# Preservation of Epistemic Properties in Security Protocol Implementations*

**Ron van der Meyden**
School of Computer Science and Engineering
University of New South Wales,
Sydney 2052, Australia
meyden@cse.unsw.edu.au

**Thomas Wilke**
Department of Computer Science
Christian-Albrechts-Universität zu Kiel,
24098 Kiel, Germany
wilke@ti.informatik.uni-kiel.de

## Abstract

We introduce (i) a general class of security protocols with private channel as cryptographic primitive and (ii) a probabilistic epistemic logic to express properties of security protocols. Our main theorem says that when a property expressed in our logic holds for an ideal protocol (where "ideal" means that the private channel hides everything), then it also holds when the private channel is implemented using an encryption scheme that guarantees perfect secrecy (in the sense of Shannon). Our class of protocols contains, for instance, an oblivious transfer protocol by Rivest and Chaum's solution to the dining cryptographers problem. In our logic we can express fundamental security properties of these protocols. The proof of the main theorem is based on a notion of refinement for probabilistic Kripke structures.

## 1 Introduction

Cryptographic protocols are an important building block for secure distributed systems. They often involve several agents with conflicting security goals, which makes the protocols difficult to design, let alone analyze. In many cases, it is already a difficult task to describe precisely the crucial security goals required of the protocols in question.

A common intuition concerning security protocols is that their specifications concern what parties to the protocol are permitted to know, and a recurrent thread in the literature (at least since [10, 11, 23]) has been the idea that they should be specified using some type of *epistemic logic* which can express properties like "agent $A$ knows message $m$" or "agent $B$ has not learned message $m$" or "for agent $C$ it is conceivable that $\varphi$ is true". Since probabilistic mechanisms play a key role in hiding information in security protocols, there has also been work (e. g. [24]) which includes operators for probability, such as "agent $A$ knows that $\varphi$ is the case with probability 3/4". In this paper, we use a logic—essentially that of Fagin and Halpern, see [21]—that combines operators for knowledge and probability. As we illustrate by a number of examples, this logic is particularly apt for specifying the security goals of multi-party protocols such as oblivious transfer protocols, bit commitment schemes, or protocols for solving problems like the dining cryptographers.

We also present a formal protocol model, in which we can encode protocols such as the ones mentioned above. The security primitive that we allow to be used in our protocols is the transfer of a message over an ideal private channel, from one agent to another. The semantics is such that a transmission over a private channel does not reveal anything about the message transferred to anyone other than the intended recipient.

The main contribution of our work is the following. We show that any implementation of a given protocol specified in our model where each transmission of a message over an ideal private channels is replaced by a broadcast of the message encrypted by an information-theoretically secure encryption scheme (in the sense of Shannon, [34]) satisfies exactly the same properties of our logic as the given protocol. In other words, all the security properties of an "ideal" protocol are preserved under implementations using information-theoretically secure encryption. Note that this means we are modelling a passive adversary.

Our result does not relieve us from designing secure protocols, but it makes verification a much simpler task. One only needs to check that an "ideal" version of a protocol is secure. It then follows that implementations are secure as well. In particular, this result provides an abstraction that enables us to reduce the size of the state space that needs to be analyzed in epistemic logic model checking analyses of security protocols, see, for instance, [29].

*Work supported in part by an Australian Research Council Discovery grant. We are grateful to Martín Abadi, Ralf Küsters, and Dusko Pavlovic for fruitful discussions on the topic and helpful pointers to references.

**Related work.** There is a whole range of work on logics for specifying cryptographic protocols, starting with Burrows, Abadí, and Needham's BAN logic [11] and many follow-ups, for instance [23, 9]; see [35] for a survey. Our definition of semantics for our protocol model is in the spirit of work on epistemic update, see, e. g., [28, 22, 7, 27]. This has also been applied to cryptographic protocols, see, e. g. [25, 26], but not aiming at preservation results. While all the logics just mentioned are inspired by modal logic, there is also a body of work which is inspired by Hoare logic, starting with a paper by Durgin, Mitchell, and Pavlovic [19, 20]. One of the more recent papers is [18].

There is a recent body of research that is concerned with relationships between abstract and cryptographic models of security protocols. One thread, starting with the seminal work of Abadi and Rogaway [3], deals with the justification of the abstract model of encryption in case of passive adversaries. This work has later been extended in different directions, including cryptographic justifications of abstract models in presence of active adversaries [30, 31, 17, 16].

At a lower level of abstraction is work that models cryptographic primitives and protocols as networks of probabilistic polynomial-time Turing machines, and uses the approach of simulation-based security [12, 14, 32, 5, 6]. In [13], oblivious transfer has been studied in a quite formal way in this framework.

Our treatment of private channels is in the spirit of body of work that deals with relationships between notions of process at various levels of abstraction. E. g., Abadi, Fournet, and Gonthier [2] refine processes using a secure channel primitive to a lower level calculus with an abstract model of encryption and active adversaries. Their notion of correctness is a type of observational equivalence of processes. Adão and Fournet [4] and Abadi et al. [1] follow a similar approach, but their low-level target structures are probabilistic polynomial-time Turing machines, that is, they work in a computational setting.

**Structure of the paper.** In Sect. 2, we give motivating examples, Sect. 3 contains the main definitions, that is, the definition of our protocol model and our epistemic logic, Sect. 4 presents the preservation theorem, and in Sect. 5, we outline its proof. Sect. 6 is a short conclusion.

## 2 Motivating Examples

In this section, we give two examples that motivate our results. We discuss an oblivious transfer protocol described by Rivest, [33], which is based on the BBCS quantum cryptographic protocol by Bennett, Brassard, Crépeau, and Skubiszewska, see [8], and Chaum's protocol for the dining cryptographers, see [15].

### 2.1 Rivest's Oblivious Transfer Protocol

In this protocol, which we call *ROT protocol*, Alice $(A)$ and Bob $(B)$ and a trusted third party Ted $(T)$ are involved. Alice is given two distinct messages $m_0, m_1 \in \{0,1\}^k$. Bob chooses $c \in \{0,1\}$ and wants to obtain $m_c$ from Alice. The protocol should be such that (ROT1) Alice does not learn $c$ and (ROT2) Bob does not learn anything about $m_{1-c}$. By (ROT0), we denote the correctness of the protocol, namely that Bob gets the desired message. To achieve all this, a trusted third party, Ted, is used.

The protocol works in four phases:

1. *Setup.* Ted chooses $r_0, r_1 \in \{0,1\}^k$ randomly and sends these values to Alice. Ted chooses $d \in \{0,1\}$ and sends $d$ and $r_d$ to Bob.
2. *Request.* Bob computes $e = c \oplus d$, where $\oplus$ denotes exclusive or, and sends $e$ to Alice.
3. *Reply.* Alice computes $f_0 = m_0 \oplus r_e$ and $f_1 = m_1 \oplus r_{1-e}$ and sends $f_0$ and $f_1$ to Bob.
4. *Result.* Bob computes $m = f_c \oplus r_d$.

One can easily prove that (ROT0) is the case and that (ROT1) and (ROT2) hold if Ted sends his messages over private channels during the setup phase.

One can also prove that (ROTA) Ted will not learn anything about $m_0$ nor $m_1$, provided Alice uses a private channel, and (ROTB) he will not learn anything about $c$, provided Bob uses a private channel.

### 2.2 The Dining Cryptographers

In this protocol, which we call *CDC protocol*, there are three cryptographers, $C_0$, $C_1$, and $C_2$, sitting at a dinner table, and just done with their meals, ready to pay. The waiter tells them their bill has already been paid for. It is immediately clear to them that either one of them has paid the bill (and thus treated the two others) or their national security agency (NSA) has paid for their expenses. They are curious and want to find out more, but if two of them were treated by the third they do not want his identity to be revealed. So the protocol should be such that (CDC0) each cryptographer finds out whether one of them has paid or whether it was the NSA, but (CDC1) if any one of the cryptographers has paid, the two others should not learn his identity.

Let us assume that we have a variable $p_i \in \{0,1\}$ for every one of the three cryptographers and that $p_i = 1$ iff $C_i$ paid for the dinner. So either all of the $p_i$'s are $0$ (the NSA has paid) or exactly one of the $p_i$'s is $1$.

The protocol Chaum suggested works as follows.

1. *Bit Sharing.* For every $i < 3$, cryptographer $C_i$ chooses a random bit $r_i \in \{0,1\}$ and sends it to his right neighbor, $C_{i+1 \bmod 3}$.

2. *Public Announcement.* For every $i < 3$, cryptographer $C_i$ computes $b_i = p_i \oplus r_i \oplus r_{i-1 \bmod 3}$ and announces $b_i$ publicly. Observe that he knows $r_{i-1 \bmod 3}$ since this is the bit he received from his left neighbor.

3. *Result.* Everyone computes $p = b_0 \oplus b_1 \oplus b_2$.

One can prove that $p = 0$ iff the NSA has paid, that is, (CDC0) holds, and that (CDC1) holds provided during bit sharing the cryptographers use private channels to communicate with each other.

What we will show for both protocols is that if the private channels (needed in the setup and the bit sharing phase, respectively, and potentially when Alice and Bob communicate) are implemented by any information-theoretically secure encryption scheme, then the resulting systems satisfy the required properties for both protocols. In fact, our result is much more general: We will show the same for any protocol in our model and for any property specified in a specific logic of knowledge and probability.

## 3 Protocol Model and Specification Logic

In this section, we describe our protocol model and our epistemic specification logic. The definition of the semantics of our protocol model uses the concept of updates for epistemic logics and thus needs the specification logic. (We note that the update semantics could be justified with respect to a standard operational semantics (cf. [28]): we work directly with the update semantics here since this is convenient for the proof of refinement.)

### 3.1 Protocols

For the purpose of this paper, a *protocol variable* is a variable in the ordinary sense, but each such variable has a fixed domain. When $v$ stands for a variable, we write $\text{dom}(v)$ for its domain. We assume that all domains are subsets of a set $U$, the *universe*. A variable assignment for a set of protocol variables $V$ is a function $\alpha\colon V \to U$ such that $\alpha(v) \in \text{dom}(v)$ for each $v \in V$.

A *protocol* is a tuple

$$\mathscr{P} = \langle \mathit{Agts}, V_{\mathit{inp}}, V_{\mathit{intr}}, \iota, \Theta \rangle \tag{1}$$

where

- *Agts* is a set of *agents,*
- $V_{\mathit{inp}}$ is a set of protocol variables, the *input variables,*
- $V_{\mathit{intr}}$ is a finite set of protocol variables, the *introduced variables,* with $V_{\mathit{inp}} \cap V_{\mathit{intr}} = \emptyset$,
- $\iota\colon \mathit{Agts} \to V_{\mathit{inp}}$ is a function that determines for every agent the set $\iota(A)$ of input variables that it can access initially, and
- $\Theta$ is the *protocol text* to be described next.

A *protocol text* is simply a sequence $A_0\colon \gamma_0$, $A_1\colon \gamma_1$, …, $A_{r-1}\colon \gamma_{r-1}$ where the $A_i$'s are agents and the $\gamma_i$'s are commands to be specified in the next paragraph. An expression of the form $A_i\colon \gamma_i$ should be understood as agent $A_i$ carries out command $\gamma_i$.

There are four types of *commands*. Each *reads* and *introduces* a set of variables. First, a *broadcast* is of the form $\texttt{broadcast}(v)$ where $v$ is any protocol variable. This command reads $v$ and introduces no variables. The intuitive meaning is that the value of $v$ is broadcast to all agents. Second, a *transmission* over a private channel is of the form $v \to B.v'$ where $B$ is an agent and $v$ and $v'$ are protocol variables. Here, the intuitive meaning is that the value of variable $v$ is sent to $B$ securely and stored in $v'$. This command reads $v$ and introduces $v'$. Third, an *expression* is of the form $v = f(v_0, \ldots, v_{s-1})$ where $v$ and the $v_i$'s are protocol variables and $f$ is any function with the right domain and range. This command reads $v_0, \ldots, v_{s-1}$ and introduces $v$. Fourth, a *randomization* is of the form $\texttt{randomize}(v)$ for some protocol variable $v$. This command reads nothing and introduces $v$.

When describing protocols one often writes something like "Alice sends $v$ to Bob", which in our framework cannot be modelled directly. We would have to use a further variable $v'$ accessible to Bob where the value of $v$ would be stored. This is the reason we also allow the following short form of transmission as syntactic sugar: $v \to B$. A protocol where such a command occurs should be viewed as a protocol where $v \to B.v'$ is executed for a new variable $v'$ and where every later occurrence of $v$ in a command executed by $B$ is replaced by $v'$. Clearly, if more than one such command occurs, the described transformation of the protocol has to be performed several times, for each such command in turn.

Given a protocol as above, agent $A$ *can access* a variable $v$ at step $i$ if

- $v \in \iota(A)$ or
- there exists $j < i$ such that $\gamma_j$ is
  - $\texttt{broadcast}(v)$,
  - $v' \to A.v$ for some variable $v'$,
  - $v = f(v_0, \ldots, v_{s-1})$ for variables $v_k$ and an appropriate function $f$ and $A_j = A$, or
  - $\texttt{randomize}(v)$ and $A_j = A$.

A protocol text is said to be *well-formed* if each command (1) reads only variables that are accessible to the agent performing the command at that point, (2) every variable introduced is in $V_{\mathit{intr}}$, and (3) no variable is introduced more than once. We only consider protocols with well-formed protocol texts.

To illustrate the protocol notations we first reconsider Rivest's oblivious transfer protocol. A compact description of a protocol in our sense corresponding to the description in Subsection 2.1 is depicted in Figure 3.1 on the left. The agents, their input variables, their domains and initial access are indicated in the headline, the introduced variables are implicit. Semicolons are replaced by new lines. Observe that $r_d$ is a *variable* and different from the variable $r_0$ even if $d = 0$. Only two functions are used, $\oplus$ (in infix notation), which stands for 'exclusive or', and $\mathtt{ite}(\cdot,\cdot,\cdot)$. The latter returns its second or third argument depending on its first argument, that is, $\mathtt{ite}(d, x_0, x_1) = x_d$.

Note that the description of the ROT protocol must be changed if Alice and/or Bob are assumed to use private channels for the communication between each other.

Next, we reconsider Chaum's solution to the dining cryptographers problem. A corresponding protocol in our sense is depicted in Figure 3.1 on the right. Note that the values $e_0$, $e_1$, and $e_2$ are the same, but since we restrict ourselves to well-formed protocols, we have to use different variables. (In our context, the variables are not really needed, which will be explained later.)

## 3.2 Epistemic Structures

The idea of the semantics of our protocols and the commands from which they are composed is that they transform an initial state of mutual information of the agents into a new state of mutual information. The transformation corresponding to a protocol is just the result of composing the sequence of transformations corresponding to its commands. The following definition describes the semantic objects used to represent these states of mutual information, a type of Kripke structure representing knowledge and probability.

An *epistemic structure* is a tuple

$$\mathscr{S} = \langle \mathit{Agts}, W, \{\sim_A\}_{A\in\mathit{Agts}}, C, \{\mathrm{P}_c\}_{c\in C}, V, \pi\rangle \qquad (2)$$

where

- *Agts* is a finite set of agents,
- $W$ is a finite set of worlds,
- $\sim_A$ is an equivalence relation on $W$, the *indistinguishability relation* for $A$, for each $A \in \mathit{Agts}$,
- $C \subset 2^W$ is a partition of $W$ into *cells*,
- $\mathrm{P}_c \colon 2^c \to [0,1]$ is a (discrete) probability measure on $c$, for each $c \in C$,
- $V$ is a set of protocol variables,
- $\pi\colon W \to V \to U$ is a function that assigns to each world $w$ an assignment $\pi(w)$ to the protocol variables.

We restrict attention to finite structures in order to avoid measure-theoretic concerns. For a world $w$, we write $\mathit{cell}(w)$ for the element $c \in C$ with $w \in c$.

## 3.3 The Specification Logic

As indicated above, part of the definition of the semantics of the protocol uses some aspects of the specification logic. This is why we next turn to the specification logic.

### 3.3.1 Syntax and Semantics.

Given a set *Agts* of agents and a set of variables $V$, the language $\mathscr{L}_{\mathrm{PK}(\mathit{Agts})}(V)$ is defined by:

- for each variable $v \in V$ and each $a \in \mathrm{dom}(v)$, $v = a$ is an atomic formula; if $\mathrm{dom}(v) = \{0,1\}$, then $v = 1$ may be written as $v$ and $v = 0$ may be written as $\neg v$,
- if $\varphi$ and $\psi$ are formulas, then $\neg\varphi$ and $\varphi \vee \psi$ are formulas,
- for each variable $v$ and variables $v_0,\ldots,v_{s-1}$ and function $f\colon \mathrm{dom}(v_0)\times\cdots\times\mathrm{dom}(v_{s-1}) \to \mathrm{dom}(v)$, the expression $v = f(v_0,\ldots,v_{s-1})$ is an atomic formula;
- for every $A \in \mathit{Agts}$, if $\varphi$ is a formula, then $\mathsf{K}_A\varphi$ is a formula,
- for every $A \in \mathit{Agts}$ and real number $r \in [0,1]$, if $\varphi$ is a formula, then $\mathrm{Pr}_A\varphi = r$ is a formula.

As usual, we use abbreviations for boolean constants and connectives such as $\top$, $\wedge$, and $\rightarrow$, but also $\mathsf{P}_A\varphi$, which stands for $\neg\mathsf{K}_A\neg\varphi$.

Before we define the satisfaction relation, we introduce some more terminology. Let $c$ be any cell and $X, S \subseteq W$. Then we write $\mathrm{P}_c(X)$ for $\mathrm{P}_c(X\cap c)$ and, similarly, provided $\mathrm{P}_c(S\cap c) \neq \emptyset$, we write $\mathrm{P}_c(X \mid S)$ for the conditional probability $\mathrm{P}_c(X\cap c \mid S \cap c)$.

The relation of satisfaction is now defined as follows:

- $\mathscr{S}, w \models v = a$ if $\pi(w)(v) = a$;
- $\mathscr{S}, w \models \neg\varphi$ if not $\mathscr{S}, w \models \varphi$;
- $\mathscr{S}, w \models \varphi\vee\psi$ if $\mathscr{S}, w \models \varphi$ or $\mathscr{S}, w \models \psi$;
- $\mathscr{S}, w \models v = f(v_0,\ldots,v_{s-1})$ if $\pi(w)(v) = f(\pi(w)(v_0),\ldots,\pi(w)(v_{s-1}))$;
- $\mathscr{S}, w \models \mathsf{K}_A\varphi$ if $\mathscr{S}, w' \models \varphi$ for all $w' \in W$ such that $w' \sim_A w$;
- $\mathscr{S}, w \models \mathrm{Pr}_A\varphi = r$ if $\mathrm{P}_{\mathit{cell}(w)}([\varphi]_{\mathscr{S}} \mid I_A(w)) = r$ where $[\varphi]_{\mathscr{S}} = \{v \in W \mid \mathscr{S}, v \models \varphi\}$ and $I_A(w) = \{v \in \mathit{cell}(w) \mid w \sim_A v\}$.

### 3.3.2 Examples.

Although we have not defined the semantics of our protocol model yet, we can specify properties of our running examples. We just have to imagine that the execution of a protocol yields an appropriate epistemic structure.

We start with the ROT protocol. First, we would like to specify (ROT0), namely that after the protocol has been

```
$T; A(m_0, m_1\colon \{0,1\}^k); B(c\colon \{0,1\})$
—Setup
$T\colon \texttt{randomize}(r_0)$
$T\colon \texttt{randomize}(r_1)$
$T\colon r_0 \to A$
$T\colon r_1 \to A$
$T\colon \texttt{randomize}(d)$
$T\colon r_d = \texttt{ite}(d, r_0, r_1)$
$T\colon d \to B$
$T\colon r_d \to B$
—Request
$B\colon e = c \oplus d$
$B\colon \texttt{broadcast}(e)$
—Reply
$A\colon r_e = \texttt{ite}(e, r_0, r_1)$
$A\colon f_0 = m_0 \oplus r_e$
$A\colon r_{1-e} = \texttt{ite}(e, r_1, r_0)$
$A\colon f_1 = m_1 \oplus r_{1-e}$
$A\colon \texttt{broadcast}(f_0)$
$A\colon \texttt{broadcast}(f_1)$
—Result
$B\colon g_c = \texttt{ite}(c, f_0, f_1)$
$B\colon m_c = g_c \oplus r_d$
```

```
$C_i(p_i\colon \{0,1\}), i < 3$
—Bit Sharing
$C_0\colon \texttt{randomize}(r_0)$
$C_0\colon r_0 \to C_1$
$C_1\colon \texttt{randomize}(r_1)$
$C_1\colon r_1 \to C_2$
$C_2\colon \texttt{randomize}(r_2)$
$C_2\colon r_2 \to C_0$
—Public Announcement
$C_0\colon b_0 = r_0 \oplus r_2 \oplus p_0$
$C_0\colon \texttt{broadcast}(b_0)$
$C_1\colon b_1 = r_1 \oplus r_0 \oplus p_1$
$C_1\colon \texttt{broadcast}(b_1)$
$C_2\colon b_2 = r_2 \oplus r_1 \oplus p_2$
$C_2\colon \texttt{broadcast}(b_2)$
—Result
$C_0\colon e_0 = b_0 \oplus b_1 \oplus b_2$
$C_1\colon e_1 = b_0 \oplus b_1 \oplus b_2$
$C_2\colon e_2 = b_0 \oplus b_1 \oplus b_2$
```

Figure 1: Formal descriptions of the ROT (left) and the CDC protocol (right)

executed Bob knows the message $m_c$:

$$\bigwedge_{z\in\{0,1\}} \left( c = z \to \bigwedge_{a\in\{0,1\}^k} (m_z = a \to \mathsf{K}_B(m_z = a)) \right) .$$

Observe that for this to be true, the last two steps of the protocol can be left out, because the semantics of the knowledge operator implies that Bob knows the value as soon as he has some information from which the value can be deduced. Also note that $c$ is a protocol variable and $z$ is just a meta variable representing $0$ and $1$ in different places of the formula.

We leave the formalization of (ROT1) to the reader and turn to (ROT2), which says that Bob does not know anything about $m_{1-c}$ except for the fact that it is different from $m_c$:

$$\bigwedge_{z\in\{0,1\}} \left( c = z \to \bigwedge_{a\in\{0,1\}^k} (a \neq m_z \to \mathsf{P}_B(m_{1-z} = a)) \right) .$$

We can modify this example slightly. Let us assume that $m_0$ and $m_1$ are not chosen non-deterministically, but randomly with equal probability, that is, every pair $(m_0, m_1)$ is chosen with probability $1/(2^k(2^k-1))$. Then we can make a stronger statement about what we expect from the protocol. (ROT0) and (ROT1) do not change, but (ROT2) does:

$$\bigwedge_{z\in\{0,1\}} \left( c = z \to \bigwedge_{a\in\{0,1\}^k} (a \neq m_z \to \left( \Pr_B(m_{1-z} = a) = \frac{1}{2^k - 1} \right) \right) \right) .$$

Next, we look at CDC. We first express (CDC0):

$$\bigwedge_{i<3} (\mathsf{K}_{C_i}(p_0 \vee p_1 \vee p_2) \vee \mathsf{K}_{C_i}\neg(p_0 \vee p_1 \vee p_2)) .$$

The other property, (CDC1), can be expressed by

$$\bigwedge_{i<3} \left( p_i \to \bigwedge_{j\neq i} \bigwedge_{k\neq j} \mathsf{P}_{C_j} p_k \right) .$$

As a further example, let us consider the following variation. Instead of three, we assume there are four agents, $C_0, \ldots, C_3$. That anyone should be so generous as to pay for everyone is a complete surprise, but suppose that it is common knowledge that $C_2$ and $C_3$ always share costs by flipping a coin to decide who pays on a given occasion. These assumptions would be represented by an initial epistemic structure $\mathscr{S}_{DC4}$ with four cells, $\{\mathrm{NSA}\}$, $\{0\}$, $\{1\}$, and $\{2,3\}$ where for the first three the probability measure assigns 1 to the only non-empty event and where for $c = \{2,3\}$, we have $\mathrm{P}_c(\{2\}) = \mathrm{P}_c(\{3\}) = 1/2$. Thus, we have

$$p_{\mathrm{NSA}} \vee p_0 \vee p_1 \vee (\Pr(p_2) = 1/2 \wedge \Pr(p_3) = 1/2))$$

before the execution of the protocol, where, for simplicity, we use $p_{\mathrm{NSA}}$ for $\bigwedge_{j<4} \neg p_j$. After running the protocol, we now require more specific properties for $C_0$ and $C_1$:

$$\bigwedge_{i<2} (\neg p_i \to (\mathsf{K}_{C_i}(p_{\mathrm{NSA}}) \vee \varphi_i))$$

where

$$\varphi_i = \mathsf{P}_{C_i}(p_{1-i}) \wedge \\ \mathsf{P}_{C_i}(\Pr_{C_i}(p_2) = 1/2 \wedge \Pr_{C_i}(p_3) = 1/2)) \quad .$$

## 3.4 Protocol Execution

Protocols are executed starting at an initial epistemic structure, and transform this structure into a new structure. Not all protocols can be executed on all structures. A protocol $\mathscr{P}$ is *fit* for an epistemic structure $\mathscr{S}$ if it has the same set of agents, all the initial variables of $\mathscr{P}$ are variables of $\mathscr{S}$, and none of the introduced variables of $\mathscr{P}$ is a variable of $\mathscr{S}$.

Each of the commands of a protocol acts as a transformer of epistemic structures. We express the transformation using the following notion. An *action* is a tuple

$$\mathscr{A} = \langle \mathit{Agts}, E, V, \Phi, \{\sim_A\}_{A\in \mathit{Agts}}, \{\mathrm{P}_\varphi\}_{\varphi\in\Phi}, V', \rho\rangle \quad (3)$$

where

- *Agts* is a set of agents,
- $E$ is a (finite) set of *events*,
- $V$ is a set of protocol variables,
- $\Phi$ is a set of mutually exclusive conditions in the variables from $V$ that have semantics in the structure to be updated,
- $\sim_A$ is an equivalence relation on $E$, for every $A \in$ *Agts*,
- $\mathrm{P}_\varphi$ is a distribution on $E$, for every $\varphi \in \Phi$,
- $V'$ is a set of *new* protocol variables disjoint from $V$,
- $\rho\colon E \to V' \to U$ is an interpretation of the new variables.

An action is *fit* for an epistemic structure $\mathscr{S}$ if it has the same set of agents, $V$ is a subset of the variables of $\mathscr{S}$ and $V'$ is disjoint from the variables of $\mathscr{S}$. We write $\mathscr{S}, w \to^{\mathscr{A}} e$ if there exists $\varphi \in \Phi$ such that $\mathscr{S}, w \models \varphi$ and $\mathrm{P}_\varphi(e) > 0$.

Given an epistemic structure

$$\mathscr{S} = \langle \mathit{Agts}, W, \{\sim_A\}_{A\in \mathit{Agts}}, C, \{\mathrm{P}_c\}_{c\in C}, V, \pi\rangle$$

and an action

$$\mathscr{A} = \langle \mathit{Agts}, E, V_\Phi, \Phi, \{\sim^{\mathscr{A}}_A\}_{A\in \mathit{Agts}}, \{\mathrm{P}^{\mathscr{A}}_\varphi\}_{\varphi\in\Phi}, V', \rho\rangle$$

fit for $\mathscr{S}$, the *update* of $\mathscr{S}$ with respect to $\mathscr{A}$ is the structure: $\mathscr{S} \times \mathscr{A}$ with components

$$\langle \mathit{Agts}, W', \{\sim'_A\}_{A\in \mathit{Agts}}, C', \{\mathrm{P}'_c\}_{c\in C}, V \cup V', \pi'\rangle$$

where

- $W' = \{(w, e) \mid w \in W$ and $\mathscr{S}, w \to^{\mathscr{A}} e\}$,
- $C' = \{c \times \mathscr{A} \mid c \in C\}$ where $c \times \mathscr{A} = \{(w, e) \mid w \in C$ and $\mathscr{S}, w \to^{\mathscr{A}} e\}$,
- $(w, e) \sim'_A (u, d)$ if $w \sim_A w$ and $e \sim^{\mathscr{A}}_A d$,
- $\mathrm{P}'_{c\times A}((w, e)) = \mathrm{P}_c(w) \times \mathrm{P}^{\mathscr{A}}_\varphi(e)$, where $\varphi \in \Phi$ with $\mathscr{S}, s \models \varphi$,
- $\pi'((w, e)) = \pi(w) \cup \rho(e)$.

We can now describe the semantics of the commands that we allow in our protocols. The command $A$: `randomize`$(v)$ corresponds to the action defined by:

- $E = \mathrm{dom}(v)$,
- $V = \emptyset$,
- $\Phi = \{\top\}$,
- $\sim_A = \{(a, a) \mid a \in \mathrm{dom}(v)\}$ and $\sim_B = \mathrm{dom}(v) \times \mathrm{dom}(v)$ for $B \in \mathit{Agts} \setminus \{A\}$,
- $\mathrm{P}_\top$ is the uniform distribution on $\mathrm{dom}(v)$,
- $V' = \{v\}$,
- $\rho(a)(v) = a$ for every $a \in \mathrm{dom}(v)$.

The command $A\colon v = f(v_0, \ldots, v_{s-1})$ corresponds to the action defined by:

- $E = \mathrm{dom}(v)$,
- $V = \{v_0, \ldots, v_{s-1}\}$,
- $\Phi = \{f(v_0, \ldots, v_{s-1}) = a \mid a \in \mathrm{dom}(v)\}$,
- $\sim_A = \{(a, a) \mid a \in \mathrm{dom}(v)\}$ and $\sim_B = \mathrm{dom}(v) \times \mathrm{dom}(v)$ for $B \in \mathit{Agts} \setminus \{A\}$,
- $\mathrm{P}_{f(v_0,\ldots,v_{s-1})=a}(a) = 1$ and zero everywhere else,
- $V' = \{v\}$,
- $\rho(a)(v) = a$.

The command $A\colon v \to B.v'$ corresponds to:

- $E = \mathrm{dom}(v)$,
- $V = \{v\}$,
- $\Phi = \{v = a \mid a \in \mathrm{dom}(v)\}$,
- $\sim_B = \{(a, a) \mid a \in \mathrm{dom}(v)\}$ and $\sim_C = \mathrm{dom}(v) \times \mathrm{dom}(v)$ for $C \in \mathit{Agts} \setminus \{B\}$,
- $\mathrm{P}_{v=a}(a) = 1$ and zero everywhere else,
- $V' = \{v'\}$,
- $\rho(a)(v') = a$.

Finally, $A$: `broadcast`$(v)$ corresponds to:

- $E = \mathrm{dom}(v)$,
- $V = \{v\}$,
- $\Phi = \{v = a \mid a \in \mathrm{dom}(v)\}$,
- $\sim_B = \{(a, a) \mid a \in E\}$ for all $B \in \mathit{Agts}$,
- $\mathrm{P}_{v=a}(a) = 1$ and zero everywhere else,
- $V' = \emptyset$,
- $\rho(a) = \emptyset$.

Finally, it is easy to define the semantics of a protocol. Given an initial epistemic structure $\mathscr{S}$ and a protocol $\mathscr{P}$ fit for $\mathscr{S}$, let $\mathscr{A}_0, \ldots, \mathscr{A}_{r-1}$ be the sequence of actions corresponding to the protocol text. We then write $\mathscr{S}[\mathscr{P}]$ for the epistemic structure $\mathscr{S} \times \mathscr{A}_0 \times \cdots \times \mathscr{A}_{r-1}$ and call this the *result* of executing $\mathscr{P}$ on $\mathscr{S}$. (It is straightforward to check that if $\mathscr{P}$ is fit for $\mathscr{S}$, then the action $\mathscr{A}_0$ is fit for $\mathscr{S}$ and for each $j \in \{1, \ldots, r-1\}$ the action $\mathscr{A}_j$ is fit for $\mathscr{S} \times \mathscr{A}_0 \times \cdots \times \mathscr{A}_{j-1}$.) When we write $\mathscr{S}[\mathscr{P}] \models \varphi$ we mean that $\mathscr{S}[\mathscr{P}], w \models \varphi$ holds for every world $w$ of the structure $\mathscr{S}[\mathscr{P}]$.

## 4 Implementing Private Channels via Encryption Schemes and Main Result

We are now in a position to formally present the main theorem of the paper, which states that secure transmission operations can be safely implemented by a broadcast of encrypted messages, provided that the encryption keys are securely distributed to the intended recipients. Safety here will mean that satisfaction of formulas of the logic is not affected by this replacement.

A *shared key encryption scheme* for a space of messages $Msg$ is a tuple $\mathscr{E} = \langle K, G, \mathrm{P}_K, E, D\rangle$ where $K$ is a finite set of *keys*, $G$ is a process for randomly generating keys, $\mathrm{P}_K$ is a discrete probability measure on $K$ such that the probability that the process $G$ will generate a key $k \in K$ is $\mathrm{P}_K(k)$, and $E\colon K \times Msg \to Msg$ and $D\colon K \times Msg \to Msg$ are, respectively, encryption and decryption operations, such that $D(k, E(k,m)) = m$ for all $k \in K$ and $m \in Msg$.

We say that the encryption scheme is *possibilistically secure* if for each pair of messages $m, m' \in Msg$ and key $k \in K$, there exists a key $k'$ such that $E(k,m) = E(k', m')$. In other words, each ciphertext $E(k,m)$ could have come from any plaintext $m'$. However, the *a priori* probability that $E(k,m)$ was produced from plaintext $m'$ might be low, so possibilistically secure encryption schemes may still be subject to statistical attacks. A condition that eliminates such attacks is Shannon's *perfect secrecy* property [34]. The encryption scheme is defined to have this property if for all messages $m, m' \in Msg$, we have $\mathrm{P}_K(\{k' \in K \mid E(k,m) = E(k',m)\}) = \mathrm{P}_K(\{k' \in K \mid E(k,m) = E(k',m')\})$. That is, using a randomly generated key, a message $m'$ is as likely to be encrypted to $E(k,m)$ as is the message $m$ itself.[1]

In order to enable use of an encryption scheme $\mathscr{E}$ in protocols, introduce commands $x = E(k,m)$, $x = D(k,m)$, and $\texttt{generate}(k)$. Semantically, the first two of these correspond to assignment actions defined just as above, and the last corresponds to an action that is identical to $\texttt{randomize}(k)$ except that the distribution $\mathrm{P}_K$ is used rather than the uniform distribution (if it differs).

If $\mathscr{P}$ is a protocol, let $\mathscr{P}^*$ be obtained from $\mathscr{P}$ by substituting for each secure transmission command of the form $A\colon m \to B.v$ occurring in $\mathscr{P}$ the sequence

$$\begin{aligned} &A\colon e = E(k_a, m)\\ &A\colon \texttt{broadcast}(e)\\ &B\colon v = D(k_b, e) \end{aligned} \tag{4}$$

and prepending

$$\begin{aligned} &A\colon \texttt{generate}(k_a)\\ &A\colon k_a \to B.k_b \end{aligned} \tag{5}$$

to the protocol where an appropriate encryption scheme and a fresh set of variables $k_a, k_b, e, d$ is used for each secure transmission command.

Note that (5) describes the process in symmetric key encryption that is typically carried out before the actual encryption takes place: the principals involved in the communication exchange a secret key. The actual encryption of the plaintext and the transmission of the ciphertext are described by (4).

Note also that in (5) it is not necessary that $A$ generates the key and sends it to $B$. Alternatively, one could also use

$$\begin{aligned} &B\colon \texttt{generate}(k_b)\\ &B\colon k_b \to A.k_a\ . \end{aligned} \tag{6}$$

Finally, note that instead of prepending (5) we could also modify the epistemic structure we start with. For each secure transmission command as above, we could add a variable $k$ to the set of protocol variables $V$ instead of using the two variables $k_a$ and $k_b$, make sure $k$ is assigned a value in a random fashion according to the distribution on the key space, and make $k$ accessible to $A$ and $B$ only. We could then replace (4) by

$$\begin{aligned} &A\colon e = E(k, m)\\ &A\colon \texttt{broadcast}(e)\\ &B\colon v = D(k, e)\ . \end{aligned} \tag{7}$$

**Theorem 1 (preservation theorem)** *Let $\mathscr{S}$ be an epistemic structure such that $\mathscr{P}$ is fit for $\mathscr{S}$. Then $\mathscr{P}^*$ (as obtained above) is is fit for $\mathscr{S}$ and for each formula $\varphi \in \mathscr{L}_{PK(Agts)}$ we have $\mathscr{S}[\mathscr{P}] \models \varphi$ iff $\mathscr{S}[\mathscr{P}^*] \models \varphi$, provided either*

1. *the encryption schemes used are possibilistically secure and $\varphi$ does not contain probability operators, or*
2. *the encryption schemes used satisfy the perfect secrecy condition.*

Intuitively, this result states that when they run $\mathscr{P}^*$ and information is sent in an encrypted broadcast form, visible to all agents, rather than through private channels, the agents to the protocol (and others) do not learn anything about the variables of $\mathscr{S}$ and $\mathscr{P}$ that they would not have learnt when running $\mathscr{P}$. Note that the result concerns security against passive adversaries, who intercept the broadcast communications, but have no powers to prevent message delivery or inject false messages.

[1]This is one of several equivalent formulations presented in [34].

## 5 Refinements and Proof of Main Theorem

In this section we outline the proof of Theorem 1; we restrict attention to the probabilistic case. The main ingredient of the proof is the notion of refinement between epistemic structures.

### 5.1 Refinements

We start with the definition of refinements. Let $\mathscr{S}$ be an epistemic structure as usual and let

$$\mathscr{T} = \langle \mathit{Agts}, W', \{\sim'_A\}_{A\in \mathit{Agts}}, C', \{\mathrm{P}'_c\}_{c\in C}, V', \pi' \rangle$$

be another epistemic structure with an identical set of agents *Agts*. A *refinement mapping* from $\mathscr{S}$ to $\mathscr{T}$ is a function $r\colon W \to W'$ such that

R1. $V' \subseteq V$;
R2. $\pi(w)(v) = \pi'(r(w))(v)$ for all $w \in W$ and $v \in V$;
R3. if $\mathit{cell}(w) = \mathit{cell}(w')$ then $\mathit{cell}'(r(w)) = \mathit{cell}'(r(w'))$, for $w, w' \in W$;
R4. if $w \sim_A w'$ then $r(w) \sim'_A r(w')$, for $w, w' \in W$ and $A \in \mathit{Agts}$;
R5. for $w \in W$ and $w' \in W'$ and $A \in \mathit{Agts}$, if $r(w) \sim'_A w'$ then there exists $w'' \in W$ such that $w \sim_A w''$ and $r(w'') = w'$;
R6. $\mathrm{P}'_{\mathit{cell}'(r(w))}(w') = \mathrm{P}_{\mathit{cell}(w)}(r^{-1}(w'))$ for $w \in W$ and $w' \in \mathit{cell}'(r(w))$.
R7. for $w \in W$, $A \in \mathit{Agts}$, and $w' \in I_A(r(w))$,

$$\mathrm{P}'_{\mathit{cell}'(r(w))}(w' \mid I_A(r(w))) = \mathrm{P}_{\mathit{cell}(w)}(r^{-1}(w') \mid I'_A(w))$$

where $I_A(w) = \{w'' \in \mathit{cell}(w) \mid w'' \sim_A w\}$ and, similarly, $I'_A(r(w)) = \{w'' \in \mathit{cell}(w) \mid w'' \sim'_A r(w)\}$.

We say that $\mathscr{S}$ is a *refinement* of $\mathscr{T}$ if there exists a refinement mapping $r$ from $\mathscr{S}$ to $\mathscr{T}$. In this case, we write $\mathscr{S} \preceq_r \mathscr{T}$, or $\mathscr{S} \preceq \mathscr{T}$ for short.

### 5.2 Basic Facts about Refinements

We state some basic facts about refinements, after which we can describe in more detail how the proof of Theorem 1 is structured.

It is straightforward to check that "being a refinement of" is a transitive relation:

**Lemma 1 (transitivity)** *Assume $\mathscr{S}$, $\mathscr{T}$, and $\mathscr{U}$ are epistemic structures such that $\mathscr{S} \preceq \mathscr{T}$ and $\mathscr{T} \preceq \mathscr{U}$. Then $\mathscr{S} \preceq \mathscr{U}$.*

Refinement is also easily seen to be preserved under application of actions:

**Proposition 1 (refinement and actions)** *Let $\mathscr{S}$ and $\mathscr{T}$ be epistemic structures such that $\mathscr{S} \preceq \mathscr{T}$ and let $\mathscr{A}$ be an action fit for $\mathscr{T}$. Then $\mathscr{A}$ is fit for $\mathscr{S}$ and $\mathscr{S}\times\mathscr{A} \preceq \mathscr{T}\times\mathscr{A}$.*

The following result states that formulas over the variables of the target structure are preserved by refinement mappings.

**Proposition 2 (preservation by refinement)** *Let $V$ be the set of variables of the epistemic structure $\mathscr{T}$ and assume $\mathscr{S} \preceq_r \mathscr{T}$. Then for all worlds $w$ of $\mathscr{S}$ and formulas $\varphi \in \mathscr{L}_{\mathit{Agts}}(V)$,*

$$\mathscr{S}, w \models \varphi \quad \textit{iff} \quad \mathscr{T}, r(w) \models \varphi\ .$$

**Sketch of proof** The proof is an induction on the structure of $\varphi$. □

### 5.3 Sketch of the Proof of Theorem 1

In view of Proposition 2, to prove Theorem 1 it suffices to show $\mathscr{S}[\mathscr{P}^*] \preceq \mathscr{S}[\mathscr{P}]$. We will prove this using an intermediate protocol. Let $\mathscr{P}^\#$ be the protocol obtained from $\mathscr{P}$ by substituting for each secure transmission command of the form $A\colon m \to B.v$ occurring in $\mathscr{P}$ the sequence

$$\begin{array}{l} A\colon \mathtt{generate}(k_a) \\ A\colon k_a \to B.k_b \\ A\colon e = E(k_a, m) \\ A\colon \mathtt{broadcast}(e) \\ B\colon v = D(k_b, e) \end{array} \qquad (8)$$

with $k_a$, $k_b$, $E$, and $D$ as above. We show that $\mathscr{S}[\mathscr{P}^*] \preceq \mathscr{S}[\mathscr{P}^\#]$ (see Subsection 5.4) and $\mathscr{S}[\mathscr{P}^\#] \preceq \mathscr{S}[\mathscr{P}]$ (see Subsection 5.5), which, by Lemma 1, then implies $\mathscr{S}[\mathscr{P}^*] \preceq \mathscr{S}[\mathscr{P}]$.

### 5.4 Commutation Rules

We start with a fairly general result on the effect of reversing the order of two consecutive actions.

An action as in (3) has *propositional preconditions* if all $\varphi \in \Phi$ are boolean combinations of atomic propositions of the form $x = y$ where $x$ and $y$ are variables or constants.

Actions $\mathscr{A}$ and $\mathscr{B}$ are *concurrently fit* for an epistemic structure $\mathscr{S}$ if $\mathscr{A}$ and $\mathscr{B}$ are fit for $\mathscr{S}$, action $\mathscr{A}$ is fit for $\mathscr{S} \times \mathscr{B}$, and $\mathscr{B}$ is fit for $\mathscr{S} \times \mathscr{A}$.

**Lemma 2 (commutation rule)** *Let $\mathscr{A}$ and $\mathscr{B}$ be any actions with propositional preconditions such that $\mathscr{A}$ and $\mathscr{B}$ are concurrently fit for $\mathscr{S}$. Then*

$$\mathscr{S} \times \mathscr{B} \times \mathscr{A} \preceq \mathscr{S} \times \mathscr{A} \times \mathscr{B}\ .$$

**Sketch of proof** The worlds of $\mathscr{S}\times\mathscr{B}\times\mathscr{A}$ and $\mathscr{S}\times\mathscr{A}\times\mathscr{B}$ have the forms $((w, b), a)$ and $((w, a), b)$, respectively, and the mapping $((w,b),a) \mapsto ((w,a),b)$, can be shown to be a refinement. $\square$

There are two important consequences of Lemma 2:

**Lemma 3 (rule for randomization)** *Let $\mathscr{A}$ be any action and let $\mathscr{B}$ be the action corresponding to $A$: $\texttt{generate}(k)$ such that $\mathscr{A}$ and $\mathscr{B}$ are fit for $\mathscr{S}$ and $\mathscr{S} \times \mathscr{A}$, respectively. Then $\mathscr{B}$ and $\mathscr{A}$ are fit for $\mathscr{S}$ and $\mathscr{S} \times \mathscr{B}$, respectively, and $\mathscr{S} \times \mathscr{B} \times \mathscr{A} \preceq \mathscr{S} \times \mathscr{A} \times \mathscr{B}$.*

**Lemma 4 (rule for private channel)** *Let $\mathscr{A}$ be any action corresponding to a protocol command that does not introduce the protocol variable $v$ and $\mathscr{B}$ be the action corresponding to $A\colon v \to B.v'$ such that $\mathscr{A}$ and $\mathscr{B}$ are fit for $\mathscr{S}$ and $\mathscr{S} \times \mathscr{A}$, respectively. Then $\mathscr{B}$ and $\mathscr{A}$ are fit for $\mathscr{S}$ and $\mathscr{S} \times \mathscr{A}$, respectively, and $\mathscr{S} \times \mathscr{B} \times \mathscr{A} \preceq \mathscr{S} \times \mathscr{A} \times \mathscr{B}$.*

From Lemmas 1, 3 and 4 and Proposition 1, we obtain by induction:

**Proposition 3** *For $\mathscr{S}$ and $\mathscr{P}$ as in Theorem 1, $\mathscr{S}[\mathscr{P}^*] \preceq \mathscr{S}[\mathscr{P}^\#]$.*

### 5.5 Implementing Secure Channels

We next note that implementing secure channels can be dealt with:

**Proposition 4 (implementing a private channel)** *Let $\mathscr{A}$ be the action corresponding to $A\colon m \to B.x$ and let $\mathscr{A}^*$ be the action sequence corresponding to (8) when cryptography is interpreted using an encryption scheme $\mathscr{E}$. Suppose $\mathscr{S} \preceq_r \mathscr{T}$.*

*Then if $\mathscr{E}$ satisfies the perfect secrecy property, then there exists $s$ such that $\mathscr{S} \times \mathscr{A}^* \preceq_s \mathscr{T} \times \mathscr{A}$.*

**Sketch of proof** Since the action $\mathscr{A}$ is deterministic, the worlds of $\mathscr{T}[\mathscr{A}]$ and $\mathscr{T}$ are in 1-1 correspondence, so we may use the same symbol $w$ to refer to a world of $\mathscr{T}$ and the world $(w,e)$ representing the effect of having performed the action $A\colon m \to B.x$ on $w$. Similarly, all actions corresponding to commands in $\mathscr{A}^*$ except $\texttt{generate}(k_a)$ are deterministic, so if $w$ is a world of $\mathscr{S}$ and $k$ is a key then we may write $w+k$ to refer to the world of $\mathscr{S}[\mathscr{A}^*]$ produced when the random choice for $k_a$ is $k$. Using these identifications, define the mapping $s$ by $s(w+k) = r(w)$. This can be shown to be a refinement mapping. $\square$

From the previous proposition and Proposition 1, we obtain by induction:

**Proposition 5** *For $\mathscr{S}$ and $\mathscr{P}$ as in Theorem 1, $\mathscr{S}[\mathscr{P}^\#] \preceq \mathscr{S}[\mathscr{P}]$.*

## 6 Conclusion

We have introduced a model for security protocols as well as an appropriate epistemic logic for specifying security properties of such protocols, and proved that a notion of implementation that transforms secure channels into encrypted broadcasts preserves these epistemic specifications, provided information-theoretic secure encryption schemes are used.

The related work on the relations between abstract and concrete models of protocols mentioned in the introduction is largely concerned with showing that implementations keep secrets from an active adversary. By contrast, our work is concerned with the preservation of a richer class of properties (involving also trusted principals) expressible in the logic of knowledge and probability. On the other hand, we consider only passive adversaries, and our notion of implementation uses a less realistic type of cryptographic primitive and does not take computational complexity concerns into account. We hope to address this concern in future work.

## References

[1] M. Abadi, R. Corin, and C. Fournet. Computational secrecy by typing for the pi-calculus. In *Proc. ASIAN Symp. on Programming Languages and Systems*, pages 253–269. Springer LNCS 4279, 2006.

[2] Martín Abadi, Cédric Fournet, and Georges Gonthier. Secure implementation of channel abstractions. *Inf. Comput.*, 174(1):37–83, 2002.

[3] Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. Cryptology*, 15(2):103–127, 2002.

[4] Pedro Adão and Cédric Fournet. Cryptographically sound implementations for communicating processes. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 83–94. Springer, 2006.

[5] M. Backes, B. Pfitzmann, and M. Waidner. A composable cryptographic library with nested operations. In S. Jajodia, V. Atluri, and T. Jaeger, editors, *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS 2003)*, pages 220–230. ACM, 2003.

[6] Michael Backes and Birgit Pfitzmann. Relating symbolic and cryptographic secrecy. *IEEE Trans. Dependable Sec. Comput.*, 2(2):109–123, 2005.

[7] Alexandru Baltag. A logic for suspicious players: epistemic actions and believe update in games. *Bulletin of Economic Research*, 54:1–46, 2002.

[8] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Marie-Hélène Skubiszewska. Practical quantum oblivious transfer. In Joan Feigenbaum, editor, *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 351–366. Springer, 1991.

[9] Stephen H. Brackin. A HOL extension of GNY for automatically analyzing cryptographic protocols. In *CSFW*, pages 62–76. IEEE Computer Society, 1996.

[10] Michael Burrows, Martín Abadi, and Roger M. Needham. A logic of authentication. *ACM Trans. Comput. Syst.*, 8(1):18–36, 1990.

[11] Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication. *Proc. Roy. Soc. London Ser. A*, 426(1871):233–271, December 1989.

[12] R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS 2001)*, pages 136–145. IEEE Computer Society, 2001.

[13] Ran Canetti, Ling Cheung, Dilsun Kirli Kaynar, Moses Liskov, Nancy A. Lynch, Olivier Pereira, and Roberto Segala. Time-bounded task-pioas: A framework for analyzing security protocols. In Shlomi Dolev, editor, *DISC*, volume 4167 of *Lecture Notes in Computer Science*, pages 238–253. Springer, 2006.

[14] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In Birgit Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 453–474. Springer, 2001.

[15] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *J. Cryptology*, 1(1):65–75, 1988.

[16] V. Cortier, S. Kremer, R. Küsters, and B. Warinschi. Computationally Sound Symbolic Secrecy in the Presence of Hash Functions. In *Proceedings of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2006)*, volume 4337 of *Lecture Notes in Computer Science*, pages 176–187. Springer, 2006.

[17] V. Cortier and B. Warinschi. Computationally Sound, Automated Proofs for Security Protocols. In *Proceedings of the 14th European Symposium on Programming (ESOP 2005)*, volume 3444 of *Lecture Notes in Computer Science*. Springer, 2005.

[18] Anupam Datta, Ante Derek, John C. Mitchell, and Arnab Roy. Protocol composition logic (pcl). *Electr. Notes Theor. Comput. Sci.*, 172:311–358, 2007.

[19] Nancy A. Durgin, John C. Mitchell, and Dusko Pavlovic. A compositional logic for protocol correctness. In *CSFW*, pages 241–272. IEEE Computer Society, 2001.

[20] Nancy A. Durgin, John C. Mitchell, and Dusko Pavlovic. A compositional logic for proving security properties of protocols. *Journal of Computer Security*, 11(4):677–722, 2003.

[21] Ronald Fagin and Joseph Y. Halpern. Reasoning about knowledge and probability. *J. ACM*, 41(2):340–367, 1994.

[22] Jelle Gerbrandy. Dynamic epistemic logic. Tech. rep. LP-97-04, Institute for Logic, Languages and Computation, Amsterdam, 1997.

[23] Li Gong, Roger M. Needham, and Raphael Yahalom. Reasoning about belief in cryptographic protocols. In *IEEE Symposium on Security and Privacy*, pages 234–248, 1990.

[24] Joseph Y. Halpern and Kevin R. O'Neill. Anonymity and information hiding in multiagent systems. *Journal of Computer Security*, 13(3):483–512, 2005.

[25] Arjen Hommersom. Reasoning about security. Master's thesis, Universiteit Utrecht, 2003.

[26] Arjen Hommersom, John-Jules Meyer, and Eric P. de Vink. Update semantics of security protocols. *Synthese*, 142:229–267, 2004. Knowledge, Rationality and Action subseries.

[27] Barteld P. Kooi. Probabilistic epistemic logic. *J. of Logic, Language and Information*, 12:381–408, 2003.

[28] Ron van der Meyden. Constructing finite state implementations of knowledge based programs with perfect recall. In *Intelligent Agent Systems: Theoretical and Practical Issues (Based on a Workshop Held at PRICAI'96, Cairns, Australia, Aug 1996)*, volume No. 1209 of *LNAI*, pages 135–152. Springer, 1997.

[29] Ron van der Meyden and Kaile Su. Symbolic model checking the knowledge of the dining cryptographers. In *CSFW*, pages 280–291. IEEE Computer Society, 2004.

[30] D. Micciancio and B. Warinschi. Completeness theorems for the abadi-rogaway logic of encrypted expressions. In *Workshop on Issues in the Theory of Security (WITS 2002)*, 2002.

[31] D. Micciancio and B. Warinschi. Soundness of Formal Encryption in the Presence of Active Adversaries. In M. Naor, editor, *First Theory of Cryptography Conference (TCC 2004)*, volume 2951 of *Lecture Notes in Computer Science*, pages 133–151. Springer, 2004.

[32] B. Pfitzmann and M. Waidner. A Model for Asynchronous Reactive Systems and its Application to Secure Message Transmission. In *IEEE Symposium on Security and Privacy*, pages 184–201. IEEE Computer Society Press, 2001.

[33] Ronald L. Rivest. Unconditionally secure commitment and oblivious transfer schemes using private channels and a trusted initializer. Unpublished, but available at `http://theory.lcs.mit.edu/∼rivest/publications.html`, November 1999.

[34] Claude E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28-4:656–715, 1949.

[35] Paul F. Syverson and Iliano Cervesato. The logic of authentication protocols. In Riccardo Focardi and Roberto Gorrieri, editors, *FOSAD*, volume 2171 of *Lecture Notes in Computer Science*, pages 63–136. Springer, 2000.