

# Formal Analysis of Two Buyer-Seller Watermarking Protocols

David M. Williams<sup>1,\*</sup>, Helen Treharne<sup>1</sup>, Anthony T.S. Ho<sup>1</sup>, and Adrian Waller<sup>2</sup>

<sup>1</sup> University of Surrey, Guildford, GU2 7XH, UK

<sup>2</sup> Thales Research and Technology (UK) Ltd., Reading, RG2 0SB, UK  
d.m.williams@surrey.ac.uk

**Abstract.** In this paper we demonstrate how the formal model constructed in our previous work [1], can be modified in order to analyse additional Buyer-Seller Watermarking Protocols, identifying which specific sections of the CSP scripts remain identical and which require modification. First, we model the protocol proposed by Memon and Wong [2], an exemplar of the Offline Watermarking Authority (OFFWA) Model, defined in the framework by Poh and Martin [3]. Second, we model the Shao protocol [4] as an example of a protocol fitting the Online Watermarking Authority (ONWA) Model. Our analysis of the protocols reaffirms the unbinding attack described by Lei *et al.* [5] on the Memon and Wong protocol and we identify a new unbinding attack on the protocol proposed by Shao.

## 1 Introduction

A major benefit of digital media is the ease with which multimedia content can be duplicated and disseminated on a large scale. However, copyright owners are faced with the task of limiting these activities so that they may make a financial gain from licencing such content. Digital Watermarking has been proposed as a suitable deterrent to illegal copying/distribution.

A digital watermark is an imperceptible mark embedded into cover material designed to degrade linearly with the degradation of the content itself (e.g. compression). Embedding a digital watermark into licenced multimedia content enables copyright owners to trace piracy to the original perpetrator. However the buyer must be assured that it is only possible for the copyright owner to gather adequate evidence if and only if an illegal act has taken place.

Qiao and Nahrstedt [6] identified that watermarking schemes alone are inadequate in fulfilling the above requirement. Consequently, asymmetric fingerprinting protocols have been developed to be used in conjunction with digital watermarking schemes. The framework constructed by Poh and Martin [3] specifies the three common requirements that asymmetric fingerprinting protocols aim to satisfy and provides a classification of such protocols into four distinct

---

\* The author's work is sponsored by an EPSRC Thales CASE Award.

protocol models. We will focus on just two of the models, the Online Watermarking Authority (ONWA) and Offline Watermarking Authority (OFWA) Models that together make up the set of protocols more commonly known as the Buyer-Seller Watermarking Protocols.

Our previous work [1] introduced the notion of rigorous analysis of buyer-seller watermarking protocols using a formal analysis technique used previously to check communication protocols [7]. In [1] we accurately represented a buyer-seller watermarking protocol as proposed by Ibrahim *et al.* [8] by constructing a model using the process algebra Communicating Sequential Processes (CSP) [9]. By describing our model in this manner and utilising the tool support associated with CSP we were able to conduct a thorough analysis of all the possible behaviour in the protocol. In this paper we demonstrate how our analysis, of the protocol proposed by Ibrahim *et al.* [8], is easily adapted for further analyses of various buyer-seller watermarking protocols.

We extend Poh and Martin's framework to include those buyer-seller watermarking protocols in which the buyer generates the watermark and the watermarking authority is used to verify that the watermark is well formed, such as [4] and [8]. We do not compare the ONWA and OFWA models, the framework in [3] supplies such a comparison. Instead we analyse a protocol conforming to each model simply to demonstrate that our formal analysis technique can be used for the analysis of each.

Firstly, we model the protocol proposed by Memon and Wong [2] as an example of a protocol that fits the OFWA Model. We analyse our model against a single requirement, which is a necessary test of the protocol's security. Our analysis is sufficient in reaffirming the unbinding attack described by Lei *et al.* [5]. Secondly, we model the ONWA protocol proposed by Shao [4]. We again analyse the protocol against a single requirement, which is sufficient in identifying a new attack found on the protocol.

In Section 2 we summarise the framework constructed by Poh and Martin [3]. Next we define the notation used throughout the paper in Section 3. We then analyse two buyer seller watermarking protocols in Sections 4 and 5. In in Section 6 we discuss how our scripts may be modified to analyse other protocols fitting the ONWA and OFWA models. Finally, we conclude with Section 7 giving a summary of the contributions of this paper and a discussion of further work on this subject.

## 2 Asymmetric Fingerprinting Protocol Framework

A detailed framework for the design and analysis of asymmetric fingerprinting protocols was proposed in [3]. According to the framework asymmetric fingerprinting protocols have the two primary requirements of: (i) Traceability and proof of illegal distribution and (ii) Framing resistance, with a third desirable requirement (iii) Anonymity and unlinkability. Asymmetric protocols fall under four models namely: (i) ZK Model, (ii) ONWA Model, (iii) OFWA Model, and (iv) TKTP Model. Buyer-Seller Watermarking Protocols fall exclusively into either of the ONWA or OFWA models. It is the ONWA and OFWA models on which this paper focuses.

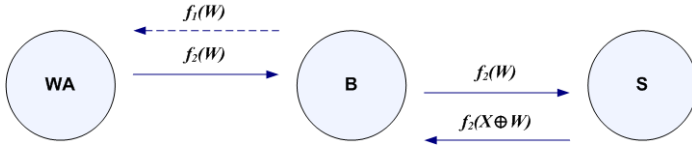


Fig. 1. Offline Watermarking Authority (OFWA) Model

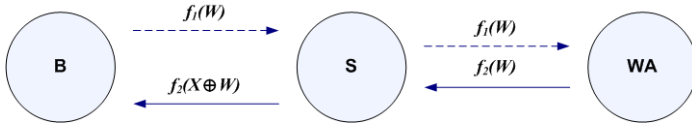


Fig. 2. Online Watermarking Authority (ONWA) Model

2.1 OFWA Model

Figure 1 shows the typical message flow of the OFWA Model. The watermarking authority *WA* generates the value  $f_2(W)$ , where  $f_2$  is a homomorphic encryption algorithm with respect to the  $\oplus$  operator, such as Paillier encryption [10], and sends this value to the buyer. The watermarking authority is regarded as offline as this message exchange between the buyer and watermarking authority can be performed independently of the watermarking insertion.

The buyer may now initiate the transaction between the buyer and the seller by forwarding the value  $f_2(W)$  to the seller. It is important that  $f_2$  is some homomorphic encryption algorithm so that the embedding may be done in the encrypted domain. The result of this embedding  $f_2(X \oplus W)$  is sent to the buyer, where  $X \oplus W$  denotes watermark embedding of the form: the cover material  $X$  is embedded with some mark  $W$  using some key conforming to a suitable watermark embedding scheme [11]. The marked content  $X \oplus W$  is retrieved by the buyer decrypting the message using some inverse function  $f_2^{-1}$ .

The dashed arrows in Figure 1 indicate the additional message flow required when the buyer chooses the watermark, omitted in the framework by Poh and Martin. The buyer generates the value  $f_1(W)$ , where  $f_1$  denotes a function of which the seller is unable to calculate the inverse  $f_1^{-1}$ . The value  $f_1(W)$  is sent, by the buyer, to the watermarking authority. The watermarking authority verifies the value  $W$  by applying the inverse function  $f^{-1}$  to the received message. The message exchange then proceeds in the same manner as when the watermarking authority chooses the watermark.

2.2 ONWA Model

In the Online Watermarking Authority Model (ONWA) the buyer is only required to contact the seller but the watermarking authority is required to participate online. Figure 2 shows the typical message flow of the ONWA Model.

The watermarking authority  $WA$  generates the value  $f_2(W)$ , where  $f_2$  is a homomorphic encryption algorithm and sends this value to the seller. Embedding is conducted in the encrypted domain, using a homomorphic encryption algorithm, and the result  $f_2(X \oplus W)$  is sent to the buyer. The buyer retrieves the marked content  $X \oplus W$  by decrypting the message using some inverse function  $f_2^{-1}$ .

The dashed arrows in Figure 2 indicate the additional message flow required when the buyer chooses the watermark. The buyer generates the value  $f_1(W)$  and sends it to the seller, where  $f_1$  denotes a function of which the seller is unable to calculate the inverse  $f_1^{-1}$ . The seller forwards  $f_1(W)$  on to the watermarking authority. The watermarking authority verifies the value  $W$  by applying the inverse function  $f^{-1}$  to the received message and then proceeds in the same manner as if the watermarking authority had chosen the watermark.

### 3 Notation

In this section we define all abbreviations used throughout the paper for clarity of presentation.

#### 3.1 Protocol Participants

- B : The Buyer, wishing to purchase the digital content.
- S : The Seller, whom owns/provides/distributes the digital content.
- WA : The Watermarking Authority, a trusted third party responsible for generating or verifying generated watermarks ready for embedding.
- CA : The Certification Authority, a trusted third party responsible for generating and distributing public-private key pairs along with public-key certificates. In practice a single agent may act as both the Watermarking Authority and the Certification Authority.
- ARB : The Arbitrator, required in an arbitration protocol to make judgement upon a case of copyright infringement.

We use the convention of using uppercase to represent a specific value where lower case represents a variable of arbitrary value.  $X'$  denotes the cover material uniquely marked with  $V$  enabling the seller to identify exactly which entry to look up in their database once a copy has been found rather than conduct an intractable exhaustive search.

#### 3.2 Cryptographic/Watermarking Primitives

- $(pk_A, sk_A)$  : A public-private key pair, associated with an adopted public key infrastructure (PKI), in which the public key is published and the secret key kept private to agent  $A$ .
- $(\overline{pk}_A, \overline{sk}_A)$  : A public-private key pair used for anonymous certification.
- $(pk_A^*, sk_A^*)$  : A *one time* public-private key pair chosen prior to each protocol run to provide anonymity/unlinkability.
- $E_{pk_A}(m)$  : A message  $m$  encrypted using the public key

- belonging to agent  $A$ .
- $D_{sk_A}(m)$  : A cipher  $m$  may be decrypted using the relevant secret key belonging to agent  $A$  i.e.  $D_{sk_A}(E_{pk_A}(m)) = m$ .
- $Cert_{CA}(pk_A)$ : A X.509 compliant digital certificate [12] constructed by the certification authority  $CA$  such that the agent  $A$  is able to convince others of the validity of  $A$ 's public key.
- $Sign_{sk_A}(m)$  : The digital signature associated with the adopted PKI.
- $h(m)$  : A cryptographic hash of message  $m$  such that anyone in possession of the message  $m$  is able to verify the hash value but the inverse argument does not hold. Anyone in possession of the hash value only is unable to construct the original message  $m$ .
- $X \oplus W$  : Watermark embedding of the form: the cover material  $X$  is embedded with some mark  $W$  using some key conforming to a suitable watermark embedding scheme [11].

### 3.3 CSP

CSP is a process algebra for describing models of interacting systems. A system model is described as a *process* (or collection of processes). CSP processes are defined in terms of the *events* that they can and cannot do. Processes interact by synchronising on events, and the occurrence of events is atomic. The set of events of a process  $P$  is denoted by  $\alpha P$ .

Events may be compound in structure, consisting of a *channel name* and some (or none) *data values*. Thus, events have the form  $c.v_1\dots v_n$ , where  $c$  is the channel name associated with the event, and the  $v_i$  are data values. The *type* of the channel  $c$  is the set of values associated with  $c$  to produce events.

For example, the channel *comm* has type *agents*  $\times$  *agents*  $\times$  *messages*, where *agents* is a set of agents, involved in some message exchange that may send and receive messages over *comm*, and *messages* is the set of all possible messages that the senders may wish to transmit. The events associated with *comm* will be of the form *comm.a.b.m*, where  $a \in \text{agents}$ ,  $b \in \text{agents}$ , and  $m \in \text{messages}$ . The syntax of CSP provides several operators for modelling processes.

$$P ::= a \rightarrow P \mid c?x!v \rightarrow P \mid P_1 \square P_2 \mid \square_i P_i \mid S(p)$$

where  $a$  is a *synchronisation event*,  $c$  is a *communication channel* accepting inputs and sending output values,  $x$  is a data variable,  $v$  is a data value, and  $S(p)$  is a process expression.

The process  $a \rightarrow P$  is initially prepared to engage in an  $a$  event, after which it behaves as  $P$ . The process  $c?x!v \rightarrow P$  is prepared to accept any value for  $x$  along channel  $c$ , provides  $v$  as output, and then behave as  $P$  (whose behaviour can be dependent on  $x$ ). The external choice process  $P_1 \square P_2$  is initially prepared to behave either as  $P_1$  or as  $P_2$ , and the choice is resolved on occurrence of the first event. This can be generalised for an indexed set of processes:  $\square_i P_i$  chooses a process from an  $i$ -indexed set of processes  $P$ .

Processes can be combined together using the parallel ‘||’ composition operator. When processes run in parallel they must synchronise on common events (otherwise the events can occur independently). E.g., in the parallel process:

$$a \rightarrow b \rightarrow Stop \parallel b \rightarrow c \rightarrow Stop$$

both *a* and *c* can occur independently of the other process, but the occurrence of *b* requires both processes to synchronise. We use an indexed parallel operator which enables us to combine the behaviour of similar processes together. During the analysis phase, we also make use of the hiding operator ‘\’ so that we can focus on particular events in a trace.

CSP has a theory of refinement that enables us to compare the behaviour of processes. If a process *P* is refined by a process *Q*, then all of the possible behaviours of *Q* must also be possible behaviours of *P*. In this paper we will make use of trace refinement checks:  $P \sqsubseteq_T Q$ .

The modelling and analysis of protocols in CSP, is supported by model checking tools, such as FDR [13]. FDR can automatically check whether a specification of a property (*P*) is satisfied by a proposed model (*Q*). If the result of a check is negative a counter example is given which provides information on the behaviour of the model which leads to the violation of the property.

## 4 Formal Analysis of an OFWA Protocol

### 4.1 A Buyer-Seller Watermarking Protocol

The Buyer-Seller Watermarking Protocol, proposed in [2], is an OFWA protocol aiming to satisfy the two primary asymmetric fingerprinting protocol requirements, namely traceability and proof of illegal distribution and framing resistance. The protocol does not attempt to provide anonymity. We use the three stage work flow illustrated in Figure 3 to perform our analysis.

We use the model checker FDR [13], which takes two processes, the protocol model and the requirement model, and verifies that the first refines the other. If the refinement check fails then the protocol does not satisfy its requirements and appropriate examples of attacks are automatically generated by FDR.

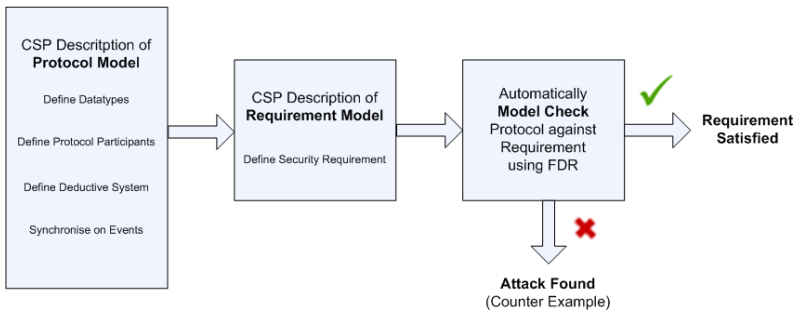


Fig. 3. Three Stage Work Flow

### 4.2 Protocol Model

Each protocol participant must have its own CSP process defined representing each event the agent is able to perform. Due to the restrictions on space we will not include the CSP descriptions of all the protocol participants. Instead we discuss the *BUYER* process as an appropriate example, as the buyer participates in every step of the protocol and thus every possible event is described. The processes representing the other two agents, the seller and watermarking authority, can be constructed similarly.

The buyer’s watermark generation process  $BUYER\_GEN(b, signed, known)$ , denotes the buyer,  $b$ , performing watermarking generation in terms of three events. Upon each successful run of this process the buyer collects an encrypted watermark signed by the chosen watermarking authority  $Sign_{sk_{wa}}(E_{pk_b}(wm))$  which is added to the buyer’s set of *signed* encrypted watermarks.

The buyer’s watermark insertion process  $BUYER\_INS(b, signed, known)$ , denotes the buyer,  $b$ , performing watermarking insertion in terms of four events. Upon each successful run of this process the buyer collects a piece of cover material embedded with their chosen watermark  $(x' \oplus wm)$  which is added to the buyer’s set of *known* watermarked material.

$$BUYER(b, signed, known) = \left( \begin{array}{l} \text{if } (signed = \emptyset) \text{ then} \\ \quad BUYER\_GEN(b, signed, known) \\ \text{else} \\ \quad BUYER\_GEN(b, signed, known) \\ \quad \square BUYER\_INS(b, signed, known) \\ \quad \square_{s \in sellers} \text{ share.b.s?k} \in known \rightarrow BUYER(b, known) \end{array} \right)$$

$$BUYER\_GEN(b, signed, known) = \left( \begin{array}{l} \square_{\substack{wm \in \text{watermarks} \\ wa \in \text{watermark\_authorities}}} \left( \begin{array}{l} comm.b.wa.Cert_{wa}(pk_b) \rightarrow \\ comm.wa.b.Sign_{sk_{wa}}(E_{pk_b}(wm)) \rightarrow \\ BUYER(b, signed \cup Sign_{sk_{wa}}(E_{pk_b}(wm)), known) \end{array} \right) \end{array} \right)$$

$$BUYER\_INS(b, signed, known) = \left( \begin{array}{l} \square_{\substack{s \in sellers \\ x \in \text{covermaterial} \\ sgn \in signed \\ v \in seller\_watermarks \\ wa \in watermark\_authorities}} \left( \begin{array}{l} comm.b.s.arg(x) \rightarrow \\ comm.b.s.Cert_{wa}(pk_b) \rightarrow \\ comm.b.s.sgn \rightarrow \\ comm.s.b.E_{pk_b}(x' \oplus wm) \rightarrow \\ BUYER(b, signed, known \cup \{x' \oplus wm\}) \end{array} \right) \end{array} \right)$$

where  $x' = x \oplus v$

**Fig. 4.** Protocol Model: *BUYER* Process

$$\begin{aligned}
 REQ\_MODEL(b, s, wa, x, wm) = & \\
 & \left( \begin{array}{l} SPEC_1(b, s, wa, x, wm) \\ \square SPEC_2(b, s, wa, x, wm) \\ \square share.b.s.(x' \oplus wm) \rightarrow ARB(b, s, wa, x, wm) \end{array} \right) \\
 \\
 SPEC_1(b, s, wa, x, wm) = & \\
 & sellerknows.evidence_1(b, s, wa, x, wm) \rightarrow SPEC_1(b, s, wa, x, wm) \\
 & \square share.b.s.(x' \oplus wm) \rightarrow ARB(b, s, wa, x, wm) \\
 \\
 SPEC_2(b, s, wa, x, wm) = & \\
 & sellerknows.evidence_2(b, s, wa, x, wm) \rightarrow SPEC_2(b, s, wa, x, wm) \\
 & \square share.b.s.(x' \oplus wm) \rightarrow ARB(b, s, wa, x, wm) \\
 \\
 ARB(b, s, wa, x, wm) = & \\
 & sellerknows.evidence_1(b, s, wa, x, wm) \rightarrow ARB(b, s, wa, x, wm) \\
 & \square sellerknows.evidence_2(b, s, wa, x, wm) \rightarrow ARB(b, s, wa, x, wm) \\
 & \square share.b.s.(x' \oplus wm) \rightarrow ARB(b, s, wa, x, wm) \\
 \\
 REQ\_MODEL = & \quad \parallel \quad REQ\_MODEL(b, s, wa, x, wm) \\
 & \begin{array}{l} b \in buyers \\ s \in sellers \\ wa \in watermark\_authorities \\ c \in covermaterial \\ wm \in watermarks \end{array}
 \end{aligned}$$

**Fig. 5.** Requirement Model: *REQ\_MODEL* Process

$$REQ\_MODEL \sqsubseteq PROTO\_MODEL \setminus \{ | comm | \}$$

**Fig. 6.** Refinement Check

We define the parameterised process  $BUYER(b, signed, known)$ , which gives the buyer,  $b$ , the choice of participating in watermark generation, watermark insertion, or illegally releasing pirated material on the *share* channel. If the buyer is not in possession of any *known* watermarked material then he must first participate in the buyer generation process  $BUYER\_GEN(\dots)$ . The *share* event models the real world event of a dishonest buyer releasing a pirated copy onto some file sharing network which the seller is monitoring. The full description of a buyer is given in Figure 4.

Minimal change is made to the deductive system constructed in [1] for our analysis of the Memon and Wong protocol. Our deductive system is a simplification of Roscoe's lazy spy model [14], used to analyse the Needham Schroeder (Lowe) Public Key protocol [15]. Our deductive system only allows the intelligent seller to act passively. That is, we construct an intelligent seller process  $INTELLIGENT\_SELLER$  that listens in on all messages sent over the *comm* and *share* channels to build up knowledge and makes deductions using this knowledge. The intelligent seller is defined using the following three events:-

- *learn* is the event that enables the intelligent seller to build up knowledge by listening in on messages sent over the *comm* and *share* channels,

- *infer* enables the intelligent seller to deduce further knowledge by making inferences using existing knowledge,
- *sellerknows* is the event we use to observe when the intelligent seller has collected sufficient knowledge to constitute evidence.

The seller's initial knowledge, denoted by the *initialknowledge* set, consists of all agents, each agent's public key, all cover material, the seller's secret key and the seller's watermarking key.

$$\begin{aligned} \text{initialknowledge} = & \text{agents} \cup \{pk_a \mid a \leftarrow \text{agents}\} \cup \\ & \text{covermaterial} \cup \{sk_{Sam}\} \cup \{wk_{Sam}\} \end{aligned}$$

Our specification is written in the form, certain evidence of illegal file sharing may be gathered by the seller if and only if the file has been illegally shared. The evidence set defines what knowledge, learnt and deduced within the deductive system, constitutes evidence. This enables us to observe, during analysis, when an intelligent seller has gathered evidence, having defined a *sellerknows* channel upon which gathered evidence can be released.

$$\begin{aligned} \text{Evidence} = & \{ \text{evidence}_1(b, s, wa, x, wm), \text{evidence}_2(b, s, wa, x, wm) \\ & \mid b \leftarrow \text{buyers}, s \leftarrow \text{sellors}, wa \leftarrow \text{watermark\_authorities}, \\ & x \leftarrow \text{covermaterial}, wm \leftarrow \text{watermarks} \} \end{aligned}$$

$$\begin{aligned} \text{evidence}_1(b, s, wa, x, wm) &= (x' \oplus wm) \\ \text{evidence}_2(b, s, wa, x, wm) &= \text{Sign}_{sk_{wa}}(E_{pk_b}(wm)) \end{aligned}$$

We construct our overall protocol model, by synchronising the protocol participant processes and the *INTELLIGENT\_SELLER* process, which can then be verified against the protocol requirements. Figure 7, illustrates the synchronisation forming the overall protocol model *PROTO\_MODEL*.

### 4.3 Requirement Model

We specify a single requirement which is a necessary test of the protocol's security. For the framing resistance property to be satisfied the seller should be in possession of the two pieces of evidence specified in the *evidence* set if and only if that piece of digital material has been illegally copied and shared. We describe this requirement, the second stage of our three stage work flow, by constructing the CSP process *REQ\_MODEL* as illustrated in Figure 5.

The parameterised process *REQ\_MODEL*(*b, s, wa, x, wm*) gives the choice of three possible behaviours. The processes *SPEC*<sub>1</sub> and *SPEC*<sub>2</sub> state that, if at first one piece of evidence is gathered, the alternative piece of evidence must not be gathered until the file has been illegally distributed, although the first piece of evidence may be collected over and over again. Otherwise, if the file is illegally distributed once along the *share* channel at any time, evidence gathering along with further illegal sharing of the same file may then happen arbitrarily as described by the process *ARB*. We use indexed parallel to generalise this requirement for all possible transactions.

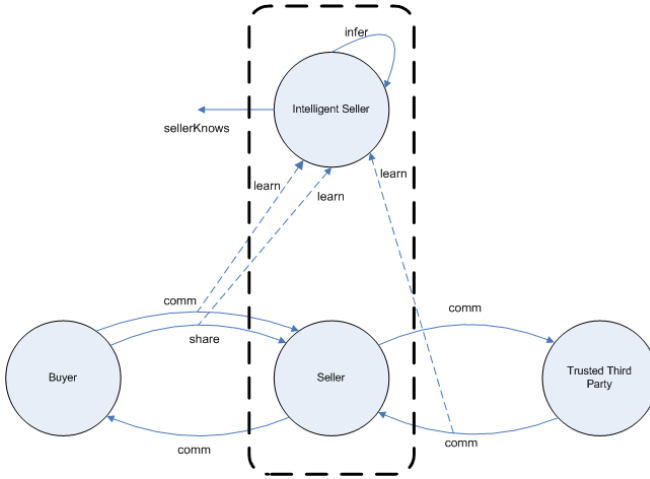


Fig. 7. Protocol Model: Synchronisation on Events

- $\alpha.1$   $Bob \rightarrow Tom : Cert_{Tom}(pk_{Bob})$
- $\alpha.2$   $Bob \rightarrow Tom : E_{pk_{Tom}}(Sign_{sk_{Bob}}(W_1))$
- $\alpha.3$   $Tom \rightarrow Bob : Sign_{sk_{Tom}}(E_{pk_{Bob}}(W_1))$
- $\alpha.4$   $Bob \rightarrow Sam : arg(X_1)$
- $\alpha.5$   $Bob \rightarrow Sam : Cert_{Tom}(pk_{Bob})$
- $\alpha.6$   $Bob \rightarrow Sam : Sign_{sk_{Tom}}(E_{pk_{Bob}}(W_1))$
- $\alpha.7$   $Sam \rightarrow Bob : E_{pk_{Bob}}(X'_1 \oplus W_1)$
- share*  $Bob \rightarrow Sam : (X'_1 \oplus W_1)$
- sellerknows*  $\rightarrow$   $: Sign_{sk_{Tom}}(E_{pk_{Bob}}(W_1))$
- sellerknows*  $\rightarrow$   $: (X'_2 \oplus W_1)$

Fig. 8. Counter Example of Unbinding Attack

#### 4.4 Analysis

Lei *et al.* identified an unbinding problem inherent in the Memon and Wong protocol. The flaw becomes apparent only once the seller has intercepted at least a single item of pirated material. In this scenario, the seller is able to extract the watermark from the cover material and embed it within a second piece of cover material so as to fabricate evidence of further piracy.

Using the model checker FDR we perform a refinement check of our protocol model against our specification process. This will search through every possible state of the interpreted state machine to check whether it is possible to reach some undesirable state. When we model check our protocol model of Memon and Wong we find that the requirement is not satisfied and a counter example generated, illustrated in Figure 8, matching the unbinding problem described by Lei *et al.*

## 5 Formal Analysis of an ONWA Protocol

### 5.1 A Privacy-Preserving Buyer-Seller Watermarking Protocol with Semi-trust Third Party

An ONWA protocol was proposed by Shao [4] to meet the three common requirements of asymmetric fingerprinting protocols. The roles of the watermarking authority (known as the Notary Authority in [4]) and the certification authority are separated so as to reduce the the level of trust required of the watermarking authority. The likelihood of a conspiracy attack made against the buyer is said to be reduced as more parties must now conspire together to frame the buyer.

### 5.2 Protocol Model

As in Section 4.2 we discuss a single process *SELLER*, omitting the processes representing the buyer and watermarking authority. The *SELLER* process involves every possible event performed in the protocol and the other two can be constructed similarly, consisting of only those events which they may perform.

The process *SELLER*(*s*), illustrated in Figure 10, enables the seller, *s*, to participate in the watermark generation/insertion process, *SELLER\_GEN\_INS*(*s*), or intercept illegally released pirated material on the *share* channel.

$\alpha.1$     *Bob*  $\rightarrow$  *Sam* :  $arg(X_1)$   
 $\alpha.2$     *Bob*  $\rightarrow$  *Sam* :  $Sign_{sk_{Bob}^*}(arg(X_1))$   
 $\alpha.3$     *Bob*  $\rightarrow$  *Sam* :  $E_{Tom}^{X_1}$   
 $\alpha.4$     *Sam*  $\rightarrow$  *Tom* :  $E_{Tom}^{X_1}$   
 $\alpha.5$     *Tom*  $\rightarrow$  *Sam* : *ew*  
 $\alpha.6$     *Tom*  $\rightarrow$  *Sam* :  $pk_{Bob}^*$   
 $\alpha.7$     *Tom*  $\rightarrow$  *Sam* :  $S_{Tom}^{X_1}$   
 $\alpha.8$     *Tom*  $\rightarrow$  *Sam* :  $E_{ARB}$   
 $\alpha.9$     *Sam*  $\rightarrow$  *Bob* :  $E_{pk_{Bob}}(X'_1 \oplus W_1)$   
 $\alpha.share$  *Bob*  $\rightarrow$  *Sam* :  $(X'_1 \oplus W_1)$   
 $\beta.1$     *Bob*  $\rightarrow$  *Sam* :  $arg(X_2)$   
 $\beta.2$     *Bob*  $\rightarrow$  *Sam* :  $Sign_{sk_{Bob}^*}(arg(X_2))$   
 $\beta.3$     *Bob*  $\rightarrow$  *Sam* :  $E_{Tom}^{X_2}$   
 $\beta.4$     *Sam*  $\rightarrow$  *Tom* :  $E_{Tom}^{X_2}$   
 $\beta.5$     *Tom*  $\rightarrow$  *Sam* : *ew*  
 $\beta.6$     *Tom*  $\rightarrow$  *Sam* :  $pk_{Bob}^*$   
 $\beta.7$     *Tom*  $\rightarrow$  *Sam* :  $S_{Tom}^{X_2}$   
 $\beta.8$     *Tom*  $\rightarrow$  *Sam* :  $E_{ARB}$   
*sellerknows*  $\rightarrow$         :  $Sign_{sk_{Tom}}(E_{pk_{Bob}}(W_1))$   
*sellerknows*  $\rightarrow$         :  $(X'_2 \oplus W_1)$

where  $E_a^{X_i} = E_{pk_a}(ew, pk_{Bob}^*, pf_{Bob}, E_{ARB}, h(arg(X_i)), Cert_{CA}(\overline{pk_{Bob}}), S_{Bob}^{X_i})$   
 $ew = E_{pk_{Bob}}(W_1)$   
 $S_a^{X_i} = Sign_{sk_a}(ew, pk_{Bob}^*, h(arg(X_i)))$

**Fig. 9.** Counter Example of Unbinding Attack

$$\begin{aligned}
\text{SELLER}(s) = & \\
& \square \left( \begin{array}{l} \text{SELLER\_GEN\_INS}(s, b, x, wm) \\ \square \\ \text{share.b.s.}(x' \oplus wm) \rightarrow \text{SELLER}(s) \end{array} \right) \\
& \begin{array}{l} b \in \text{buyers} \\ x \in \text{covermaterial} \\ wm \in \text{watermarks} \end{array} \\
\text{SELLER\_GEN\_INS}(s, b, x, wm) = & \\
& \square \left( \begin{array}{l} \text{comm.b.s.} \arg(x) \rightarrow \\ \text{comm.b.s.} \text{Sign}_{sk_b^*}(\arg(x)) \rightarrow \\ \text{comm.b.s.} E_{wa} \rightarrow \\ \text{comm.s.wa.} E_{wa} \rightarrow \\ \text{comm.wa.s.} ew \rightarrow \\ \text{comm.wa.s.} pk_b^* \rightarrow \\ \text{comm.wa.s.} S_{wa} \rightarrow \\ \text{comm.wa.s.} E_{arb} \rightarrow \\ \text{comm.s.b.} E_{pk_b^*}(x' \oplus wm) \rightarrow \\ \text{SELLER}(s) \end{array} \right) \\
& \begin{array}{l} wa \in \text{watermark\_authorities} \\ arb \in \text{arbitrators} \end{array}
\end{aligned}$$

where  $x' = x \oplus v$

$$\begin{aligned}
E_{wa} &= E_{pk_{wa}}(ew, pk_b^*, pf_b, E_{arb}, h(\arg(x)), \text{Cert}_{ca}(\overline{pk_b}), S_b) \\
E_{arb} &= E_{pk_{arb}}(ew, pk_b^*, pf_b, E_{arb}, h(\arg(x)), \text{Cert}_{ca}(\overline{pk_b}), S_b) \\
ew &= E_{pk_b}(wm) \\
S_b &= \text{Sign}_{sk_b}(ew, pk_b^*, h(\arg(x))) \\
S_{wa} &= \text{Sign}_{sk_{wa}}(ew, pk_b^*, h(\arg(x)))
\end{aligned}$$

**Fig. 10.** Seller Description in CSP

Our deductive system remains unchanged apart from the *initialknowledge* and *evidence* sets given below.

$$\begin{aligned}
\text{initialknowledge} = & \text{agents} \cup \{pk_a \mid a \leftarrow \text{agents}\} \cup \\
& \text{covermaterial} \cup \{sk_{Sam}\} \cup \{wk_{Sam}\}
\end{aligned}$$

$$\begin{aligned}
\text{Evidence} = & \{ \text{evidence}_1(b, s, wa, x, wm), \text{evidence}_2(b, s, wa, x, wm) \\
& \mid b \leftarrow \text{buyers}, s \leftarrow \text{sellors}, wa \leftarrow \text{watermark\_authorities}, \\
& x \leftarrow \text{covermaterial}, wm \leftarrow \text{watermarks} \}
\end{aligned}$$

$$\begin{aligned}
\text{evidence}_1(b, s, wa, x, wm) &= (x' \oplus wm) \\
\text{evidence}_2(b, s, wa, x, wm) &= \text{Sign}_{sk_{wa}}(ew, pk_b^*, h(\arg(x)))
\end{aligned}$$

We construct our overall protocol model *PROTO\_MODEL*, by synchronising the protocol participant processes and the *INTELLIGENT\_SELLER* process, which can then be verified against the protocol requirements.

### 5.3 Requirement Model

We use the same single requirement, in Figure 5, which is a necessary test of the protocol's security. The requirement differs only in what constitutes evidence, as given in the *Evidence* set defined in the deductive system above.

**Table 1.** CSP Sections Consistent Between Protocols

CSP Script Section	Unmodified
Cryptographic Primitives	✓
Protocol Participants	×
Message Formats	×
Deductive Rules	✓
Deductive System	✓
- Initial Knowledge	×
- Evidence Set	×
Composition	✓
Specification	✓

## 5.4 Analysis

When we model check our protocol model of Shao we find that the requirement is not satisfied. The flaw again occurs once the seller has intercepted at least a single item of pirated material. This is illustrated in Figure 9 by a complete run  $\alpha$  of the protocol, followed by an associated share event. The buyer may then choose to use the same values for each variable other than the cover material, i.e. the watermark and *one time* public/private key pair  $(pk_{Bob}^*, sk_{Bob}^*)$ , on a subsequent run of the protocol,  $\beta$ . The seller is now able to extract the watermark from the cover material and embed it within a second piece of cover material, purchased in a second run of the protocol  $\beta$ , so as to falsely gain evidence of further piracy.

As our previous work indicated [1], the consequence of the attack is that if a number of files are purchased, copied and illegally distributed in this manner, then the seller is only able to prove that at least one of the files has been copied and is not able to identify specifically which one(s).

## 6 Discussion

The CSP scripts, available at [www.cs.surrey.ac.uk/personal/pg/D.M/](http://www.cs.surrey.ac.uk/personal/pg/D.M/), used to model and analyse the protocols, are identical in their structure consisting of seven parts. This section identifies which parts of the scripts remain unchanged between protocol analyses, and which must be modified, as illustrated in Table 1. The reader is also referred to Roscoe's CSP script(s) of the Needham Schroeder (Lowe) protocol, released in conjunction with [14], which forms the basis of our work.

All the cryptographic primitives defined in Section 3 have been included in our CSP scripts, expressed as functions. The rules that govern such functions are described by the set of deductive rules. All cryptographic primitives and associated deductive rules remain unmodified in the analyses of protocols, although only a subset may be required. If the list of cryptographic primitives is found to be incomplete then a new primitive, and associated deductive rules, can be added to the current list.

Each protocol will have differing CSP descriptions of the protocol participants. Although these processes will need to be rewritten, each buyer-seller watermarking protocol will closely follow the typical message flows, illustrated in Figures 1 and 2, thus the protocol participants will be similar in form to those in our analyses of [2] and [4], respectively. The manner in which the synchronisations between the protocol participants and the deductive system are composed is identical between protocols.

The set of possible message formats are unique to each protocol but require little extra work to that of writing the protocol participants themselves as they match identically the form of the messages used by the protocol participants.

Our reduced deductive system aims only to be sufficient in identifying the unbinding attacks discussed in our analyses. The deductive system remains unchanged for the analyses of protocols with the exception of the *initialknowledge* and *Evidence* sets. It must, however, be generalised to provide a sufficient verification of all protocol requirements.

Our single requirement is similar for every protocol analysis differing only in what constitutes evidence, as defined in the *Evidence* set. However, the set of requirements must be expanded in order to provide a sufficient verification of all protocol requirements, although each protocol may only aim to satisfy a subset of these common requirements.

## 7 Conclusion and Further Work

We have demonstrated how the model constructed in [1] may be adapted for further analyses of various buyer-seller watermarking protocols. Having extended the Poh and Martin's framework [3], we analysed two Buyer-Seller Watermarking Protocols, using the process algebra CSP, to identify flaws in the OFWA and ONWA protocols, proposed by Memon and Wong [2] and Shao [4] respectively. Our analyses of the protocols reaffirmed the unbinding attack described by Lei *et al.* on the Memon protocol and identified a new attack found on the protocol by Shao. We then highlighted which sections of the script require modification and which sections can remain unchanged, for the analyses of additional protocols.

Currently our deductive system restricts the intelligent seller to passive behaviour, listening in on insecure communications. This system should be made more robust by allowing other protocol participants, as well as an external intruder (man-in-the-middle), to also act maliciously and by allowing more aggressive behaviour by the intruder in line with Roscoe's lazy spy model allowing all behaviour defined in the Dolev Yao model. The set of formally specified requirements must also be extended to provide both a necessary and sufficient test of each protocol's security.

Further research is required to understand how formal analysis techniques may be used to verify protocols conforming to the remaining asymmetric fingerprinting models. In particular, it would be appropriate to study how our model may be modified in order to analyse TKTP protocols which use a trusted computing platform, rather than a watermarking authority, as a centre of trust.

**Acknowledgement.** The authors would like to thank Steve Schneider for his technical discussions.

## References

1. Williams, D.M., Treharne, H., Ho, A.T.S., Culnane, C.: Using a formal analysis technique to identify an unbinding attack on a buyer-seller watermarking protocol
2. Memon, N., Wong, P.W.: A buyer seller watermarking protocol. *IEEE Transactions on Image Processing* 10(4), 643–649 (2001)
3. Poh, G.S., Martin, K.M.: A framework for design and analysis of asymmetric fingerprinting protocols. In: *Third International Symposium on Information Assurance and Security*, pp. 457–461 (2007)
4. Shao, M.H.: A privacy-preserving buyer-seller watermarking protocol with semi-trust third party
5. Lei, C.L., Yu, P.L., Tsai, P.L., Chan, M.H.: An efficient and anonymous buyer-seller watermarking protocol. *IEEE Transactions on Image Processing* 13(12), 1618–1626 (2004)
6. Qiao, L., Nahrstedt, K.: Watermarking schemes and protocols for protecting rightful ownership and customer’s rights. *Journal of Visual Communication and Image Representation* 9(3), 194–210 (1998)
7. Roscoe, A.W., Ryan, P., Schneider, S., Goldsmith, M., Lowe, G.: *The Modelling and Analysis of Security Protocols*. Addison-Wesley, Reading (2001)
8. Ibrahim, I.M., Nour El-Din, S.H., Hegazy, A.F.A.: An effective and secure buyer seller watermarking protocol. In: *Third International Symposium on Information Assurance and Security*, pp. 21–28 (2007)
9. Hoare, C.A.R.: *Communicating sequential processes*. Prentice-Hall International, Englewood Cliffs (1985)
10. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) *EUROCRYPT 1999*. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
11. Cox, I.J., Miller, L., Bloom, J.A.: *Digital Watermarking*. Morgan Kaufmann, San Francisco (2002)
12. Housley, R., Ford, W., Polk, W., Solo, D.: *Internet x.509 public key infrastructure certificate and crl profile* (1999)
13. Formal Systems. FDR 2.82. Formal Systems Ltd. (2005)
14. Roscoe, A.W.: *The Theory and Practice of Concurrency*. Prentice Hall, Englewood Cliffs (1998)
15. Lowe, G.: Breaking and fixing the needham-schroeder public-key protocol using fdr. In: *Proceedings of the Second International Workshop on Tools and Algorithms for Construction and Analysis of Systems*, pp. 147–166 (1996)