Information Processing Letters 116 (2016) 279–283

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Introducing the counter mode of operation to Compressed Sensing based encryption

Robin Fay

Chair for Data Communications Systems, University of Siegen, Hoelderlinstr. 3, 57068 Siegen, Germany

ARTICLE INFO

Article history:
Received 13 July 2015
Received in revised form 2 November 2015
Accepted 13 November 2015
Available online 19 November 2015
Communicated by L. Viganò

Keywords:
Compressed Sensing
Cryptography
Modes of operation

ABSTRACT

Compressed Sensing based encryption is computationally secure in a one time key scenario, but it does not resist chosen-plaintext attacks (CPA) due to the deterministic encryption process. This paper introduces the counter mode of operation to Compressed Sensing based encryption in order to achieve probabilistic encryption with security against chosen-plaintext attacks. In particular, the proposed scheme addresses the case where multiple signals are encrypted under one master key. The security of the proposed scheme is solely based on the inherent secrecy of the compressed measurements, meaning that no additional ciphers are utilized to ensure CPA-security. To achieve this objective, a method for updating the secret sensing matrix on every signal is presented, such that each signal is encrypted under a fresh pseudorandom matrix.

1. Introduction

In the Compressed Sensing (CS) framework confidentiality is achieved by treating the coefficients of the underlying underdetermined system of linear equations as a shared secret between sender and recipient. By this means, compression and encryption can be performed in a single operation where the additional costs are limited to key management only. Prior studies showed that breaking this kind of encryption is computationally hard for an eavesdropping adversary in the case where just one signal is encrypted under a particular matrix [1]. Conversely, it is clear that the adversary learns useful information if a matrix is used to encrypt multiple signals, since encryption is performed in Electronic Codebook (ECB) mode. This circumstance is a primary motivation to develop and investigate methods towards protecting Compressed Sensing based encryption against chosen-plaintext attacks (CPA).

Huang et al. [2] proposed an image encryption scheme, where CPA-security is achieved through Compressed Sensing followed by additional substitution and permutation layers. While this approach is application specific, it presents a general concept to achieve CPA-security in the framework of Compressed Sensing that may be called *compress-then-encrypt*, because resistance against CPA is achieved by supplementary block cipher components. Indeed, these components introduce further costs, although the secrecy of the compressed measurements may be exploited directly to ensure CPA-security. Apart from that, Zhang et al. [3] suggested to use a bi-level protection where a distinct key is used in order to generate a key-related sparsifying basis besides the secret sensing matrix. This approach is more general than the compress-then-encrypt concept but it might prove hard to find suitable sparsifying bases for each particular application.

In modern cryptography, security against an active attacker is achieved by running the block cipher in some mode of operation, which turns the deterministic behavior of the block cipher after fixing the key into a probabilistic encryption scheme. This paper introduces a mode of operation to Compressed Sensing based encryption in order to ensure confidentiality when one shared master key is used to encrypt multiple messages. More precisely, this paper

E-mail address: robin.fay@uni-siegen.de.

http://dx.doi.org/10.1016/j.ipl.2015.11.010
0020-0190/

describes a general model in order to alter the secret sensing matrix on every new signal.

The rest of this paper is organized as follows: Section 2 presents related works that deal with general properties and requirements of Compressed Sensing and the secrecy of Compressed Sensing based encryption. The overall challenges with CPA-security in Compressed Sensing are discussed and the core idea of the proposed method is explained. Section 3 presents the design of the so called *Compressed Sensing Counter Mode of operation* (CS-CTR) as well as its properties and some implementation details followed by a short experimental proof of concept. Section 3.5 discusses the security of the proposed mode. The final section draws a conclusion and states the future work.

2. Related work

The main challenge with CPA-security in Compressed Sensing is the linearity of the encryption process. Let $\vec{x} \in \mathbb{R}^N$ be the plaintext signal, which is assumed to be s-sparse or compressible in some domain Ψ meaning that there exists some $\vec{s}$ with

$$\vec{s} = \Psi\vec{x} \tag{1}$$

where only s entries in $\vec{s}$ are nonzero. If $\vec{x}$ is s-sparse, we may set $\Psi = I_N$.

Further let $A \in \mathbb{R}^{m\times N}(N \gg m)$ be the secret sensing matrix and let $\vec{y} \in \mathbb{R}^m$ be the ciphertext vector. Then, the sensing process respectively encryption function is defined as:

$$\vec{y} = A\vec{x} \tag{2}$$

The randomness which allows reconstruction of the sparse signal [4,5], is also necessary for the purpose of encryption and is introduced by the sensing matrix A whose entries are chosen at random from a (sub) Gaussian distribution.

The common sensing matrices for practical applications are binary sometimes called Bernoulli sensing matrices, meaning that their entries are drawn uniformly at random from the set $\{-1, 1\}$. The bit sequences used in Compressed Sensing are assumed to be from the set $\{-1, 1\}^*$ instead of $\{0, 1\}^*$, as long as not mentioned otherwise. From a cryptographic point of view, the matrix generation can be modeled using a shared secret key k as a seed for a secure pseudorandom number generator. The key needs to be random and sufficient large, say $|k| \geq 128$-Bit. Rachlin and Baron showed in [1], that it is computationally hard for an adversary to reconstruct the original signal from eavesdropped measurements without knowing the sensing matrix. An exhaustive search of all binary sensing matrices of size $m \times N$ would have complexity $2^{m\cdot N}$. In practical scenarios it can be assumed that $N \cdot m > |k|$ holds. Hence, if a pseudorandom number generator is used for matrix generation, the computational complexity of a brute force attack is reduced to the size of the shared secret key.

As described independently by Bianchi et al. [6] and Cambareri et al. [7], the encryption process preserves the signals energy so that an adversary is able to distinguish between the encryption of two signals with different energy. Cambareri et al. proved that, for large enough N, Compressed Sensing based encryption with sub Gaussian matrices leaks no information about the signal but its energy and they named this *asymptotic spherical secrecy*. Furthermore, Bianchi et al. claimed, that information theoretic secrecy can be obtained if Gaussian random matrices are used when the measurements are normalized to the same energy. Since the measurements energy needs to be known to the recipient it must be transmitted over a secure channel, which is protected using classical cryptography.

However, this strategy does not protect against an active attacker performing a chosen-plaintext attack as long as multiple signals are encrypted under the same matrix. In the CPA scenario, the adversary has access to an encryption oracle which encrypts arbitrary plaintexts of his/her choice. In order to break Compressed Sensing based encryption in ECB mode, an adversary would ask his/her encryption oracle for the encryption of all unit vectors of the standard basis and he/she would obtain the columns of the secret sensing matrix. Even if the measurements are normalized to the same energy, the attacker would gain enough useful information to break the system. For example, assume that binary random matrices are used. In this case, the adversary is just interested in the measurements sign, which does not change due to normalization. With Gaussian random matrices, the captured sensing matrix is equal to the original matrix up to a scaling factor, thus still useful for reconstruction [8, chap. 3]. Based on the fact that Compressed Sensing based encryption is deterministic for a fixed A, a general solution to achieve a probabilistic encryption scheme is to use a different random A on every new signal. This will render the previously mentioned attack useless, since an adversary would only obtain the columns of independent sensing matrices.

The main contribution of this letter is to lift the theoretical results from [6] and [7] to a more practical level, by exploiting the inherent secrecy of Compressive Sensing in order to achieve security against CPAs in a multiple encryption scenario.

3. The compressed sensing counter mode

The proposed solution is based upon the fact that encryption by Compressed Sensing leaks no information about the plaintext but its energy. It is stressed that the proposed encryption scheme does not leak additional information to an attacker even if he/she has access to an encryption oracle. The general design of the proposed *CS-CTR* mode of operation is shown in Fig. 1.

3.1. Algorithm description

At first, assume that the sender and recipient are honest parties sharing a secret key k. Both sides agree publicly on a function *rec* from the family of suitable Compressed Sensing reconstruction algorithms and an optional sparsifying basis Ψ. Further details about the reconstruction function are omitted here for the sake of adaptability, since there are many suitable candidate functions depending on the application (see [8, chap. 4/5]). If the total number of plaintexts is denoted by l, let $A_i \in \{-1, 1\}^{m\times N}$ be the sens-

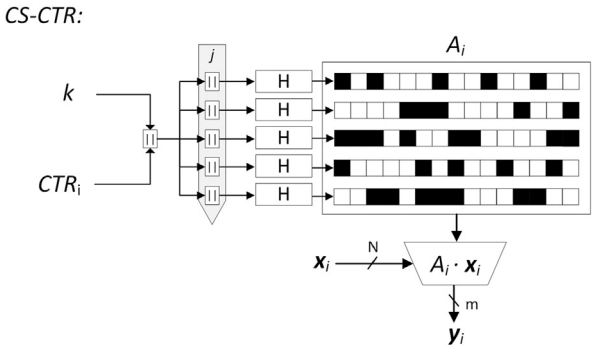

Fig. 1. Conceptual image of the Compressed Sensing Counter Mode encryption. The scheme generates random binary matrices by evaluating a pseudorandom function H at different points consisting of a shared secret key k, a counter CTR_i and the matrix row index j (where $\|$ denotes concatenation). Section 3.1 explains the notation as well as encryption and decryption in more detail.

ing matrix used to compress and encrypt the i-th plaintext $\vec{x}_i \in \mathbb{R}^N$ by

$$\vec{y}_i = A_i \vec{x}_i, \text{where } i = 1, \ldots l;\ \vec{y}_i \in \mathbb{R}^m. \tag{3}$$

Further, let $\alpha_i^j \in \{-1, 1\}^N$ be the j-th row vector of A_i with $j = 1, \ldots, m$. It is assumed that the length of plain- and ciphertext i.e. N respectively m are fixed for the whole encryption process. The set of all plain- and ciphertexts is determined as $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_l\}$ and $Y = \{IV, \vec{y}_1, \vec{y}_2, \ldots, \vec{y}_l\}$ respectively. Note that $|Y| = l + 1$. This is due to the fact that CS-CTR requires an n-Bit initialization vector IV, which should be sent in the clear at the beginning of the communication. The initialization vector is used as a randomized *counter* (see *NIST SP 800-38A* or *ISO/IEC 10116*) such that:

$$CTR_1 = IV,\ CTR_{i+1} = (CTR_i + 1) \bmod 2^n. \tag{4}$$

Finally, fix a cryptographic hash function $H : \{0, 1\}^* \to \{-1, 1\}^{L_H}$ with output bit-length $L_H \geq N$ and generate the matrix A_i as follows:

$$A_i = \begin{pmatrix} \alpha_i^1 \\ \vdots \\ \alpha_i^m \end{pmatrix} \text{ where } \alpha_i^j = H(\mathrm{k} \,\|\, CTR_i \,\|\, j). \tag{5}$$

The keyed hash function H is required to be a pseudorandom function (PRF), so that each row is a series of uniformly distributed random bits, guaranteeing the necessary conditions for reconstruction. Note that for sake of simplicity the range of H is $\{-1, 1\}^{L_H}$. However, H can be implemented using a common cryptographic hash function with range $\{0, 1\}^{L_H}$ where the outputted zeros are replaced with minus ones. More details on appropriate candidates for H and modifications for the case where $L_H < N$ are given in section 3.3.

Now, the sender is able to jointly encrypt and compress the plaintext signals. He chooses an n-Bit IV at random and computes the corresponding counter and secret sensing matrix using the equations (4) and (5). After that, he/she computes (3) for each signal and sends Y to the recipient.

The recipient obtains the IV as part of the ciphertext and derives the counter using equation (4). He/she generates the associated sensing matrix with (5) and computes

$$\hat{x}_i = rec(\vec{y}_i, A_i, [\Psi]). \tag{6}$$

The Compressed Sensing reconstruction algorithm *rec* takes as input the measurement vector together with the corresponding sensing matrix and an optional sparsifying basis Ψ. The function outputs the plaintext or more precisely the signal $\hat{x}_i = \vec{x}_i + \epsilon$ that is reconstructed from the compressed measurements and may contain some noise ϵ.

3.2. Properties

Similar to classical block cipher modes of operations, the proposed mode should be analyzed with respect to *error propagation*, *synchronization* and *parallelizability*.

Error propagation: An error during the transmission of some $\vec{y}_i$ has no effect on A_{i+1}, hence only the corresponding plaintext vector is corrupted. However, depending on the kind of error, Compressed Sensing offers some robustness e.g. against noise.

Synchronization: Encryption and decryption work synchronously as long as the counters are in sync. For example, the loss of some $\vec{y}_i$ during transmission has the effect, that the sender is using a different matrix then the recipient leading to improper decryption. A workaround to this problem is for instance to use the packet numbers of connection-oriented transport protocols to derive the correct counter value.

Parallelizability: One benefit of Compressed Sensing is that sampling and reconstruction can be performed in parallel for many signals. CS-CTR supports this parallel processing, since there are no dependencies between two or more distinct ciphertexts. Therefore, this mode also ensures *random access*, meaning that for the decryption of a particular vector, the recipient does not need to decrypt any previous vector as long as he/she knows the corresponding counter value e.g. derived from a packet number.

3.3. Implementation details

The function H is required to be a PRF when it is used as a keyed hash function of the form $H(k \,\|\, x)$. Hash functions that follow the Merkle–Damgård construction (e.g. *SHA* 1 or *SHA* 2) are no PRFs when the key is used as the prefix of the hash functions input due to their length extension property [9]. If CS-CTR should be implemented using a Merkle–Damgård hash, it is recommended to run H in *HMAC* mode. Moreover, *HMAC* was proven to be a PRF even if the underlying compression function is not weakly collision resistant, which applies to *SHA* 1 [10]. In general, H is proposed to be a hash function which was designed to be indistinguishable from a random oracle when used as a keyed hash function, like *Keccak* [11] or *Skein* [12]. The indistinguishability of these two hash functions in construction (5), assuming no flaws in the underlying primitives, was proven in [13] respectively [14].

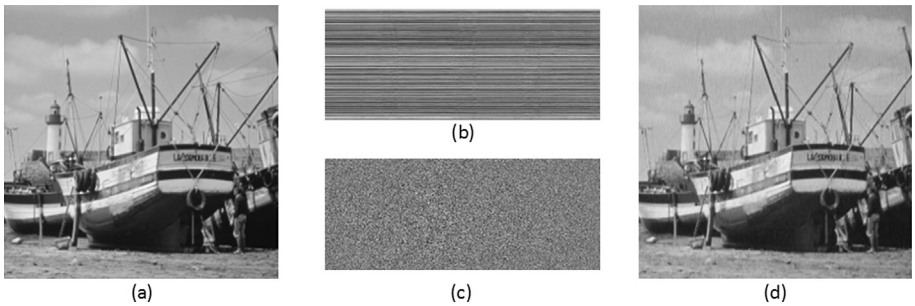

Fig. 2. (a) The original test image. (b) The result of Compressed Sensing electronic codebook encryption. (c) The compressed and encrypted image using CS-CTR mode of operation with a compression ratio of nearly 40% ($N = 512$, $m = 205$, $H \hat{=}$ HMAC-SHA512, 128-Bit random IV). (d) Image reconstructed from (c) using *TVAL3* [15].

In addition, the output length of the hash function must satisfy $L_H \geq N$, since the output is used as the rows of the sensing matrix. Fortunately, both *Keccak* and *Skein* support variable output lengths. In the case where $L_H > N$ the hash functions output should be truncated to fit to the rows of the sensing matrix. In some practical applications of Compressed Sensing, N might be bigger than the output length of some popular hash functions. When it is not possible to use a hash function with arbitrary output length, A_i should be computed as follows ($\lceil\cdot\rceil$ denotes the ceiling function):

$$A_i = \begin{pmatrix} \alpha_i^1 \\ \vdots \\ \alpha_i^m \end{pmatrix} \text{ where } \alpha_i^j = (\beta_i^{j,1}, \ldots, \beta_i^{j,\lceil N/L_H \rceil}) \tag{7}$$

with

$$\beta_i^{j,t} = H(\mathrm{k} \,\|\, CTR_i \,\|\, j \,\|\, t), \text{ for } t = 1, \ldots, \lceil N/L_H \rceil \tag{8}$$

This way, the hash function is called $\lceil N/L_H \rceil$ times per row. The parameter t ensures that the hash function input changes on every call. In general, the computational overhead of CS-CTR compared to Compressed Sensing in ECB mode is limited to $\lceil N/L_H \rceil \cdot m$ hash function calls for each matrix generation.

3.4. Experimental results

For better illustration, the signal vectors are taken from an $N \times N$ image such that each column of the image represents one signal. However, plain Compressed Sensing as well as CS-CTR are lacking the diffusion property of classical (image) ciphers and will only guarantee secrecy up to the point where an adversary learns no information about the plaintext but its energy. Fig. 2 shows the encryption of the *boats* test image in CS-ECB and CS-CTR mode and the reconstruction with *TVAL3* [15]. This example demonstrates the weakness of ECB encryption and shows that CS-CTR preserves the randomness of the sensing matrix which is required for signal reconstruction.

3.5. Security

While the compressed measurements are computational secure against eavesdropping adversaries in a one time key scenario [1], it is clear that the CS-ECB mode is deterministic and therefore it cannot be secure against more powerful attackers. Secrecy for multiple encryption can be achieved if a fresh random sensing matrix is used on every signal, which reduces the adversary's task to multiple instances of Compressive Sensing based encryption [6,7]. In this case, even if the adversary has access to an encryption oracle the encrypted measurements are uncorrelated which is why he/she will not learn useful information through his/her CPA-queries. Instead of drawing a fresh matrix at random, CS-CTR uses the output of a PRF to create random matrices. Therefore, the secrecy of the PRF in the proposed construction must be examined. While a brute force attack against the PRF runs in time exponential in the size of k, it is clear that after the encryption of 2^n signals the sensing matrix repeats due to counter reuse. Hence, the adversary is able to predict matrix reuse, which is why we stress that $l < 2^n$. Even in this case it must be ensured that the adversary cannot distinguish H from a random oracle.

Let $L_H \geq N$ and further $l < 2^n \ll 2^{d/2}$, where $d = \min(N, c)$ denotes the security parameter of H when truncated to N bits, and let c be the size of its inner stage, e.g. the capacity in case of *Keccak* [11]. Now, the complexity of distinguishing the pseudorandom matrix generation from i.i.d. random matrices can be stated as

$$\min\left(\frac{2^d}{l \cdot m}, 2^{|k|}\right), \tag{9}$$

where l is the number of encrypted signals and m is the number of rows in the secret sensing matrix.

4. Conclusion and future work

This paper introduced the Counter Mode of operation to Compressed Sensing based encryption. The proposed mode of operation is as secure as the underlying Compressed Sensing based encryption but in a scenario where one master key is used to encrypt multiple messages. Compared to the methods based on the *compress-then-encrypt* structure, security against chosen-plaintext attacks is directly integrated in the Compressed Sensing process without the need of additional block ciphers such that the system joins encryption and compression in one step. CS-CTR is suitable for all applications where CPA-secure encryption should be integrated directly in the sensing process while

hiding the signals energy is not crucial. Our future work involves formal security definitions for modes of operation in Compressed Sensing based encryption and a detailed security proof of CS-CTR with respect to this definitions.

Acknowledgements

This research was funded by the German Research Foundation (DFG) under grant number Ru 600/11-1.

References

[1] Y. Rachlin, D. Baron, The secrecy of compressed sensing measurements, in: 46th Annual Allerton Conference on Communication, Control, and Computing, 2008, pp. 813–817.
[2] R. Huang, K. Rhee, S. Uchida, A parallel image encryption method based on compressive sensing, Multimed. Tools Appl. 72 (1) (2014) 71–93.
[3] L.Y. Zhang, K.-W. Wong, C. Li, Y. Zhang, Towards secure compressive sampling scheme, arXiv preprint arXiv:1406.1725, 2014.
[4] E.J. Candes, T. Tao, Near-optimal signal recovery from random projections: universal encoding strategies?, IEEE Trans. Inf. Theory 52 (12) (2006) 5406–5425.
[5] E.J. Candes, The restricted isometry property and its implications for compressed sensing, C. R. Math. 346 (9) (2008) 589–592.
[6] T. Bianchi, V. Bioglio, E. Magli, On the security of random linear measurements, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, May 2014, pp. 3992–3996.
[7] V. Cambareri, M. Mangia, F. Pareschi, R. Rovatti, G. Setti, Low-complexity multiclass encryption by compressed sensing, IEEE Trans. Signal Process. 63 (9) (May 2015) 2183–2195.
[8] R. Baraniuk, M.A. Davenport, M.F. Duarte, C. Hegde, An introduction to compressive sensing. Connexions e-textbook, 2011.
[9] G. Tsudik, Message authentication with one-way hash functions, in: Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, INFOCOM'92, IEEE, May 1992, pp. 2055–2059.
[10] M. Bellare, New proofs for NMAC and HMAC: security without collision-resistance, in: Advances in Cryptology, CRYPTO 2006, Springer, Berlin, Heidelberg, 2006, pp. 602–619.
[11] G. Bertoni, J. Daemen, M. Peeters, G. Assche, The Keccak reference, Submission to NIST (Round 3), 2011.
[12] N. Ferguson, S. Lucks, B. Schneier, D. Whiting, M. Bellare, T. Kohno, J. Callas, J. Walker, The Skein hash function family, Submission to NIST (Round 3), vol. 7, no. 7.5, p. 3, 2010.
[13] G. Bertoni, J. Daemen, M. Peeters, G. Assche, On the security of the keyed sponge construction, in: SKEW, 2011.
[14] M. Bellare, T. Kohno, S. Lucks, N. Ferguson, B. Schneier, D. Whiting, J. Callas, J. Walker, Provable security support for the Skein hash family, available online: https://www.schneier.com/skein-proofs.pdf, 2009.
[15] C. Li, An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing, Ph.D. dissertation, Rice University, 2009.