This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2011 proceedings.

Performance Evaluation of an Hybrid Mesh and Sensor Network

Roberto Riggio and Tinku Rasheed
CREATE-NET
Via Alla Cascata, 56/C
38123 Trento, Italy
Email: {roberto.riggio, tinku.rasheed}@create-net.org

Sabrina Sicari
Università degli studi dell'Insubria
Dipartimento di Informatica e Comunicazione
Via Mazzini, 5
21100 Varese, Italy
Email: sabrina.sicari@uninsubria.it

***Abstract*—Wireless Sensor Networks (WSNs) provide an extensible and effective means to monitor large and geographically diverse areas. Nodes in a WSN are characterized by very strict constraints in terms of computing power and energy consumption, as a result of which efficient and secure data-aggregation techniques are being proposed as enabling technology for reducing network load, while at the same time providing high sensing accuracy and data integrity. In this paper, we present an hybrid mesh/sensor network architecture based on a sharing of tasks between mesh routers and sensor nodes. 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. Simulation analyses have shown that the proposed scheme can significantly reduce the network load while preserving data confidentiality and integrity. Finally, a real-world prototype has been implemented and tested over a small scale testbed confirming the simulation results.**

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

I. Introduction

Wireless Sensor Network technologies support data collection and distributed data processing by means of very small sensing devices [1] characterized by limited computation and energy capabilities. WSNs are used in many contexts, such as telemedicine, surveillance systems, assistance to disabled and elderly people, environmental monitoring, localization of services and users, industrial process control, and systems supporting traffic monitoring/control in urban/suburban areas, military and/or anti-terrorism operations.

An important goal when designing WSNs is minimizing the number of transmissions and the length of each communication, thus reducing the overall power consumption of the network. Using data aggregation algorithms (e.g., see [2], [3], [4]) can significantly reduce the number of bytes exchanged across the network. However, such solutions raise several privacy and security challenges in that, data aggregation is potentially vulnerable to attackers who may inject bogus information without being detected. Moreover, typical security and privacy solutions, used in wireless networks, are not applicable to WSNs due to their relatively high requirements in terms of computing power. As a result, secure and energy efficient WSNs are receiving considerable attention from the research community [5], [6], [7], [8], [9], [10].

In this paper, we propose an hybrid architecture combining WSNs with wireless mesh networking (WMN)[1]. More specifically, sensor nodes use their resources (i.e. power) only to implement sensing functionalities, while mesh routers perform the secure aggregation of the incoming data, and then relay the aggregated messages to the *Sink* reducing the amount of traffic exchanged over the network and thus the overall power consumption. As a result, our architecture is a perfect candidate to implement the networking backend for an application agnostic middleware, able to support multiple sensing applications while ensuring both end–to–end encryption *and* hop–by–hop authentication. To the best of the authors' knowledge, there are no other works that exploit an hybrid WSN/WMN architecture to jointly address security and power consumption issues.

The paper is organized as follow. Section II summarizes the security model employed by this work. In Section III we introduce the proposed network architecture. Section IV discusses the design choices we made while designing our prototype. Section V presents the results of our simulation campaign, while the results from our prototype–based evaluation are reported in Section VI. A brief overview of the state of the art is presented in Section VII. Finally, Sec. VIII draws some conclusions and provides hints for future works.

II. Security Model

Secure data 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. The main contributions can be grouped into hop-by-hop [3], [12], [13] and end-to-end [2], [14] secure aggregation. The solution presented in this work belongs to the latter category and builds on top of the additively homomorphic stream cipher proposed by Castelluccia et al. in [2]. Such a cipher uses modular additions and is thus very well suited for resources–constrained devices. Data aggregated using this cipher can be used to efficiently compute statistical values such as mean, variance and standard deviation enabling significant bandwidth gain.

[1] This extends our previous work on secure aggregation in hybrid wireless sensors and mesh networks [11] by simplifying the communication protocol and by validating the proposed architecture using both simulations and a real–world prototype.

978-1-4244-9268-8/11/$26.00 ©2011 IEEE

For readers' convenience, the homomorphic encryption scheme proposed in [2] is briefly sketched here. 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:

$$c_i = Enc(m_i; k_i; M) = m_i + k_i(modM) \quad (1)$$

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

$$c = \sum_{i=1}^{k} c_i(modM) \quad (2)$$

The cleartext message can then be obtained by:

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

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 result 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 network architecture for secure aggregation proposed in this work is sketched in Fig. 1. A multi-hop wireless backhaul is exploited by clusters of sensors nodes in order to deliver the sensed data to the *Sink*. Each cluster consists of a variable number of sensors, one *Sensor Head* and one mesh router, which acts as *Cluster Head*.

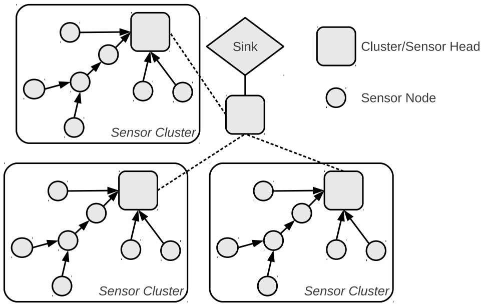


Fig. 1: Reference network model for the hybrid mesh/sensor secure aggregation scheme.

Albeit in the pictures they are shown as single entity, *Cluster Head* and *Sensor Head* functionalities are conceptually separated and as such can be implemented by different nodes, one equipped with a WSN interface (e.g. IEEE802.15.4) and the other equipped with a WMN interface (e.g. IEEE 802.11). In such a case, the sensor node that is directly connected to the *Cluster Head* is termed *Sensor Head*. *Sensor Heads* are in charge for gathering encrypted messages coming from local sensor nodes, while *Cluster Heads* implement 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, however, are not required to reply to all requests. This design choice stems from the consideration that, in a WSN, nodes can be unavailable for a number of reasons ranging from a temporary lack of connectivity, a limited battery, or simply hardware failures or a malicious removal. Nevertheless, if the cleartext message is to be obtained from the aggregated message, the network *Sink* must be able to derive the list of 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). In–cluster aggregation is also supported, allowing sensor nodes to both perform ciphertexts addition and message forwarding. In this configuration, each sensor concatenates the *ids* of the messages being relayed creating a new In-Cluster Aggregated Message (IAMEX). It is worth noticing that, if a locally generated sample is added to the aggregated ciphertext then also the local sensor's *id* shall be appended to the IAMEX message. Please note that the evaluation of the in–cluster aggregation is out of the scope of this work.

An high-level overview of the *Cluster Head*'s architecture is sketched in Fig. 2. Continuous lines represent communication channels that use the IAMEX format, while dashed lines represent communication channels 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

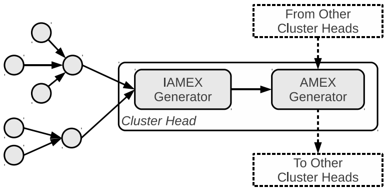


Fig. 2: Architecture of the *Cluster head*.

ciphertexts and appending the nodes' *ids*.

The entire procedure, exploited in order to securely convey and aggregate the samples collected by the sensor nodes to the network *Sink*, can be decomposed into the following steps:

1) *Cluster Heads* periodically poll all the sensors in their cluster. Polling packets can be either flooded across the entire cluster or, if broadcast is not supported, they can be sent using unicast transmissions.
2) Upon polling, each sensor generates a packet containing a single encrypted sample that is then forwarded to the *Sensor Head*.
3) *Sensor Heads* receive the packets coming from sensor nodes in their cluster and store them in a local queue. As soon as N packets are received or when a timeout has expired, each *Sensor Head* aggregates each encrypted samples and generates a *IAMEX* packet that is then sent to the *Cluster Head*.
4) *Cluster Heads* receive both the *IAMEX* packets coming from its cluster and the *AMEX* packets coming from the neighbouring cluster head, then aggregate them into a new single *AMEX* packet which is sent to the *Sink*.
5) The *Sink* receives all the *AMEX* packets, deciphers the ciphertext and computes the relevant statistical information (e.g. average and/or variance).

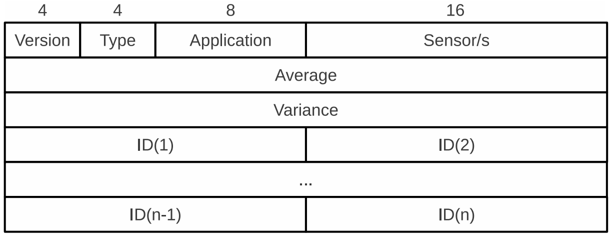
IV. Design Choices

In order to prove its viability in a realistic scenario, we implemented a specific use case on top of our hybrid architecture. The ensuing application computes the average and the variance of the physical phenomena monitored by the WSN (e.g. the temperature). In this section, we will describe the design choices we made while designing the application, while in Sec. V and in Sec. VI, we will report on its evaluation using respectively simulations and a prototypical implementation.

In the application scenario envisioned in this work, each sensor node periodically samples the environmental temperature. The collected data is then forwarded to the *Cluster Head* through the sensor cluster, where the secure aggregation scheme is implemented. 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 measured 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 $Var(x)$:



4	4	8	16
Version	Type	Application	Sensor/s
Average			
Variance			
ID(1)			ID(2)
...			
ID(n-1)			ID(n)

Fig. 3: Message format used in our secure aggregation scheme.

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

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

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.

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 assumed 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. 3. Here, follows a detailed description of the various fields:

- *Version (4–bits)*. The protocol version (set to 0).
- *Type (4–bits)*. The message type:
  - *IAMEX*. Aggregated message emitted by a *Sensor Head*. 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.

- *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.

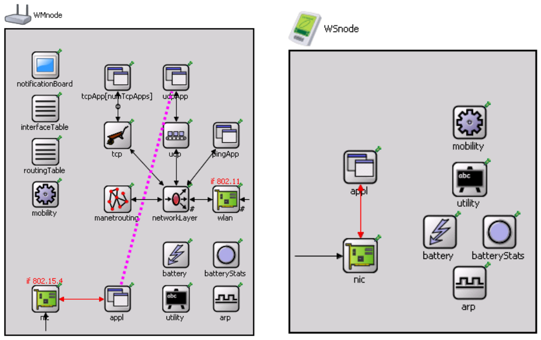
Please note that padding is used in order to ensure that the whole message contains an integral number of 32-bit words.

V. Simulations

In this section, we aim at evaluating the bandwidth efficiency of our secure aggregation architecture (*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 the end–to–end security concerns. The experimental data on which this work is based together with all the scripts used during the post–processing phase are made available to the research community[2]. Notice that, the evaluation of the data confidentiality and integrity features supported by our hybrid architecture has already been provided by Castelluccia et al. in [2] and is thus out of scope for this work.

A. Simulation Environment

The evaluations were carried out using the OMNET++ simulator (version 4.1). The INETMANET and the MiXiM models have been used in order to simulate respectively the IEEE802.11–based mesh backhaul and the IEEE802.15.4–based sensor clusters. Each cluster is composed of one mesh router (See Fig. 4a) equipped with two radio interfaces and one or more wireless sensor/s (See Fig. 4b) each of them equipped with a single radio interface. The primary mesh router interface, derived from the INETMANET framework, is an IEEE 802.11 (WiFi) interface operating in the ISM 2.4 GHz frequency band while the secondary, derived from MiXiM framework, is an IEEE 802.15.4 interface operating in ISM 868 MHz frequency band. It is worth stressing that, the mesh router is equipped with two different interfaces in that it is implementing both *Sensor Head* and *CLuster Head* functionalities. Mesh connectivity is implemented by means of the AODV mesh routing protocol. The sensor nodes are arranged in a star topology around the mesh router. One mesh router acts as a gateway implementing *Sink* functionalities.



(a) Mesh router equipped with two interfaces.

(b) Sensor node equipped with a single interface.

Fig. 4: Simulation environment's setup.

B. Simulation Scenarios

The following scenarios have been considered:

- *No–Agg*. In this scenario, when a *Cluster Head* receives an encrypted packet from either its sensors cluster or neighbouring cluster head, it immediately forwards it to the *Sink*. No aggregation is performed in this scenario which serves as baseline for the rest of the evaluation.
- *Aggregation N*. Every packet received by the *Cluster Head* is stored into a FIFO queue. After the N^{th} arrivals the queue is emptied and an *AMEX* packet is generated and forwarded to the *Sink*. The values of N considered for this study have been $4, 8, 12$.

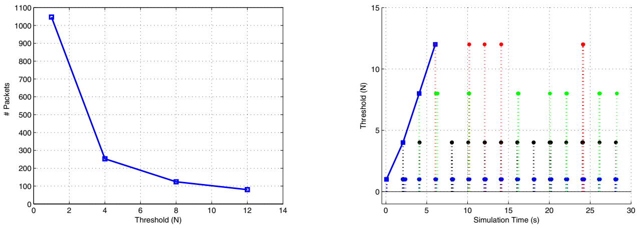
The duration of each simulation has been set to 240 seconds. The results reported in this work are the average of 10 runs executed with different seed values for the random number generator. This section reports the results obtained with 3 sensor clusters each of them containing one mesh router and 7 sensors distributed over a 500x500 meters square field where mesh routers and sensors nodes are randomly distributed at initialization time. Both mesh and sensor node are not mobile.

C. Results

Figure 5a reports the number of packets delivered to the *Sink* over the WiFi interface during the entire simulation time for increasing values of the aggregation threshold N. As expected, increasing the value of N results in a significant reduction in the number of packets delivered to the *Sink*, and thus forwarded across the network which in time results in a lower channel utilization and energy consumption. The initial transition time, which can be noticed in Fig. 5b, is strictly related with the *Cluster Head*'s polling period, which for this simulations was set to 2 seconds.

Finally, Fig. 6 reports the histogram of the AMEX messages' inter-arrival times for different values of the aggregation

[2] Online resources at http://www.wing-project.org/.



(a) Number of packets delivered to the *Sink* for increasing values of N.

(b) Inter–arrivals times of packets for increasing values of N.

Fig. 5: Outcomes of the simulations campaign.

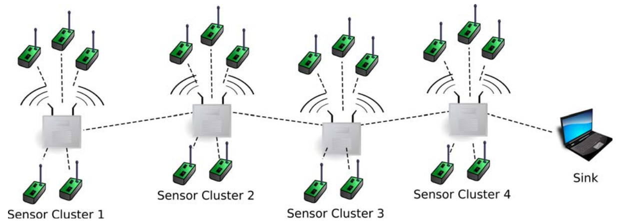


Fig. 7: The linear network topology exploited during our study.

threshold. As expected it can be noticed, the inter-arrival time increases with the value of N, in particular for $N = 12$, intervals as long as 10 seconds can be observed.

VI. Prototype

A prototype has also been implemented and tested in order to demonstrate the practical viability of our approach in realistic settings. This study has been conducted exploiting 4 mesh routers organized in a linear topology (see Fig. 7) and implementing both *Cluster Head* and *Sensor Head* functionalities. 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 mesh router. 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. The mesh backhaul has been implemented using the WING toolkit, an experimental IEEE 802.11 wireless mesh network [15], [16].

In the envisioned application, the WSN is required to monitor the temperature of a certain area, and as a result, each sensor periodically generates a random temperature sample uniformly distributed in $[28, 32]$. Period is set to 5 seconds.

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 the clusters.

As it can be seen, in the *No-Agg* scenario, nodes that are closer to the *Sink* transmit 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, in the *Agg* scheme, the number of transmissions is constant while the amount of bytes exchanged at each hops increases. Such a behavior is due to the *ids* of the non–responding nodes that need to be appended to the aggregated samples being transmitted. Such a list becomes larger and larger as the sample get closer to the *Sink*.

TABLE I: Number of packets relayed at each hop.

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 at each hop.

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

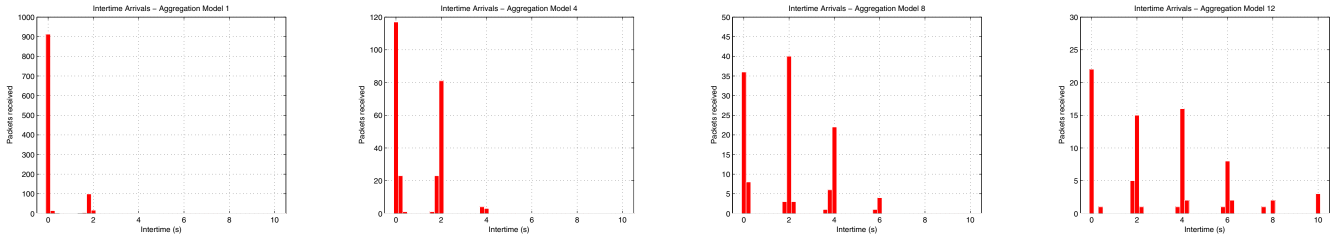
VII. Related work

State-of-the-art solutions for secure data aggregation can be classified as hop–by–hop data aggregation and end–to–end data aggregation. In the former approach, data is encrypted by the sensing nodes and decrypted at each hop before being delivered to the *Sink*. In the latter approach, data is decrypted only by the *Sink*.

Different hop–by–hop solutions [3], [12], [13] 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 the integrity of its two hops neighbors' data. 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. Nevertheless, all hop–by–hop secure data aggregation solutions are vulnerable to attacks at the intermediate nodes, that can be tampered, leaving the attackers with complete access to the sensor readings.

End-to-end techniques, such as [2], [14], [10], overcome this limitation by requiring all the nodes to share an encryption key only with the *Sink* possibly using novel distribution schemes [17], [18], [19]. Particularly, SeDap [10] addresses the privacy as well as security aggregation issues adopting an end-to-end additively homomorphic encryption. An alternative approach is represented by the use of public–key encryption scheme, such as the one presented in [20]. The drawback of this solution is represented by the high computational requirements imposed by public–key schemes.

As opposed to the aforementioned solutions, our work exploits an hybrid sensor/mesh network architecture where an homomorphic encryption scheme is implemented by the sensor nodes, while data aggregation operations are performed by



(a) $N = 1$. (b) $N = 4$. (c) $N = 8$. (d) $N = 12$.

Fig. 6: Histogram of the AMEX messages' inter-arrival times for different values of the aggregation threshold N.

mesh routers that are not required to know the actual content of the message being processed. Our architecture is capable of providing data confidentiality and integrity while, at the same time, reducing the amount of traffic exchanged over the network and thus the overall power consumption.

VIII. Conclusion

In order to make WSNs a viable choice for distributed data collection and processing, security and privacy issues and energy efficiency shall be addressed as a single challenge rather than two incompatible requirements. The hybrid mesh/sensor architecture proposed in this paper is a first step toward such an ambitious goal. The sharing of task that characterize our solution is particularly suitable for designing and implementing an application agnostic mesh backhaul capable of supporting multiple WSN applications while ensuring both end-to-end secure data aggregation. 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 applications measuring average temperature of the environment, as well as to multimedia sensor networks whose nodes may exchange audio and video signals.

In order to prove the viability of our architecture in a realistic scenario, we exploited it to implement a specific use case, namely collecting and processing the mean and the average value of a physical phenomena measured by a WSN (the temperature). The performance of our approach has been deeply analyzed by means of a simulation campaign and a real world prototype. The results showed the our hybrid approach significantly outperform the currently available solutions. As a future work, we plan to address both node location information's privacy issues and *run-time* data trustworthiness. Moreover, we will introduce other security mechanisms able to reveal malicious behaviours. 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, Y. S. W. Su, and E. Cayirci, "A survey on wireless sensor network," *IEEE Wireless Communications*, vol. 40, no. 8, pp. 102 – 104, 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*, Orlando,Florida,USA, 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] M.Monga and S. Sicari, "Assessing data quality by a cross-layer approach," in *Proc. of IEEE International Conference on Ultra modern telecommunications*, St.-Petersburg, Russia, 2009.

[6] M. Monga and S. Sicari, "On the impact of localization data in wireless sensor networks with malicious nodes," in *Proc. of SIGSPATIAL ACM GIS international workshop on security and privacy in GIS and LBS*, New York, NY, USA, 2009.

[7] N. Gatti, M. Monga, and S. Sicari, "A localization game in wireless sensor networks," in *Decision and game theory for security*, ser. SpringerLecture Notes in Computer Science, T. Alpcan, L. Buttyan, and J. Baras, Eds., vol. 6442, Berlin, Germany, 2010.

[8] ——, "Localization security in wireless sensor networks as a non-cooperative game," in *Proc. of IEEE international conference on Ultra modern telecommunications and control systems*, Moscow, Russia, 2010, best paper award winner.

[9] A. Coen-Porisini, P. Colombo, and S. Sicari, "Dealing with anonymity in wireless sensor networks," in *Proc. of ACM International Symposium On Applied Computing (ACM-SAC10)*, Losanna, Switzerland, 2010.

[10] A. Coen-Porisini and S. Sicari, "Sedap: Secure data aggregation protocol in privacy aware wireless sensor networks," in *Springer Proceeding of the 2nd International Conference on Sensor Systems and Software*, Miami, Florida,USA, 2010.

[11] R. Riggio and S. Sicari, "Secure aggregation in hybrid mesh/sensor networks," in *Proc. of SASN*, Saint Petersburg, Russia, 2009.

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

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

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

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

[20] 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.