

Dynamic Authentication-Key Re-assignment for Reliable Report Delivery

Weijia Li, Youtao Zhang
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260

Jun Yang
Department of Electrical and Computer Engineering
University of Pittsburgh
Pittsburgh, PA 15260

Abstract—Sensor networks deployed in hostile environments are subject to various types of attacks. While multipath routing and en-route authentication schemes have been proposed to defend packet dropping and injection attacks respectively, it is challenging to defend both at the same time. The paper addresses this problem through an annulus based authentication-key reassignment scheme in multipath routing with en-route authentication. Our experimental results show that the proposed scheme achieves better trade-offs in true report delivery, false report filtering and energy consumption.

I. INTRODUCTION

Sensor networks, while they can collect crucial data from nontraditional environments such as hostile battle fields, are subject to many security attacks. As the example in Fig. 1, data reports are forwarded through a multi-hop routing path from the sensors that detect the enemy tank to the sink node (a soldier). Both sensing and relay nodes may be compromised — they may drop reports (e.g. node A) or inject false reports (e.g. node B) such that the sink may reach sub-optimal or even wrong decisions with severe consequences. In addition, sensors are usually powered with batteries that are difficult to re-charge. Wasting the energy to route false reports significantly reduces the lifetime of the network.

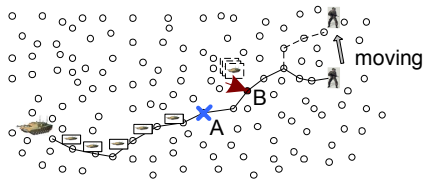


Fig. 1. Collecting data in hostile environments.

Security enhancement schemes have been proposed separately to defend different types of attacks. More true reports can be delivered if they are forwarded along multiple routing paths, which effectively defends

packet dropping along some paths. The topology of those paths may be *braided* [1], mesh-based [9] or even *broadcasting*-based [12]. En-route false report filtering schemes [11], [15], [10] were designed to reject false reports as early as possible in routing, which minimizes the energy wasted to route the false reports.

However it is still challenging to defend both attacks simultaneously. Our observation is that with multipath routing, false reports at a relay node may be forwarded to multiple next hop nodes. While existing en-route filtering schemes may be adopted, unless these reports are detected and dropped along all paths, it is possible that some copies can still reach the sink wasting a large amount of the routing energy. To effectively defend both types, we proposed an annulus based key re-assignment scheme with following contributions.

- We identify security problems in multipath routing: (i) copies of false reports may be delivered; (ii) a new attack which intentionally attaches one wrong MAC, may result in wasting more routing energy.
- We propose an annulus region based authentication reassignment scheme for more reliable report delivery. Based on a recent group rekeying scheme PCGR [14], we introduce a novel access control mechanism to achieve dynamic re-assignment.
- We simulate and evaluate the proposed scheme. Our experimental results show that the proposed scheme achieves better trade-offs in true reports delivery, early false reports filtering, and low energy consumption per delivered report.

The remainder of the paper is organized as follows. Section II discusses the related work. Section III defines the problem and elaborates different types of attacks that we are to defend. Our algorithm is discussed in Section IV with experimental evaluation presented in Section V. Finally, Section VI concludes the paper.

II. RELATED WORK

A. Packet dropping and multipath routing

With the focus on routing, Karlof *et al.* [4] identified a variety of security attacks including packet dropping attack in which a compromised node selectively forwards its received packets, or creates a sinkhole by luring all traffic around and then drops the packets. [4] suggests to counter packet dropping through multipath routing [1] where multiple node-disjoint or link-disjoint routing paths are set up to route data reports.

Ganesan *et al.* proposed braided link-disjoint multipath [1] which has one primary path and several alternative paths that are braided around the primary one. Other schemes [3] have also been developed. The study in [1], [9] showed that link-disjoint schemes are energy efficient and can counter most individual failures in sensor networks. Similar robustness from redundant routing can also be achieved from the gradient broadcast scheme [12] in which packets are forwarded to a gradient that is defined by a cost function. In this paper we adopt link-disjoint multipaths.

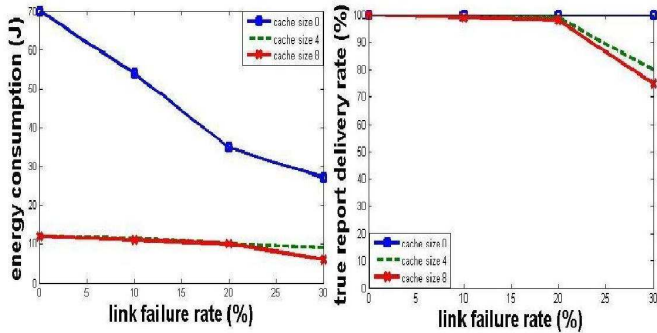


Fig. 2. Merging same copies saves the energy while maintaining similar delivery rate.

When a relay node receives multiple copies of the same report, it can either forward or suppress the same copy. Fig. 2 plots the energy consumption and the delivery rate under different link failure rates. As we can see, suppressing duplicate copies maintains similar report delivery rates while consuming significant less energy. For this reason, we maintain a four-entry cache on each node to remove duplicate copies if found in cache.

B. En-route false report filtering

En-route false report filtering schemes have been recently developed to defend false reports injected by compromised nodes. Ye. *et al.* proposed a statistical en-route authentication algorithm. Zhu *et al.* [15] proposed an interleaved hop-by-hop authentication scheme. Yang *et al.* proposed to incorporate location information to achieve better resilience [10].

The following briefly reviews SEF [11] which motivates our design. Before node deployment, a global key list is determined and then divided to several partitions. A sensor node is randomly assigned to one partition and loaded with a subset of keys in that partition. To endorse a sense reading, each sensing node also generates a MAC using its authentication key. From received readings and MACs, an aggregation node constructs a report and forwards it to the sink. A relay node, if it shares a key for generating one of these MACs, can verify whether the report has been tampered with. The report is forwarded further if the verification succeeds and dropped otherwise.

C. Secure information exchange and key management

Key management schemes have been proposed for secure information exchange in the sensor network. Perrig *et al.* [8] proposed μ TELSA to authenticate information transferred from the sink. [5] proposed a general framework to setup pairwise keys between two sensor nodes. Pairwise keys can be used to encrypt or authenticate exchanged information.

Zhang *et al.* [14] proposed to periodically update authentication keys and exclude the compromised nodes from updating the keys. A node distributes a polynomial function to the neighbors and dynamically generates a new key by collecting shares from its neighbors. Our scheme is developed based on [14]. The difference between their scheme and ours is that in their scheme nodes are statically grouped and only the key content gets updated after deployment. The access control mechanism proposed in our algorithm enables us to flexibly regroup nodes after deployment and update keys accordingly.

III. PROBLEM STATEMENT

A. Network model

In this paper we consider sensor networks deployed in hostile environments. Such a network consists of a trustworthy sink node and a large number of battery powered unattended sensor nodes.

The deployed nodes monitor events of interests and send the data reports back to the sink. We assume that an interesting event is detected by multiple surrounding nodes and the majority of them are trustworthy. Readings are first aggregated to data reports by some selected cluster head nodes [13] and then forwarded along the routing path to the sink. We adopt a multi-hop link-disjoint braided multipath routing scheme [1].

We assume sensor nodes are quasi-stationary after deployment whose locations may change within a small

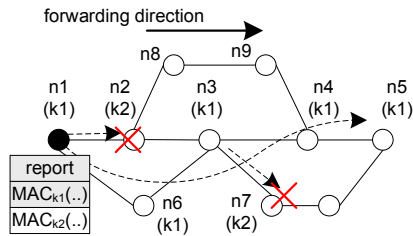


Fig. 3. False reports may be delivered in multipath routing.

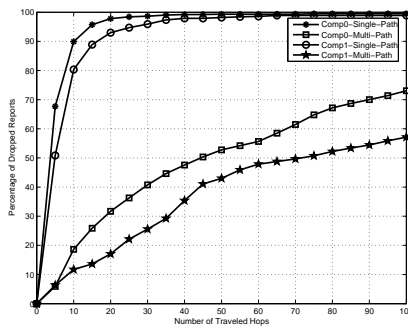


Fig. 4. Degraded false report filtering.

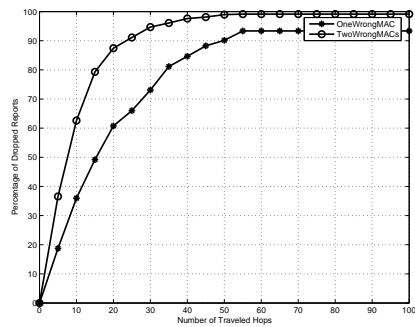


Fig. 5. True reports can be dropped.

range. As shown later the location inaccuracy is tolerable in our scheme. On the other hand, the sink may move around for better flexibility and security, i.e. the soldier not captured by the enemy.

B. Attack model

We assume that an unattended sensor node may be compromised with its stored secure information extracted. We further assume that the security enhancement schemes (the new algorithm and others) are well-known.

We are to defend three types of attacks.

A1: Report dropping attack. A compromised node may selectively forward some or completely drop all of its received reports. There are two types — (i) a compromised relay node drops its received packets; (ii) a compromised data aggregation node refuses to generate a report even if it receives multiple sensing readings. We further assume that compromised nodes can eventually be determined by neighboring watchdog nodes [7]. However it may not be done timely due to the signal conflicting, sensing range limitation, or the compromise of the watchdog nodes (discussed in [7]).

A2: False data injection attack. A compromised node may selectively inject false reports. When an en-route authentication scheme [11], [10], [14] is adopted, a data report is attached with multiple MACs. The adversary can generate consistent MACs for the false report depending on if it has the corresponding authentication keys. For example, with one compromised node and its authentication key, the adversary can generate one consistent MAC but have to guess other MACs.

A3: False MAC injection attack. This is a new type of attack that we develop in this paper, targeting at defeating en-route authentication. Since a data aggregation node has limited authentication keys (usually the same as other nodes), it cannot verify the MAC generated from a sensing node. Instead the received MACs are blindly used. A compromised sensing node thus can send in the correct sensing reading but a wrong MAC. In this

case, the constructed report contains a wrong MAC but not known to the aggregation node resulting in a high possibility of being dropped in the middle of routing. It is always dropped at the sink even if it is received. This is a complementary attack to A2 which constructs a false report with wrong report content.

The false MAC injection attack can also happen at a compromised relay node which arbitrarily modifies some bits of the Bloom filter (or MACs) but not the content. The attack is worse than either packet dropping or false data injection since it not only blocks the delivery of true reports but also wastes the routing energy.

C. Problem description

While schemes have been proposed to defend A1 and A2 attacks, they are developed independently. On the other hand, we cannot restrict attack types in a more realistic environment. In this section we study if simply combining developed schemes can effectively defend A1–A3 attacks simultaneously.

a) Problem 1: false reports are delivered in multipath routing: Fig. 3 illustrates why a false report from node n1 can reach the sink node n5 in multipath routing. Two keys (k1 and k2) are used for discussion purpose while the results shown next are evaluated with the same setting as that in [11].

We assume node n1 (with k1) is compromised such that a consistent MAC_{k1} can be generated for the false content. When routing this false report in the network, nodes with k2 (i.e. n2 and n7) can detect and discard it. However due to the path n6–n3–n4–n5 on which no node has k2, the false report will be delivered to the sink without en-route detection. To investigate why SEF loses its statistical effectiveness in multipath routing, we further study the routing behavior in more detail. There are multiple copies for the same report. These copies are generated (e.g. at n1 and n4), and merged (at n3) at different times. The latter is due to the reason in Section II (using a small cache of 4 entries at each

node). As we can see, dropping this report at node n2 only saves the routing energy on *partial path n2-n8-n9*. Node n4 still forwards the report regardless of the detection/dropping at node n2. This is opposite to the case in single path where dropping a false report at one node stops its forwarding in the future path.

Fig. 4 compares the effectiveness of en-route authentication with single path and multipath routing. The experimental setting is in Section V. Fig. 4 shows the accumulated percentage of false reports dropped regarding the number of hops they travel in the network. In single path routing we see effective defense of detecting and dropping all false reports. However in multipath routing, around 28% and 43% false reports are delivered if zero and one key are compromised respectively. With zero key compromised, none of the attached MACs is consistent with the false content.

Since duplicate copies of a report are generated and merged at different times, different copies travel different number of hops. A more accurate evaluation of wasted energy is to count how many wireless data transfers (defined as one report receive/send at a node) are wasted for routing these false reports. Without authentication protection, around 70K transfers are wasted to route 100 false reports in multipath setting (the setting is in Section V). With SEF authentication, 30K and 22K transfers are saved with zero and one compromised key respectively, i.e. 57% and 68% are still wasted. This indicates that SEF should be enhanced in multipath routing.

b) Problem 2: true reports are discarded due to false MACs: To evaluate attack type A3, we assume that a compromised data aggregation node tampers with one or two received MACs and includes them in constructing Bloom filter bit vector. Fig. 5 shows the percentage of packets that are dropped at different times (with single path routing). The point (x=20,y=60) indicates that 60% of these packets are dropped within 20 hops. With one and two introduced false MACs respectively, 93% and 99% of the packets are dropped and on average they travel 16 and 10 hops before being dropped. Since these reports cannot be accepted by the sink anyway, the large number of traveled hops indicates more wasted routing energy than that from A2.

If it is a sensing node that injects the false MACs, the data aggregation node may not select these false MACs and thus some true reports may be delivered.

D. Design goals

To summarize, our design goal is to defend all three types of attacks (A1/A2/A3) in a multipath routing

sensor network with en-route authentication. We strive to achieve better trade offs among true report delivery, early false reports filtering, and energy control.

IV. OUR ALGORITHM

A. Algorithm overview

Our algorithm is divided into following phases.

Phase 1: Network initialization.

Phase 2: Setting up the routing paths and selecting multiple data aggregation nodes.

Phase 3: Endorsing the report.

Phase 4: En-route MAC authentication.

Phase 5: Region-based group key reassignment.

Phase 6: Sink verification.

B. Algorithm description

Network initialization. Before deployment each sensor node is loaded with a unique ID. It also gets secure keys for μ TESLA [8] (which is to authenticate the control package from the sink). Within a short and non-compromised time interval after the deployment, sensor nodes exchange information with their neighbors and set up pairwise keys for securing the future communication between each node pair. This assumption is reasonable as shown in [14].

For security purposes, the authentication keys need update after the network initialization. So we use a t-degree g-polynomial function $g(x)$ to do the key re-generation. Firstly, all sensor nodes are loaded with the same function $g(x)$ and a random function f_{random} . In order to protect $g(x)$ from compromising, each node u randomly picks a bivariate e-polynomial $e_u(x,y)$ and distributes the share $e_u(x,v_i)$ to its n neighbors ($i=0,\dots,n-1$), i.e. v_i is a neighbor node id. Node u keeps

$$g'(x) = g(x) + e_u(x,u)$$

but removes $g(x)$ and $e_u(x,y)$ after initialization. After receiving the key re-generation notice from the sink node, each node asks its neighbours for the share $e_u(c,v_i)$, and then re-construct the function $e_u(c,y)$. As it keeps the function $g'(x)$, it can calculate $g(c)$ by

$$g(c) = g'(c) - e_u(c,u)$$

which is its new key.

The theoretical discussion of generating a group key from $g()$ is in [14]. The difference here is, [14] assigns a different $g_i(x)$ for nodes in each statically decided group i and lacks the ability to change groups after deployment. However in our algorithm, while the same $g()$ is assigned to all nodes, we design an access control mechanism

to dynamically regroup these nodes and securely regenerate authentication keys.

Each node also needs to estimate its location according to secure localization algorithms [6] and distributes its coordinate $[loc_{ux}, loc_{uy}]$ to its neighbors that receive e_u shares. The relationship is then fixed after initialization (we will use redundant shares to tolerate more node compromising in future work). While a sensor may slightly change its place, the location is not re-estimated — our experimental results show that reasonable location errors work sufficiently well.

Event sensing, data aggregation and report routing.

Before discussing key reassignment, let us first discuss the routing and en-route authentication behavior.

As shown in SEF [11], each interesting event is sensed by a certain number of surrounding sensor nodes. We distinguish two cases.

- Most reports are transferred under the *normal* case, i.e. sensors have been refreshed with new keys, multiple routing paths have been set up, multiple aggregation nodes (ANs, determined using clustering algorithm [13]) have been selected and advertised. Reports are then routed along these paths to the sink. We will focus on the normal case.
- If an event is detected in a region with expired or non-trustworthy authentication keys, the sensing node generates the MAC using its private key (i.e. only known to the sink). The reading together with its MAC are sent back through multi-hop multi-path routing *without* en-route authentication.

This case exists for a short period of time and thus is not an issue. While false reports may not be timely detected (no en-route authentication), some copies of the report can still be received due to multipath routing. Authentication keys can then get refreshed in an on-demand fashion after the sink is waken up and sends out a control packet.

A sensor node alarms the sink if it stays in this state for a long time, i.e. forwarded many such reports but did not get a control packet.

An aggregation node (AN) generates a data report of the format the same as that in SEF [11].

$$\{Content, [k1, \dots, k5], BloomFilterVector[]\}$$

i.e. the report contains the content, a key list, and a Bloom filter generated from corresponding MACs.

Data report endorsement and authentication. While a group of authentication keys may be active at any time, each non-compromised node, including the AN node,

only keeps one at most. Therefore, an AN cannot verify all of its received MACs. To effectively defend attack A3, we determine multiple ANs and each AN selects a subset of its received MACs, reducing the possibility that the wrong MAC is selected.

We then study its effectiveness in more detail. Assuming there are c AN nodes, and each receives $m1$ MACs from which $m2$ MACs are selected. Assume one sensing node sends back a false MAC. Unless all c AN nodes select this wrong MAC, a correct packet can be constructed and delivered. Without considering other attacks, the probability of packet loss is

$$Prob_{packet\ loss} = \left[\frac{\binom{m1-1}{m2-1}}{\binom{m1}{m2}} \right]^c = \left[\frac{m2}{m1} \right]^c \quad (1)$$

For example, when $c=3$, $m1=10$, $m2=5$, 87.5% of true reports are delivered while without multiple ANs and random selection, only 7% are delivered (Fig. 5).

When these reports are forwarded along multiple paths, it is possible that a relay node receives multiple copies of the same report but with different Bloom filters. Suppose a relay node receives

$$\{C, [k1, k2, k3], BV1\} \quad \text{and} \quad \{C, [k3, k4, k5], BV2\},$$

i.e. the same content C but two different Bloom filters $BV1$ and $BV2$ using keys $k1, k2, k3$ and $k3, k4, k5$ respectively. If the second one is received before the node forwards the first one, they are merged into

$$\{C, [k1, k2, k3], BV1, [k3, k4, k5], BV2\}$$

If the second one is received after the first one has been forwarded, the relay node checks the packet in its 4-entry cache, if it is not there, the packet is sent normally; otherwise if found and with different filter vector, the relay node sends out the key list and the different filter vector. The content is omitted since the original scheme (section II) will not send it again. The packet is

$$\{MessageId, [k3, k4, k5], BV2\}.$$

If a relay node receives such a packet but couldn't find the corresponding message id in its cache, the packet is dropped immediately. We included these extra packets in the experiments.

If a relay node has $k3$, it can verify both $BV1$ and $BV2$; if it has $k4$, it can verify $BV2$ only. The report is not forwarded if all filter vectors are bad — the inconsistent MACs are dropped immediately while the content is saved in the cache. This is to defend false MAC attack as it may be combined with a later Bloom filter-only

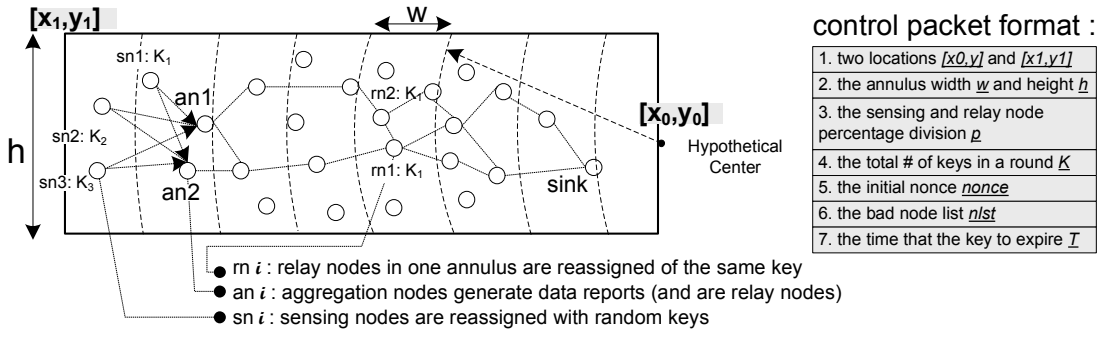


Fig. 6. The dynamic region based key reassignment algorithm.

packet. At that time, the saved content and MAC report can be re-combined and get forwarded. The content is dropped if it is kicked out the cache.

Dynamic authentication key reassignment. Our annulus-based dynamic authentication key reassignment scheme has following steps.

S1. *The sink broadcasts the key reassignment control packet before normal data transfers.* To ensure the security, the key reassignment can only be initiated by the sink. The packet contains the information as shown in Fig. 6 and itself is authenticated using μ TELSA [8].

The packet is designed to update authentication keys for sensor nodes within the rectangle region defined by $\{[x1,y1], [x0,y1+h]\}$. These nodes are divided into *sensing nodes* and *relay nodes* with keys updated differently. As shown next, the sink decides a percentage division p while a sensor u and its neighbors use the same pre-distributed $f_{random}()$ and u 's identity to determine u 's role in this round. A compromised sensor node cannot fake its role since its neighbors determine its role independently (more details are in step S2 and S3). On the other hand, the role of a sensor can change if a new control package is used.

The control packet defines a hypothetic center $[x0,y0]$ from which all sensor nodes can only roughly know where the sink resides. It is secure since the real location is hidden, which is important in a hostile environment.

In addition to sending the control packet, the sink generates and saves K new authentication keys using the predetermined polynomial $g(x)$ ($0 \leq i \leq K-1$)

$$NewKey_i = g(nonce + i).$$

That is, the *nonce* is used as a one time initial value for generating the new keys. A simple way to maintain the nonce is to initiate it to be zero and

update ($nonce += K$) every time a control packet is sent. The new keys are of indices from 0 to $K-1$. The control packet is broadcasted to all nodes in the region. Each node performs μ TELSA protocol [8] to verify that the control packet is authentic and has not been tampered with.

S2. *Reassigning the same key to relay nodes within an annulus.* The reassignment is always performed collaboratively between the node u (who is to update its key) and its neighboring nodes (who help the update). That is (here MAX is the maximal possible value that the random function can generate),

$$\begin{aligned} & \text{if } (u \notin nlst) \\ & \text{if } ((x1 \leq loc_{ux} \leq x0) \text{ and } (y1 \leq loc_{uy} \leq y1 + h)) \\ & \text{if } (f_{random}(nonce + u) / MAX \leq p) \\ & \quad dist = \sqrt{(loc_{ux} - x0)^2 + (loc_{uy} - y0)^2} \\ & \quad Ind = f_{random}([dist/aw]) \text{ mod } K \end{aligned}$$

If u is not an identified compromised node, both node u and its neighbors decide if u is in the region defined by $\{[x1,y1], [x0,y1+h]\}$. Then they decide if node u should be a relay node in this round. This is done by comparing the percentage parameter p and a randomly generated value using $(nonce + u)$ as the seed to the random function. The node is a relay node if the value is within $[0, p]$. Next the index of the new authentication key is decided. **Note:** Both u and its neighbors reach the same decision independently after initialization. That is, neighbors calculate u 's role such that u cannot fool them regarding if u should update the key, if u is a relay node, and the index of the new authentication key that u is to get.

By computing $[dist/aw]$ we effectively define an annulus such that all relay nodes in this annulus would have the same index as shown in Fig. 6.

Each of u 's neighboring nodes v_i ($0 \leq i \leq n-1$), if it receives an e_u share in the initialization, computes $e_u(nonce + Ind, v_i)$ and send it back to u . Regarding

each control packet received from the sink, only one such share is sent back to generate one key (the key with index Ind) in this round.

After collecting all $e_u(nonce + Ind, v_i)$ shares from the neighbors, node u reconstructs the polynomial $e_u(nonce + Ind, y)$ and compute its new authentication key as follows. Note only y is the variable for polynomial and thus only one key can be generated after constructing this polynomial.

$$XKey = g'(nonce + Ind) - e_u(nonce + Ind, u).$$

Since $g(nonce + Ind) = g'(nonce + Ind) - e_u(nonce + Ind, u)$, the newly updated key $XKey$ is the Ind -th key in the new group.

$$XKey = g(nonce + Ind) = NewKey_{Ind}$$

S3. *Random key reassignment for sensing nodes.* If a node is a sensing node, the new key index is assigned *randomly*.

if ($u \notin nlst$)
if ($((x1) \leq loc_{ux} < x0) \text{ and } (y1 \leq loc_{uy} \leq y1 + h))$
if ($f_{random}(nonce + u) / MAX > p$)
 $Ind = f_{random}(\lceil (loc_{ux} - x0) * (loc_{uy} - y0) \rceil) \bmod K$

Sensing nodes are with new authentication keys randomly without considering the annulus as above. This ensure the generation of sufficient different MACs for an event in the field. The update step is performed among neighboring nodes similar as above.

The mobile sink and sink authentication. Since the sink knows all the keys, the integrity of the packet can be verified after receiving the delivered report ([11], [15]). If a mobile sink moves out of the region defined in the control packet, it can send a new packet to refresh those who do not have new keys. If the same $nonce$ and K are used, the generated keys on these nodes will be consistent with those in the old region.

C. Defending various attacks

Each node gets only one key in each round. With the same polynomial $g(x)$ distributed in the network, a concern arises, that is, if every node can generate any new authentication key in a new round. This is impossible since each node only have $g'(x)$ and can collect shares to calculate $e_u(nonce + Ind, y)$. Note $nonce$ and Ind are constants so that the node u cannot regenerate $g(x)$ but only have the ability to compute one authentication key according to its role. It is discussed in step S2.

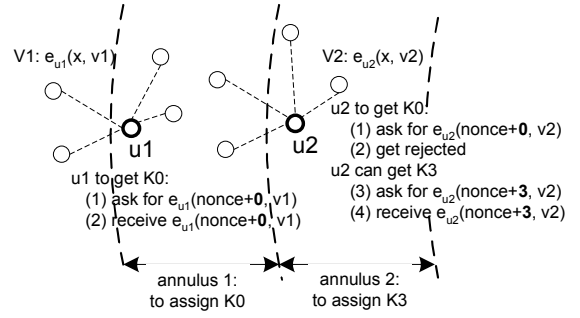


Fig. 7. Each node gets only one key in each round.

The security depends on the receiving of shares from its neighbors. Unless it can either break these neighbors or fool them, the node cannot generate an arbitrary key as it wishes. The former is hard since with sufficient distribution ($n=20$), it is hard to compromise all of them. The latter is hard since the control packet is sent from the sink and authenticated; the location of u is advertised in the trusted initialization phase. The neighbors make the decision by themselves. For example in Fig. 7, u_2 is in an annulus to get K_3 . If u_2 asks for shares for generating K_0 , its neighbors won't reply such requests. Actually only shares for generating K_3 can be collected by u_2 in this round. In addition, by compromising some but not all neighbors, an adversary cannot extract $g()$ either. The protocol is secure and only one key is received per node in each round.

Role switching attack. Since nodes are divided to sensing and relay nodes in each round, a concern may arise — if adversaries can benefit from switching the role of a compromised node u . As in step S2 and S3, after the initialization, u 's neighbors determine its role independently and send back shares for generating the corresponding key. u cannot even get shares for generating the key according to another role. On the other hand, a relay node may still possible to generate a sensing reading or even a report after collecting some MACs. While en-route nodes may not detect it, the sink has sufficient information ($nonce$, u 's id, and function $f()$) to check if the sender should be a sensing or an aggregation node in this round, and identify this attack immediately after receiving one such report.

Previous discussed attacks are defended. Since multi-path routing is used, dropping nodes from one compromised node (A1) does not terminate the delivery of true reports. With random MAC selection at multiple data aggregation nodes, a false MAC or a false Bloom filter (A3) may not block the generation and delivery of true reports unless all data aggregation nodes select

the false MAC. It is further defended by independent verification and dropping of Bloom filters at relay nodes. False data report attack (A2) is defended by region based key reassignment. We assign the width to be large than the detection range of a sensor node. Consider three consecutive annulus R1, R2 and R3, no matter which path a packet is to take, a node within R2 must be selected for packets sent from R1 to R3. This enforces the verification of corresponding MACs using the key in R2. The authentication strength accumulates from consecutive annulus that a packet travels to the sink.

D. Limitations

While our algorithm defends a broader range of attacks, it has some limitations. First the scheme has several threshold parameters similar as those in [11], [14]. The network loses the security protection if all group keys in one round are compromised. In addition each node distributes $e_u(x,y)$ to n neighbors. If all of them are compromised then the adversary can construct the $g(x)$ function and retrieve all old and new keys. They may be countered with more frequent key update together with schemes to isolate them [7]. We can also select a bivariate e-polynomial with larger degree and with redundant distribution among neighbors.

Second, we divide sensor nodes into sensing and relay nodes. This is not a problem for densely deployed network. However with limited number of nodes, sensing or routing functions may be affected if evenly partitioned. It is possible to enhance the control packet adaptively such that we allocate more nodes around the events for sensing while more nodes between the stimulus and the sink for routing.

Third an adversary may isolate a group of nodes from key updating. While he cannot extract the key, a victim node without the current key drops its received reports as the authentication will fail. There are two ways to alleviate such dropping. The sink can periodically fresh the region with the current control information and a node stops dropping packets or even participating in routing if it does not receive such packet for a while. Another way is to deploy some new nodes to replace isolated ones as in [14].

V. PERFORMANCE EVALUATION

A. Settings

We have implemented our algorithm by simulating 1000 sensor nodes deployed in a 1600×40 m² region. The detection range of each node is 20m. The aggregation nodes and the sink are set at two opposite ends

of the region with about 100 routing hops in between. There are 20 keys generated in each round. Each report is attached with 5 MACs. In our algorithm, we use 3 data aggregation nodes and each relay node forwards packets to three next hops nodes for multipath routing. Each aggregation node receives 10 MACs for each event. Similar to that in [11], it takes 16.25/12.5 μ J to transmit/receive a byte [11]. We ignore the computation energy cost since it is usually small comparing to routing energy consumption [11], [4].

B. False reports filtering and annulus width

We first study the effectiveness of our algorithm in defending injected false reports by varying the annulus width. Since relay nodes in one annulus are assigned with the same authentication key, the annulus width is an important parameter to decide. Intuitively, if the width is smaller than the detection range of a node, some annulus may be skipped and thus we lose the desired authentication from nodes in those annulus.

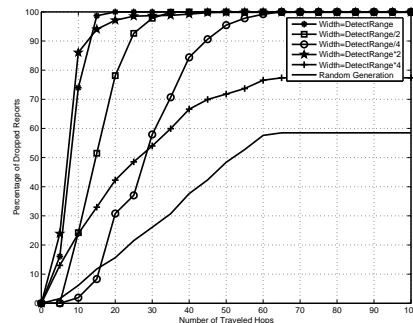


Fig. 8. Early detection of false data reports.

Annulus width	# of hops
1/4 Detection Range	27.8
1/2 Detection Range	14.2
Detection Range	7.2
2 \times Detection Range	6.8
4 \times Detection Range	20.6
Random (i.e. SEF)	30.2

Fig. 9. Average number of hops that false data reports travel.

The results are shown in Fig. 8. We assume one authentication key is compromised. The random case indicates that keys are reassigned randomly without using our annulus based algorithm, i.e. the SEF approach. From the figure, random key assignment cannot effectively filter false reports in multipath routing — around 41% false packets are delivered. With annulus based key assignment, we drop 100% false packets in most cases. If the width is too large (four times the detection range),

there are not enough number of annulus on the routing path and thus lack the full coverage of all authentication keys. False packets with missing keys may be delivered. The best filtering results are achieved when the annulus width ranges from one to two times of the detection range. Almost all false reports are dropped within ten hops with averages 7.2 hops and 6.8 hops respectively. When the width shrinks, the annulus skipping effect (as described above) happens and thus false report can travel more hops before completely dropped. For example false reports travel on average 27.8 hops when the width is 1/4 of the detection range (Fig. 9). The results also suggest that small location errors can be tolerated if annulus width is assigned to be two times of the detection range.

C. True reports delivery

Next we study the delivery of true reports under the false MAC injection attack. As we discussed we defend it using multiple data aggregation nodes and random MAC selection. We assume each aggregation data receives ten difference MACs of which some are false MACs.

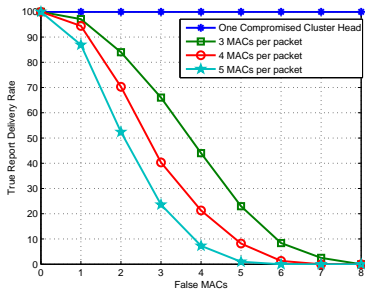


Fig. 10. The impact of attached MACs.

Fig. 10 plots the percentage of received reports when different number of sensing nodes send false MACs. When there is one such node, we receive 97% and 88% of the reports while without our algorithm, only 7% can be received (Fig. 5). As more sensing nodes are compromised, more true reports are dropped accordingly. It is more resilient if each packet contains fewer number of MACs, which reduces the probability that the wrong MAC is selected.

If two different MACs based on the same key are received by an aggregation node, then at least one is a false MAC. In this case both MACs are included in a separate report to the sink for the identification of the compromised node, which terminates its future attack. On the other hand, a compromised data aggregation node can arbitrarily tamper with its received MACs. However it cannot prevent other aggregation nodes from receiving correct MACs. Since Bloom filters are verified

independently. As a result 100 % of reports are received while those injected false MACs are dropped in the network Fig. 10.

D. Multipath routing and report delivery

Since multipath routing is designed to tolerate various types of failures in the network, we then evaluate the report delivery under two types similar to the ones in [12]: (i) each node has a uniform probability to fail (Fig. 11); (ii) each link has a uniform probability to fail (Fig. 12). From the figures our algorithm achieves consistent report delivery in different fault models. For example, for failure rates smaller than 20%, false reports, true reports with correct MACs, and true reports with one wrong MAC are dropped or received with same rates as that there is no fault.

We then evaluate the malicious report dropping at compromised nodes in the network. These nodes drop some or all its received packets. We evaluate with the failure model that has 20% uniform link failure probability. From Fig. 13 we see that these compromised nodes have little impact on report delivery regarding all three types of data reports we evaluate in the paper.

E. Authentication key update overhead

Till now the results are compared without considering network initialization and key re-assignment overhead. Here we discuss these extra costs which the original SEF does not have.

To update authentication keys, each sensor node receives a broadcasted control packet from the sink and n ($=20$) $e_u(x, y)$ shares from its neighbors — each share is 8 bytes [14] while the control packet is of 24 bytes. Thus there are 184 bytes transferred and received for updating the key on one node. In our experiments we employ passive key update which, instead of updating the key right after receiving the control packet, a node asks for the shares only if it is involved either as a sensing node or a relay node. A 2-byte short request packet is sent to neighbors for updating.

The network initialization overhead is a one time cost. Each node broadcasts shares to its neighbors. This cost is about the same as that to update keys of all nodes in one round.

The sum of these costs is approximately the same as the cost to route 10 extra data reports. It is amortized by continuous report transfers in one round.

F. Overall evaluation

The last experiment is to study the effectiveness of our algorithm in a hostile environment with all three types of

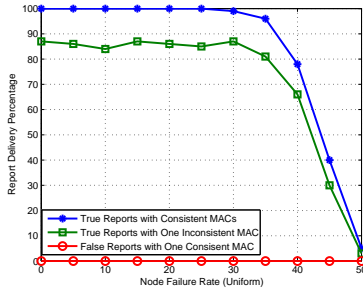


Fig. 11. Report delivery with uniform node failure.

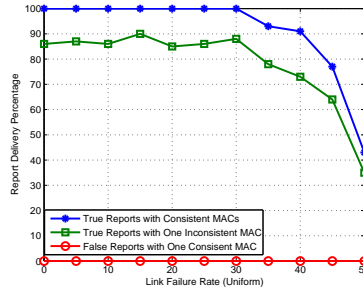


Fig. 12. Report Delivery with uniform link failure.

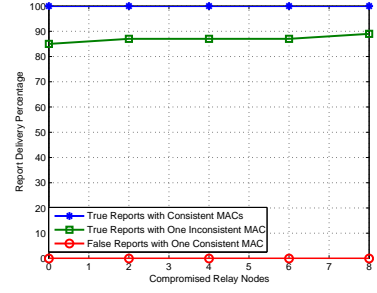


Fig. 13. The impact of malicious report dropping.

attacks A1, A2 and A3 (Fig. 14). We consider all costs of our scheme including the initialization and key update costs.

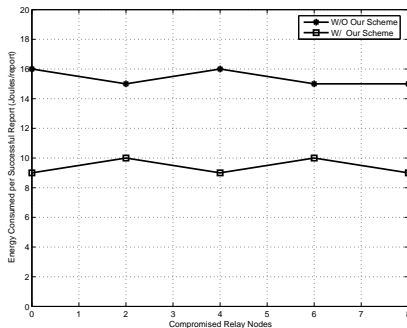


Fig. 14. Energy consumption per accepted true report.

We assume only one authentication key compromised. There are one compromised sensing node, one compromised data aggregation node, and some compromised relay nodes. The sensing node always sends out the reading with a wrong MAC while the compromised data aggregation node injects false reports with wrong data content and one consistent MAC using the compromised key. The compromised relay nodes keep dropping packets they receive. The compromised relay nodes drop their received reports. The number of false reports is $2 \times$ the number of true packets while the number of reports with a wrong MAC is half of the true reports. We route a total of 1000 reports.

Fig. 14 shows that our algorithm achieves better energy consumption per true report accepted at the sink node. Comparing with the one without using multiple data aggregation nodes and region based key assignment scheme, we save on average 33% of the energy for each successful routing.

VI. CONCLUSIONS

In this paper we proposed the dynamic region based authentication key reassignment algorithm for en-route

report authentication in multipath routing environment. Experimental results show that it defends a broader range of data manipulation attacks, and achieves better trade-offs for the delivery of more true reports, early dropping of false reports, and low energy consumption per delivered true report.

ACKNOWLEDGEMENT

This work is partially supported by the U.S. National Science Foundation under grants CCF CAREER 0447934 and CCF 0430021.

REFERENCES

- [1] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, Energy-efficient Multipath Routing in Wireless Sensor Networks," In *ACM Mobile Computing and Communications Review*, Vol.5(4), 2001.
- [2] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: a Scalable and Robust Communication in Wireless Sensor Networks," In *MOBICOM*, 1999.
- [3] S. De, C. Qiao, and H. Wu, "Meshed Multipath Routing with Selective Forwarding: an efficient Strategy in Wireless Sensor Networks," In *Computer Networks*, 43(2003) pages 481-497, Elsevier 2003.
- [4] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," In *IEEE international workshop on Sensor Network Protocols and Applications*, pages 113-127, 2003.
- [5] D. Liu, P. Ning, and R. Li, "Establishing Pairwise Keys in Distributed Sensor Networks," In *ACM Trans. Info. Sys. Security*, 8(1): 41-77 (2005).
- [6] D. Liu, P. Ning, and W. Du, "Attack-Resistant Location Estimation in Sensor Networks," In *IPSN* 2005.
- [7] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," In *MOBICOM*, 2000.
- [8] A. Perrig, R. Szewczyk, V. Wen, D.E. Culler, and J.D. Tygar, "SPINS: security protocols for sensor networks," In *MOBICOM*, 2001.
- [9] A. Srinivas, and E. Modiano, "Minimum Energy Disjoint Path Routing in Wireless Ad-hoc Networks," In *MOBICOM*, 2003.
- [10] H. Yang, F. Ye, Y. Yuan, S. Lu and W. Arbaugh, "Toward Resilient Security in Wireless Sensor Networks," In *ACM MOBIHOC'05*, 2005.
- [11] F. Ye, H. Luo, S. Lu and L. Zhang, "Statistical En-route Detection and Filtering of Injected False Data in Sensor Networks," In *IEEE INFOCOM*, 2004.
- [12] F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient broadcast: A robust data delivery protocol for large scale sensor networks," In *ACM WINET*, vol. 11(2), 2005.
- [13] O. Younis, and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach," In *INFOCOM*, 2004.
- [14] W. Zhang and G. Cao, "Group Rekeying for Filtering False Data in Sensor Networks: A Predistribution and Local Collaboration-Based Approach," In *INFOCOM*, 2005.
- [15] S. Zhu, S. Setia, S. Jajodia, P. Ning, "An Interleaved Hop-by-Hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks," In *IEEE Security and Privacy*, California, 2004.