POLITECNICO DI TORINO
Repository ISTITUZIONALE

TTP-free asymmetric fingerprinting protocol based on client side embedding

Original
TTP-free asymmetric fingerprinting protocol based on client side embedding / Bianchi, T., Alessandro, P.. - (2014), pp. 3987-3991. (2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) Firenze May 2014) [10.1109/ICASSP.2014.6854350].

Availability:
This version is available at: 11583/2560954 since:

Publisher:
IEEE - INST ELECTRICAL ELECTRONICS ENGINEERS INC

Published
DOI:10.1109/ICASSP.2014.6854350

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

11 October 2026

TTP-FREE ASYMMETRIC FINGERPRINTING PROTOCOL BASED ON CLIENT SIDE EMBEDDING

Tiziano Bianchi

Dept. of Electronics and Telecomm.
Politecnico di Torino
C. Duca degli Abruzzi 24, 10129, Torino, Italy

Alessandro Piva

Dept. of Information Engineering
Università di Firenze
via S. Marta 3, 50139, Firenze, Italy

ABSTRACT

In this paper, we propose a scheme to employ an asymmetric fingerprinting protocol within a client-side embedding distribution framework. The scheme is based on a novel client-side embedding technique that is able to transmit a binary fingerprint. This enables secure distribution of personalized decryption keys containing the Buyer's fingerprint by means of existing asymmetric protocols, without using a trusted third party. Simulation results show that the fingerprint can be reliably recovered by using non-blind decoding, and it is robust with respect to common attacks. The proposed scheme can be a valid solution to both customer's rights and scalability issues in multimedia content distribution.

Index Terms— Fingerprinting, Buyer-Seller watermarking protocol, Client-side embedding, secure watermark embedding.

1. INTRODUCTION

The recent proliferation of various platforms for the distribution of multimedia contents requires the adoption of effective protection measures for preventing copyright violations. In the most common case, distribution tracing is made possible by letting the Seller insert a distinct watermark, called a *fingerprint*, identifying the Buyer, within any copy of data that is distributed. Whenever an unauthorized published content is found, this fingerprint can be used to trace the author of the illegal redistribution [1, 2, 3].

Existing watermarking techniques for multimedia content protection have been developed to face two important practical issues. The customer's rights problem is due to the fact that the distribution server should not know the actual fingerprint embedded into the content, since an accused customer could claim that he/she has been framed by a malicious seller. To cope with this issue, asymmetric fingerprinting schemes [4] have been proposed: here, only the buyer has access to the fingerprinted content; however, if the seller later finds a copy of the content, the buyer can still be identified and proved guilty. Several such protocols suitable for multimedia contents, referred to as Buyer-Seller Watermarking Protocols (BSWP), exist [5, 6, 7, 8]: a special class include those relying only on messages exchanged between buyer and seller, without requiring a dedicated trusted third party (TTP) [9, 10].

The second issue is related to the system scalability. In a classical distribution model, adopted also by the BSWP, individually watermarked copies are generated and distributed by the distribution server to each user. Since both the computational burden due to watermark embedding and the required bandwidth grow linearly with the number of users, in large-scale systems the server could consume a prohibitive amount of resources. An effective solution to this problem is provided by client-side embedding [11]: here, the server distributes the same encrypted copy of the content to all the clients, along with different client-specific decryption keys allowing each user to decrypt a slightly different version of the content, bearing a different watermark. Secure client-side embedding methods suitable for realistic multimedia content have been developed taking into account spread-spectrum watermarking [12], informed embedding [13], and vector quantization [14].

Although client-side embedding provides an elegant solution to the system scalability problem, it still suffers of the customer's rights problem, since the server has access to the decryption keys that carry the client-specific watermarks. Some works [15, 16] have proposed to introduce a TTP managing the distribution of the decryption keys. However, such a TTP can quickly become overloaded, thus hindering the advantages of client-side embedding. To the best of our knowledge, there is no existing solution that incorporates the aforementioned techniques into an asymmetric fingerprinting protocol, solving both the customer's rights problem and the scalability issue.

In this paper, we propose a simple scheme to exploit existing secure asymmetric fingerprinting protocols within a client-side embedding distribution framework. Namely, we modify the client-side embedding technique proposed in [12] so that it can be used to reliably transmit a binary fingerprint, which enables the secure distribution of decryption keys by means of existing TTP-free buyer-seller watermarking protocols. Thanks to the used protocol, the server can distribute personalized decryption keys without knowing the actual fingerprint embedded in each key, which eliminates the need of a TTP. At the same time, since the size of a decryption key is much lower than the size of a multimedia content, and a single key can be used for multiple contents, the complexity of running an existing TTP-free buyer-seller protocol, like e.g. that in [9], for the distribution of the keys is still reasonable.

2. PRELIMINARIES

2.1. Watermarking Model

Given a vector $\mathbf{x} = [x_1, x_2, \ldots, x_M]$, representing either the original host signal samples or, more generally, a set of features of the host signal, and some to-be-hidden information, represented as a binary vector $\mathbf{b} = [b_1, b_2, \ldots, b_L]$, an *embedder* inserts the watermark code $\mathbf{b}$ into the host signal to produce a watermarked signal $\mathbf{y}$, usually making use of a secret key sk to control some parameters of the embedding process and allow the watermark recovery only to authorized users. It is often useful to describe the embedding function by introducing a watermarking signal $\mathbf{w}$, so that the watermarked signal can be expressed as $\mathbf{y} = \mathbf{x} + \mathbf{w}$.

2.2. Homomorphic Cryptosystems

A cryptosystem is said to be *homomorphic* with respect to an operation $\star$ if there exists an operator $\phi(\cdot,\cdot)$ such that for any two plain messages m_1 and m_2, we have:

$$\phi([\![m_1]\!],[\![m_2]\!]) = [\![m_1 \star m_2]\!] \quad (1)$$

where $[\![\cdot]\!]$ denotes the encryption operator. Homomorphic encryption allows to perform a set of operations by working on encrypted data. In particular, an additively homomorphic cryptosystem usually maps an addition in the plaintext domain to a multiplication in the ciphertext domain. Given two plaintexts m_1 and m_2, the following equalities are then satisfied: $[\![m_1]\!] \cdot [\![m_2]\!] = [\![m_1 + m_2]\!]$ and, as a consequence, $[\![m]\!]^a = [\![am]\!]$ where a is a public integer. Additively homomorphic cryptosystems allow then to perform in the encrypted domain additions, subtractions and multiplications with a known (non-encrypted) value (but not division, since it could lead to non integer values), thus providing a way of applying any linear operator in the encrypted domain. A well known additively homomorphic asymmetric encryption scheme was proposed by Paillier [17].

2.3. Asymmetric Fingerprinting

In *asymmetric fingerprinting* [4] the Buyer first commits to a secret that only he/she knows (registration phase), then Buyer and Seller follow a protocol (named Buyer-Seller watermarking protocol) after which only the Buyer receives a copy of the watermarked work. However, if the copy is illegally distributed, the Seller can identify the Buyer from whom the copy originated, and prove it to a Judge by using a proper dispute resolution protocol.

A fundamental building block of asymmetric fingerprinting is a functionality that allows Seller and Buyer to jointly perform watermark embedding, in such a way that the original content $\mathbf{x}$ is a private input of the Seller, whereas the fingerprint data $\mathbf{b}$ and thus the watermark $\mathbf{w}$ are a private input of the Buyer.

2.4. LUT-based Secure Embedding

In the secure embedding proposed by Celik *et al.* in [18, 12], a distribution server generates a long-term master encryption look-up table $\mathbf{E}$ of size T, whose entries, denoted by $\mathbf{E}(0), \mathbf{E}(1), \ldots, \mathbf{E}(T-1)$, are i.i.d. random variables following a Gaussian distribution $\mathcal{N}(0, \sigma_E)$. The LUT $\mathbf{E}$ will be used to encrypt the content to be distributed to the K_U clients. Next, for the k-th client, the server generates a personalized watermark LUT $\mathbf{W}_k$ whose entries follow a Gaussian distribution $\mathcal{N}(0, \sigma_W)$, and builds a personalized decryption LUT $\mathbf{D}_k$ by combining componentwise the master encryption LUT $\mathbf{E}$ and the watermark LUT $\mathbf{W}_k$:

$$\mathbf{D}_k(t) \quad = \quad -\mathbf{E}(t) + \mathbf{W}_k(t) \quad (2)$$

for $t = 0, 1, \ldots, T-1$. The personalized decryption LUTs are then transmitted once to each client over a secure channel. It is worth noting that the generation of the LUTs is carried out just once at the setup phase.

Driven by the content dependent key sk, a set of $M \times R$ values t_{ih} in the range $[0, T-1]$ is generated, where $0 \le i \le M-1$, $0 \le h \le R-1$. Each of the M content features x_i is encrypted by adding R entries of the encryption LUT identified by the indexes $(t_{i0}, \ldots, t_{i(R-1)})$, obtaining the encrypted feature c_i as follows:

$$c_i \quad = \quad x_i + \sum_{h=0}^{R-1} \mathbf{E}(t_{ih}). \quad (3)$$

Joint decryption and watermarking is accomplished by reconstructing with the content dependent key sk the same sequence of indexes t_{ih} and by adding R entries of the decryption LUT $\mathbf{D}_k$ to each encrypted feature c_i:

$$y_{k,i} = c_i + \sum_{h=0}^{R-1} \mathbf{D}_k(t_{ih}) = x_i + \sum_{h=0}^{R-1} \mathbf{W}_k(t_{ih}) = x_i + w_{k,i} \quad (4)$$

where the i-th watermark component is given as the sum of R entries of the LUT $\mathbf{W}_k$. The result of this operation is the watermarked content $\mathbf{y}_k = \mathbf{x} + \mathbf{w}_k$ identifying the k-th user.

3. PROPOSED METHOD

The key idea of the proposed method is that the decryption LUT in (2) can be alternatively seen as the negative version of the encryption LUT watermarked by a proper signal $\mathbf{W}$ corresponding to the watermarking LUT. Hence, existing buyer-seller watermarking protocols can be used to securely distribute personalized decryption LUTs in such a way that the server does not have access to plaintext versions of those decryption LUTs. However, since existing TTP-free protocols require the buyer to be identified by a unique binary fingerprint, the watermarking LUT must be properly modified so as to embed a binary message into the content and guarantee that the embedded message can be reliably decoded from a possibly modified watermarked content.

3.1. Distribution of Personalized Decryption LUTs

Let us assume that the k-th user is identified by the L-bit fingerprint $\mathbf{b}_k$. The fingerprint is encoded using a binary antipodal modulation, yielding the to be transmitted message $\mathbf{m}_k$, where $m_{k,l} = \sigma_W(2b_{k,l} - 1)$, $0 \le l \le L-1$. Hence, the watermarking LUT of the k-th user is obtained as

$$\mathbf{W}_k = \mathbb{G}\mathbf{m}_k \quad (5)$$

where $\mathbb{G}$ is a $T \times L$ encoding matrix. Namely, $\mathbb{G}$ can be thought as the generator matrix of a linear block code over the set of real numbers [19, 20].

Several choices are possible for $\mathbb{G}$. A really simple solution is to use a repetition code, i.e., $\mathbb{G}$ has only one entry equal to one for each row and approximately T/L entries equal to one for each column. Another solution is to generate the elements of $\mathbb{G}$ as i.i.d. Gaussian variables with zero mean and variance $1/L$.

Since the encoding is linear, the personalized decryption LUT $\mathbf{D}_k$ can be obtained in a secure way by using a simple protocol based on an additively homomorphic cryptosystem. Let us assume that by executing a secure buyer-seller protocol like the one described in [10] the Server obtains an encryption of the Client's fingerprint $[\![\mathbf{b}_k]\!]$, encrypted with the Client's public key, together with a proper proof of identity. Thanks to the homomorphic properties of the cryptosystem, the Server can compute the encrypted message as $[\![m_{k,l}]\!] = [\![b_{k,l}]\!]^{2\sigma_W} [\![\sigma_W]\!]^{-1}$. In a similar way, each entry of the Client's personalized LUT can be directly computed in the encrypted domain as

$$[\![\mathbf{D}_k(j)]\!] = [\![\mathbf{E}_k(j)]\!]^{-1} \prod_{l=0}^{L-1} [\![m_{k,l}]\!]^{\mathbb{G}(j,l)}. \quad (6)$$

Finally, the Server can send the encrypted LUT $[\![\mathbf{D}_k]\!]$ to the Client, who decrypts it with his/her private key obtaining

$$\mathbf{D}_k = -\mathbf{E}_k + \mathbb{G}\mathbf{m}_k. \quad (7)$$

As done in [10], the Server can randomize $b_{k,l}$ in the encrypted domain in order to prevent the Client from removing the watermark from $\mathbf{D}_k$.

3.2. Watermark Decoding

The decryption operation described in (4) can be modeled by adding to the encrypted signal the product of the decryption LUT $\mathbf{D}$ and a proper binary matrix $\mathbb{T}$ defined according to the sequence of indexes t_{ih}, i.e.,

$$\mathbf{y} = \mathbf{c} + \mathbb{T}\mathbf{D}_k = \mathbf{x} + \mathbb{T}\mathbf{W}_k \tag{8}$$

where $\mathbb{T}$ is a $M \times T$ binary matrix defined as

$$\mathbb{T}(i,j) = \begin{cases} 1 & t_{ih} = j,\ h = 0, \ldots, R-1 \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

As a result, the watermark associated to the kth user is given by

$$\mathbf{w} = \mathbb{T}\mathbf{W}_k = \mathbb{T}\mathbb{G}\mathbf{m}_k = \tilde{\mathbb{G}}\mathbf{m}_k \tag{10}$$

that is, the watermark is equal to the message $\mathbf{m}_k$ encoded by the linear block code defined by the $M \times L$ generator matrix $\tilde{\mathbb{G}} = \mathbb{T}\mathbb{G}$.

Since the scheme is asymmetric, the decoder does not know the messages $\mathbf{m}_k$, so it can not employ a correlation detector as in [12]. Instead, the detector obtains an estimated fingerprint $\hat{\mathbf{b}}_k$ and verifies whether it matches with a recorded Client, using the proof of identity provided by the underlying buyer-seller protocol. Let us assume that the watermark decoder receives a copy of the watermarked signal corrupted by an additive noise, i.e., the received signal is

$$\mathbf{y}' = \mathbf{y} + \mathbf{n} = \mathbf{x} + \tilde{\mathbb{G}}\mathbf{m}_k + \mathbf{n}. \tag{11}$$

When the original signal is available at the decoder, its interference can be removed and decoding can be performed on the signal $\mathbf{y}'' = \mathbf{y}' - \mathbf{x} = \tilde{\mathbb{G}}\mathbf{m}_k + \mathbf{n}$. Otherwise, blind decoding can be obtained by directly using the received signal $\mathbf{y}'$ and considering $\mathbf{x}$ as an additional noise term.

Several decoding strategies can be considered to recover the Client's fingerprint $\hat{\mathbf{b}}_k$. In this paper, we will consider the *Matched Filter (MF)* decoder

$$\hat{\mathbf{b}}_k = \text{sgn}\left\{\tilde{\mathbb{G}}^T\mathbf{y}''\right\} \tag{12}$$

and the *Pseudo-Inverse (PI)* decoder

$$\hat{\mathbf{b}}_k = \text{sgn}\left\{(\tilde{\mathbb{G}}^T\tilde{\mathbb{G}})^{-1}\tilde{\mathbb{G}}^T\mathbf{y}''\right\}. \tag{13}$$

MF and PI decoders are based on standard suboptimal receiver commonly adopted in digital communications. Namely, the PI decoder corresponds to zero-forcing equalization followed by hard decision.

4. EXPERIMENTAL RESULTS

For the experimental validation of the proposed technique, we have simulated a system performing client-side embedding on digital images. We have considered a dataset of 20 gray scale uncompressed 8 bit images, each having resolution 1024×1024. For each image, the signal $\mathbf{x}$ has been obtained by applying a 8×8 discrete cosine transform (DCT) to the image and taking 4 DCT coefficients for each 8×8 block, corresponding to the coefficients between the 7th and 10th positions according to the zig-zag ordering used by JPEG standard. This resulted in a vector $\mathbf{x}$ of 2^{16} components.

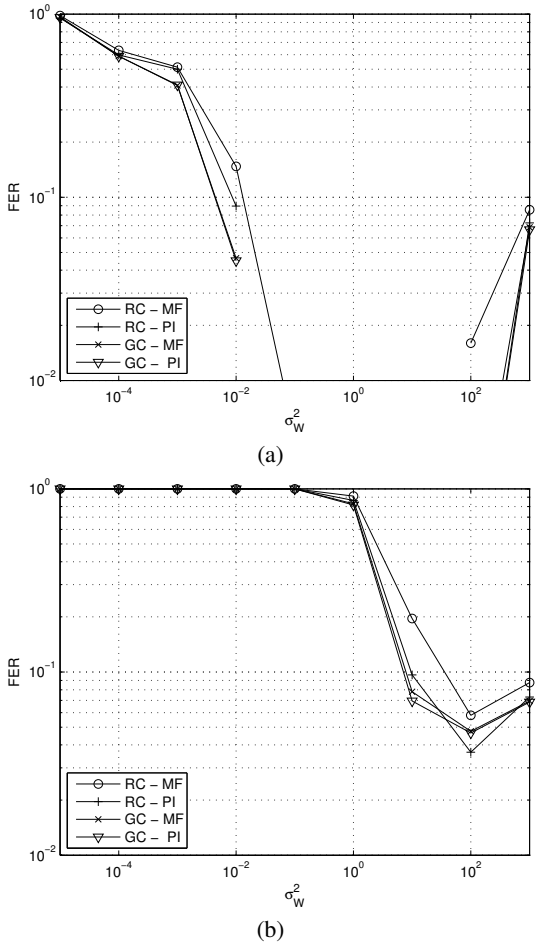

Fig. 1. FER performance of different encoding a decoding strategies: (a) nonblind decoding; (b) blind decoding.

Each image has been encrypted by using an encryption LUT $\mathbf{E}$ with power $\sigma_E^2 = 10^6$. After adding the elements of $\mathbf{E}$ to the selected DCT coefficients, the images have been reconstructed by using an inverse block DCT and pixel values have been mapped to 8 bit values by applying rounding and a modulo 256 operation. An analogous sequence of operations have been performed when decrypting the images with the decryption LUT $\mathbf{D}$. The use of the modulo operation guarantees that the encryption is perfectly reversible as long as $\mathbf{D} = -\mathbf{E}$. However, when $\mathbf{D} = -\mathbf{E} + \mathbf{W}$, some pixels may exceed the range $[0, 255]$ in the watermarked image and wrap around after the modulo operation. In order to prevent this problem, the histogram of each image has been compressed between 2 and 253. Moreover, possible wrong pixels in the reconstructed image are identified by comparing the value of each pixel with the value of a 5×5 median filtered version of the image, and pixels whose value differ by more than 192 from the median filtered image are replaced by the corresponding value in the median filtered image.

In all experiments, the LUT size was set to $T = 2^{16}$ and $R = 4$ LUT entries are added to encrypt each element. We simulated embedding and subsequent decoding of a 128 binary fingerprint. Two encoding strategies were considered, repetition coding (RC) and i.i.d. random Gaussian coding (GC), as described in Section 3. As to decoding, we considered MF and PI decoding. For each im-

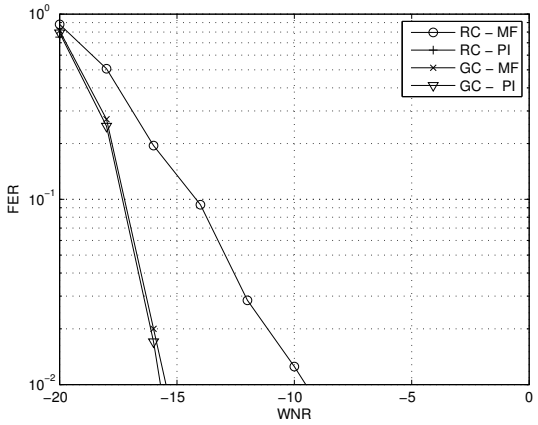

Fig. 2. FER performance of different encoding a decoding strategies in the presence of AWGN attack, considering nonblind decoding.

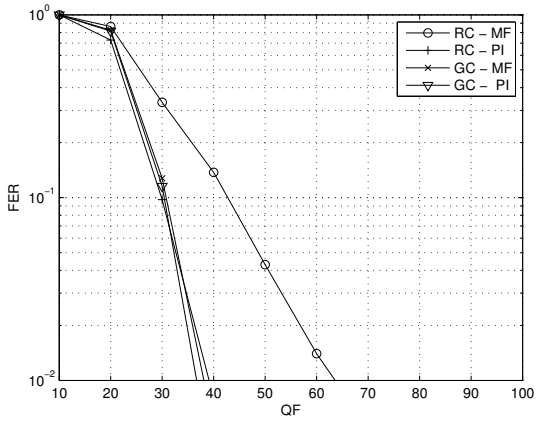

Fig. 3. FER performance of different encoding a decoding strategies in the presence of JPEG attack, considering nonblind decoding.

age, 100 independent tests were performed by randomly generating different encryption LUTs, different fingerprints, and different encoding matrices $\mathbb{G}$. This yielded $20 \times 100 = 2000$ transmitted fingerprints and $2000 \times 128 = 256000$ transmitted bits for each experiment.

The decoding performance has been evaluated by measuring the fingerprint error rate (FER), corresponding to the ratio of erroneously decoded fingerprints to the overall transmitted fingerprints. In order to evaluate the quality of the watermarked image with respect to the original image, we also measured the mean structural similarity (MSSIM) index [21].

A first set of experiments considered the decoding performance in the absence of attacks, by testing different watermarking powers corresponding to values $\sigma_W^2 \in [10^{-5}, 10^3]$. Fig. 1-(a) shows the decoding performance in the nonblind case. The results show that for $\sigma_W^2 > 10^{-2}$ all the strategies achieve a good fingerprint decoding performance, with GC being slightly better than RC and PI decoding being slightly better than MF decoding. Namely, for $\sigma_W^2 = 10^{-2}$ GC using PI decoding is able to correctly decode more than 95% of the fingerprints. Interestingly, some errors appears for $\sigma_W^2 \geq 10^2$ due to the fact that the large watermarking power causes several wrap around errors in the reconstructed image. Fig. 1-(b) shows the decoding performance in the blind case. As expected, the performance is significatively worse than in the nonblind case. Namely, all strategies, except RC using MF decoding, can correctly decode more than 90% of the fingerprints only when $\sigma_W^2 \geq 10$. As to the effect of the watermarking strength on the reconstructed images, for $\sigma_W^2 = 1$ the value of MSSIM index in the worst case is 0.9941 for RC and 0.9949 for GC, meaning that when using nonblind decoding the proposed method can achieve very good decoding performance without significantly affecting the quality of the watermarked image.

A second set of experiments were conducted in the presence of attacks, for $\sigma_W^2 = 1$. For brevity, only the results obtained by the nonblind decoder are presented. The watermarked images were either corrupted by AWGN or compressed using the JPEG standard. In the first case, we considered watermark-to-noise ratios (WNRs) in the range $[-20, 0]$, where we define $\mathrm{WNR} = 10 \log_{10} \frac{R\sigma_W^2}{\sigma_N^2}$, being σ_N^2 the variance of the additive noise. In the second case, we considered different JPEG quality factors (QF), from 10 to 100.

Fig. 2 shows the performance of nonblind decoding in the presence of AWGN attack. With the exception of RC using MF decoding, all the strategies achieve very similar performance and guarantee almost error-free decoding of the fingerprint for $\mathrm{WNR} > -15$, which demonstrates a great robustness in the presence of AWGN. For $\mathrm{WNR} = -16$, the average MSSIM index value after the AWGN attack is 0.5599 for both RC and GC, indicating that the image is so degraded as to be of no practical value.

Fig. 3 shows the performance of nonblind decoding in the presence of JPEG attack. Similarly to the AWGN case, all the strategies achieve very similar performance except RC using MF decoding. In general, the proposed scheme can withstand JPEG compression with a quality factor as low as 40 without showing significant decoding errors and can still correctly decode about 20% of the fingerprints for a quality factor equal to 20. For a quality factor equal to 20, the values of MSSIM index after the JPEG attack range from 0.7960 to 0.9268, with an average value of 0.8826, indicating that most of the images have to be largely degraded in order to impede the correct decoding of the fingerprint.

5. CONCLUSIONS

In this paper, we have proposed a novel client-side embedding technique enabling the distribution of multimedia content through an asymmetric fingerprint protocol. The main idea behind the proposed scheme is that existing asymmetric protocols, that do not require a dedicated trusted third party, can be used to securely exchange the personalized decryption keys needed by client-side embedding. In order to make this approach feasible, the Buyer's binary fingerprint is encoded in the personalized decryption key via linear block coding, which can be securely implemented at the Seller's side by using homomorphic encryption. Since the size of a decryption key is much lower than the size of a multimedia content, and a single key can be used for multiple contents, the proposed solution offers significant advantages with respect to a traditional server-side asymmetric protocol. Moreover, simulation results show that the embedded fingerprint can be reliably decoded from the watermarked content, even when using low watermarking power and in the presence of common attacks, like additive Gaussian noise and JPEG compression. The proposed scheme can offer a valid solution in multimedia content distribution, since it is able to protect both seller's and customer's rights, and, at the same time, it effectively solves scalability issues.

6. REFERENCES

[1] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*, Marcel Dekker, 2004.

[2] W.S. Lin, H.V. Zhao, and K.J.R. Liu, "Game-theoretic strategies and equilibriums in multimedia fingerprinting social networks," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 191 –205, Apr. 2011.

[3] T. Bianchi and A. Piva, "Secure watermarking for multimedia content protection: A review of its benefits and open issues," *Signal Processing Magazine, IEEE*, vol. 30, no. 2, pp. 87–96, 2013.

[4] B. Pfitzmann and M. Schunter, "Asymmetric fingerprinting," in *Adv. in Cryptology - EUROCRYPT'96*, 1996, LNCS 1070, pp. 84–95.

[5] N. Memon and P. Wong, "A buyer-seller watermarking protocol," *IEEE Transactions on Image Processing*, vol. 10, no. 4, pp. 643–649, Apr. 2001.

[6] M. Kuribayashi and H. Tanaka, "Fingerprinting protocol for images based on additive homomorphic property," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2129–2139, Dec. 2005.

[7] J. P. Prins, Z. Erkin, and R. L. Lagendijk, "Anonymous fingerprinting with robust QIM watermarking techniques," *EURASIP Journal on Information Security*, vol. 2007, Article ID 31340, 13 pages, 2007.

[8] Minoru Kuribayashi, "On the implementation of spread spectrum fingerprinting in asymmetric cryptographic protocol," *EURASIP J. Inf. Secur.*, vol. 2010, pp. 1:1–1:11, Jan. 2010.

[9] Mina Deng, Tiziano Bianchi, Alessandro Piva, and Bart Preneel, "An efficient buyer-seller watermarking protocol based on composite signal representation," in *Proceedings of the 11th ACM workshop on Multimedia and security*, Princeton, New Jersey, USA, 2009, pp. 9–18, ACM New York, NY, USA.

[10] A. Rial, Mina Deng, T. Bianchi, A. Piva, and B. Preneel, "A provably secure anonymous buyer-seller watermarking protocol," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 920 –931, Dec. 2010.

[11] R. J. Anderson and C. Manifavas, "Chameleon—a new kind of stream cipher," in *Proceedings of the 4th International Workshop on Fast Software Encryption — FSE'97*, London, UK, 1997, pp. 107–113, Springer-Verlag.

[12] M. Celik, A. Lemma, S. Katzenbeisser, and M. van der Veen, "Look-up table based secure client-side embedding for spread-spectrum watermarks," *IEEE Trans. on Information Forensics and Security*, vol. 3, no. 3, pp. 475–487, 2008.

[13] A. Piva, T. Bianchi, and A. De Rosa, "Secure client-side ST-DM watermark embedding," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 13 –26, Mar. 2010.

[14] Chih-Yang Lin, Panyaporn Prangjarote, Li-Wei Kang, Wei-Lun Huang, and Tzung-Her Chen, "Joint fingerprinting and decryption with noise-resistant for vector quantization images," *Signal Processing*, vol. 92, no. 9, pp. 2159 – 2171, 2012.

[15] Stefan Katzenbeisser, Aweke Lemma, Mehmet Utku Celik, Michiel van der Veen, and Martijn Maas, "A buyer–seller watermarking protocol based on secure embedding," *IEEE Trans. on Information Forensics and Security*, vol. 3, no. 4, pp. 783–786, Dec. 2008.

[16] Geong Poh and Keith Martin, "An efficient buyer-seller watermarking protocol based on chameleon encryption," in *Digital Watermarking*, Hyoung-Joong Kim, Stefan Katzenbeisser, and Anthony Ho, Eds., vol. 5450 of *Lecture Notes in Computer Science*, pp. 433–447. Springer Berlin / Heidelberg, 2009.

[17] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology - EUROCRYPT 1999*, J. Stern, Ed. 1999, number 1592 in Lecture Notes in Computer Science, pp. 223–238, Springer Verlag.

[18] M. Celik, A. Lemma, S. Katzenbeisser, and M. van der Veen, "Secure embedding of spread-spectrum watermarks using look-up tables," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP'07)*. 2007, IEEE Press.

[19] Jr. Marshall, T., "Coding of real-number sequences for error correction: A digital signal processing problem," *IEEE J. Sel. Areas Commun.*, vol. 2, no. 2, pp. 381–392, 1984.

[20] Zhengdao Wang and G.B. Giannakis, "Complex-field coding for OFDM over fading wireless channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 707–720, 2003.

[21] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.