Chapter 1

USING DIGITAL WATERMARKING FOR SECURING NEXT GENERATION MEDIA BROADCASTS

Abstract The Internet presents a problem for the protection of intellectual property. Those who create content must be adequately compensated for the use of their works. Rights agencies who monitor the use of these works exist in many jurisdictions. In the traditional broadcast environment this monitoring is a difficult task. With Internet Protocol Television (IPTV) and Next Generation Networks (NGN) this situation is further complicated.

In this work we focus on Digitally Watermarking next generation media broadcasts. We present a framework which provides the ability to monitor media broadcasts that also utilises a Public Key Infrastructure (PKI) and Digital Certificates. Furthermore, the concept of an independent monitoring agency, that would operate the framework and act as an arbiter, is introduced. We finally evaluate appropriate short signature schemes, suitable Watermarking algorithms and Watermark robustness.

Keywords: Next Generation Networks, broadcast monitoring, public key watermarking, IPTV, PKI, short signature

1. Introduction

Radio and television are audio and video broadcasting services typically broadcasted over the air, cable networks or satellite networks. Since the advent of the Internet, the distribution of media content has always been a principal goal, however for many years this was not realised due to the prohibitive cost and limited capabilities of personal computers.

With the transition to digital media streams received over the Internet, new challenges loom. Today, the practices of recording, distribution and copying multimedia content is easy and straightforward [Peitz and Waelbroeck, 2003]. Due to these facts, it is more and more difficult to enforce copyright and to safeguard intellectual property for broadcast media.

Digital Watermarking [Coxy et al., 1996], which may be considered as a form of steganography [Lu, 2005], attempts to address this problem by embedding information within the digital signal. The embedded Watermark is invisible to the user, should not affect the perceived aesthetic quality of the final signal, nor should the Watermark reveal any clues about the technique used to embed it. However, it is debatable whether traditional Watermarking systems, which are based on disclosure of the key needed to embed and to detect the watermark are generally suitable for proving ownership or authentication. Therefore, we established a framework based on asymmetric public-key cryptography which is used for exhaustive authentication with the help of *Blind Watermarking* techniques.

In many jurisdictions broadcasters have regulatory obligations which attempt to protect the intellectual property [Lu, 2005] and copyrights of authors, songwriters, performers, actors, publishers, etc. Furthermore, in some jurisdictions there exists bodies charged with the defense of the rights of intellectual property and copyright holders and the calculation, charging and collection of performance royalties on the use of these protected works. Currently, there are several cases in which broadcasters cannot confidentially confirm that their royalties liabilities are correctly calculated. This is because they currently do not employ a viable automated system to measure what protected works are broadcasted, how often and when. Therefore a gap has opened up in the actual amount charged by the rights bodies and the correct payable royalties liability of the broadcaster.

This paper describes a specific Watermarking concept that may be used for identifying next generation media broadcast streams based on PKI authentication. We introduce a formal PKI framework in section 3 allocating authentication methods and then focus on procedures and measures for Watermarking media streams in legacy networks as well as NGNs using PKI. We prove our proposal through an exemplified scenario on video stream Watermarking.

2. Framework Overview

A general overview of the framework with its three parties can be seen in Figure 1.1. It makes use of two additional frameworks, the PKI and the Watermarking Framework.

The PKI Framework, described in chapter 3, is used for establishing a trust network between all of the involved entities. The *broadcaster* (BC) can initialise the monitoring process for metering his use of protected works and hence the royalties payable rights entity can also launch the monitoring process for billing purposes. In practice, the PKI procedures (1) should be established as the first step in the deployment of the framework. The PKI is necessary for establishing a trusted relationship with the purpose of distributing authenticated private and public keys utilising digital certificates. To start the process of monitoring, a ”Request for Monitoring” (2) is sent to the *monitoring agency*

(MA).
Afterwards, the broadcaster selects a piece of content which he wants to stream (3) and computes the corresponding hash table (see 4.0.0). This hash table is carried over a secure and authenticated channel to the MA as well as to the *rights entity* (EX). Afterwards, the broadcaster initiates the process defined by the Watermarking Framework.

The Watermarking Framework specifies procedures for Watermark embedding, retrieval and verification in media streams (4). The rights entity is the entity which charges broadcasters for distributing media content over a chosen distribution network. It also attempts to track and process broadcast media with the help of information obtained by the monitoring agency. The broadcaster will sign the stream which is about to be broadcasted with his private key. Subsequently, the corresponding signature will be embedded into the media stream with known Watermarking techniques. Later in the process, the monitoring agency will extract the Watermark and verify the signature. So, the agency can be sure that only the original broadcaster broadcasted the media content, due to the fact that additional security metadata, such as timestamps and identifiers, are used. Additionally, EX can also verify the signature in order to prevent abuse by the MA (5).

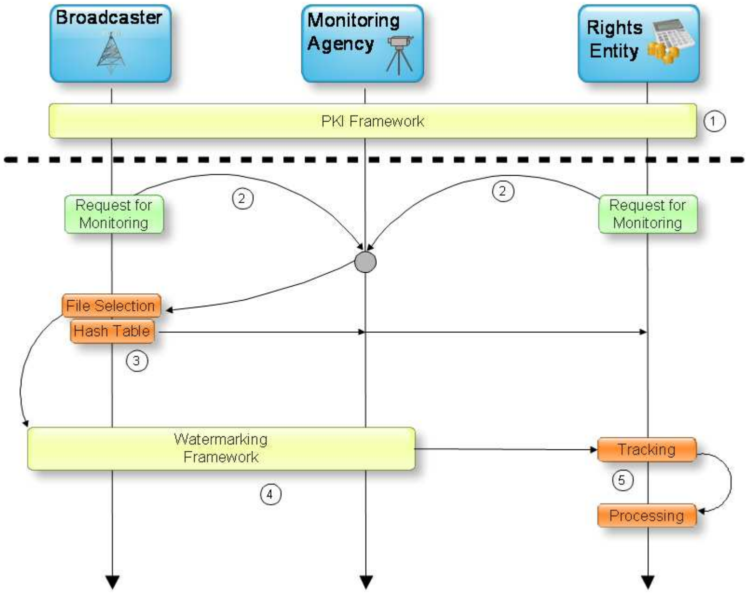

Figure 1.1. General Framework Overview

The objective of the whole framework is to let the broadcaster mark the file stream uniquely but also provides the monitoring agency with the possibility to identify the broadcast stream and therefore the corresponding broadcaster. Within this framework, non-repudiation is also provided. This means that the broadcaster cannot deny having broadcasted a Watermarked media stream.

3. PKI Framework

The PKI framework makes use of a root Certificate Authority (CA) in which each participating entity must trust. The monitoring agency, rights entity and the broadcaster submit their created public keys (*PK*) or create the keys directly at the CA for receiving the corresponding Certificate and the Certificates of all other participants.

Overview

Trust forms the basis of all communication, be it physical or electronic. In the case of electronic communication, building trust is quite difficult as the identity of the other entity remains concealed. While a PKI normally provides confidentiality, non-repudiation, authentication and integrity, our framework mainly focuses on authentication and non-repudiation.
A detailed description of the three stages in Figure 1.2 will be given in the following section.

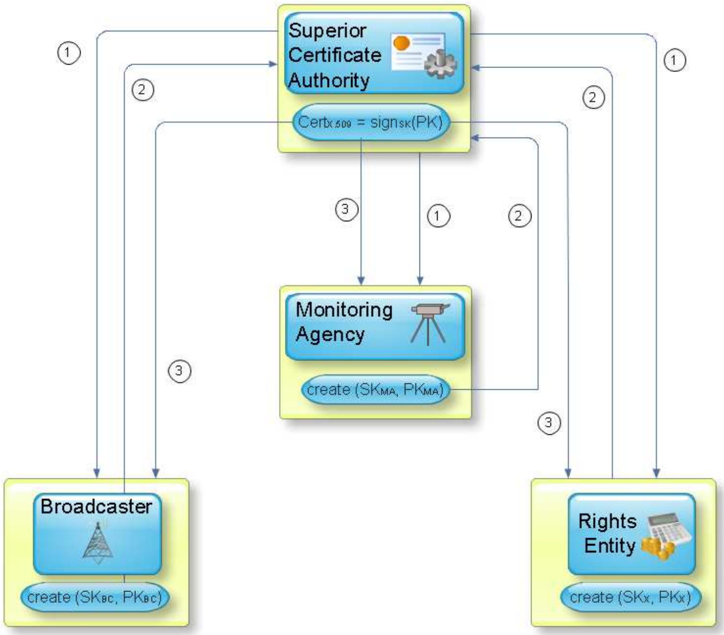
Superior Certificate Authority
Cert$_{X.509}$ = sign$_{SK}$(PK)
1
2
3
1
2
1
2
3
3
Monitoring Agency
create (SK$_{MA}$, PK$_{MA}$)
Broadcaster
create (SK$_{BC}$, PK$_{BC}$)
Rights Entity
create (SK$_X$, PK$_X$)

Figure 1.2. PKI Framework Overview

1 This step demonstrates a particular need of a PKI. The public key (PK_{CA}) of the CA has to be pre-distributed in an authenticated manner to any involved entity, otherwise no secure communication with the CA is possible.

2 As soon as the entities have received the authenticated PK_{CA} over a secure, authenticated channel, they create their own secret (SK_X, SK_{MA} and SK_{BC}) and public key (PK_X, PK_{MA} and PK_{BC}) and subsequently send a PKCS#10 Certificate request to the Certificate Authority. With Digital Signatures in Certificate requests, the CA can be sure that the sender has a private key related to the public key. Therefore, the sender has a proof of possession [Choudhury, 2002] but the receiver needs to assure that the entity with which he is communicating is not spoofed.

3 If the CA receives the Certification request, it behaves like a Registration Authority (RA) and tries to validate the information stored in the PKCS#10 file. If it is valid, a X.509 Certificate is issued by signing the corresponding *PK* of the entity. Afterwards, all Certificates are distributed to all entities for authentication reasons. So, the broadcaster owns now a Certificate concerning the *PK* of EX which was issued by the corresponding CA. The Certificate will be used during the Watermarking processes in order to authenticate the sender.

As previously mentioned, the main purpose of these protocol steps is providing full authentication. Now, the broadcasting entity could sign a message and send it to the monitoring agency or indeed to the rights entity and both entities could be assured, that the message was sent by the broadcaster.

4. Watermarking Framework

The Watermarking Framework, illustrated in Figure 1.3, specifies the communication protocol between the broadcaster and the monitoring agency in which the rights entity is not involved. Furthermore, the Watermarking Framework provides a detailed insight into procedures for creating, detecting, extracting and verifying the Watermark.

Overview

The framework is initialised at the moment the broadcaster had chosen a content file and transferred the corresponding hash table to the MA (see Algorithm 4.0.0). Afterwards, no further information needs to be sent to the MA due to the use of message recovering signatures. So the MA can be sure about who broadcast the stream and what stream has been broadcasted. This information is sufficient for metering and charging purposes.

The chief characteristic of a Watermarking scheme for copyright protection, or DRM, is that the Watermark cannot be separated from the medium without

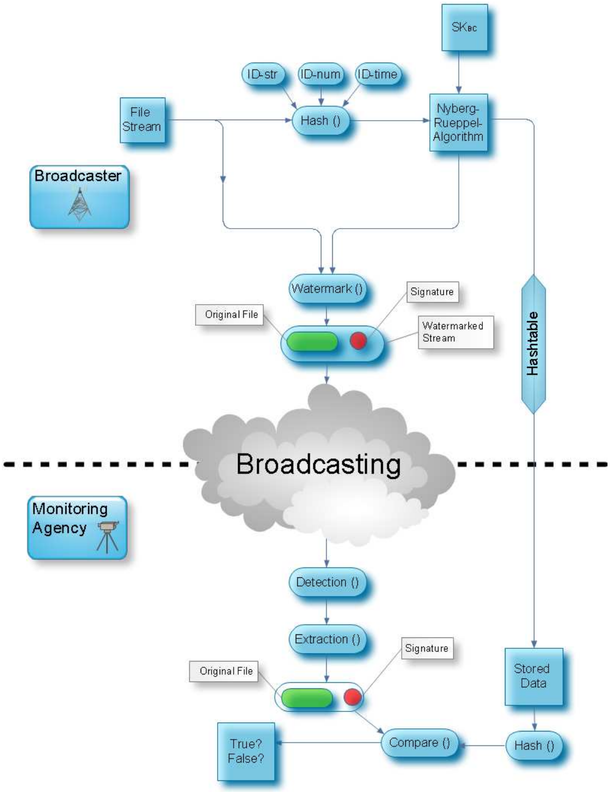

Figure 1.3. Watermarking Framework Overview

knowledge of a secret value. We, in our specific case, target on another characteristic: sender authentication. It should be possible to identify the broadcasting station unambiguously and show exactly who broadcasted what stream and when.

Therefore, our Watermark information contains a digital signature issued by the broadcaster. Each entity that receives the broadcast stream and owns the corresponding broadcaster certificate, can clearly verify the distributed stream with the help of the corresponding *PK*.

Below, we discuss suitable signature schemes and Watermarking algorithms. We introduce adequate procedures for embedding and retrieving the Watermark with the help of a beacon in addition to verifying the signature.

Signature Schemes

A principal requirement to all Watermarking systems is the need for a small Watermark. The larger the Watermark, the larger are the chances for adversely affecting the quality of the streamed media. In our case, the Watermark depends on the corresponding signature which has to authenticate the media stream. Interference and transaction defects could cause problems in extracting the Watermark. Therefore, the signature scheme output has to be as small [Naccache and Stern, 2001] as is possible to be able to embed the Watermark as often as possible and to be repeated multiple times throughout the stream.

The Nyberg-Rueppel ([Nyberg and Rueppel, 1993], hereafter NR) signature scheme focuses on the size of the input and output and is a DSA-like signature with message recovery.

The Nyberg-Rueppel Signature Scheme

NR is perfectly suited to messages shorter than ten bytes but leaves the question of dealing with short messages, of say fifteen bytes, unanswered. In our specific case, the hash to be signed is exactly 10 bytes long and brings only a marginal risk of collision (see section 4.0.0 for further details).

Message recovery [Ateniese and de Medeiros, 1999], another characteristic of NR signatures, provides means so that the original message can be extracted out of the signature. In our given case, this characteristic aligns with our goals. The hash value of the message *m* does not need to be created by the monitoring agency, due to the fact that it can be extracted from the signature due to the aforementioned message recovery characteristic. However, it is necessary to transfer a hash table (see section 4.0.0) once from the BC to the MA. This could happen in periodical time-frames.

The complete NR algorithm is shown in Algorithm 1 using the standard discrete logarithm (DL) problem.

Short Hash Methods

Hash functions are often used in digital signature algorithms. The message *m* that is about to be hashed, in our case, consists of an identifier string *ID-str* concatenated with an ID number *ID-num* and an unique times-tamp *ID-time*:

$$m = \mathit{ID\text{-}str} \mid\mid \mathit{ID\text{-}num} \mid\mid \mathit{ID\text{-}time} \tag{1.1}$$

The ID-str could be represented through the name of the media content, for instance. The ID-num could be an identification number. The ID-time is a unique time-stamp which prevents replay-attacks. This means, that an adversary may not record the stream and broadcast it later again on an authorised channel which is also monitored.

Hash Table for Verification Purposes

A hash table *ht* in our specific case is a data structure that associates the hash value with ID-str, ID-num and ID-time. The hash table contains several important attributes and is essential for the verification process by the MA (see

Algorithm 1 Nyberg-Rueppel signature generation and verification

Summary: the broadcaster signs a message $m \in \mathcal{M}$. The monitoring agency can verify the broadcaster's signature and recover the message m from the signature.

1 ***Signature Generation***. The broadcaster has to do the following:

- **(a)** Compute $\tilde{m} = R(m)$.
- **(b)** Select a random secret integer k, $1 \leq k \leq q-1$ and compute $r = \alpha^{-k} \bmod p$.
- **(c)** Compute $e = \tilde{m}r \bmod p$
- **(d)** Compute $s = ae + k \bmod q$.
- **(e)** The broadcaster's signature for the specific m is the pair (e,s).

2 ***Verification***. To verify the broadcaster's signature (e,s) on m, the monitoring agency should do the following:

- **(a)** Obtain the broadcaster's authentic public key (p,q,α,y) and verify it with the corresponding certificate delivered by the CA earlier (see Figure 1.2).
- **(b)** Verify that $0 < e < p$; if not, reject the signature.
- **(c)** Verify that $0 \leq s < q$; if not, reject the signature.
- **(d)** Compute $v = \alpha^s y^{-e} \bmod p$ and $\tilde{m} = ve \bmod p$
- **(e)** Verify that $\tilde{m} \in \mathcal{M}_{\mathcal{R}}$; if $\tilde{m} \notin \mathcal{M}_{\mathcal{R}}$ then reject the signature.
- **(f)** Recover $m = R^{-1}(\tilde{m})$.

Figure 1.4).

Transferring the hash table to the MA, can be compared to the cryptographical *commitment scheme*, visualized in Algorithm 2, except that the hash table has no hidden value.

```
Algorithm 2 Secure and Authentic Hash Table Distribution
Summary: during the commitment phase, the hash table is transferred to MA
and EX. During the opening phase, BC proves to MA and EX that he is broad-
casting one of the items in the hash table.

1  commitment phase:
     1. BC ⟶ MA: $\mathrm{enc}_{PK_{MA}}(\mathrm{sign}_{SK_{BC}}(ht))$
     2. MA ⟶ EX: $\mathrm{enc}_{PK_{EX}}(\mathrm{sign}_{SK_{BC}}(ht))$
2  opening phase:
     1. BC ⟶ MA: watermark($\mathrm{sign}_{SK_{BC}}(hs)$)
     2. MA: extract signature from stream with the help of beacon
     3. MA ⟶ EX: $\mathrm{enc}_{PK_{EX}}(\mathrm{sign}_{SK_{BC}}(hs))$
```

The prover, respectively the BC, sends the ”commitment” in form of the hash table *ht* to the verifier (the MA). MA will forward the signed hash table to the rights entity but encrypts it with the corresponding PK_{EX} in order to guarantee secrecy which is needed to prevent other parties from viewing the hash table. This can be seen as the *commitment phase* and takes place directly after having chosen the file to be streamed. The encryption is necessary due to the possibility that the hash table of a potential business rival might be seen by another party. Later, after broadcasting the media content, the verifier can scrutinise, with the help of the message recovery characteristic of the signature, whether the BC broadcast the content correctly or not (*opening phase*). As can be seen in Figure 1.4 that the hash value is only made of 10 bytes/ 20 hex characters. For verification, the MA needs to extract the hash value out of the signature with the help of the message recovery characteristic. Afterwards, he will look up the hash in the transferred hash table and check whether the corresponding ID-fields are valid. The same procedures can be done by the EX in order to be

Hash	ID-str	ID-num	ID-time
`e80c78299acc041ffd23`	Title A	42133	34523312
`9a2002a978b5c7538952`	Title B	87565	56245323
`65ae7da24e501c95a0ae`	Title C	52332	6345231

Figure 1.4. Structure of the Hash table

sure that the MA is not cheating.
For instance, if a video stream was recorded and replayed at a later point in time, the MA will recognise that due to this fact it will not match the ID-time field in the hash table. A video stream can only be validly broadcast once. If the BC tries to cheat by changing the ID-str field for a piece of media content with a lower or no payable royalty, the MA will detect that.

Case Study: Video Broadcaster

The *Internet Movie Database (IMDB)* published interfaces for several systems to access the IMDB locally. For our case study, we downloaded the complete IMDB title textfile which contains currently 1.206.730 different movie titles. We used the movie title as a ID-str and created a unique number used as the ID-num. The time-stamp ID-time was the current date parceled as a unixtimestamp. An example of an assignment between unixtimestamp and normal time can be seen in equation 1.2.

$$05/07/1982@00:00 \Longrightarrow 389592000 \tag{1.2}$$

For instance, in our simulation, *m* looked like this:

$$\text{m} = \underbrace{\text{Title A}}_{\textbf{ID-str}} \quad || \quad \underbrace{23754}_{\textbf{ID-num}} \quad || \quad \underbrace{534056}_{\textbf{ID-time}} \tag{1.3}$$

We created 1,206,730 different messages *m* and subsequently hashed them with MD5 and SHA-1. Afterwards, we extracted the first 10 bytes which satisfy the first 20 characters of the output HEX value. No collisions were detected for both hash functions, MD5 and SHA-1, even with only using the first 10 bytes of the hash-sum.

$$hs = [0...9]hash(m) \tag{1.4}$$

Suitable Watermarking Algorithms

A substantial body of research in Watermarking algorithms can be found in literature [Seitz, 2005]. However, in our specific case, the Watermark should have special control characteristics which are required to guarantee the ability to verify the embedded signature by the monitoring agency.

- **Robustness**
  Robust Watermarks are designed to resist against heterogeneous manipulations and therefore not substantial for our framework [Chen and Wornell, 1999]. Our framework focuses on authentication, not on robustness against

manipulation. Only robustness against accidental manipulation or signal interference would be useful in our case.

- **Invisibility**
  The Watermark embedded into a video stream should be visually imperceptible.

- **Inaudibility**
  The Watermark of an audio stream or the audio track in a video stream should be unaudible.

- **Complexity**
  Watermarking and Watermark retrieval should, in principle, have low complexity. Due to the fact that our case focuses on streaming applications, the functions for embedding and retrieving of the Watermark should be as simple as possible so that on the fly, or faster than realtime, Watermarking is possible.

- **Compressed domain processing**
  We assume that the broadcaster will store the media files in a compressed format. Referring to the above complexity requirement, embedding the watermark into the compressed video stream should be possible, specific decode and recode steps for watermarking are undesirable as not to affect the overall performance of the system.

In our case, there is no need to keep the Watermark private. Each participant in the framework may extract the Watermark via the known *beacon* needed to locate the Watermark. Afterwards, the signature can be extracted from the Watermark and verified with the help of the corresponding public key.

Proposed Watermarking Algorithm

Basically, a Watermarking system for our purposes can be described by a tuple $\langle \mathcal{O}, \mathcal{S}, \mathcal{W}, \mathcal{H}, \mathcal{P}, \mathcal{G}, C_S, E_H, D_H, V_P \rangle$ where $\mathcal{O}$ is the set of all original data, a video stream for instance. The set $\mathcal{S}$ contains all secret keys needed for creating an unforgeable signature. $\mathcal{W}$ represents the set of all Watermarks (signatures, in our case) and $\mathcal{H}$ the set of all beacons. Beacons in our scenario are markers that signify the presence and start of a Watermark bit sequence in the signal. The beacon substitutes the key in normal Watermarking systems. $\mathcal{P}$ describes the set of public keys which are needed to verify the signature and $\mathcal{G}$ represents the set of Certificates issued by the CA.

Four functions are described as followed:

$$C_S : \mathcal{O} \times \mathcal{S} \longrightarrow \mathcal{O} \tag{1.5}$$

$$E_H : \mathcal{O} \times \mathcal{S} \times \mathcal{W} \times \mathcal{H} \longrightarrow \mathcal{O} \tag{1.6}$$

$$D_H : \mathcal{O} \times \mathcal{H} \longrightarrow \mathcal{W} \tag{1.7}$$

$$V_P : \mathcal{W} \times \mathcal{P} \times \mathcal{G} \longrightarrow \{1,0\} \tag{1.8}$$

C_S focuses on creating the corresponding Watermark through a signature. E_H describes the function for embedding the Watermark and D_H respectively the function for extracting it. Furthermore, V_P stands for the verification function needed to check if the Watermark is valid.
The Watermark w is created with

$$w = sign_{SK_{BC}}(hs) \tag{1.9}$$

and outputs a short bit-string which contains the signature of the reduced hash-sum. See equation 1.4 for further details about the reduced hash-sum hs.

Embedding the Watermark

In this subsection we focus on the embedding process of the signature/ Watermark. Hartung and Girod proposed in 1998 [Hartung and Girod, 1998] a method which focuses on Watermarking MPEG-2 video data. We adopt the proposed methods for our purposes of embedding the signature into a given video broadcast stream. For further information, the interested reader is referred to [Hartung and Girod, 1998].
The basic concept of Hartung and Girod [Hartung and Girod, 1998] was to present a Watermarking scheme for MPEG-2 encoded as well as uncompressed video based on spread-spectrum methods [Cox et al., 1997].
Let

$$a_j, \quad a_j \in \{-1,1\}, \quad j \in \mathbf{N} \tag{1.10}$$

be the Watermark bit sequence to be hidden in a linearised video stream. In our case, this bit sequence contains the signature which was created by signing the reduced hash with a specific short signature method based on the NR algorithm (see section 4.0.0). This discrete signal is up-sampled by a factor cr called the chip-rate, to obtain a sequence

$$b_i = a_j, \quad j \cdot cr \leq i < (j+1) \cdot cr, \quad i \in \mathbf{N} \tag{1.11}$$

so as to provide redundancy. The new bit sequence b_i is modulated by a pseudo-noise signal, respectively the beacon in our specific case, p_i whereas $p_i \in \{-1,1\}, i \in \mathbf{N}$ and previously scaled by a constant $\alpha_i \geq 0$. Therefore, the spread spectrum Watermark now consists of

$$w_i = \alpha_i \cdot b_i \cdot p_i \quad i \in \mathbf{N} \tag{1.12}$$

Afterwards, the spread spectrum Watermark w_i is added to the line-scanned digital video signal v_i yielding the new, Watermarked video signal

$$\tilde{v} = v_i + w_i = v_i + \alpha_i \cdot b_i \cdot p_i \tag{1.13}$$

Due to the noisy appearance of p_i, the spread spectrum watermark w_i is also noise-like and therefore difficult to detect and remove.
In ordinary Watermarking schemes, p_i is typified as the secret key. As already noticed, our proposed scheme doesn't need a secret key, therefore the p_i sequence is a beacon known to both the broadcaster and the monitoring agency. Hartung and Girod proposed to create p_i with the help of feed-back shift registers producing m-sequences or chaotic physical processes. However we propose to use a public and arranged sequence which does not itself cause interference itself and is frequently repeated.

Retrieval of the Watermark

The proposed methods rely on *Blind Watermarking* techniques and therefore do not need the original video stream in the retrieval process. To correctly decode the information, the ”secret” beacon p_i must be known. Due to the public nature of p_i in our case, the monitoring agency knows the sequence.
Optionally, the Watermarked stream can be high-pass filtered in order to improve the performance of the overall Watermarking system. Afterwards, the possibly Watermarked video stream $\overline{v}$ is multiplied by the same noise-like beacon stream p_i that was used in the embedding process.

$$\begin{aligned} s_j &= \sum_{j \cdot cr \leq i < (j+1) \cdot cr} p_i \cdot \overline{\tilde{v}} = \underbrace{\sum_{j \cdot cr \leq i < (j+1) \cdot cr} p_i \cdot \overline{v}}_{\Sigma_1} \\ &+ \underbrace{\sum_{j \cdot cr \leq i < (j+1) \cdot cr} p_i \cdot \overline{p_i \cdot \alpha_i \cdot b_i}}_{\Sigma_2} \approx \sum_{j \cdot cr \leq i < (j+1) \cdot cr} p_i^2 \cdot \alpha \cdot b_i \end{aligned} \tag{1.14}$$

We now assume, in accordance with Hartung and Girod, that Σ_1 is zero due to the fact that the video signal has been filtered out. Furthermore, we assume that $\overline{p_i \cdot \alpha_i \cdot b_i} \approx p_i \cdot \alpha_i \cdot b_i$ which therefore means, that the high-pass filtering has negligible influence on the white pseudo-noise Watermark signal. Following the proposal of Hartung and Girod, the sign of the correlation sum is the embedded information bit

$$sign(s_j) = sign(a_j) = a_j \tag{1.15}$$

Verifying the Signature

It is possible for the monitoring agency to verify the signature which is represented by the extracted bit sequence. The method V_P verifies the signature with the help of the corresponding public key which is taken from the Certificate in order to be sure, that only the public key belonging to the correct broadcaster is used.

But verification alone is not sufficient in our case. The monitoring agency has to be sure, that the broadcaster keeps to the preassigned streaming plan. Therefore, a signature algorithm was used which has the characteristic called message recovery. This means, the message can be extracted from the signature after verifying it. See the algorithm 1 for more details about the verification procedure.

Due to the fact, that the broadcaster signed a hash of *m* (see 1.3), the monitoring agency has to look up the hash in his hash table transferred to him beforehand (4.0.0). So he can be sure that the extracted and verified hash belongs to the correct broadcaster. Furthermore, the MA can be sure, that the broadcaster broadcasted the video stream in a given time-frame by comparing the embedded time-frame ID-time in *m*. If the time-stamp is not within a 60 second time-frame, it could be possible, that someone attempted to replay the recorded video stream. It might also be possible, that is is not the original broadcaster is trying to cheat, but that another broadcaster has recorded the stream and is attempting to stream/broadcast it.

5. Conclusions and Future Work

The schemes proposed in this paper may be viewed as attractive to both broadcasters and rights agencies. This model provides the broadcaster and the rights entity with an automated and trust worthy method for measuring the exploitation of protected works. The paper introduces the concept of an independent third party that monitors and balances the interests of the broadcaster and rights entity.

We discuss the new technologies and distribution models faced by the entertainment and broadcasting sectors. We evaluate established short signature schemes, such as Nyberg-Rueppel, that could be integrated into a final system. Boneh et.al [Boneh and Franklin, 1999] proposed a public key encryption scheme in which there is one public encryption key, but many private decryption keys. This scheme could be used if multimedia content should be encrypted and distributed over given channels. However, in our specific case, we do not focus on encrypting the content. The goal is authentication and non-repudiation for the broadcaster, so that the MA is able to uniquely identify the sender.

Though, a similar schmeme could be used for the same purposes. Due to the fact, that only one unique public key exists, but many corresponding private keys, the broadcaster could encrypt a secret value with this public key and put the ciphertext into the media stream. This has several advantages. The MA as well as the EX could obtain a private key for the ciphertext ans used to decrypt the content. This means that more than one corresponding private key could be used for different monitoring agencies. In addition, authentication is also given in this context. Due to the fact that there is only one public key which is obtained by the broadcaster, only this broadcaster could encrypt the secret value. But this postulates that the public key is not ”public” in general.

References

Alfred J. Menezes, Paul C. van Oorschot and Vanstone, Scott A. (2001). *Handbook of Applied Cryptography*.

Ateniese, G. and de Medeiros, B. (1999). A signature scheme with message recovery as secure as discrete logarithm.

Ateniese, G. and de Medeiros, B. (2003). Efficient group signatures without trapdoors.

Ateniese, Giuseppe and de Medeiros, Breno. A provably secure nyberg-rueppel signature variant with applications.

Boneh, Dan and Franklin, Matthew (1999). An efficient public key traitor tracing scheme. pages 338–353. Springer-Verlag.

Carlisle Adams, Steve Lloyd (2002). *Understanding PKI: Concepts, Standards, and Deployment Considerations*, volume 2. Addison-Wesley Professional.

Chen, B and Wornell, G W (1999). Provably robust digital watermarking. In *in Proc. SPIE Multimedia Systems and Applications II*, pages 43–54.

Choudhury, Suranjan (2002). *Public Key Infrastructure and Implementation and Design*, volume 1. Wiley & Sons.

Cox, Ingemar, Kilian, Joe, Leighton, Tom, and Shamoon, Talal (1997). Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687.

Coxy, Ingemar J., Kiliany, Joe, Leightonz, Tom, and Shamoony, Talal (1996). A secure, robust watermark for multimedia.

Eggers, J J, Su, J K, and Girod, B (2001). Performance of a practical blind watermarking scheme. In *in Proc. of SPIE Vol. 4314: Security and Watermarking of Multimedia Contents III*.

Halevi, S and Krawczyk, H (2006). Strengthening digital signatures via randomized hashing. In *In Cynthia Dwork, editor, Advances in Cryptology - CRYPTO 2006, volume 4117 of Lecture*, page 00. Springer.

Hartung, Frank and Girod, Bernd (1998). Watermarking of uncompressed and compressed video. *Signal Processing*, 66(3):283–301.

Kirovski, D, Malvar, H, and Yacobi, Y (2001). A dual watermarking and fingerprinting system. Technical report, ACM Multimedia.

Lu, Chun-Shien (2005). *Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property*. IDEA GROUP PUBLISHING.

Mauro Barni, Franco Bartolini (2004). *Watermarking Systems Engineering - Enabling Digital Assets Security and Other Applications*. Marcel Dekker, Inc.

Naccache, David and Stern, Jacques (2001). Signing on a postcard. *Lecture Notes in Computer Science*, 1962:121–123.

Nyberg, Kaisa and Rueppel, Rainer A. (1993). A new signature scheme based on the dsa giving message recovery. In *Proceedings of the 1st ACM CCCS*, Fairfax. ACM.

Peitz, M and Waelbroeck, P (2003). Piracy of digital products: A critical review of the economics literature. cesifo working paper series no. *Information Economics and Policy*, (1071):2003.

Ruanaidh, O, Dowling, W J, and Boland, F M (1996). Watermarking digital images for copyright protection. pages 250–256.

Seitz, Juergen (2005). *Digital Watermarking for Digital Media*. Information Science Publishing.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27.

Wojciech Mazurczyk, Zbigniew Kotulski (2007). Covert channel for improving voip security. In *Advances in Information Processing and Protection*.

Zhao, Jian (1997). Applying digital watermarking techniques to online multimedia commerce. In *In Proc. of the International Conference on Imaging Science, Systems, and Applications, Las Vegas*.

Zuccherato, Robert (2000). Elliptic curve cryptography support in entrust.