# Security and Flexibility Contribution for Wireless Sensor Networks

**Nourreddine Mitta, Rachid Elgouri, Lamari Hlou**

Laboratory of Electrical Engineering and Energy Systems, Faculty of Sciences, Ibntofail University-Kenitra Morocco

**Summary**
We propose a flexible and secure routing algorithm for wireless sensor networks. It guarantees confidentiality, authenticity and integrity of messages transporting data. These properties were experienced with CryptoVerif. Our experimental results show that the flexibility of our algorithm against several attack scenarios is better than many other routing protocols, especially in sparse networks. In addition, our algorithm adapts to face attackers whose behavior changes over time.

***Key words:***
Routing, Sensor Networks, Security, Flexibility.

## 1. Introduction

Wireless sensor networks are wireless mesh networks, consisting of multiple sensors. These sensors operate using batteries. They are limited in memory and computing capacity. Finally, they communicate by radio. Routing is a central problem in these networks. We consider here that the routing converges: i.e. there is a particular node in the network, called pit, and data from other nodes, called sources, are intended to wells. Network characteristics of wireless sensor systems are prone to attacks. We consider a critical scenario where an intruder has compromised multiple nodes. The intruder full control nodes compromise, in particular, it has both access to internal data of these nodes (e.g., cryptographic keys) and the messages they have received. The attacker can then disrupt the routing on two levels. It can address the data, e.g., counterfeit messages deliver erroneous information to the application. It can also attack the same routing, e.g., it may lose messages or create to degrade the quality of network service.

We propose a converge routing algorithm SFRR (Secure Flexible Reputation Routing), which fights against both attacks at the packet level and routing level. SFRR uses cryptographic primitives adapted to wireless sensor networks [1] - symmetric cryptography, nonces (fresh and unpredictable random values) and hash functions - to provide several security properties: confidentiality routed data and the inability to forge the messages carrying (this property implies the authenticity and integrity of messages). Then SFRR resists attacks routing level by ensuring a high level of flexibility. This concept has been defined as the ability of a network to continue to provide a reasonable quality of service when it is under attack [2]. In our case, flexibility is measured by the overall message delivery and the equity between different rates of honest nodes. We compare experimentally SFRR against several protocols: standard protocols such as uniform random walk (RW)[3], geographic routing (GFG) [4] and gradient routing (GBR) [5-6] and protocols aimed flexibility. For the latter, we consider the three solutions (RGBR, PRGBR and PRDGBR) proposed in [2]. These algorithms are variations of GBR protocol that route messages according to a spanning tree rooted in the well width. These three variants are to introduce randomness and duplications in GBR. Our study shows that the flexibility of SFRR is better than these protocols. Moreover, unlike these algorithms SFRR fits against the attackers whose behavior changes over time.

## 2. Presentation of SFRR

We consider related bidirectional networks, where all honest nodes regularly have to route data to wells. Each node has a unique identifier, a shared key with the well, and can use symmetric cryptography, hash functions and nonce. The nodes also know their neighbors. The code of our algorithm, as well as the details of our model and our evidence is available on ask.

### 2.1 Principle

Introduce randomness in a routing algorithm is interesting for its flexibility, as it is the unpredictable routing by the attacker. So, SFRR is designed as a reinforced random walk, based on a reputation mechanism to calculate the probability of choosing a neighbor to the next hop. The idea is to increase the probability to route a message through a neighbor if it performs well.

To do this, SFRR is based on acknowledgments. Our algorithm ensures that for each delivery valid received; the corresponding message is best delivered to the wells. In this way, the node can legitimately increase their confidence in the neighbor to whom it sent the message first. After a time, all nodes routed preferably their messages via their trusted neighbors. Thus, messages tend to follow routes which surely lead to the well.

Manuscript received August 5, 2013
Manuscript revised August 20, 2013

SFRR is based on nonces. Each message is routed SFRR identified by a nonce $N_v$, which is generated by the initiator node *v*. Node *v* figure this nonce with the data to send, using the key it shares with the well $k_{vs}$. Encryption C and the identifier of node v are then routed to the wells. Once validated message, the well produces acknowledgment $<N_v;v>$. It is returned to the original sender *v*. Thus, the receipt of the acknowledgment, *v* is certain that the well has received the message.

However, an attacker can modify the blind content of a received packet. To avoid the well that pourd erroneous application information, we add to the message digest *H* of a nonce $N_v$, created with a public hash function. Upon receipt of a message, decrypts the well with the *C* key of the sender claimed for $N_v$, and applies the hash function. The result should be equal to *H* for the message to be considered valid. In this way, if an attacker alters the encryption *C* or the identity of the sender of a message, the well detects these changes and rejects the message.

## 2.2 Reputation of mechanism

Upon receipt of an acknowledgment, the node *v* that initiated the routing of the message corresponding *m* can conclude that *m* is delivered. In this case, *v* increases the probability associated with the neighbor in which *m* is sent first. To do this, the first consignment of *m*, *v m* stores the nonce and the identifier of the first recipient in $L_{Queue}$ list. Upon receipt of an acknowledgment, *v* checks if the recipient of the accused, and if the nonce is contained in $L_{Queue}$. In this case, *v* retrieves the identifier of the corresponding neighbor growing reputation and removes the entry from $L_{Queue}$. If *v* is the recipient of the acknowledgment, but the list does not contain the corresponding input, the accused is simply ignored.

Due to memory limitations, the size of $L_{Queue}$ merely $s_Q$ elements. When a node needs to route a message, but the list is full, the oldest element is deleted. In this way the most recent information is a priority. We also note that the lost messages, or that the accused was lost ultimately removed from the list.

Finally, it is possible that a data message *m* returns the node *v* that generated because of a cycle in the network. In this case, the validity of the message is verified, then the routing m is reset and the oldest entry in $L_{Queue}$ is replaced by the new.

## 2.3 Assessment of reputation

To select the next node which will send a message; a node performs a random choice among its neighbors, weighted by their reputation. The reputation of a neighbor is the number of occurrences of the identifier in the list $L_{Routing}$: each receive a receipt confirming delivery of a message, the initiator node increases its confidence in the neighbor chosen as the first relay by adding their username in this list.

The choice of the next hop follows the law of the next probability: for a node *v*, either X is the random variable representing the neighbor of *v* that sends the message.

Let $\delta_v$ the number of neighbors of *v*,

$|L_{Routing}|_x$ the total occurrences number of a node *x* in $L_{Routing}$

$|L_{Routing}|$ the sum number of elements in this list.

The probability of forwarding a message to x is:

$$\Pr(X = x) = (|L_{Routing}|x + \delta^{-1}{}_v)/|L_{Routing}| + 1 \qquad (1)$$

Intuitively, if a node must route a message, it will select a random value from $L_{Routing}$ or a joker. If a node ID is pulled, the message will be transmitted. In the case of joker, a neighbor will be chosen uniformly at random.

Thus, the most trusted node to a neighbor, the more it will send him messages. However, there is always a positive probability of choosing a node without considering the reputations.

To ensure a strong flexibility to attackers who change their behavior over time, $L_{Routing}$ is a FIFO list size bounded by $s_R$. Thus, the information stored will always be cooler. In this way, if an attacker performs well at first, its reputation will be good. If then, he decided to lose messages, reputation fall gradually through acknowledgments through other paths messages.

## 2.4 Routing acknowledgments

A receipt is issued only if the corresponding message was delivered by *m* pit or well. We can therefore assume that the path followed by *m* is sure. As our relationships are bidirectional, we route the acknowledgment as possible, following the reverse path *m*.

To do this, each data message leaves a trace of its passage through all the nodes that relay. This trace is stored at each node in the list $L_{AckRouting}$: after each reception of a message, the relay nodes store the footprint of the message nonce and the identifier of the neighbor who relayed above. This information will then be reused for routing acknowledgments: when a node v receives an acknowledgment, it calculates the footprint of the announcement of this acknowledgment, and searches $L_{AckRouting}$ if the corresponding path is known. If this is the case, the acknowledgment is relayed to the associated node. Otherwise, it is referred to a neighbor chosen uniformly at random.

If data messages loop and revisits a node, the most relevant information on its provenance is the oldest. Therefore, when additions in $L_{AckRouting}$, the node must always check if they already have information about this message: If this is the case, the old entry is retained.

Acknowledgments, however, can be lost through compromised nodes. To avoid cluttering memory nodes with routing information for these defendants is $L_{AckRouting}$ maximum size $s_A$, and additions to a full list remove the oldest items.

Finally, an attacker can forge a receipt with a false destination. These false acknowledgments can be relayed indefinitely. Since a node cannot trust its neighbors, we cannot rely on them whether an accused should be relayed or stopped. To work around this problem, each node chooses not to transmit an acknowledgment with a probability *1 / N*, where *N* is an upper bound on the number of nodes. Thus, an acknowledgment will be in *N* jumps before being deleted. An advantage of this mechanism is that it favors shorter routes. Indeed, acknowledgments of receipt of long roads are often removed before reaching their destination. However, the length of the route followed by an accused is correlated to the length of the route followed by the message corresponding. Thus, the reputation of neighbors participating in long roads will seldom strengthen.

### 2.5 Security

We have analyzed the security properties of SFRR with CryptoVerif [7]. From games created by protocol, and the desired properties of cryptographic primitives used CryptoVerif determines a bound on the ability of an attacker to break these properties. This bound depends on the sizes of parameters and properties of cryptographic primitives that are used. We proved three properties SFRR: first, the protocol guarantees the confidentiality of data routed.

Second, it is impossible to counterfeit data messages (this property implies authenticity and integrity of these messages). Third, it is possible to create a receipt for a message that has not yet been delivered to the wells, or pits.

## 3. Experimental Analysis Result

### 3.1 Methodology and parameters

We have experimentally evaluated the flexibility against SFRR, RW, GFG, GBR and its variants (RGBR, PRGBR, PRDGBR) in various attack scenarios. We used Sinalgo [8] a simulator for wireless sensor networks. Here we present some representative results.

For our simulations, we consider networks related unitary disk generated by placing nodes uniformly at random on a square surface. The compromised nodes are randomly chosen from the sensors, and the well is located in the center of the simulation area. Communications are asynchronous and FIFO transfer times follow an exponential distribution, as well as the interval between two generations of message for a node. Our simulations had time to process 500,000 messages, and we test 20 topologies for each experiment. SFRR requires four parameters: N (an upper bound on the number of nodes in the network), and bounds on the size of each list ($s_R$, $s_Q$ and $s_A$). In each simulation, *N* is set to the number of nodes in the network. We set respectively $s_R$, $s_Q$ $s_A$ and 10, 3 and 5 elements. These values were determined by a detailed experimental evaluation in the technical report [1], and guarantee good performance while keeping low memory consumption.

### 3.2 Results

Figure 1 shows the rate of delivery of messages observed in networks of average degree

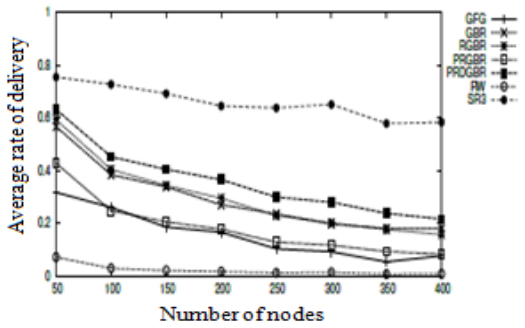


Figure 1.Average of delivery, 30% BH, δ = 8

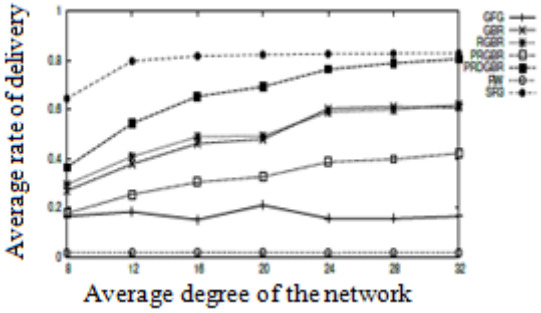


Figure 2. Average of delivery, 30% BH, *n* = 200

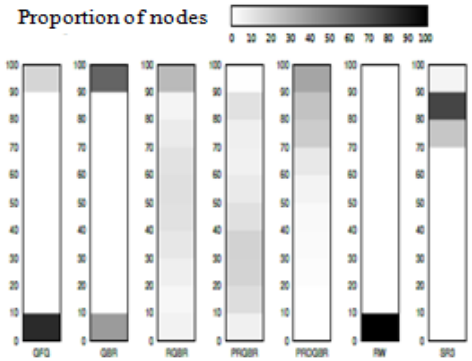


Figure 3. Rates distribution of delivery, 30% BH, *n*=200, δ=32

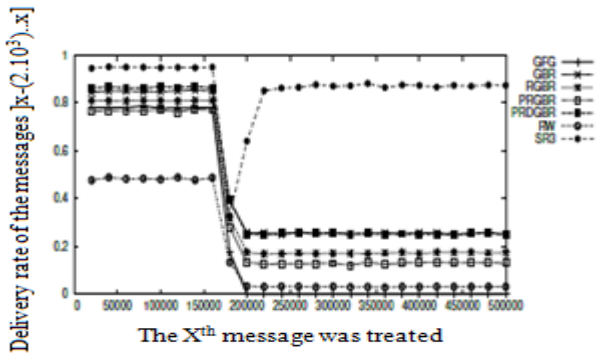


Figure 4. Average rate of delivery per packages of 20000 messages, 5% WH→BH, 5% BH, n = 200, δ = 8

δ = 8, confronted with 30% of blackholes (BH), i.e., nodes compromise lose all the messages they receive. By varying the number of nodes in the network, we see that SFRR always gives a delivery rate superior to other protocols in our panel. The gap is widening in large networks. Figure 2 shows the rate of delivery of messages observed in networks of 200 nodes, confronted with 30% of blackholes, varying δ from 8 to 32 Again, SFRR offers the best performance. Moreover, as RW and GFG, we note that SFRR is insensitive to variations in δ. Instead, the rates observed for GBR and its variants are low in low-density networks.

In very dense networks, the performance of PRDGBR approximates SFRR. However, PRDGBR duplicate messages at each hop, which leads to a significant communication overhead.

We then measured the fairness of SFRR, i.e., how honest nodes have comparable delivery rates. Figure 3 shows the distribution of the delivery rate of nodes in networks of 200 nodes and average degree 32. GBR is not fair, because clearly distinguishes between two classes of nodes: those in which all messages are delivered and all those messages are lost. Conversely, SFRR is fair, since almost all nodes have a delivery rate close to 80%. We then evaluated the adaptability of SFRR. Figure 4 shows the average sliding window of 20,000 messages delivery rate in a network of 200 nodes and average degree 8, against 5% of blackholes and 5% of wormholes, which initially act as nodes connected directly to well (generating an excellent reputation), then as blackholes after the first third of the simulation.

## 4. Conclusion

Our protocol actually gets a good delivery rate quickly after the change in behavior, which is not the case for other protocols.

Finally, we found that the attackers blocking a portion of the message (selective forwarding) or declaring multiple identities (Sybil nodes) do not have special impact on our protocol compared to other presented here. We also evaluated the performance of wireless networks in SFRR attackers, and our results indicate routes reasonable length: for example, in networks of 400 nodes and average degree 8, the roads have an average length of 20 jumps.

## References

[1] T. Eisenbarth and S. Kumar. A survey of lightweight-cryptography implementations. Design & Test of Computers, IEEE, 24(6) :522–533, 2007.

[2] O. Erdene-Ochir, A. Kountouris, M. Minier, and F. Valois. Enhancing resiliency against routing layer attacks in wireless sensor networks Gradient-based routing in focus. International Journal On Advances in Networks and Services, 4(1, 2) :38–54, 2011.

[3] Izumi Kubo Satoshi Ikeda and Masafumi Yamashita. The hitting and cover times of random walks on finite graphs using local degree information. Theoretical Computer Science, 410 :94–100, 2009.

[4] Kuruvila, Nayak & Stojmenovic. Progress based localized power and cost aware routing algorithms for ad hoc and sensor wireless networks. International Journal of Distributed Sensor Networks, vol. 2, no. 2, pages 147–159, 2006.

[5] J. Faruque and A. Helmy, "Gradient-based routing in sensor networks," ACM SIGMOBILE Mobile Computing and Communications Review, Special feature on MOBICOM 2003 posters, vol. 4, no. 4, pp. 50–52, October 2003.

[6] K.-H. Han, Y.-B. Ko, and J.-H. Kim, "A novel gradient approach for efficient data dissemination in wireless sensor networks," in IEEE International Conference on Vehicular Technology Conference (VTC), 2004.

[7] B. Blanchet. A computationally sound mechanized prover for security protocols. IEEE Trans. Dependable Sec. Comput., 5(4) :193–207, 2008.

[8] Sinalgo : Distributed Computing Group, Laboratory TIK, ETH Zurich, 2013, http://disco.ethz.ch/projects/sinalgo/