# AN ANONYMOUS WATERMARKING SCHEME FOR CONTENT DISTRIBUTION PROTECTION USING TRUSTED COMPUTING

Adrian Leung* and Geong Sen Poh

*Information Security Group*
*Royal Holloway, University of London*
*Egham, Surrey, TW20, 0EX, UK*

Keywords: Buyer-Seller Watermarking, Asymmetric Fingerprinting, DRM, Trusted Computing, Ubiquitous Computing.

Abstract: Many Content Distribution Protection (CDP) schemes (e.g. Buyer-Seller Watermarking and Asymmetric Fingerprinting) have been proposed to address the problem of illegal distribution of copyrighted content. All of the existing CDP schemes rely on a Trusted Third Party in one way or another to achieve the desired security objectives. In this paper, using the functionalities of Trusted Computing, we present an anonymous CDP watermarking scheme, which minimises the reliance on a Trusted Third Party. Our scheme allows a buyer to anonymously purchase digital content, whilst enabling the content provider to blacklist the buyers that are distributing content illegally.

## 1 INTRODUCTION

Illegal distribution of copyrighted digital content (e.g. music and movies) through computer networks poses a major challenge to the digital content industries. On the other hand, the ease of content distribution also presents an opportunity for content providers to reach a large pool of consumers efficiently. Hence, the challenge for content providers is how to prevent or deter illegal distribution of copyrighted materials, whilst embracing this new opportunity.

One of the technical means for detering illegal content distribution is for the content provider to embed a unique watermark into every piece of content. If illegal copies of the content are found, the content provider should be able to trace it back to the original buyer. This approach suffers from two problems. Firstly, an honest buyer may be wrongly accused (framed) of illegal distribution (e.g. if the content provider matches the wrong identity to the illegal copies of the content). Secondly, it is also possible for a malicious buyer to claim that an illegal copy was in fact leaked by the content provider.

To address these problems, two types of content distribution protection (CDP) scheme have been proposed to protect the interests of both buyers and sellers, namely, the *Buyer-Seller Watermarking* (BSW) schemes (Memon and Wong, 2001) and the *Asymmetric Fingerprinting* (AF) schemes (Pfitzmann and Schunter, 1996). These schemes require a buyer watermark in addition to a watermark generated by the seller. Subsequently, in order to preserve a buyer's privacy, several anonymous BSW and AF schemes (Pfitzmann and Waidner, 1997; Camenisch, 2000; Ju et al., 2002; Choi et al., 2003; Lei et al., 2004) have also been proposed.

In BSW schemes, a *Trusted Third Party* (TTP) generates buyer watermarks, while in the AF schemes a buyer generates its own watermark, which is proven to be well-formed to the content provider (using zero-knowledge proofs). In general, both of these approaches prevent an honest buyer from being framed, as well as a malicious buyer from denying that he has illegally distributed copyrighted content. If buyer privacy is desired, then a TTP can be employed to provide buyers with certified pseudonyms.

**Motivation** The requirement for an (online) trusted third party in existing BSW and AF schemes, either to generate the buyer watermarks, or to provide

*This author is supported by the British Chevening/Royal Holloway Scholarship, and the European Commission under contract IST-2002-507932 (ECRYPT).

Sen Poh G. and Leung A. (2007).
AN ANONYMOUS WATERMARKING SCHEME FOR CONTENT DISTRIBUTION PROTECTION USING TRUSTED COMPUTING.
In *Proceedings of the Second International Conference on Security and Cryptography*, pages 319-326
DOI: 10.5220/0002124803190326

pseudonyms for buyers, represents a major constraint. We are interested in removing this constraint, so that the schemes are more scalable and suitable for use in distributed environments. Trusted Computing offers some interesting security functionalities which may be used to meet this objective.

Trusted Computing (TC) is a technology that has been developed to enhance the security of computing platforms in increasingly ubiquitous environments. This objective is realised through the incorporation of a hardware component, known as a Trusted Platform Module (TPM), into computing platforms. The TPM provides the platform with a foundation of trust (so-called "roots of trust") as well as the basis on which a suite of TC security functionalities can be built. As a result, users can gain greater assurance that the platform with which they are interacting is behaving in the expected manner (Balacheff et al., 2003; Mitchell, 2005).

Tomsich and Katzenbeisser (Tomsich and Katzenbeisser, 2000) proposed a watermarking framework that uses tamper-proof hardware to protect copyrighted content. Using the functionalities of TC, we take this approach a step further, and offer a concrete construction of an anonymous CDP scheme.

**Contributions** In this paper, we propose an anonymous CDP watermarking scheme using two TC functionalities, namely Direct Anonymous Attestation (DAA) and Integrity Measurement, Storage and Reporting (IMSR). Using the DAA protocol, our scheme minimises reliance on a TTP for privacy protection as the buyer can generate verifiable pseudonyms on its own. As a result, we are able to reduce the communication overheads, and hence improve the overall efficiency compared to BSW and AF schemes. A second contribution of our scheme is that, through the use of IMSR, the content provider is able to obtain assurance that a buyer-generated watermark is well formed. Our scheme also provides the following security features: framing resistance, user anonymity, content information confidentiality, unlinkability (even against the TTP), and transaction linkability.

**Organisation** The remainder of this paper is organised as follows. In Section 2, we discuss the various CDP security issues. Section 3 describes the TC functionality that is used in our CDP watermarking scheme. In Section 4, we present our anonymous CDP watermarking scheme. In the penultimate section, the security of the scheme is analysed, and, finally, conclusions are drawn in Section 6.

# 2 CDP SECURITY ISSUES

In this section, we examine various security issues arising from content distribution.

## 2.1 A CDP Threat Model

The potential security threats that may be posed to content buyers and content providers are as follows.

1. **Illegal Content Distribution** A malicious user may illegally distribute content (which may have earlier been legally purchased from a content provider), resulting in the content being used by others without the appropriate payment being made to the content provider. This translates to a potential loss of revenue for the content provider.
2. **Framing** To deter illegal content distribution, the content provider can employ a digital watermarking scheme, whereby a unique seller-generated watermark is embedded into every piece of content bought by the buyer. Such a scheme, however, does not prevent an honest buyer from being falsely accused (framed) of illegal content distribution. This is a problem if there is no way for the buyer to challenge the decision and prove his/her innocence.
3. **Information Disclosure**
   - **Buyer's Personally Identifiable Information (PII)** During the process of content purchase, a buyer's PII, such as his/her identity or physical location, may be revealed (either willingly or unwillingly) to a content provider or passive eavesdropper.
   - **Content Information** By observing the type of content that a buyer purchases, a passive adversary may gradually build up a profile of the buyer. This information may later be used to infer or predict future patterns and habits of the buyer. The privacy of the buyer is potentially compromised as a result.
4. **Profile Linking** Colluding content providers may buy, sell or exchange information about their buyers. Such collusion could not only provide content providers with monetary benefits, but also enhance their business intelligence as they are able to build a more comprehensive profile of their buyers. With the aid of a TTP, buyers can employ privacy enhancing mechanisms to protect their identity when they interact with content providers. The consequences for buyer privacy could be even more serious if a TTP decides to collude with content providers.

## 2.2 CDP Security Requirements

Based on the aforementioned threats, we derive a corresponding set of security requirements:

1. **Framing Resistance** It should not be possible for the content provider to falsely accuse an honest buyer of illegal content distribution.
2. **User Anonymity** Unique identifying information for a buyer (such as a long lived key) should not be divulged to a content provider during the content purchasing process. A buyer may interact with content providers using pseudonyms.
3. **Content Information Confidentiality** Eavesdroppers (listening to the communications between a buyer and content provider) should not be able to determine the type of content that is being purchased by the buyer.
4. **Unlinkability** Colluding content providers should not be able to link the activities of the same buyer. Similarly, when a TTP colludes with a content provider, they should not be able to correlate the actions of a particular buyer. In other words, it should be impossible for colluding content providers to tell if two sets of prior content purchase transactions (made with different providers) had originated from the same or different buyers.
5. **Transaction History** For billing or other purposes (e.g. loyalty rewards), it may be necessary for a content provider to maintain the transaction histories of its buyers. That is, a content provider may need to be able to identify whether a particular buyer is a repeat buyer (and, if so, which one) or a first time buyer, whilst still being unable to determine the unique identity of the buyer.
6. **Blacklisting of Rogue Buyers** In the event that illegal copies of copyrighted content are found (e.g. on the Internet), content providers should be able to blacklist the buyers of these copies of the content.

# 3 TRUSTED COMPUTING OVERVIEW

In this section, we introduce the core Trusted Computing functionalities (according to v1.2 of the TCG TPM specifications (Trusted Computing Group (TCG), 2004)) that are employed in our Anonymous CDP Watermarking Scheme, namely the Integrity Measurement/Reporting Mechanisms and the Direct Anonymous Attestation Protocol.

## 3.1 Integrity Measurement, Storage and Reporting

Integrity Measurement, Storage and Reporting (IMSR) is one of the key features of Trusted Computing. IMSR is built upon the three Roots of Trust in a trusted platform: a *root of trust for measurement* (RTM), a *root of trust for storage* (RTS), and a *root of trust for reporting* (RTR). Together, they allow a verifier to reliably ascertain the exact operational state of a platform, and hence obtain evidence of a platform's behaviour. This functionality is extremely important, as a platform may potentially enter one of a wide range of operational states, including insecure and undesirable states.

**Integrity Measurement** IMSR begins with the process of integrity measurement. The RTM, a computing engine in the TPM, measures a platform's operational state and characteristics. The measured values are known as integrity metrics, since they convey information about the platform's current state (and hence trustworthiness).

**Integrity Storage** Using the RTS, these integrity metrics are then put into a log called the *Stored Measurement Log* (SML). At the same time, a digest (i.e. a cryptographic hash computed using Secure Hash Algorithm 1 (SHA-1) (National Institute of Standards and Technology (NIST), 2002)) of the same integrity metrics is saved in one of TPM's internal *Platform Configuration Registers* (PCRs). The SML contains the sequence of all measured events, and each sequence shares a common measurement digest. Since an SML may become fairly large, it does not reside in the TPM. Furthermore, the SML does not require the protection provided by the TPM, as attacks against the SML can easily be detected. On the other hand, there are only a limited number of PCRs in the TPM to hold the measurement digests. So, to ensure that previous and related measured values are not ignored, and the order of operations is preserved, new measured values are appended to the previous measurement digest values and re-hashed. This technique is also known as *extending* the digest.

**Integrity Reporting** The final phase of the IMSR process is Integrity Reporting. The RTR has two main responsibilities during Integrity Reporting:

1. to retrieve and supply a challenger with the requested integrity metrics (i.e. the relevant portion of the SML and the corresponding PCR values).

2. to *attest to* (prove) the authenticity of the integrity metrics (in step 1) to a challenger. This is done by signing the PCR values using one of the TPM's trusted platform identities, also known as an *Attestation Identity Key* (AIK).

To verify the integrity measurements, the verifier computes the expected measurement digest (using the relevant portion of the SML) and compares it with the corresponding PCR values. The verifier also checks the signature on the PCR values. In the context of Trusted Computing, the process of integrity reporting is also often referred to as *Attestation*.

## 3.2 Direct Anonymous Attestation

Direct Anonymous Attestation (DAA) (Brickell et al., 2004) is a special type of signature scheme that can be used to anonymously authenticate a TCG v1.2 compliant platform to a remote verifier. The key feature that DAA provides, in the context of Trusted Computing, is the capability for a TPM (a prover) to convince a remote verifier that:

- it is indeed a genuine TPM without revealing any unique identifiers;
- an AIK is held by a TPM, without allowing multiple verifiers to collude and link transactions involving different AIKs from the same platform.

These features help to protect the privacy of a TPM user. Another important feature of DAA is that the powers of the supporting TTP (DAA Issuer) are minimised, as it cannot link the actions of users, and thus compromise the user's privacy.

The DAA scheme is made up of two sub-protocols: *DAA Join* and *DAA Sign*. We now provide a simplified description of these two sub-protocols.

**DAA Join Protocol** The Join protocol enables the TPM to obtain a DAA Certificate from the DAA Issuer.

Let $(n, S, Z, R)$ be the public key of the DAA Issuer, where $n$ is an RSA modulus, and $S$, $Z$ and $R$ are integers modulo $n$. We assume that the TPM is already authenticated to the DAA Issuer via its *Endorsement Key*, $EK$. Each TPM will only have one EK key pair (usually created by a TPM manufacturer), and a TPM may be uniquely identified by its EK.

The platform (TPM) first generates a DAA secret value, $f$, and makes a commitment to $f$ by computing $U = R^f S^{v'} \bmod n$, where $v'$ is a value chosen randomly to "blind" $f$. The platform (TPM) also computes $N_I = \zeta_I^f \bmod \Gamma$, where $\zeta_I$ is derived from the DAA Issuer's name, and $\Gamma$ is a large prime. The platform (TPM) then sends $(U, N_I)$ to the DAA Issuer, and convinces the DAA Issuer that $U$ and $N_I$ are correctly formed (using a Zero Knowledge Proof (Goldwasser et al., 1989)). If the DAA Issuer accepts the proof, it will sign the hidden message, $U$, by computing $A = (\frac{Z}{US^{v''}})^{1/e} \bmod n$, where $v''$ is a random integer and $e$ is a random prime. The DAA Issuer then sends the platform (i.e. the TPM) the triple $(A, e, v'')$, and proves that $A$ was computed correctly. The DAA Certificate is then $(A, e, v = v' + v'')$.

**DAA Sign Protocol** The Sign protocol allows a platform to prove to a verifier that it is in possession of a DAA Certificate, and, at the same time, to sign and authenticate a message.

The platform signs a message, $m$, using its DAA Secret, $f$, its DAA Certificate, and the public parameters of the system. The message, $m$, may be an Attestation Identity Key (AIK) generated by the TPM, or an arbitrary message. The platform also computes $N_V = \zeta^f \bmod \Gamma$ as part of the signature computation (the selection of $\zeta$ will be be discussed in the next section). The output of the Sign protocol is known as the DAA Signature, $\sigma$.

The verifier verifies the DAA Signature, $\sigma$, and, upon successful verification of $\sigma$, is convinced that:

1. The platform has a DAA Certificate $(A, e, v)$ from a specific DAA Issuer, and hence it is a valid TPM. This is accomplished by a zero-knowledge proof of knowledge of a set of values $f, A, e$ and $v$ such that $A^e R^f S^v \equiv Z \pmod{n}$.
2. A message, $m$, was signed by the TPM using its DAA secret, $f$, where $f$ is the same as the value in the DAA Certificate.

In summary, once a platform (TPM) has obtained a DAA Certificate (which only needs to be done once), it is able to subsequently DAA-Sign as many AIKs as it wishes, without involving the DAA Issuer.

**Variable Anonymity** Anonymity and unlinkability are afforded to a user via the use of two parameters: $\zeta$, also referred to as the "base", and the AIK. The choice of the base directly affects the degree of anonymity afforded to a user of a TPM. If perfect anonymity is desired, then a different, random, base value should be used for every interaction with a verifier. Conversely, if the same base value is used for every interaction with a verifier, then the verifier can identify that this is the same TPM. In addition, if the same base value is used to interact with different verifiers, then they are able to correlate the activities of a particular TPM.

A TPM is capable of generating multiple platform identities, simply by generating different AIK key pairs. Different AIKs may therefore be used to interact with different verifiers so that the TPM remains unlinkable (provided the base is different).

# 4 OUR PROPOSED SCHEME

In this section, we present our anonymous content distribution protection watermarking scheme. The primary objective of the scheme is for the buyer to anonymously purchase digital content, whilst allowing a seller to blacklist any buyer platforms that are distributing content illegally. Using the aforementioned TC functionalities, our scheme also allows a buyer to generate verifiable pseudonyms, and to convince a content provider that the buyer generated watermark is well formed, both without the involvement of a TTP. Our proposed solution is also designed to meet all the security requirements set out in section 2.2.

First, we introduce the entities participating in the protocol. Next, we state the assumptions upon which the scheme is based. Finally, we describe the operation of the scheme.

## 4.1 The Entities

The entities participating in our scheme are:

- the **buyer** of digital content (e.g. music, video, podcasts, and etc).
- the **platform**, which consists of the TPM and its host. The platform is also the device which a content buyer will use to interact with other entities.
- the **seller** (also referred to as the content provider) of some digital content.
- the **DAA Issuer**, which is also the authority that issues DAA Certificates to legitimate platforms.

## 4.2 Assumptions

The correct working of our scheme relies upon a number of assumptions:

- The content buyer is already authenticated to the platform (via some out of band mechanism such as the one given in (Gehrmann et al., 2004)) that is used for the CDP watermarking scheme. As such, the buyer and the platform will collectively be referred to as the Buyer Platform.
- The device/platform running the CDP scheme is equipped with TCG functionality conforming to v1.2 of the TCG specifications (Trusted Computing Group (TCG), 2004).
- The parties involved have agreed on the use of a homomorphic encryption algorithm $Enc_K(\cdot)$ (e.g. the Paillier probabilistic encryption scheme (Paillier, 1999) that is homomorphic with respect to addition).
- The embedding operation $\otimes$ is public knowledge and the security of the embedding relies on the key used to embed the watermark $W$. (In our case this key is a random permutation $\rho$). In addition, the watermark $W$ embedded with $\otimes$ is *collusion resistant*, which means that it is computationally infeasible for the attackers to remove $W$ by comparing different copies of the content.

## 4.3 The Scheme

Before describing the scheme, it is first necessary to introduce some notation (see Table 1).

Table 1: Notation.

| *Notation* | Description |
|---|---|
| $BP$ | The Buyer Platform |
| $S$ | The Seller or Content Provider |
| $DI$ | The DAA Issuer |
| $f$ | A DAA secret value generated by the TPM |
| $ID_A$ | The identity of a principal, $A$ |
| $(EK_{pk}, EK_{sk})$ | The pair of Public and Private Endorsement Keys |
| $(AIK_{pk}, AIK_{sk})$ | A pair of Public and Private Attestation Identity Keys |
| $X'$ | Watermarked Content |
| $X \otimes W$ | Embed W into X with the embedding operation, $\otimes$ |
| $\rho$ | A random permutation function |
| $H$ | A cryptographic hash-function |
| $Enc_K(M)$ | The encryption of a message, $M$, using the key $K$ |
| $Dec_K(M)$ | The decryption of a message, $M$, using the key $K$ |
| $Sig_K(M)$ | A signature on a message, $M$, signed using the key $K$ |

The proposed CDP watermarking scheme involves three distinct phases, namely, the *Join Phase*, the *Watermarking Phase*, and the *Content Acquisition Phase*. We now describe the workings of each phase in greater detail.

**Join Phase** The objective of the *Join Phase* is for a buyer platform to obtain a *DAA Certificate* from a *DAA Issuer*. Since the Join Phase of our scheme is identical to the DAA Join Protocol of Section 3.2, we do not describe the sequence of Join Phase steps again. Note that the Join Phase may have taken place before a device is shipped to the content buyer.

**Watermarking Phase** The aim of this phase is for a buyer to contribute a watermark, and for the seller to embed the buyer's watermark into a piece of copyrighted content. The entities involved in this phase are the *Buyer Platform*, *BP* and the *Seller*, S. The sequence of events is as follows:

1. BP generates a watermark, $W$, using the watermark generation function of a reliable watermarking algorithm (e.g. the spread spectrum watermarking algorithm in (Cox et al., 1997)).
2. BP generates an encryption key pair ($BEK_{pk}, BEK_{sk}$), and encrypts the watermark, $W$, using $BEK_{pk}$, to create:
$$Enc_{BEK_{PK}}(W).$$
3. BP (TPM) generates a non-migratable signing key pair ($BSK_{pk}$,$BSK_{sk}$). BP then signs the encrypted watermark, $Enc_{BEK_{pk}}(W)$ (from step 2), and $BEK_{pk}$, to obtain:
$$Sig_{BSK_{sk}}(Enc_{BSK_{pk}}(W), BEK_{pk}).$$
4. BP generates an AIK key pair, ($AIK_{pk}$,$AIK_{sk}$).
5. BP retrieves the Stored Measurement Log (SML), and the corresponding Platform Configuration Register (PCR) values. BP then signs the PCR values using $AIK_{sk}$ (from step 4):
$$Sig_{AIK_{sk}}(PCR).$$
The SML and PCR values provide the evidence that a particular watermarking algorithm was used (by the buyer) to generate the watermark.
6. BP computes $\zeta$ = H($ID_S$). It then creates a pseudonym, $N_v = \zeta^f$ (where $f$ is the DAA Secret generated during the join phase) for use when interacting with the seller.
7. To prove (to the seller) that the AIK (from steps 4) originates from a genuine TPM, the platform DAA-Signs $AIK_{pk}$ using $f$, *DAA Certificate*, and the other public parameters of the system. The output of DAA Sign is the DAA Signature, $\sigma$ (which also includes $\zeta$ and $N_v$).
8. To prove that *BSK* originates from the TPM, BP signs (certifies) $BSK_{pk}$ using $AIK_{sk}$:
$$Sig_{AIK_{sk}}(BSK_{pk}).$$
9. BP sends the following to the Seller:
$$\begin{aligned} BP \to S :& Enc_{BEK_{pk}}(W), AIK_{pk}, BSK_{pk}, BEK_{pk}, \\ & \sigma, Sig_{BSK_{sk}}(Enc_{BSK_{pk}}(W), BEK_{pk}), \\ & Sig_{AIK_{sk}}(BSK_{pk}), SML, Sig_{AIK_{sk}}(PCR). \end{aligned}$$

Upon receiving the last message from the buyer, and, to subsequently incorporate the buyer's watermark into a piece of content, the seller performs the following steps:

1. Verifies the DAA Signature, $\sigma$, and is convinced that:
   - BP is in possession of a legitimate DAA Certificate from a specific DAA Issuer, which implies that a genuine TPM is contained in BP.
   - $AIK_{pk}$ was signed using BP's DAA Secret, $f$. Even though the value of $f$ is never revealed to the seller, the seller knows that the value is related to the one in the DAA Certificate
2. Examines the integrity measurements of the buyer platform. This is achieved by recursively hashing the values in the SML, and then comparing them with the corresponding PCR values. If the outcome is satisfactory, the seller is convinced that a reliable watermarking algorithm was used by the buyer platform to generate its watermark, $W$.
3. Verifies $Sig_{AIK_{sk}}(BSK_{pk})$.
4. Verifies $Sig_{BSK_{sk}}(BEK_{pk})$.
5. Generates a seller watermark, $V$, and then embeds it into the Content, $X$, to create:
$$X' = X \otimes V.$$
6. Encrypts $X'$ (from step 3) using $BEK_{pk}$ to get:
$$E(X') = Enc_{BEK_{pk}}(X').$$
7. Permutes $Enc_{BEK_{pk}}(W)$ (received from buyer) to get $E(\rho W)$.
8. Permuted watermark is then embedded into $X'$ as follows:
$$E(X' \otimes \rho W) = E(X') \otimes E(\rho W),$$
which follows because of the homomorphic property of the encryption algorithm.
9. The encrypted, watermarked content is then sent back to the buyer.
$$S \to B : E(X' \otimes \rho W).$$

**Content Acquisition Phase** When the buyer receives the encrypted, watermarked content, $E(X' \otimes \rho W)$, from the seller, he decrypts it using $BEK_{sk}$, to retrieve the watermarked content:
$$(X' \otimes \rho W).$$

The watermarked content is now ready for the buyer's consumption (e.g. viewing or listening).

## 5 SECURITY ANALYSIS

We now consider how the proposed scheme meets the security requirements outlined in Section 2.2.

**Framing Resistance** Framing of the buyer by the content provider is not possible since neither of them have knowledge of the watermark embedded in the final copy possessed by the buyer. This can be observed from the watermarking phase, in which $W$ is embedded into content in encrypted form and is permuted with $\rho$. The embedding through homomorphic encryption prevents the content provider from knowing the watermark, while $\rho$ randomises $W$ and thus prevents the buyer from knowing the embedded watermark in the content.

**User Anonymity** The Endorsement Key, *EK*, which is also the long-lived and unique identity of a platform, is never disclosed to a content provider during content purchase. Buyers interact with content providers using *AIKs*, which act as pseudonyms. Since it is computationally infeasible for content providers to make any association between a specific *EK* and an *AIK* from the same platform, buyers will remain anonymous to content providers.

**Transaction History** It may be necessary for content providers to link a repeat content buyer (e.g. for customer loyalty rewards or discounts). This can be achieved, without any compromise of a buyer's privacy or anonymity, if a content buyer uses the same $N_v$ value to interact with a particular content provider. Note that it is not necessary for a content buyer to store the value $N_v$, as the same value will be recovered during re-computation (since the values of $\zeta$ and $f$ should remain unchanged).

**Content Information Confidentiality** The piece of copyrighted content is encrypted with the buyer's public encryption key, $BEK_{pk}$. As such, the content is protected from eavesdroppers.

**Unlinkability/Collusion Resistance** Buyers interact with different content providers using different *AIK* keys and $N_v$ values. It is computationally infeasible for colluding content providers to link these keys and values to a particular content buyer. Hence a buyer's content purchasing activities with different content providers are unlinkable.

Since a DAA Issuer knows which TPMs with *EKs* possess valid DAA Certificates, it may collude with a content provider in an attempt to link these *EKs* with the corresponding *AIKs*. To be able to make this link, an entity would require knowledge of the TPM's DAA Secret value, $f$. Again this is computationally infeasible because of the way that a DAA Certificate is created, and since $f$ never leaves the TPM.

Our scheme is therefore resistant to two or more colluding content providers as well as a DAA Issuer colluding with one or more content providers.

**Rogue Blacklisting** A content provider may blacklist malicious content buyers (i.e. those found to be distributing content illegally), so as to prevent them purchasing content in future. In other words, if a malicious buyer revisits the content provider, it should be possible for the content provider to recognise that this buyer platform is malicious, whilst remaining anonymous. This can be achieved by blacklisting the pseudonyms, i.e. $N_v$ values, of all known platforms of rogue buyers. The only way in which a rogue buyer could avoid detection would be to obtain a new pseudonym, $N_v$. This would require the buyer to have a new value for $f$. Although it is possible for a TPM to generate a new value for $f$, it is unlikely that the buyer platform will be able to obtain a new DAA Certificate for it from a DAA Issuer.

Furthermore, if a DAA Certificate and the value $f$ are found in the public domain (e.g. on the Internet), then they should be distributed to all potential content providers, who should add them to their lists of rogue keys. These rogue platform identification methods could have the advantage of eliminating the need for a centralised revocation authority.

**Efficiency** Our watermarking scheme is more efficient than existing schemes, since there is no need for the buyer to interact with a TTP to obtain a pseudonym every time the buyer wishes to buy some content. Once the buyer platform has obtained a DAA Certificate, it is able to generate an arbitrary number of verifiable pseudonyms (AIKs) on its own.

## 6 APPLICATION SCENARIO

Since an online TTP is no longer required, our watermarking scheme is particularly suitable for environments which are highly dynamic and mobile, such as the mobile ubiquitous environment (depicted in figure 1) as envisaged by the Mobile VCE[2] Core 4 research programme on Ubiquitous Services.

In a mobile ubiquitous environment, consumers (through one of their mobile devices and via some network access technologies) will be able to seamlessly discover, select, and access a rich offering of services and content from an array of service and content providers. Consumers will be able to interact with content providers, and have access to content,

[2] http://www.mobilevce.com/

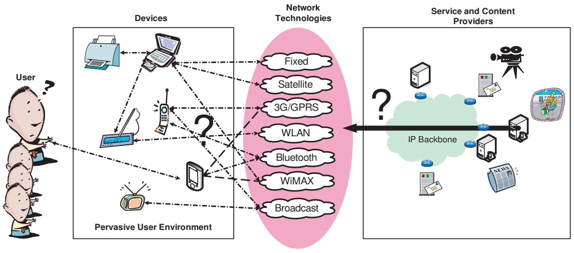

Figure 1: A Mobile Ubiquitous Environment.

instantly, while on the move. Unfortunately, in such environments, the tasks of (illegally) distributing or propagating content is also made easier.

Our proposed CDP scheme may thus be employed to address this problem, as it would be infeasible to have an online TTP in such environments.

## 7 CONCLUSION

In this paper, we identified the security threats that may arise during the process of content purchase and distribution. We derived a corresponding set of security requirements. We then presented an anonymous CDP watermarking scheme, using TC functionality. Our subsequent security analysis has shown that our scheme is able to satisfy all the identified security requirements. We also showed a potential application scenario, a mobile ubiquitous environment, in which our scheme could be employed.

## ACKNOWLEDGEMENTS

The work reported in this paper has formed part of the Ubiquitous Services Core Research Programme of the Virtual Centre of Excellence in Mobile & Personal Communications, Mobile VCE, www.mobilevce.com. This research has been funded by the DTI-led Technology Programme and by the Industrial Companies who are Members of Mobile VCE. Fully detailed technical reports on this research are available to Industrial Members of Mobile VCE.

The authors would like to thank Chris Mitchell and Keith Martin for their helpful comments.

## REFERENCES

Balacheff, B., Chen, L., Pearson, S., Plaquin, D., and Proudler, G. (2003). *Trusted Computing Platforms: TCPA Technology in Context*. Prentice Hall, NJ, USA.

Brickell, E., Camenisch, J., and Chen, L. (2004). Direct anonymous attestation. In *11th ACM Conf. on Computer and Communications Security*, pages 132–145. ACM Press.

Camenisch, J. (2000). Efficient anonymous fingerprinting with group signatures. In *6th Intl. Conf. on the Theory and Application of Cryptology and Information Security*, pages 415–428. Springer LNCS 1976.

Choi, J.-G., Sakurai, K., and Park, J.-H. (2003). Does it need trusted third party? Design of buyer-seller watermarking protocol without trusted third party. In *1st Intl. Conf. on Applied Cryptography and Network Security*, pages 265–279. Springer LNCS 2846.

Cox, I. J., Killian, J., Leighton, T., and Shamoon, T. (1997). Secure spread spectrum watermarking for multimedia. *IEEE Trans. on Image Processing*, 6(12):1673–1687.

Gehrmann, C., Mitchell, C. J., and Nyberg, K. (2004). Manual authentication for wireless devices. *Cryptobytes*, 7(1):29–37.

Goldwasser, S., Micali, S., and Rackoff, C. (1989). The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208.

Ju, H. S., Kim, H. J., Lee, D. H., and Lim, J. I. (2002). An anonymous buyer-seller watermarking protocol with anonymity control. In *5th Intl. Conf. on Information Security & Cryptology*, pages 421–432. Springer LNCS 2587.

Lei, C.-L., Yu, P.-L., Tsai, P.-L., and Chan, M.-H. (2004). An efficient and anonymous buyer-seller watermarking protocol. *IEEE Trans. on Image Processing*, 13(12):1618–1626.

Memon, N. and Wong, P. W. (2001). A buyer-seller watermarking protocol. *IEEE Trans. on Image Processing*, 10(4):643–649.

Mitchell, C. J., editor (2005). *Trusted Computing*. IEE Press, London.

National Institute of Standards and Technology (NIST) (2002). Secure Hash Standard. Federal information processing standards publication (FIPS) 180-2.

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT'99*, pages 223–238. Springer LNCS 1592.

Pfitzmann, B. and Schunter, M. (1996). Asymmetric fingerprinting. In *EUROCRYPT'96*, pages 84–95. Springer LNCS 1070.

Pfitzmann, B. and Waidner, M. (1997). Anonymous fingerprinting. In *EUROCRYPT'97*, pages 88–102. Springer LNCS 1233.

Tomsich, P. and Katzenbeisser, S. (2000). Towards a secure and de-centralized digital watermarking infrastructure for the protection of intellectual property. In *1st Intl. Conf. in E-Commerce & Web Technologies*, pages 38–47. Springer LNCS 1875.

Trusted Computing Group (TCG) (2004). TCG Specification Architecture Overview. Version 1.2, The Trusted Computing Group, Portland, Oregon, USA.