Dynamic Epistemic Logic for Protocol Analysis

Francien Dechesne, Jan van Eijck, Wouter Teepe, Yanjing Wang

The next day, the participants in the discussion on protocol analysis reconvene. This time, they have an in-depth exchange of ideas on possible uses of epistemic logic.

Logician: Yesterday we concluded that there are many things that need to be formalized for the analysis of security protocols. After a good night's sleep, I have the feeling that this may be a nice field of application for some kind of *dynamic* epistemic logic. It is about updates of knowledge after the passing of messages, and the protocols are designed to fulfill requirements in terms of knowledge or belief.

Computer Scientist: Yes, that sounds good, but how does one get started? I guess we should avoid going down the road of BAN-logic.

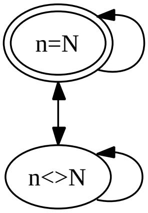
Logician: We definitely want to have a clear semantics. So why don't we take possible worlds semantics, as for modal logics, as the starting point? We then have a set of *possible worlds*, on which there is also a valuation function that gives the truth value for each primitive proposition. For each agent, his uncertainties about the real world are modeled by an *accessibility* relation on those worlds. A world that is accessible for an agent from a given world, is held to be *possible* by that agent in that world. For an agent to know ϕ, means that ϕ holds in each world that he considers possible from the *actual* world.

Security Analyst: But I see a complexity problem popping up in this semantics with respect to the cryptographic primitives. For example, suppose that we model all possible values that nonces could have. If the nonces don't have an upper bound, or even if the agents just don't know they have an upper bound, we would have to put an infinite number of possible worlds in our model. For each nonce whose value you don't know, there is then an infinite number of possible worlds you can't distinguish…

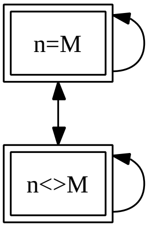
Logician: Yes, I've been pondering about that. But I think I have a nice solution! Suppose the actual value of the number n is N. Then one should lump together all worlds where n is different from N. So two worlds are enough to represent your uncertainty about N.

Computer Scientist: I see. I propose we call your new-style worlds *condensed worlds.* Instead of a single valuation, a condensed world has a non-empty set of valuations. I suppose this will work, but it seems rather awkward. Suppose one wishes to check whether $n = M$ is true in a condensed world. Then you may not get a single answer.

Logician: Still there is no need to go for a logic of partiality. Remember that what we have done till now is essentially a succinct representation of the huge possible worlds space. When we evaluate a formula we just need to split the condensed world to get relevant information. Here we take the dynamic approach. We replace evaluation in condensed worlds by updating with an appropriate *evaluation action model.* Let me draw some pictures. Here is the situation where you don't know the actual value N of n:

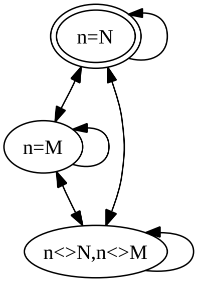


And here is an action model for checking whether the value is M or not:



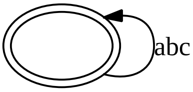
Notice that the equalities and inequalities in the boxes are the preconditions for the corresponding actions. An action can only happen on the worlds which satisfy the precondition of it.

Then the result of updating the condensed model with the action model is the following condensed Kripke model:

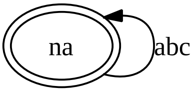


Computer Scientist: I see what you are getting at, and I will explain this general method of updating to the others in a while [1]. It makes the semantics pretty complicated when you want to evaluate a complex modal formula on a condensed world. Anyway, suppose it is well-defined, then I see another useful update action: valuation expansion. I suppose generating a nonce can be seen as a combination of first expanding the valuations in our representation with a new register, and next filling the register with a value.

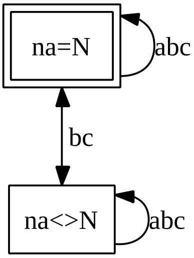
Logician: That's right. Let us try our hand at the analysis of the Needham-Schroeder protocol. We start with a situation of blissful ignorance, with an empty list of valuation registers. Assume there are three agents a, b, c.



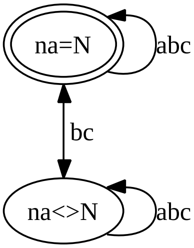
The first thing that happens is that a generates a nonce n_a. This consists of valuation expansion followed by generating the value. The effect of valuation expansion:



Private generation of value N for the new register is represented by the following action model:



Updating with this gives a situation where only a knows the value of the nonce:



Cognitive Scientist: The only difference between the last two pictures is that in the first I see boxes and in the second ovals. Can anyone explain, please?

Computer Scientist: The ovals represent worlds in a Kripke model and the boxes represent actions that take place and that transform Kripke models. This is called “action update”. It was invented by Baltag, Moss and Solecki [1]. Action update is a product operation: Worlds in the updated Kripke model are pairs consisting of an old world and an action. Arrows in the updated Kripke model relate pairs where both the world component and the action component were related by the same arrow.

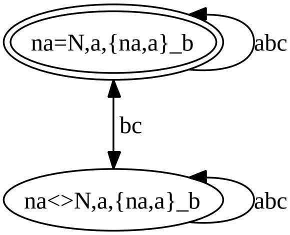
Cognitive Scientist: And I suppose the actual worlds after the updates are those pairs of worlds and actions where the world was an actual world before the update and where the action was an actual action. For the double boxes indicate the actual actions, don’t they?

Computer Scientist: Yes, you got it.

Security Analyst: Now how about the action of sending the nonce to b? There is also the issue of encryption with the public key of b. How should we represent that? And a is also putting her own name inside the message.

Logician: First, we need to create appropriate registers.

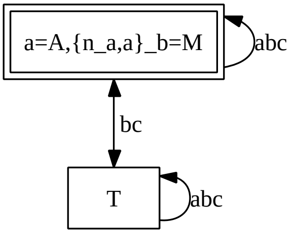
Computer Scientist: Register expansion again. Can we agree to use the obvious conventions for naming the registers? Then registers a and $\{n_a, a\}_{\mathrm{PK}_b}$ need to be created. Here is the effect of valuation expansion:



Logician: Let us suppose that the actual value of a equals A, and the actual value of $\{n_a, a\}_{\mathrm{PK}_b}$ equals M.

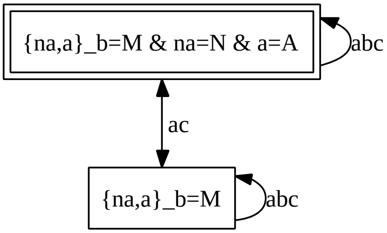
Cognitive Scientist: What do you mean by the value of a?

Logician: Think of A as the number that represents the name of a, maybe the encoding of a's name in ASCII. It is just a number that everyone recognizes as the name of a. Agent a decides to use her name A to sign a message. This means that a can distinguish the true value of the register a from other possible values, and the other agents cannot. And similarly for M. M is the number that results from encoding the pair consisting of the nonce number N and the name A with b's public key. One can say that M equals the number $\{N, A\}_{\mathrm{PK}_b}$, where $\{\cdot\}_{\mathrm{PK}_b}$ now stands for computing with the public key encryption function for b. Again, since a generated this, she knows about it. Here is the update model for this:



Cognitive Scientist: I see. So this expresses that a generates a message for the pair consisting of n_a and her own name, encrypted in the public key of b.

Computer Scientist: Now notice that M means something for b, since it is supposed to be a result of encryption in b's public key. But to the others M means nothing. So the act of making the encrypted message public can be neatly encoded in an update action, as follows:



What this says is that b is the only agent who uses the number M in register $\{n_a, a\}_{\mathrm{PK}_b}$ to find the correct register contents for n_a and a, namely N and A.

Logician: Yes, that's right. In the action model, the actual action provides the link between the encoding M and the plain text that it encodes, but other agents (in our example, a and c) confuse this with the update where nothing happens.

Computer Scientist: Note that the action model could be decomposed into an action model for private communication of the implication

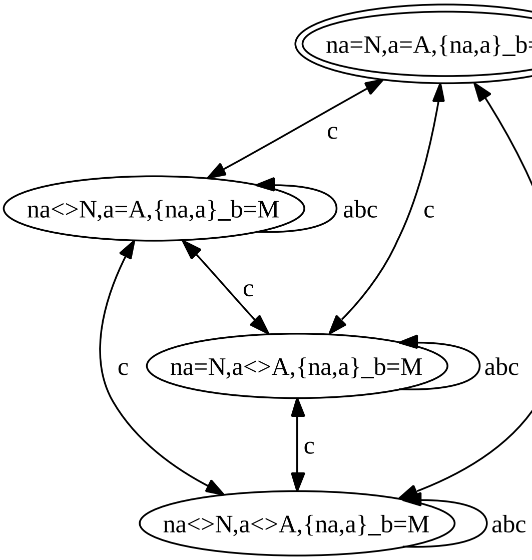
$$\{n_a, a\}_{\mathrm{PK}_b} = M \Rightarrow n_a = N \wedge a = A$$

to b, plus a public announcement of $\{n_a, a\}_{\mathrm{PK}_b} = M$.

Logician: That's right. What matters is that b is the only agent that can combine the two actions and derive $n_a = N \wedge a = A$ by modus ponens.

Cognitive Scientist: Which means that the others do not get the message. In the case of a this makes no difference, as the message originates with her. But the point is that c will not get informed.

Logician: Indeed. Now look at the result of updating the previous Kripke model with these two action models:



Philosopher: But wait a minute. It is not clear to me yet how to read this picture. Doesn't the fact that $\{n_a, a\}_{\mathrm{PK}_b} = M$ is true in all worlds imply that c knows that $\{n_a, a\}_{\mathrm{PK}_b}$ equals M, in other words that M is the result of applying b's public key to the values of n_a and a?

Computer Scientist: That would be a mistake. Please recall that the register naming scheme is just a convenience. To c, M is just a number, and $\{n_a, a\}_{\mathrm{PK}_b}$ is just a register to store this number. If c could use M for computing A and N then the c-indistinguishability arrows would be absent from the picture. Having the number M stored somewhere does not help c at all.

Philosopher: Ah, now I see.

Computer Scientist: I hope it is clear now how this should go on, at least in principle. By the way, calculating these updates by hand is madness. Fortunately there is an implementation of a powerful dynamic update logic: the version described in [3]. It is called DEMO [8].

Logician: Yes, I have heard of this. It is an epistemic model checker, right? It has been used for checking the so-called dining cryptographers protocol [7]. It would be useful to extend this into a tool for a wider range of protocols.

Computer Scientist: I like your idea of the condensed worlds very much. However, the model is still essentially infinite. If my understanding of the possible worlds semantics for knowledge is correct, you are forced to represent all possible ways of making $n = N$ false to represent the agent's ignorance that the value of n equals N. What this means is that you need to represent all possibilities $n = M$ where $M \neq N$, since there is no particular value for M that is of special interest.

Security Analyst: Actually, I don't think you need to talk about those possible values for things like nonces, under some strong assumptions about the cryptographic primitives. Suppose we assume that the agents can't make effective guesses about the value of a nonce, then either they know the value or they don't. You can reformulate the relevant part, whether an agent *knows* a value, or stated differently (non-epistemically!): whether you *possess* some piece of information or not. After yesterday's discussion, I remembered I saw a nice way of modeling this in a paper by Ramanujam and Suresh [12]. This is in the context of some temporal logic, and I think it was inspired by Paulson's work [10].

Logician: OK, I see the point. So an agent either "has" a nonce, or he doesn't have it. That leaves the actual value totally implicit. And "a has nonce n_a" is actually a proposition, without any epistemic operators…

Security Analyst: Yes, that's what I mean! Formally, we can build these propositions using a predicate on agents a and messages m: $a \cdot \mathbf{has} \cdot m$. On top of the propositions $a \cdot \mathbf{has} \cdot m$, we can build up a full dynamic epistemic logic, with the possible world semantics as usual. For example, we can express "a knows that b has the key k" with the formula $K_a(b \cdot \mathbf{has} \cdot k)$ in our language.

Logician: Ah, now it's getting very interesting! There is a problem expressing this using epistemic operators and values. If we formalize "a knows b knows k" as $K_a K_b(k = N)$, this would necessarily imply also that $K_a(k = N)$ in the classical setting. But this is very problematic! For agent a is assumed to know that agent b knows the value of his own private key, and this is common knowledge, but agent a does *not* know the value of b's private key. That is the essential feature of this type of encryption.

Philosopher: Doesn't this have to do with the *de dicto* — *de re* distinction? I would say a better way to express that "a knows that b knows the value of k" would also involve a quantification: $K_a(\exists N K_b(k = N))$, as opposed to $\exists N K_a K_b(k = N)$. The disadvantage of such quantification is that it makes the modeling even more complicated... I like this idea of separating "real" knowledge –knowledge of facts– from "possession of bits of information." By the way, this discussion reminds me of Plaza's formalization of agents knowing the value of the two secret numbers in the sum-and-product puzzle [11]. Let's continue in this direction!

Logician: Intuitively, I would say indeed that $K_a(b \cdot \mathbf{has} \cdot k)$ should not imply $a \cdot \mathbf{has} \cdot k$. But how does the evaluation of the propositions of the form $a \cdot \mathbf{has} \cdot m$ work formally? In possible worlds semantics, every world comes with a valuation for the basic propositions. So I guess we need to extend these valuations in some way?

Security Analyst: Yes, we could do so by assigning in each world, to each agent i, a set of messages, to which we will refer as a's *information set* in that world. The elements of this set represent the keys he possesses (like his private key), the nonces he generated, and the messages he received. The proposition $a \cdot \mathbf{has} \cdot m$ is then defined to be true in that world, if either m is in i's information set, or m can be constructed by a from the elements in his information set.

Computer Scientist: You could think of the messages as terms generated as follows: *(writes on the whiteboard)*

$$m ::= a \quad | \quad n \quad | \quad k \quad | \quad \{m\}_k \quad | \quad (m, m')$$

Philosopher: I guess $\{m\}_k$ stands for a message m encrypted with key k. But what is (m, m') supposed to mean?

Security Analyst: Just pairing of messages. Now, for example, if you have some message m and a key k in your information set, you should be able to construct $\{m\}_k$. So we have some rules determining which messages an agent can construct from his information set: *(writes on the whiteboard again)*

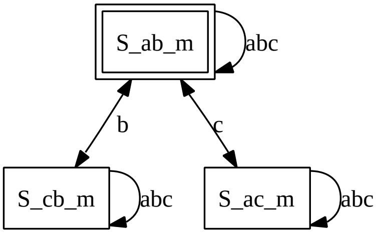
$$\frac{\{m\}_k \quad \overline{k}}{m} \qquad \frac{m \quad m'}{(m,m')} \qquad \frac{(m,m')}{m} \qquad \frac{(m,m')}{m'} \qquad \frac{m \quad k}{\{m\}_k}$$

Philosopher: So now we can say that $a \cdot \mathbf{has} \cdot m$ is true if the message m is in the closure of a's information set under these rules... Quite nice!

Logician: (to himself) And then it can easily be the case that b actually has the key k in all worlds a considers possible, but a still doesn't possess k herself. So, indeed, $K_a(b \cdot \mathbf{has} \cdot k)$ does not imply $(a \cdot \mathbf{has} \cdot k)$.

Cognitive Scientist: That sounds reasonable, but how do you model the communication between agents? I mean, the actions in the protocol are all communicative actions.

Logician: For this we can use the action models again. Let me give you an example of the action model to get a flavor. Suppose there are three agents a, b, c, and c is the special name for an intruder. The action model "a sends b the message m" would be like *(draws on the whiteboard)*:



Computer Scientist: Let me see... So the actual action is "a sends m to b successfully", indicated by *S_ab_m* in the picture. But b is not sure whether he received it from a or from the intruder c, and a is not sure whether the message she sent got intercepted by c or not.

Logician: That's exactly right.

Security Analyst: I guess the precondition for each action is that the sender "has" the message m?

Logician: That's right. Moreover, we have a "postcondition" for each action as well. For example, the postcondition for *S_ab_m* should be that b learns

m but both a and c learn nothing. We just need to update the information set of b in the worlds that satisfy the precondition of S_ab_m, by adding the message m.

Security Analyst: OK, I can sense the general direction. Still, these action models seem rather ad hoc to me.

Logician: Yes, I agree that there is still a lot that we need to make clear. For example: what is it exactly that the agents observe when communication takes place? This depends on assumptions we make about the channels, whether it is observable that messages are passed among agents, and between whom. Such things are crucial for building action models.

Philosopher: I am beginning to wonder whether this Needham-Schroeder protocol is the best test case for dynamic epistemic logic. Could anyone come up with a more convincing example maybe?

Logician: Of course, there is the famous muddy children example [2]. I guess you all know that?

Security Analyst: Is that a protocol? I don't think I know it...

Logician: Here's how it goes: Among n children, there are k (which is at least one) of them with mud on their foreheads. They can see each other but not themselves. Now their father confronts them and says aloud: 'At least one of you has mud on his forehead. Will all the children who know they have mud on their heads please step forward?' First, none of the children step forward. When the father repeats his question, he will still get no response until he asks the question for the k-th time. Then, miraculously, all muddy children step forward.

Security Analyst: Hey, I did know this problem, but I know it as the *unfaithful wives problem.*

Philosopher: That must be the politically incorrect version.

Logician: If you care for political incorrectness you should also look at the *unfaithful husbands* variation [9]. Well, the reasoning always amounts to the same, and nowadays everyone knows it. I suppose I could convince all of you that a dynamic epistemic analysis can be used nicely to explain what is going on. There are similar problems, for example "Product and Sum" and "Russian cards", that have also been analyzed using these logical techniques [6; 5].

Computer Scientist: Ahem, to me these examples don't sound like protocols at all. They don't *prescribe* actions to fulfill a certain goal. Instead, they seem to *describe*, or explain if you wish, how smart logically thinking agents could solve puzzles about knowledge and ignorance.

Cognitive Scientist: And didn't we decide that protocol analysis was about checking whether some given requirements are fulfilled after each possible run of the protocol? In the above cases, I don't see directly what the requirements are. And how would you check them, in practice?

Security Analyst: A possibly better example that comes to my mind is the so-called "Dining Cryptographers Protocol" [4], which is a way of doing an anonymous broadcast. Three cryptographers are dining out and at the end of the evening they are informed that their bill has been paid. Moreover, they know that either one of the cryptographers has paid for the dinner, or otherwise the National Security Agency (NSA) has. The cryptographers want to achieve common knowledge on whether it was the NSA that paid or one of them, in the latter case without revealing which individual footed the bill. By flipping coins and announcing bits, this can be achieved. An epistemic analysis is in [7] and in the chapter on 'Eating from the Tree of Ignorance' (page **??**).

Computer Scientist: However, this still sounds like an epistemic puzzle to me: Initially, the agents have some uncertainties about the facts, but the facts themselves are already established. Through making announcements following a certain pattern and in accordance with the epistemic states of the agents, the agents get to know the desired facts. In terms of the protocol, the communicative actions have epistemic preconditions and the requirements to be fulfilled after the protocol are also purely epistemic.

Security Analyst: Yes, you have a point. Moreover, there is a crucial element in security protocol analysis that is missing in these puzzles.

Computer Scientist: Let me guess: These puzzles don't really have *runs*.

Security Analyst: Exactly! In these puzzles, there is usually one (and only one) sequence of actions and it leads to the desired outcome. In that sense, I would say they are not really protocols as we usually understand them. There are no intruders or compromised players. In the analysis of security protocols on the other hand, we consider all possible sequences of actions, which could also result from interleavings of several instantiations of the same protocol. Think for example of Lowe's attack on the Needham-Schroeder protocol, where two instantiations of the protocol were smartly connected by the intruder.

Computer Scientist: I do not see yet how we can generate all possible runs in this dynamic epistemic framework.

Cognitive Scientist: In those puzzles, you have assumed implicitly that the agents all attended a course on epistemic logic and they can reason with this in a perfect way, and even that this assumption itself is common knowledge among the participants. Such assumptions may be too strong for protocol analysis in general. You shouldn't rely on the reasoning power of agents unless the preconditions of the actions require some kind of epistemic reasoning.

Philosopher: And you assume that not only the protocol but also the epistemic reasoning are common knowledge among agents.

Logician: Ahem, these are very useful insights, thanks a lot! Anyway, you never get anywhere if you don't start somewhere. It would be a perfect starting point if we can find a real protocol not only *about* knowledge but also having epistemic preconditions for actions. A good indication of the epistemic nature of a protocol could be when the requirements to be achieved by the protocol involve nested modal knowledge operators. I feel that the strengths of epistemic logic would really come to the fore in such cases.

Computer Scientist: You will have to actively look around to find such protocols. Standard protocols like Needham-Schroeder definitely don't seem to fit in the category.

Security Analyst: Yes, it may not be easy to find such protocols in the practice of computer communications. With these subjective perspectives, they look too complicated.

Philosopher: Still, there might be communication scenarios between humans that require this kind of analysis, maybe? Indeed, why don't you try to find real-life scenarios and make up the protocols that fit yourselves?

Logician: Good idea! Would you guys give me some suggestions?

Philosopher: Did you go to Wouter Teepe's talk in this workshop? He talked about this funny scenario that might be interesting to you.

Cognitive Scientist: All I remember is that Wouter talked about gossiping.

Always interesting, I suppose...

Philosopher: His example story went like this. Geertje tells her friend Wouter that she is pregnant. A few days later, Wouter meets the secretary at the coffee corner. The secretary looks expectantly at Wouter. It seems she wants to gossip with Wouter about something, perhaps Geertje's pregnancy. However, as a good friend of Geertje, Wouter promised her not to disclose the secret. So Wouter can start gossiping about Geertje's pregnancy only if he is sure that the secretary also knows the secret. The question is: Is there a protocol that will allow Wouter to find out whether he can safely start his gossip?

Logician: That looks promising. Let us try to list the requirements *(writes on the whiteboard)*:

- After the protocol execution, Wouter knows whether the secretary has the secret.
- If the secretary did not have secret, neither does she after the protocol execution.
- If the secretary has the secret, she knows Wouter has it too after the protocol execution.

Security Analyst: Maybe we should also require that no one else learns the secret and no one else learns whether Wouter and the secretary share a secret.

Logician: We can formalize such requirements in a straightforward way in our epistemic language. And I can see that the actions of the protocol must have some sort of epistemic preconditions since the secretary should respond to Wouter according to the information she has.

Computer Scientist: Yes, actually Wouter himself gave several protocols of such scenarios in his thesis [13].

Security Analyst: And I think they are real protocols which are quite useful in the cases when you need to compare information without leaking it.

Computer Scientist: In fact, maybe I should have spoken up earlier, but in my view there is one type of logics that is very suitable for protocol analysis, but which we hardly discussed: temporal logics! Model checking of those logics is very well developed in computer science. There are several mature model checking tools around, and also some standards and languages for the modeling of protocols. Maybe a framework combining the best of both worlds is also worth investigating.

Logician: Yes, there are many promising directions. Anyway, the application of dynamic epistemic logic in protocol analysis deserves to be pursued. I am quite confident it will turn out to be useful at least for some of the cases.

Cognitive Scientist: That sounds hopeful. Unfortunately I need to dash now to catch my train.

Security Analyst: May I join you? *(The security analyst and the cognitive scientist amble off together towards the bus stop.)*

Computer Scientist: I hope they won't gossip about us . . .

Logician: Let them gossip. *Looking smugly at the philosopher.* At least *I* do not have anything to hide.

Philosopher: I already gathered that you frown upon my behavior at yesterday's NIAS dinner. Ah, the wonderful Beaujolais they served! I must admit that I do not remember much of what transpired. Well, as long as our two colleagues use that fellow Wouter Teepe's protocol for their gossiping, they will not learn anything new about *me*, either.

Computer Scientist: Indeed, ignorance is bliss [1]. *(Looks dreamily into the distance.)*

[1] The theme of ignorance as bliss is developed further in the chapter 'Eating from the tree of ignorance', starting on page **??**.

References

[1] A. Baltag, L.S. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. Technical Report SEN-R9922, CWI, Amsterdam, 1999.

[2] J. Barwise. Scenes and other situations. *The Journal of Philosophy*, 78:369–397, 1981.

[3] J. van Benthem, J. van Eijck, and B. Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, 2006.

[4] D. Chaum. The dining cryptographers problem: unconditional sender and receiver untraceability. *Journal of Cryptology*, 1:65–75, 1988.

[5] Hans van Ditmarsch, Wiebe van der Hoek, Ron van der Meyden, and Ji Ruan. Model checking Russian cards. *Electronic Notes Theoretical Computer Science*, 149(2):105–123, 2006.

[6] H.P. van Ditmarsch, J. Ruan, and R. Verbrugge. Sum and product in dynamic epistemic logic. *Journal of Logic and Computation*, 18:563–588, 2008.

[7] J. van Eijck and S. Orzan. Modelling the epistemics of communication with functional programming. In Marko van Eekelen, editor, *Sixth Symposium on Trends in Functional Programming TFP 2005*, pages 44–59, Tallinn, 2005. Institute of Cybernetics, Tallinn Technical University.

[8] Jan van Eijck. DEMO — a demo of epistemic modelling. In Johan van Benthem, Dov Gabbay, and Benedikt Löwe, editors, *Interactive Logic — Proceedings of the 7th Augustus de Morgan Workshop*, number 1 in Texts in Logic and Games, pages 305–363. Amsterdam University Press, 2007.

[9] Yoram Moses, Danny Dolev, and Joseph Y. Halpern. Cheating husbands and other stories: A case study of knowledge, action, and communication. *Distributed Computing*, 1(3):167–176, September 1986.

[10] Lawrence C. Paulson. Proving properties of security protocols by induction. In *10th Computer Security Foundations Workshop*, pages 70–83. IEEE Computer Society Press, 1997.

[11] J. A. Plaza. Logics of public communications. In M. L. Emrich, M. S. Pfeifer, M. Hadzikadic, and Z. W. Ras, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216, 1989.

[12] R. Ramanujam and S. P. Suresh. Deciding knowledge properties of security protocols. In *TARK '05: Proceedings of the 10th conference on Theoretical aspects of rationality and knowledge*, pages 219–235, Singapore, Singapore, 2005. National University of Singapore.

[13] Wouter Teepe. *Reconciling Information Exchange and Confidentiality — A Formal Approach.* PhD thesis, Rijksuniversiteit Groningen, 2007.