Fast identification algorithms for forensic applications

Beekhof, Fokko Pieter; Voloshynovskyy, Svyatoslav; Koval, Oleksiy; Holotyak, Taras

FAST IDENTIFICATION ALGORITHMS FOR FORENSIC APPLICATIONS

Fokko Beekhof, Sviatoslav Voloshynovskiy, Oleksiy Koval and Taras Holotyak

University of Geneva
Department of Computer Science
7 route de Drize, CH 1227, Geneva, Switzerland

ABSTRACT

In this work a novel fast search algorithm is proposed that is designed to offer improved performance in terms of identification accuracy whilst maintaining acceptable speed for forensic applications involving biometrics and Physically Unclonable Functions. A framework for forensic applications is presented, followed by a review of optimal and existing fast algorithms. We show why the new algorithm has the power to outperform the other algorithms with a theoretic analysis and confirm this using simulations on a large database.

1. INTRODUCTION

The present work targets fast identification systems based on biometrics or Physically Unclonable Functions (PUFs). Both biometrics and PUFs are well-known techniques in forensic applications [1] because of their ability to serve as a unique identifyer for many people and objects.

For reasons of computational complexity, privacy and security, it is undesirable for an identification system to retain the biometrics or PUFs in their full form, yet the performance in terms of successful identification should be minimally reduced, moreover, the reduction in performance should be predictable which requires a thorough understanding and analysis.

The theoretical framework in which the identification can be analysed has been reported in previous work [2], but a quick overview is given here for reference, see Figure 1. A codebook is generated by recording biometrics or PUFs of each person or item to be identified during the enrollment stage, the source of the data is modelled by a continuous memoryless source $\mathbf{X}$ with some distribution $p(\mathbf{x})$. One of the differences with classical communication setups is that the distribution $p(\mathbf{x})$ is given rather than chosen. The receiver observes a noisy version of the biometric or PUFs of a given person or item, where $\mathbf{Y}$ is the observation and the probabilistic mismatch between $\mathbf{X}$ and $\mathbf{Y}$ is modelled

The contact author is S. Voloshynovskiy (email: svolos@unige.ch). http://sip.unige.ch

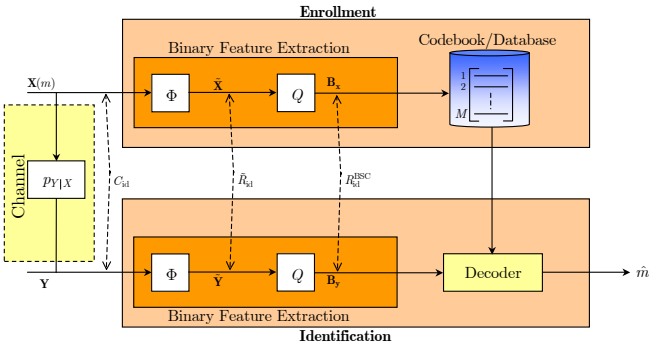

Fig. 1. A schematic overview of a framework for privacy-preserving identification.

by the channel $p(\mathbf{y}|\mathbf{x})$. In a classic communication setup, the receiver has access to the codebook and attempts to find the entry corresponding to the channel output. The identification capacity C_{id} is $I(\mathbf{X};\mathbf{Y})$ where $I(.;.)$ is the mutual information [3]. The second step of the enrollment is to reduce the dimensionality from N to L to extract a so-called *template*. The reduction is accomplished by applying random projections [4]; we use an approximation of a so-called *orthoprojector* $\mathbf{\Psi}$, where each $\Psi_{i,j} \sim \mathcal{N}(0, \frac{1}{N})$. The dimensionality reduction step transforms $\mathbf{X}$ into $\tilde{\mathbf{X}}$ and $\mathbf{Y}$ into $\tilde{\mathbf{Y}}$; as a consequence, the achievable rate at this stage is denoted as $\tilde{R}_{id}$ and is equal to $I(\tilde{\mathbf{X}};\tilde{\mathbf{Y}})$ [5].

For reasons discussed earlier, the original codewords have not been preserved, but only L-length binary templates are derived from the projected data by taking the sign, producing $\mathbf{B_x}$ and $\mathbf{B_y}$. The equivalent channel between these binary vectors follows the Binary Symmetric Channel (BSC) model, resulting in an achievable rate $\tilde{R}_{id}^{BSC} = I(\mathbf{B_x};\mathbf{B_y})$ [6].

Once enrolled, a typical codebook size can be of the order of millions or even billions, which is why fast and reliable identification based on binary data is emerging as an important open problem.

Once the data has been enrolled, the question is how to

query the resulting database in such a way that the performance in terms of the probability of error and complexity is optimized. Unfortunately, the problem of decoding using unstructured codebooks is known to be NP-hard, and thus fast approximative methods are required to create practical systems.

The proposed design can be analyzed within the framework of Locality Sensitive Hashing (LSH) [7, 8], which has at least one form similar to our proposed setup. In that particular case, each hash function is the sign of the projection of the data onto a normal unit vector, which satisfies the requirements of an orthoprojector as in our work. In another variant, the projection is quantized and the index of the bin is the basic hash. Peculiarly, nothing is mentioned about the possible use of Gray codes instead of regular indices, which might reduce the number of bit-errors. A particular case is where quantization yields only one bit, for example by taking the sign, an approach taken in our proposed design as well.

Another scheme that can be analyzed within the framework of LSH is due to Kalker and Haitsma [9]. A key difference is that in LSH a number of complete hashes g are created each by concatenating the outputs of k basic hash-functions, whereas in the original work of Kalker and Haitsma only one complete hash is calculated from audio-data first, which is then divided into blocks. In this work, the hash is created by taking the sign from projections onto Gaussian basis vectors. Under these conditions, each resulting bit can be considered as the output of a single basic hash function, and each block of k bits as one hash function g. In that case, this variant of the Kalker and Haitsma Scheme (KHS) can be considered equivalent to LSH, where the number of functions g is equal to the number of blocks. As in [7], a set of candidates is selected by consulting hash tables corresponding to each block or function g, and from these candidates a final answer is derived.

Practical systems following the framework have access only to a codebook of templates $\mathbf{b_x}$ and the channel output $\mathbf{y}$. The contribution of this paper consists in the evaluation of different fast decoders that perform this identification and the introduction of a new decoder. The new decoder leverages knowledge about the probability of a change of sign in the projected channel output resulting in several advantages: first, the accuracy can be enhanced; second, the time-complexity can be more carefully managed.

The paper is organized as follows: Section 2 reviews different known decoders that achieve optimal performance for the considered channel model and existing fast approximative decoders; additionally it introduces a novel fast approximate decoder. Section 3 contains the result of computer simulations and finally Section 4 concludes the paper.

2. DECODERS

In this paper, we will focus on the Additive White Gaussian Noise (AWGN) channel $p(\mathbf{y}|\mathbf{x})$ where $\mathbf{Y} = \mathbf{X} + \mathbf{Z}$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \sigma_Z^2 \mathbf{I}_N)$. We assume that all M codewords are equiprobable and independent, generated from $\mathbf{X}(m) \sim \mathcal{N}(\mathbf{0}, \sigma_X^2 \mathbf{I}_N)$ where $1 \leq m \leq M$. All vectors in the direct domain are of length N, in the projected and binary domain the length is L where $L < N$.

2.1. Minimum Euclidean Distance Decoder

Under these assumptions, the optimal maximum likelihood decoder reduces to a minimum Euclidean distance decoder:

$$\hat{m} = \arg \min_{1 \leq m \leq M} ||\mathbf{x}(m) - \mathbf{y}||,$$

where $||.||$ denotes the Euclidean distance. This kind of decoding requires $O(MN)$ operations, which is too much for most practical applications.

To introduce the best achievable bound for all approximative decoders, we will use the average probability of error for a Euclidean distance decoder [10]:

$$\begin{aligned} p_e &= 1 - \int_{-\infty}^{\infty} \left(1 - Q\left(\frac{t + \frac{1}{2}||\mathbf{x}||^2}{\sqrt{\sigma_Z^2 ||\mathbf{x}||^2}}\right)\right)^{M-1} \\ &\quad \times \exp\left[-\frac{1}{2\sigma_Z^2 ||\mathbf{x}||^2}(t - \frac{1}{2}||\mathbf{x}||^2)^2\right] dt. \end{aligned} \quad (1)$$

2.2. Minimum Projected Euclidean Distance Decoder

To reduce the complexity as well as to cope with the curse of dimensionality, many practical fast decoders map the original data of length N into a space of a lower dimensionality L. In the scope of this dimensionality reduction, the input data $\mathbf{x}$ is mapped into:

$$\tilde{\mathbf{x}} = \mathbf{\Psi}\mathbf{x},$$

where $\mathbf{x} \in \mathbb{R}^N$, $\tilde{\mathbf{x}} \in \mathbb{R}^L$, $\mathbf{\Psi} \in \mathbb{R}^{L \times N}$ where $L \leq N$. Due to the assumed Gaussian distribution of the basis vectors $\mathbf{\Psi}(i)$ where $i \in \{1, \ldots, L\}$, the resulting vectors will also be Gaussian, i.e., $\tilde{\mathbf{X}} \sim \mathcal{N}(\mathbf{0}, \sigma_X^2 \mathbf{I}_L)$ and $\tilde{\mathbf{Z}} \sim \mathcal{N}(\mathbf{0}, \sigma_Z^2 \mathbf{I}_L)$. The optimal decoder is therefore still the minimum Euclidean distance decoder, to which we will refer as the Minimum Projected Euclidean Distance or "Proj Euclid" decoder. It simply finds the codeword whose projection has the smallest Euclidian distance to the projected channel output by exhaustive search:

$$\begin{aligned} \hat{m} &= \arg \min_{1 \leq m \leq M} ||\tilde{\mathbf{x}}(m) - \tilde{\mathbf{y}}|| \quad (2) \\ &= \arg \min_{1 \leq m \leq M} \tilde{\mathbf{x}}^T(m)\tilde{\mathbf{x}}(m) - 2\tilde{\mathbf{x}}^T(m)\tilde{\mathbf{y}} + \tilde{\mathbf{y}}^T\tilde{\mathbf{y}}. \end{aligned}$$

Under condition that $\tilde{\mathbf{x}}^T(m)\tilde{\mathbf{x}}(m)$ is equal for all m, the "Proj Euclid" decoder is equal to the maximization of the empirical correlation $\rho_{\tilde{\mathbf{X}}(m),\tilde{\mathbf{Y}}} = \tilde{\mathbf{x}}^T(m)\tilde{\mathbf{y}}$.

The complexity of this decoder is $O(ML)$ and the achievable rate is $I(\tilde{\mathbf{X}};\tilde{\mathbf{Y}})$. The performance of this decoder is also defined by Equation (1) with the replacement of $||\mathbf{x}||$ by $||\tilde{\mathbf{x}}||$.

2.3. Minimum Hamming Distance Decoder

Biometric or PUF templates can be obtained from the projected vectors by binarization by taking the sign:

$$b_{x,i}(m) = sign(\mathbf{\Psi}(i)^T\mathbf{x}(m)), \qquad \forall i \in \{1,\ldots,L\}.$$

Any noisy channel output $\mathbf{y}$ derived from a given $\mathbf{x}(m)$ will also be converted into the binary domain, the result being a BSC between $\mathbf{B_x}$ and $\mathbf{B_y}$ with a crossover probability:

$$\bar{p}_b = \frac{1}{\pi}\arccos\rho_{X,Y}, \tag{3}$$

where $\rho_{X,Y} = \pm\sqrt{\frac{\sigma_X^2}{\sigma_X^2+\sigma_Z^2}}$ is the cross-correlation coefficient between $\mathbf{X}$ and $\mathbf{Y}$.

The optimal maximum-likelihood decoder for the BSC reduces to a minimum Hamming distance decoder. Let $d_H(.,.)$ denote the Hamming distance between two binary sequences, then the minimum Hamming distance decoder is:

$$\hat{m} = \arg\min_{1\le m\le M} d_H(\mathbf{b_x}(m),\mathbf{b_y}).$$

Like the previous exhaustive search decoder, its complexity is $O(ML)$, but in contrast to the Euclidean distance decoder it operates on binary data, which is significantly faster in practice. The achievable rate is $I(\mathbf{B_x};\mathbf{B_y})$.

2.4. Locality Sensitive Hashing

A KHS-decoder has been included as a representative of the LSH family of hash functions. The L-length vector is partitioned in several blocks of size k. The idea is that there is a significant chance that in a short block no bit has flipped, which implies that with large probability, it points to the right entry in a corresponding hash table. There are hash tables for each different block rather than one combined table as in the original design by Kalker and Haitsma. The hash tables are consulted for each block to find codewords that have a matching sequence of bits in the same position as the hashed channel output. The codewords that have the highest number of matching blocks form the set of candidates. From these candidates, the winning index $\hat{m}$ is chosen as the one that has the smallest Hamming distance to the binary template of the channel output.

The decoder tries to minimize the Hamming distance, but unlike the reference implementation by exhaustive search, there is a risk that the appropriate codeword does not show in the lookup tables more frequently than all others. The output is thus an approximation of the minimum Hamming distance:

$$\hat{m} \approx \arg\min_{1\le m\le M} d_H(\mathbf{b_x}(m),\mathbf{b_y}).$$

The achievable rate is upper bounded by $I(\mathbf{B_x};\mathbf{B_y})$.

The complexity of the decoder is $O(\frac{L}{k}\frac{M}{2^k})$ and the chance of a block of length k to be error-free is $p_b^C = (1-\bar{p}_b)^k$, which shows the tradeoff between performance in terms of accuracy and runtime: both change exponentially as a function of the block size.

2.5. NP soft Decoder

The LSH-decoder operates purely on binary data, hence its rate is limited by $I(\mathbf{B_x};\mathbf{B_y})$, even though a decoder has access to the full channel output $\mathbf{Y}$. The logical result is that the performance is degraded with respect to the "Proj Euclid" decoder.

In order to move closer to the accuracy of the "Proj Euclid" decoder whilst maintaining acceptable complexity, we propose the "NP soft" decoder, that uses the real-valued (soft) information $\tilde{\mathbf{y}}$.

The "NP soft" decoder is a heuristic-guided backtracking algorithm designed to find the most likely match in the binary codebook by flipping a fraction of the L bits starting with the least reliable ones, inspired by DPLL-solvers for the Satisfiability problem [11, 12]. This covers the most likely original codewords and is therefore an approximation of an ML-decoder. The depth of the tree that is expanded during the recursive search is limited to a number $D \le L$.

If a binary alphabet $\{-1,1\}$ is used to represent the codebook, the following metric is used to assign a score to different items that are found in the codebook during the search:

$$\hat{\rho}_{\tilde{\mathbf{X}}(m),\tilde{\mathbf{Y}}} = \sum_{i=1}^{L} b_{x,i}(m)b_{y,i}\tilde{y}_i^2. \tag{4}$$

This metric is an estimate of the correlation; note that $\tilde{x}(m)$ can be decomposed as $\tilde{x}_i(m) = sign(\tilde{x}_i(m))\,|\tilde{x}_i(m)|$ where $|\tilde{x}_i(m)|$ denotes the magnitude. $\tilde{y}$ can be decomposed similarly, then the correlation can be computed as:

$$\begin{aligned}
\rho_{\tilde{\mathbf{X}}(m),\tilde{\mathbf{Y}}} &= \sum_{i=1}^{L}\tilde{x}_i(m)\tilde{y}_i, \\
&= \sum_{i=1}^{L} sign(\tilde{x}_i(m))|\tilde{x}_i(m)|sign(\tilde{y}_i)|\tilde{y}_i|, \\
&= \sum_{i=1}^{L} -(b_{x,i}(m)\oplus b_{y,i})|\tilde{x}_i(m)||\tilde{y}_i|. \qquad (5)
\end{aligned}$$

Because $\tilde{\mathbf{x}}$ is not available at the decoder, it is estimated as $\tilde{\mathbf{y}}$, applying this estimation in (5) leads to (4), showing that $\hat{\rho}_{\tilde{\mathbf{X}}(m),\tilde{\mathbf{Y}}}$ is an estimator for $\rho_{\tilde{\mathbf{X}}(m),\tilde{\mathbf{Y}}}$. This demonstrates the relation with the "Min Proj" decoder introducted in Section 2.2.

It should be noted that if the depth of the search is properly chosen, only one entry should be encountered whilst flipping bits, assuming that the Hamming distance between the entries is sufficient to guarantee that the hash of the channel output is practically always closest to the hash of the original codeword. The fact that the depth of the search is limited implies that the algorithm produces only an approximation of the estimated value, hence the decoding metric is:

$$\hat{m} \approx \arg \max_{1 \leq m \leq M} \hat{\rho}_{\tilde{X},\tilde{Y}}(\tilde{\mathbf{x}}(m), \tilde{\mathbf{y}})).$$

Instead of searching over all $\mathbf{b}_{\mathbf{x}}(m)$, the algorithm flips bits in $\mathbf{b}_{\mathbf{y}}$ and tests for the existence of the resulting bitstring in the codebook. If a codeword $\mathbf{b}_{\mathbf{x}}(m)$ is found, the score can be computed knowing that each flipped bit $b_{y,i}$ corresponds to a mismatch with $b_{x,i}(m)$. As the existence of an entry in the codebook is sufficient, the security and privacy could be enhanced by storing hashes of the templates rather than the templates themselves, provided that the resulting codebook remains sufficiently sparse to avoid collisions.

The achievable rate is upper bounded by $I(\mathbf{B}_{\mathbf{x}}; \tilde{\mathbf{Y}})$, the complexity is $O(2^D \log_2 M)$, where D is the fixed limit on the depth of the search. To search through the full codebook, one should set $D = L$, but this is unnecessary for the following reason: using the BSC as a model after binarization, a parameter $\bar{p}_b$ can be determined, and then the number of bit flips that occur follows a Binomial distribution with parameters L and $\bar{p}_b$. For any arbitrarily chosen large probability $1 - \epsilon$, the maximum number of bit-flips can be determined through the inverse of the cumulative distribution function.

3. SIMULATION RESULTS

The different decoders were tested using a database of synthetic data that is independently and identically normally distributed, i.e. $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_N)$; a total of $M = 2^{20}$ samples of length $N = 1024$ were produced to form the codebook, the dimensionality is reduced to $L = 32$. The Signal-to-Noise Ratio is defined as SNR $= 10 \log_{10} \frac{\sigma_X^2}{\sigma_Z^2}$.

For the "NP soft" decoder, the value of D is given by:

$$D = \min \left(12, \frac{3}{2} F^{-1}(1 - 10^{-6}, L, \bar{p}_b) \right),$$

where $F^{-1}(.,.,.)$ is the inverse cumulative distribution function of the Binomial distribution with parameters L and $\bar{p}_b$. In case of such a Gaussian setup, the probability of a bit flipping can be computed by developing Equation (3):

$$\bar{p}_b = \frac{1}{\pi} \arccos \sqrt{\frac{1}{1 + \frac{\sigma_Z^2}{\sigma_X^2}}}. \quad (6)$$

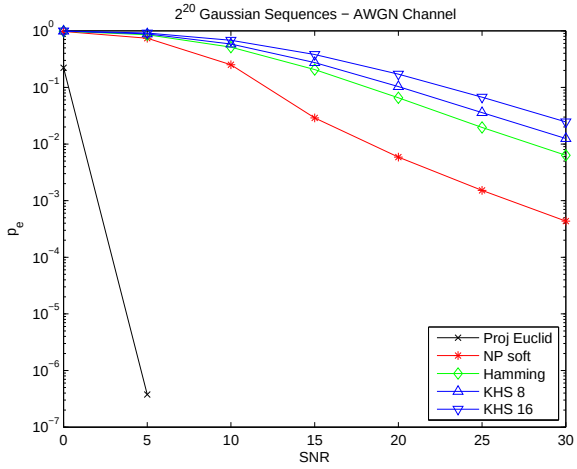

Fig. 2. The probability of error as a function of the SNR for the tested data for $M = 2^{20}$.

The tests have been run with two versions of the KHS decoder, one using $k = 8$ (KHS8) and one where $k = 16$ (KHS16). See Table 1 for a list of the used limits on the search depths.

SNR	0	5	10	15	20	25	30
D	12	12	12	12	9	7	6

Table 1. Search depth of the "NP soft" decoder per SNR.

The probability of error is defined as

$$p_e = \frac{1}{M} \sum_{m=1}^{M} \Pr[\hat{m} \neq m | m] \quad (7)$$

and has been plotted in Figure 2 as a function of the SNR. The data for the minimum Euclidean distance in the projected domain proved too computationally intensive, so Equation (1) has been used to determine the results for that decoder. The "NP soft" decoder outperforms all binary decoders.

Regarding the complexity, the runtime depends very much on the particular platform, which limits our ability to make sensible comparisons to the big-O notations. A plot of the complexity is displayed in Figure 3 for the values used in the simulations. The difference between the complexity of exhaustive search over a codebook of N-length data such as the minimum Euclidean distance or cross-correlations in the direct domain (Exhaustive N) and the complexity of

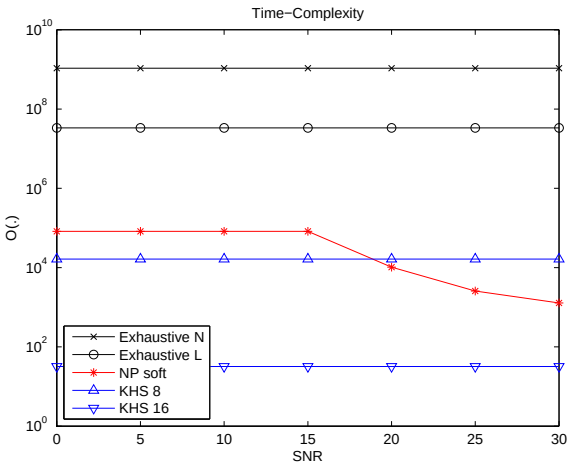

Fig. 3. The order of the time-complexity as a function of the SNR for the tested data.

algorithms using L-length data such as exhaustive search over the minimum Euclidean distance in the projected domain or the Hamming distance in the binary domain (both Exhaustive L) shows the benificial effect of dimensionality reduction. The effectiveness of the fast search algorithms is clearly visible as a significant drop in complexity when compared to both forms of exhaustive search. The "NP-soft" decoder has a complexity comparable to the "KHS" decoders for higher SNRs, but its accuracy is much higher.

4. CONCLUSIONS

We have reviewed a framework for identification based on noisy data such as biometrics and Physically Unclonable Functions. In light of this framework, we have investigated different decoders and introduced an advanced decoder that offers superior performance by approximating the optimal decoder without excessive computational requirements.

Using computer simulations on a large database, the proposed decoder proved to be able to outperform several existing decoders in terms of accuracy, whilst maintaining comparable complexity for SNRs that can realistically be expected in forensic applications.

Acknowledgments

This work is supported by SNF projects 111643 and 1119770.

5. REFERENCES

[1] P. Tuyls, B. Skoric, and T. Kevenaar (Eds.), *Security with Noisy Data: On Private Biometrics, Secure Key Storage and Anti-Counterfeiting*, Springer, 2007.

[2] S. Voloshynovskiy, O. Koval, F. Beekhof, and T. Pun, "Conception and limits of robust perceptual hashing: toward side information assisted hash functions," in *Proceedings of SPIE Photonics West, Electronic Imaging / Media Forensics and Security XI*, San Jose, USA, 2009.

[3] F. Willems, T. Kalker, J. Goseling, and J-P. Linnartz, "On the capacity of a biometrical identification system," in *In: Proc. of the 2003 IEEE Int. Symp. on Inf. Theory*, 2003, pp. 8–2.

[4] J. Fridrich, "Robust bit extraction from images," in *Proceedings ICMCS'99*, Florence, Italy, June 1999, vol. 2, pp. 536–540.

[5] S. Voloshynovskiy, O. Koval, and T. Pun, "Multimodal authentication based on random projections and distributed coding," in *Proceedings of the 10th ACM Workshop on Multimedia & Security*, Oxford, UK, September 22–23 2008.

[6] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley and Sons, New York, 1991.

[7] Aristides Gionis, Piotr Indyk, and Rajeev Motwani, "Similarity search in high dimensions via hashing," in *VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases*, San Francisco, CA, USA, 1999, pp. 518–529, Morgan Kaufmann Publishers Inc.

[8] Mayur Datar, Nicole Immorlica, , Piotr Indyk, and Vahab Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the ACM Symposium on Computational Geometry*. 2004, pp. 253–262, ACM Press.

[9] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system with an efficient search strategy," *Journal of New Music Research*, vol. 32, no. 2, pp. 211–221, 2003.

[10] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*, Prentice Hall, Englewood Cliffs, NJ, 1998.

[11] M. Davis and H. Putnam, "A computing procedure for quantification theory," *J. ACM*, vol. 7, no. 3, pp. 201–215, 1960.

[12] M. Davis, G. Logemann, and D. Loveland, "A machine program for theorem-proving," *Commun. ACM*, vol. 5, no. 7, pp. 394–397, 1962.