Adaptive compressive sensing for energy efficient smart objects in IoT applications

Alexandros Fragkiadakis, Pavlos Charalampidis, and Elias Tragos
Institute of Computer Science
Foundation for Research and Technology-Hellas (FORTH-ICS)
Heraklion, Crete, Greece
email: {alfrag, pcharala, etragos}@ics.forth.gr

***Abstract*—The IoT (Internet-of-Things) concept has been introduced as a strategic innovation aspect that will benefit society in many ways. Environmental monitoring, smart traffic control, pollution monitoring, crime prevention, smart metering, etc., have now been made feasible due to the ongoing research and development towards IoT. Smart Objects (SOs) are among the fundamental blocks of an IoT architecture consisting of a sensing element (sensor) that senses the environment, and other software/hardware entities. SOs are interconnected forming wireless sensor networks (WSNs) that often convey sensitive information in a multi-hop fashion. As SOs are severe resource-constrained devices, energy-efficient mechanisms for data collection and transmission are of paramount importance. A popular technique for achieving energy efficiency is Compressive Sensing (CS). CS's major advantage is that it can sample and compress information in a single step. At the same time, CS offers lightweight data encryption during compression. In this work, we propose an adaptive CS scheme where a central SO with enhanced capabilities utilizes a learning phase, associating different sparsity levels with the reconstruction error. Based on this scheme, this SO provides feedback to the rest of the SOs to adapt their CS parameters in order to serve applications with a diverge range of quality-of-service requirements.**

***Index Terms*—adaptive compressive sensing, sparsity, reconstruction error, security, encryption, quality of service**

I. Introduction

WSNs comprise the fundamental blocks of IoT (Internet-of-Things) architectures used to gather a diverse range of measurements. Current technology advances in the electromechanical systems' area have enabled the design of off-the-shelf miniature SOs with relatively enhanced communication and processing capabilities. This along with the proliferation of energy-efficient communication protocols (e.g. IEEE 802.15.4), have given a considerable boost to WSN deployment for serving a large number of applications. Typical WSN applications collect measurements like the ambient temperature, light, humidity, barometric pressure, etc. [1].

Despite the wide use of WSNs in the IoT architectures, there are a few limitations that should be taken into consideration. First of all, SOs are still severe-resource constrained devices, despite the technological advancements in this domain. Many IoT applications are supported by battery-operated SOs; hence, there is always the risk of operation disruption when one or more SOs fail to properly function due to energy shortage. This may not consist a major issue for several applications (e.g. [2]) but for other mission-critical implementations and in possible life-threatening situations (e.g [3], [4]), prolonging SOs lifetime is of major importance. For this reason, energy-efficient mechanisms are a top priority in WSN research domain with a number of significant contributions.

In general, SOs consume energy when performing three main tasks:

- Data sampling that mainly involves sensing from the environment (e.g. ambient temperature)
- Data processing that follows sampling and involves operations like storage, de-noising, etc.
- Communication that includes all necessary networking tasks like packet transmissions and receptions, protocol overheads due to control traffic, etc.

Among all tasks, the communication task consumes the highest amount of energy as packet transmissions and receptions use the radio circuitry of the SO, a hardware component that requires a high amount of energy. As SOs usually operate on the ISM band that is overcrowded and interference is present, further energy is consumed as packet collisions and retransmissions often take place. In this work, we use CS in order to compress the data in SOs prior to transmission, and hence to minimize the energy consumed for communication.

Except the energy efficiency in WSNs, another issue that is of high priority is security and privacy [5]. This is because WSNs often convey sensitive and private information [6]. Security in WSNs is difficult to achieve for two main reasons: (i) robust and energy-efficient encryption algorithms are still in their infancy, (ii) SOs are often placed in unattended areas; hence, can be easily compromised. In this work, we use CS for lightweight encryption of the data SOs transmit to a central node (cluster head). Although there are several works that define clustering with semantic criteria (i.e. using social relationships as in [7]), here we assume that clustering is being done using geographical criteria and the location of the nodes that can be easily identified using in band signalling as proposed in [8].

Related work contains several important contributions. The authors in [9] also consider adaptive CS, computing signal's sparsity. Our work has two main differences: (i) we consider Gaussian and Toeplitz measurement matrices that provide higher secrecy, and (ii) we adapt the feedback sent to SOs based on the QoS requirements of specific applications. In [10], a data gathering CS scheme using Gaussian measurements and exploiting linear spatial correlation between sensor data is proposed. Differently to this approach, we assume compression across temporal dimension and consider also a Toeplitz measurement matrix, which is more suitable for limited-resource systems. The algorithm described in [11] is based on the adaptive CS theory, and jointly optimizes compression and routing steps to obtain optimal, in the information gain sense, measurements. Although improving the accuracy of the reconstruction, this framework substantially increases complexity. Adaptivity of compressive measurement rate, based on the heterogeneity of resource consumption in the nodes of a WSN, is studied in [12]. In our work, however, we consider the sparsity of the sampled signal in order to adjust the measurement rate, taking into consideration the time-varying nature of the signals. In [13], an adaptive CS scheme is proposed but the focus is mainly on designing efficient dictionaries. Furthermore, none of these contributions consider CS adaptation based on specific QoS requirements.

The rest of the paper is organized as follows. In Section II we give an overview of CS background and explain how it is used for simultaneous compression and encryption. Section III describes the proposed adaptive CS-based scheme. In Section IV we present the performance evaluation results. Conclusions and further work appear in Section V.

II. Compressive sensing for IoT applications

A. Background

CS [14] is a relatively new theory that has attracted a lot of interest as it unifies sampling and compression in a single step, and enables a significant reduction in the sampling and computation costs. CS has been used in many research areas, like in wireless intrusion detection [15], energy-efficiency [16], indoor localization [17], etc.

Suppose that the information a SO collects are symbolized by $\mathbf{x} \in \mathbb{R}^N$. According to CS theory, if $\mathbf{x}$ is sparse in some domain, it can be reconstructed exactly with high probability using M randomized projections of signal $\mathbf{x}$ to a measurement matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$, where $M \ll N$. Signal $\mathbf{x}$ is said to be K-sparse in domain $\mathbf{\Psi} \in \mathbb{R}^{N \times N}$ (e.g. FFT, DCT) if it can be written as $\mathbf{x} = \mathbf{\Psi b}$, and $||\mathbf{b}||_0 = K$. Therefore, a signal is K-sparse if only K of its elements in basis $\mathbf{\Psi}$ are non-zero.

The general CS measurement model is expressed as follows:

$$\mathbf{y} = \mathbf{\Phi x} = \mathbf{\Phi \Psi b} = \mathbf{\Theta b} \quad (1)$$

where $\mathbf{\Theta} = \mathbf{\Phi \Psi}$. The original vector $\mathbf{b}$, and consequently the sparse signal $\mathbf{x}$, is estimated by solving the following ℓ_1-norm constrained optimization problem:

$$\hat{\mathbf{b}} = \arg\min ||\mathbf{b}||_1 \quad s.t. \quad \mathbf{y} = \mathbf{\Theta b}. \quad (2)$$

Finally, the reconstructed signal is given by $\hat{\mathbf{x}} = \mathbf{\Psi} \hat{\mathbf{b}}$.

CS performance is evaluated using the reconstruction error (e) defined as $e = \frac{||\mathbf{x}-\hat{\mathbf{x}}||_2}{||\mathbf{x}||_2}$. Error e essentially expresses how much signal $\mathbf{x}$ and $\hat{\mathbf{x}}$ differ. The smaller e is, the higher the fidelity of $\hat{\mathbf{x}}$ to $\mathbf{x}$, therefore, the higher CS performance is. The number of projected measurements M affects error e, as a large M provides a lower compression to the original signal that further leads to a smaller error during reconstruction. In general, a K-sparse signal $\mathbf{x}$ can be reconstructed exactly with high probability if $M \geq CK \log(N/K)$, where $C \in R^+$ [14].

B. Lightweight compression and encryption

As (1) shows, signal $\mathbf{x} \in \mathbb{R}^N$ is multiplied by the measurement matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$, producing signal $\mathbf{y} \in \mathbb{R}^M$. As $M \ll N$, y is a compressed version of the original signal $\mathbf{x}$, and error e depends on the number of projections M, it is now clear that CS enables a lightweight and lossy compression of the original data.

Except the lossy compression capability of CS, referring again to (1), observe that $\mathbf{\Phi}$ can play the role of an encryption matrix in a symmetric-key cipher. Similar ciphers like [18] employ a multiplication of the plaintext with a matrix similar to $\mathbf{\Phi}$ that produces the ciphertext. Assuming $\mathbf{x}$ is the plaintext, $\mathbf{\Phi}$ the encryption matrix, and y the ciphertext, we conclude that CS, except for compression, it also enables encryption, in a single step. The difference of CS with the traditional symmetric-key ciphers is that it forms an under-determined system (more unknowns than equations) that is solved using (2). The authors in [19] show that although CS-based encryption does not achieve Shannon's definition for perfect secrecy, it can however provide a computational guarantee of secrecy. Furthermore, Orsdemir et al. [20], by studying brute force and structured attacks against CS-based encryption, show that the computational complexity for launching these attacks make them infeasible in practice.

The size of matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ determines the compression rate of the original data. The higher M is, the less the data are compressed. At the same time, as $\mathbf{\Phi}$ is used for encryption, its size determines the complexity of guessing it. Hence, here there is clear trade-off: the higher M gets, the less data are compressed, and the more secure encryption is. Smaller data compression can save energy but at the same time makes CS-based encryption weaker. Furthermore, the more data are compressed, the higher error e gets.

Except the size of $\mathbf{\Phi}$ that affects reconstruction quality and encryption strength, its type should be also considered. Related works have shown that when considering measurement matrices built using values selected independently from certain distributions, exact signal recovery can be achieved with high probability. Measurement matrices built from Gaussian distributions have been widely used. However, the generation of a Gaussian distribution may not be easily achieved in practical implementations due to hardware limitations. In [21], the authors show that Toeplitz matrices with entries drawn

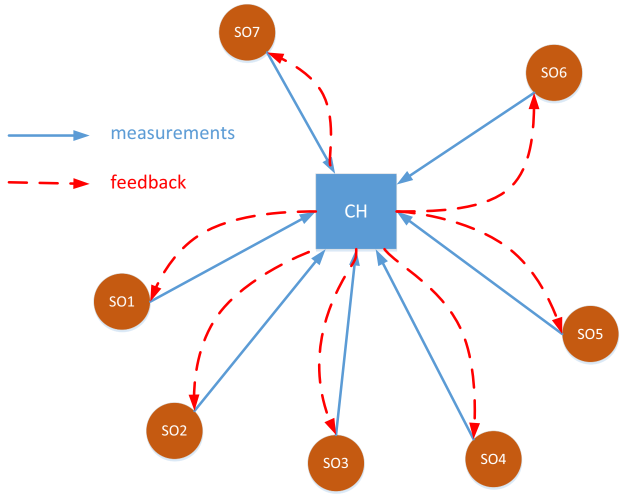


Figure 1: Network model

from the same distributions (e.g. Gaussian) are also sufficient to reconstruct a signal with high probability. In these matrices, all elements belonging to the same diagonal have a common value. Compared to the Gaussian matrices, the Toeplitz have a number of advantages: (i) they require the generation of $O(N)$ random variables instead of $O(MN)$ for the Gaussian case, (ii) the multiplication with a Toeplitz matrix can be performed using FFT and requires only $O(Nlog_2(N))$ operations instead of $O(MN)$ for the Gaussian matrices. At the other hand, Toeplitz matrices usually have a higher reconstruction error, and CS-based encryption is weaker as the elements on the same diagonals take a common value; hence, it is easier for an attacker to derive the encryption key.

III. Adaptive CS-based framework

In this section, we present a novel framework based on adaptive CS for the minimization of energy consumption for data gathering, compression and encryption, based on higher layer constraints in order to support QoS. The ultimate goal is to enable simultaneous compression and encryption of the data SOs collect, taking into account the sparsity of the observed data. In this way, each SO transmits the compressed and encrypted information with parameters defined by a cluster head (CH) so as certain QoS constraints are met.

A. Network Model

The network model on which we deploy our framework is depicted in Figure 1. We assume that there is a number of SOs located at a specific area. The SOs form a federation with one of them that has the dual role of the cluster head (CH) and gateway. The mechanism to select CH is out of the scope of this work. Here, we assume that CH is a more powerful SO (in terms of processing, memory, and energy) that performs the highly computational task of CS reconstruction (decryption). As shown in this figure, the SOs send their measurements to the CH. The latter after receiving these measurements, and for each SO, it computes the optimal compression rate so as certain QoS constraints are met. The optimal compression rate is then sent to the corresponding SO that it further adjusts CS operation based on the received feedback.

B. Proposed framework

The proposed framework for simultaneous compression and encryption in IoT applications based on adaptive CS is shown in Figure 2. This framework is partially based on the idea proposed in [9], however in that work, the complete initial signal is required in order to compute its sparsity, and further derive the optimal compression ratio. In this work, we assume having no knowledge of the complete initial signal; rather we transmit parts of this information. Moreover, we employ Gaussian and Toeplitz measurement matrices that offer enhanced CS-based encryption, compared to [9], where a binary matrix is used; hence, it is more vulnerable to attackers.

The basic goal of our work is to identify a framework for privacy-preserving and energy-efficient data transmission in IoT applications. Assume that the compression rate is CR for N packets, resulting in M compressed packets. Suppose the energy consumed by the SO for transmission of a single packet is E_p, the reconstruction error (computed at CH) is symbolized by e, and the threshold for the reconstruction error is set to Th_{er}, the problem can be formulated as follows:

$$\begin{aligned} \underset{N}{\text{maximize}} \quad & CR(N) \\ \text{subject to} \quad & min(M * E_p), \\ & e < Th_{er} \\ & max(Encryption) \\ & \text{for a given QoS} \end{aligned} \tag{3}$$

Actually, the compression rate computed as $\frac{N-M}{N}$ for a given N, and as the encryption strength is inversely proportional to the compression rate, the previous problem can be transformed to the following simpler problem:

$$\begin{aligned} \underset{N}{\text{minimize}} \quad & M \\ \text{such that} \quad & e < Th_{er} \\ & \text{for a given QoS} \end{aligned} \tag{4}$$

Figure 2 shows our proposed framework. Initially, a SO transmits part of its sensed data to the CH without using CS; hence, data are transmitted un-compressed and in an un-encrypted fashion. After CH receives this data portion, it computes its sparsity. Then, it enters a learning phase where it continuously compresses and decompresses the specific data for different compression rates, computing error e for each rate. Essentially, CH builds a profile where it associates sparsity and error e. This process repeats each time a different sparsity level is detected, and only once for each level.

A wide range of applications execute in IoT architectures with varying QoS characteristics. Mission-critical applications

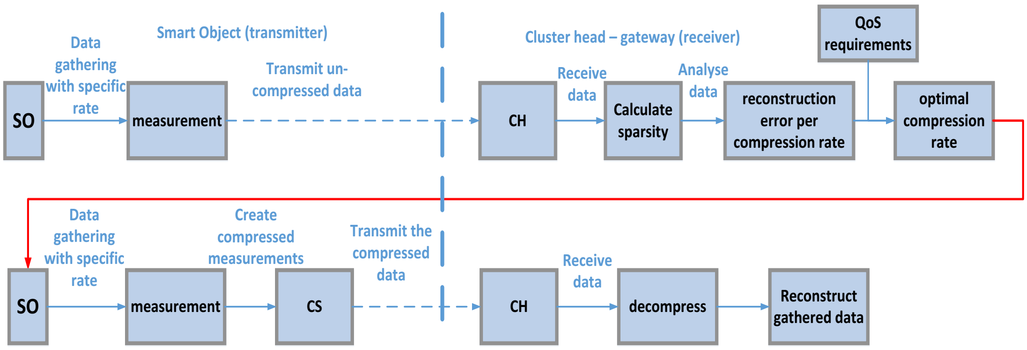

Smart Object (transmitter)
Cluster head – gateway (receiver)
QoS requirements
Data gathering with specific rate
Transmit un-compressed data
SO
measurement
CH
Receive data
Calculate sparsity
Analyse data
reconstruction error per compression rate
optimal compression rate
Data gathering with specific rate
Create compressed measurements
Transmit the compressed data
SO
measurement
CS
CH
Receive data
decompress
Reconstruct gathered data

Figure 2: Adaptive CS-based framework

usually require a very small error e that is directly affected by the data sparsity in this CS-based scheme. The proposed algorithm, after building the profile for each different sparsity level and based on application requirements, it sends a feedback to the SOs so as to adjust their CS parameter. Actually, SOs modify the number of projections M (1) that directly affects the compression rate and error e. Our scheme is flexible enough to cope with sudden changes in sparsity, and to provide the appropriate feedback to the SOs. A sudden sparsity change can happen when for example a fire occurs and temperature abruptly increases.

IV. Performance evaluation

Here, we evaluate the performance of the proposed CS-based scheme described in the previous sections by means of reconstruction error (e) of synthetically generated signals that are k-sparse in the DCT domain. In particular, we generate blocks of $N = 100$ samples with increasing sparsity levels from 10% to 50%, with a step of 10%, while we draw the non-zero DCT coefficients independently from a normal distribution $\mathcal{N}(0,1)$. We create 50 blocks for each sparsity level resulting in a total of 25000 samples. The first block of each sparsity level is used for the learning phase of the reconstruction error by the CH where 100 independent trials of compression-decompression pairs are executed with compression rate (CR) that varies in $[0.1, 0.9]$. Having estimated the simulated reconstruction error per compression rate for a specific sparsity level, the CH evaluates the minimum number of measurements dictated by the QoS requirements (expressed by means of threshold Th). After incremented by a small safety fraction, which in the following is fixed in 5%, this number M_{min} is sent back to the SO, which transmits M_{min} measurements for each of the next 49 blocks.

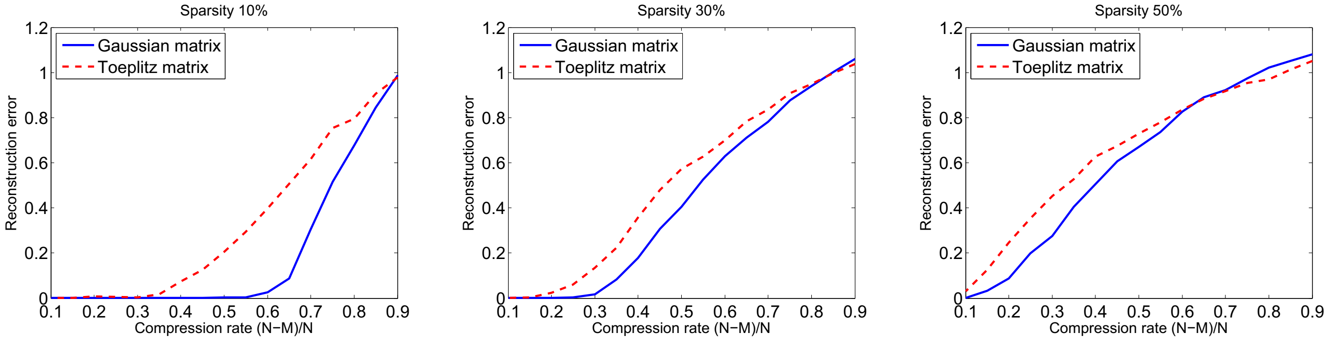
Figures 3a, 3b and 3c show the learned reconstruction error trend (averaged over the 100 trials) against the compression rate for both Gaussian and Toeplitz measurement matrices $\mathbf{\Phi}$, and for the sparsity levels 10%, 30% and 50%, respectively. As shown, the compressed signal can be accurately reconstructed up to a critical value of compression rate, for which the reconstruction error begins to increase rapidly. This critical value decreases with the sparsity level of the signal, clearly following the CS theory.

Regarding the performance of the two different measurement matrices, it can be seen that a Gaussian measurement matrix $\mathbf{\Phi}_G$ generally outperforms a Toeplitz measurement matrix $\mathbf{\Phi}_T$ of the same size. This builds on the fact that $\mathbf{\Phi}_T$ includes much fewer independent random entries than $\mathbf{\Phi}_G$ which, consequently, suggests that $\mathbf{\Phi}_T$ yields less incoherent projections. Additionally, the difference in performance is more profound in lower values of sparsity and especially for small compression rates. For example, if sparsity equals 10%, accurate reconstruction is feasible for $CR < 0.45$ in case of $\mathbf{\Phi}_G$ and $CR < 0.25$ in case of $\mathbf{\Phi}_T$, while if sparsity equals 30% the initial block can be reconstructed for $CR < 0.25$ and $CR < 0.15$, respectively.

Tables I and II present the mean reconstruction error averaged over the 49 per sparsity level evaluation blocks, for two different error thresholds, namely $Th_{er1} = 0.1$ and $Th_{er2} = 0.01$. From these results we make two basic observations. Firstly, our scheme is able to reach the QoS requirements for all sparsity levels and for both measurement matrices. Secondly, using a Gaussian matrix is clearly more efficient than a Toeplitz matrix by means of M_{min}, regardless of sparsity level and reconstruction error threshold.

TABLE I: Mean reconstruction error for $Th_{er1} = 0.1$

Sparsity level	Gaussian		Toeplitz	
	M_{min}	Reconstruction error	M_{min}	Reconstruction error
10%	37	0.0478	63	0.0539
20%	53	0.0849	69	0.0715
30%	69	0.0193	79	0.0395
40%	79	0.0097	90	0.0144
50%	85	0.0252	95	0.0014



(a)

(b)

(c)

Figure 3: Mean learned reconstruction error as a function of the compression rate

TABLE II: Mean reconstruction error for $Th_{er2} = 0.01$

Sparsity level	Gaussian		Toeplitz	
	M_{min}	Reconstruction error	M_{min}	Reconstruction error
10%	48	0.0040	74	0.0019
20%	69	0.0016	79	0.0006
30%	79	0.0009	85	0.0008
40%	90	0.0005	90	0.0009
50%	95	0.0004	95	0.0007

V. Conclusions-Further work

In this paper we presented a novel scheme for adaptive CS, meeting the QoS requirements of IoT applications. CS offers simultaneous compression and encryption in a lightweight fashion. The proposed scheme employs a learning phase where the sparsity level is associated with a specific reconstruction error. This process executes in CH, an advanced SO in terms of processing, memory, and energy. During runtime, and based on the specific QoS requirements, CH transmits a feedback to the SOs defining the compression rate to be used for the CS. Our results show that this mechanism achieves its goal by meeting applications' requirements in terms of the reconstruction error.

Further work will include the investigation of the trade-offs between duration of the learning phase with the energy consumption and the learning phase for different QoS requirements. Furthermore, we will study the trade-off between the compression rate and the encryption strength for different measurement matrices.

Acknowledgment

This work has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 609094.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks, Elsevier*, vol. 52, pp. 2292–2330, 2008.

[2] I. Hakala, M. Tikkakoski, and I. Kivela, "Wireless sensor network in environmental monitoring - case foxhouse," in *Proc. of SENSORCOMM*, 2008, pp. 202–208.

[3] E. Basha, S. Ravela, and D. Rus, "Model-based monitoring for early warning flood detection," in *Proc. of SenSys*, 2008, pp. 295–308.

[4] N. Javaid, N. Khan, M. Shakir, M. Khan, S. Bouk, and Z. Khan, "Ubiquitous healthcare in wireless body area networks - a survey," *CoRR*, vol. abs/1303.2062, 2013.

[5] H. Pohls, V. Angelakis, S. Suppan, K. Fischer, G. Oikonomou, R. Rodriquez, T. Mouroutis, and E. Tragos, "Rerum: Building a reliable iot upon privacy- and security- enabled smart objects," in *Proc. of WCNC*, 2014.

[6] A. Fragkiadakis, I. Askoxylakis, and E. Tragos, "Secure and energy-efficient life-logging in wireless pervasive environments," in *Proc. of the 1st International Conference on Human Aspects of Information Security, Privacy and Trust*, 2013.

[7] P. Karamolegkos *et al.*, "User - profile based communities assessment using clustering methods," in *18th IEEE PIMRC 2007*, Sept 2007, pp. 1–6.

[8] C. Mensing *et al.*, "Location determination using in-band signaling for mobility management in future networks," in *18th IEEE PIMRC 2007*, Sept 2007, pp. 1–5.

[9] W. Chen and I. Wassell, "Energy efficient signal acquisition via compressive sensing in wireless sensor networks," in *Proc. of ISWPC*, 2011.

[10] J. Wang, S. Tang, B. Yin, and X. Li, "Data gathering in wireless sensor networks through intelligent compressive sensing," in *Proc. of Infocom*, 2012, pp. 603–611.

[11] C. Chou, R. Rana, and W. Hu, "Energy efficient information collection in wireless sensor networks using adaptive compressive sensing," in *Proc. of LCN*, 2009, pp. 443–450.

[12] Y. Shen, W. Hu, R. Rana, and C. Chou, "Non-uniform compressive sensing in wireless sensor networks: Feasibility and application," in *Proc. of ISSNIP*, 2011, pp. 271–276.

[13] A. Soni and J. Haupt, "Learning sparse representations for adaptive compressing sensing," in *Proc. of ICASSP*, 2012, pp. 2097–2100.

[14] E. Candes and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[15] A. Fragkiadakis, S. Nikitaki, and P. Tsakalides, "Physical-layer intrusion detection for wireless networks using compressed sensing," in *Proc. of WiMob*, 2012, pp. 845–852.

[16] A. Fragkiadakis, I. Askoxylakis, and E. Tragos, "Joint compressed-sensing and matrix-completion for efficient data collection in wsns," in *Proc. of Camad*, 2013, pp. 84–88.

[17] S. Nikitaki, P. Scholl, K. Laerhoven, and P. Tsakalides, "Localization in wireless networks via laser scanning and bayesian compressed sensing," in *Proc. of SPAWC*, 2013, pp. 739–743.

[18] L. Keliher and A. Delaney, "Cryptanalysis of the toorani-falahati hill ciphers," *IACR Cryptology ePrint Archive*, vol. 2013, 2013.

[19] Y. Rachlin and D. Baron, "The secrecy of compressed sensing measurements," in *Proc. of the Annual Allerton Conference on Communication, Control and Computing*, 2008, pp. 813–817.

[20] A. Orsdemir, H. Altun, and M. B. G. Sharma, "On the security and robustness of encryption via compressed sensing," in *Proc. of MILCOM*, 2008, pp. 1–7.

[21] W. Bajwa, J. Haupt, G. Raz, S. Wright, and R. Nowak, "Toeplitz-structured compressed sensing matrices," in *Proc. of SSP*, 2007, pp. 295–298.