SECURITY APPLICATIONS OF DISTRIBUTED ARITHMETIC CODING

Marco Grangetto†, Enrico Magli, Gabriella Olmo**

* Dip. di Elettronica
Politecnico di Torino - Italy
enrico.magli@polito.it

†Dip. di Informatica
Università degli Studi di Torino - Italy
marco.grangetto@di.unito.it

ABSTRACT

Distributed arithmetic coding (DAC) has recently been proposed for compression in the Slepian-Wolf setting. With respect to syndrome coding, DAC allows to easily adapt to nonstationary statistics of the signal to be coded, and works well for short and medium block lengths. In this paper we develop security applications of DAC in the field of biometric authentication. We show that DAC can be used for authentication by employing the codeword as secure hash, and using the outcome of DAC decoding for authentication. Moreover, we introduce a second powerful security feature, namely the randomization of the DAC intervals. This allows to protect the hash from attacks. We assess the authentication performance of the proposed scheme with respect to template matching and turbo codes.

1. INTRODUCTION

Many authentication and access control systems employ biometrics such as fingerprints, irises, and faces. These have several advantages over passwords. Being linked to the user, they cannot be lost, forgotten or given away. They are more secure than passwords, as they cannot be poorly chosen [1]. However, biometric authentication systems pose significant security problems. In short, authentication works by taking a biometric reading from the user, and comparing it with an enrollment biometric stored on the device. If they are close enough, then access is granted. The original biometric has to be stored on the authentication device. This poses a significant security threat, as an attacker that gains physical access to the device then also gains access to the original biometric.

Several authors have investigated the security of biometric authentication schemes. Schemes based on encryption have been proposed [2]. The use of error correcting codes was first proposed in [3]. Recently, the connections between biometrics and Slepian-Wolf (S-W) coding have been pointed out [4].

S-W coding refers to coding of correlated sources X and Y. It has been shown [5] that separate encoding coupled with joint decoding (a.k.a. distributed source coding) is optimal, in that a sum rate equal to the joint entropy $H(X,Y)$ is sufficient to guarantee that the decoder error probability is vanishingly small as the block size goes to infinity. The correlation can be described through a "virtual correlation channel", i.e. $Y = X + N$, with N a noise process. Coding with side information refers to the case that the decoder knows Y exactly, and the encoder transmits a description of X at rate at least equal to $H(X|Y)$. In this case, the receiver observes the channel output and has to estimate the input X from the corrupted observations. The codeword of X is obtained as the parity bits of X obtained using a good channel code or, even better, as the syndrome of X using the channel code. Several practical algorithms have been developed using low-density parity-check codes (LDPC) [6] and turbo codes [7].

The connection between S-W coding and biometrics can be explained in terms of the virtual correlation channel, i.e., $Y = X + N$. The side information Y is taken as the biometric reading provided by the user, while X is the original biometric stored on the device. The conditional entropy $H(X|Y)$ determines the average degree of "difference" between X and Y. Under this paradigm, X is stored on the device using a S-W codeword with rate R, which determines how much "difference" between the original and the reading can be tolerated to successfully authenticate the user. This yields two advantages. First, X is stored in compressed format, requiring less space. Second and more important, the S-W codeword does not represent a complete description of X, and hence cannot be decoded individually. The authentication process attempts to decode X using the reading Y as side information. Ideally, only if the reading is authentic the system is able to decode X. This means that, if the system is designed properly, only authorized users may have access to the original biometric X.

In [8] this concept is applied to image authentication. In [9] LDPC codes are applied to fingerprint biometrics, and in [10] they are used for gait authentication. Recently, a new approach has been proposed, which extends arithmetic coding to the S-W setting, and is termed distributed arithmetic coding (DAC) [11, 12]. DAC has better performance than turbo- and LDPC-based designs at short and medium block lengths, and allows to easily account for nonstationary probability distributions of the data. DAC is a potentially good match to biometric applications, in which the data blocks to be compressed are not necessarily large, e.g., 792 samples in [10].

S-W coding indeed improves the security of a biometric authentication scheme, as an attacker that gains possession of the device can not have direct access to the user hashes. However, coding of the templates can not prevent an attacker from carrying out attacks based on decoding the hashes with purposely designed biometric readings, in order to obtain more information on the hashes or on the system. E.g., given enough time, an attacker could try a brute force approach by attempting to decode an arbitrarily high number of readings until a positive verification is obtained.

In this paper we propose a secure biometric authentication scheme using S-W coding, which addresses the problems mentioned above. In particular, we leverage on DAC to develop a general-purpose secure authentication scheme. We model the feature vectors as binary strings, and describe a practical procedure to acquire these feature vectors dur-

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n. 216715 (NEWCOM++)

 ISSN 2076-1465

ing the enrollment stage, in such a way that all information needed for S-W decoding, particularly the marginal and conditional source probability distributions, are made available to the verification stage. The use of S-W coding provides a first degree of securization of the user templates. However, we also employ interval randomization as in [13] to define a randomized DAC (RDAC) that can secure the template from *any* unauthorized use. Therefore, besides hiding the templates, the proposed scheme also ensures that no malicious use can be made of these templates.

2. RANDOMIZED DAC

2.1 Review of DAC

Let $X = [X_1, \ldots, X_i, \ldots X_N]$ be a binary memoryless source with probabilities $p_0 = P(X_i = 0)$ and $p_1 = 1 - p_0$. The classical binary arithmetic coding process for X uses the probabilities p_0 and p_1 to partition the $[0,1)$ interval into subintervals associated to possible occurrences of the input symbols. At initialization the current interval is set to $I_0 = [0,1)$. For each input symbol, the current interval I_i is partitioned into two adjacent sub-intervals of lengths $p_0|I_i|$ and $p_1|I_i|$, where $|I_i|$ is the length of I_i. The sub-interval corresponding to the actual value of X_i is selected as the next current interval I_{i+1}, and this procedure is repeated for the next symbol. After all N symbols have been processed, the sequence is represented by the final interval I_N. The codeword C_X can consist in the binary representation of any number inside I_N.

DAC is based on the principle of inserting some ambiguity in the source description during the encoding process. This is obtained employing a set of intervals whose lengths are proportional to the modified probabilities $\widetilde{p}_0 \geq p_0$ and $\widetilde{p}_1 \geq p_1$. In order to fit the enlarged sub-intervals into the $[0,1)$ interval, they are allowed to partially overlap. The encoding procedure is exactly the same as for arithmetic coding, but employs the modified probabilities. The codeword C_X is shorter, i.e., the bit-rate is smaller, so much so as the interval overlap is large. The amount of overlap is a parameter that can be selected so as to achieve the desired rate, which should be no less than $H(X|Y)$. Decoding employs a correlated side information sequence, and is carried out using a sequential decoding algorithm (see [12] for details).

2.2 Randomization of DAC

There has recently been a lot of interest in secure arithmetic coding schemes. In [13] a randomized arithmetic coder has been proposed. The basic idea is that the encoding is done as in Sect. 2.1, but the intervals are randomly swapped at each input bit based on a pseudorandom sequence. Specifically, a key is used to initialize a pseudorandom number generator. For each input bit X_i, a number $R_i \in \{0,1\}$ is drawn with probability $P(R_i = 0) = 0.5$. For the encoding of X_i, if $R_i = 0$ then the convention that the interval of the most significant symbol precedes the interval of the least significant symbol is used, and vice versa. Given knowledge of the key, the normal arithmetic decoder can initialize the pseudorandom generator and track the interval orderings used at the encoder. Conversely, without knowledge of the key, it is not possible to decode the input string, as the loss of synchronization will produce an output sequence with high distance from the correct sequence.

This concept can also be applied to DAC in order to increase its security at no extra computational cost. In particular, we obtain RDAC as described before, swapping (or not swapping) the intervals based on a pseudorandom number R_i. The interval overlap is of no concern, as the total amount of overlap remains the same, while only the interval positions are changed. Although the sequential decoder is very different from an arithmetic decoder, it can be easily adjusted to the randomized case. The decoder also initializes a pseudorandom generator and produces R_i. The decoder is bit-synchronous, and at each stage it uses the intervals, with the correct ordering yielded by the value of R_i, to test for the two possible values of the input source symbol.

It should be noted that the concept of randomized arithmetic coder has been extended to the case of "interval splitting", in which the unit interval is partitioned into subintervals, and the subintervals are associated to the input values such that the subinterval lengths match the input probability distribution [14]. This technique could also be applied to RDAC.

3. PROPOSED SCHEME

3.1 System model

We assume the following reference system model. We refer to the object that handles all enrollment and verification procedures as the "device".

Enrollment. When a user wants to enroll in the system, she provides a biometric reading. From the reading, the device computes a feature vector, that is, a sequence of numbers that capture most information about the reading, and can be directly used to verify a new reading against the template reading.

Authentication. When a user that has been previously enrolled wants to be verified, she provides a new biometric reading. The device computes from the reading the corresponding feature vector. Subsequently, it compares the new feature vector with the templates of all users available in the database. If a template is found that is "sufficiently" similar to the new template, then the user is positively authenticated.

It should be noted that the biometric reading and the feature vector are quite different. The reading depends on the biometric measurement device. E.g., for a fingerprinting system, the reading would typically be an image of the fingerprint. However, the image cannot be used for verification directly, for several reasons. Imperfections in the acquisition process, e.g. misalignment with respect to a reference image or illumination changes between the reference and the new template, would easily lead to rejection of an authorized user. Therefore, it is more convenient to extract from the image a set of numbers (the feature vector) that exhibit features such as rotation and illumination invariance, and make decisions on the feature vectors rather than on the original readings. The template is usually a relatively short string (e.g., 512 samples in [15]), making a good case for the use of RDAC, which is known to outperform other channel codes (turbo and LDPC codes) at these block lengths. In this paper we purposely keep the system model rather general, so that it can be applied to several biometric systems (e.g., fingerprinting, iris recognition, and so on). Obviously, every specific system will use a different biometric reader and a specific algorithm to produce the template. Moreover, every template will need tailoring of the RDAC to its specific characteristics, e.g. block lengths, statistical distributions, binarization of templates belonging to M-ary alphabets with

$M > 2$. Nevertheless, the present study models the RDAC performance in a general but still realistic context, and analyzes the performance loss with respect to the ideal (but not secure) reference algorithm based on template matching.

In practice, our model works as follows. During the enrollment stage, the user provides a first biometric reading, from which a feature vector X is obtained. For authentication, the user provides a new reading from which the corresponding feature vector Y is computed. We assume that X and Y are binary strings of length N, and that $P(X_i = 0) = p_0$. We denote as $\|X,Y\|_H$ the Hamming distance between X and Y, i.e. the number of positions in which X and Y differ. If the feature vector is not binary, both the template matching and RDAC algorithms can be modified to accommodate larger alphabet sizes, as described later on. We also assume that the relation between the enrollment and the authentication feature vectors can be described through a binary symmetric channel (BSC) model. Specifically, Y is obtained as the output of a BSC with crossover probability q_D, with X as input. We also consider the case that a non-authorized user maliciously attempts to make herself authorized by the system. This user provides a reading that is mapped to a feature vector Z, and we model Z as the output of a BSC with crossover probability q_F, with X as input. Clearly, two feature vectors of the same user will have small Hamming distance, hence q_D will be small. Conversely, the feature vector of a non-authorized user will be rather distant from that of any authorized user; hence, q_F will be relatively large.

3.2 Template matching

The classical authentication setting can be cast as a hypothesis testing problem. The two hypotheses are that "the user is legitimate" or "the user is not legitimate". Verification of these hypotheses is performed comparing the reading Y to the templates X of all users in the database. If there is at least one template X^* that is "sufficiently close" to Y, then the user is declared legitimate. In particular, if X^* is such that $\|X^*,Y\|_H \leq \|X,Y\|_H \ \forall X \neq X^*$, then the decision statistics for this problem can be written as

$$\|X^*,Y\|_H \lessgtr T \tag{1}$$

where T determines the trade-off between the probability of detection P_D, i.e. the probability of positively authenticating an enrolled user, and the probability of false alarm P_F, i.e. the probability of positively authenticating a non-authorized user.

Given the system model described above, and specifically the use of a BSC to model the differences between feature vectors, the Hamming distance between two feature vectors follows a Binomial distribution:

$$f(k|n,p) = \binom{n}{k} p^k (1-p)^{n-k}$$

where n is the total number of random drawings and p is the success probability. In the case of authentication of an enrolled user, detection occurs if the BSC introduces a number of errors not greater than T over the BSC channel with crossover probability q_D. Therefore the probability of detection over a block of length N can be written as

$$P_D = \sum_{i=0}^{T} \binom{N}{i} q_D^i (1-q_D)^{N-i}. \tag{2}$$

Similarly, a false alarm occurs when the BSC introduces a number of errors not greater than T over the BSC channel with crossover probability q_F. The false alarm probability can be written as

$$P_F = \sum_{i=0}^{T} \binom{N}{i} q_F^i (1-q_F)^{N-i}. \tag{3}$$

The two equations above allow to derive the receiver operating characteristic [16] that determines the overall performance of the authentication system. Parameters q_D and q_F are application-specific, as they depend on the ability of the selected feature vectors to discriminate between different users. It is worth noting that template matching can be easily extended to the case of nonbinary feature vectors, using the Euclidean or weighted Euclidean distance instead of the Hamming distance metric.

3.3 RDAC authentication algorithm

For the RDAC scheme, and in general S-W based authentication schemes, the operation is different from the template matching case. The main differences are the following.

- Every user that wishes to enroll or authenticate herself in the system provides a key that will be used by the RDAC to initialize the pseudorandom generator used for interval swapping. In this paper we do not consider specific key management issues; we simply note that, without the key, correct decoding of a S-W codeword will be impossible.
- During the enrollment stage, the feature vector X is not stored on the device "as is". Lossless storage of the feature vector would require at least $H(X)$ bits per sample (bps). Instead, we store on the device a codeword C_X obtained applying to X a RDAC at the selected rate of R bps. Moreover, a cyclic redundancy check (CRC) codeword is also stored along with C_X. This enables the decoder to make a decision on the authentication of a given biometric reading. This is explained in more detail in the following.
- During the enrollment stage, a second biometric reading X' of the same user is acquired. It will be used to compute statistics between X and X', as will be explained in the following.
- During the authentication stage, the feature vector Y obtained from the new biometric reading is used as "side information" in the decoding of C_X. The decoding process yields a decoded template $\widehat{X}$. If X and $\widehat{X}$ are identical, i.e. the CRC codeword computed on $\widehat{X}$ is identical to that stored along with C_X, then the user is authenticated, otherwise she is rejected.

The selected rate R is very important, as it has the same function as the detection threshold T. If the rate is very high, P_D will be very close to one, as also feature vectors that are not too similar to X will be "sufficiently good" side information signals. Indeed, as an extreme case, when the rate R is larger than $H(X)$, C_X contains all information about X, and the decoder will positively authenticate all users, either enrolled or non-authorized. On the contrary, if R is very small, the probability of detection may be somewhat small because if Y is not extremely close to X the decoder will declare an error; however, the false alarm rate will also be very small. Thus, a proper selection of R is critical in order to achieve the desired trade-off between P_D and P_F.

The binary decision is enabled through the CRC codeword. The CRC does not guarantee that X and $\widehat{X}$ are identical if their hashes are also identical; however, the error probability of the CRC can be made arbitrarily small (e.g., the worst case error probability is equal to 2^{-32} for a 32-bit CRC.

It should be noted that S-W coding requires knowledge of the source probability p_0 and the amount of correlation between X and Y. These parameters can be used to select the rate R, and are needed by the S-W decoder to construct a good source and channel model to be used for the soft decoding of RDAC, as well as turbo and LDPC codes. This has often been a problem in previous designs based on S-W coding. In this paper we use an innovative approach. In order to facilitate statistical modeling, we modify the enrollment stage in such a way that, when a new user is enrolled in the system, not one but two biometric readings (X and X') are acquired. This requirement is not a limitation in that, as a matter of fact, most existing systems acquire multiple fingerprints in order to cope with bad acquisitions. In our system, the multiple acquisition has an additional use: it allows to compute and compare two correlated feature vectors of the same user, and hence to compute marginal and conditional statistics between the pair of correlated sources, and particularly the prior probability p_0 and the crossover probability q_D that are needed by the decoder. Then, once the statistics have been computed, the second reading X' is erased, and only the first one is coded using a RDAC. The statistics are stored along with the coded template C_X and the CRC codeword, and subsequently used by the decoder. Since typical biometric authentication systems acquire several readings from the same user, the RDAC scheme may also benefit from more than two measurements, making the estimation of p_0 and q_D even more accurate.

The RDAC scheme can be easily extended to the case of nonbinary feature vectors, applying the DAC to the bit-planes (or other binarization) of X.

4. EXPERIMENTAL RESULTS

We have tested the RDAC scheme in order to derive its receiver operating characteristic (ROC). Note that the ROC refers to the case that the device is used "properly", i.e., it has not been stolen or otherwise manipulated. In other terms, the ROC only measures the ability of the biometric authentication system to discriminate the feature vectors of authorized and unauthorized users.

To derive the ROC, the following simulation has been made. We consider feature vectors of size $N = 200$ and $N = 1000$, which are typical of practical biometric authentication systems. For each vector length, we also consider feature vectors with symmetric ($p_0 = 0.5$) and skewed ($p_0 = 0.9$) probability distribution. For $p_0 = 0.5$, we take $q_D = 2.154 \cdot 10^{-2}$ and $q_F = 9.41 \cdot 10^{-2}$. These values correspond to a conditional entropy $H(X|Y)$ approximately equal to 0.15 for the verification of an authorized user, and 0.45 for an unauthorized user. A point of the ROC curve is obtained setting a given rate for C_X; we consider rates from 0.6 bps to 0.35 bps. For $p_0 = 0.9$ we take $q_D = 2.737 \cdot 10^{-2}$ and $q_F = 1.038 \cdot 10^{-1}$. If a block is not correctly decoded, this yields a negative authentication. Therefore, the probability of detection is equal to one minus the frame error rate of the RDAC decoder when the BSC crossover probability is equal to q_D, and the probability of false alarm is equal to one minus the frame error rate when the BSC crossover probability is equal to q_F. In every case, the results are averaged over a number of blocks such that at least 10^7 feature vector bits are simulated for every point of the ROC curve.

We compared the ROC curves of the DAC with that of template matching (TM) with the same block length, probability distributions and BSC crossover probabilities. We remark that template matching is optimal but *not* secure. Hence, we expect the ROC of TM to outperform that of RDAC, although a device using TM could be easily violated by an attacker if stolen or otherwise manipulated. Moreover, we also compare with an algorithm similar to RDAC, which employs turbo codes (TC) instead of the RDAC. In particular, the RDAC encoder and decoder are replaced by a punctured turbo code with rate-$\frac{1}{2}$ generator (31,27) octal (16 states), S-random interleavers, and 15 decoder iterations. For the skewed source, the turbo BCJR decoder has been modified by adding to the decoder metric the *a priori* term, as done in [7]; moreover, we also modify the metric in order to give large likelihood to the parity bits, which are not corrupted under the "virtual channel" model of S-W coding.

Fig. 1 shows the results for $N = 200$. As can be seen, TM consistently outperforms RDAC and TC. However, RDAC is extremely close in the case of $p_0 = 0.5$. For $p_0 = 0.9$, the performance of both algorithms is degraded, more seriously so for TC.

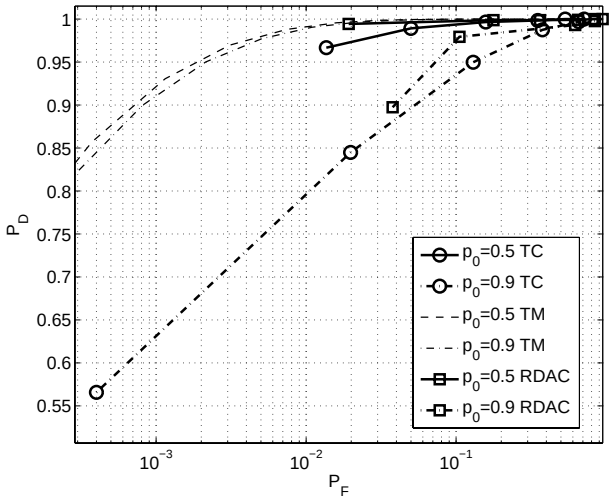

Figure 1: ROC curve for TM, RDAC and TC for $N = 200$.

For $N = 1000$, the results are shown in Fig. 2. In this case, although the best results are achieved by TM, RDAC and TC are extremely close. Note that the x-axis scale is different in Fig. 1 and 2. For $p_0 = 0.5$ the performance is almost identical to TM, and for $p_0 = 0.9$, P_D is still above 98% also for very small values of P_F. This demonstrates the suitability of RDAC for biometric authentication.

In case that the device is stolen or manipulated, TM provides no security. An attacker that gains access to the device will be able to read all feature vectors of all users, and will be able to impersonate any user creating fake biometric readings. The TC scheme avoids this using S-W coding. The attacker then will only gain access to the hash of every user. While the hash can not be used directly, the attacker

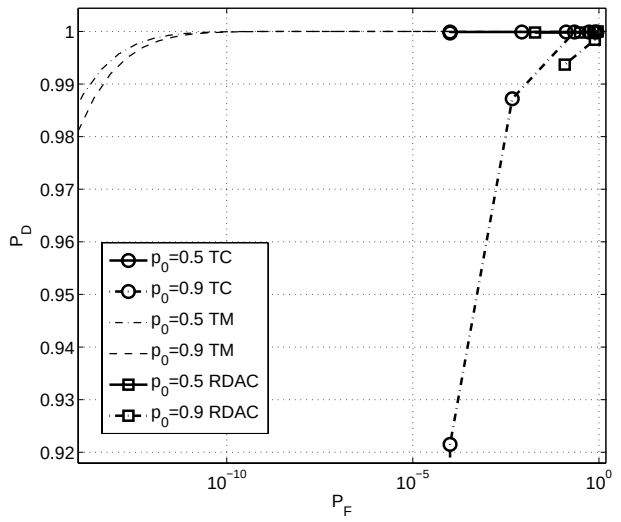

Figure 2: ROC curve for TM, RDAC and TC for $N = 1000$.

could still carry out a brute force or other attack, because the CRC will give the attack a way to understand whether a decoding attempt has been successful or not. On the other hand, RDAC provides a double level of security. It hides the hashes through S-W coding in much the same way as the TC. However, it also prevents an attacker from extracting more information from the device, as the built-in encryption due to the interval randomization leads to failure of virtually every decoding attempt. It should also be noted that certain attacks [14] that can be carried out on the randomized arithmetic coder cannot be applied to the RDAC due to the very different nature of the decoding process. While the randomized arithmetic coder is close to a standard arithmetic decoder, the RDAC decoder is a sequential search decoder aided by the side information. As a consequence, a state of the RDAC decoder cannot be forced easily, as it depends not only on the sequence of input bits, but also on the side information sequence, which is not available at the encoder and is not known to the attacker.

5. CONCLUSIONS

We have proposed a biometric authentication scheme based on RDAC. The proposed scheme provides a dual level of security, using S-W coding to store the feature vectors, and randomizing the encoding process to ensure unusability of the information on the device by a non-authorized user. Moreover, the proposed scheme uses multiple biometric readings to compute statistics needed during the authentication (S-W decoding) stage. The performance is close to that of template matching, showing that this scheme is very promising for biometric applications.

REFERENCES

[1] A.K. Jain, S. Pankanti, S. Prabhakar, L. Hong, and A. Ross, "Biometrics: a grand challenge," in *Proceedings of International Conference on Pattern Recognition*, 2004.

[2] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. of EUROCRYPT*, 2004.

[3] G.I. Davida, Y. Frankel, and B.J. Matt, "On enabling secure applications through off-line biometric identification," in *Proc. of IEEE Symposium on Security and Privacy*, 1998.

[4] E. Martinian, S. Yekhanin, and J.S. Yedidia, "Secure biometric via syndromes," in *Proc. of Allerton Conference*, 2005.

[5] D. Slepian and J.K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[6] A. Liveris, Z. Xiong, and C. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, Oct. 2002.

[7] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Communications Letters*, vol. 5, no. 10, pp. 417–419, Oct. 2001.

[8] Y.-C. Lin, D. Varodayan, and B. Girod, "Image authentication based on distributed source coding," in *Proc. of IEEE International Conference on Image Processing*, 2007.

[9] S.C. Draper, A. Khisti, E. Martinian, A. Vetro, and J.S. Yedidia, "Using distributed source coding to secure fingerprint biometrics," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007.

[10] S. Argyropoulos, D. Tzovaras, D. Ioannidis, and M.G. Strintzis, "Gait authentication using distributed source coding," in *Proc. of IEEE International Conference on Image Processing*, 2008.

[11] M. Grangetto, E. Magli, and G. Olmo, "Distributed arithmetic coding," *IEEE Communications Letters*, vol. 11, no. 11, pp. 883–885, Nov. 2007.

[12] M. Grangetto, E. Magli, and G. Olmo, "Distributed arithmetic coding for the Slepian-Wolf problem," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2245–2257, June 2009.

[13] M. Grangetto, E. Magli, and G. Olmo, "Multimedia selective encryption by means of randomized arithmetic coding," *IEEE Transactions on Multimedia*, vol. 9, no. 5, pp. 905–917, Oct. 2006.

[14] H. Kim, J. Wen, and J.D. Villasenor, "Secure arithmetic coding," *IEEE transactions on Signal Processing (to appear)*.

[15] A.K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerpring matching," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 846–859, May 2000.

[16] H.L. Van Trees, *Detection, Estimation and Modulation Theory. Part I*, Wiley, New York, 1968.