensor network is to collect and return data from the sensor nodes [1]. Data fusion plays an important role because it can save communication bandwidth and power while improving the efficiency of data collection [2]. The popular data fusion methods for wireless networks include address-centric routing [3] and data-centric routing [4]. In address-centric routing, each source node will forward its data along the shortest path while not considering the routing of data fusion; in the data centric routing, the node will analyze the content of the data and process the data with filtering or merging operation [5]. Wireless sensor networks (WSNs) [6] are ideal candidates for applications to report detected events of interest, such as military surveillance and forest fire monitoring. A WSN comprises battery-powered senor nodes with extremely limited processing capabilities. With a narrow radio communication range, a sensor node wirelessly sends messages to a base station via a multi-hop path. However, the multi-hop routing of WSNs often becomes the target of malicious attacks. An attacker may tamper nodes physically, create traffic collision with seemingly valid transmission, drop or misdirect messages in routes, or jam the communication channel by creating radio interference [8]. This paper focuses on the kind of attacks in which adversaries misdirect network traffic by identity deception through replaying routing information. Based on identity deception, the adversary is capable of launching harmful and hard-to-detect attacks against routing, such as *selective* forwarding, wormhole attacks, sinkhole attacks and *Sybil* attacks [9].These current solutions for data fusion exhibit the following disadvantages [11]: Incorporating data fusion operations among nodes introduces significant delay, thus sacrificing the efficiency of the whole network. There is no mature solution in the application layer which fully facilitates the distributed database technology. The middle layer nodes cannot communicate with each other. There is no sharing of information between different branches of the sensor network.

Several protocols have been introduced for Wireless routing. The issues related to the design of the wireless routing protocols are inherently related to the wireless application. Routing protocols are designed for purposes such as quality of service provisioning, energy management and security. A noteworthy on-demand protocol called Dynamic Source Routing (DSR) protocol was developed by Johnson *et al.* [2]. DSR was designed to restrict the bandwidth consumed by control packets in wireless networks by eliminating the periodic table update messages required in the proactive routing protocols. The problem of routing was divided into two areas - route discovery and route maintenance. In order for one host to communicate with another, it must initially discover a suitable route to use in sending packets to that destination. As long as conditions remain unchanged, this route should be maintained as long as it is needed.

Many secure versions such as QoS [13], SQoS [14], Ariadne [15] and CONFIDANT [16] have been developed from the basic design of DSR. The process of identifying routes based on the trust level of nodes has not been addressed in such previous work [3-10]. Also, the major issue is to determine the trust metric based on a given set of parameters/attributes. A secure wireless network has to meet different security requirements [11]: *Confidentiality*: Data which has been transmitted should only be interpreted by the intended receiver. To meet this requirement data encryption is used. *Integrity:* Data should not change during the process of sending. Data integrity must be ensured. *Availability*: Network services should be available all the time and it should be possible to correct failures to keep the connection stable. However, the level of trust that may be associated with each available route should be determined and advertised to the user. Most of the existing research work [3] focuses on confidentiality and integrity. TARF will focus more on the availability as an important factor in securing wireless networks. The proposed protocol has been simulated using OPNET [15] for two important attributes, the battery power and the software configuration. The results show that TARF is able to improve network availability and reduce the routing traffic sent and received by 37.7% and more than 70%, respectively, while still maintaining an acceptable route discovery time and an acceptable delay.

II. TECHNICAL BACKGROUND

The distributed database technology is widely used in the process of collecting data in sensor networks. Researchers exploit languages similar to SQL (Structured Query Language) to aggregate data in the application layer of the sensor networks [8]. The queries with requests of data collecting are sent to all the nodes in the network. Each upper node processes data from its lower nodes and decide what data to be sent to its upper node. The advantages of such mechanism include: it hides the details of implementation under the application layer; it helps users express their requirement easily; it is convenient to make query optimization through in network processing [9]. However, the disadvantage of this mechanism makes it less attractive. The effectiveness of the data collection is low. Every node needs to wait for the responses from other nodes even if those nodes may not fulfill the query conditions. Such loss in efficiency is not acceptable in wireless sensor networks which are extremely sensitive to power consumption.

Another major approach of data fusion is aggregation tree. Greedy incremental tree (GIT) is one of the most popular aggregation tree algorithms [10]. The tree is created gradually through first building the trunk of the tree and then adding different leaves. The initial tree only has a shortest route between the aggregating node and its closest source nodes. The algorithm then selects the closest node to connect to from the remaining source nodes until all the nodes are connected to the tree. The advantage of the tree algorithm is that it is well organized especially in event-driven applications. It can efficiently aggregate data while saving energy effectively. The problem with this approach is that it is difficult to fulfill all the assumptions in the real circumstance. For most sensor networks, the number of the nodes reaches magnitude of thousands and the dispatching area is easily beyond the limit of the aggregation tree. Mobile agents are processes (e.g. executing programs) that can migrate from one machine to another machine (usually in the same system) in order to satisfy requests made by their clients [11]. Mainly, a mobile agent executes on a machine that hopefully provides the resource or service that it needs to perform its job. If the machine does not contain the needed resource/service, or if the mobile agent requires a different resource/service on another machine, the state information of the mobile agent is saved in pre-defined manner first, then the transfer to a machine containing the necessary resource/service is initiated, and the mobile agent resumes execution at the new machine. Advantages of using MAs include low network bandwidth since they only move when they need to continue execution even when disconnected from the network (typically for disconnected mode), ability to clone itself to perform parallel execution, easy implementation, deployment, and reliability. There are many additional good reasons to use mobile agent in an wireless sensor networks environment [12]: Mobile agent reduces wireless sensor network load. Mobile agents allow users to package a conversation and dispatch it to a destination host where interactions take place locally. Mobile agents are also useful when reducing the flow of raw data in the network. When very large volumes of data are stored in remote hosts, that data should be processed in its locality rather than transferred over the network. Mobile agent overcomes network latency. Controlling sensors through a sensor network of substantial size involves significant latencies. For critical wireless sensor networks, such latencies are not acceptable. Mobile agents offer a solution, they can be dispatched from a central controller to act locally and execute the controller‘s directions directly. Mobile agent encapsulates protocols. When data is exchanged in a distributed system, each host owns the code that implements the protocols needed to properly code outgoing data and interpret incoming data. However, as protocols evolve to accommodate new requirements for efficiency or security, it is cumbersome if not impossible to upgrade protocol code properly. As a result, protocols often become a legacy problem. Mobile agents, on the other hand, can move to remote hosts to establish channels based on proprietary protocols. Mobile agent executes asynchronously and autonomously. Mobile motes often rely on expensive or fragile network connections. Tasks requiring a continuously open connection between a mobile device and a fixed network are probably not economically or technically feasible. To solve this problem, tasks can be embedded into mobile agents, which can then be dispatched into the network. After being dispatched, the agents become independent of the process that created them and can operate asynchronously and autonomously. The mobile device can reconnect at a later time to collect the agent. Mobile agent adapts dynamically. Mobile agents can sense their execution environment and react autonomously to changes. Multiple mobile agents have the unique ability of distributing themselves among the hosts in the network to maintain the optimal configuration for solving a particular problem. Network computing is fundamentally heterogeneous, often from both hardware and software perspectives. Because mobile agents are generally computer- and transport layer-independent (dependent on only their execution environments), they provide optimal conditions for seamless system integration. Mobile agents are robust and fault-tolerant. Mobile agents‘ ability to react dynamically to unfavorable situations and events makes it easier to build robust and fault tolerant distributed systems. If a host is being shut down, all agents executing on that machine are warned and given time to dispatch and continue their operation on another host in the network.

A. *TRUST AWARE ROUTING PROTOCOL*

TARF selects routes to the destination based not only on the shortest path but also on several other security oriented attributes of the nodes. Only nodes that match the sender requirements would forward the packet. To protect WSNs from the harmful attacks exploiting the replay of routing information, we have designed and implemented a robust trust-aware routing framework, TARF, to secure routing solutions in wireless sensor networks. Based on the unique characteristics resource-constrained WSNs, the design of TARF centers on trustworthiness and energy efficiency. Though TARF can be developed into a complete and independent routing protocol, the purpose is to allow existing routing protocols to incorporate our implementation of TARF with the least effort and thus producing a secure and efficient fully-functional protocol. Unlike other security measures, TARF requires neither tight time synchronization nor known geographic information. Most importantly, TARF proves resilient under various attacks exploiting the replay of routing information, which is not achieved by previous security protocols. Even under strong attacks such as *sinkhole* attacks, *wormhole* attacks as well as *Sybil* attacks, and hostile mobile network condition, TARF demonstrates steady improvement in network performance. The effectiveness of TARF is verified through extensive evaluation with simulation and empirical experiments on large-scale WSNs. Finally, we have implemented a ready-to-use TARF module with low overhead, which as demonstrated can be integrated into existing routing protocols with ease; the demonstration of a proof-of-concept mobile target detection program indicates the potential of TARF in WSN applications. The main objectives of the proposed TARF suite are:

1 implement security that is inherently built into the routing protocol,
2 deliver messages that are received with a user defined or best available level of confidence, (c) allow users and applications to prescribe their required level of security,
3 achieve efficiency in routing that is improved by limiting control message exchanges,
4 optimize resource usage,
5 obtain graceful network performance degradation, and
6 Develop a protocol suite that adapts to changes in the environment, such as the network topology, the power-level of nodes, etc.

In TARF, the security parameters considered in computing the trust-level of a node in a given route include: software configuration, hardware configuration, battery power, credit history, exposure and organizational hierarchy. Each node evaluates the trust level of its neighbors based on the above parameters and includes it in computing the next hop node in the overall shortest route computation. Due to page limitations, this paper will focus on the implementation and evaluation of the battery power and the software configuration attributes. Below is a description of the battery power and software configuration attributes:

1 Power

In wireless networks, the battery power with which nodes operate is a limited resource. Each node uses its power to not only send and receive, it also behaves as a router by forwarding routing messages and updates. The cryptographic techniques that provide security is computationally intensive, which further increase the power consumption of a node. The node's trust level should be set to low since it cannot guarantee its service. This illustrates that power is an important parameter for evaluating the trust level of a node.

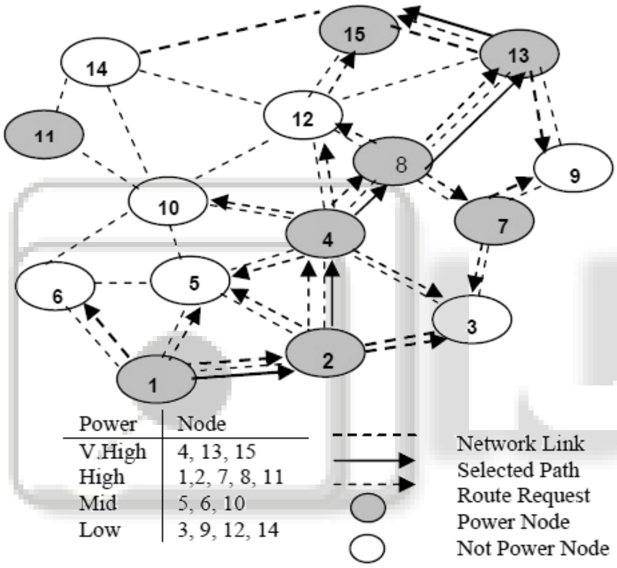
2 Software Configuration

The software configuration includes the encryption ability of a node. To satisfy CAI (Confidentiality, Availability and Integrity), different cryptographic mechanisms have been proposed. Some are based on symmetric encryption and others on asymmetric encryption. Each node is given either a shared secret key or a public/private key pair depending on the type of cryptographic mechanism. Different encryption algorithms are available such as RSA, DES/3DES, BLOWFISH, IDEA, and SEAL RC2/RC4/RC5/RC6 [12]. Strong encryption is often discerned by the key length used by the algorithm. In general, a node with a stronger encryption algorithm has a higher trust level than a node with a weaker encryption algorithm.

B. *Battery Power*

We will adopt the DSR mechanism in finding the shortest path to the destination. However, DSR does not take into consideration the node power factor. We will modify the packet format for the Route Request in the Route discovery mechanism to carry additional two bits, which will allow the sender to choose among four levels of power: LOW, MID, HIGH, V.HIGH.

If node S wishes to communicate with node D, it needs to find a route on demand by using a route discovery mechanism. Node S broadcasts a Route Request packet in the network. This Route Request contains the address of the initiator, the address of the target, a field sequence number (sets the initiator and used to identify the request) and a route record (where a record of the sequence of hops taken by the Route Request is accumulated dynamically). Each node in the network maintains a table in order to detect a duplicate Route Request packet received. A node propagates the Route Request if (i) the intermediate node is not the target, (ii) it is not the first time it receives this packet and (iii) if its power is greater than or equal to the sender ReqPower. The first node receiving this request that has a valid route in its route cache for node D initiates a Route Reply packet back to node S. This Route Reply contains the list of nodes along the path from node S to node D. (Route cache entries will maintain one half the time required for regular DSR, because the node power is changing by time). The first part is the information gathered along the path of the Route Request (that is, from node S to the node replying); the rest of the list is the information found in the route cache of the replying node. Moreover, it may occur that destination node D itself receives a Route Request packet, e.g., no node along the way before node D has an accurate route from node S to node D in its route cache. In this case, node D sends a Route Reply packet containing the path just created dynamically from source S to destination D, i.e., the path traversed by the first Route Request packet received by node D. This path is the minimum delay route from node S to node D. Node D discards all RouteRequest packets corresponding to the same route discovery process after the arriving of the first one. The route maintenance ensures that the paths stored in the route cache are valid. If the data link layer of a node detects a transmission error, the uplink node creates a RouteError packet and transmits it to the original sender of the data packet. This RouteError packet indicates which link is broken, i.e., the node that detected the error and the node it was trying to reach. When a node receives a RouteError packet, it removes the link in error from its route cache and for each route containing this link, truncates the route from the hop before the broken link. TARF ignores the ReqPOWER in the route maintenance procedure. However, the requested power will be considered in the next scheduled transmission. In order to have feedback on the status of each node, several acknowledgement mechanisms may be used, e.g., ACK at the MAC layer level, request of an explicit ACK from the next-hop receiver in the data packet header, or passive ACK (that is, a node overhears the next node forwarding its packets). In Fig. 2, the source node is node 1 and the destination node is node 15. Node 1 will send a RouteRequest to all its neighbors 2, 5 and 6. Node 1 is requesting HIGHPower for its transmission. Each node receiving this request will read the eqPower (found on the sender RouteRequest field of the packet) and compare it to its current power level. In this example, only node 2 will forward the message. The same procedure will be performed by nodes 2, 4, 8, 13 and 15.



Power	Node
V.High	4, 13, 15
High	1,2, 7, 8, 11
Mid	5, 6, 10
Low	3, 9, 12, 14

Figure 2. Route Discovery in Power TARP.

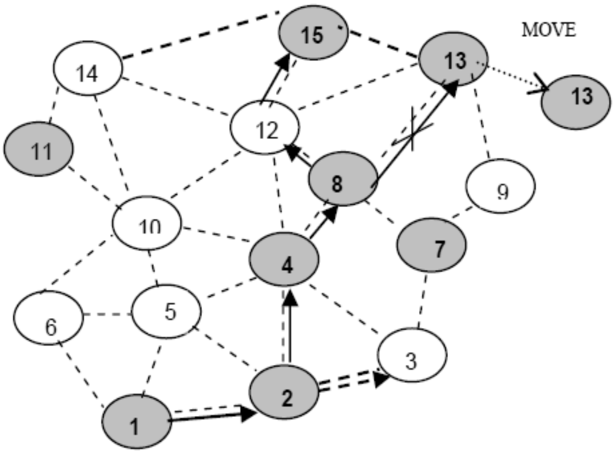


Figure 3. Route Maintenance in Power TARP.

A route reply will take the same route (the destination node, node 15, has also the option to do a route discovery based on the sender ReqPower). If the link between nodes 8 and 13 is unavailable, (see Fig. 3) because node 13 may have shut down or relocated, then node 8 will start a timer and will try to find a PowerRoute to the destination. If node 8 does not find a PowerRoute, it will establish a NONE PowerRoute. However, this path will not be selected the next time the sender initiates a Route Discovery. If there is no PowerRoute to the Destination, the sender will get a message indicating a route error because of the ReqPower.

III. DESIGN CONSIDERATIONS

Before elaborating the detailed design of TARF, we would like to clarify a few design considerations first, including certain assumptions in Section 3.1 and the goals in Section 3.2.

A. *Assumptions*

We target secure routing for data collection tasks, which are one of the most fundamental functions of WSNs. In a data collection task, a sensor node sends its sampled data to a remote base station with the aid of other intermediate nodes, as shown in Figure 1. Though there could be more than one base station, our routing approach is not affected by the number of base stations; to simplify our discussion, we assume that there is only one base station. An adversary may forge the identity of any legal node through replaying that node‘s outgoing routing packets and spoofing the acknowledgement packets, even remotely through a *wormhole*. Additionally, to merely simplify the introduction of TARF, we assume no data aggregation is involved. Nonetheless, our approach can still be applied to cluster based WSNs with static clusters, where data are aggregated by clusters before being relayed [24]. Cluster-based WSNs allows for the great savings of energy and bandwidth through aggregating data from children nodes and performing routing and transmission for children nodes. In a cluster-based WSN, the cluster headers themselves form a sub-network; after certain data reach a cluster header, the aggregated data will be routed to a base station only through such a sub-network consisting of the cluster headers. Our framework can then be applied to this sub-network to achieve secure routing for cluster based WSNs. TARF may run on cluster headers only and the cluster headers communicate with their children nodes directly since a static cluster has known relationship between a cluster header and its children nodes, though any link-level security features may be further employed. Finally, we assume a data packet has at least the following fields: the sender id, the sender sequence number, the next-hop node id (the receiver in this onehop transmission), the source id (the node that initiates the data), and the source‘s sequence number. We insist that the source node‘s information should be included for the following reasons because that allows the base station to track whether a data packet is delivered. It would cause too much overhead to transmit all the onehop information to the base station. Also, we assume the routing packet is sequenced. 2.2 Authentication Requirements though a specific application may determine whether data encryption is needed, TARF requires that the packets are properly authenticated, especially the broadcast packets from the base station. The broadcast from the base station is asymmetrically authenticated so as to guarantee that an adversary is not able to manipulate or forge a broadcast message from the base station at will. Importantly, with authenticated broadcast, even with the existence of attackers, TARF may use Trust Manager and the received broadcast packets about delivery information to choose trustworthy path by circumventing compromised nodes. Without being able to physically capturing the base station, it is generally very difficult for the adversary to manipulate the base station broadcast packets which are asymmetrically authenticated. The asymmetric authentication of those broadcast packets from the base station is crucial to any successful secure routing protocol. It can be achieved through existing asymmetrically authenticated broadcast schemes that may require loose time synchronization. As an example, µTESLA [14] achieves asymmetric authenticated broadcast through a symmetric cryptographic algorithm and a loose delay schedule to disclose the keys from a key chain. Other examples of asymmetric authenticated broadcast schemes requiring either loose or no time synchronization are found in [25], [26]. Considering the great computation cost incurred by a strong asymmetric authentication scheme and the difficulty in key management, a regular packet other than a base station broadcast packet may only be moderately authenticated through existing symmetric schemes with a limited set of keys, such as the message authentication code provided by TinySec [13]. It is possible that an adversary physically captures a non-base legal node and reveals its key for the symmetric authentication [27]. With that key, the adversary can forge the identity of that non-base legal node and joins the network ―legally”. However, when the adversary uses its fake identity to falsely attract a great amount of traffic, after receiving broadcast packets about delivery information. Other legal nodes that directly or indirectly forwards packets through it will start to select a more trustworthy path through Trust Manager.

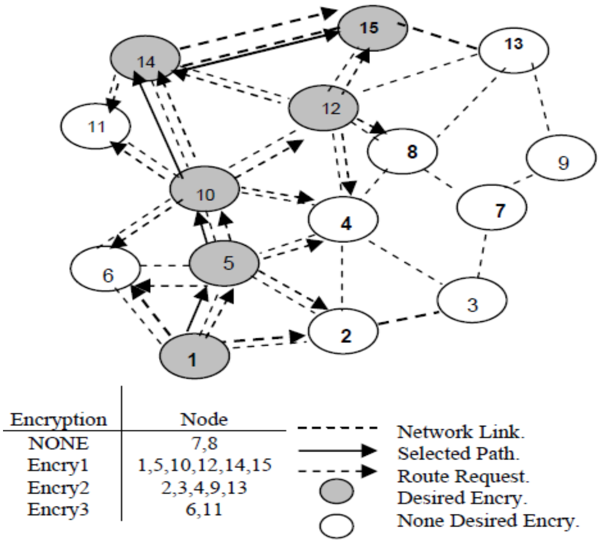
B. *Goals*

TARF mainly guards a WSN against the attacks misdirecting the multi-hop routing, especially those based on identity theft through replaying the routing information. This paper does not address the denial-of-service (DoS) [3] attacks, where an attacker intends to damage the network by exhausting its resource. For instance, we do not address the DoS attack of congesting the network by replaying numerous packets or physically jamming the network. TARF aims to achieve the following desirable properties:

High Throughput is defined as the ratio of the number of all data packets delivered to the base station to the number of all sampled data packets. In our evaluation, throughput at a moment is computed over the period from the beginning time (0) until that particular moment. Note that single-hop re-transmission may happen, and that duplicate packets are considered as one packet as far as throughput is concerned. Through-put reflects how efficiently the network is collecting and delivering data. Here we regard high throughput as one of our most important goals. Energy Efficiency Data transmission accounts for a major portion of the energy consumption; we evaluate energy efficiency by the average energy cost to successfully deliver a unit-sized data packet from a source node to the base station. Note that link-level re-transmission should be given enough attention when considering energy cost since each re-transmission causes a noticeable increase in energy consumption. If every node in a WSN consumes approximately the same energy to transmit a unit-sized data packet, we can use another metric hop-per-delivery to evaluate energy efficiency. Under that assumption, the energy consumption depends on the number of hops, i.e. the number of one-hop transmissions occurring. To evaluate how efficiently energy is used, we can measure the average hops that each delivery of a data packet takes, abbreviated as hop-per-delivery. Scalability & Adaptability TARF should work well with WSNs of large magnitude under highly dynamic contexts. We will evaluate the scalability and adaptability of TARF through experiments with large-scale WSNs and under mobile and hash network conditions. Here we do not include other aspects such as latency, load balance, or fairness. Low latency, balanced network load, and good fairness requirements can be enforced in specific routing protocols incorporating TARF.

IV. DESIGN OF TARF

Data will be distributed among a variety of mobile devices, such as laptops, personal digital assistants (PDAs), mobile phones, in vehicle computer systems, etc. Different secure ad hoc routing protocols might exist in any ad hoc network and the sender should have the right to choose which secure route he might utilize to send data or the sender might choose not to have any security requirements. We will modify the packet format for the RouteRequest in the Route discovery mechanism to carry an additional two bits, in which it will give the sender the chance to select among four encryption mechanisms NONE, Encry1, Encry2 and Encry3 (Encry1, Encry2 and Encry3 could be one of the following: RSA, DES/3DES, BLOWFISH, IDEA, SEAL RC2/RC4/RC5/RC6). Fig. 4 shows the Route Discovery in Encrypted TARF. The sender, node 1, will initiate a RouteRequest to the destination, node 15 asking for Encry1. Nodes 2, 5 and 6 will get the RouteRequest. All the nodes will compare their Encryption mechanism to the Requested Encryption by reading the encryption field in the RouteRequest. Node 5 is the only one which matches the Requested Encryption. Only Node 5 will forward the RouteRequest to all its neighbors.



Encryption	Node
NONE	7,8
Encry1	1,5,10,12,14,15
Encry2	2,3,4,9,13
Encry3	6,11

Figure 4. Route Discovery in Encrypted TARP.

Nodes 10, 12 and 14 will do the same. Node 15 will respond only to the first RouteRequest. If the RouteRequest through node 14 is the first to reach the destination, then 1-5-10-14-15 route will be selected for forwarding the packets and the connection will be established. If node 14 moves away or shuts down, the uplink node (10) will be responsible for finding another secure route to the destination. Node 10 will set a timer, and start the route maintenance procedure in finding the secure route. In this example, the alternative way is through node 12. The new route will then be established. If node 12 fails to establish a route to the destination then it will send back an error message to the sender. The sender will then have the choice to retry and find a secure route with the same encryption criteria or may change to a different one. Power

TARF and Encrypted TARF can work together to form a strong trusted routing protocol. The sender can specify the amount of power and the type of encryption for the transmission.

A. *Routing Procedure*

TARF, as with many other routing protocols, runs as a periodic service. The length of that period determines how frequently routing information is exchanged and updated. At the beginning of each period, the base station broadcasts a message about data delivery during last period to the whole network consisting of a few contiguous packets (one packet may not hold all the information). Each such packet has a field to indicate how many packets are remaining to complete the broadcast of the current message. The completion of the base station broadcast triggers the exchange of energy report in this new period. Whenever a node receives such a broadcast message from the base station, it knows that the most recent period has ended and a new period has just started. No tight time synchronization is required for a node to keep track of the beginning or ending of a period. During each period, the Energy Watcher on a node monitors energy consumption of one-hop transmission to its neighbors and processes energy cost reports from those neighbors to maintain energy cost entries in its neighborhood table; its *Trust Manager* also keeps track of network loops and processes broadcast messages from the base station about data delivery to maintain trust level entries in its neighborhood table. To maintain the stability of its routing path, a node may retain the same next-hop node until the next fresh broadcast message from the base station occurs. Meanwhile, to reduce traffic, its energy cost report could be configured to not occur again until the next fresh broadcast message from the base station. If a node does not change its next-hop node selection until the next broadcast message from the base station that guarantees all paths to be loop-free, as can be deducted from the procedure of next-hop node selection. However, as noted in our experiments, that would lead to slow improvement in routing paths. Therefore, we allow a node to change its next-hop selection in a period when its current next-hop node performs the task of receiving and delivering data poorly.

V. References

[1] F. Akyildiz, W. L. Su, Y. Sankarasubramaniam, and E. Cayirci, ―Asurvey on sensor network,‖ IEEE Commun. Mag., vol. 40, pp. 102- 114, 2002.

[2] W. R. Heizelman, A. Chandrakasan, and H. Balakrishnan, ―Energy- efficient communication protocol for wireless microsensor networks,‖ in Proc. 33rd Hawaii Intl. Conf. Syst. Sci., (HICSS‘00), 2000, pp. 1-10.

[3] C. Intanagonwiwat, R. Govindan, and D. Estrin, ―Directed diffusion: A scalable and robust communication paradigm for sensor networks,‖ in Proc. 6th Annu. ACM/IEEE Intl. Conf. Mobile Comput. Network., (MobiCOM‘00), Boston, MA, August 2000.

[4] A. Manjeshwar, and D. P. Agrawal, ―TEEN: A routing protocol for enhanced efficiency in wireless sensor networks,‖ in Proc. 15th Intl. Parallel Distributed Process Symp., (IPDPS‘01), San Francisco, CA, April 2001.

[5] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar, ―Next century challenges: Scalable coordinate in sensor network,‖ in Proc. 5th ACM/IEEE Int. Conf. Mobile Comput. Network., 1999, pp. 263-270.

[6] G. Zhan, W. Shi, and J. Deng, ―Tarf: A trust-aware routing framework for wireless sensor networks,‖ in Proceeding of the 7th European Conference on Wireless Sensor Networks (EWSN‘10), 2010.

[7] F. Zhao and L. Guibas, Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann Publishers, 2004.

[8] A. Wood and J. Stankovic, ―Denial of service in sensor networks,‖ Computer, vol. 35, no. 10, pp. 54–62, Oct 2002.

[9] C. Karlof and D. Wagner, ―Secure routing in wireless sensor networks: attacks and countermeasures,‖ in Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications, 2003.

[10] M. Jain and H. Kandwal, ―Asurvey on complex wormhole attack in wireless ad hoc networks,‖ in Proceedings of International Con- ference on Advances in Computing, Control, and Telecommunication Technologies (ACT ‗09), 28-29 2009, pp. 555 –558.

[11] I. Krontiris, T. Giannetsos, and T. Dimitriou, ―Launching a sinkhole attack in wireless sensor networks; the intruder side,‖ in Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications(WIMOB ‗08), 12-14 2008, pp. 526 –531.

[12] J. Newsome, E. Shi, D. Song, and A. Perrig, ―The sybil attack in sensor networks: Analysis and defenses,‖ in Proc. of the 3rd International Conference on Information Processing in Sensor Networks (IPSN‘04), Apr. 2004.

[13] L. Bai, F. Ferrese, K. Ploskina, and S. Biswas, ―Performance analysis of mobile agent-based wireless sensor network,‖ in Proceedings of the 8th International Conference on Reliability, Maintainability and Safety (ICRMS 2009), 20-24 2009, pp. 16 –19.

[14] L. Zhang, Q. Wang, and X. Shu,

—A mobile-agent-based middleware for wireless sensor networks data

[15] C. Siva Ram Murthy and B. S. Manoj, —Ad Hoc Wireless Networks: Architectures and Protocols," Prentice Hall, Chapter 7, 1994.

[16] D. B. Johnson and D. A. Maltz, —Dynamic Sources Routing in ad Hoc Wireless Networks," Mobile Computing, 1996

[17] Don Coppersmith and Markus Jakobsson, —Almost Hash Sequence Traversal," In proceeding of the Fourth Conference on Financial Cryptography (FC '02), Lecture Notes in computer Science, 2002.

[18] Yih-Chun Hu and D. B. Johonson, —Securing Quality-of- Service Route Discovery in On-Demand Routing for Ad Hoc Networks," Proceedings of ACM SASN '04, October 20, 2004.

[19] Yih-Chun Hu, A. Perrig, and D. B. Johonson, —Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," Proceedings of ACM MobiCom '02, September 23-26, 2002.

[20] S. Buchegger and Jean-Yves Le Boudec, —Performance Analysis of the CONFIDANT Protocol Cooperation Of Nodes—Fairness In Dynamic Ad-hoc Networks," Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), 2002.

[21] W. Lou and W. Liu, and Y. Fang, —SPREAD: Enhancing Data Confidentiality in Mobile Ad Hoc Networks," Proceedings of IEEE INFOCOM 2004, 2004.

[22] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, —A Secure Routing Protocol for Ad Hoc Networks," Proceedings of IEEE Network Protocols, 2002, pp. 78-87, November 12-515, 2002.

[23] S. Yi, P. Naldurg, and R. Kravets, —A Security-Aware Routing Protocol for Wireless Ad Hoc Networks," Proceedings of ACM MobiHoc ‗01, 2001.

[24] Yih-Chun Hu, D. B. Johnson, and A. Perrig, —SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," Proceedings of IEEE Mobile Computing Systems and Applications, 2002, pp. 3-13, 2002.

[25] D. B. Johnson, —Routing in Ad Hoc Networks of Mobile Hosts," Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications, 1995.

[26] A. Murat Fiskiran and Ruby B. Lee, —Performance Impact of Addressing Modes of Encryption Algorithms," IEEE International Conference on Computer Design (ICCD'01), pp. 542- 545, 2001.

[27] V. D. Park and M. S. Corson, —A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," Proceedings of IEEE INFOCOM '97, pp. 1405- 1413, 1997.

[28] C. E. Perkins and E. M. Royer, —Ad-hoc On-Demand Distance Vector Routing," Proceedings of 2nd IEEE Workshop Mobile Computer Systems and Applications, pp. 90-100, 1999.

[29] OPNET University Program: http://www.opnet.com/services/university/

[30] P. Michiardi and R. Molva, —CORE: A Collaborative Reputation Mechanism to enforce node cooperation in Mobile Ad hoc Networks," IFIP-Communicatin and Multimedia Securtiy Conference 2002.

[31] M. Guerrero Zapata, —Secure Ad hoc On-Demand Distance Vector Routing," Mobile Computing and Com