

Using a Formal Analysis Technique to Identify an Unbinding Attack on a Buyer-Seller Watermarking Protocol

David M. Williams^{*}
d.m.williams
@surrey.ac.uk

Helen Treharne
h.treharne@surrey.ac.uk

Anthony T. S. Ho
a.ho@surrey.ac.uk

Chris Culnane
c.culnane@surrey.ac.uk

Department of Computing
University of Surrey
Guildford, GU2 7XH

ABSTRACT

In this paper we provide a novel approach to the analysis of buyer-seller watermarking protocols by tailoring an existing formal technique that has not previously been used in this context. We accurately represent a buyer-seller watermarking protocol as proposed by Ibrahim *et al.* [6] by constructing a model using the process algebra CSP. By describing our model in this manner and utilising the tool support associated with CSP we are able to conduct a thorough analysis of all the possible behaviour in the protocol. Through formal analysis we have discovered an unbinding attack on the protocol. In this paper we also highlight other weaknesses that exist in the protocol and propose verifiable solutions to correct these weaknesses.

Categories and Subject Descriptors

I.6 [Computing Methodologies]: Simulation and Modelling; E.m [Data]: Miscellaneous—*Digital Watermarking*

General Terms

Security, Verification

Keywords

Buyer-Seller Watermarking Protocol, Formal Model, Formal Analysis, CSP, Unbinding, Customers' Rights

1. INTRODUCTION

Multimedia content is vulnerable to large scale copying and redistribution through easily accessible networks. Copy-

^{*}The author's work is sponsored by an EPSRC Thales CASE Award.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM&Sec'08, September 22–23, 2008, Oxford, United Kingdom.
Copyright 2008 ACM 978-1-60558-058-6/08/09 ...\$5.00.

right owners wish to deter such activity, detect it when it occurs and even trace the original perpetrator. Digital watermarking schemes aim to imperceptibly embed an identifying mark within the multimedia content itself [2]. The object of a robust watermarking scheme is to make it impossible to remove the mark without using the appropriate extraction algorithm and corresponding key.

Much attention has been paid to the protection of the rights of the seller during transactions involving copyrighted content [3]. Qiao and Nahrstedt [10] first identified the inadequacy of existing watermarking procedures when considering a customer's rights. Memon and Wong [9] proposed a solution to the customers' rights problem and subsequently many authors have identified other desirable properties and constructed protocols to satisfy them [1], [14]. Buyer-seller watermarking protocols are now expected to satisfy the following security properties: copy deterrence, customers' rights, unbinding, protocol practicality, conspiracy, buyer participation, and resilience to man-in-the-middle attack.

This paper demonstrates how a formal modelling technique can be used to identify any failure to meet the security requirements of such buyer-seller watermarking protocols. If all desired security requirements are satisfied by the protocol we say that the protocol is secure. This is the notion of security we shall use throughout this paper. The formal modelling technique we use is Communicating Sequential Processes (CSP) [5] which has a mathematical foundation and can be used to provide precise, compact and unambiguous high-level models of systems. CSP has been used to model many cryptographic and communication protocols and has helped find several successful attacks and ambiguities in published security protocols [12]. A common example is Lowe's Man-in-the-Middle attack on the Needham Schroeder Public Key protocol [8].

The first contribution of this paper is the formal model we have developed of the Ibrahim *et al.* protocol. Our second contribution is the formal analysis of this model and the unbinding attack on the protocol that our analysis uncovers. By assuming that the underlying watermarking is perfectly secure we are able to abstract away from the watermarking layer and analyse the protocol alone. If the protocol is found to be insecure the overall system is insecure.

We start by defining the CSP, cryptography and water-

marking notation, used throughout this paper, in Section 2. Section 3 gives the necessary background to buyer-seller watermarking protocols, discusses the protocol proposed by Ibrahim *et al.* and describes each security requirement the protocol aims to satisfy. We define our model in Section 4 and provide the formal analysis of the model in Section 5, including details of an unbinding attack as found by the model checker FDR [13]. A second unbinding attack and a relevant revision to the protocol are described in Section 6. Section 7 then discusses the contribution of our work and how it may be generalised. This final section also suggests ideas for related work and areas of interest for further reading.

2. NOTATION

2.1 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 all events is denoted by Σ . The set of events of a process P is denoted by αP .

Events may be compound in structure, consisting of a *channel name* and some (possibly 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 that can be associated with c to produce events.

For example, consider the channel name *comm* and $\text{agents} \times \text{agents} \times \text{messages}$ as its type, 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 along the *comm* channel. The events associated with *comm* will be of the form $\text{comm}.s.r.m$, where $s \in \text{agents}$, $r \in \text{agents}$, and $m \in \text{messages}$. For example, $\text{comm}.Bob.Sam.message_1$ is one such event.

The syntax of CSP provides several operators for modelling processes.

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

where a is an *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 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: $\bigsqcup_i P_i$ chooses a process from an i -indexed set of processes P . The process expression $S(p)$ expresses a recursive call. Finally, processes can be defined using (recursive) definitions of the form $S(p) \hat{=} P$.

Processes can be combined together using composition operators: parallel ‘ \parallel ’ and interleaving ‘ $\parallel\parallel$ ’. During the analysis phase, we also make use of the hiding operator ‘ \backslash ’. The process $P \parallel\parallel Q$ means that none of the events from P and Q

synchronise. When processes run in parallel they must synchronise on common events (otherwise the events can occur independently). For example, in the parallel process:

$$a \rightarrow b \rightarrow \text{Stop} \parallel b \rightarrow c \rightarrow \text{Stop}$$

a and c can occur independently of the other process, but the occurrence of b requires both processes to synchronise. We use the alphabetised parallel operator, and its indexed version, which explicitly states the alphabets of the processes, i.e., in $P \parallel [\alpha P \parallel \alpha Q] Q$. The intersection of αP and αQ is the set of events that requires synchronisation. In this paper we omit the alphabets for clarity of presentation.

There are three semantic models in CSP which enable us to describe process behaviour and in this paper we use the simplest model, i.e. the *traces* model. Full details of the models can be found in [11]. The traces model captures the traces of events which a CSP process might exhibit. A sequence tr is a *trace* of a process P if there is some execution of P in which exactly that sequence of events is performed. The empty trace, containing no events, is written $\langle \rangle$. More generally, a trace may be written as a sequence of events $\langle e_1, e_2, \dots, e_n \rangle$.

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 of systems in CSP and their formal verification is supported by model checking tools, such as FDR. 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. In this paper we will encode desired properties of the buyer-seller watermarking protocol as CSP processes (and will be used as P above). For any given model there can be a number of properties which need to be preserved. CSP theory enables us to deduce that if the model satisfies each property individually then all properties are satisfied. It is important to be able to split up the verification into smaller parts because it is impossible to model check unboundedly large systems all at once in FDR.

2.2 Cryptography

Throughout the paper we make use of the following notation regarding cryptographic primitives. We use $\{m\}_{PK(a)}$ to denote a message m encrypted under the public key of agent a . The same agent possesses a corresponding secret decryption key which is used to sign the message m . We denote this signed message using $\{m\}_{SK(a)}$. More commonly the decryption key will be used to decipher an encrypted message. It is also important to note that any agent can recover the original message from a signed message using the corresponding encryption key. We define this equivalence as follows: $\{\{m\}_{PK(a)}\}_{SK(a)} \equiv \{\{m\}_{SK(a)}\}_{PK(a)} \equiv m$.

A specific occurrence of a message signed using an agent’s secret key is the digital certificate $\{a, PK(a)\}_{SK(t)}$, constructed of the agent’s identity and their public key as signed by the trusted certification authority. Digital certificates are sent between agents to indicate their public key and are trusted by the receiver as the signature authenticates its source as a trusted origin.

Step.1	$b \rightarrow s : \text{arg}(c)$
Step.2	$s \rightarrow b : \{s, PK(s)\}_{SK(t)}$
Step.3	$b \rightarrow s : \{wm\}_{PK(b)}$
Step.4	$b \rightarrow s : \{\{wm\}_{SK(b)}\}_{PK(t)}$
Step.5	$b \rightarrow s : \{b, PK(b)\}_{SK(t)}$
Step.6	$b \rightarrow s : \{H(\text{arg}(c))\}_{SK(b)}$
Step.7	$b \rightarrow s : \{H(wm, \text{arg}(c))\}_{SK(b)}$
Step.8	$s \rightarrow t : \{\{wm\}_{SK(b)}\}_{PK(t)}$
Step.9	$s \rightarrow t : \{b, PK(b)\}_{SK(t)}$
Step.10	$t \rightarrow s : \{\{wm\}_{PK(b)}\}_{SK(t)}$
Step.11	$s \rightarrow b : \{\{[c \oplus wm]_{WK(s)}\}_{PK(b)}\}_{SK(s)}$

Figure 1: Steps in Ibrahim *et al.* protocol

It is useful to have some way of denoting hash functions. Let $H(m)$ define a hash of the message m such that any agent in possession of the message m is able to produce the hash $H(m)$. The opposite does not hold true i.e., no agent is able to deduce m from $H(m)$. We denote $\text{arg}(c)$ to be the unique identifier of the cover material, c . Consider it to be an image title, track name or, more generally, a purchase order. Note that anyone in possession of the file knows its title but knowing the title does not necessarily mean possession of the file.

2.3 Watermarking

In order to include watermarking in this cryptography notation we use $[c \oplus wm]_{WK(s)}$ to denote the watermark wm embedded in the cover material, c , using the watermarking key of some seller $WK(s)$. We need not concern ourselves with the particular watermarking embedding algorithm. Our analysis can be conducted independently of it.

3. BUYER-SELLER WATERMARKING PROTOCOLS

Qiao and Nahrstedt [10] first identified the inadequacy of watermarking procedures when considering customers' rights. Memon and Wong [9] and subsequent work [1] [14] proposed solutions to the customers' rights problem along with a growing set of other desirable security requirements.

Ibrahim *et al.* [6] proposed a single protocol to address a number of issues, namely copy deterrence, customers' rights, unbinding, protocol practice applicability, conspiracy, buyer participation, and man in the middle. Claims were made of the security of the protocol without any formal verification. Three of the properties highlighted in the paper were trivially satisfied by design. The remaining four properties are summarised below. In Section 5, we will show the protocol fails to preserve the unbinding property. Figure 1 is a step by step representation of the protocol.

- **Copy Deterrence**
An honest seller must be able to trace piracy to the dishonest buyer at the source of a leak.
- **Customers' Rights**
An honest buyer requires assurance that a dishonest seller cannot fabricate evidence of piracy by the buyer.
- **Man-In-The-Middle**
All honest agents involved in the protocol require that

no dishonest intruder is able to discover the cover material, the buyer or seller's watermark, watermarked content or any watermarking and/or secret cryptographic keys used, even if the intruder has full control over the communication channels used as described.

- **Unbinding**

The unbinding problem in [7] describes the unique issue that arises when a seller has intercepted a single piece of cover material pirated by a semi-dishonest buyer. The seller is thus able to extract the buyer's watermark and embed this within a number of other pieces of cover material. This alone must not constitute evidence of further piracy, a mechanism must be put in place to *bind* the watermark to a specific piece of cover material.

Ibrahim *et al.* [6] assume that a Trusted Certification Authority (CA) exists as a single trust pivot and that each agent has received a digital certificate from the CA prior to participation in the protocol. This digital certificate is made up of the public key of the agent along with the agent's name all signed under the secret key of the CA. It is not necessary for the CA to keep a record of all digital certificates as they are returned to the CA in the protocol itself.

A buyer, b , initiates the protocol by sending a purchase order, corresponding to a unique piece of cover material, to a seller, s . This purchase order is defined as the one way function $\text{arg}(c)$. Anyone in possession of the cover material knows the associated $\text{arg}(c)$ but the cover material cannot be deduced when only the title is known. The seller proceeds to send his own digital signature $\{s, PK(s)\}_{SK(t)}$ to the buyer indicating receipt of the purchase order. The buyer must store this signature so that they may verify the final message as authentic.

The buyer then sends the seller five further messages:-

1. The seller receives the buyer's watermark encrypted with the buyer's public key $\{wm\}_{PK(b)}$. This will be stored as it will be required in order to embed the watermark wm into the cover material, c ;
2. The watermark signed using the buyer's watermarking key and then encrypted under the public key of the CA $\{\{wm\}_{SK(b)}\}_{PK(t)}$. This will be used by the judge during dispute resolution, after collaboration with the CA, to deduce the buyer's watermark without it being necessary for the buyer to reveal it themselves;
3. The buyer's digital certificate $\{b, PK(b)\}_{SK(t)}$. The seller will use this to verify authentic origin of messages signed using the buyer's secret key and also encrypt messages using the buyer's public key ready for secure transmission to the buyer;
4. The signed hash of the purchase order $\{H(\text{arg}(c))\}_{SK(b)}$ is said to provide evidence that the buyer made the purchase order;
5. A signed hash of the dual of the buyer's watermark and the argument $\{H(wm, \text{arg}(c))\}_{SK(b)}$ binding some watermark wm to a particular purchase order $\text{arg}(c)$.

The seller then proceeds to forward two of the messages to the trusted CA, t . These are the buyer's digital certificate $\{b, PK(b)\}_{SK(t)}$ and the buyer's signed watermark encrypted under the public key of the CA $\{\{wm\}_{SK(b)}\}_{PK(t)}$.

The CA has sufficient knowledge in order to strip off the levels of cryptography, thus deducing the original watermark wm . The CA then encrypts the watermark using the public key of the buyer and signs it with his own private key i.e., $\{\{wm\}_{PK(b)}\}_{SK(t)}$. Upon receiving this signed message the seller is able to verify that the buyer's watermark encrypted with the buyer's public key $\{wm\}_{PK(b)}$ received in step 3 matches the signed correspondence just received in step 10.

The seller is now ready to embed the watermark into the cover material in the encrypted domain thus creating $\{[c \oplus wm]_{WK(s)}\}_{PK(b)}$. Ibrahim *et al.* also discuss the embedding of a second watermark used to index the file. This enables 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. Our original CSP script included the seller's watermark. However, we have not included this additional embedding in the models within this paper for the sake of simplicity as the analysis finds the same attack regardless of whether this step is included or excluded from the model.

Finally, in step 11, the seller signs the encrypted watermarked document and sends this message, $\{\{[c \oplus wm]_{WK(s)}\}_{PK(b)}\}_{SK(s)}$, to the buyer. The buyer uses the seller's public key $PK(s)$ deduced from his digital certificate $\{s, PK(s)\}_{SK(t)}$ to strip off the seller's signature and verify it did indeed originate from the seller and then uses their own secret key $SK(b)$ to decipher message into the watermarked document, $[c \oplus wm]_{WK(s)}$. Figure 1 shows each step of the protocol.

The watermark embedding is conducted in the encrypted domain using the privacy homomorphism property resulting in equivalence 3. This is how the seller is able to conduct the embedding without ever knowing the watermark.

$$\{[c \oplus wm]_{WK(s)}\}_{PK(b)} \equiv \{\{c\}_{PK(b)} \oplus \{wm\}_{PK(b)}\}_{WK(s)}$$

The construction of the dual signature of the watermark and purchase order aims to prohibit the seller from transplanting a watermark from one piece of cover material into another provided both are purchased by the same buyer which assumes the protocol will be run multiple times for multiple transactions. It also assumes that more than a single buyer will use the protocol to purchase digital content. The protocol must remain secure for more than just a single run. Our analysis in Section 5 shows that an attack can be made when running two consecutive runs of the protocol.

4. THE MODEL

In order to build a formal model of the protocol behaviour we must have defined datatypes and all the agents of the protocol. Each agent modelled can be described as a separate CSP process. As will be shown later, we also use the notion of an inference engine which is a simplification of Roscoe's deductive system [11] that enables a fact to be deduced from a set of facts. Figure 2 illustrates how a buyer-seller watermarking protocol is modelled as above, and how we subsequently reason about the model.

4.1 Defining Data Types

The set $messages$ is the set of all messages that pass between agents. We note from Figure 1 that there are different message formats in the 11 steps of the protocol. Some steps use the same message format, for example, steps 2, 5 and 9. Each message format, $message_i$, is defined separately and

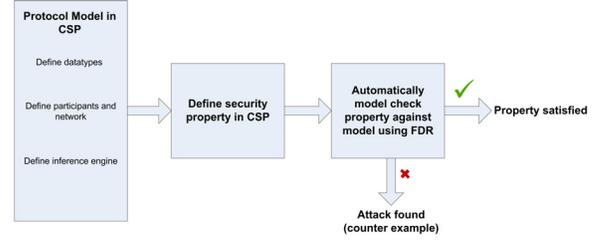


Figure 2: Adopted Workflow

$$\begin{aligned}
 message_1 &= \{arg(c) & | c \leftarrow covermaterial\} \\
 message_2 &= \{\{a, PK(a)\}_{SK(t)} & | a \leftarrow agents, \\
 & & t \leftarrow thirdpartys\} \\
 message_3 &= \{\{wm\}_{PK(b)} & | b \leftarrow buyers, \\
 & & wm \leftarrow watermarks\} \\
 message_4 &= \{\{\{wm\}_{SK(b)}\}_{PK(t)} & | t \leftarrow thirdpartys, \\
 & & b \leftarrow buyers, \\
 & & wm \leftarrow watermarks\} \\
 message_5 &= \{\{H(arg(c))\}_{SK(b)} & | c \leftarrow covermaterial, \\
 & & b \leftarrow buyers\} \\
 message_6 &= \{\{H(wm, arg(c))\}_{SK(b)} & | b \leftarrow buyers, \\
 & & wm \leftarrow watermarks, \\
 & & c \leftarrow covermaterial\} \\
 message_7 &= \{\{\{wm\}_{PK(b)}\}_{SK(t)} & | b \leftarrow buyers, \\
 & & t \leftarrow thirdpartys, \\
 & & wm \leftarrow watermarks\} \\
 message_8 &= \{\{\{[c \oplus wm]_{WK(s)}\}_{PK(b)}\}_{SK(s)} & | b \leftarrow buyers, \\
 & & s \leftarrow sellers, \\
 & & c \leftarrow covermaterial, \\
 & & wm \leftarrow watermarks\} \\
 message_9 &= \{[c \oplus wm]_{WK(s)} & | s \leftarrow sellers, \\
 & & c \leftarrow covermaterial, \\
 & & wm \leftarrow watermarks\} \\
 messages &= \bigcup_{i \in 1..9} message_i
 \end{aligned}$$

Figure 3: Message types

these combined to form the set $messages$ as shown in Figure 3. Messages 1 to 8 are the message formats for use in the Ibrahim *et al.* watermark generation/insertion protocol. While, $message_9$ denotes the unencrypted watermarked document that a buyer may subsequently illegally distribute over some file sharing network.

4.2 Protocol Agents

Each agent involved in a protocol run both sends and receives messages. We define a parameterised process, $HONEST_BUYER(b, \dots)$, denoting the buyer b in terms of eight events representing the messages sent and received. These eight events match the first seven steps along with step 11 in Figure 1, i.e., all the steps of the protocol in which the buyer participates. For example, $\{H(wm, arg(c))\}_{SK(b)}$ symbolises b sending to seller s the dual signature of the watermark wm and the purchase order $arg(c)$. The overall definition of the buyer depends on whether the buyer chooses to complete an honest run of the protocol or share previously purchased watermarked content. The share event, $share.b.s.k$, where k is some watermarked document, signifies b releasing the watermarked document onto some open file sharing network in the knowledge that sellers may be

monitoring. The watermark document k is drawn from the set $known$, which is the set of all watermarked content previously purchased by b . The full description of a buyer is given in Figure 4.

$$BUYER(b, known) = \square \left(\begin{array}{l} HONEST_BUYER(b, s, t, c, wm, known) \\ \square \\ share.b.s?k \in known \rightarrow BUYER(b, known) \end{array} \right)$$

$s \in sellers$
 $t \in thirdpartys$
 $c \in covermaterial$
 $wm \in watermarks$

$$HONEST_BUYER(b, s, t, c, wm, known) = \left(\begin{array}{l} comm.b.s.arg(c) \rightarrow \\ comm.s.b.\{s, PK(s)\}_{SK(t)} \rightarrow \\ comm.b.s.\{wm\}_{PK(b)} \rightarrow \\ comm.b.s.\{\{wm\}_{SK(b)}\}_{PK(t)} \rightarrow \\ comm.b.s.\{b, PK(b)\}_{SK(t)} \rightarrow \\ comm.b.s.\{H(arg(c))\}_{SK(b)} \rightarrow \\ comm.b.s.\{H(wm, arg(c))\}_{SK(b)} \rightarrow \\ comm.s.b.\{\{c \oplus wm\}_{WK(s)}\}_{PK(b)}_{SK(s)} \rightarrow \\ BUYER(b, known \cup \{c \oplus wm\}_{WK(s)}) \end{array} \right)$$

Figure 4: Buyer Description in CSP

We define a parameterised process, $HONEST_SELLER(s, \dots)$, denoting the seller s in terms of eleven events. This process describes every message exchanged in the protocol because the seller participates in every step. The overall definition of the seller is similar to the buyer. The seller is also willing to participate in an honest run of the protocol or listen in on the share channel so that they may retrieve any illegally distributed watermarked content. The full description of a seller is given in Figure 5.

$$SELLER(s) = \square \left(\begin{array}{l} HONEST_SELLER(s, b, t, c, wm) \\ \square \\ share.b.s.[c \oplus wm]_{WK(s)} \rightarrow SELLER(s) \end{array} \right)$$

$b \in buyers$
 $t \in thirdpartys$
 $c \in covermaterial$
 $wm \in watermarks$

$$HONEST_SELLER(s, b, t, c, wm) = \left(\begin{array}{l} comm.b.s.arg(c) \rightarrow \\ comm.s.b.\{s, PK(s)\}_{SK(t)} \rightarrow \\ comm.b.s.\{wm\}_{PK(b)} \rightarrow \\ comm.b.s.\{\{wm\}_{SK(b)}\}_{PK(t)} \rightarrow \\ comm.b.s.\{b, PK(b)\}_{SK(t)} \rightarrow \\ comm.b.s.\{H(arg(c))\}_{SK(b)} \rightarrow \\ comm.b.s.\{H(wm, arg(c))\}_{SK(b)} \rightarrow \\ comm.s.t.\{\{wm\}_{SK(b)}\}_{PK(t)} \rightarrow \\ comm.s.t.\{b, PK(b)\}_{SK(t)} \rightarrow \\ comm.t.s.\{\{wm\}_{PK(b)}\}_{SK(t)} \rightarrow \\ comm.s.b.\{\{c \oplus wm\}_{WK(s)}\}_{PK(b)}_{SK(s)} \rightarrow \\ SELLER(s) \end{array} \right)$$

Figure 5: Seller Description in CSP

We also construct a process of all the events in which some trusted CA t participates. We model steps 8, 9 and 10 of the protocol, as the process $TRUSTED_THIRD_PARTY(t)$, describing the interactions with any seller regarding any buyer's purchase.

$$TRUSTED_THIRD_PARTY(t) = \square \left(\begin{array}{l} comm.s.t.\{\{wm\}_{SK(b)}\}_{PK(t)} \rightarrow \\ comm.s.t.\{b, PK(b)\}_{SK(t)} \rightarrow \\ comm.t.s.\{\{wm\}_{PK(b)}\}_{SK(t)} \rightarrow \\ TRUSTED_THIRD_PARTY(t) \end{array} \right)$$

$b \in buyers$
 $s \in sellers$
 $wm \in watermarks$

Figure 6: Trusted Third Party Description in CSP

4.3 Overall Network

The process $NETWORK$ is a composition of the $BUYER$, $SELLER$ and $TRUSTED_THIRD_PARTY$ processes that synchronise on each of the events they may perform. For example, each time $BUYER(b)$ performs $comm.b.s.arg(c)$, $SELLER(s)$ must also perform this same event. Each process is initiated with concrete parameters, $BUYER$ with Bob and the empty set \emptyset , $SELLER$ with Sam and the $TRUSTED_THIRD_PARTY$ with Tom .

The $NETWORK$ process is a complete description of a simple system of transactions in which just a single buyer, Bob , may purchase watermarked content from a single seller whom communicates with a single CA, Tom . Multiple purchases may be made consecutively but not simultaneously. The buyer, Bob , initially owns no watermarked content thus the known parameter is equivalent to the empty set \emptyset . Figure 7 illustrates our $NETWORK$ process. In general we would construct a network made up of several buyers and sellers: but the $NETWORK$ process we have defined is sufficient in finding an attack.

$$NETWORK = \begin{array}{l} BUYER(Bob, \emptyset) \\ \parallel SELLER(Sam) \\ \parallel TRUSTED_THIRD_PARTY(Tom) \end{array}$$

Figure 7: Overall Network Description in CSP

4.4 Inference Engine

In order to analyse the model of the network we need to augment our model with additional observable events. We must be able to track these events so that evidence may be gathered regarding transactions. We want to define an intelligent seller that is able to build up knowledge by making deductions based on all messages that the seller receives over any channel. We model this as a separate process that listens in on incoming messages sent to the seller along the comm and share channels. The $learn$ event enables the intelligent seller to listen in on communications sent to the buyer allowing him to build up his knowledge. The intelligent seller is defined using the following three events:-

- $learn$ is the event that enables the intelligent seller to expand his knowledge by learning the messages that have been sent to the seller
- $infer$ models the deductive behaviour of the intelligent seller who is able to build up further knowledge by making inferences using existing knowledge
- $sellerknows$ enables us to observe when the knowledge built up by the intelligent seller constitutes evidence

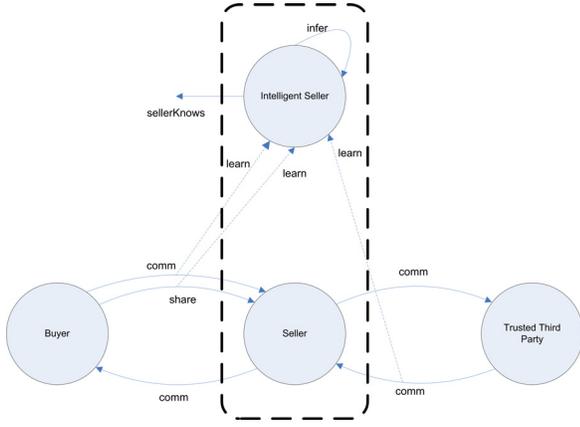


Figure 8: Overview of CSP process structure

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

$$\text{initialknowledge} = \text{agents} \cup \{PK(a) \mid a \leftarrow \text{agents}\} \cup \{SK(\text{Sam}), WK(\text{Sam})\} \cup \text{covermaterial}$$

The intelligent seller's knowledge extends as new messages are received and inferences are made. Knowledge built up only constitutes evidence when it is also an element of the evidence set defined below. This is the set of all possible embedded documents and dual signatures binding watermarks to purchase orders. Evidence is released along the *sellerknows* channel so that we may observe when this knowledge has been deduced for use in the analysis.

$$\text{evidence} = \{[c \oplus wm]_{WK(s)}, \{H(wm, arg(c))\}_{SK(b)} \mid b \leftarrow \text{buyers}, c \leftarrow \text{covermaterial}, wm \leftarrow \text{watermarks}\}$$

We then define the overall system as the *NETWORK* run in parallel with the *INTELLIGENT_SELLER*. This overall system is illustrated by Figure 8, which identifies the events involved when running the processes concurrently.

$$\text{SYSTEM} = \text{NETWORK} \parallel \text{INTELLIGENT_SELLER}$$

5. ANALYSIS

Having constructed our model in machine readable CSP we use the FDR model checker to test whether assertions made about the model hold true. In order to prosecute a dishonest buyer the seller must gather two pieces of evidence. The buyer must have the signed hash of the dual $\{H(wm, arg(c))\}_{SK(b)}$ that binds the watermark to a specific piece of cover material and also the corresponding watermarked document $[c \oplus wm]_{WK(s)}$. To protect the customers' rights, the same argument must hold in the opposite direction. That is, a dishonest seller must not be in possession of both pieces of evidence if an honest buyer has not illegally distributed the material. We analyse our model against this single requirement, which is a necessary test of the customers' rights and resistance to unbinding properties,

$$\text{SPEC}(b, s, c, wm) = \left(\begin{array}{l} \text{sellerknows}.\{H(wm, arg(c))\}_{SK(b)} \rightarrow \\ \text{SPEC}_1(b, s, c, wm) \\ \square \\ \text{sellerknows}.[c \oplus wm]_{WK(s)} \rightarrow \\ \text{SPEC}_2(b, s, c, wm) \\ \square \\ \text{share}.\text{b.s}.[c \oplus wm]_{WK(s)} \rightarrow \\ \text{ARB}(b, s, c, wm) \end{array} \right)$$

$$\text{SPEC}_1(b, s, c, wm) = \text{sellerknows}.\{H(wm, arg(c))\}_{SK(b)} \rightarrow \text{SPEC}_1(b, s, c, wm) \square \text{share}.\text{b.s}.[c \oplus wm]_{WK(s)} \rightarrow \text{ARB}(b, s, c, wm)$$

$$\text{SPEC}_2(b, s, c, wm) = \text{sellerknows}.[c \oplus wm]_{WK(s)} \rightarrow \text{SPEC}_2(b, s, c, wm) \square \text{share}.\text{b.s}.[c \oplus wm]_{WK(s)} \rightarrow \text{ARB}(b, s, c, wm)$$

$$\text{ARB}(b, s, c, wm) = \text{sellerknows}.\{H(wm, arg(c))\}_{SK(b)} \rightarrow \text{ARB}(b, s, c, wm) \square \text{sellerknows}.[c \oplus wm]_{WK(s)} \rightarrow \text{ARB}(b, s, c, wm) \square \text{share}.\text{b.s}.[c \oplus wm]_{WK(s)} \rightarrow \text{ARB}(b, s, c, wm)$$

Figure 9: Specification of desired property

but not sufficient for a thorough verification of the protocol's security. We do not consider the other security properties in this paper, for example, security against a man-in-the-middle attack.

Our analysis shows that the protocol by Ibrahim *et al.* fails to satisfy this requirement, thus we say that the protocol is insecure. The requirement fails to be satisfied due to the unbinding attack, given in Figure 10, such that the customers' rights are preserved as described in Section 3.

We first describe this requirement as a CSP specification, *SPEC*(*b, s, c, wm*), given in Figure 9. It is broken down into three separate processes. *SPEC*₁ says that if the file has been illegally distributed once along the *share* channel, evidence gathering along with further illegal sharing of the same file may then happen arbitrarily as described by the process *ARB*. Otherwise, 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. These behaviours are described by *SPEC*₂ and *SPEC*₃ collectively.

We then use trace refinement and let FDR automatically check whether our protocol model *SYSTEM* refines our specification *SPEC*. Therefore, we specify that the following assertion must hold for any buyer and any seller, who participate in the various consecutive runs of the protocol, and for any watermark and piece of cover material, chosen by

$\alpha.1 \quad Bob \rightarrow Sam : arg(C1)$
 $\alpha.2 \quad Sam \rightarrow Bob : \{Sam, PK(Sam)\}_{SK(Tom)}$
 $\alpha.3 \quad Bob \rightarrow Sam : \{\{WM\}_{SK(Bob)}\}_{PK(Tom)}$
 $\alpha.4 \quad Bob \rightarrow Sam : \{WM\}_{PK(Bob)}$
 $\alpha.5 \quad Bob \rightarrow Sam : \{Bob, PK(Bob)\}_{SK(Tom)}$
 $\alpha.6 \quad Bob \rightarrow Sam : \{H(arg(C1))\}_{SK(Bob)}$
 $\alpha.7 \quad Bob \rightarrow Sam : \{H(WM, arg(C1))\}_{SK(Bob)}$
 $\alpha.8 \quad Sam \rightarrow Tom : \{\{WM\}_{SK(Bob)}\}_{PK(Tom)}$
 $\alpha.9 \quad Sam \rightarrow Tom : \{Bob, PK(Bob)\}_{SK(Tom)}$
 $\alpha.10 \quad Tom \rightarrow Sam : \{\{WM\}_{PK(Bob)}\}_{SK(Tom)}$
 $\alpha.11 \quad Sam \rightarrow Bob : \{\{[C1 \oplus WM]_{WK(Sam)}\}_{PK(Bob)}\}_{SK(Sam)}$

$\alpha.share \quad Bob \rightarrow Sam : [C1 \oplus WM]_{WK(Sam)}$

$\beta.1 \quad Bob \rightarrow Sam : arg(C2)$
 $\beta.2 \quad Sam \rightarrow Bob : \{Sam, PK(Sam)\}_{SK(Tom)}$
 $\beta.3 \quad Bob \rightarrow Sam : \{\{WM\}_{SK(Bob)}\}_{PK(Tom)}$
 $\beta.4 \quad Bob \rightarrow Sam : \{\{WM\}_{SK(Bob)}\}_{PK(Tom)}$
 $\beta.5 \quad Bob \rightarrow Sam : \{Bob, PK(Bob)\}_{SK(Tom)}$
 $\beta.6 \quad Bob \rightarrow Sam : \{H(arg(C2))\}_{SK(Bob)}$
 $\beta.7 \quad Bob \rightarrow Sam : \{H(WM, arg(C2))\}_{SK(Bob)}$

Figure 10: Counter Example of Unbinding Attack

the buyers; achieved below using indexed parallel.

$$\parallel_{\substack{b \in buyers \\ s \in sellers \\ c \in covermaterial \\ wm \in watermarks}} SPEC(b, s, c, wm) \sqsubseteq SYSTEM \setminus \{comm\}$$

If the statement is found to be true the model is shown to be resilient to unbinding. Our analysis reveals that the protocol does not satisfy the property. An appropriate counter example is provided in Figure 10.

The counter example, involving two consecutive runs, illustrates one such instance where the security requirement has been breached. An initial run α of the protocol is completed between the buyer, *Bob*, and the seller, *Sam*, in which the buyer purchases the cover material, *C1*, embedded with their chosen watermark, *WM*. The buyer then proceeds to illegally distribute the file in step $\alpha.share$. Subsequently, a second protocol β runs through to step 7 between the same buyer, *Bob*, and seller, *Sam*, with *Bob* choosing to use the same watermark but requesting new content, *C2*.

Following each of the steps of the attack in Figure 10 *Sam* is able to gather the evidence required to prove that *Bob* has illegally redistributed a watermarked document without *Bob* having ever shared this file. At step 7 *Sam* receives $\{H(WM, arg(C2))\}_{SK(Bob)}$. By step 7 *Sam* is also able to construct $[C2 \oplus WM]_{WK(Sam)}$ like so: once the first piece of content has been illegally distributed by *Bob*, *Sam* has in his possession $[C1 \oplus WM]_{WK(Sam)}$ and $WK(Sam)$. Hence, *Sam* is able to extract the watermark *WM* (using one of the deduction rules given in the Appendix). *Sam* also knows the cover material *C2* and can therefore infer $[C2 \oplus WM]_{WK(Sam)}$ (using the same deduction rule). By deducing these two pieces of evidence, *Sam* is able to prove that *Bob* has illegally redistributed $[C2 \oplus WM]_{WK(Sam)}$ without *Bob* having ever shared this watermarked document.

The impact of the attack can be explained as follows: the buyer *Bob* purchases 1,000 files from the seller *Sam* each one embedded with the same watermark. *Bob* then illegally distributes multiple copies of all 1,000 files. *Sam* may subsequently retrieve any number of these files. *Sam* is only able

$\alpha.1 \quad Bob \rightarrow Sam : arg(C1)$
 $\alpha.2 \quad Sam \rightarrow Bob : \{Sam, PK(Sam)\}_{SK(Tom)}$
 $\alpha.3 \quad Bob \rightarrow Sam : \{\{WM1\}_{SK(Bob)}\}_{PK(Tom)}$
 $\alpha.4 \quad Bob \rightarrow Sam : \{WM1\}_{PK(Bob)}$
 $\alpha.5 \quad Bob \rightarrow Sam : \{Bob, PK(Bob)\}_{SK(Tom)}$
 $\alpha.6 \quad Bob \rightarrow Sam : \{H(arg(C1))\}_{SK(Bob)}$
 $\alpha.7 \quad Bob \rightarrow Sam : \{H(WM2, arg(C2))\}_{SK(Bob)}$
 $\alpha.8 \quad Sam \rightarrow Tom : \{\{WM1\}_{SK(Bob)}\}_{PK(Tom)}$
 $\alpha.9 \quad Sam \rightarrow Tom : \{Bob, PK(Bob)\}_{SK(Tom)}$
 $\alpha.10 \quad Tom \rightarrow Sam : \{\{WM1\}_{PK(Bob)}\}_{SK(Tom)}$
 $\alpha.11 \quad Sam \rightarrow Bob : \{\{[C1 \oplus WM1]_{WK(Sam)}\}_{PK(Bob)}\}_{SK(Sam)}$
 $\alpha.share \quad Bob \rightarrow Sam : [C1 \oplus WM1]_{WK(Sam)}$

Figure 11: Alternative Unbinding Attack

to prove that at least one of the 1,000 files has been illegally redistributed. *Sam* is unable to prove how many have been copied or identify which particular files. This is not sufficient evidence to prosecute *Bob* even though he was indeed acting maliciously. It is therefore in *Bob*'s interest to choose the same watermark in each run of the protocol. To protect the seller from such an attack the protocol must not allow a buyer to use the same watermark twice.

6. ALTERNATIVE UNBINDING ATTACK

The protocol is vulnerable to another attack. In this section we describe the attack and propose a modification to the protocol which solves the problem.

A dishonest buyer can send false data in step 7 of the protocol illustrated in Figure 11. Here the values for the watermark and/or purchase order do not match the data sent in the previous run. However, the seller cannot construct the message $\{H(WM2, arg(C2))\}_{SK(Bob)}$ from his current knowledge and is therefore unable to detect that the watermark *WM2* and/or purchase order *arg(C2)* do not match data sent in previous steps. The seller is therefore unable to build a case against the dishonest buyer, even if the buyer has illegally shared the watermarked document, as he is unable to gather both of the pieces of evidence required for prosecution.

As in the previous attack, by enabling unbinding in the protocol the buyer is able to avoid prosecution. Even if multiple illegal copies are made of multiple files, the seller is only able to prove that at least one watermarked document has been illegally distributed. He cannot prove how many of the purchased files have been copied or identify specifically which documents.

The attack is possible only because the seller is unable to verify the dual signature $\{H(wm, arg(c))\}_{SK(b)}$ received in step 7 of the protocol. We modify this dual signature into something that the seller can verify during the protocol run. By replacing the watermark, *wm*, with the watermark as encrypted with the buyer's public key, $\{wm\}_{PK(b)}$, the attack can be avoided. The seller is able to construct and verify the hash value, held within this new dual signature $\{H(\{wm\}_{PK(b)}, arg(c))\}_{SK(b)}$, using the other verifiable information he receives in the protocol run, specifically $\{wm\}_{PK(b)}$ and *arg(c)*.

7. DISCUSSION AND FURTHER WORK

In this paper we provided a novel approach to the analysis of buyer-seller watermarking protocols. We tailored an existing formal analysis technique, used previously to analyse communications and security protocols, to model the buyer-seller watermarking protocol as proposed by Ibrahim *et al.* We accurately represented their protocol by constructing a model using the process algebra CSP. By utilising the tool support associated with CSP we were able to conduct a thorough analysis of all the possible behaviour of the model. An unbinding attack on the protocol was discovered during the analysis and we have shown the counter example that was automatically generated by FDR in order to illustrate this attack. A second unbinding attack was described and revisions to both vulnerabilities were proposed in this paper.

Our protocol model can be extended, and additional specifications defined, in order to analyse the other properties as described by Ibrahim *et al.* We have modelled the protocol by Memon and Wong [9] in the same manner and our analysis reaffirmed the unbinding attack described in [7]. We are currently modelling and analysing other buyer-seller watermarking protocols that claim to satisfy a growing list of desirable security requirements. Our future work will examine how to provide a proof of such security properties in the general case, i.e. where the proof is independent of the number of participants and of particular cover material and watermarks. This will include reasoning about multiple runs of the protocol made by multiple agents of each type both consecutively and concurrently. This work is to be part of our longer term goal to develop a framework for CSP modelling of buyer-seller watermarking protocols.

8. ACKNOWLEDGMENTS

The authors would like to thank Johann Briffa and Steve Schneider for useful discussions related to this paper; and the reviewers for their valuable suggestions.

9. REFERENCES

- [1] C.-C. Chang and C.-Y. Chung. An enhanced buyer seller watermarking protocol. In *International Conference on Communication Technology Proceedings*, pages 1779–1783, 2003.
- [2] I. J. Cox, L. Miller, and J. A. Bloom. *Digital Watermarking*. Morgan Kaufmann, 2002.
- [3] S. Craver, N. Memon, B. L. Yeo, and M. Yeung. Resolving rightful ownerships with invisible watermarking techniques: limitations, attacks, and implications. *IEEE Journal on Selected Areas in Communications*, 16(4):573–586, 1998.
- [4] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.
- [5] C. A. R. Hoare. *Communicating sequential processes*. Prentice-Hall International, 1985.
- [6] I. M. Ibrahim, S. H. N. El-Din, and A. F. A. Hegazy. An effective and secure buyer seller watermarking protocol. In *Third International Symposium on Information Assurance and Security*, pages 21–28, 2007.
- [7] C.-L. Lei, P.-L. Yu, P.-L. Tsai, and M.-H. Chan. An efficient and anonymous buyer-seller watermarking

protocol. *IEEE Transactions on Image Processing*, 13(12):1618–1626, 2004.

- [8] G. Lowe. 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*, pages 147–166, 1996.
- [9] N. Memon and P. W. Wong. A buyer seller watermarking protocol. *IEEE Transactions on Image Processing*, 10(4):643–649, 2001.
- [10] L. Qiao and K. Nahrstedt. Watermarking schemes and protocols for protecting rightful ownership and customers' rights. *Journal of Visual Communication and Image Representation*, 9(3):194–210, 1998.
- [11] A. W. Roscoe. *The theory and practice of concurrency*. Prentice Hall, 1998.
- [12] A. W. Roscoe, P. Ryan, S. Schneider, M. Goldsmith, and G. Lowe. *The Modelling and Analysis of Security Protocols*. Addison-Wesley, 2001.
- [13] F. Systems. *FDR 2.82*. Formal Systems Ltd., 2005.
- [14] J. Zhang, W. Kou, and K. Fan. Secure buyer-seller watermarking protocol. *IEE Proceedings Information Security*, 153(1):15–18, 2006.

APPENDIX

A. FURTHER DETAILS OF DEDUCTIVE SYSTEM IN CSP

In this section we present the deductive process *INTELLIGENT_SELLER* which augments the model of the network so that we may analyse the overall behaviour of the system. We simplify Roscoe's lazy spy model [11] used in the analysis of the Needham Schroeder Public Key protocol. We no longer use the Dolev Yao model [4] used by Roscoe to model an outside intruder but instead construct an intelligent seller as illustrated in Figure 8.

Unlike Roscoe's lazy spy the intelligent seller cannot intercept or fabricate messages. The intelligent seller can only act passively, listening in to communications sent to the seller and making inferences about the information he attains. The intelligent seller is given the initial knowledge and a set of deduction rules let him break down a message sequence into its composite parts, encrypt/decrypt messages, embed/extract watermarks and hash messages. The intelligent seller then releases messages along the *sellerknows* channel once some knowledge learnt constitutes evidence, predefined in the *evidence* set. This communication on *sellerknows* enables us to make necessary observations during analysis. Similar to Roscoe's lazy spy model, *INTELLIGENT_SELLER* is defined in this manner to minimise issues caused by state space explosion.

$$\begin{aligned}
 \text{deductions}_1(X) &= \{(\{Sq.m\}, nth(j, m)), \\
 &\quad (\{nth(i, m) \mid i \leftarrow \{0..\#m - 1\}\}, Sq.m) \mid \\
 &\quad Sq.m \leftarrow X, j \leftarrow \{0..\#m - 1\}\} \\
 \text{deductions}_2(X) &= \{(\{m, PK(a)\}, \{m\}_{PK(a)}), \\
 &\quad (\{\{m\}_{PK(a)}, SK(a)\}, m), \\
 &\quad (\{\{m\}_{SK(a)}, SK(a)\}, m) \mid \\
 &\quad m \leftarrow \text{message}, a \leftarrow \text{agents}, \\
 &\quad \{m\}_{PK(a)} \leftarrow X\} \\
 \text{deductions}_3(X) &= \{(\{c, wm, WK(s)\}, [c \oplus wm]_{WK(s)}), \\
 &\quad (\{[c \oplus wm]_{WK(s)}, WK(s)\}, c), \\
 &\quad (\{[c \oplus wm]_{WK(s)}, WK(s)\}, wm) \mid
 \end{aligned}$$

$$\begin{aligned}
& c \leftarrow \text{covermaterial}, \\
& wm \leftarrow \text{watermarks}, \\
& s \leftarrow \text{sellors}, \\
& [c \oplus wm]_{WK(s)} \leftarrow X \\
\text{deductions}_4(X) &= \{(\{m\}, H(m)) \mid \\
& m \leftarrow \text{messages}, H(m) \leftarrow X\} \\
\text{deductions}_5(X) &= \{(\{c\}, \text{arg}(c)) \mid \\
& c \leftarrow \text{covermaterial}, \text{arg}(c) \leftarrow X\} \\
\text{deductions}(X) &= \bigcup_{i \in 1..5} \text{deductions}_i(X) \\
\\
\text{allfacts} &= \bigcup \{ \text{explode}(m) \mid \\
& m \leftarrow \text{messages} \} \\
\text{alldeductions} &= \text{deductions}(\text{allfacts}) \\
\text{possiblebasicknowledge} &= \text{known} \cup \text{messages} \\
\text{knowablefacts} &= \text{Close}(\text{possiblebasicknowledge}) \\
\text{learnablefacts} &= \text{knowablefacts} - \text{known} \\
\\
\text{Deductions} &= \{(X, f) \mid (X, f) \leftarrow \text{alldeductions}, \\
& f \in \text{learnablefacts}, \\
& f \notin X, X - \text{knowablefacts} = \emptyset\} \\
\\
\text{initialknowledge} &= \text{agents} \cup \{PK(a) \mid a \leftarrow \text{agents}\} \cup \\
& \{SK(\text{Sam}), WK(\text{Sam})\} \cup \text{covermaterial} \\
\\
\text{known} &= \text{Close}(\text{initialknowledge}) \\
\text{evidence} &= \{[c \oplus wm]_{WK(s)}, \{H(wm, \text{arg}(c))\}_{SK(b)} \mid \\
& b \leftarrow \text{buyers}, c \leftarrow \text{covermaterial}, \\
& wm \leftarrow \text{watermarks}\}
\end{aligned}$$

$$\begin{aligned}
\text{IGNORANTOF}(f) &= \\
& \langle f \in \text{messages} \rangle \text{learn}.f \rightarrow \text{KNOWS}(f) \\
& \square \\
& (\text{infer}?t \in \{(X, f') \mid (X, f') \leftarrow \text{Deductions}, f' = f\} \\
& \rightarrow \text{KNOWS}(f)) \\
\\
\text{KNOWS}(f) &= \\
& \langle f \in \text{messages} \rangle \text{learn}.f \rightarrow \text{KNOWS}(f) \\
& \square (\text{infer}?t \in \{(X, f') \mid (X, f') \leftarrow \text{Deductions}, f \in X\} \\
& \rightarrow \text{KNOWS}(f)) \\
& \square \langle f \in \text{evidence} \rangle \text{sellerknows}.f \rightarrow \text{KNOWS}(f) \\
\\
\text{LEARNKNOWN} &= (\text{learn}?f \in \text{known} \cap \text{messages} \\
& \rightarrow \text{LEARNKNOWN}) \\
\text{LEARN} &= \text{chase}(\langle \parallel \text{IGNORANTOF}(f) \rangle \setminus \{ \mid \text{infer} \mid \}) \\
& f \in \text{learnablefacts} \\
\\
\text{INTELLIGENT_SELLER} &= \\
& (\text{LEARN} \parallel \text{LEARNKNOWN}) \\
& \llbracket \text{comm}, \text{share}/\text{learn}, \text{learn} \rrbracket \\
\\
\text{SYSTEM} &= \text{NETWORK} \parallel \text{INTELLIGENT_SELLER}
\end{aligned}$$