Approximate Message Authentication Codes for N-ary Alphabets

Renwei Ge, *Student Member, IEEE*, Gonzalo R. Arce, *Fellow, IEEE*, and Giovanni Di Crescenzo

***Abstract*—Approximate message authentication codes (AMACs) for binary alphabets have been introduced recently as noise-tolerant authenticators. Different from conventional "hard" message authentications that are designed to detect even the slightest changes in messages, AMACs are designed to tolerate a small amount of noise in messages for applications where slight noise is acceptable, such as in multimedia communications. Binary AMACs, however, have several limitations. First, they do not naturally deal with messages having N-ary alphabets ($N > 2$). AMACs are distance-preserving codes; i.e., the distance between two authentication tags reflects the distance between two messages. Binary representation of N-ary alphabets, however, may destroy the original distance information between N-ary messages. Second, binary AMACs lack a means to adjust authentication sensitivity. Different applications may require different sensitivities against noise. AMACs for N-ary alphabets are designed as a cryptographic primitive to overcome the limitations of binary AMACs. N-ary AMACs not only directly process messages having N-ary alphabets but also provide sensitivity control on the authentication of binary and of N-ary messages. The generalized N-ary AMAC algorithm and its probabilistic model are developed. A statistical analysis characterizing the behavior of N-ary AMACs is provided along with the simulations illustrating their properties. Security analysis under chosen message attack is also developed.**

***Index Terms*—Chosen message attack, hash function, message authentication code (MAC), multimedia message authentication.**

I. Introduction

MESSAGE authentication codes (MACs) are cryptographic primitives used extensively in the construction of security services for general digital data. Security services here include authentication, nonrepudiation, and integrity. Message digest schemes used in conventional MACs, such as keyed Hash MAC (HMAC) [1] with Message Digest algorithm 5 (MD5) or Secure Hash Algorithm (SHA), are "hard" [2], deliberately constructed to be as unforgiving as possible where modifying a single message bit would lead to a security check breakdown. These MACs fit those applications where the security requirement is to reject any message that has been altered to the slightest extent. Multimedia digital content, however, is continuously modified and manipulated as a result of compression and format conversion. In addition, data errors are frequent due to noise in communication channels and/or packet losses. In many multimedia applications, incidental noise or admissible modification, such as watermarking or format conversion, is acceptable as long as all-out forgeries and substantial modification of content can be identified. This scenario, incompatible with traditional "hard" cryptographic mechanisms, motivates the development of "soft" message authentication by which slight noise or acceptable modifications can be tolerated by the authenticator.

In the literature, there are several different approaches working on the authentication of multimedia messages, which can be basically grouped into two classes [3]. One class is to generate an authentication tag based on the extraction of the content or features from a message. The second class is to generate an authentication tag based on the modified message. Detailed review of different approaches can be found in Section II. The common point of these two classes is that both of them use traditional cryptographic primitives, such as HMAC or Digital Signature, as the core of authentication, only with a modified input.

Different from all these approaches, binary approximate message authentication codes (AMACs) [4]–[7] have been developed as a new cryptographic primitive for noise-tolerant security. It is a variant of MACs, whereby "certain," perhaps imperceptible, modifications in the message propagate to "minor" modifications in the authentication tag, and thus still retain security against other unacceptable modifications. The codes are probabilistic in nature and have the property of preserving distance; i.e., the probability of a given bit change in the authentication tag varies monotonically as a function of bit changes in the message. Such distance-preserving tags give users the freedom to decide the authentication boundaries among tags.

To date, AMACs have been restricted to the messages that are binary in nature. N-ary AMACs presented in this paper generalize the original binary construction of AMACs into the one that accepts messages with N-ary alphabets ($N > 2$). The first motivation is to provide a suitable means of dealing with information that does not tend naturally to be a binary representation such as graphics, multilevel and color halftones, and other signals such as biological DNA and protein sequences [8], [9]. Given an N-ary message, the N-ary representation is often converted into a binary one. However, AMACs are distance-preserving codes, which means the distance between AMAC tags reflects the distance between the messages. Binary representations usually do not keep the distance information among orig-

Manuscript received July 1, 2004; revised March 8, 2005. This work was prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Pierre Moulin.

R. Ge and G. R. Arce are with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: ge@ece.udel.edu; arce@ece.udel.edu).

G. Di Crescenzo is with Telcordia Technologies, Inc., Piscataway, NJ 08854 USA (e-mail: giovanni@research.telcordia.com).

Digital Object Identifier 10.1109/TIFS.2005.863504

inal N-ary messages. For instance, given an 8-ary message on the alphabet $\{0, 1, \ldots, 7\}$, the distance between 3 ("011") and 4 ("100") equals 1 in 8-ary but 3 in binary; contrarily, the distance between 0 and 4 in 8-ary is 4 but just 1 in binary. Although Gray codes can keep some distance information, they only preserve distance between adjacent numbers. Hence, AMACs operating directly in an N-ary domain are more effective in keeping distance information. The second motivation of this work lies in the fact that binary AMACs cannot be tuned for sensitivity. Given the same amount of noise, it is desirable that the distances between authentication tags can be adjusted according to different sensitivity requirements. Binary AMACs, however, lack a means to adjust the authentication sensitivity. N-ary AMACs, on the other hand, can provide sensitivity control to binary messages by grouping binary bits into N-ary symbols. As it is illustrated in Section V, such grouping changes the original distance information in binary messages and makes the authentication more sensitive to message variations.

The rest of the paper is organized as follows. Section II states and compares some related work. Section III gives definitions and notions of AMACs. Section IV describes the N-ary AMAC algorithm. The probabilistic model and several properties are developed in Section V. Section VI generalizes the iterated N-ary AMACs with round operations. Section VII gives some applications using N-ary AMACs. Security analysis is provided in the Section VIII. Section IX concludes the paper.

II. Related Work

Several authentication schemes have been proposed in recent years for authenticating multimedia messages. Generally, a multimedia message along with a secret key is input into an authenticator generation system to produce an authenticator. The authenticator can be either embedded into the original message, such as watermarking schemes, or just attached with the message as a tag. The latter can be called as an authentication tag scheme. Reference [2] provides a compact introduction to some existing approaches of both watermarking schemes and authentication tag schemes. Based on the content of this paper, we only discuss the approaches belonging to the latter category.

Reference [10] proposed a content based digital signature scheme for image authentication. Basically, a feature vector that represents the media content is extracted from the original message and hashed into a small digest. The digest is then signed by a standard digital signature algorithm. Since only the semantic information is extracted for authentication, the incidental noise can be tolerated. Although the authors pointed out that different features could be used to represent the content of the image such as edge information, DCT coefficients, and color or intensity histogram, only the histogram feature was used in the paper. Following the same idea, [11] proposed a similar approach using average gray values of image blocks as a feature factor; [12] proposed an image authentication scheme by using an extremely low-bit-rate compression to extract the features.

Since JPEG compression is a popular image compression method, [13] proposed an image authentication method designed to accept JPEG compression yet reject other data manipulations. The feature factor is based on the invariance of the relationships between any two discrete cosine transform (DCT) coefficients at the same frequency of different blocks in an image. These relationships are preserved when DCT coefficients are quantized in JPEG compression. The extracted relations are then encrypted by a public key algorithm to form a digital signature. Similar to the feature extraction from DCT coefficients, [14] exploited the inter-scale relationships of wavelet coefficients of an image and picked those pairs whose magnitude differences were greater than a preset threshold.

All methods above can be referred to as content-based approaches, where different characteristics of an image are used as feature vectors. A problem of these approaches is that it is often hard to devise a succinct yet sufficient set of features that defines a given image. The extracted features are lengthy in terms of storage volume even after a lossless compression. These approaches often prefer to encrypt the feature vector into a digital signature without further hashing. The advantage is that the receiver can obtain a complete feature vector decrypted from digital signature and compare the similarity with the feature vector extracted from the received image. Then a preset threshold can be used to separate the admissible changes from the inadmissible manipulations. The end result, however, is that the signature length may be long. For example, the length of the signature in [13] for a 320×240 size image is 6136 bits, which is more than order of magnitude longer than a conventional 128 or 256-bit message authentication tag.

Different from the content-based approaches, [15] proposed an authentication system by first modifying the original message in such a way to tolerate some predictable distortion. For example, in order to tolerate $\pm v$ distortion of each pixel, this scheme quantized the image with a uniform quantization function with step size equal to $2v+1$ and treated the resulting image as an "original" image which was then authenticated by conventional cryptographic primitives such as HMAC or digital signature. [3] used a similar idea to modify an image after a DCT transformation in order to tolerate the admissible changes. A set of quantization functions are used in this approach to provide the "smoothness" and tolerate the minor changes. The challenge with approaches like these is how to modify the original image in an effective way to accommodate the incidental changes but reject the malicious data manipulation.

Different from all above approaches, AMACs develop a new cryptographic primitive that can be used on multimedia messages. It can be viewed as a new keyed-hash algorithm that hashes an original message into a small digest. Such digest has the property of distance preservation. In terms of multimedia data hashing, the "intermediate" hash in [16], which is a black-white representation of the original image, has the feature of distance preservation; but the "final" hash is generated in such a way as to keep the hashes identical if the two images are similar. This is quite different from the goal of the AMAC hash that tries to keep the property of distance preservation in the final tags.

III. Definitions

In this section, we recall the formal definition of MACs, as used in the cryptography literature, and we extend this definition to obtain a definition for AMACs (following the approach used in [7]). The security requirements of both MACs and AMACs are defined and studied in Section VIII.

A. Message Authentication Codes

Let the integer s be a security parameter. A *message authentication code* (MAC) is a triple $(\mathcal{K}, \mathcal{T}, \mathcal{V})$, where algorithms $\mathcal{K}, \mathcal{T}, \mathcal{V}$ run in time polynomial in s and satisfy the following syntax. The key generation algorithm $\mathcal{K}$ takes as input a random string and returns a secret key k of length s. The authenticating algorithm $\mathcal{T}$ takes a message m and a secret key k as inputs and produces a string *tag*. The verifying algorithm $\mathcal{V}$ takes a message m, a secret key k and a string *tag* as inputs and returns a value $\in \{1, 0\}$. A MAC has to satisfy a *correctness* requirement: after k is generated by $\mathcal{K}$ and *tag* is generated by $\mathcal{T}_k(m)$, $\mathcal{V}_k$, on input (m', tag), outputs 1 if $m' = m$.

B. Approximate Message Authentication Codes

We now define approximate MACs (AMACs), using the above definition for regular MACs as a starting point.

An *Approximate Message Authentication Code* (AMAC) *with parameters* (s, d_m, e, α) is a triple $(\mathcal{K}, \mathcal{T}, \mathcal{V})$, where algorithms $\mathcal{K}, \mathcal{T}, \mathcal{V}$ run in time polynomial in s and satisfy the same syntax as in the definition of MACs; d_m is a distance function on the set of messages input to $\mathcal{T}$; and the following (d_m, e, α)-*noise-tolerance* requirement holds. After k is generated using $\mathcal{K}$, if *tag* is generated using algorithm $\mathcal{T}_k$ on input message m, then algorithm $\mathcal{V}_k$, on input (m', tag), outputs 1 with probability at least α, if $d_m(m, m') \leq e$. Here, e stands for the acceptable number of errors. Also, note that the correctness requirement is a particular case of the (d_m, e, α)-noise-tolerance requirement, when $e = 0$ and $\alpha = 1$.

We will also consider the following requirement. Let d_t be a distance function on the set of authentication tags t's. We say that an AMAC is $(d_m, d_t, \delta_m, \delta_t)$-*distance-preserving* if the following is true: for any m_1, m_2, such that $d_m(m_1, m_2) \leq \delta_m$, it holds that the expected value of $d_t(t_1, t_2)$ is $\leq \delta_t$, where $t_i = \mathcal{T}_k(m_i)$, for $i = 1, 2$, and k has been generated using algorithm $\mathcal{K}$. (We note that in the rest of the paper the Hamming distance function is used for both d_m and d_t.)

We now observe that an $(d_m, d_t, \delta_m, \delta_t)$-distance-preserving AMAC can be easily modified into a (d_m, e, α)-noise-tolerant AMAC, for $e = \delta_m$ and $\alpha = 1/2$. Specifically, the receiver algorithm is modified so that, on input m', tag, it computes $tag' = \mathcal{T}_k(m')$ and returns 1 if $d_t(tag, tag') \leq 2\delta_t$ or 0 otherwise. Furthermore, the following modification results in a (d_m, e, α)-noise-tolerant AMAC, for $e = \delta_m$ and $\alpha = 1 - c^r$, for some constant $0 < c < 1$ and positive integer r. The key generation algorithm is modified so that it generates q independent pseudo-random keys $k_1, \ldots, k_q$. The authenticating algorithm is modified so that it generates q independent tags $tag_1, \ldots, tag_q$, where $tag_i = \mathcal{T}_{k_i}(m)$ and the receiver algorithm is modified so that, on input $m', tag_1, \ldots, tag_k$, it computes $tag'_i = \mathcal{T}_{k_i}(m')$ and returns 1 if the majority of the tests $d_t(tag_i, tag'_i) \leq 4\delta_t$ is satisfied or 0 otherwise. As a consequence, to show that an AMAC satisfies the noise-tolerance requirement, it is enough to show that it satisfies the distance-preservation requirement.

For any positive integer N, an N-ary AMAC is an AMAC where algorithms $\mathcal{K}, \mathcal{T}, \mathcal{V}$ operate over inputs represented as sequences of digits in the alphabet $\{0, \ldots, N-1\}$.

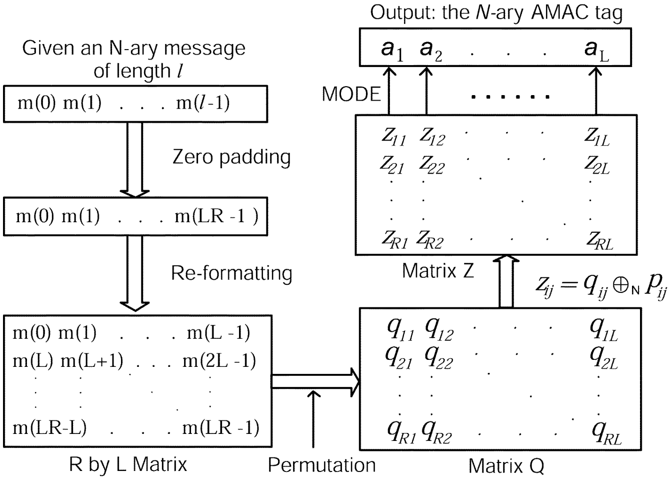

Output: the N-ary AMAC tag
a_1 a_2 . . . a_L
Given an N-ary message of length l
m(0) m(1) . . . m(l-1)
Zero padding
m(0) m(1) . . . m(LR -1)
Re-formatting
m(0) m(1) . . . m(L -1)
m(L) m(L+1) . . . m(2L -1)
m(LR-L) . . . m(LR -1)
R by L Matrix
Permutation
MODE
z_{11} z_{12} z_{1L}
z_{21} z_{22} z_{2L}
z_{R1} z_{R2} z_{RL}
Matrix Z
$z_{ij} = q_{ij} \oplus_N p_{ij}$
q_{11} q_{12} q_{1L}
q_{21} q_{22} q_{2L}
q_{R1} q_{R2} q_{RL}
Matrix Q

Fig. 1. The N-ary AMAC construction algorithm. $\oplus_N$ denotes the modulo N sum operation.

IV. The N-Ary AMAC Algorithm

The N-ary AMAC is a probabilistic checksum calculated by using pseudo-random permutation, masking via a modulo sum operation, and MODE function, such that a small difference between two messages tends to result in a small difference between their N-ary AMACs. For $N = 2$, the N-ary AMACs reduces to the binary AMACs introduced in [5], where modulo sum operator reduces to XOR operation and MODE function reduces to MAJORITY function.

Let m be the input N-ary message of length l. The ith element in the message is denoted as $m(i) \in \{0, \ldots, N-1\}$. Given a secret key k generated by $\mathcal{K}$ and a pseudo-random number generator PRG, the algorithm for constructing N-ary AMACs is given as follows. As with conventional MACs, the length of AMACs, L, is typically chosen in the range $128 \leq L \leq 512$.

A. Initialization

Secret key k and initialization vector I are input to the N-ary pseudo-random generator PRG as a seed. I varies the output of PRG from one instance to another and must be made available to both sender and receiver. Thereafter, PRG is used repeatedly as a source of N-ary pseudo-random numbers. The construction of the N-ary AMAC is shown in Fig. 1.

B. Formatting and Randomization

First, the N-ary message of length l, denoted as $\overrightarrow{m} = [m(0), m(1), \ldots, m(l-1)]$, is padded with zeros to the length $L \times R$ if $l < L \times R$, yielding the padded message

$$\overrightarrow{m_e} = [m(0), \ldots, m(l), \ldots, m(LR-1)],$$

where L is the length of the AMAC tag and R is the minimum positive integer satisfying the inequality $l \leq L \times R$. The padded message is then re-formatted into an R by L matrix, denoted as $\mathbf{M}$:

$$\mathbf{M} = \begin{pmatrix} m(0) & m(1) & \cdots & m(L-1) \\ m(L) & m(L+1) & \cdots & m(2L-1) \\ \vdots & \vdots & \ddots & \vdots \\ m(LR-L) & m(LR-L+1) & \cdots & m(LR-1) \end{pmatrix}.$$

Next, the PRG is used to form a permutation table such that each element in the message matrix is permutated to the new position accordingly. Let $\pi(\cdot)$ denote permutation and matrix $\mathbf{Q}$ denote the message after permutation. Then $\mathbf{Q} = \pi(\mathbf{M})$.

The purpose of the pseudo-random permutation is to not only destroy any existing spatial correlation within the neighboring elements but also enhance the security against attack.

Let $\mathbf{P}$ be the pseudo-random L by R matrix generated from PRG. The matrix $\mathbf{Q}$ is then masked by a modulo N operator with the pseudo-random matrix $\mathbf{P}$, element by element. Denote the masked matrix as $\mathbf{Z}$, $\mathbf{Z} = (\mathbf{P} + \mathbf{Q})_{\mathrm{N}}$ where $z_{ij} = (p_{ij} + q_{ij})_N$.

The modulo operation leads to the variables z_{ij}, which are independent of each other and unbiased whenever the samples $\{p_{ij}\}$ are mutually independent and unbiased. "Unbiased" here means they obey a discrete uniform distribution on $\{0, \ldots, N-1\}$.

C. AMAC MODE Calculation

As shown in Fig. 1, each symbol of the AMAC, a_i, is obtained by computing the MODE of each corresponding column, $a_i = \mathrm{MODE}[z_{1i}, z_{2i}, \ldots, z_{Ri}]$, for $i = 1, 2, \ldots L$. The MODE is defined as the most common value in a set. If a "tie" occurs, the MODE operation breaks the tie by comparing the adjacent values. The resultant N-ary AMAC tag, $\overrightarrow{A}$, together with the initialization data I are sent along with the message m. The receiver compares the received AMAC $\overrightarrow{A}$ and the AMAC $\overrightarrow{A'}$ constructed from the received message m'. The Hamming distance between two AMACs is measured. Over an N-ary alphabet $(N \geq 2)$, the definition of Hamming distance between two vectors is the number of positions in which they differ. Although other distance functions like Euclidean distance are also taken into account here, the Hamming distance between two AMACs is effective in showing the differences between two messages. The larger the distance between $\overrightarrow{A}$ and $\overrightarrow{A'}$, the larger the difference between m and m' is judged to be.

The algorithm described here only shows one round AMAC operation. The generalized multi-round algorithm is discussed in Section VI.

V. Probabilistic Properties of N-Ary AMACS

A. Probability That One N-Ary AMAC Symbol Changes

Given the distance δ_m between message m and m', the probability that one N-ary AMAC symbol changes, denoted as P_A, is derived in Appendix A. Fig. 2 shows the curves of P_A versus the fraction of changes in messages with various length l_m. The monotonicity of the curves exhibits that AMACs keep the property of $(d_m, d_t, \delta_m, \delta_t)$-distance-preservation, for some value δ_t as a function of δ_m. In particular, Fig. 2 shows that the probability of an AMAC symbol change increases monotonically with the increase of differences in the message, and therefore the expected value δ_t can be simply computed according to the value of P_A when the fraction of changes in the message is δ_m/l_m. As already observed in Section III, the distance-preservation property implies a related (d_m, e, α)-noise-tolerance property for N-ary AMACs, where $e = \delta_m$ and α can be $1/2$ or closer to 1.

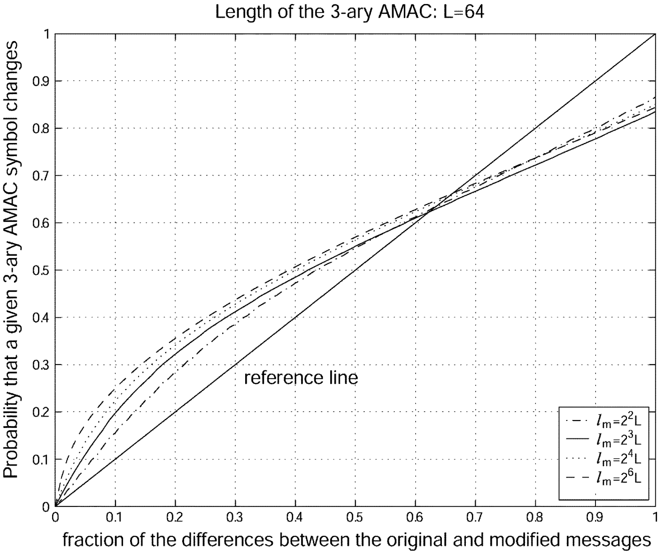


Fig. 2. Probability that one AMAC symbol changes versus the fraction of changes in the 3-ary messages.

When all elements in the message are altered, the curves in Fig. 2 show that $P_A \neq 1$, which means a given AMAC symbol still has a small probability not to change. This is due to the fact that it is possible to have several changing patterns that leave the MODE unchanged. When $(\delta_m/l_m) = 1$, with increasing N, P_A tends to 1. At this case, the probability that the MODE of a column does not change is basically determined by

$$\binom{R - r_x}{r_y} \left(\frac{1}{N-1}\right)^{r_y} \left(\frac{N-2}{N-1}\right)^{R - r_x - r_y} \tag{1}$$

where r_x and r_y represent the number of elements belonging to the unchanged MODE before and after the message changes, respectively. As N gets larger, this probability tends to zero, which means P_A tends to 1.

B. Distribution of the N-Ary AMAC Differences in the Message Space

An l_m-dimension message space with alphabets N is constructed from all possible messages of length l_m. One of the values $\{0, \ldots, N-1\}$ is assigned to each dimension, such that the message space contains N^{l_m} possible messages. Given a key k and an initialization vector I, the N-ary AMAC algorithm maps each message to an AMAC tag of length L. The following theorems hold for the AMACs and the message space.

Theorem V.1: Assume the existence of a pseudo-random[1] generator, AMAC symbols are mutually independent.

Proof: When we calculate an AMAC tag of length L, a message is partitioned into L nonoverlapping sets after the operations with the outputs of PRG, the pseudo-random number generator. Each set contains (l_m/L) elements. Each AMAC symbol is calculated from the corresponding set. Since the permutation

[1] A sequence $\{X_n\}$ is pseudo-random if it is indistinguishable from a uniformly distributed sequence $\{U_n\}$ by any polynomial-time algorithm.

and modulo operations eliminate the correlations between sets, the AMAC symbols are mutually independent. ♢

Theorem V.2: Each N-ary AMAC symbol divides the message space into N equal-size classes. The message space is divided into N^L equal-size classes by L AMAC symbols.

Proof: From the Proof of Theorem V.1, we know that each AMAC symbol is calculated from each set containing (l_m/L) elements, which means there are totally $N^{(l_m/L)}$ possible message-segments in this set and $N^{l_m-(l_m/L)}$ message-segments out of this set. One N-ary AMAC symbol further divides this set into N equivalent classes, each class has $N^{(l_m/L)-1}$ possible message-segments with same AMAC value from $\{0,\ldots,N-1\}$. Combine each class with the remaining $N^{l_m-(l_m/L)}$ message-segments in the message space, we have $(N^{(l_m/L)-1})(N^{l_m-(l_m/L)}) = N^{l_m-1}$ possible messages corresponding to one AMAC symbol. Therefore, each N-ary AMAC symbol divides the message space into N equal-size classes.

Furthermore, since each AMAC symbol is independent of the other AMAC symbols, the message space is divided into N^L equal-size classes by L AMAC symbols. Each class contains N^{l_m-L} messages with same AMACs. ♢

From above discussion, we can derive that, given an N-ary message m, the number of messages with δ_t different AMAC symbols from m's AMAC tag is $\binom{L}{\delta_t}(N-1)^{\delta_t}N^{l_m-L}$. Denote the fraction of such messages in the message space as $F_M(\delta_t)$, then

$$F_M(\delta_t) = \binom{L}{\delta_t}(N-1)^{\delta_t}N^{l_m-L}N^{-l_m} = \binom{L}{\delta_t}\left(1-\frac{1}{N}\right)^{\delta_t}\left(\frac{1}{N}\right)^{L-\delta_t}. \quad (2)$$

F_M is thus a binomial distribution with parameters $(L, 1-(1/N))$, which represents the probability that δ_t out of L symbols change in a AMAC tag. The probability that each AMAC symbol changes is $1-(1/N)$.

To verify the correctness of the calculation of P_A, we use P_A to calculate $F_M(\delta_t)$ and denote the result as $F'_M(\delta_t)$. Assuming each element in the message space is equally likely to be one of the values in $\{0,\ldots,N-1\}$, $F'_M(\delta_t)$ can be calculated as follows:

$$F'_M(\delta_t) = \binom{L}{\delta_t}\sum_{\delta_m=0}^{l_m}\left\{((P_A)|_{\delta_m})^{\delta_t}(1-(P_A)|_{\delta_m})^{L-\delta_t} \cdot \binom{l_m}{\delta_m}\left(1-\frac{1}{N}\right)^{\delta_m}\left(\frac{1}{N}\right)^{l_m-\delta_m}\right\} \quad (3)$$

where $(P_A)|_{\delta_m}$ denotes P_A, given that there are δ_m different elements between the messages. Fig. 3 shows the comparison between $F_M(\delta_t)$ and $F'_M(\delta_t)$, where 3-ary AMACs of length $L = 64$ are obtained from messages with different lengths. It can be seen that $F_M(\delta_t)$ and $F'_M(\delta_t)$ are very close when l_m gets larger, which verifies the computation of P_A.

C. Distance Measure in N-Ary AMAC Symbols

Authenticity decisions do not rely on AMAC symbol individually but as a group on the distance between AMACs. The statistical analysis of the distance between AMAC tags, δ_t, is thus important. As mentioned before, Hamming distance is sufficient to measure the difference here. Table I shows the distance of AMACs between the original 3-ary message and its modified version with $\delta_m = 64, 128, 256, 512, 1024$. The length of AMAC is 128, and the message length is 512 K. According to the table, when $\delta_m = 64, \hat{\delta}_t$, the average of the observed distance between AMACs obtained by simulations, equals 0.906, which means about 1 of the 128 symbols is affected. Similarly, when $\delta_m = 1024, \hat{\delta}_t$ equals 10.004, which means about ten of the 128 symbols are changed. Assume AMAC symbols are mutually independent, then δ_t, the distance between AMACs with length L, follows a binomial distribution with parameters (L, P_A). Denote the expected value of δ_t as $E\{\delta_t\}$ and the variance as $\mathrm{VAR}\{\delta_t\}$, then

$$E\{\delta_t\} = L \cdot P_A, \qquad \mathrm{VAR}\{\delta_t\} = L(P_A - P_A^2). \quad (4)$$

The comparisons of the results between simulations and analysis are also shown in Table I, where $\hat{v}$ is the observed variance from simulations. It can be observed that the results from the probabilistic model are consistent with those of the simulations.

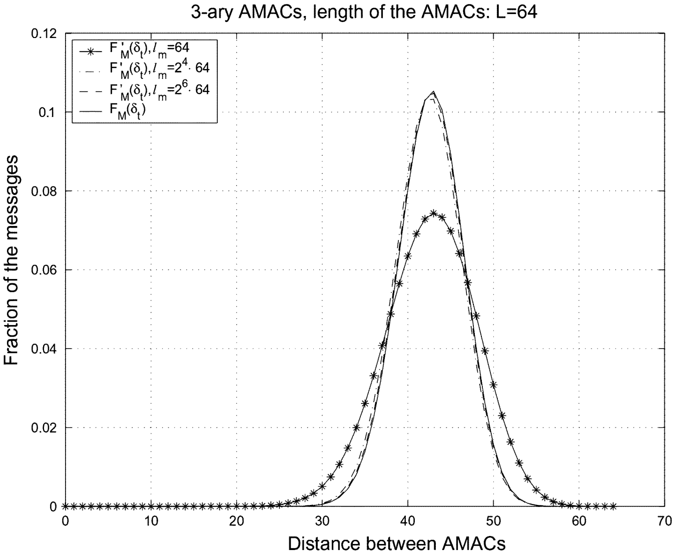


Fig. 3. Fraction of the messages versus the distance of AMACs. Different curves are computed by (2) and by (3), as indicated in the legend.

TABLE I
RESULTS FROM ANALYSIS AND SIMULATIONS, 3-ARY AMACS

δ_m	64	128	256	512	1024
$\hat{\delta}_t$	0.906	1.798	3.406	5.988	10.004
$E\{\delta_t\}$	0.885	1.724	3.276	5.960	10.106
$\hat{v}$	0.857	1.897	2.769	5.168	8.840
$\mathrm{VAR}\{\delta_t\}$	0.879	1.701	3.193	5.683	9.308

D. Impact of the Size of Alphabets N

The sensitivity of N-ary AMACs varies according to the different alphabets N. Fig. 4 shows how the value of N influences the sensitivity of AMAC symbols. All three probability curves, binary and 3-ary as well as 4-ary AMACs, increase gradually in concert with the increasing changes in messages. When half of the message is changed, the probability of binary AMAC reaches 0.5; whereas, probabilities of 3-ary and 4-ary AMACs

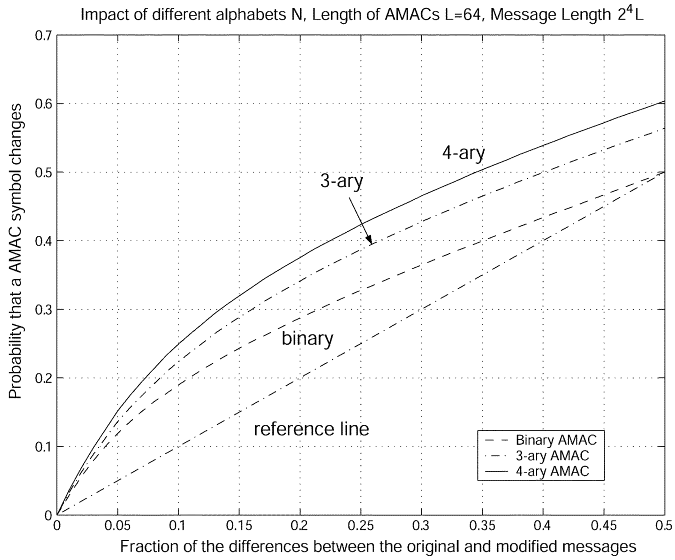


Fig. 4. Probability of one AMAC symbol changing for different alphabets N.

are greater than 0.5. It can be concluded that the larger the alphabets N, the higher the sensitivity of the resultant AMACs. This result also provides a method to control the sensitivity of binary AMACs. Certain number of bits can be grouped together into an N-ary symbol and the N-ary AMAC can be computed based on the N-ary representation of original message. Thus, authentication can be adjusted to different levels of sensitivity. The reason is that the original distance information in the message is reorganized when bits are grouped into N-ary symbols. The resultant N-ary AMACs then reflect the new distance information and provide higher levels of sensitivity.

To illustrate this concept, consider an AMAC computation of a 256 gray-level image with each pixel represented by 8 bits. The binary AMAC can be computed directly from the bit image representation. A 4-ary AMAC can be computed by grouping every two binary bits into a 4-ary symbol to form a 4-ary message. Similarly, 8-ary, 16-ary, and even 256-ary AMACs can be computed by grouping the appropriate number of bits. Fig. 5 shows how the size of the alphabet influences the sensitivity of AMAC symbols. The original image is the uncompressed version of Fig. 9. The image is contaminated by different densities of salt-and-pepper noise generated by Matlab's imnoise function. At each level of noise density, the sensitivity of AMACs increases with an increasing value of N. It can be concluded that for the same input message, the larger the alphabets N, the higher the sensitivity of the AMACs.

VI. ITERATED N-ARY AMACS

As seen in Fig. 2, P_A, the probability that an AMAC symbol changes is a function of (δ_m/l_m), the fraction of changes in the message. If (δ_m/l_m) is fixed, the value of P_A can be still tuned by using iterated round operations described as follows.

Let the message length be $L\prod_{i=1}^{U} R_i$, where L is the length of the AMAC, U is the number of rounds. The generalized algorithm is depicted in Fig. 6. Similar to the algorithm described in Section IV, the N-ary message is pseudo-randomly permuted and formatted into a matrix of $\prod_{i=1}^{U} R_i$ rows and L columns.

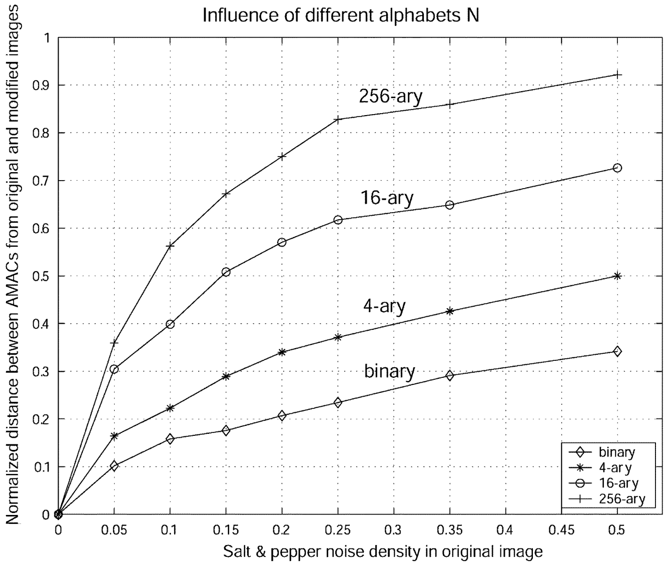


Fig. 5. Various densities of salt-and-pepper noise in the original image versus the normalized distances between N-ary AMACs generated from the original and the contaminated images. For the fair comparison, all these AMACs have same 512 bit-length but vary in symbol-length L: 512 in binary, 256 in 4-ary, 128 in 16-ary, 64 in 256-ary. Y-coordinate represents $d_t(\vec{A},\vec{A}')/L$. $d_t(\cdot)$: distance function.

The matrix elements are fed into a modulo operator along with the pseudo-random elements generated by PRG and split into $\prod_{i=1}^{U-1} R_i$ sub-matrices with R_U rows and L columns each. In the first round, the MODE of every column in each sub-matrix is computed, and the message is reduced to $L\prod_{i=1}^{U-1} R_i$ elements. As the round operation goes on, the message matrix continues to reduce. After U rounds are processed, an AMAC of length L is the final tag. Note a "Modulo N" step with fresh pseudo-random numbers is performed in each round prior to the MODE calculation to further secure the process.

The previous probabilistic model for one round operation can be extended to generalize the U rounds algorithm. Let P_m^i be the probability of the MODE changing in the ith round calculation $(i \geq 1)$ and $P_C^i(d)$ be the probability that d out of R_{U-i+1} elements change in a column. Then, $P_C^i(d)$ equals

$$\begin{cases}\binom{R_{U-i+1}}{d}\left(P_m^{i-1}\right)^d\left(1-P_m^{i-1}\right)^{R_{U-i+1}-d}, & \text{for } i \geq 2\\ \binom{R_U}{d}p_0^d(1-p_0)^{(R_U-d)}, \quad p_0=\frac{\delta_m}{l_m}, & \text{for } i=1\end{cases} \tag{5}$$

Hence, P_m^i is as follows:

$$P_m^i = \sum_{d=0}^{R_{U-i+1}} P_C^i(d) P_{C_m}(R_{U-i+1}, d) \tag{6}$$

where P_{C_m} is calculated by (11) in Appendix A.

P_C^i and $P_m^i (i \geq 2)$ are calculated recursively given the value of P_m^1, which is obtained from (8) by replacing R with R_U and P_A with P_m^1. After U rounds, $P_A = P_m^U$ is the probability that a given AMAC symbol changes.

P_A is now a function of $U, R_1, \ldots, R_U$ besides (δ_m/l_m). Its value can be fine tuned by choosing different values of these parameters. To show the impact of different number of rounds on the authentication, a 3-ary message of length 48×2^{12} with

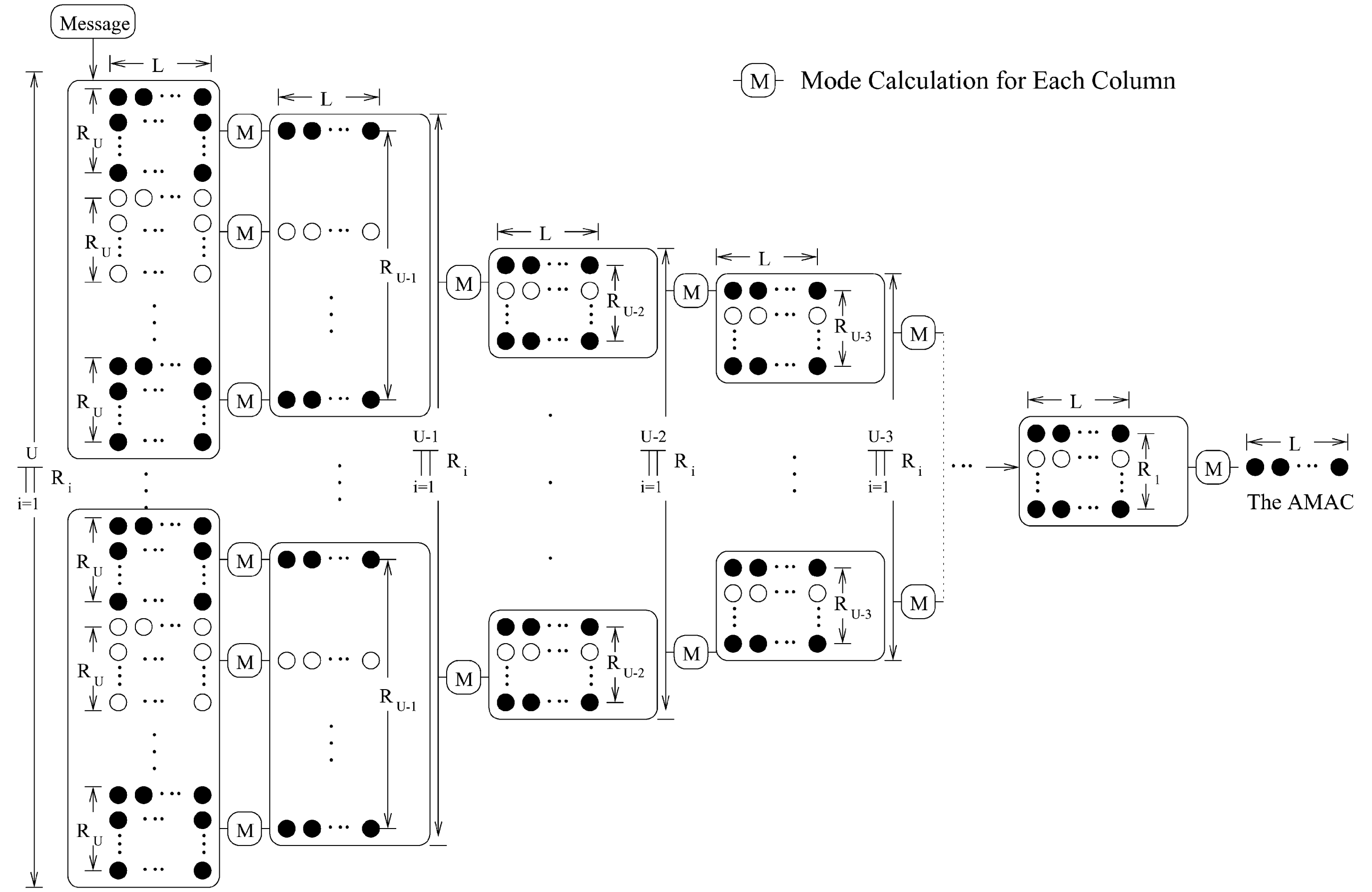


Fig. 6. MAC obtained by U rounds of MODE calculation. MODULO N operation is performed in each round prior to the MODE calculation but is not shown in the figure due to space limitation.

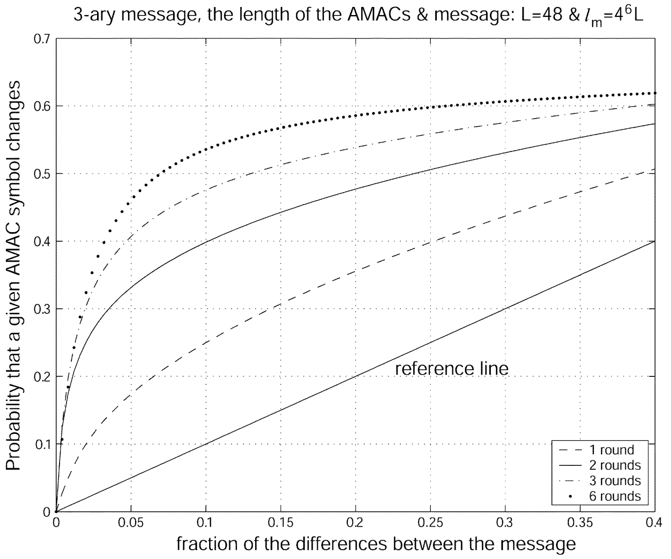


Fig. 7. Impact of different number of rounds. Alphabet $N = 3$. The fraction (δ_m/l_m) from 0 to 0.4 is shown.

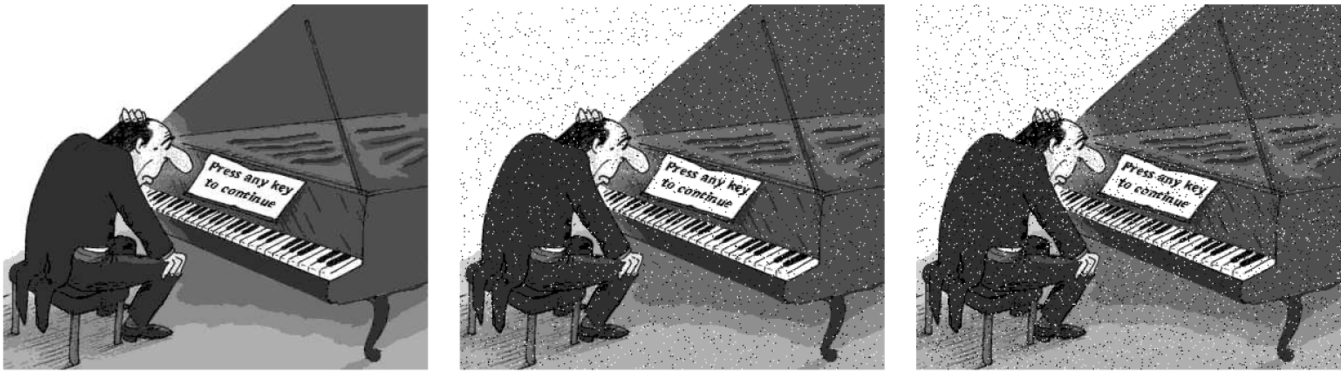
$L = 48$ and $U = 1, 2, 3, 6$ is investigated. In each $U, R_1 = R_2 \cdots = R_U$. The result is shown in Fig. 7, where P_A versus (δ_m/l_m) is plotted. The line of $P_A = (\delta_m/l_m)$ is also plotted as a reference. With the increasing of the number of rounds, the P_A curves deviate further from the reference line.

Therefore, in the generalized AMAC scheme, the probabilistic property of the AMAC is tunable by adjusting the number of rounds. The sensitivity to the small amount of changes increases quickly when the number of rounds increases. For larger amount of changes, however, the curves with more rounds become flatter and thus less sensitive to the increasing changes.

VII. Application of N-Ary AMACs

To illustrate the attributes of N-ary AMACs, consider the example shown in Fig. 8. The original graphics image in Fig. 8(a) has eight different gray values. Fig. 8(b) and (c) depict the graphics images distorted by salt and pepper noise. The noise density is 3% and 5%, respectively. The AMAC tags from the original image and the noisy versions are computed. The normalized distance[2] between the original tag and the tag of the noisy version is measured to judge the authenticity between the original message and the distorted image. The binary AMAC tag is computed from the binary representation of the 8-ary graphics. The 8-ary AMAC tag is computed directly from the image. It can be seen that the binary AMAC is less sensitive than the 8-ary AMAC. When the noise density is increased from 3% to 5%, the quality of the image downgrades significantly; but the normalized distance between binary AMACs only increases

[2] Hamming distance divided by the length of the vectors.

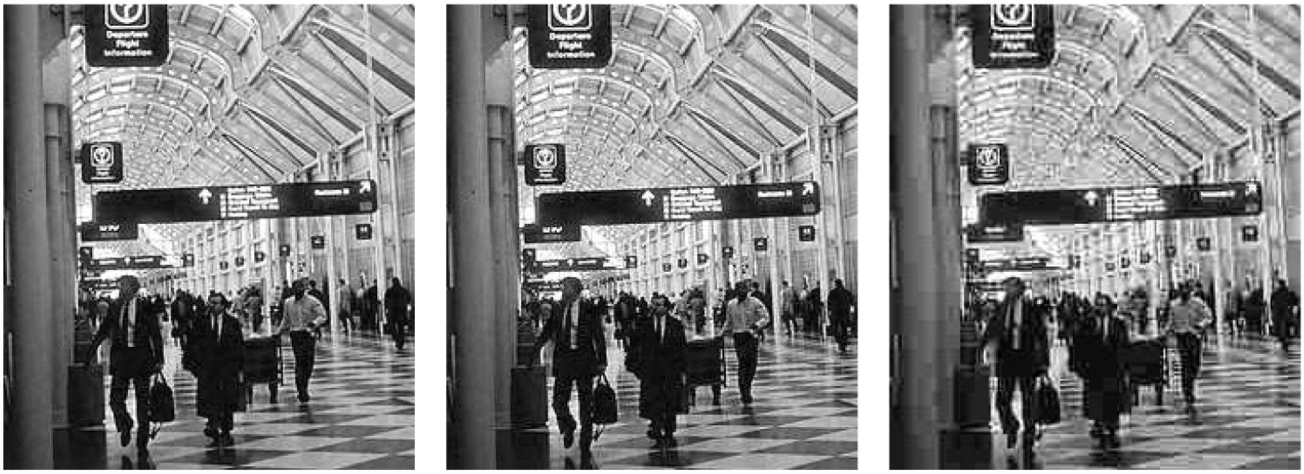


(a) Original 8-ary graphics

(b) 3% salt & pepper noise
Distance between binary AMACs: 0.042
Distance between 8-ary AMACs: 0.185

(c) 5% salt & pepper noise
0.069
0.258

Fig. 8. AMACs computed from the 8-ary graphics images versus the AMACs computed from the binary representations. All the authentication tags are 384 bits long.



(a) Q:99; MAE: 0.160
HMAC-MD5: 0.484
Binary AMAC: 0.109
Quaternary AMAC : 0.141

(b) Q:75; MAE: 4.820
0.532
0.370
0.516

(c) Q:10; MAE: 13.788
0.498
0.487
0.609

Fig. 9. Distances between authentication tags are listed including HMAC-MD5 and binary AMACs as well as 4-ary AMACs. All the authentication codes are 128 bits long. A quaternary message is constructed by converting each two binary bits into one quaternary symbol. All distances of authentication codes are normalized by their lengths.

from 0.042 to 0.069, representing about 7% change of tag bits. This low sensitivity may mislead the receiver to believe that the noise level is still low. Meanwhile, the distance between 8-ary AMACs increases from 0.185 to 0.258, representing more than 25% change in the tag. This indicates a clear degradation in image quality. As stated before, binary AMACs are less sensitive in this example in that the binary representation changes the distance information in the original 8-ary graphics image. In particular, the normalized Hamming distance between the noise-free image and the noisy image with 5% noise density is 0.043 in 8-ary but only 0.025 in their binary representations.

The following example further illustrates the characteristics of N-ary AMACs. There are three JPEG representations of an "airport terminal" image shown in Fig. 9. The left, center, and right images are the representations having quality factors of $Q = 99, Q = 75$ and $Q = 10$, respectively. As indicated by the normalized mean absolute error (MAE), the distortion of the image increases when the quality factor decreases. For each of these images, and for the original uncompressed image, a conventional MAC (using MD5) and a binary AMAC as well as a 4-ary AMAC tag are computed. Quaternary AMACs are computed by converting the images into 4-ary representations first. The distances of tags between source image and compressed images are measured. For the image with severe compression artifacts $(Q = 10)$, or for the slightly modified image $(Q = 99)$, about 50% of the MD5 tag bits differ from those of the original image. It is the goal of MD5 that each tag bit has a 0.5 probability to change no matter how many errors occur. Both binary AMACs and quaternary AMACs, on the other hand, show that the distance of the tags between the original and the distorted images gradually increases in concert with the distortion in the images. In all three different quality settings, the distances of quaternary AMACs are larger than their binary counterparts. It is also interesting to see how fast the distance increases. When the image quality factor decreases from $Q = 99$ to $Q = 75$, the distance of 4-ary AMACs increases from 0.141 to 0.516; whereas, the distance of binary AMACs only increases from 0.109 to 0.37. For large amount of distortion, such as when the quality factor decreases from $Q = 75$ to $Q = 10$, the increasing rate of the distance of 4-ary AMACs becomes smaller than the binary AMACs, only increasing from 0.516 to 0.609. Such characteristics show that N-ary AMACs are more suitable for slight message differences, which is exactly the desired "tuning" feature we sought.

Although not elaborated in this paper, another application of N-ary AMACs can be the noise-tolerant message authentication in a wireless multicast group. Imagine that one node in a multicast group wants to send a multimedia message to all other nodes in the group. Since the originator has different wireless paths reaching different receivers, the copies of the message received by different destinations may have different levels of interference. The challenge of authentication in such scenario is not only the message being contaminated by noise but also the message experiencing different amounts of noise in different paths. Traditional MACs cannot tolerate any errors that may be acceptable. Any premeasures taken at the sender side to tolerate the noise are not sufficient due to the different paths. AMACs thus are a good solution for such application. Although the multicast message copy at a given receiver may contain a different number of errors than the copies received by other receivers, the copy still can be accepted as long as $d_m(m, m') \leq e$, where e is an acceptable number of errors. This is also an example of group authentication. The group secret session key should be agreed before computing AMACs. Any authenticated group Diffie–Hellman key exchange can be applied in the AMAC key generation phase.

VIII. Security Analysis of N-Ary AMACs

First note that the AMAC construction in Section IV already satisfies some kind of security property. Specifically, assume that the construction is proved $(d_m, d_t, \delta_m, \delta_t)$-distance-preserving, for some distance functions d_m, d_t and parameters δ_m, δ_t. (We showed this fact in Section V). Then consider an adversary trying to produce two messages m_1, m_2 such that $d_m(m_1, m_2) > c\delta_m$ and $d_t(t_1, t_2) \leq \delta_t$, where $t_i = \mathcal{T}_k(m_i)$, for $i = 1, 2$, and k has been generated using the key-generation algorithm $\mathcal{K}$. (Such an adversary could convince a verifier that the same tag is valid for both m_1 and m_2.) Then the monotonicity of the probability P_A that a given symbol in the AMAC tag changes as a function of the differences between m_1 and m_2, implies that there exists a constant $c > 1$ for which this adversary can only succeed with probability at most $1/2$. Precisely, c can be chosen as the constant such that P_A at value $c\delta_m$ is at least twice as large as P_A at value δ_m. Furthermore, this probability can be decreased by modifying the construction so that multiple independently generated tags are returned by the authenticated algorithm. Finally, we note a particular case ($\delta_t = 0$) that will be of interest later; that is, the adversary is able to compute m_1, m_2 such that $d_m(m_1, m_2) > c\delta_m$ and $t_1 = t_2$ only with probability $(1 - P_A)^L$, where P_A is the probability that a given symbol in the AMAC tag changes when m_1 and m_2 differ by at least $c\delta_m$ symbols, L is the tag length.

While this security property can be satisfactory for some applications, in order to further investigate the applicability of AMACs, we now define a stronger type of security requirement, security against a chosen message attack, by adapting the strongest known type of security for conventional MACs in the cryptography literature. We show that the AMAC construction of Section IV does not fully satisfy this stronger definition (a similar claim can be seen to hold for several content-based multimedia authentication techniques in the literature). However, we show how to use standard cryptographic algorithms to modify our construction so that it satisfies this stronger definition as well.

A. Insecurity of the N-Ary AMAC Against Chosen Message Attacks

Let $\mathcal{MA} = (\mathcal{K}, \mathcal{T}, \mathcal{V})$ be an arbitrary message authentication scheme. A chosen message attack against message authentication codes, MACs, is performed in two phases. The first is adversary's "learning" phase in which she is given the oracle access to $\mathcal{T}_k(\cdot)$, where the key k was *a priori* chosen by $\mathcal{K}$ (thus, k is fixed during the attack). "Oracle access" means that the adversary can choose whatever message m_i she wants as input and ask $\mathcal{T}_k(\cdot)$ to return the corresponding tag t_i; but the key k and other randomness of $\mathcal{T}_k(\cdot)$ are kept unknown to her. She can query the oracle up to q times. Then she enters "forgery" phase to output a pair of a message and its tag (m, t). The adversary succeeds if $\mathcal{V}_k(m, t) = 1$. Note that m is not necessarily a given one, but was never a query in adversary's "learning" phase.

Then, we say that an AMAC is (q, γ, ϵ)-*secure* against chosen message attack if the following holds: for any efficient adversary algorithm, Adv, if Adv queries algorithm $\mathcal{T}_k(\cdot)$ q times with adaptively chosen messages, thus obtaining pairs $(m_1, t_1), \ldots, (m_q, t_q)$, and then returns a pair (m, t), the probability that $\mathcal{T}_k(m, t) = 1$, where $d_m(m, m_i) \geq \gamma$, for $i = 1, \ldots, q$, is at most ϵ.

Since AMACs are noise-tolerant, the distance between query message m_i and forgery message m should exceed some noise-tolerant boundary γ, i.e., $d_m(m, m_i) \geq \gamma$. Indeed, when $\gamma = 1$, this definition is identical to the known definition for MACs of (q, ϵ)-security against chosen message attack. If by s we denote the security parameter, as for MACs, the most desirable security level for AMACs is (q, γ, ϵ)-security against chosen attack for any q polynomial in s and any ϵ negligible in s. (A function $f : \mathrm{N} \rightarrow \mathrm{N}$ is *negligible* if for all positive constants c, there exists a positive integer n_c such that, for all $n \geq n_c$, it holds that $f(n) \leq n^{-c}$. Intuitively, a negligible probability event is so unlikely that is not expected to be observed.)

We first show that the one-round N-ary AMAC construction in Section IV is not secure against a chosen message attack where the adversary is allowed to query a large enough (but polynomial in the security parameter) number q of queries.

Theorem VIII.1: Consider the N-ary AMAC algorithm in Section IV, taking as input messages of length l_m and returning tags of length L, and assume that it is $(d_m, d_t, \delta_m, \delta_t)$-distance preserving, where $\delta_m < l_m/2$. Then, for any $\gamma > \delta_m$, there exists an efficient adversary, Adv, performing $q = (l_m + \log_N(l_m!) + \log_N \epsilon)/L$ queries to oracle $\mathcal{T}_k(\cdot)$ and returning with probability at least ϵ a pair (m, t) such that $\mathcal{V}_k(m, t) = 1$ and $d_m(m, m_i) \geq \gamma$, for $i = 1, \ldots, q$.

This analysis is compatible with the binary case, and, when $N = 2$, it is essentially the same as in [7]. Furthermore, it extends to the U-round AMAC construction, where the number of queries $q = U((l_m + \log_N(l_m!) + \log_N \epsilon)/L)$ is still polynomial in the security parameter, which is considered insecure from a cryptographic point of view simply because in practical applications, an adversary can make either zero queries (no access to oracle) or any polynomial number of them.

This theorem is proved by exhibiting a specific Adv that makes q queries and output a valid pair (m, t). The strategy of Adv is that of guessing both permutation, π, and pseudo-random sequence, rs, using the answers of her queries to $\mathcal{T}_k(\cdot)$, which are computed as so that each AMAC tag symbol will help Adv reduce the set of possible π, rs by a factor of N on average. The detailed proof is shown in Appendix B.

B. Securing the N-Ary AMAC Against Chosen Message Attacks

We stress that it should not come as a surprise that for some polynomially large value q, the presented AMAC is not (q, ϵ)-secure under chosen message attack, since this construction only uses a polynomially large amount of pseudo-randomness, contrarily to other MAC constructions that assume, for instance, the existence of a pseudo-random function (that can provide an exponential amount of pseudo-randomness). We now show how, using standard cryptographic algorithms such as a symmetric encryption scheme and a conventional MAC algorithm, the AMAC presented in Section IV can be modified so that it satisfies the stronger definition of security against chosen message attack (even if the adversary can make an arbitrary polynomial number of queries). Our main result here is the following.

Theorem VIII.2: Consider the N-ary AMAC algorithm in Section IV, taking as input messages of length l_m, and assume that it is $(d_m, d_t, \delta_m, \delta_t)$-distance preserving, where $\delta_m < l_m/2$. Also, assume the existence of a symmetric encryption scheme and a conventional MAC secure against chosen message attack. Then there exists $\gamma > \delta_m$ and an N-ary AMAC construction that is (q, γ, ϵ)-secure against chosen message attack, for any polynomial q and any negligible ϵ.

In summary, Theorem VIII.2 indicates that there exists a scheme to make a previous $(d_m, d_t, \delta_m, \delta_t)$ distance-preserving AMAC that is not secure against chosen message attack with an arbitrary polynomial number of queries into a secure one against such attack.

Let (K_1, T_1, V_1) denote the AMAC algorithm discussed in previous sections; let (Kg, E, D) denote a symmetric encryption scheme; and let (K_2, T_2, V_2) denote a conventional MAC scheme secure again chosen message attack. The modification of our AMAC goes as follows.

The key-generation algorithm K returns three random keys k_1, k_2, k_3, generated using algorithms K_1, K_2, Kg, respectively.

The tagging algorithm first computes an AMAC tag t_1 for the input message m exactly as before using T_1 and key k_1; then it computes an encryption c of t_1 using the encryption algorithm E and key k_3; finally, it computes a conventional MAC tag t_2 of c using algorithm T_2 and key k_2. The final tag is the pair $t = (c, t_2)$.

The receiving algorithm, on input m', c, t_2, and keys k_1, k_2, k_3, first verifies that t_2 is a valid MAC tag for c by running the conventional MAC receiver algorithm V_2 using key k_2; if so, it decrypts c as t_1 using the decryption algorithm D and key k_3; then, it runs the AMAC receiver algorithm V_1 on input m', t_1 and key k_1.

Appendix C sketches the proof that the resulting AMAC is secure against chosen message attack for an arbitrary (polynomial) number of queries from the adversary.

IX. Discussion and Conclusion

In this paper, we proposed a new noise-tolerant AMACs for N-ary alphabets. The codes are probabilistic in nature and have the property of distance preservation. The theoretical analysis was performed on the probabilistic properties of N-ary AMACs along with the simulations. Some application examples were presented to show the potential application areas of such codes. As discussed in Section II, our approach takes a different perspective from other approaches in the literature to authenticate messages that may have been affected with admissible changes. AMACs can be viewed as a new keyed-hash algorithm that hashes the original message into a small digest. Such digest has the property to reflect the differences between messages; i.e., small differences between messages lead to small differences between AMAC tags, and vice versa. One argument of such distance preservation is that it might be hard to determine the boundary between admissible changes and inadmissible manipulations based on the authentication tags. Such distance preservation, however, brings the flexibility of authenticating messages with different levels of sensitivity, such as tightening the boundary for high sensitive messages.

We also adopt rigorous security analysis commonly used in cryptography literature to perform the security analysis on the AMAC scheme. We believe that such security analysis is important for multimedia message authentication.

Appendix

A. Derivation of the Probability That One N-Ary AMAC Symbol Changes

Let l_m be the length of the padded message and assume that δ_m elements in m are changed. Let $\overrightarrow{C}$ be a column in $\mathbf{Z}$ that contains R elements and let C_m be the MODE of $\overrightarrow{C}$. We next derive the probability that C_m changes given that d elements in $\overrightarrow{C}$ are changed. For simplicity, we compute the case of $N = 3$ as starting point, then generalize the results to $N > 3$.

Let X be the event that C_m changes. Denote $(A_{r_0 r_1 r_2})_i$ as the ith possible combination in $\overrightarrow{C}$, where r_0 elements are of value 0, r_1 elements are of value 1, and r_2 elements are of value 2, and $r_0 + r_1 + r_2 = R$. Denote $(B_{d_0 d_1 d_2})_j$ as the jth possible combination in which d_0, d_1, d_2 elements change their values from $0, 1, 2$, respectively. Denote $(T_{s_0 s_1 s_2})_k$ as the event of the kth possible combination in which s_0, s_1, s_2 elements change their original values into $0, 1, 2$, respectively. Clearly, $s_0 + s_1 + s_2 = d_0 + d_1 + d_2$. Subscripts i, j, k represent the indices referring to each of all the possible combinations in events A, B, T, respectively. Let $P_{C_m}(R, d)$ be the probability that C_m changes given that there are d errors in column $\overrightarrow{C}$. Then, we have

$$P_{C_m}(R,d) = \sum_i \frac{R!}{r_{0i}! r_{1i}! r_{2i}!} \left(\frac{1}{3}\right)^R \sum_j \frac{\binom{r_{0i}}{d_{0j}} \binom{r_{1i}}{d_{1j}} \binom{r_{2i}}{d_{2j}}}{\binom{R}{d}} \cdot \sum_k P_{jk} P(X|R, d, A_i, B_j, T_k), \quad (7)$$

where A_i, B_j, T_k represent $(A_{r_0 r_1 r_2})_i, (B_{d_0 d_1 d_2})_j, (T_{s_0 s_1 s_2})_k$ in short. How d elements change from their original values to the new values can be regarded as a Markov Chain. Then P_{jk} is the transition probability of such Markov Chain. Assume each element has equal probability to change to one of the other two values, then $P_{jk} = n(1/2)^d, \sum_k P_{jk} = 1, n$ is the number of possible changes from one pattern to the other.

If the new $\mathrm{MODE}(r_{0_i} - d_{0_j} + s_{0_k}, r_{1_i} - d_{1_j} + s_{1_k}, r_{2_i} - d_{2_j} + s_{2_k})$ is different from the original $\mathrm{MODE}(r_{0_i}, r_{1_i}, r_{2_i})$, then $P(X|R, d, A_i, B_j, T_k) = 1$, otherwise 0.

Next, we calculate the probability that one AMAC symbol changes given that δ_m elements change in message m, denoted as P_A

$$P_A = \sum_{d=0}^{R} P_C(d) \cdot P_{C_m}(R, d), \quad (8)$$

where $P_C(d)$ is the probability of d differences in one column given δ_m differences in whole message and computed by

$$P_C(d) = \binom{R}{d} \binom{l_m - R}{\delta_m - d} / \binom{l_m}{\delta_m}. \quad (9)$$

Note that when δ_m and l_m are large in relation to R, $P_C(d)$ can be approximated by a binomial distribution with parameters (R, p_0) where $p_0 = (\delta_m / l_m)$. Therefore

$$P_C(d) \approx \binom{R}{d} p_0^d (1 - p_0)^{(R-d)}. \quad (10)$$

From the probability equations derived for the 3-ary case, we can generalize the calculation to the N-ary case. Equation (8) still holds for calculating P_A, where

$$P_{C_m}(R,d) = \sum_i \frac{R!}{r_{0_i}! r_{1_i}! \cdots r_{(N-1)_i}!} \left(\frac{1}{N}\right)^R \cdot \sum_j \frac{\binom{r_{0_i}}{d_{0_j}} \binom{r_{1_i}}{d_{1_j}} \cdots \binom{r_{(N-1)_i}}{d_{(N-1)_j}}}{\binom{R}{d}} \cdot \sum_k P_{jk} P(X|R, d, A_i, B_j, T_k) \quad (11)$$

and $P_C(d)$ is as same as in (9), as well as $P_{jk} = n(1/N - 1)^d, \sum_k P_{jk} = 1$. P_{jk} is the transition probability, n is the number of possible changes from one pattern to the other.

B. Proof of Theorem VIII.1

We define two message spaces: S_{pseudo} and S_{random}. The former is defined as the space induced by the random choices after running oracle $\mathcal{T}$ in the chosen message attack by Adv; the latter is defined as the former except that the pseudo-random numbers generated by generator PRG are replaced with uniformly and independently distributed random numbers. Due to the pseudo-randomness properties of PRG, S_{pseudo} and S_{random} are indistinguishable between polynomial time computation. Then, in order to prove that Adv's probability of success is at least ϵ, it is enough to show that in space S_{random}Adv can perform q adaptive queries $(m_1, t_1), \ldots, (m_q, t_q)$ to oracle $\mathcal{T}_k(\cdot)$ and return with probability at least ϵ a pair (m, t) such that $\mathcal{V}_k(m, t) = 1$ and $d_m(m, m_i) \geq \gamma$, for $i = 1, \ldots, q$. Thus, the proof is performed in space S_{random}. (Note that we are showing a stronger claim since construction of AMACs can only be more secure in S_{random} than in S_{pseudo}.)

We now prove that Adv is successful with q queries as claimed. As proved in Section V, each N-ary AMAC symbol partitions the message space into N equal size subsets, we observe that each symbol t_{ij} of AMAC tag t_i for message m_i restricts the values of possible (π, rs) used by $\mathcal{T}_k(\cdot)$ at Adv's point of view. We define T_0 as the largest set of pairs (π, rs) possibly used by $\mathcal{T}_k(\cdot)$ before the first query by Adv, and define T_i as the largest possible set of pairs (π, rs) consistent with the transcript $((m_1, t_1), \ldots, (m_i, t_i))$ obtained by Adv, for $i = 1, \ldots, q$. Now we observe that, since Adv uniformly and independently chooses all queries, on average it holds that $|T_i| = |T_{i-1}|/N^L$. Then, since $|T_0| = N^{l_m} l_m!$, we obtain that $q = (l_m + \log_N(l_m!) + \log_N \epsilon)/L$ queries are enough, on average, to obtain $|T_q| = 1/\epsilon$, from which it follows that with probability at least ϵ, the pair (m, t) returned by Adv satisfies $\mathcal{V}_k(m, t) = 1$.

C. Proof of Theorem VIII.2

We start the proof of this theorem by recalling the definition of a symmetric encryption scheme. Let (Kg, E, D) be a triple of probabilistic polynomial-time algorithms with the following syntax. On input an n-bit security parameter, the key-generation algorithm Kg returns a random n-bit key k. On input k and a message m, the encryption algorithm E returns a ciphertext c. On input k and a ciphertext c, the decryption algorithm D returns a message m or a failure symbol $\perp$. The triple (Kg, E, D) is a *secure symmetric encryption scheme* if the following two requirements of correctness and security are satisfied. Correctness requires that if k is generated by Kg and c is returned by E on input k, m, then the output of D on input k, c, is equal to m. Security (in the "real-or-random" sense) requires that an adversary (not knowing k) who is given access to an oracle O, and after making a polynomial number of queries, can distinguish if O is equal to $E(k, \cdot)$ or equal to a random function with the same length parameters only with negligible probability.

Also, we use a conventional MAC scheme (K_2, T_2, V_2) secure under chosen message attack.

We now sketch the proof that the resulting AMAC is secure under chosen message attack for an arbitrary (polynomial) number of queries from the adversary. Recall that the adversary is successful only if, given several (m_i, t_i), for any $i = 1, \ldots, q$ and *any* polynomially large q, and for adaptively chosen $m_1, \ldots, m_q$, she can generate with probability at least ϵ a pair (m', t'), where m' has distance at least γ from all m_i and $t' = (c', t'_2)$ is a valid tag for m' according to the modified AMAC. Then note that after each query m_i, the adversary only obtains an encryption c_i of the N-ary AMAC tag $(t_1)_i$ and a conventional MAC tag of c_i, and therefore obtains no information about the randomness used to generate AMAC tags. Because of the latter fact, and using the fact that m' has large distance from all m_i, with very high probability, it holds that ${t_1}' \neq (t_1)_i$ for all i, where ${t_1}'$ is the decryption, using D, of c', or otherwise we can use the adversary to break the encryption scheme E. Therefore, the adversary needs to use a value c' different from all c_i associated with the queried m_i; but for this c', the adversary cannot produce a valid tag ${t_2}'$ as she does not have key k_2. Specifically, if she did that, she would violate the security under chosen message attack of the conventional MAC used here.

Acknowledgment

The authors thank P. Moulin and the reviewers for their constructive comments and suggestions that have significantly improved the manuscript.

References

[1] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed. New York: Wiley, 1996.

[2] B. B. Zhu, M. D. Swanson, and A. H. Tewfik, "When seeing isn't believing: Current multimedia authentication technologies and their applications," *IEEE Signal Process. Mag.*, pp. 40–49, Mar. 2004.

[3] C. W. Wu, "On the design of content-based multimedia authentication systems," *IEEE Trans. Multimedia*, vol. 4, no. 3, pp. 385–393, Sep. 2002.

[4] R. Graveman and K. Fu, "Approximate message authentication codes," in *Proc. 3rd Annu. Fedlab Symp. Advanced Telecommunications/Information Distribution*, vol. 1, College Park, MD, Feb. 1999.

[5] G. R. Arce, L. Xie, and R. F. Graveman, "Approximate image authentication codes," in *Proc. 4th Annu. Fedlab Symp. Advanced Telecommunications/Information Distribution*, vol. 1, College Park, MD, Mar. 2000.

[6] L. Xie, G. R. Arce, and R. F. Gravemen, "Approximate image message authentication codes," *IEEE Trans. Multimedia*, vol. 3, no. 2, pp. 242–252, Jun. 2001.

[7] G. Di Crescenzo, R. Graveman, G. Arce, and R. Ge, "A formal security analysis of approximate message authentication codes," in *Proc. CTA Communications and Networks*, College Park, MD, Apr. 2003, pp. 217–221.

[8] K. M. Bloch and G. R. Arce, "Analyzing protein sequences using signal analysis techniques," in *Computational and Statistical Approaches to Genomics*. Norwell, MA: Kluwer, 2002.

[9] T. A. Brown, *GENETICS A Molecular Approach*, 2nd ed. London, U.K.: Chapman & Hall, 1995.

[10] M. Schneider and S. Chang, "A robust content based digital signature for image authentication," in *Proc. IEEE Intl. Conf. Image Processing*, Lausanne, Switzerland, Sept. 1996.

[11] D.-C. Lou and J.-L. Liu, "Fault resilient and compression tolerant digital signature for image authentication," *IEEE Trans. Consumer Electron.*, vol. 46, no. 1, pp. 31–39, Feb. 2000.

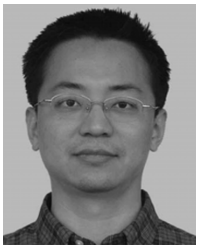
[12] E. Chang, M. Kankanhalli, and X. Guan, "Robust image authentication using content based compression," *ACM Multimedia Syst.*, vol. 9, pp. 121–130, Aug. 2003.

[13] C.-Y. Lin and S.-F. Chang, "A robust image authentication method distinguishing JPEG compression from malicious manipulation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 2, pp. 153–168, Feb. 2001.

[14] C.-S. Lu and H.-Y. Liao, "Structural digital signature for image authentication: An incidental distortion resistant scheme," *IEEE Trans. Multimedia*, vol. 5, no. 2, pp. 161–173, Jun. 2003.

[15] N. Memon, P. Vora, B. Yeo, and M. Yeung, "Distortion bounded authentication techniques," in *Proc. SPIE, Security and Watermarking of Multimedia Contents*, vol. 3971, San Jose, CA, Feb. 2000, pp. 164–174.

[16] M. K. Mihcak and R. Venkatesan, "New iterative geometric methods for robust perceptual image hashing," in *Proc. ACM Workshop on Security and Privacy in Digital Rights Management*, Philadelphia, PA, Nov. 2001.



Renwei Ge (S'97) was born in Shanghai, China. He received the B.S. in electrical engineering and M.S. in communication and electronic system both from Shanghai University, Shanghai, China, in 1995 and 1998, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Delaware, Newark, where he is a Research Assistant.

From 1998 to 2000, he was an Assistant Professor with the Department of Electrical and Information Engineering, Shanghai University, focusing mainly on video compression and communications. His research interests include information security in wireless networks, multimedia signal processing, and communications.

Gonzalo R. Arce (F'02) received the Ph.D. degree from Purdue University, W. Lafayette, in 1982.

Since 1982, he has been with the faculty of the Department of Electrical and Computer Engineering, University of Delaware, Newark, where he is the Charles Black Evans Distinguished Professor and Department Chairman. His research interests include statistical and nonlinear signal processing, multimedia security, electronic imaging, and signal processing for communications and networks. He is coauthor of the textbooks *Digital Halftoning* (Marcel Dekker, 2001), *Nonlinear Signal Processing and Applications* (CRC Press, 2003), and *Nonlinear Signal Processing: A Statistical Approach* (Wiley, 2004). He holds nine U.S. patents.

Dr. Arce received the NSF Research Initiation Award. He is a Fellow of the IEEE for his contributions on nonlinear signal processing and its applications.

Giovanni Di Crescenzo received the Ph.D. degree in computer science from the University of California, San Diego, La Jolla, and the Ph.D. degree in applied mathematics from the University of Naples, Italy.

He is a Senior Research Scientist at Telcordia Technologies, Piscataway, NJ. In his 14-year research career, his main activity has been in various areas, such as computer, information and network security, cryptography, and computational complexity. He has six patent applications awarded or currently pending, and more than 75 scientific publications in major refereed conferences and journals in his research areas. He is a coauthor of *Contemporary Cryptology* (Boston, MA: Birkhauser).