Selective encryption in the CCSDS standard for lossless and near-lossless multispectral and hyperspectral image compression

Selective encryption in the CCSDS standard for lossless and near-lossless multispectral and hyperspectral image compression

Migliorati, Andrea, Bianchi, Tiziano, Magli, Enrico

Andrea Migliorati, Tiziano Bianchi, Enrico Magli, "Selective encryption in the CCSDS standard for lossless and near-lossless multispectral and hyperspectral image compression," Proc. SPIE 11533, Image and Signal Processing for Remote Sensing XXVI, 1153312 (20 September 2020); doi: 10.1117/12.2572991

SPIE.

Event: SPIE Remote Sensing, 2020, Online Only

Selective encryption in the CCSDS standard for lossless and near-lossless multispectral and hyperspectral image compression

Andrea Migliorati[1], Tiziano Bianchi[1], and Enrico Magli[1]

[1]Politecnico di Torino, Torino, IT

ABSTRACT

In this paper, we investigate low-complexity encryption solutions to be embedded in the recently proposed CCSDS standard for lossless and near-lossless multispectral and hyperspectral image compression. The proposed approach is based on the randomization of selected components in the image compression pipeline, namely the sign of prediction residual and the fixed part of Rice-Golomb codes, inspired by similar solutions adopted in video coding. Thanks to the adaptive nature of the CCSDS algorithm, even simple randomization of the sign of prediction residuals can provide a sufficient scrambling of the decoded image when the encryption key is not available. Results on the standard CCSDS test set show that the proposed technique uses on average only about 20% of the keystream compared to a conventional stream cipher, with a negligible increase of the rate of the encoder.

Keywords: Satellite Imaging, Remote Sensing, On-board Image Compression, Encryption

1. INTRODUCTION

Images generated by Earth observation (EO) satellites are nowadays used in several applications providing basic services, including environmental monitoring and assessment, emergency management, civilian security. While the utility of these applications is apparent, at the same time the availability of large amounts of data representing the Earth surface poses significant risk in terms of both security and privacy.

The Consultative Committee for Space Data Systems (CCSDS) recognized the need for specific security protocols[1] and cryptographic algorithms[2] in order to protect the communications between the satellite and the ground segment and ensure that payload data are received only by the intended targets. However, deploying such solutions on actual satellites may prove difficult, especially when large payloads have to be transmitted to ground with very low latency. For example, considering current missions, MetOp satellites offer encryption of the downlink at 3.5 Mbps,[3] which is adequate for MetOp instruments but would be clearly insufficient for instruments like the Sentinel-2 satellite sensor providing a swath width of 290 Km at 10 meter resolution.[4]

Moreover, in order to deliver the final EO products to the final users with very low latency, recent trends in satellite imaging tends to move on-board several tasks that were traditionally performed on the ground segment, like image generation and image processing. An example of this approach can be found in the EO-ALERT project [*], which aims at providing alerts to end users with very low latency.[5] While on-board processing maximizes the usefulness of EO images, eliminating the latency due to transmission of raw data and subsequent processing on ground, it also increases on-board computational requirements, hindering the implementation of on-board fast encryption modules.

Further author information:
Andrea Migliorati andrea.migliorati@polito.it
Tiziano Bianchi: tiziano.bianchi@polito.it
Enrico Magli: enrico.magli@polito.it

[*]EO-ALERT, Next Generation Satellite Processing Chain for Rapid Civil Alerts: http://eo-alert-h2020.eu/

Image and Signal Processing for Remote Sensing XXVI, edited by Lorenzo Bruzzone, Francesca Bovolo
Emanuele Santi, Proc. of SPIE Vol. 11533, 1153312 · © 2020 SPIE
CCC code: 0277-786X/20/$21 · doi: 10.1117/12.2572991

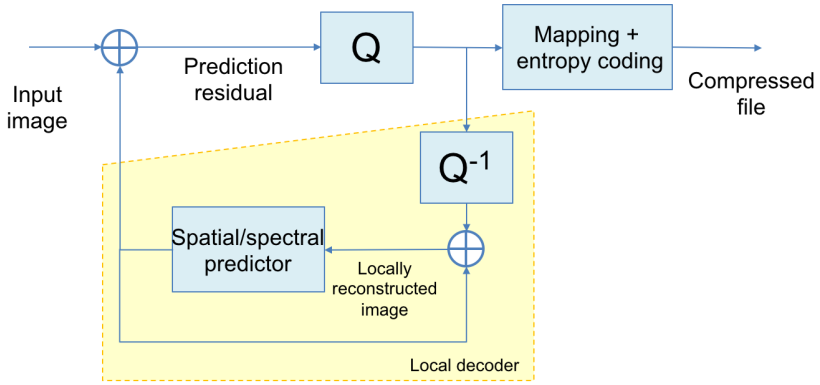


Figure 1. Block diagram of CCSDS prediction-based near-lossless compression.

In order to the address the above issues, we propose a flexible low-complexity encryption modality that can be embedded in the image compression algorithm. Namely, we propose to extend the recent CCSDS standard for lossless and near-lossless image compression[6] by introducing selective encryption of some of its syntax elements. The proposed technique is inspired by selective encryption modalities often employed for encoded video streams.[7–11] However, an important difference is that the CCSDS standard is not transform-based like most video codecs, but employs an adaptive predictor that introduces different constraints in order to achieve a format-compliant and rate-preserving solution. Moreover, while selective encryption of videos often produces only a visual degradation which preserves some macro features, selective encryption of prediction residuals usually results in a dramatic propagation of decoding errors, which makes the image decoded by a non-legitimate receiver completely scrambled.

2. BACKGROUND

2.1 CCSDS standard

In this section, we briefly introduce the CCSDS standard, which is based on a DPCM prediction loop as in Fig. 1. The mathematical model is based on a spatial and spectral predictor that aims at estimating the value of the current pixel to be encoded. The algorithm predicts the value of each sample by using the values of previously decoded samples located in a small neighborhood across the two spatial dimensions and the spectral dimension, so that the decoder can compute the predictor without having access to the original pixels.

Prediction is accomplished by using a least-mean square filter with a weight update performed via the sign algorithm in order to minimize the energy of the residual, i.e. the difference between the true pixel value and its estimate. In particular, the prediction residual is quantized to a signed integer quantization index which is in turn used to calculate a positive *mapped quantizer index*. The mapped quantizer index is then used to update the weight vector by means of the sign algorithm, such that the new weight vector will be used for the prediction of the next sample to be compressed.

Finally, the mapped quantized indexes go through a sample-adaptive entropy coding stage, based on the Rice-Golomb codes,[12] where the parameters of the code for the current index are adapted based on already coded indexes.

2.2 Pseudo-Random Bit Generator

The proposed joint encryption and compression algorithm requires a source of random bits. Ideally, one would be able to have at their disposal a sequence of truly random bits. Since this is not feasible, algorithms were developed to output pseudo-random bit sequences that can be considered unpredictable even if previously generated bits are disclosed, short of a very huge (close to infinite) computational power. In this work, we assume that the complexity of randomizing the compression algorithm is negligible with respect to the toll of the algorithm itself; in such fashion, the most computationally intensive encryption part consists in generating pseudo-random bits.

The pseudo-random generator we use is the Keccak cryptographic function,[13] a hash function that can be deployed as a primitive for cryptographic algorithms. The function relies on the so-called sponge construction, which consists of an *absorption* and a *squeezing* phase that interact with the function's internal state. In the absorbing phase, input data is XORed into a subset of the state that is then modified via a permutation function *f*. In the squeezing phase, the same state subset is read as to output the desider pseudo-random bit sequence, and the state is then updated by applying the f permutation. The fraction of the state that is iteratively modified and read is called *rate* (r), while the remaining part that is not affected by the absorption phase, i.e. the hidden one, is called *capacity* (c). In particular, the maximum security level of the Keccak function is $2^{\frac{c}{2}}$, which is reasonably high for suitable values of c.

3. PROPOSED FRAMEWORK

The proposed solution considers a low-complexity encryption framework in which a given part of the CCSDS compression algorithm is randomized so that encryption can be embedded into the compression stage, avoiding full encryption of the transmitted packets at lower layers in the transmission protocol. In the following, we evaluate different randomization targets and show which one of them is the most suitable. In general, this algorithm is inspired by solutions already employed in video encryption to reduce complexity and achieve high data rates, such as in[7,14] where a trade-off between security and complexity is achieved.

In order to combine compression and encryption, different parts of the compression algorithm can be randomized via a reliable source of randomness, i.e. a pseudo-random bit generator. According to the scheme in Fig. 1, randomization can be applied to data at different stages of the pipeline, and in particular on input samples, prediction residuals signs, and Rice-Golomb entropy coding. However, randomizing input samples is a solution that we will not consider since randomization is expected to increase the entropy of the samples and make them less predictable, leading to an increase of the rate of the encoder. Hence, in the following, we will consider the randomization of the signs of the prediction residuals and the entropy coding.

3.1 Requirements

The encryption algorithm should follow the Kerckhoffs' principle, which requires the system to be secure even if everything about the algorithm is known, except for the secret key. Also, when combining encryption and compression, the encryption algorithm should be rate-preserving, i.e. there should be no differences in compression rate with respect to the no-encryption case. Possibly, the algorithm should be also format-compliant, meaning that a CCSDS decoder should be able to decode the encrypted bitstream without errors even when the encryption key is unknown.

Considering the CCSDS adaptive entropy coding strategy, it is generally not possible to obtain an encryption algorithm that is both format-compliant and rate-preserving. Format compliance requires that the context of the encoder should be perfectly replicated at the decoder. If the encryption process modifies the context of the decoder, the same modification should be replicated at the encoder. Hence, the encoder will make sub-optimal choices due to the changed context, leading to increased rate. Conversely, if the encoder keeps using the right context hence preserving the rate, format compliance is lost due to the loss of synchronization between encoder and decoder. As a general consideration, however, one could assume that format compliance would most probably be not strictly necessary for space applications.

3.2 Adopted Solution

We outline three possible configurations for the the proposed joint compression / encryption algorithms: **(i) randomization of the signs of the prediction residuals**: this straightforward solution offers the advantage of being very easily implemented and also format-compliant; however, since CCSDS uses an optimized mapping of residuals depending on their sign, exact rate preservation is not guaranteed; **(ii) randomization of the fixed part of the Rice-Golomb coded mapped quantized residuals**: this approach consists in outputting a random bit sequence of k bits which is then XORed with the last k bits of the Rice-Golomb code for the mapped quantized residuals. The main advantage in this case is that the encoder rate is preserved; however, format compliance is lost, since the decoder observes a different context and may estimate wrong coding parameters; **(iii) combined randomization**: it is also possible to jointly implement the previous approaches at the same

time; however, in order to keep the encoder and the decoder perfectly synchronized, two different pseudo-random bit streams would be required, hence causing a doubling of the computational overhead needed to deploy this solution.

All the above solutions require that a common source of pseudo-random bits is available at the encoder and the decoder. Moreover, even if part of this pseudo-random sequence is disclosed, it should not be possible to recover the remaining bits. This can be easily provided using a standard stream cipher based on a pre-shared secret key and a random initialization vector that is generated at the encoder and transmitted to the decoder. For example, AES in counter mode offers a viable implementation.[15] It is worth noting that data encapsulation must be implemented in order to be able to include the initialization vector in the compressed and encrypted file and operate this mechanism; however, the extra effort required for implementation is negligible.

4. SECURITY ANALYSIS

A rigorous evaluation of the security of selective encryption techniques is often difficult. Although selective encryption provides a sort of visual protection, making it hard for an attacker to reconstruct a high quality version of the media,[16,17] its confidentiality is often disputed.[18,19]

Concerning random sign flipping of prediction residuals, under the assumption that the predictor is optimal and residuals are independent, every sign pattern would be equally likely. Hence, this scheme would achieve information theoretic security since an adversary would have to choose among an exponential number of equally likely possible solutions. In reality, prediction residuals are not truly independent, so it is reasonable to assume that an adversary has only a small set of plausible solutions to choose from. Here, the question is whether there is a computationally efficient procedure to select the correct solution.

In the case of a fixed linear predictor, breaking sign flipping is close to solving a phase recovery problem from magnitude measurements.[20] Since the acquired images are likely sparse in some domain, sparse phase recovery algorithms based on a convex relaxation can be used to recover the original image in polynomial time with high probability,[21] as suggested by similar results for transform coding.[18] However, in the case of CCSDS the predictor is adaptive, so the above solution cannot be directly applied. Our conjecture is that breaking sign flipping of prediction residuals in CCSDS compression requires much more complex algorithms, close to enumerating all possible solutions. Further research is needed in order to verify this assumption.

Concerning randomization of Rice-Golomb codes, due to the mapping used in CCSDS compression algorithm this is equivalent to randomizing the sign of residual and adding a perturbation when $k > 0$. Here, the problem is that for low rates the adaptive entropy encoder often enforces $k = 0$, making it less secure that simple sign randomization.

5. EXPERIMENTAL EVALUATION

In this section, we report experimental results obtained when applying our joint compression / encryption solution. As stated before, the algorithm requires a source of random bits. The pseudo-random generator we chose to work with is that described in[22] and based on the very efficient Keccak cryptographic function,[23] which has recently won the SHA-3 standardization competition.[24]

We employed four evaluation metrics in order to be able to compare the (i), (ii), and (iii) approaches as presented in the previous section, and hence choose the most suitable configuration. The metrics are the following: MAD: maximum absolute difference over the whole image between a pixel value in the original image and the corresponding reconstructed value; $MAD = 0$ refers the lossless compression case; SNR *(dB)*: signal-to-noise Ratio computed as the energy of the image over the energy of the compression error. We compute SNR for a *key-oblivious decoder*, e.g., a decoder that does not have the key to decipher the encrypted bitstream. SNR provides a measure of the security of the encryption system with respect to a passive observer which intercept the compressed data but does not put any additional decoding effort; **Rate Increase (RI, %)**: the rate increase with respect to the rate of the baseline with no encryption algorithm applied; **Key-stream Rate (KSR, %)**: the measure of the amount of random bits one needs to generate over the total number of encoded bits; this metric indicates how better the algorithm performs with respect to a stream cipher XORing each bit of the compressed file (100% key-stream rate);

Table 1. Average performance over the 9-image dataset; rate is measured in *bits per sample*.

AVG	MAD				
	0	1	2	4	8
	(i) residuals sign randomization				
RATE	5.02	3.53	2.92	2.30	1.76
SNR	-14.15	-14.23	-14.17	-14.09	-14.10
RI	0.47	0.63	0.57	0.44	0.31
KSR	18.99	22.09	22.38	21.62	19.01
	(ii) rice-golomb fixed part randomization				
RATE	5.01	3.52	2.91	2.29	1.76
SNR	-13.78	-13.83	-13.83	-13.94	-13.83
KSR	57.41	38.14	28.96	17.97	5.93
	(iii): (i) and (ii) combined				
RATE	5.02	3.53	2.92	2.30	1.76
SNR	-13.86	-13.83	-13.79	-13.91	-13.82
RI	0.47	0.63	0.57	0.44	0.31
KSR	76.40	60.23	51.34	39.60	24.94

The experiments have been performed on a few images from the test set employed to assess the CCSDS compression algorithm, which comprises the *agriculture*, *airs_gran9*, *aviris_sc0*, *aviris_sc3*, *aviris_sc10*, *aviris_sc11*, *aviris_sc18*, *hawaii_sc01*, and *IASI_desert* images.[25]

5.1 Encryption Techniques Comparison

Table 1 reports the average performance over the 9-image dataset coupled with the three proposed randomization techniques. The metrics are evaluated for MAD values in $[0, 1, 2, 4, 8]$, corresponding to a lossless and near-lossless scenario, which is the required target for reliable satellite imaging applications.

As mentioned, (i) is format-compliant but not rate-preserving. However, the average rate increase is always lower than 1%, and therefore can be considered negligible. On the contrary, (ii) is a rate-preserving technique by design so the rate increase is always 0% and has not been reported in the table. The SNR of the key-oblivious decoder is comparable for both cases and is much below 0 *dB* in all cases, indicating that the image reconstructed without knowledge of the encryption key does not convey useful information. The key-stream rate, instead, is on average much higher for (ii), because a greater amount of random bits is required in order to scramble the fixed part of the Rice-Golomb code. The (i)+(ii) configuration offers Rate increase and Key-stream rate values which can be roughly estimated as the sum of the ones for (i) and (ii) ones, at the expense of a double overhead necessary to manage two different random bits generators, without any SNR advantage. From Table 1, the preferable solution is the residuals sign randomization (i), which comes with lower resource requirements, a negligible increase in the rate and a lower Key-stream rate. In the following, we will consider (i) to be the adopted encryption solution.

5.2 Residuals Sign Randomization Encryption

Table 2 reports the performance in terms of rate increase and key-stream rate for each single image in the test dataset. It can be observed once more that, except for the *hawaii_sc01* image, the rate increase is always lower than 1%. As per the RI and KSR, they are largely dependent on the visual content of the image to be compressed and encrypted, so it may be possible that these two metrics experience different trends according to each different case.

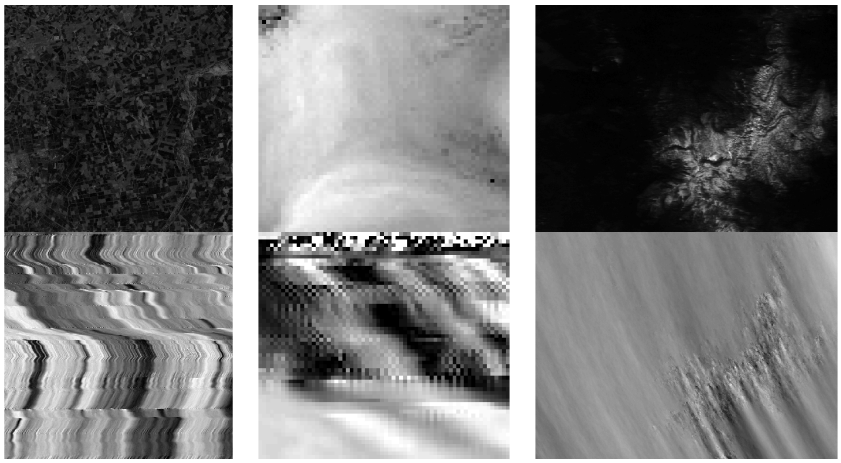
About the security of the algorithm, is important to evaluate the quality of an image decoded by a key-oblivious decoder. In Fig. 2, we reported three qualitative examples of images that have been reconstructed without knowledge of the encryption key. Fig. 2 shows that obtained images are meaningless and therefore they cannot be used to infer useful information. It is also interesting to observe that, in Fig. 2 **(c)**, even if the key-oblivious decoder is not able to infer meaningful data, in the presence of very high discontinuities in the original image some structure can be preseved in the decoded data.

Table 2. Performance detail over the 9-image dataset for the residuals sign randomization.

AVG	MAD				
	0	1	2	4	8
	agriculture				
RI	0.89	1.45	1.25	0.77	0.35
KSR	22.59	24.81	22.64	16.65	9.59
	airs_gran9				
RI	0.30	0.83	0.90	0.79	0.48
KSR	21.12	25.76	26.26	22.82	14.53
	aviris_sc0				
RI	0.02	0.07	0.13	0.20	0.23
KSR	15.67	19.75	21.90	24.13	24.97
	aviris_sc3				
RI	0.04	0.18	0.28	0.36	0.34
KSR	17.27	21.85	23.90	25.45	23.53
	aviris_sc10				
RI	0.02	0.11	0.21	0.28	0.30
KSR	16.78	21.31	23.51	25.48	25.17
	aviris_sc11				
RI	0.02	0.08	0.14	0.21	0.22
KSR	15.98	20.06	22.10	24.09	24.52
	aviris_sc18				
RI	0.02	0.09	0.015	0.22	0.25
KSR	16.46	20.75	22.91	25.01	25.40
	hawaii_sc01				
RI	2.62	2.38	1.54	0.65	0.24
KSR	26.08	21.74	14.32	7.05	3.41
	IASI_desert				
RI	0.34	0.50	0.51	0.49	0.36
KSR	18.94	22.75	23.88	23.94	19.95

6. CONCLUSIONS

We have presented different randomization strategies that can be used to selectively encrypt parts of the CCSDS image compression pipeline. We concentrated on solutions that approximately preserve the rate of the encoder, possibly preserving the format compliance. Due to the adaptive nature of the CCSDS algorithm, achieving both rate preservation and format compliance is in general not possible. However, our experimental results show that even a straightforward solution such as sign randomization of the prediction residuals can be very effective in order to embed encryption into the CCSDS compression pipeline, with a negligible rate increase.



(a) **(b)** **(c)**

Figure 2. Examples of images reconstructed by a key-oblivious decoder (bottom row) against the original images (top row). (a) *agriculture* (b)*iasi_desert* (c)*aviris_18*

Moreover, the proposed solution has a complexity significantly lower than applying a stream cipher to the compressed bit stream. A preliminary security analysis shows that the cost of performing image restoration without knowledge of the encryption key should be sufficiently high, since standard optimization techniques cannot be directly applied. However, further research is needed to verify this assumption in the presence of active adversaries.

ACKNOWLEDGMENTS

The research leading to this publication has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776311.

REFERENCES

[1] [*Space Data Link Security Protocol*], no. 355.0-B-1 in Recommendation for Space Data System Standard, Blue Book, Issue 1, Consultative Committee for Space Data Systems (CCSDS), Washington, DC, USA (Sept. 2015).

[2] [*CCSDS Cryptographic Algorithms*], no. 352.0-B-2 in Recommendation for Space Data System Standard, Blue Book, Issue 2, Consultative Committee for Space Data Systems (CCSDS), Washington, DC, USA (Aug. 2019).

[3] "MetOp operations." https://www.esa.int/Our_Activities/Observing_the_Earth/Meteorological_missions/MetOp/Operations. Accessed: 2019-10-08.

[4] "Sentinel-2 resolution and swath." https://sentinel.esa.int/web/sentinel/missions/sentinel-2/instrument-payload/resolution-and-swath. Accessed: 2019-10-08.

[5] Kerr, M., Cornara, S., Latorre, A., Tonetti, S., et al., "EO-ALERT: Next generation satellite processing chain for rapid civil alerts," in [*6th Int. Workshop on On-Board Payload Data Compression*], 1–10 (September 2018).

[6] [*Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression*], no. 123.0-B-2 in Recommendation for Space Data System Standard, Blue Book, Issue 2, Consultative Committee for Space Data Systems (CCSDS), Washington, DC, USA (Feb. 2019).

[7] Stutz, T. and Uhl, A., "A survey of H.264 AVC/SVC encryption," *IEEE Trans. Circuits Syst. Video Technol.* **22**, 325–339 (March 2012).

[8] Van Wallendael, G., Boho, A., De Cock, J., Munteanu, A., and Van De Walle, R., "Encryption for high efficiency video coding with video adaptation capabilities," *IEEE Trans. on Consumer Electronics* (2013).

[9] Wang, Y., O'Neill, M., and Kurugollu, F., "A tunable encryption scheme and analysis of fast selective encryption for CAVLC and CABAC in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.* (2013).

[10] Hofbauer, H., Uhl, A., and Unterweger, A., "Transparent encryption for HEVC using bit-stream-based selective coefficient sign encryption," in [*2014 IEEE Int. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*], (2014).

[11] Boyadjis, B., Bergeron, C., Pesquet-Popescu, B., and Dufaux, F., "Extended selective encryption of H.264/AVC (CABAC)- and HEVC-encoded video streams," *IEEE Trans. Circuits Syst. Video Technol.* (2017).

[12] Kiely, A., "Selecting the Golomb parameter in Rice coding," *IPN progress report* **42**, 159 (2004).

[13] Bertoni, G., Daemen, J., Peeters, M., and Van Assche, G., "The keccak sha-3 submission, january 2011."

[14] Liu, F. and Koenig, H., "A survey of video encryption algorithms," *Computers & security* (2010).

[15] Dworkin, M. J., "SP 800-38A 2001 edition. recommendation for block cipher modes of operation: Methods and techniques," tech. rep., Gaithersburg, MD, United States (2001).

[16] Asghar, M. N., Ghanbari, M., Fleury, M., and Reed, M. J., "Confidentiality of a selectively encrypted h.264 coded video bit-stream," *Journal of Visual Commun. and Image Repr.* **25**(2), 487 – 498 (2014).

[17] Fezza, S. A., Hamidouche, W., Kamraoui, R. A., and Déforges, O., "Visual security assessment of selective video encryption," in [*2019 11th Int. Conf. on Quality of Multimedia Exp. (QoMEX)*], 1–3 (June 2019).

[18] Amir Said, "Measuring the strength of partial encryption schemes," in [*IEEE Int. Conf. on Image Processing 2005*], (Sep. 2005).

[19] Hofbauer, H. and Uhl, A., "Identifying deficits of visual security metrics for images," *Signal Processing: Image Communication* **46**, 60 – 75 (2016).

[20] Candès, E. J., Strohmer, T., and Voroninski, V., "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on Pure and Applied Mathematics* **66**(8), 1241–1274 (2013).

[21] Li, X. and Voroninski, V., "Sparse signal recovery from quadratic measurements via convex programming," *SIAM Journal on Mathematical Analysis* **45**(5), 3019–3033 (2013).

[22] Bertoni, G., Daemen, J., Peeters, M., and Van Assche, G., "Duplexing the sponge: Single-pass authenticated encryption and other applications," in [*Selected Areas in Cryptography*], Miri, A. and Vaudenay, S., eds., 320–337, Springer Berlin Heidelberg (2012).

[23] Bertoni, G., Daemen, J., Peeters, M., and Van Assche, G., "Keccak sponge function family main document," *Submission to NIST (Round 2)* (2009).

[24] Dworkin, M. J., "SHA-3 standard: Permutation-based hash and extendable-output functions," tech. rep. (2015).

[25] Auge, E., Santalo, J., Blanes, I., Serra-Sagrista, J., Kiely, A., et al., "Review and implementation of the emerging CCSDS recommended standard for multispectral and hyperspectral lossless image coding," in [*2011 First Int. Conf. on Data Compression, Communications and Processing*], (2011).