Secure Aggregation in Hybrid Mesh/Sensor Networks

Roberto Riggio
CREATE-NET
Via Alla Cascata, 56/c
38123 Trento, Italy
Email: roberto.riggio@create-net.org

Sabrina Sicari
University of Insubria
Dipartimento di Informatica e Comunicazione
Via Mazzini, 5
21100 Varese, Italy
Email: sabrina.sicari@uninsubria.it

***Abstract*—Several researchers are proposing information systems–based Wireless Sensor Networks (WSNs) that provide an extensible and effective means to monitor large and diverse geographical areas. Nodes in a WSN are characterized by very limited computing capabilities and energy consumption is a major concern, which implies that communications should be minimized, thus unorthodox solutions are required for many situations. The definition of secure and privacy aware solutions, ensuring at the same time limited power consumption of transmitted data is then a great challenge. In this paper we present an hybrid mesh/sensor network, which allows to deliver a transparent multi–hop wireless backhaul able to handle in a secure way different kinds of data (temperature, humidity, etc.), coming from different kinds of wireless sensor networks. The main idea is based on a sharing of tasks between wireless mesh networks and wireless sensor networks. Our architecture is particularly suitable to realize an application agnostic mesh backhaul able to concurrently support multiple WSNs, while ensuring both end–to–end encryption *and* hop–by–hop authentication. Hence, in order to evaluate the performance of the proposed architecture an ad-hoc prototype is realized.**

***Index Terms*—wireless networks, IEEE 802.11, sensors networks, mesh architecture, secure aggregation, testbed**

I. Introduction

Wireless Sensor Networks (WSNs) have been object of a deep study and analysis from scientific community [1]. In the last few decades many solutions have been provided and the research on some issues, i.e. routing protocol, localization algorithm, has reached a mature step. The research on WSN has been characterized by the power limits of the sensor nodes. In fact, sensor nodes are tiny devices with limited resources in terms of storage, power and processing. It is the main reason for which it is necessary to reduce the amount of transmitted data. In order to achieve such a goal many data aggregation algorithm have been defined [2], [3], [4].

The range of application of WSNs is wide and spreads over multimedia surveillance systems, traffic monitoring and control in urban/suburban areas, support to military and/or anti-terrorism operations, telemedicine, assistance to disabled and/or elderly people, environmental monitoring, secure localization of services and users, industrial process control. In order to ensure a broad deployment of such innovative services, strict requirements on security and privacy should be satisfied, taking also into account the limited technological resources (in term of energy, computation, bandwidth, and storage) of sensor nodes. In fact, data aggregation is potentially vulnerable to attackers who may inject bogus information or forge aggregated values without being detected. At the aggregation layer many security services can be provided, but well defined security solution is expensive in terms of power consumption. This is the main reason for which the research on security and privacy in WSNs is still under investigation. Security solutions require power, WSNs have limited power resources, so WSNs should not be secure. In order to overcome such a limit we propose a hybrid network architecture combining WSNs with the wireless mesh networking paradigm. A Wireless mesh networks (WMNs) [5] consist of several nodes (mesh routers and mesh clients), which exploit multi-hopping in order to build and maintain a wireless backhaul.

The hybrid mesh/sensor network, proposed in this paper, aims at delivering a transparent multi–hop wireless backhaul, able to handle different kinds of data (temperature, humidity, etc.) coming from a WSN. More specifically, in such a context sensor nodes use their resources (i.e. their power) just for sensing and encrypting, while mesh routers implements the secure aggregation of the encrypted data and relay the aggregated data to the network *Sink*. Our architecture is particularly suitable to realize an application agnostic mesh backhaul able to concurrently support multiple WSNs application while ensuring both end–to–end encryption *and* hop–by–hop authentication. Finally, the proposed architecture addresses power consumption issues in large WSNs (i.e. a metropolitan area) in that the encrypted data can is relayed by the WSN to the closest mesh router where it is first aggregated with other samples and then delivered over the mesh backhaul to the appropriate network *Sink*. It is worth noting that albeit mesh router are typically directly connected to the electrical grid, their (relatively) small physical footprint makes them suitable for a wide range of deployments, as for example mesh routers can be deployed as completely autonomous units with solar, wind, or hydro power. To the best of the authors' knowledge there are no other works that exploit an hybrid WSN/WMN architecture to jointly address e security and power consumption issues.

The paper is organized as follow: Sec. II provides a brief overview about the security model; Sec. III introduces the

9781-4244-3941-6/09/$25.00

reference scenario; Sec. IV provides technical details about the prototype. Sec. V compares the obtained results with solutions without aggregation; Sec. VI analyzes the state of the art. Finally, Sec. VII draws some conclusions and provides hints for future works.

II. Security Model

Secure aggregation becomes especially challenging if end-to-end privacy between sensors and the *Sink* is required. In literature there are several works defined in order to guarantee security of the aggregated data. More specifically, the main contribution are cataloged into hop-by-hop [3], [6], [7] and end-to-end [2], [8] secure aggregation. The aim of our work is to provide a solution that guarantees security to data that are aggregated by mesh nodes. In our scenario, in fact a mesh network has the function to aggregate data coming from a WSN, before reaching the *Sink*. In order to achieve our goal we choose the algorithm of Castelluccia et al. [2] because it is based on a simple and secure additively homomorphic stream cipher that allows efficient aggregation of encrypted data. The new cipher only uses modular additions and is therefore very well suited for CPU-constrained devices. Aggregation based on this cipher can be used to efficiently compute statistical values such as mean, variance and standard deviation of sensed data, enabling significant bandwidth gain.

An homomorphic encryption scheme allows arithmetic operations to be performed on ciphertexts. One example is a multiplicatively homomorphic scheme, whereby the multiplication of two ciphertexts followed by a decryption operation yields the same result as, say, the multiplication of the two corresponding plaintext values. Homomorphic encryption schemes are especially useful in scenarios where someone who does not have decryption keys needs to perform arithmetic operations on a set of ciphertexts. The main idea of [2], is to replace the XOR (Exclusive-OR) operation, typically found in stream ciphers, with modular addition.

For readers' convenience, the homomorphic encryption scheme proposed in [2] is here briefly sketched. Each sensor represents its message m_i as an integer $m_i \in [0; M-1]$, where M is a large integer. Let k_i be a randomly generated keystream, where $k \in [0; M-1]$, the encrypted ciphertext c_i is given by:

$$c_i = Enc(m_i; k_i; M) = m_i + k_i (mod M) \tag{1}$$

The sensor then forwards the ciphertext c_i to its parent, who aggregates all the c_i received from its children by simply:

$$c = \sum_{i=1}^{k} c_i (mod M) \tag{2}$$

The cleartext message can then be obtained by:

$$s = Dec(c, k, M) = c - k (mod M); \quad k = \sum_{i=1}^{k} k_i \tag{3}$$

Where *Enc()* and *Dec()* respectively denote the encryption and decryption scheme; M is the message space and C the ciphertext space such that M is a group under operation $\oplus$ and C is a group under operation $\otimes$. In other words, the result of the application of function $\oplus$ on plaintext values may be obtained by decrypting the result of $\otimes$ applied to the corresponding encrypted values.

Besides, m assumes value in the range $0 \leq m \leq M$. Due to the commutative property of addition, the above scheme is additively homomorphic. In fact, if $c_1 = Enc(m_1; k_1; M)$ and $c_2 = Enc(m_2; k_2; M)$ then $c_1 + c_2 = Enc(m_1 + m_2; k_1 + k_2; M)$.

Note that if n different ciphers c_i are added, then M must be larger than $\sum m_i$, otherwise correctness is not provided. In fact if $\sum m_i$ is larger than M, decryption will results in a value m' that is smaller than M. In practice, if $p = max(m_i)$ then M should be selected as $(M = 2^{\log(p*n)})$. The keystream k can be generated by using a streamcipher, such as RC4, keyed with a node's secret key and a unique message id. Finally, each sensor node shares a unique secret key with the *Sink*. Such keys are derived from a master secret (known only to the *Sink*) and distributed to the sensor nodes. However, the key distribution protocol is outside the scope of this work.

III. Network Architecture

The reference network model is sketched in Fig. 1. As it can been seen from the figure, clusters of sensor nodes exploit a multi-hop wireless backhaul in order to deliver the sensed data to a network *Sink*. Each cluster is composed by a variable number of sensor nodes and one mesh router, which acts as *Cluster Head*. *Cluster Heads* are in charge for both gathering encrypted messages coming from local sensor nodes and implementing the secure aggregation scheme by combining local messages with aggregated messages coming from other *Cluster Heads*. Sensor nodes within a cluster may as well exploit multi–hopping in order to reach their *Cluster Head*. The proposed aggregation scheme requires that all sensors in a cluster send their data within the same sampling period. Such a goal can be achieved either by having synchronized sensor nodes, or by implementing a polling scheme at the *Cluster Head* level. Our architecture implements the latter solution.

Sensor nodes are not required to reply to all requests. As a matter of fact, nodes in a WSN can be unavailable for a number of reason ranging from temporary lack of connectivity, limited battery, or simply hardware failures or malicious removal. However, in order to properly obtain the cleartext from an aggregated message, the network *Sink* needs to know the *ids* of the non-responding sensor nodes. In order to address this issue we introduced a message, named *Aggregated Message* (AMEX), generated by the *Cluster heads* and containing a list of the non–responding nodes in a cluster. Such a list can be easily computed by the *Cluster head* using the message received from the sensor nodes and the list of sensor nodes in its cluster (obtained using an initial raging procedure).

It is worth stressing that, encryption is performed only by sensor nodes. *Cluster heads*, on the other hand, perform only

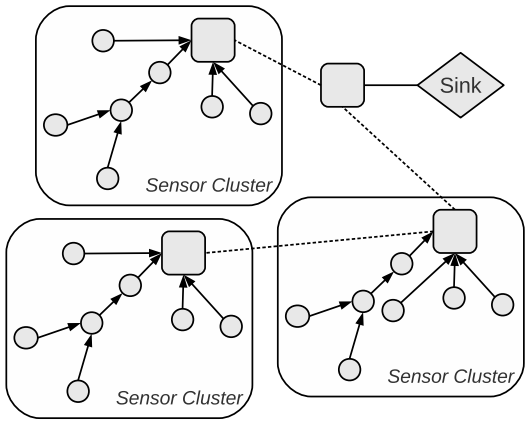

Fig. 1. Reference network model for the hybrid mesh/sensor secure aggregation scheme.

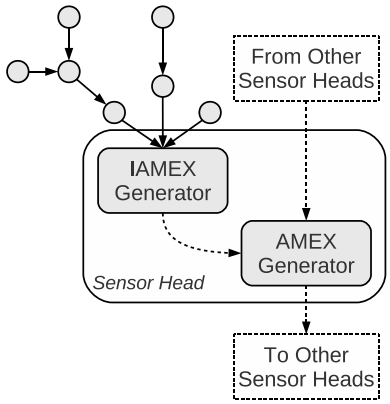

Fig. 2. Architecture of the *Cluster head*. IAMEX and AMEX data paths are represented respectively by continuous and dashed lines.

the addition of ciphertexts, while the ciphertext is decrypted only at the *Sink*. As a result, *Cluster heads* are only required to know the *ids* of the sensors in their cluster making our architecture particularly suitable to realize an application agnostic mesh backhaul, able to support multiple WSNs while ensuring both end–to–end encryption *and* hop–by–hop authentication.

In–cluster aggregation is also supported. In this case sensor nodes are in charge of both message forwarding and addition of ciphertexts. In this scenario, each sensor appends its node *id* to the relayed message creating an *In-Cluster Aggregated Message* (IAMEX). Fig. 2 sketches the architecture of the *Cluster Head* in its most general deployment scenario. Continuous lines represent communication paths that use the IAMEX format, while dashed lines represent communication paths that use the AMEX format. It is worth stressing that, thanks to the homomorphic additive encryption scheme, messages of the same type can be aggregated in a end–to–end fashion by simply adding their ciphertexts and appending the nodes' *ids*. Please note that the evaluation of the in–cluster aggregation is out of the scope of this work.

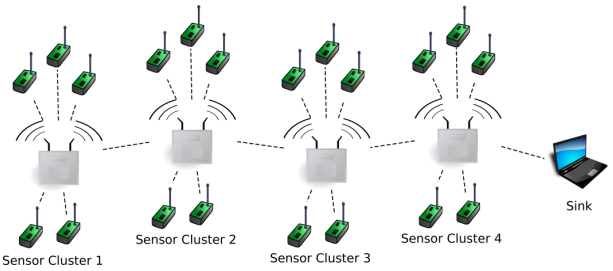

Fig. 3. The linear network topology exploited during our study. Sensor nodes are emulated by a software process running within each *Cluster Head*.

IV. Technical details

The mesh backhaul has been implemented using the WING toolkit[1]. WING [9], [10] is an experimental IEEE 802.11 wireless mesh network built on top of the Roofnet platform [11] originally developed by MIT in Cambridge, MA, USA. Roofnet routes packets using a link quality aware DSR–like routing protocol, called *Srcr*, exploiting the Estimated Transmission Time (ETT) as routing metric [12]. We decided to exploit the WING toolkit due to both its open-source nature[2] and its flexibility in terms of supported platforms, which allowed us to implement and deploy a testbed exploiting off–the–shelf components.

Each *Cluster head* is built using a PCEngines Alix 2C processor board and is equipped with a 500 MHz processor and 256 MB of RAM. Operating system and application are stored on a 1GB Compact Flash card. Connectivity is provided by 2 Ethernet channels, and 2 Mikrotik RB52 WiFi IEEE 802.11a/b/g cards based on the Atheros AR2412 chipset.

This study has been conducted exploiting 4 mesh routers organized in a linear topology (see Fig. 3). A Dell D630 laptop connected through an Ethernet cable to the fourth *Cluster Head* has been exploited as network *Sink*. Sensor nodes have been emulated by means of a software process running within each *Cluster head*. This process emulates a flat WSN computing both the average and the variance of the physical phenomena monitored by the WSN (e.g. the temperature). Each sensors cluster is composed by 60 nodes.

In order to obtain average and variance, sensor nodes are required to compute:

$$S = \sum_{i=1}^{n} X_i \quad V = \sum_{i=1}^{n} X_i^2 \tag{4}$$

where X_i is the individual value measure by a sensor node and n is the total number of answering sensors. The sink will then receive two distinct values, which can be used to compute both the average $E(x)$ and the variance $Var(x)$:

$$E(x) = \frac{\sum_{i=1}^{n} X_i}{n} \quad E(x^2) = \frac{\sum_{i=1}^{n} X_i^2}{n} \tag{5}$$

$$Var(X) = E(x^2) - E(x)^2 \tag{6}$$

[1]Online resources available at *http://www.wing-project.org/*

[2]All the source code is freely available being release under a BSD License.

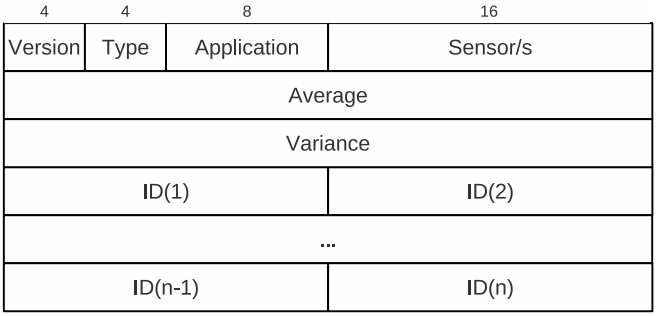

Fig. 4. Message format used by the secure aggregation protocol. The fields in the header are packed with the most significant byte first (big endian).

It is worth noting that, in computing the average, the modulus M must be large enough to prevent any overflow. The modulus is thus chosen as follows: $M = n * p$, where $p = max(m_i)$ is the maximum value that can be assumed by the message, and n is the total number of sensor nodes in the network. Therefore each ciphertexts will be $log(M) = log(p)+log(n)$ bits long. Moreover, if also the variance of the measured data has to be derived an additional modulus M' is necessary for the sum of the squares. As for the average, also M' must be large enough to prevent overflow and it is then chosen as follows: $M' = n * p^2$. The size of the ciphertext is therefore $log(M') = 2 * log(p) + log(n)$ bits.

In the application scenario envisioned in this paper each sensor node periodically samples the environmental temperature. The collected sample is then forwarded to the *Cluster Head*, were the secure aggregation scheme is implemented.

Two strings, each of them 32 bits long, have been used to encode, respectively, the sum of the values reported by each sensor node ($\sum_{i=1}^{n} X_i$) and sum of their squares ($\sum_{i=1}^{n} X_i^2$). Setting the maximum number of sensor nodes allowed in the WSNs to $n = 2^8 = 256$, leaves us with 24 bits to represent p^2. As a result we have the following constraint on the range temperatures that can be represented: $m_i \in [0, 2^{12}]$.In fact, in order to represent the square of the maximum value that can be assume by m_i ($2^{12} = 4096$) without incurring in any overflow, 24 bits are necessary.

The message format, devised in order to implement the secure aggregation scheme, introduces 4 different headers and consists of 6 fields plus an optional list of sensor nodes IDs appended at the end of the message and used only in the AMEX and the IAMEX message types. The fields in the header are packed with the most significant byte first (big endian). The most significant bit is numbered 0, so the *Version* field is actually found at the fourth most significant bits of the first byte. The message format is illustrated in Fig. 4. Here, follows a detailed description of the various fields:

- *Version (4–bits).* The protocol version. At the moment the only possible value is 0.
- *Type (4–bits)*. Message type. At the moment the following message types are used:
  - *IAMEX*. Aggregated message emitted by a sensor node. The *Sensor/s* field contains the number of sensors that contributed to this value. The header is followed by the *ids* of the nodes whose samples have been summed to produce the aggregated value.
  - *AMEX*. Aggregated message emitted by a *Cluster head*. The *Sensor/s* field contains the number of sensors that failed to produce a sample. The header is followed by the *ids* of the non–responding nodes.
  - *Sink*. Sink message emitted by a *Sink*. This message contains the aggregated value in cleartext. The *Sensor/s* field contains the number of sensors that contributed to this value.
  - *Probe*. Probe message emitted by a sensor node. This message carries a single sample encrypted with the sensor's stream cipher. In this case, the *Sensor/s* field contains the node's id rather than the number of sensor nodes that contributed to this reading.
- *Application (8–bits).* Used to distinguish among different set of monitored information (e.g. humidity, pressure, etc.). It can be used to map up to 256 different WSN applications over the same mesh–backhaul.
- *Sensor/s (16–bits).* Different meanings according to the particular message type, as you read above.
- *Average (32–bits).* Sum of the readings produced by the sensor node/s.
- *Variance (32–bits).* Sum of the squares of the readings produced by the sensor node/s.
- *ID(i).* List of sensor nodes' *ids* (16–bit each). Their meaning depends on the particular message type.

Please note that padding is used in order to ensure that the whole message contains an integral number of 32-bit words.

V. Evaluation

In this section we aim at evaluating the bandwidth efficiency of our secure aggregation implementation (*Agg*) in comparison with a baseline scenario where no aggregation is used (*No–Agg*). The hop–by–hop (HBH) aggregation scheme discussed in [2] is not considered in that, albeit characterized by a slightly higher bandwidth transmission gain, it does not address end–to–end security concerns. In fact, in the HBH schemes each node, performing node packet aggregation, has to decrypt the message, before executing the aggregation operations; so at each hop sensing data are in clear. Moreover, HBH requires each node to share same secret key, as a result it is enough for an attacker to compromise a single node in order to gain complete knowledge the monitored environment. The experimental data on which this work is based together with all the scripts used during the post–processing phase are available to the research community at *http://www.wing-project.org/*.

Our reference Wireless Mesh/Sensor Network is composed of 4 cluster organized in a string topology. Each cluster consist of 60 sensor nodes directly connected to their *Cluster Head*. In our emulated scenario, the WSN is required to monitor the temperature of a certain area, as a result each sensor periodically generates a random temperature sample uniformity distributed in $[28, 32]$. Period is set to 5 seconds and the temperature is given in Celsius degrees.

TABLE I
NUMBER OF PACKETS RELAYED BY EACH MESH ROUTER AT EACH HOP. THIS TABLE REPORTS THE RESULTS FOR THE *No–Agg* AND FOR THE *Agg* SCENARIOS.

Hops	No–Agg	Agg	Agg (90%)	Agg (70%)
1	10860	180	180	180
2	21660	180	187	193
3	32520	180	188	193
4	43380	181	188	193

TABLE II
NUMBER OF BYTES RELAYED BY EACH MESH ROUTER AT EACH HOP. THIS TABLE REPORTS THE RESULTS FOR THE *No–Agg* AND FOR THE *Agg* SCENARIOS.

Hops	No–Agg	Agg	Agg (90%)	Agg (70%)
1	434400	7200	8552	10628
2	866400	7200	10784	16036
3	1300800	7200	12532	20096
4	1735200	7240	14086	24084

Table I and II respectively report the number of packets and bytes sent at each hop of the network. As in [2], we consider three scenarios: (i) all sensor nodes reply; (ii) 90% of the nodes replies; and (iii) only 70% of the sensor nodes replies. *Cluster heads* (i.e. mesh routers) do not generates any sample, moreover, we assume that the distribution of non–responding nodes is uniform across all clusters.

As it can be seen, in the *No-Agg* scenario, nodes, closer to the *Sink*, send an amount of data that is significantly higher (see *Hop 4* in the tables) than the data transmitted by the previous *Cluster Heads*. On the other hand, the *Agg* scheme shows a constant number of both single transmissions and amount of data exchanged at each hop. In the remaining two scenarios, the number of transmission remains constant, while the amount of bytes exchange increases at each hop. Such a behavior is due to the *ids* of the non–responding nodes that need to be appended to aggregated sample being transmitted. Such a list becomes larger and larger as the sample get closer to the *Sink*.

Finally, Fig. 5 and 6 report the average and the variance of the data samples gathered using respectively the *No–Agg* scheme and the *Agg* scheme. As it can be seen by the figures, the two approaches lead to similar results, proving the validity of the secure aggregation scheme in a realistic environment.

VI. RELATED WORK

In data aggregation the security issues, data confidentiality and integrity, become vital when sensor nodes are deployed in a hostile environment. In literature there are many works that address such security issues. These works have been classified in hop–by–hop encrypted data aggregation and end-to-end encrypted data aggregation. In the former the data is encrypted by the sensing nodes and decrypted by the aggregator nodes. The aggregator nodes, then, decrypt data coming from the sensing nodes, aggregate data and encrypt the aggregated data again. At last, the *Sink* gets the final encrypted aggregation result and decrypts it. In the end-to-end encrypted data aggregation the intermediate aggregator nodes have not the key and can only do aggregations on the encrypted data. Different hop–by–hop related works [3], [6], [7] assumes that data security is guaranteed by means of some key distribution schemes; for example SEDAN [4] proposes a secure hop–by–hop data aggregation protocol, in which each node can verify immediately the integrity of its two hops neighbours' data and the aggregation of the immediate neighours by means a management of new type of key, called *two hops pairwise key*. SEDAN [4] provides a totally distributed scheme to guarantee data integrity. The SEDAN performance, evaluated by means of ad-hoc simulation, shows a better behavior than other solutions, i.e., SAWAN [3], in terms of overhead and mean time to detection. All hop–by–hop proposed solutions are vulnerable because the intermediate aggregator nodes are easy to tamper and the sensor readings are decrypted on those aggregators. End-to-end encrypted techniques overcome this weakness of hop-by-hop techniques. Notice that end–to–end secure data aggregation techniques also use a key scheme. Some approaches [2], [8] suggest to share a key among all sensing nodes and the *Sink*, the aggregator nodes have not the key because the aggregator nodes handle data without making any encryption/decryption operation. The limitation of such a solution is that the whole network is compromised in case the key is compromised in a sensing nodes.

An alternative approach is represented by the adoption of public-key encryption [13], but in this case the drawback is represented by a high computation consumption. After this short overview, notice that all proposed solutions are based on the adoption of encryption techniques, ad-hoc key distribution schemes [14], [15], [16], authentication, access control solutions in a WSN. Our solution, instead, focuses on the system architecture adopting a hybrid network architecture, composed of Wireless Sensor Network and Wireless Mesh network. More specifically, to guarantee data security a end-to-end secure data aggregation is used, but the aggregation operations are performed by mesh routers, reducing the power consumption of sensor nodes by means a sharing of functions.

VII. CONCLUSION

This paper has proposed a hybrid mesh/sensor network architecture, characterized by a sharing of tasks in order to satisfy the security requirements and the power constrains. Our architecture is suitable to realize an application agnostic mesh backhaul able to support multiple WSNs, while ensuring both end-to-end secure data aggregation.

The performance of the solution has been evaluated by an ad-hoc prototype. The evaluation on the testbed shows the performance of the secure hybrid aggregated approach, defined by us, is better than the available solutions that guarantee security, but without providing aggregated data. More specifically, the performance of secure aggregation approach, besides being better than no-aggregation scheme in terms of amount of both transmitted packets and bytes (clearly reducing the

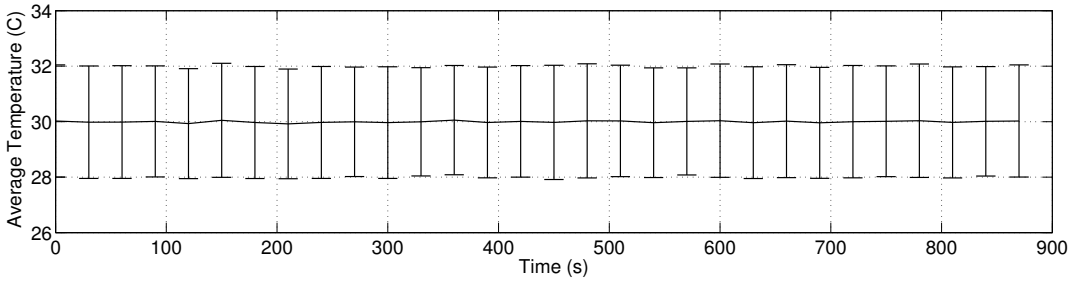

Fig. 5. Average environmental temperature as reported by each sensor node in the *No–Agg* scenario. Samples are averaged every 5 seconds. The variance of the experimental data is reported as errorbar.

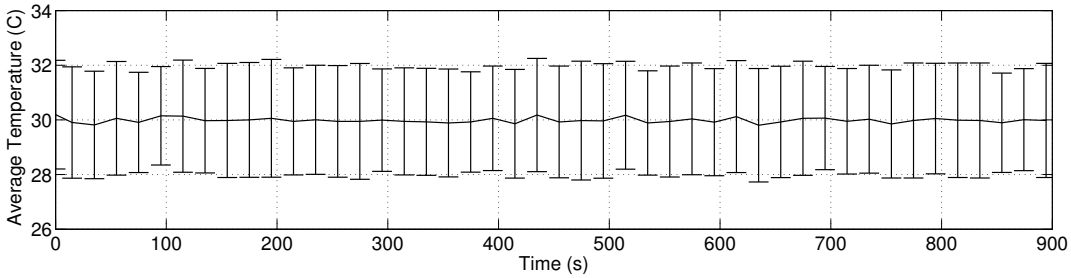

Fig. 6. Average environmental temperature as reported by the entire network *Agg* scenario. The variance of the experimental data is reported as errorbar.

power consumption), preserves data quality. In fact, the *Agg*-approach obtains similar results, i.e., variance and average, to *No-Agg* scheme, validating our approach in a realistic context. The proposed solution is also independent from the types of data that are sensed and handled by the nodes; hence it can be applied to simple networks that sensed the temperature of the environment, as well as to multimedia sensor networks whose nodes may exchange audio and video signals.

Future works concern the development of extensions of the protocol and then the prototype in order guarantee the privacy of node location information and *run-time* data trustworthiness. Moreover, we will introduce other security mechanisms able to reveal malicious behaviors, exploiting the improved power resources of WSNs, thanks to the hybrid architecture. We also plan to exploit our hybrid architecture as reference platform for the development of innovative and really dynamic applications, such as the new Internet of Things applications.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on wireless sensor network," *IEEE Wireless Communications*, vol. 40, no. 8, pp. 102 – 114, August 2002.

[2] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *Proc. of MobiQuitous*, San Diego, CA, USA, 2005.

[3] L. Hu and D. Evans, "Secure data aggregation in wireless sensor networks," in *Proc. of IEEE WSAAN*, 2003.

[4] M. Bagaa, N. Lasla, A. Ouadjaout, and Y. Challal, "Sedan: Secure and efficient protocol for data aggregation in wireless sensor networks," in *Proc. of IEEE LCN*, Dublin, Ireland, 2007.

[5] I. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Elsevier Computer Networks*, vol. 47, no. 4, pp. 445 – 487, Mar. 2005.

[6] A. Mahimkar and T. Rappaport, "Securedav: A secure data aggregation and verification protocol for sensor networks," in *Proc. of IEEE Globecom*, Dallas, TX, USA, 2004.

[7] B. Przydatek, D. Song, and A. Perrig, "Sia: Secure information aggregation in sensor networks," in *Proc. of ACM SenSys*, Los Angeles, CA, USA, 2003.

[8] J.Girao, D.westhoff, and M. Schneider, "Cda: concealed data aggregation for reverse multicast traffic in wireless sensor networks," in *Proc. of IEEE ICC*, Seoul, Korea, 2005.

[9] R. Riggio, N. Scalabrino, D. Miorandi, F. Granelli, Y. Fang, E. Gregori, and I. Chlamtac, "Hardware and software solutions for wireless mesh network testbeds," *IEEE Communication Magazine*, vol. 46, no. 6, pp. 156 – 162, Jun. 2008.

[10] R. Riggio, K. Gomez, T. Rasheed, M. Gerola, , and D. Miorandi, "Mesh your senses: Multimedia applications over wifi-based wireless mesh networks," in *Proc. of Secon*, Rome, Italy, 2009.

[11] "MIT roofnet." [Online]. Available: http://pdos.csail.mit.edu/roofnet/

[12] R. Draves, J. Padhye, and B. Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks," in *Proc. of ACM MOBICOM*, Philadelphia, Pennsylvania, USA, 2004.

[13] E.Mykletun, J.Girao, and D. Westhoff, "Public key based cryptoschemes for data concealment in wireless sensor networks," in *Proc. of IEEE ICC*, Istanbul, Turkey, 2006.

[14] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks." in *Proc. of ACM CCS*, Washington, DC, USA, 2002.

[15] R. D. Pietro, A. Mei, and L. V. Mancini, "Random key assignment for secure wireless sensor networks," in *Proc. of ACM SASN*, Fairfax, VA, USA, 2003.

[16] R. D. Pietro, C. Soriente, A. Spognardi, and G. Tsudik, "Collaborative authentication in unattended wsns," in *Proc. of ACM WiSec*, Zurich, Switzerland, 2009.