Which directions for asymmetric watermarking?

Gaël Hachez
Université Catholique de Louvain
UCL Crypto Group,
Place du Levant 3
B-1348 Louvain-la-Neuve, Belgium.
E-mail: hachez@dice.ucl.ac.be

Jean-Jacques Quisquater
Université Catholique de Louvain
UCL Crypto Group,
Place du Levant 3
B-1348 Louvain-la-Neuve, Belgium.
E-mail: jjq@dice.ucl.ac.be

ABSTRACT

A perfect asymmetric watermark is what a lot of researchers are looking for. So far, no asymmetric scheme is perfect (at least from a security point of view). We here review some schemes proposed so far. We put an emphasis on classical cryptography to see how it has been and can be used, imitated in the watermarking world.

Keywords. asymmetric watermarking, copyright protection

1 Introduction

A lot of people talk about asymmetric watermarking. But the definition can have different meanings.

It is a hot topic because the industry expects something from the research that does not exist up to now. What are the characteristics of such a perfect watermark?

- robust against any attack,
- embedded with a secret key,
- can be verified with the knowledge of a public key or no key at all but without the knowledge of the (a part of the) secret key,
- can be verified off-line without contacting any authority (secure server),
- the full knowledge of the verification algorithm and the public key cannot allow to remove or alter the watermark.

Progress has been made towards such scheme but no one perfectly match the requirements set by the industry.

It is also worth noting that these requirements do not imply the use of a pair private / public key. Others schemes are possible.

In the following sections we will investigate the different methods that have been chosen and others that can be used.

2 Partly known key

This was one the first idea to obtain a kind of asymmetric watermark and was introduced by Hartung and Girod [12]. It is a simple modification of spread-spectrum watermarking. The recipient of the watermarked signal receive only a part of the watermark. The recipient can check the presence of this part of the watermark. Unfortunately, he is also able to remove this part of the watermark. That means that a public detector of the watermark cannot detect it anymore although the private part of the watermark remains.

3 Based on autocorrelation

By embedding some specific values, a correlation appears between the signal and a transformation of the signal. The embedded values are the private keys and the transformation matrix the public key.

3.1 Legendre sequence insertion

The first scheme based on this technique was introduced by van Schyndel, Tirkel and Svalbe in [14]. They embed a Legendre sequence in the signal. The Legendre sequence is invariant in a Fourier transform. Therefore it correlates with its conjugate Fourier transform. The private key is the Legendre symbol embedded and the public key is the length of the sequence.

3.2 Eigenvector insertion

The second scheme is a variation on the first scheme and was introduced by Eggers, Su and Girod in [7]. Here the watermark is an eigenvector of a transformation matrix. Once again as it is an eigenvector, it correlates with its transformation by the matrix. The private key is an eigenvector of the transformation matrix and the public key is the transformation matrix itself

Unfortunately, attacks that remove these watermarks have been found on these two schemes [6].

4 Inspired by cryptography

Here the idea is to look at what has been done in cryptography to solve such problem. As it works in cryptography why not reuse it?

In order to obtain public key systems, cryptographic protocols use one-way functions with trapdoors. The principle is quite simple: without the trapdoor (the secret key), the function is quasi impossible to invert. With the trapdoor, it is easy to invert the function.

As mentioned by Cox, Miller and Bloom in [1], the similarity is not perfect. The cryptography mapping between the cleartext and the ciphertext is one-to-one and a small change in the clear text induces a random change in the ciphertext. In watermarking, a small change in the signal should allow the detection of the same watermark.

The first tentative was made by Furon and Duhamel in [9, 10]. They did not really use a one-way trapdoor function. Instead, they use a signal processing one-way function: the power density spectrum (PDS). The result of this function does not allow a perfect reconstruction of the signal.

The signal O is randomly permuted $\tilde{O}$ to obtain a flat PDS. The watermark W is a filtered white noise signal. The watermarked permuted signal is $\tilde{O}'$. As W is independent of $\tilde{O}$, the PDS of $\tilde{O}'$ is the sum of the PDS of the two signals. This PDS is the public key. The random permutation is reverted and the resulting watermarked signal is O'.

To verify the watermark, the signal O' is randomly permuted, his PDS is computed and a hypothesis test is used to verify it is the same as the public key.

But once again, an attack able to remove the watermark was found against the system [6].

5 Using watermarking as subset of a cryptographic protocol

As the security of asymmetric cryptographic protocols has already been assessed, we can be a little bit more confident. However, it must be integrated inside the watermarking process which is a difficult step. The problem is not that the added component is secure (we here suppose it) but is in the interaction of all components of the system. Jointly with François Koeune in [11], we already showed that integrating new components in a protocol is likely to lead to new attacks on the system if the integration is not carefully crafted.

5.1 Based on zero knowledge

Craver in [2] and jointly with Katzenbeisser in [3, 4] use a zero knowledge protocol to prove the presence of a watermark in the signal. The secure cryptographic primitive used is a zero-knowledge proof based on the isomorphism of graphs.

The private key is a permutation τ on a finite group of n elements. The public key is a graph G and its permutation $\tau(G)$. The signal to be watermarked O is treated as an array of n samples. A classic symmetric watermark W is inserted in the signal O. Then O' is the watermarked signal. The signature of O' is the value $< \tau(O'), \tau(W) >$.

The protocol needs multiple rounds to gain confidence in the fact that W is embedded with O'. The probability to cheat the protocol is $1/2^m$ where m is the number of rounds. At each round, Alice (the prover) finds two permutations σ_i and ρ_i so that $\sigma_i \circ \rho_i = \tau$ and computes $G_i = \rho_i(G)$ and $O'_i = \rho_i(O')$. Alice sends two commitments to Bob (the verifier): $< C_1(\rho_i), C_2(\sigma_i), \mathcal{H}(O'_i), \mathcal{H}(G_i) >$ where $\mathcal{H}$ is a hash function. Bob flips a coin and asks to Alice to open the commitment C_1 or C_2. If Alice opens C_1, Bob can verify that $\mathcal{H}(\rho_i^{-1}(\tau(O'))) = \mathcal{H}(O'_i)$ and that $\mathcal{H}(\rho_i^{-1}(\tau(G))) = \mathcal{H}(G_i)$. He also verifies that $W_i = \rho_i^{-1}(\tau(W))$ is embedded within O'_i. In the other case, if Alice opens C_2, Bob checks that $\mathcal{H}(\sigma_i(\tau(O'))) = \mathcal{H}(O'_i)$ and that $\mathcal{H}(\sigma_i(\tau(G))) = \mathcal{H}(G_i)$.

In order to avoid that the verifier gets any information in the verification protocol, the signal must be preconditioned as atypical values could give information on the secret permutation. On the other side, if the preconditioning step is too aggressive, the watermark could be destroyed. The authors find a suitable function: the Hilbert space-filling curve.

As we mentioned at the beginning of this section, integrating two components can lead to new attacks and it is the case here. The authors describe a chosen plaintext attack against their protocol which usually does not pose a threat in a classical zero-knowledge proof.

Eventually, this scheme does not really solve the industrial problem of the asymmetric watermark because it requires a lot of data to prove the existence of the watermark and it needs a lot of computations and exchanges between Bob and Alice.

5.2 Ambiguity attacks

In [4], Craver and Katzenbeisser introduce a subtle mechanism where they combine classical cryptography (discrete log and zero-knowledge), classical watermarking and classical attacks against watermarking.

The basis here is the ambiguity attack (a.k.a. invertibility attack). When you have a signal O, it is possible to extract a watermark W so that $O = \tilde{O} + W$ and then claim that the original signal is $\tilde{O}$, the watermark is W and the watermarked signal O.

Alice computes a watermark $W = a^h \bmod p$ from her secret h. Computing h from W knowing a is know as the discrete log problem and is believed to be intractible for large p. She embeds the watermark in her signal $O' = O + W$. Using the ambiguity attack, she find k different watermarks W_i embedded in O'. She publishes $k+1$ randomized watermarks (the fake ones and the real one). If Bob want to remove the correct watermark, he needs to remove all watermarks and therefore seriously damage the signal O.

The verification protocol is as follows: Alice chooses blinding exponents h_i for all watermarks still detectible in O'. She sends to Bob the blinded watermarks $B_i = W_i a^{h_i}$. Bob flips a coin. In one case, Alice has to reveal the blinding exponent and Bob verifies that the B_i are correct. In the other case, Alice has to give the discrete log of one B_i.

The authors pointed that a chosen plain text attack was possible against this version of the protocol. They had to modify a little bit the protocol. This shows again that the interaction between different kind of security components must be studied carefully.

This protocol is not suitable either for an off-line verification as it requires a lot of exchanged data between Alice and Bob.

5.3 Extension of the zero knowledge schemes

The two previous protocols are based on zero knowledge interactive proofs. In classical cryptography, you can use several techniques to prove that you know something. One method is the asymmetric cryptography where the secret is protected by the one way functions with trapdoors as mentioned earlier. Another method is the zero knowledge interactive proof where you prove you own something without giving any information about your secret. This is why such technique has been used by Craver and Katzenbeisser as there is no real one way watermarking function with trapdoor.

A zero knowledge interactive proof can be transformed in a signature scheme [8]. When such transformation is applied, we lose the zero knowledge property. A replay is now possible but this is not a problem in our case.

We can now verify the signature off-line but if somebody remove the attached signature, we cannot verify the watermark even if there is a watermark embedded.

A not completely satisfactory solution: If the device does not find any signature associated with the signal, it contacts a server and tries to find the associated signature (i.e. by using a robust hash of the image).

With such protocol, we usually avoid an on-line verification and still conserve most of the properties of the original schemes. This is a small step in the right direction but we do not reach the goal.

5.4 Key distribution

A well studied problem is the distribution of keys for video broadcasting. The goal is to find the customer who released a pirated copy of the video. The basic idea is to distribute a different subset of keys to each customer. By analyzing the pirated copy it is possible to retrieve which was the subset of keys and hence the pirate.

We can apply this method to watermarking. A watermark is embedded within the image. The watermark is divided in several parts in the same way as the first exposed scheme in this paper. Each customer receives a subset of key parts.

Obviously he will be able to remove the known parts of the watermark. However, the watermark will be detected by each customer that does not have the same subset of key parts and it is possible to find the culprit. Lastly, a collusion attack is always possible, the number of people needed to remove the complete watermark depends on the exact key distribution scheme used.

6 Using cryptography as subset of watermarking

The payload of the watermark can be any data. So why not put some cryptographic data inside the watermark. By using such method, we can tightly link the payload of the watermark to the signal to be watermarked. This could solve the problem of forging a new watermarking but this usually cannot address the problem of the removal of watermark.

6.1 Identity scheme

A classical cryptographic identity scheme is used to prove your identity to another person. The scheme involves three entities: Alice (the prover), Bob (the verifier) and a Trusted Third Party (TTP) that Alice and Bob trust. The protocol goes as follows: the TTP signs with his private key some data O related to Alice (i.e. her picture) certifying that O is really linked to Alice. With the public key of the TTP, Bob can verify the signature and thus link O to Alice.

Let us use that in the context of watermarking. We need to link the signal to Alice. So Alice sends some intrinsic data of the signal (i.e. a robust image hash $\mathcal{H}(O)$)) and her identity $I_a$to the TTP. The TTP returns the signature $S = \mathcal{S}_k(\mathcal{H}(O), I_a)$ to Alice. Alice embeds the signature as watermark inside O and publish the watermarked signal O'. Bob verifies with the public key of the TTP that the watermark embedded within O' is the signature made by Alice.

With such a scheme, it is very difficult to make fake watermarks because the signature can only be generated by the TTP and because the content of the signature is tightly linked to the signal.

What is really lacking in this scheme, it is that the watermark is not protected and therefore can be easily removed.

7 Function watermark

The method we will expose here is not related to classical watermarking (in the image or audio domain). However this kind of watermark has some nice properties. How it can be used in the signal domain is unknown.

This method is very old and has already been used centuries ago. Computing a logarithm without any computer is difficult task and can take hours, even days to obtain good precision. Some people built tables of logarithms and sold these tables. To avoid that other people steal their tables and sell as their own, they introduced special rounding methods instead of the classical one. These methods could take into account the position where the rounding occurs and even the original number.

To prove ownership of their table, they could give the particular instance used to compute a randomly chosen logarithm. The watermark on this logarithm could be removed but as the author had not given the general rounding formula, the rest of the watermarks remains intact.

In the same spirit, Naccache, Shamir and Stern created in [5] a watermark for function.

8 Conclusion

There is no real conclusion. Can we create a perfect asymmetric watermarking scheme? Nobody knows. Some people even believe that it is impossible but as we say in French "*Impossible n'est pas français*". So we will see in the future.

Still, we can point out that looking at classical cryptography is a good source of inspiration as the best schemes pick some idea or protocols there.

References

[1] Ingemar J. Cox, Matthew L. Miller, and Jeffrey A. Bloom. *Digital Watermarking*, pages 299–302. Morgan Kaufmann, 2002.

[2] Scott Craver. Zero knowledge watermarking detection. In Pfitzmann [13], pages 101–116.

[3] Scott Craver and Stefan Katzenbeisser. Copyright Protection Protocols Based on Asymmetric Watermarking. In M. Steinebach R. Steinmetz, J. Dittmann, editor, *Proceedings of the Fifth Conference on Communication and Multimedia Security (CMS'01)*, pages 159–170. Kluwer Academic Publishers, 2001.

[4] Scott Craver and Stefan Katzenbeisser. Security Analysis of Public-Key Watermarking Schemes. In *Proceedings of the SPIE, Mathematics of Data/Image Coding, Compression, and Encryption IV*, volume 4475, pages 172–182, July 2001.

[5] Adi Shamir David Naccache and Julien P. Stern. How to Copyright a Function? In H. Imai and Y. Zheng, editors, *Proceedings of the Second International Workshop on Practice and Theory in Public Key Cryptography (PKC'99)*, volume 1560 of *Lectures Notes in Computer Science (LNCS)*, pages 188–196. Springer-Verlag, 1999.

[6] J. J. Eggers, J. K. Su, and B. Girod. Asymmetric Watermarking Schemes. *Sicherheit in Mediendaten*, September 2000.

[7] J. J. Eggers, J. K. Su, and B. Girod. Public Key Watermarking by Eigenvectors of Linear Transforms. In *Proceedings of the European Signal Processing Conference (EUSIPCO 2000)*, 2000.

[8] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86*, volume 263 of *Lectures Notes in Computer Science (LNCS)*, pages 186–196. Springer-Verlag, 1987.

[9] Teddy Furon and Pierre Duhamel. An Asymmetric Public Detection Watermarking Technique. In Pfitzmann [13], pages 88–100.

[10] Teddy Furon and Pierre Duhamel. Robustness of an Asymmetric Watermarking Technique. In *IEEE International Conference on Image Processing*, volume 3, pages 21–24, 2000.

[11] Gaël Hachez, François Koeune, and Jean-Jacques Quisquater. Biometrics, Access Control, Smart Cards: A not so Simple Combination. In Josep Domingo-Ferrer, David Chan, and Anthony Watson, editors, *Smart Card Research and Advanced Applications*, IFIP, Bristol, United Kongdom, 2000. Kluwer Academic Press.

[12] F. Hartung and B. Girod. Fast Public-Key Watermarking of Compressed Video. In *Proceedings of the IEEE International Conference on Speech and Signal Processing*, 1997.

[13] A. Pfitzmann, editor. *Information Hiding '99*, volume 1768 of *Lectures Notes in Computer Science (LNCS)*, Dresden, Germany, 2000. Springer-Verlag.

[14] R. G. van Schyndel, A. Z. Tirkel, and I. D. Svalbe. Key Independent Watermark detection. In *IEEE International Conference on Multimedia Computing and Systems*, volume 1, 1999.