Volume 4, Issue 3, March 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Intrusion Prevention System (IPS) Countering Ddos Attack in Internetworks

M.Rajakumaran
Department of Computer Science & Engineering
PG Scholar / EGS Pillay Engineering College
India

B.Saravanakumaran
Department of Information Technology
Associate Professor / EG S Pillay Engineering College
India

Abstract*— Distributed denial-of-service (DDoS) attacks remain a major security problem, the mitigation of which is very hard especially when it comes to highly distributed botnet-based attacks. The early discovery of these attacks, although challenging, is necessary to protect end-users as well as the expensive network infrastructure resources. Here, we address the problem of DDoS attacks and present the theoretical foundation, architecture, and algorithms of FireCol. The core of FireCol is composed of Intrusion Prevention Systems (IPSs) located at the Internet Service Providers (ISPs) level. The IPSs form virtual protection rings around the hosts to defend and collaborate by exchanging selected traffic information. The evaluation of FireCol using extensive simulations and a real dataset is presented, showing FireCol effectiveness and low overhead, as well as its support for incremental deployment in real network.

Keywords*— botnet based attacks, DDoS, FireCol, IPSs, Virtual protections

I. INTRODUCTION

A botnet is a large network of compromised machines (bots) controlled by one entity (the master). The master can launch synchronized attacks, such as DDoS, by sending orders to the bots via a Command & Control channel. Unfortunately, detecting a botnet is also hard, and efficient solutions may require participating actively to the botnet itself which raises important ethical issues, or to first detect botnet-related malicious activities which may delay the mitigation. To avoid these issues, this focuses on the detection of DDoS attacks and their underlying vectors. Although non distributed denial-of-service attacks usually exploit vulnerability by sending few carefully forged packets to disrupt a service, DDoS attacks are mainly used for flooding a particular victim with massive traffic as highlighted.

In fact, the popularity of these attacks is due to their high effectiveness against any kind of service since there is no need to identify and exploit any particular service-specific flaw in the victim. Hence, this focuses exclusively on flooding DDoS attacks.A single intrusion prevention system (IPS) or intrusion detection system (IDS) can hardly detect such DDoS attacks, unless they are located very close to the victim. Even in that latter case, the IDS/IPS may crash because it needs to deal with an overwhelming volume of packets. A DDoS resistant communication mechanism is proposed for end-hosts by using acknowledgments. Another solution relies on tokens delivered to each new TCP flow. In each router between the source and the destination marks the path to detect spoofed addresses. Detection of specific SYN flooding attacks at the router level is investigated in. The correlation between the requests and replies to detect flooding attacks to limit overhead is analysed.

The FireCol system maintains virtual rings or shields of protection around registered customers. A ring is composed of a set of IPSs that are at the same distance (number of hops) from the customer. Each FireCol IPS instance analyzes aggregated traffic within a configurable detection window. The metrics manager computes the frequencies and the entropies of each rule A rule describes a specific traffic instance to monitor and is essentially a traffic filter, which can be based on IP addresses or ports the selection manager measures the deviation of the current traffic profile from the stored ones, selects out of profile rules, then forwards them to the score manager. Using a decision table, the score manager assigns a score to each selected rule based on the frequencies, the entropies, and the scores received from upstream IPS.

Using a threshold, quite low score is marked as a low potential attack and is communicated to the downstream IPS that will use to compute its own score. A quite high score on the other hand is marked as high potential attack and triggers ring-level (horizontal) communication in order to confirm or dismiss the attack based on the computation of the actual packet rate crossing the ring surpasses the known, or evaluated, customer capacity the collaboration manager is only invoked for the few selected candidate rules based on resource-friendly metrics.

This paper proposes FireCol protects subscribers based on defined rules. A FireCol rule matches a pattern of IP packets. Generally, this corresponds to an IP sub network or a single IP address. The rule definition can include any other monitorable information that can be monitored, such as the protocols or the ports used. The ring level of a FireCol-enabled router (IPS) is regularly updated based on the degree of stability of IP routing. This is done using a two phase process. First, the router sends a message RMsg to the protected customer containing a counter initialized to 0. The counter is incremented each time it passes through a FireCol-enabled router.

The customer then replies to the initiating router with the value of its ring level. This procedure is optimized through aggregation when several routers are requesting a ring-level update.

II. RELATED WORK

Direct Anonymous Attestation (DAA) is a scheme developed for remote authentication of a hardware module, called Trusted Platform Module (TPM), while preserving the privacy of the user of the platform that contains the module [7]. CL signature scheme and the related protocols as underlying building blocks. For private-key-based revocation, the coding part still use verifier-local revocation, i.e., the revocation check is done only at the verifier's side. For the other two types of revocation, it developers proof of knowledge protocols for proving that a user's membership private key is not listed in the revocation list. This proof of knowledge protocols may be of independent interest in other applications as well. To provide high security scheme a concept called DRAFT has been implemented to check the integrity status of the intermediate node to gain the attestation. DAA-certificates need to be issued only once (no bottleneck) Issuer and verifier cannot link DAA-certificates and DAA-signatures, even if they are the same entity ("repairs the broken business model"). Anonymity degradation is possible (named base vs. random base).

Different attestations are linkable, if the same AIK is used multiple times. Thus owners should always create fresh AIKs. The Privacy-CA is a very sensitive entity. Therefore it must be carefully protected and maintained to guarantee security.

Privacy-preserving routing is crucial for some ad hoc networks that require stronger privacy protection. It define stronger privacy requirements regarding privacy-preserving routing in mobile ad hoc networks [4]. In order to provide location authentication and location privacy simultaneously .the protocol also authenticates the routing paths taken by individual messages. Achieving anonymity is a different problem than achieving data confidentiality. While data can be protected by cryptographic means, the recipient node address (and may be the sender node address) of a packet cannot be simply encrypted because they are needed by the network to route the packet. ALERT can offer high anonymity protection at a low cost when compared to other anonymity algorithms. Route messages to desired geographic locations in the presence of malicious nodes. Detect and avoid bad geographic regions containing malicious or faulty nodes. Authenticate self-generated public keys and geographic locations of nodes on the routing path.

Distributed denial of service is flooding of network with unrelated information by malicious node [6]. A technique called iHoneyCol which effectively mitigate distributed denial of service rather than present filtering approach. iHoney Col is integration of Firecol and Honey pot. Attack prevention and pre-emption, where the attack is prevented at client side itself so that the mitigation is done far from the destination. Pre-emption is when the attacker is authorized to send any malicious data. They get swapped by neighbouring net-work devices. Attack detection and filtering, where an attack is detected and they are filtered according to the traffic pattern registered at network devices. This filtering technique can be embedded into firewall through software or it can use separate hardware devices. Attack source trace back and identification, once the attack has been identified the main source of at-tack is detected. Their individual IP address is added to black list by honey pot severs

Botnets are a very real and quickly evolving problem that is still not well understood [1]. It outline the problem and investigate methods of stopping bots it identify three approaches for handling botnets: (1) prevent systems from being infected, (2) directly detect command and control communication among bots and between bots and controllers, and, (3) detect the secondary features of a bot infection such as propagation or attacks. The first approach is to prevent systems from being infected. It's preventing all systems on the Internet from being infected by attackers. It's detecting botnets by correlating secondary detection information to pinpoint bots and botnet communication. It is a multi-detector correlational approach will provide a more robust and longer-term botnet detection system.

III. DESIGN CONSIDERATIONS

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 Fig. 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 acknowledgment packets, even remotely through a wormhole.

B. Authentication requirements

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 [8]. It is possible that an adversary physically captures a nonbase legal node and reveals its key for the symmetric authentication [9]. With that key, the adversary can forge the identity of that nonbase 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.

C. Goals

TARF mainly guards a WSN against the attacks misdirecting the multihop 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. 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 retransmission may happen, and that duplicate packets are considered as one packet as far as throughput is concerned. Throughput reflects how efficiently the network is collecting and delivering data.

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 and 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. CONSTRUCTION OF FIRECOL AND HONEYPOT

The integration of "Firecol" and "Honeypot" helps in mitigating distributed denial of service to acceptable amount. The honeypot provides an organization information on their own security risk and vulnerabilities. It should consist of similar system and application that one used by organisation's for its productive environment.

So to give the attacker a real world feeling and to be able to implement the learned lessons in productive environment. So we have planned to integrate the core concept of firecol and honeypot in order to achieve higher efficiency and provide better performance.

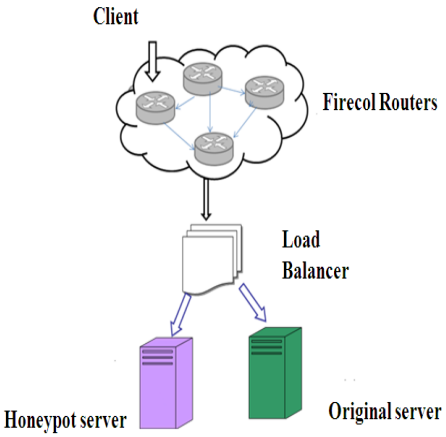

Fig1 Architecture of firecol and honeypot

The above fig 1 shows the flow of traf-fic and identity of malicious traffic in an efficient way. In any network environment the client register themselves with their own ISP,s. After the registration is fulfilled, they are notified as an authorized client. Here two major problem of a network are addressed. They are clone at-tack and PoD attack.

A. Clone attacks

These attacks are nowadays called as "twin attack". Its nothing but when an unauthorized client spoof the IP address of any authorized client to flood the network.

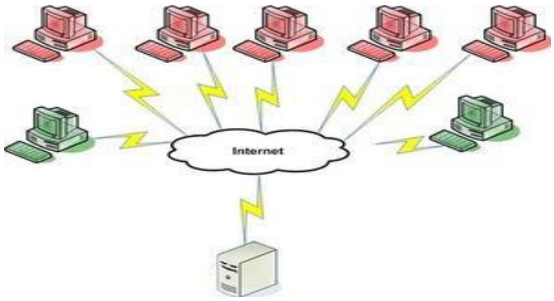

Fig 2. Clone attack

It also has a special case when an authorized client himself spoofs the IP address of any other authorized cli-ent. Here comes the function of firecol in a smart way. The proposed solution is when all the client register themselves with ISP,s they also send their individual IP address, their location and time to the firecol router. The firecol router in turn response with the ACK packet and generate individual random number to all nodes. These random numbers are assigned to all nodes. The IP ad-dress, location and random number assigned to nodes. This information gets stored in routing table of firecol router.

B. ***PoD attack***

PoD, in general known as ping of death. This attack can be defined as every data packet contains ICMP head-er which sends ECHO REQUEST and ECHO REPLY. If the ICMP data header exceeds 65,536 bytes crashes the entire system.

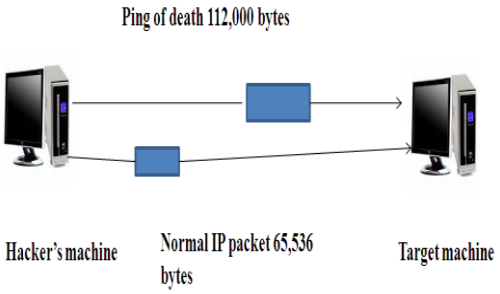

Fig 3. PoD attack

In existing work these can be overcome by fragment-ing the data. The main drawback of this system is that the destination system cannot handle larger fragment of data, so they crashes. The proposed solution is disallowing the client themselves to send larger amount of data. So that the traffic has been blocked away from the des-tination. The client attempt to send such a traffic flood will be recorded as black listing client by honeypot server. The other solution is "virtual fragmentation" of ICMP header packet.

That is, as soon as the client sends such a larger amount of ICMP data, they get fragmented at firecol and pass the original data to original server. These activities are broadcasted to honeypot server by firecol. So at final honeypot server disconnects particular clients TCP connections. So that they are not eligible to transfer the data to any of the neigh boring nodes. So by virtual fragmenting the data the information loss can be reduced. The virtual fragmentation is done by firecol router without the knowledge of client. So the user thinks that they are going to crash the sys-tem (disguises themselves).

V. VERIFICATION OF ISSUER'S PUBLIC KEY

Given the group public key (N, g', g, h, R, S, Z, p, q, u) and the proof that g, h, S, Z, R are formed properly, any user in the system can verify the correctness of the group public key as follows:

1. Verify the proof that g,h€ g' and R,S,Z £ h.
2. Check whether p and q are primes, q | (p-), q|(p-1)/q and uq =1(mod p).
3. Check whether all public key parameters have the required length.

If g, h, R, S, Z are not formed correctly, it could potentially mean that the security properties for the users do not hold. However, it is sufficient if the users verify the proof that g, h, R, S, Z are computed correctly only once.

Also, if u does not generate a subgroup of Z_p, the issuer could potentially use this to link different signatures. As argued in [1], it is not necessary to prove that N is a product of two safe primes for the anonymity of the users. In fact, it would be very expensive for the issuer to prove that N is a safe-prime product [10].

VI. PROOF OF MEMBERSHIP FOR RESOURCE-CONSTRAINED DEVICES

If the prover is a resource-constrained device, such as a TPM, a smart card, or a secure coprocessor; it can outsource part of the signing operation to a semi trusted host. Essentially, the signing operation is split between a computationally weak device (denoted as the principal prover) and a resource abundant
but less-trusted host. Observe that if the host does not operate, then it is a denial of service. Thus, the host platform is trusted for performing its portion of computation correctly. However, the host is not allowed to learn the private key of the prover or to forge a signature without the principal prover's involvement. This model is used in the original DAA scheme [1] with a concrete security model.

For EPID, the same technique from [1] can be applied. Let (A; e; f; v) be the principal prover's private key. The principal prover sends (A; e) to the host but keeps (f; v) secretly. The signing operation in the proof of membership can be conducted as follows:

1. The principal prover picks a random B ← (u) and computes K: =¼ B2 mod p.
2. The principal prover sends (B,K) to the host.
3. The host randomly chooses two integers (w, r) ← (0, 1).

VII. ANALYSIS OF ENERGY WATCHER AND TRUST MANAGER

Now that a node N relies on its Energy Watcher and Trust Manager to select an optimal neighbour as its next-hop node, we would like to clarify a few important points on the design of Energy Watcher and Trust Manager. the energy cost report is the only information that a node is to passively receive and take as "fact." It appears that such acceptance of energy cost report could be a pitfall when an attacker or a compromised node forges false report of its energy cost.

Note that the main interest of an attacker is to prevent data delivery rather than to trick a data packet into a less efficient route, considering the effort it takes to launch an attack. As far as an attack aiming at preventing data delivery is concerned, TARF well mitigates the effect of this pitfall through the operation of Trust Manager. Note that the Trust

Manager on one node does not take any recommendation from the Trust Manager on another node. If an attacker forges false energy report to form a false route, such intention will be defeated by Trust Manager: when the Trust Manager on one node finds out the many delivery failures from the broadcast messages of the base station, it degrades the trust level of its current next-hop node; when that trust level goes below certain threshold, it causes the node to switch to a more promising next-hop node.

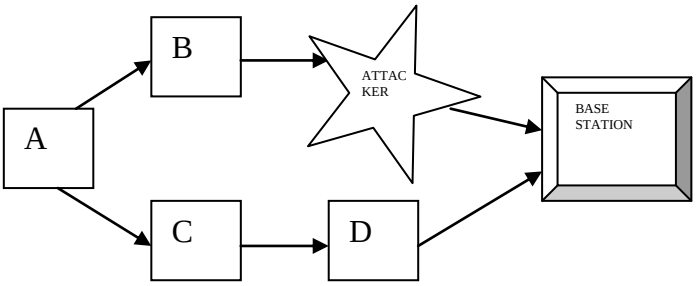

Fig. 4. An example to illustrate how Trust Manager works.

Second, Trust Manager identities the low trustworthiness of various attackers misdirecting the multihop routing, especially those exploiting the replay of routing information. It is noteworthy that Trust Manager does not distinguish whether an error or an attack occurs to the next-hop node or other succeeding nodes in the route. It seems unfair that Trust Manager downgrades the trust level of an honest next-hop node while the attack occurs somewhere after that next-hop node in the route. Contrary to that belief, Trust Manager significantly improves data delivery ratio in the existence of attack attempts of preventing data delivery. First of all, it is often difficult to identify an attacker who participates in the network using an id "stolen" from another legal node. For example, it is extremely difficult to detect a few attackers colluding to launch a combined wormhole and sinkhole attack [4]. Additionally, despite the certain inevitable unfairness involved, Trust Manager encourages a node to choose another route when its current route frequently fails to deliver data to the base station.

Though only those legal neighbouring nodes of an attacker might have correctly identified the adversary, our evaluation results indicate that the strategy of switching to a new route without identifying the attacker actually significantly improves the network performance, even with the existence of wormhole and sinkhole attacks. Fig. 4 gives an example to illustrate this point. In this example, nodes A, B, C, and D are all honest nodes and not compromised. Node A has node B as its current next-hop node while node B has an attacker node as its next-hop node. The attacker drops every packet received and thus any data packet passing node A will not arrive at the base station. After a while, node A discovers that the data packets it forwarded did not get delivered. The Trust Manager on node A starts to degrade the trust level of its current next-hop node B although node B is absolutely honest. Once that trust level becomes too low, node A decides to select node C as its new next-hop node. In this way, node A identifies a better and successful route (A - C - D - base).

Finally, we would like to stress that TARF is designed to guard a WSN against the attacks misdirecting the multihop routing, especially those based on identity theft through replaying the routing information. Other types of attacks such as the denial-of-service [3] attacks are out of the discussion of this paper. For instance, we do not address the attacks of injecting into the network a number of data packets containing false sensing data but authenticated (possibly through hacking). That type of attacks aims to exhaust the network resource instead of misdirecting the routing. However, if the attacker intends to periodically inject a few routing packets to cause wrong route, such attacks can still be defended by TARF through Trust Manager.

VIII. **IMPLEMNTATION**

In order to evaluate TARF in a real-world setting, we implemented the Trust Manager component on TinyOS 2.x, which can be integrated into the existing routing protocols for WSNs with the least effort. Originally, we had implemented TARF as a self-contained routing protocol [1] on TinyOS 1.x before this second implementation.

However, we decided to redesign the implementation considering the following factors. First, the first implementation only supports TinyOS 1.x, which was replaced by TinyOS 2.x; the porting procedure from TinyOS 1.x to TinyOS 2.x tends to frustrate the developers. Second, rather than developing a self-contained routing protocol, the Second implementation only provides a Trust Manager component that can be easily incorporated into the existing protocols for routing decisions. The detection of routing loops and the corresponding reaction are excluded from the implementation of Trust Manager since many existing protocols, such as Collection Tree Protocol [12] and the link Connectivity-based protocol [11], already provide that feature.

As we worked on the first implementation, we noted that the existing protocols provide many nice features, such as the analysis of link quality, the loop detection and the routing decision mainly considering the communication cost. Instead of providing those features, our implementation focuses on the trust evaluation based on the base broadcast of the data delivery, and such trust information can be easily reused by other protocols. Finally, instead of using TinySec [8] exclusively for encryption and authentication as in the first implementation on TinyOS 1.x, this re-implementation let the developers decide which encryption or authentication techniques to employ; the encryption and authentication techniques of TARF may be different than that of the existing protocol.

IX. CONCLUSIONS

We have designed and implemented TARF, a robust trust aware routing framework for WSNs, to secure multihop routing in dynamic WSNs against harmful attackers exploiting the replay of routing information. TARF focuses on trustworthiness and energy efficiency, which are vital to the survival of a WSN in a hostile environment. With the idea of trust management, TARF enables a node to keep track of the trustworthiness of its neighbours and thus to select a reliable route. Our main contributions are listed as follows:

1. Unlike previous efforts at secure routing for WSNs, TARF effectively protects WSNs from severe attacks through replaying routing information; it requires neither tight time synchronization nor known geographic information.
2. The resilience and scalability of TARF are proved through both extensive simulation and empirical evaluation with large-scale WSNs; the evaluation involves static and mobile settings, hostile network conditions, as well as strong attacks such as wormhole attacks and Sybil attacks.
3. We have implemented a ready-to-use TinyOS module of TARF with low overhead; as demonstrated in the paper, this TARF module can be integrated into existing routing protocols with the least effort, thus producing secure and efficient fully functional protocols.
4. Finally, we demonstrate a proof-of-concept mobile target detection application that is built on top of TARF and is resilient in the presence of an ant detection mechanism that indicates the potential of TARF in WSN applications.

And iHoneycol provides a collaborative solution for the early detection of flooding DDoS attacks by making use of "Firecol-IPS" system and "Honeypot-IDS" system. It prevent the attack as close to the source and as far from destination, providing a protection to sub-scribed customers and saving valuable network re-sources. Also, the study of iHoneyCol demonstrated its light computational as well as communication overhead.

Being offered as an added value service to customers, the accounting for iHoneyCol is therefore facilitated, which represents a good incentive for its deployment by ISPs. In general, iHoneycol which overcomes twin attack and ping of death attack in an efficient manner using a collaborative technique.

REFERENCES

[1] Cooke.E, Jahanian.F and McPherson.(2005) "The zombie roundup: Understanding, detecting, and disrupting botnets," in Proc. SRUTI, pp. 39–44.

[2] Bhattacharyya.S, Zhang.Z.L AND Xu.K(2008) "Internet traffic behavior profiling for network security monitoring," IEEE/ACM Trans. Netw., vol. 16, no. 6, pp. 1241–1252.

[3] Biersack.E, Dahl.F, Freiling.F and Steiner.M(2008) "Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm," in Proc. USENIX LEET, Article no. 9.

[4] Gil.T.M and Poletto.M.(2001)"ALERT: An Anonymous Location-Based Efficient Routing Protocol in MANETs ," in Proc. 10th USENIX Security Symp, pp. 23–38.

[5] Leckie.C,Pen.T and Ramamohanarao.(2003)"Design and Implementation of TARF: A Trust-Aware Routing Framework for WSNs," in Proc. IEEE ICC, vol. 1, pp. 482–486.

[6] Maglaris.Band Siaterlis.c,(2004)"IHONEYCOL: A COLLABORATIVE TECHNIQUE FOR MITIGATION OF DDoS ATTACK ," in Proc. Int. Symp. Comput. Commun. vol. 1, pp. 339 –344.

[7] Mao.Z.M, Zhang.M and Zhang.Y. (2009),"Enhanced Privacy ID: A Direct Anonymous Attestation Scheme with Enhanced Revocation Capabilities," in Proc. ACM SIGCOMM Conf. Internet Meas. pp. 103–115.

[8] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: A Link Layer Security Architecture for Wireless Sensor Networks," Proc. ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys '04), Nov. 2004.

[9] P. De, Y. Liu, and S.K. Das, "Modeling Node Compromise Spread in Wireless Sensor Networks Using Epidemic Theory," Proc. Int'l Symp. World of Wireless, Mobile and Multimedia Networks (WoWMoM '06), pp. 237-243, 2006.

[10] J. Camenisch and M. Michels, "Proving in Zero-Knowledge that a Number is the Product of Two Safe Primes," Proc. Int'l Conf. Theory and Application of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT '99), pp. 106-121, 1999.

[11] A. Woo, T. Tong, and D. Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks," Proc. First ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys '03), Nov. 2003.

[12] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection Tree Protocol," Proc. Seventh ACM Conf. Embedded Networked Sensor Systems (SenSys '09), pp. 1-14, 2009.

