Hindawi Publishing Corporation
EURASIP Journal on Information Security
Volume 2008, Article ID 195238, 10 pages
doi:10.1155/2008/195238

*Research Article*

# Markov Modelling of Fingerprinting Systems for Collision Analysis

**Neil J. Hurley, Félix Balado, and Guénolé C. M. Silvestre**

*School of Computer Science and Informatics, University College Dublin, Belfield, Dublin 4, Ireland*

Correspondence should be addressed to Neil J. Hurley, neil.hurley@ucd.ie

Received 8 May 2007; Revised 19 October 2007; Accepted 3 December 2007

Recommended by S. Voloshynovskiy

Multimedia fingerprinting, also known as robust or perceptual hashing, aims at representing multimedia signals through compact and perceptually significant descriptors (hash values). In this paper, we examine the probability of collision of a certain general class of robust hashing systems that, in its binary alphabet version, encompasses a number of existing robust audio hashing algorithms. Our analysis relies on modelling the fingerprint (hash) symbols by means of Markov chains, which is generally realistic due to the hash synchronization properties usually required in multimedia identification. We provide theoretical expressions of performance, and show that the use of $M$-ary alphabets is advantageous with respect to binary alphabets. We show how these general expressions explain the performance of Philips fingerprinting, whose probability of collision had only been previously estimated through heuristics.

## 1. INTRODUCTION

Multimedia fingerprinting, also known as robust or perceptual hashing, aims at representing multimedia signals through compact and perceptually significant descriptors (hash values). Such descriptors are obtained through a hashing function that maps signals surjectively onto a sufficiently lower-dimensional space. This function is akin to a cryptographic hashing function in the sense that, in order to perform nearly unique identification from the hash values, perceptually different signals—according to some relevant distance—must lead with high probability to clearly different descriptors. Equivalently, the probability of collision ($P_c$) between the descriptors corresponding to perceptually different signals must be kept low. Differently than in cryptographic hashing, signals that are perceptually close must lead to similar robust hashes. Despite this difference with respect to cryptographic hashing, the probability of collision remains the parameter that determines the "resolution" of a method for identification purposes.

A large number of robust hashing algorithms have been proposed recently. This flurry of activity calls for a more systematic examination of robust hashing strategies and their performance properties. In this paper, we take a step in that direction by examining the probability of collision of a certain general class of robust hashing systems, rather than analyzing a particular method. In its binary alphabet version, the class considered broadly encompasses several existing algorithms, in particular, a number of robust audio hashing algorithms [1–4]. We will show that the $M$-ary alphabet version of the class provides an advantage over the binary version for fixed storage size. In order to keep our exposition simple, other issues such as robustness to distortions or to desynchronization are not considered in this analysis. The study of the tradeoffs brought about by the simultaneous consideration of these issues is left as further work. We must also note that we will be dealing with *unintentional* collisions due to the inherent properties of the signals to be hashed. A related problem not tackled in this paper is the analysis of *intentional* forgeries of signals—perhaps under distortion constraints—in order to maximize the probability of collision.

The class of fingerprinting systems that we will study in this paper can be considered as consisting of two independent blocks. Denoting the multimedia signal to be hashed by a continuous-valued $N$-dimensional vector $\mathbf{x} = (x[1], \dots, x[N])$, in the first *feature extraction block*, a function, $f(\cdot)$, is applied to extract a set of $L$ feature vectors, which we assume to be real-valued with dimension $K$. The feature extraction function is

$$f(\cdot) : \mathbb{R}^N \longrightarrow \mathbb{R}^K \underbrace{\times \cdots \times}_{L-1} \mathbb{R}^K, \tag{1}$$

so that $f(\mathbf{x}) = (\mathbf{D}_1, \ldots, \mathbf{D}_L)$ with $\mathbf{D}_m = (D_m[1], \ldots, D_m[K])$ for $m = 1, \ldots, L$.

The second block can be termed as the *hashing block*, in which the continuous feature vector values are mapped to a finite alphabet of hash symbols, that is, quantized. In many methods, this hashing block is implemented through the application of a scalar hashing function to each scalar feature vector value, which we denote as

$$h(\cdot) : \mathbb{R} \longrightarrow \mathcal{H}, \tag{2}$$

where $\mathcal{H}$ is the alphabet of hash symbols whose size is given by $M \triangleq |\mathcal{H}|$.

In any hashing system, a distance measure must be established in order to determine the closeness between hash values. The commonly used distance for comparing sequences formed by discrete-alphabet symbols is the Hamming distance. This distance is defined as the number of times that symbols with the same index differ in the two sequences. Therefore, when comparing any two $M$-ary symbols their Hamming distance can only take the values 0 or 1.

As already stated, our aim is to investigate the probability of collision—also termed in some works false positive probability—of the general type of system described above, under certain assumptions that we will give next. Given a distance measurement, the probability of collision is simply the probability that the fingerprints (hashes) of two independent signals are closer than some preestablished threshold according to the distance measurement established. Our analysis will rely on the fact that the feature vector values are generally highly correlated, due to the synchronization requirements of a fingerprinting system. This high degree of correlation frees the observer of a segment of $\mathbf{x}$ (or a distorted version of it) from the need to know its exact alignment with the complete original signal used to store the fingerprint during the acquisition process (in which the reference hash is obtained for subsequent comparisons). For example, in the Philips method [5] the features are extracted by processing $\mathbf{x}$ frame-by-frame on a set of heavily overlapped frames, which creates the conditions for our analysis. In the following, we will consider the case in which dependencies within a feature vector can be modelled as a continous-valued, discrete-time *Markov chain*. In particular, we assume that

$$\Pr\{D_m[i] \mid D_m[1], \ldots, D_m[i-1]\} = \Pr\{D_m[i] \mid D_m[i-1]\} \tag{3}$$

for all $m = 1, \ldots, L$. Furthermore, we assume that the process is stationary, that is, with statistics independent of $i$. We will also focus without loss of generality on one particular element $m$ of the feature vector. Hence, we will write the relevant random variables of the feature vector as $D$ and $D'$ to represent the distributions of the feature value at $i$ and $i-1$, respectively, for any $i$, dropping the implicit index $m$.

We characterize next the Markov chain of the hash symbols. Define $F \triangleq h(D)$ to be the discrete hash symbol generated by application of the hashing function to a particular element of the feature vector. We will assume that the sequence $F[i]$ forms a discrete-valued, discrete-time Markov chain, with transition probabilities defined by

$$\pi_{s,r} \triangleq \Pr\{F = k_s \mid F' = k_r\} \tag{4}$$

for all the $M^2$ pairs $(k_s, k_r) \in \mathcal{H}^2$.

Finally note that, although methods which deal with real-valued fingerprints could be deemed in principle to belong to this class (using very large values of $M$), they rely on the use of mean square error distances instead of the Hamming distance. Thus, their study is not covered by the class of methods studied here.

### *Notation*

Lowercase boldface letters such as $\mathbf{x}$ represent column vectors, while matrices are represented by upper case Roman letters such as X. diag($\mathbf{x}$) is a matrix with the elements of $\mathbf{x}$ in the diagonal and zero elsewhere. The symbols I and O denote the identity and the all-zero matrices, respectively, whereas $\mathbf{1}$ denotes an all-ones vector, all of suitable size depending on the context. tr($X$) denotes the trace of X. The vec($\cdot$) operator stacks sequentially the columns of an $n \times m$ matrix into an $nm \times 1$ column vector. The symbol $\otimes$ denotes the Kronecker (or direct) product of two matrices, and $\odot$ denotes their Hadamard (component-wise) product. Finally, $\delta_{ij}$ denotes the Kronecker delta function.

## 2. PROBABILITY OF COLLISION

We firstly define $s$ as the amount of bits required to store a single $M$-ary hash symbol, that is,

$$s \triangleq \log_2 M. \tag{5}$$

To fix a point of operation, we consider hash sequences of $n/s$ symbols (assumed integer) which have fixed bit size $n$ (storage size). We investigate the probability of collision between two such independent sequences of symbols generated from the Markov chain with $M \times M$ transition matrix $\Pi \triangleq \{\pi_{s,r}\}$, whose elements are defined in (4). Note that $\Pi$ is a column-stochastic matrix, so that $\mathbf{1}^T \Pi = \mathbf{1}^T$.

The probability of collision is simply the probability that two such hash sequences are closer than a given threshold under the distance measure established. Write $d_n$ to represent the Hamming distance between the sequences. Let $\gamma n/s$ be the Hamming distance below which we consider two sequences of storage size $n$ bits to be identical, with $0 \leq \gamma < 1$ and assuming $\gamma n/s$ integer for simplicity. Using this threshold, the probability of collision between two sequences of storage size $n$ is

$$P_c = \Pr\{d_n \leq \gamma n/s\}. \tag{6}$$

In order to approximate this probability, observe that for any two $n/s$-length sequences of symbols their overall Hamming distance is

$$d_n = \sum_{i=1}^{n/s} d[i] \tag{7}$$

with $d[i]$ the Hamming distance between the $i$th elements of the two sequences. If the random variables $d[i]$ were independent, we could apply the central limit theorem (CLT) to $d_n$ for large $n$, in order to compute the probability (6). Although there are short-term dependencies created by the Markov chain, these vanish in the long term. Then we may invoke a broader version of the CLT for locally correlated signals [6]. In summary, the result in [6] states that, provided the second and third moments of $|d[i]|$ are bounded, then $\sum d[i]$ tends to the normal distribution. Finally, notice that $d_n$ is discrete, and then applying the CLT entails approximating a distribution with support in the positive integers using a distribution with support in the whole real line.

Assuming that the distribution of $d_n$ may be approximated by a Gaussian for large $n$, we only need its mean $\mathrm{E}\{d_n\}$ and variance $\mathrm{V}\{d_n\}$ to characterize it. The probability of collision can then be approximated as

$$P_c \approx \mathcal{Q}\left(\frac{\mathrm{E}\{d_n\} - \gamma n/s}{\sqrt{\mathrm{V}\{d_n\}}}\right) \tag{8}$$

with $\mathcal{Q}(x) \triangleq (1/\sqrt{2\pi})\int_x^\infty \exp(-\xi^2/2)d\xi$. We tackle the computation of the statistics required for this approximation in Section 3, and particular cases in Section 5.

Alternatively, the exact computation of (6) involves enumerating all cases generating a Hamming distance lower than or equal to $\gamma n/s$, that is,

$$P_c = \sum_{k=0}^{\gamma n/s} \Pr\{d_n = k\}. \tag{9}$$

We investigate this direct approach in Section 4. Finally, in Section 6 we propose a Chernoff bound to $P_c$, which is useful when the CLT assumption is not accurate or when the exact computation presents computational difficulties.

## 3. MEAN AND VARIANCE OF HAMMING DISTANCE

In this section, we derive the mean and variance of the Hamming distance using the Markov chain of symbol transitions $\Pi$, defined by (4). To proceed, we assume that $\Pi$ represents an irreducible, aperiodic Markov chain.

We denote as $\mathbf{v}_i \in \mathcal{H}^2$ the pair of simultaneous values of two independent hash sequences at time $i$. The Hamming distance between the elements of $\mathbf{v}_i$ is denoted by $d(\mathbf{v}_i)$ such that $d(\cdot) : \mathcal{H}^2 \to \{0,1\}$. Also, for convenience we denote the nonnegative integer associated with the concatenation of the bit representation of the two components of $\mathbf{v}_i$ by $c(\mathbf{v}_i)$. For instance, with $M = 4$, a possible value of $\mathbf{v}_i$ is $(1,3)$; in this particular case, $d(\mathbf{v}_i) = 1$ and $c(\mathbf{v}_i) = 7$, as the bit representation of the components is 01 and 11, respectively. We define next the $M^2 \times 1$ vector $\boldsymbol{\mu}_i$ with components $\Pr\{\mathbf{v}_i = \mathbf{h}\}$, for all possible $M^2$ values of $\mathbf{h} \in \mathcal{H}^2$ sorted in natural order, that is, according to $c(\mathbf{h})$. The pairs thus defined constitute a new Markov chain with column-stochastic transition matrix $\mathrm{B} \triangleq \Pi \otimes \Pi$, with $\otimes$ the Kronecker product. Therefore,

$$\boldsymbol{\mu}_i = \mathrm{B}\boldsymbol{\mu}_{i-1} = \mathrm{B}^{i-1}\boldsymbol{\mu}_1, \tag{10}$$

for all indices $i > 1$. Denote the equilibrium distribution of this Markov chain as $\boldsymbol{\mu}$; then

$$\mathrm{B}\boldsymbol{\mu} = \boldsymbol{\mu}, \qquad \mathrm{B}^i \longrightarrow \boldsymbol{\mu}\mathbf{1}^T \quad \text{as } i \longrightarrow \infty. \tag{11}$$

If B is symmetric, then the symbols are equally likely in equilibrium and $\boldsymbol{\mu} = 1/M^2\mathbf{1}$.

Some more definitions will be required in order to formalize the derivation of the probabilities associated with a given Hamming distance sequence. Firstly, we define two indicator vectors $\mathbf{i}_0$ and $\mathbf{i}_1$, both of size $M^2 \times 1$. The elements of the vector $\mathbf{i}_k$ are defined to be all zeros except for those elements at positions in $\boldsymbol{\mu}$ such that $\Pr\{\mathbf{v} = (v_1, v_2)\}$ corresponds to a pair with Hamming distance $d(v_1, v_2) = k$, which are set to 1. It is easy to see that $\mathbf{i}_0 = \mathrm{vec}(\mathrm{I})$ and $\mathbf{i}_1 = \mathrm{vec}(\mathbf{1}\mathbf{1}^T - \mathrm{I})$. Now, defining $\boldsymbol{\beta}_i \triangleq (\Pr\{d[i] = 0\}, \Pr\{d[i] = 1\})^T$, we can write the distribution of elemental Hamming distances at the index $i$ as

$$\boldsymbol{\beta}_i^T = \left(\mathbf{i}_0^T\boldsymbol{\mu}_i, \mathbf{i}_1^T\boldsymbol{\mu}_i\right). \tag{12}$$

Observe next that the element at the position $(n, m)$ of the matrix $\mathrm{B}^{j-i}\mathrm{diag}(\boldsymbol{\mu}_i)$, with $j > i$, gives the joint probability $\Pr\{\mathbf{v}_j = c^{-1}(n-1), \mathbf{v}_i = c^{-1}(m-1)\}$ with $c^{-1}(\cdot)$ the unique inverse of $c(\cdot)$. Using this matrix, we can write the joint probability of a pair of elemental distances as

$$\Pr\{d[j] = k, d[i] = l\} = \mathbf{i}_k^T\mathrm{B}^{j-i}\mathrm{diag}(\boldsymbol{\mu}_i)\mathbf{i}_l \tag{13}$$

with $j > i$.

Using the probabilities (12) and (13), we can derive the mean and variance of the Hamming distance between two independent hash sequences of $n/s$ symbols, *assuming that the process starts in the equilibrium distribution* (11). This is tantamount to assuming $\boldsymbol{\mu}_1 = \boldsymbol{\mu}$, in which case $\boldsymbol{\mu}_i = \boldsymbol{\mu}$ and $\boldsymbol{\beta}_i = \boldsymbol{\beta} \triangleq [\mathbf{i}_0, \mathbf{i}_1]^T\boldsymbol{\mu}$, that is, we can drop the index $i$ and write $\Pr\{d[i] = k\} = \Pr\{d = k\}$. When the initial symbol is chosen with uniform probability from $\mathcal{H}$ this condition holds if the transition matrix is symmetric. Even if all values for the initial symbol are not equiprobable in reality, the assumption is not too demanding whenever convergence to equilibrium is fast. We investigate a more general case for binary hashes in Section 5.

Noting that (7) is a sum of dependent variables, we have

$$\mathrm{E}\{d_n\} = \sum_{i=1}^{n/s} \mathrm{E}\{d[i]\}, \tag{14}$$

$$\mathrm{V}\{d_n\} = \sum_{i=1}^{n/s} \mathrm{E}\{d^2[i]\} + 2\sum_{j>i} \mathrm{E}\{d[i]d[j]\} - \mathrm{E}^2\{d_n\}. \tag{15}$$

Notice that, as $d^2[i] = d[i]$ because the Hamming distance only takes values in $\{0, 1\}$, the first summand in (15) is just (14). We compute next the different summands required to obtain $\mathrm{E}\{d_n\}$ and $\mathrm{V}\{d_n\}$. Denote the equilibrium mean and variance of $d[i]$ as $\mathrm{E}\{d\}$ and $\mathrm{V}\{d\}$, respectively. The aforementioned mean and second moment are given by

$$\mathrm{E}\{d\} = \Pr\{d = 1\} = \mathbf{i}_1^T \boldsymbol{\mu}, \tag{16}$$

where we have used (12) and the equilibrium assumption. Hence (14) is given by

$$\mathrm{E}\{d_n\} = \frac{n}{s}\mathrm{E}\{d\}. \tag{17}$$

Next, consider the sum of the elemental distance covariances. If the elemental distances were independent, we would have

$$\mathrm{E}\left\{\sum_{j>i} d[i]d[j]\right\} = \sum_{j>i} \mathrm{E}\{d[i]\}\mathrm{E}\{d[j]\} = \frac{n(n-s)}{2s^2}\mathrm{E}^2\{d\}. \tag{18}$$

Taking into account the dependencies, we have instead,

$$\mathrm{E}\left\{\sum_{j>i} d[i]d[j]\right\} = \sum_{j>i} \Pr\{d[i] = 1, d[j] = 1\}. \tag{19}$$

Using next (12), (13), and the equilibrium assumption we can compute (19) as

$$\mathrm{E}\left\{\sum_{j>i} d[i]d[j]\right\} = \mathbf{i}_1^T\left(\sum_{j>i}\mathrm{B}^{j-i}\right)\mathrm{diag}(\boldsymbol{\mu})\mathbf{i}_1. \tag{20}$$

In Appendix A, we develop this expression to show that the variance (10) of the Hamming distance between two *n*/*s*-length hash sequences is

$$\mathrm{V}\{d_n\} = \frac{n}{s}\mathrm{V}\{d\} + 2\mathbf{i}_1^T\mathrm{G}\,\mathrm{diag}(\boldsymbol{\mu})\mathbf{i}_1 \tag{21}$$

with G given by (A.9).

## 4. THE STOCHASTIC PROCESS OF ELEMENTAL DISTANCES

In this section, we will investigate the stochastic process of elemental distances, that is, the process that generates the sequence $\{d[1], d[2], \ldots, d[n]\}$. Through an analysis of this process, we arrive at a full expression for the probability of collision, which is exact in the case of binary hashing sequences with symmetric transition matrices. This is possible because, as we will show, the elemental distance process is itself a Markov chain when $s = 1$ and the transition matrix is symmetric. Even for the case $s > 1$, we note that the elemental distance process is well approximated by a Markov chain, and then the expression obtained for the probability of collision can be interpreted as a good approximation to the true collision probability.

To understand the process of elemental distances, $\{d[1], d[2], \ldots, d[n]\}$, we consider the conditional probability of $d[i+1]$ given $d[i]$. Define the matrix A with components $a_{kl} \triangleq \Pr\{d[i+1] = k-1 \mid d[i] = l-1\}$. From (12) and (13) we have that

$$a_{kl} = \frac{\mathbf{i}_{k-1}^T \mathrm{B}\,\mathrm{diag}(\boldsymbol{\mu}_i)\mathbf{i}_{l-1}}{\Pr\{d[i] = l-1\}} = \frac{\mathbf{i}_{k-1}^T(\Pi \otimes \Pi)\mathrm{diag}(\boldsymbol{\mu}_i)\mathbf{i}_{l-1}}{\mathbf{i}_{l-1}^T\boldsymbol{\mu}_i}. \tag{22}$$

Define $\Psi_i$ as the matrix such that $\boldsymbol{\mu}_i = \mathrm{vec}\,\Psi_i$. Using $\mathbf{i}_o = \mathrm{vec(I)}$, note that $\mathrm{diag}(\boldsymbol{\mu}_i)\mathbf{i}_0 = \mathrm{vec}(\Psi_i \odot \mathrm{I})$, where $\odot$ is the Hadamard product. Now using the identity $(\mathrm{vec\,P})^T(\Pi \otimes \Pi)(\mathrm{vec\,Q}) = \mathrm{tr}\,\mathrm{Q}\Pi^T\mathrm{P}^T\Pi$ for any matrices P and Q of appropriate size [7], we have that

$$a_{11} = \frac{\mathrm{tr}[(\Psi_i \odot \mathrm{I})\Pi^T\Pi]}{\mathrm{tr}[\Psi_i \odot \mathrm{I}]}. \tag{23}$$

Equation (23) represents a weighted sum of the diagonal elements of $\Pi^T\Pi$, with the weights depending on $\boldsymbol{\mu}_i$ and summing to 1. Similarly, using $\mathbf{i}_1 = \mathrm{vec}(\mathbf{1}\mathbf{1}^T - \mathrm{I})$ and $\mathrm{diag}\,(\boldsymbol{\mu}_i)\mathbf{i}_1 = \mathrm{vec}(\Psi_i - \Psi_i \odot \mathrm{I})$, we have

$$a_{12} = \frac{\mathrm{tr}[(\Psi_i - \Psi_i \odot \mathrm{I})\Pi^T\Pi]}{\mathrm{tr}[\Psi_i - \Psi_i \odot \mathrm{I}]}. \tag{24}$$

Note that (24) is a weighted sum of the off-diagonal elements of $\Pi^T\Pi$ with weights depending on $\boldsymbol{\mu}_i$ and summing to one. The remaining two components of A are given by $a_{21} = 1 - a_{11}$ and $a_{22} = 1 - a_{21}$.

It follows that, whenever the diagonal elements of $\Pi^{\mathrm{T}}\Pi$ are all equal *and* the off-diagonals are all equal, the dependence of A on $\boldsymbol{\mu}_i$ factors from (23) and (24), and A is independent of the time-step *i*. In this case, the process of elemental distances is itself a stationary Markov chain. Let us assume that $\Pi$ has the structure $\Pi = a\mathrm{I} + b\mathrm{S}$ with $\mathrm{S} \triangleq \mathbf{1}\mathbf{1}^T - \mathrm{I}$ and $a + (M-1)b = 1$. In this case, as $\mathrm{S}^2 = (M-2)\mathrm{S} + (M-1)\mathrm{I}$, we can see that $\Pi^T\Pi = \Pi^2 = a'\mathrm{I} + b'\mathrm{S}$ with $a' \triangleq a^2 + b^2(M-1)$ and $b' \triangleq 2ab + b^2(M-2)$. As we have discussed above, this is the structure that allows to cancel the dependence on $\boldsymbol{\mu}_i$ in (23) and (24). For $M = 2$, observe that symmetry implies that $\Pi$ is always of the form above, and then the conditions are always fullfilled in that case.

On the other hand, even when the elemental distances do not follow a Markov chain, since $\boldsymbol{\mu}_i \rightarrow \boldsymbol{\mu}$, the equilibrium probability, the elemental distance process is well approximated by the Markov chain with transition matrix A obtained by replacing $\Psi_i$ in (23) and (24) with $\Psi$, such that $\mathrm{vec}\,\Psi = \boldsymbol{\mu}$. From now on, we will refer loosely to the *elemental distance Markov chain*, meaning, when appropriate, the Markov chain derived from this approximation.

### 4.1. Probability of collision

Using (23) and (24), define $p \triangleq a_{11}$, the probability of a transition from 0 → 0, and $q \triangleq 1 - a_{12}$, the probability of a transition 1 → 1, in the elemental distance Markov chain. Let $\boldsymbol{\beta}_1 = (\beta_{10}, \beta_{11})^T$ be the initial distribution of the elemental distance. Consider a sequence, $\mathbf{d} = (d[1], \dots, d[n])^T$, such that $d_n = \sum_{i=1}^{n} d[i] = k$. Then there are $k$ positions in $\mathbf{d}$ at which $d[i] = 1$. Presume for the moment that $d[1] = 1$. Starting with a block of ones, $\mathbf{d}$ consists of blocks of ones, interweaved with blocks of zeros. Let $n_0$ be the number of blocks of zeros and $n_1$ be the number of blocks of ones. Consider the case $n_1 = r \geq 1$. Then either $n_0 = r$, in which case, the sequence ends with a block of zeros, or $n_0 = r - 1$ in which case the sequence ends with a block of ones. Given that there are in total $k$ ones in the sequence, it is possible to count the number of different types of transitions that occur in the sequence and hence the probability that this sequence can occur. Indeed, if $\mathbf{D}$ represents the random variable modelling an $n$-bit Hamming distance sequence, then

$$\Pr\{\mathrm{D} = \mathbf{d} \mid d[1] = 1\} = \begin{cases} q^{k-r} p^{n-k-r} (1-q)^r (1-p)^{r-1}, \\ \qquad n_1 = n_0 = r, \\ q^{k-r} p^{n-k-r+1} (1-q)^{r-1} (1-p)^{r-1}, \\ \qquad n_1 = r, n_0 = r - 1. \end{cases} \tag{25}$$

For $l = 0$ and $l = 1$, define $P_l(r) \triangleq \Pr\{d_n = k, n_1 = r \mid d[1] = l\}$. To evaluate $P_1(r)$, we enumerate all the different ways that a sequence $\mathbf{d}$ with $d_n = k$ and $n_1 = r$ can occur. This amounts to counting the number of ways that $k$ ones can be subdivided into $r$ blocks and $n - k$ zeros can be subdivided into $r$ or $r - 1$ blocks. With the blocks constructed, interweaving the blocks creates the sequence $\mathbf{d}$. Indeed, from the total of $k - 1$ possible positions at which the sequence of ones can be split, it is necessary to choose $r - 1$ positions. Hence there are $\binom{k-1}{r-1}$ different ways to select $r$ blocks of ones, and similarly $\binom{n-k-1}{r-1}$ to select $r$ blocks of zeros, and $\binom{n-k-1}{r-2}$ to select $r - 1$ blocks of zeros. Thus,

$$\begin{aligned} P_1(r) = {} & \binom{k-1}{r-1} \binom{n-k-1}{r-1} \\ & \times q^{k-r} p^{n-k-r} (1-q)^r (1-p)^{r-1} \\ & + \binom{k-1}{r-1} \binom{n-k-1}{r-2} \\ & \times q^{k-r} p^{n-k-r+1} (1-q)^{r-1} (1-p)^{r-1}. \end{aligned} \tag{26}$$

Now,

$$\Pr\{d_n = k\} = \sum_{r=1}^{k} \beta_{11} P_1(r) + \beta_{10} P_0(r). \tag{27}$$

Assuming $k < n-k$; $p, q > 0$, using an analogous argument to derive $P_0(r)$ and gathering terms, we arrive at the expression

$$\begin{aligned} \Pr\{d_n = k\} = {} & p^{n-k-1} q^k \sum_{r=0}^{k-1} \binom{k-1}{r} \phi_q^{r+1} \phi_p^r \\ & \times \left\{ \binom{n-k-1}{r+1} \beta_{10} \phi_p + \binom{n-k-1}{r} \beta_{11} \right\} \\ & + p^{n-k} q^{k-1} \sum_{r=0}^{k-1} \binom{n-k-1}{r} \phi_q^r \phi_p^{r+1} \\ & \times \left\{ \binom{k-1}{r} \beta_{10} + \binom{k-1}{r+1} \beta_{11} \phi_q \right\}, \end{aligned} \tag{28}$$

where $\phi_p \triangleq (1-p)/p$ and $\phi_q \triangleq (1-q)/q$.

Expression (28) gives the exact probability of collision when the sequence of elemental distances is a Markov chain. In other cases, it will lead to an approximation. Consequently, the analysis is exact for $s = 1$ and $\Pi$ symmetric, in which case $p\ (= q)$ can be determined easily from $\mathrm{A} = \Pi^2$.

## 5. BINARY HASHES WITH SYMMETRIC TRANSITION MATRIX

In this section, we derive expressions for the particular case $s = 1$ with $\Pi$ symmetric. In this case, some simplifications on the general expressions derived above are possible. Define firstly the $2 \times 2$ matrices

$$\mathrm{H}_{11} \triangleq \frac{1}{2} \mathbf{1}\mathbf{1}^T, \qquad \mathrm{H}_{12} \triangleq \mathrm{I} - \mathrm{H}_{11}. \tag{29}$$

Note that the first matrix is idempotent, that is, $\mathrm{H}_{11}^2 = \mathrm{H}_{11}$, and then so is the second, $\mathrm{H}_{12}^2 = \mathrm{H}_{12}$; a further consequence of the definitions is $\mathrm{H}_{11}\mathrm{H}_{12} = \mathrm{H}_{12}\mathrm{H}_{11} = \mathrm{O}$. Assuming symmetry, then for some $-1 \leq \theta < 1$, we can write the binary transition matrix as

$$\Pi = \mathrm{H}_{11} + \theta \mathrm{H}_{12}. \tag{30}$$

With $\theta$ so defined, it can be checked that as $n \to \infty$, (17) and (21) reduce to

$$\begin{aligned} \mathrm{E}\{d_n\} &= \frac{n}{2}, \\ \mathrm{V}\{d_n\} &= \frac{n}{4} \left( \frac{1+\theta^2}{1-\theta^2} \right) - \frac{\theta^2}{2(1-\theta^2)^2}. \end{aligned} \tag{31}$$

While (31) holds under the assumption that the distribution of $\boldsymbol{\beta}_1$ is the equilibrium distribution, it is also possible to derive the exact mean and variance of $d_n$ from an arbitrary initial distribution. This case is interesting, since, although the symbol sequences are assumed to be generated from independent sources, at the application level, the first bit of the hash sequence corresponding to the input signal is sometimes aligned with that of the hash sequences in the database. We can handle this scenario by assuming that the distance between the initial pair of bits is zero.

Before proceeding, note that the transition matrix for the elemental distance process is $\mathrm{A} = \Pi^2$ and, from (30), we can write

$$\mathrm{A} = \mathrm{H}_{11} + \theta^2 \mathrm{H}_{12}. \tag{32}$$

### 5.1. *Exact mean and variance*

With $\boldsymbol{\beta}_1 = (\beta_{10}, \beta_{11})^T$, as before, the initial distribution of the elemental distances, it is convenient to define the vectors $\mathbf{h}_1 \triangleq (1/2)(1,1)^T$ and $\mathbf{h}_2 \triangleq (1/2)(1,-1)^T$ and write $\boldsymbol{\beta}_1 = \mathbf{h}_1 + \psi \mathbf{h}_2$ with

$$\psi \triangleq \beta_{10} - \beta_{11}. \tag{33}$$

Note that $\mathrm{H}_{1i}\mathbf{h}_j = \delta_{ij}\mathbf{h}_j$ and $\mathbf{h}_i^T\mathbf{h}_i = 1/2$. Following the same argument as previously, and defining $\mathbf{e}_1 \triangleq \mathbf{h}_1 - \mathbf{h}_2$, we obtain analogous expressions to (16) and (20) for this case as follows:

$$\mathrm{E}\{d_n\} = \sum_{i=1}^{n} \mathbf{e}_1^T \mathrm{P}^{2i-2} \boldsymbol{\beta}_1, \tag{34}$$

$$\sum_{j>i} \mathrm{E}\{d[i], d[j]\} = \sum_{j>i} \mathbf{e}_1^T \Pi^{2(j-i)} \mathrm{diag}(\Pi^{2i-2}\beta_1)\mathbf{e}_1. \tag{35}$$

The summands in (34) are sums of terms of the form $\mathbf{h}_u^T\mathrm{H}_{1v}\mathbf{h}_w$, which are nonzero only when $u = v = w$. Furthermore, since the coefficient of $\mathrm{H}_{12}$ in $\Pi$ is $\theta$, it follows that the coefficient of $\mathrm{H}_{12}$ in $\Pi^{2i-2}$ is $\theta^{2i-2}$. Hence, summing the geometric series,

$$\mathrm{E}\{d_n\} = \sum_{i=1}^{n} \left(\mathbf{h}_1^T\mathrm{H}_{11}\mathbf{h}_1 - \psi\theta^{2i-2}\mathbf{h}_2^T\mathrm{H}_{12}\mathbf{h}_2\right) = \frac{n}{2} - \frac{\alpha}{2}\psi, \tag{36}$$

where

$$\alpha \triangleq \frac{1-\theta^{2n}}{1-\theta^2}. \tag{37}$$

On the other hand, the summands in (35) are sums of terms of the form $\mathbf{h}_p^T\mathrm{H}_{1q}\mathrm{diag}(\mathrm{H}_{1u}\mathbf{h}_v)\mathbf{h}_w$, which are nonzero only when $u = v$ and $p = q$, in which case they take the value $\mathbf{h}_p^T\mathrm{diag}\,(\mathbf{h}_u)\mathbf{h}_w$. Now, observe that $\mathrm{diag}\,(\mathbf{h}_1)\mathbf{h}_w = \mathbf{h}_w/2$ and $\mathrm{diag}\,(\mathbf{h}_2)\mathbf{h}_w = \mathbf{h}_{3-w}/2$. Hence, (35) reduces to a sum over four terms, $T_1, T_2, T_3$, and $T_4$, where

$$\begin{aligned}
T_1 &= \mathbf{h}_1^T\mathrm{H}_{11}\mathrm{diag}(\mathrm{H}_{11}\mathbf{h}_1)\mathbf{h}_1 = \frac{1}{4},\\
T_2 &= -\mathbf{h}_1^T\mathrm{H}_{11}\mathrm{diag}\Big(\theta^{2(i-1)}\psi\mathrm{H}_{12}\mathbf{h}_2\Big)\mathbf{h}_2 = -\frac{1}{4}\theta^{2(i-1)}\psi,\\
T_3 &= \mathbf{h}_2^T\theta^{2(j-i)}\mathrm{H}_{12}\mathrm{diag}(\mathrm{H}_{11}\mathbf{h}_1)\mathbf{h}_2 = \frac{1}{4}\theta^{2(j-i)},\\
T_4 &= -\mathbf{h}_2^T\theta^{2(j-i)}\mathrm{H}_{12}\mathrm{diag}\Big(\theta^{2(i-1)}\psi\mathrm{H}_{12}\mathbf{h}_2\Big)\mathbf{h}_1 = -\frac{1}{4}\theta^{2(j-1)}\psi.
\end{aligned} \tag{38}$$

In Appendix B, we use (38) to show that the variance of a symmetric binary hash is

$$\mathrm{V}\{d_n\} = \frac{n}{4}\left(\frac{1+\theta^2}{1-\theta^2}\right) - \frac{\alpha\theta^2}{2(1-\theta^2)} - \frac{\alpha^2\psi^2}{4}. \tag{39}$$

Noting that $\alpha \to (1-\theta^2)^{-1}$ as $n \to \infty$, this expression coincides with (31) as $n \to \infty$ when $\psi = 0$.

## 6. CHERNOFF BOUNDING

For large $n$ and small probabilities the CLT can exhibit large deviations from the true probabilities. This is due to the fact that the CLT gives an approximation based only on the two first moments of the real distribution. Also, the exact computation (28) can run into numerical difficulties due to the combinatorials involved. Then, it is interesting to see what can be obtained by means of Chernoff bounding on (6). Apart from the interest of a strict upper bound, this strategy also provides the error exponent followed by the integral of the tail of the distribution of $d_n$.

The Chernoff bound on the probability of collision is given by

$$\begin{aligned}
P_c &\leq \min_{\xi>0} \mathrm{E}\Big\{\exp\left(-\xi(d_n - \gamma n)\right)\Big\}\\
&= \min_{\xi>0} \exp(\xi\gamma n)\cdot\mathrm{E}\{\exp(-\xi d_n)\}.
\end{aligned} \tag{40}$$

The expectation in (40) cannot be expanded as a product of elemental expectations due to the implicit dependencies. However, using the transition matrix A of the elemental distance Markov chain and defining $\boldsymbol{\sigma} \triangleq (1\,\exp(-\xi))^T$, we can efficiently compute it as

$$E\{\exp(-\xi d_n)\} = \boldsymbol{\sigma}^T(\mathrm{A}\,\mathrm{diag}(\boldsymbol{\sigma}))^{(n/s)-1}\boldsymbol{\beta}_1. \tag{41}$$

It is not possible to optimize this expression analytically in closed-form. Nonetheless, numerical optimization can be easily undertaken, as (41) is just a weighted sum of powers of $\exp(-\xi)$.

## 7. EMPIRICAL RESULTS

Matlab source code and data assoicated with the empirical results given below can be downloaded from http://www.ihl.ucd.ie.

### 7.1. *Synthetic Markov chains*

To test the validity of the expressions presented and the accuracy of the CLT approximation, random binary and 4-ary hash sequences were drawn from the Markov chain model. For the binary case, the transition matrix $\Pi$ in (30) is used with $\theta = 0.8$. The generator matrix used for the 4-ary hashes used $\Pi_4 \triangleq \Pi \otimes \Pi$ (note: no relationship with B here). The initial hash symbols were drawn from the equilibrium (uniform) distribution. This corresponds to 4-ary sequences generated by concatenation of binary pairs. The collision probability was measured empirically, using $1.9 \times 10^6$ trials in the binary case and $4.9 \times 10^7$ trials in the 4-ary case. In Figure 1, these empirical probabilities are plotted against the CLT approximation, using the mean and variance given by (17) and (21), respectively. Also shown is the theoretical expression, calculated as $\sum_{k=0}^{\lceil \gamma n/s \rceil} \Pr\{d_n = k\}$ using (28) and the elemental distance Markov chain. This demonstrates the accuracy

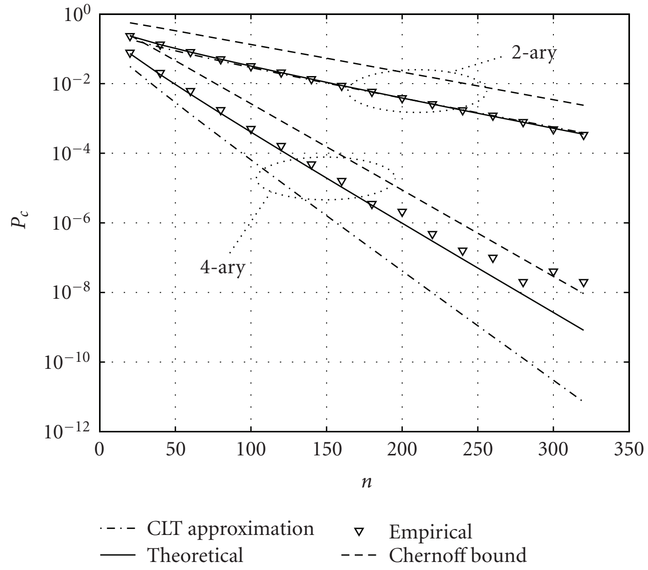

FIGURE 1: Probability of collision for independent hash sequences generated from the Markov chain with transition matrices $\Pi$ given by (30) with $\theta = 0.8$ (binary case) and $\Pi \otimes \Pi$ (4-ary case), plotted against the storage size $n$. Collisions are determined by the threshold $\gamma n/s$ in expression (6) with $\gamma = 0.3$.

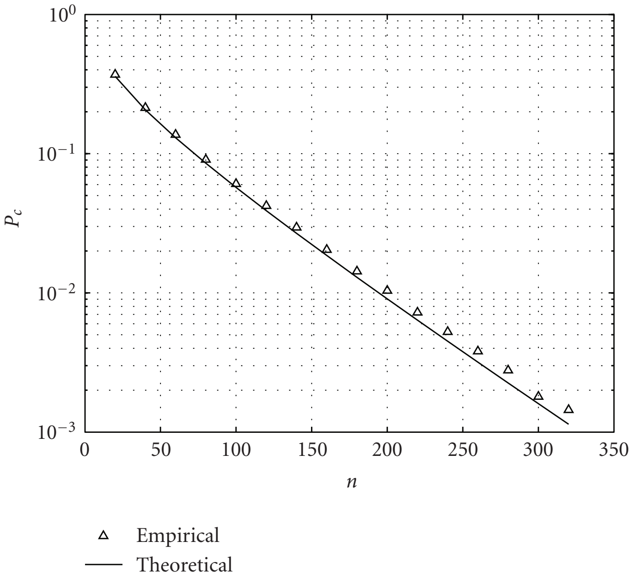

FIGURE 2: The empirical probability of collision of the Philips method is plotted against storage size $n$ and compared with the theoretical expression (28). The theoretical plot uses a binary transition matrix with $p_\Delta(m)$ calculated using (42) and the correlation coefficient $\rho_\Delta(m)$ determined empirically from hash sequence data. Hashes are generated from normally distributed i.i.d input signals. Each frame corresponds to 0.37 seconds of a 44.1 kHz signal.

of the elemental distance Markov chain approximation for 4-ary hashes.

The CLT approximation has good agreement in the binary case for $n > 20$, but is significantly less accurate for 4-ary hashes. This is due to the fact that in the second case, the pdf of $d_n$ is significantly skewed as zero distances are more likely to happen. Due to this, the CLT approximation underestimates the tail of the true distribution. The Chernoff bound, also shown in Figure 1, follows the same shape as the exact distribution and is tighter for high values of $n$ than the CLT approximation.

### 7.2. The Philips method

We show in this subsection how the Markov modelling that we have described is applicable to the hashing method proposed by Haitsma et al. [1], commonly known as the Philips method. Moreover we show how previous work on modelling this particular method allows to obtain analytically the parameters of the Markov chain.

In previous work [8], we developed a model that allows the analysis of the performance of the Philips method under additive noise and desynchronisation. Using this model, the transition matrix of the Markov chain associated to the bitstream of the Philips hash can be determined analytically as follows. In [8] we analysed the bit error that results from desynchronization, the lack of alignment between the original framing used in the acquisition stage and the framing that takes place in the identification stage.

In particular, we showed that for a given band (i.e., a particular feature value $D_m$ in this paper) the probability of error for a desynchronization of $k$ indices in $\mathbf{x}$ is well approximated by

$$p_k(m) \triangleq \frac{1}{\pi} \arccos\left(\rho_k(m)\right), \qquad (42)$$

where $\rho_k$ is the correlation coefficient corresponding to that band and that level of desynchronization. This model was shown therein to give very good agreement with empirical results, even with real audio (and hence nonstationary) input signals.

This same formula can be applied to determine the transition probabilities $0 \rightarrow 1$ or $1 \rightarrow 0$ of the hash bits within a given signal. To this end we only need to observe that two overlapped frames which generate consecutive hash bits are in fact desynchronized by the number of indices where there is no overlap. Denoting this value by $\Delta$ and using $k = \Delta$ in (42), it follows that the binary Markov chain model of Section 5 with $\theta = 2p_\Delta - 1$ can be used to determine the probability of collision for this method. Figure 2 shows the accuracy of this model against empirical results, for a range of hash sequence lengths from $n = 20$ to $n = 320$, with the Philips method applied to the hashing of normally distributed i.i.d input signals.

It is relevant to compare our Markov chain analysis with the collision probability for the Philips method previously examined in [5], in which it is referred to as the "probability of false alarm." Therein, it was assumed that $d[i]$ were mutually independent, leading straightforwardly to $\mathrm{E}\{d_n\} = n/2$ and $\mathrm{V}\{d_n\} = n/4$. With the CLT approximation, from (8), this yields the following expression for the collision probability,

$$P_c \approx \mathcal{Q}\big((1-2\gamma)\sqrt{n}\big), \tag{43}$$

which is independent of the transition probability. To obtain agreement with empirical data, in [5] this expression is modified to account for dependencies using a heuristic correction factor 1/3, that is,

$$P_c \approx \mathcal{Q}\Big(\frac{1}{3}(1-2\gamma)\sqrt{n}\Big). \tag{44}$$

Considering our own CLT approximation (8), we observe that, letting $n \to \infty$ in (36) and (39), the correction factor with respect to the independent case actually tends to

$$\sqrt{\frac{1+\theta^2}{1-\theta^2}}. \tag{45}$$

In the results presented in Figure 2, $\theta = -0.83$ and hence the correction factor for this value of $\theta$ is $1/2.33 \approx 0.43$. In summary, our analysis is able to tackle dependencies without resorting to any heuristics.

### 7.2.1. Real audio signals

We examine the validity of our analysis for real audio signals, by carrying out a collision analysis on hashes generated using the Philips method on three real audio signals already used in [1, 8]: “O Fortuna” by Carl Orff, “Say what you want” by Texas, and “Whole lotta Rosie” by AC/DC (16 bits, 44.1 kHz). Using the parameters of the original algorithm described in [1], a 32-bit block, corresponding to $N_b = 32$ frequency bands, is extracted from each frame. Each frame corresponds to 0.37 seconds of audio and the degree of overlap between frames is 1/32. Hence, from each audio file, a hash block of $N_f \times 32$ bits is extracted, where the number of frames $N_f$ is between 20000 and 30000. Our collision analysis is applied by estimating a single empirical correlation coefficient $\hat{\rho}$ from the entire hash block. We then use our model to predict the probability of collision between hash sequences drawn from the first 200 000 elements of the entire sequence of $N_f \times 32$ bits. The results are shown in Figure 3.

Although our model assumes stationarity, which is clearly not the case for real audio signals, good agreement is found between the model predictions and empirical data. The greatest discrepancy appears in the AC/DC audio and may be due to greater dynamics in this song. To improve the results, we could apply the approach used in [8], where real audio signals are approximated by stationary stretches and apply our model separately to each stretch. While this approach can provide the probability of collision within each stationary stretch, combining these into an overall probability of collision could prove problematic.

## 8. CONCLUSION

We have examined the probability of collision of a certain general class of robust hashing systems that can be described by means of Markov chains. We have given theoretical expressions for the performance of general chains of $M$-ary hashes, by deriving the mean and variance of the distance between independent hashes and applying a CLT approximation for the probability distribution. We have been able to derive an expression for the distribution, which is exact for binary symmetric hashes and gives a very good approximation otherwise. We have confirmed the accuracy of the Gaussian distribution on binary hashes once the hash sequence is sufficiently large. Moreover, we derived the binary transition matrix for the Philips method and showed that the Markov chain model has very good agreement with empirical results for this method. While we have shown that for $M > 2$, $M$-ary chains have an advantage over binary chains from the point of view of collision, higher order alphabets will inevitably lead to a degradation of performance under additive noise and desynchronisation error. The performance tradeoffs that result will be examined in future work.

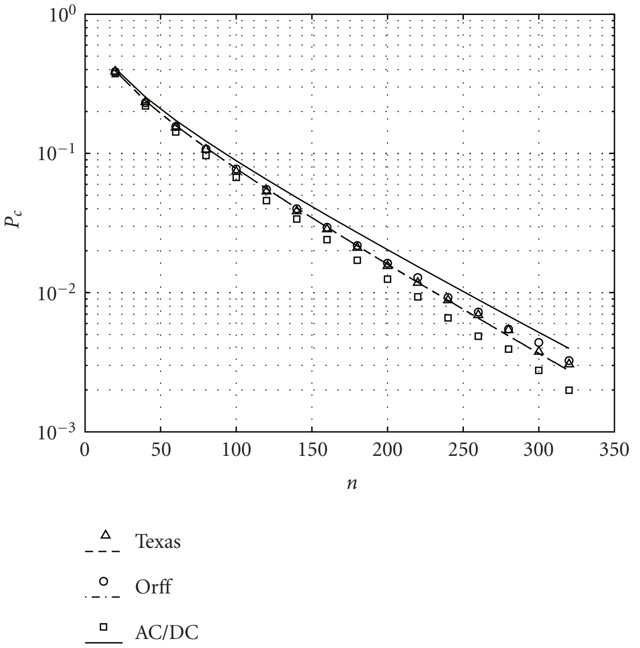

Figure 3: The empirical probability of collision of the Philips method for three real audio signals is plotted against storage size $n$ and compared with the theoretical expression (28). Dots stand for empirical values whereas lines stand for theoretical results.

## APPENDICES

## A. VARIANCE OF AN $M$-ARY HASH SEQUENCE

In this appendix, we detail the computation of (20) in order to obtain $\mathrm{V}\{d_n\}$. Firstly, see that the following identity that holds:

$$\sum_{j>i} \mathrm{B}^{j-i} = \sum_{i=1}^{n/s-1} \Big(\frac{n}{s} - i\Big)\mathrm{B}^i = \frac{n}{s}\sum_{i=1}^{n/s-1} \mathrm{B}^i - \sum_{i=1}^{n/s-1} i\mathrm{B}^i. \tag{A.1}$$

Define $\mathrm{T} \triangleq \sum_{i=1}^{n/s-1} i\mathrm{B}^i$ and $\mathrm{S} \triangleq \sum_{i=1}^{n/s-1} \mathrm{B}^i$. Then

$$\mathrm{T}(\mathrm{I}-\mathrm{B})^2 = \mathrm{B}^{n/s}\Big(\frac{n}{s}(\mathrm{B}-\mathrm{I})-\mathrm{B}\Big)+\mathrm{B}. \qquad \text{(A.2)}$$

Since $\mathbf{1}$ is an eigenvector of B, (I−B) is not invertible. Instead, notice that

$$\mathrm{T}\boldsymbol{\mu} = \sum_{i=1}^{n/s-1} i\boldsymbol{\mu} = \frac{n(n-s)}{2s^2}\boldsymbol{\mu} \qquad \text{(A.3)}$$

which implies

$$\mathrm{TW} = \frac{n(n-s)}{2s^2}\mathrm{W} \qquad \text{(A.4)}$$

with $\mathrm{W} \triangleq \boldsymbol{\mu}\mathbf{1}^T$. Similarly,

$$\mathrm{S}(\mathrm{I}-\mathrm{B}) = \mathrm{B}-\mathrm{B}^{n/s}, \qquad \mathrm{SW} = \frac{n-s}{s}\mathrm{W} \qquad \text{(A.5)}$$

and therefore,

$$\mathrm{S}(\mathrm{I}-\mathrm{B})^2 = \mathrm{B}-\mathrm{B}^2+\mathrm{B}^{n/s+1}-\mathrm{B}^{n/s}. \qquad \text{(A.6)}$$

Using (A.2), (A.4), (A.5), and (A.6), we get

$$\begin{aligned}&\Big(\frac{n}{s}\mathrm{S}-T\Big)\big((\mathrm{I}-\mathrm{B})^2+\mathrm{W}\big)\\ &\quad = \Big(\frac{n-s}{s}\Big)\mathrm{B}-\Big(\frac{n}{s}\Big)\mathrm{B}^2+\mathrm{B}^{n/s+1}+\frac{n(n-s)}{2s^2}\mathrm{W}.\end{aligned} \qquad \text{(A.7)}$$

Observe that, since $\mathrm{WB} = \boldsymbol{\mu}(\mathbf{1}^T\mathrm{W}) = \boldsymbol{\mu}\mathbf{1}^T = \mathrm{W}$,

$$\mathrm{W}\big((\mathrm{I}-\mathrm{B})^2+\mathrm{W}\big) = \mathrm{W}, \qquad \text{(A.8)}$$

which implies that $\big((\mathrm{I}-\mathrm{B})^2+\mathrm{W}\big)^{-1}$ is a right identity of W. Hence, using the definition

$$\mathrm{G} \triangleq \mathrm{B}\Big(\frac{n-s}{s}\mathrm{I}-\frac{n}{s}\mathrm{B}+\mathrm{B}^{n/s}\Big)\big((\mathrm{I}-\mathrm{B})^2+\mathrm{W}\big)^{-1} \qquad \text{(A.9)}$$

(A.7) can be rewritten as

$$\Big(\frac{n}{s}\mathrm{S}-T\Big) = \frac{n(n-s)}{2s^2}\mathrm{W}+\mathrm{G}. \qquad \text{(A.10)}$$

Note also that

$$\mathbf{i}_1^T\cdot\mathrm{W}\,\mathrm{diag}(\boldsymbol{\mu})\cdot\mathbf{i}_1 = \big(\mathbf{i}_1^T\boldsymbol{\mu}\big)^2 = \mathrm{E}^2\{d\}. \qquad \text{(A.11)}$$

Using (A.10) and (A.11), the sum of the covariances (20) is found to be

$$\sum_{j>i}\mathrm{E}\{d[i]d[j]\} = \frac{n(n-s)}{2s^2}\mathrm{E}^2\{d\}+\mathbf{i}_1^T\mathrm{G}\,\mathrm{diag}(\boldsymbol{\mu})\mathbf{i}_1. \qquad \text{(A.12)}$$

As $n \to \infty$,

$$\mathrm{G} \longrightarrow \mathrm{B}\Big(\frac{n-s}{s}\mathrm{I}-\frac{n}{s}\mathrm{B}\Big)\big((\mathrm{I}-\mathrm{B})^2+\mathrm{W}\big)^{-1}+\mathrm{W}. \qquad \text{(A.13)}$$

Using (17) and (A.12) in (15) we finally obtain (21).

## B. VARIANCE OF BINARY SYMMETRIC HASH SEQUENCE

In this appendix, we compute the sum of covariances (35), necessary to obtain the variance of a symmetric binary hash using (15). We will use (38) for this computation. We note firstly the following identities:

$$\begin{aligned}\sum_{j>i}\theta^{2(j-i)} &= \sum_{i=1}^{n-1}(n-i)\theta^{2i},\\ \sum_{j>i}\theta^{2(j-1)} &= \sum_{i=1}^{n-1}i\theta^{2i},\\ \sum_{j>i}\theta^{2(i-1)} &= \sum_{1=1}^{n-1}(n-i)\theta^{2i-2},\\ \sum_{i=1}^{n-1}i\theta^{2i} &= \frac{\theta^2-\theta^{2n}\big(\theta^2+n(1-\theta^2)\big)}{(1-\theta^2)^2}.\end{aligned} \qquad \text{(B.1)}$$

Using the definition in (37), we can write

$$\begin{aligned}\sum_{i=1}^{n-1}i\theta^{2i} &= \frac{\theta^2}{(1-\theta^2)}\alpha-\frac{n\theta^{2n}}{(1-\theta^2)}\\ &= \frac{\theta^2}{(1-\theta^2)}\alpha+n\alpha-\frac{n}{(1-\theta^2)}.\end{aligned} \qquad \text{(B.2)}$$

Therefore,

$$\begin{aligned}\sum_{j>i}\mathrm{E}\{d[i]d[j]\} &= \sum_{j>i}\frac{1}{4}\big(1+\theta^{2(j-i)}\big)-\frac{\psi}{4}\big(\theta^{2(i-1)}+\theta^{2(j-1)}\big)\\ &= \frac{n(n-1)}{8}+\frac{n}{4}\sum_{i=1}^{n-1}\theta^{2i}-\frac{1}{4}\sum_{i=1}^{n-1}i\theta^{2i}\\ &\quad -\frac{\psi}{4}\left(\frac{n}{\theta^2}\sum_{i=1}^{n-1}\theta^{2i}-\frac{1}{\theta^2}\sum_{i=1}^{n-1}i\theta^{2i}+\sum_{i=1}^{n-1}i\theta^{2i}\right).\end{aligned} \qquad \text{(B.3)}$$

Using (37), (B.1), and (37), (B.3) becomes

$$\begin{aligned}\sum_{j>i}\mathrm{E}\{d[i]d[j]\} &= \frac{n(n-1)}{8}+\frac{n}{4}(\alpha-1)-\frac{1}{4}\sum_{i=1}^{n-1}i\theta^{2i}\\ &\quad -\frac{\psi}{4}\left(\frac{n}{\theta^2}(\alpha-1)-\frac{1-\theta^2}{\theta^2}\sum_{i=1}^{n-1}i\theta^{2i}\right).\end{aligned} \qquad \text{(B.4)}$$

Inserting (B.2) into the expression above, we get

$$\begin{aligned}\sum_{j>i}\mathrm{E}\{d[i]d[j]\} &= \frac{n(n-1)}{8}-\frac{n}{4}-\frac{\theta^2\alpha}{4(1-\theta^2)}+\frac{n}{4(1-\theta^2)}\\ &\quad -\frac{\psi}{4}\Big(\frac{n}{\theta^2}\alpha-\frac{n}{\theta^2}-n\alpha\frac{1-\theta^2}{\theta^2}-\alpha+\frac{n}{\theta^2}\Big)\\ &= \frac{n(n-1)}{8}+\frac{\theta^2(n-\alpha)}{4(1-\theta^2)}-\frac{\psi}{4}(n-1)\alpha.\end{aligned} \qquad \text{(B.5)}$$

Finally, inserting (36) and (B.5) into (15), we arrive at (39).

## REFERENCES

[1] J. Haitsma, T. Kalker, and J. Oostveen, "Robust audio hashing for content identification," in *Proceedings of the International Workshop on Content-Based Multimedia Indexing (CBMI '01)*, pp. 117–125, Brescia, Italy, September 2001.

[2] M. K. Mihçak and R. Venkatesan, "A perceptual audio hashing algorithm: a tool for robust audio identification and information hiding," in *Proceedings of the 4th International Workshop on Information Hiding (IHW '01)*, vol. 2137 of *Lecture Notes In Computer Science*, pp. 51–65, Springer, Pittsburgh, Pa, USA, April 2001.

[3] S. Baluja and M. Covell, "Content fingerprinting using wavelets," in *Proceedings of the 3rd European Conference on Visual Media Production (CVMP '06)*, pp. 209–212, London, UK, November 2006.

[4] S. Kim and C. D. Yoo, "Boosted binary audio fingerprint based on spectral subband moments," in *Proceedings of the 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '07)*, vol. 1, pp. 241–244, Honolulu, Hawaii, USA, April 2007.

[5] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR '02)*, pp. 107–115, Paris, France, October 2002.

[6] M. Blum, "On the central limit theorem for correlated random variables," *Proceedings of the IEEE*, vol. 52, no. 3, pp. 308–309, 1964.

[7] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1999.

[8] F. Balado, N. J. Hurley, E. P. McCarthy, and G. C. M. Silvestre, "Performance analysis of robust audio hashing," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 2, pp. 254–266, 2007.