Formal Reasoning About Trusted Third Party Protocols

Aaron Hunter
BC Institute of Technology, 3700 Willingdon Avenue, Burnaby, Canada

Keywords: Trust, Belief Revision, Formal Verification.

Abstract: A trusted third party (TTP) is an entity that facilitates communication between agents by acting as an intermediary. Typical roles for a trusted third party include the establishment of session keys or the validation of commitment schemes. In a formal setting, this requires a model that provides some mechanism for representing trust and reasoning about dynamic beliefs. In this paper, we demonstrate how this can be captured using a combined modal logic of trust and belief. Our formalism uses plausibility models and model transformations to capture belief revision in a protocol run. It is novel in that it uses the modal accessibility relations in the logic to define a notion of trust, without requiring any additional formal machinery. We define the formal semantics of the logic, sketch the axiomatization, and demonstrate the basic verification methodology. Challenges are discussed, as well as issues related to practical deployment.

1 INTRODUCTION

In network communication, a trusted third party (TTP) is an agent that participates in a protocol to ensure that the information exchanged is correct (Zissis et al., 2011). In principle, there is no way to guarantee that a TTP acts in the interest of either party in the protocol; they must be trusted to act in a manner that is satisfactory to the other participants. Having a TTP participate in a protocol is one way to give the other participants confidence in the information when they are not able to trust each other. In practice, this kind of protocol can be implemented through mechanisms such as the so-called Web of Trust (Ulrich et al., 2011). However, provable guarantees of security are difficult to achieve in the general case.

In this paper, we argue that TTP protocols can be effectively analyzed and verified using dynamic logics of trust and belief. We note that formal logics of belief have a long history in protocol verification. However, formal logics of belief have rarely incorporated a precise notion of trust. Obviously this is an essential aspect to consider when modelling and reasoning about TTP protocols. So our approach in this paper is to define a new modal logic of knowledge and belief that captures the trust that each agent holds in the others. This logic is based on comparatively recent formal work on trust-sensitive belief revision, where the extent to which an agent can be trusted is an explicit component of the logic. We propose that this new logic can be used to prove precisely when an honest TTP can be a useful intermediary.

This is a preliminary paper introducing a novel approach to reasoning about trusted third parties in communication protocols. As such, the focus is on developing the basic logic and we leave the details of deployment on practical protocols for future work. Nevertheless, this work makes several contributions to the literature. First, the paper shows how trust can be captured in a standard dynamic epistemic logic, without introducing any new formal machinery. Second, the proposed logic explicitly captures the trust held in a TTP with respect to the beliefs of the agents participating in the protocol This allows the logic to be used for protocol verification by simply checking a modal entailment. We remark also that, while our focus here is on communication protocols, the logic is sufficiently general to capture mutual trust in other settings, such as social network communication.

2 PRELIMINARIES

2.1 Motivating Example

To facilitate the discussion, we describe a simple protocol. The protocol involves the exchange of messages between three parties: A, B and T. In this protocol, T is acting as a TTP to allow A and B to establish a session key. We use the standard notation

Hunter, A.
Formal Reasoning About Trusted Third Party Protocols.
DOI: 10.5220/0013237600003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 3*, pages 926-932
ISBN: 978-989-758-737-5; ISSN: 2184-433X

established in (Burrows et al., 1990) to describe the protocol:

Simple Key Agreement
1. $A \rightarrow B : N_A, A$
2. $B \rightarrow T : N_A, N_B, A, B$
3. $T \rightarrow A : \{K\}_{K_A}$
4. $T \rightarrow B : \{K\}_{K_B}$

In this notation, $A \rightarrow B : M$ means that A sends the message M to the agent B. An expression of the form $\{M\}_K$ denotes the message M encrypted with the key K. Messages of the form N_A are *nonces*, which are random numbers generated at the time of protocol execution. In this protocol, T is a trusted party that is responsible distributing sessions keys for communication between agents. We assume that T shares a secret key with A which is denoted by K_A, as well as a secret key with B which is denoted by K_B. The goal of this protocol is to give A and B a new key that they can use for secure communication. This protocol is a simplification of a protocol previously presented in (Perrig et al., 2001).

Proving that this kind of protocol actually works can be difficult. There are at least two challenges. The first is a question of honesty: A needs to trust that T is going to give them a key that is secure and not available to other parties. As noted in the introduction, we are not going to be concerned with this issue; we are going to assume that T has no malicious intent. But the second problem is a problem of knowledge. Why should A and B believe that T has a suitable collection of keys available, which are each secure? This problem requires an analysis of the beliefs of A and B, and how they change when information is exchanged on the network.

2.2 Logics for Protocol Verification

Logics of knowledge and belief have a long history as tools for proving the security of protocols. This approach was introduced in the pioneering work on BAN logic, in which a simple model of knowledge is used for the analysis of authentication protocols (Burrows et al., 1990). This basic logic has been explanded and modified to address different kinds of protocols. Essentially, all variations of BAN logic allow us to express both the information exchanged in a protocol and the goal of the protocol as logical formulas. In this manner, we are able to preciesly prove when the goal of a protocol is true following a successful run. We generally need to use a logic of belief, because authentication protocols are fundamentally concerned with the beliefs of the participating agents.

While the original BAN logic was never sophisticated enough to address real protocols, the logical tradition in protocol analysis continues. Logics of knowledge and belief have been used in recent years for the analysis of IOT protocols (Hofer-Schmitz and Stojanović, 2020), for smart-home protocols (Fakroon et al., 2020), and for health-record protocols (Kim et al., 2020). Similar logical methods have also been employed for the verification of smart contracts (Tolmach et al., 2021). For a recent survey on the use of formal methods for protocol verification, we refer the reader to (Erata et al., 2023).

When logical methods are used for protocol verification, we generally start with some established logic from the AI community and then we modify it suitably to capture all aspects of some class of protocols. However, we will see in the next two sections that logical models of belief dynamics are generally focused entirely on how to model the beliefs of an agent when new information is received as some kind of infallible announcement. When a protocol involves agents that have different levels of trust in each other, then we need more more complex logic that captures this fact.

2.3 Dynamic Epistemic Logic

The problem of reasoning about nested beliefs can be addressed in Dynamic Epistemic Logic (DEL). We briefly introduce DEL in this section. However, we assume the reader is familiar with basic modal logic, as described in (Chellas, 1980).

We start with an underlying propositional signature P, which is just a set of atomic sentence symbols that can be true or false. The variables in P represent properties of the world that may be true or false. A propositional formula is defined in the usual manner, by using connectives like $\wedge$ (and), $\vee$ (or), and $\neg$ (not).

Standard modal logics of belief use a static modal operator to represent the beliefs of an agent. In other words, they use formulas like $B_i\varphi$ to mean "agent i believes φ is true." DEL extends standard epistemic logic by adding dynamic operators of the form $[\varphi]\psi$; this means roughly that ψ is true following the announcement of φ. There are many variations on this logic for different kinds of announcements, and we refer the reader to (van Bentham, 2014) for a full discussion.

We are concerned with reasoning about belief dynamics in DEL. This requires a notion of plausbility. For this reason, belief revision in DEL is typically captured semantically through a *plausibility model* (van Bentham, 2007).

Definition 1. *A* plausibility model $M = \langle W, \{\leq_i\}_{i\in I}, V\rangle$ *consists of a set of worlds W, a well-ordering $\leq_i$ over W for each $i \in I$, and a valuation V.*

Informally, the ordering $\leq_i$ is a plausibility ordering for the agent i; the minimal elements of the ordering are considered the most plausible. Note that $\{\leq_i\}_{i\in I}$ defines an accessibility relation on the set of worlds for each agent i. In particular, write $w \sim_i j$ as a shorthand for the statement that w is comparable to v: either $w \leq_i v$ or $v \leq_i w$. Then $\sim_i$ defines a $KD45$ accessibility relation, familiar from standard doxastic logic.

2.4 Belief Revision in Dynamic Epistemic Logic

Belief revision is the process that occurs when an agent receives some new information about the world. The new information is understood to be more accurate, therefore the agent would like to believe the new information while giving up as little as possible from their initial beliefs. This process has been studied extensively. The most widely studied formal approach for single-shot belief revision is the so-called AGM approach (Alchourrón et al., 1985). For iterated belief change, the Darwiche-Pearl approach (Darwiche and Pearl, 1997) is the natural generalization. However, in both of those cases, we can not consider nested beliefs or beliefs about trust relationships; these are both important concepts for reasoning about trusted third parties. For this reason, we focus here on belief revision in the context of DEL.

Belief revision can be captured in DEL through tranformations on models. In the simplest case, we revise by a formula φ. For each formula φ and each model M, we define a new model M' where the plausibility ordering is modified according to some reasonable revision policy. Each revision policy can then be defined in terms of a dynamic epistemic modality. We illustrate with a simple example.

Example 1. *Suppose that the propositional signature is $\{p,q\}$, there is just a single agent, and that there is one possible world for each interpretation. The $\Uparrow$ modality will be defined such that the truth of $\Uparrow[p\wedge\neg q]\psi$ will be true at any state where ψ would be true following revision by $p\wedge\neg q$. This is checked by checking the truth of ψ in the model $M*$ that is defined by performing lexicographic update of the plausibility ordering by ψ.*

One important iterated revision operator is the so-called lexicographic revision operator (Nayak et al., 2003). In lexicographic revision by φ, the ordering is shifted so that every φ-state precedes every $\neg\varphi$-state. Within those regions, the ordering on states is left unchanged.

Given a plausiblity model M and a formula φ, let $M \Uparrow \varphi$ denote the model obtained from M by simply modifying all of the orderings according to the lexicographic revision policy. We can then introduce a lexicographic update modality $[\Uparrow \varphi]$ into the vocabulary such that

$$M,s \models [\Uparrow \varphi]\psi \iff M \Uparrow \varphi, s \models \psi.$$

The logic includes several modalities:

- $K_i\varphi$: Agent i *knows* φ.
- $B_i^{\varphi}\psi$: Agent i would believe ψ if they were given φ.
- $[\Uparrow \varphi]$: The dynamic modality for update by φ.

This logic can be axiomatized completely for lexicographic update, as well as other well-known revision policies (van Bentham, 2014).

3 LOGICAL FRAMEWORK

3.1 Motivation

We have seen that there are well-established logics for reasoning about belief revision. However, these logics simply show how an agent can revise by a *formula*. We need a logic that allows an agent to revise by a *report*, which is a formula that has been received from some other agent.

We follow the basic approach outlined in (Hunter and Booth, 2019). That is, we start with the logic of belief revision introduced in the previous section, and we extend it with additional modal operators for trust. Our goal is to be able to express statements of the following form:

- Agent i would believe agent j if they said the formula φ.

While the details in the next sections are quite formal, the end result is just a natural logic that allows us to express this sentiment.

3.2 Trust-Sensitive Plausibility Models

For each pair of agents i and j, we introduce a new modal operator TR_i^j with the following property:

- $s \models TR_i^j\varphi$ if and only if agent i would trust agent j if they reported φ in the state s Agent i would believe agent j if they said the formula φ.

This operator can be defined by introducing some new binary relations on the set of states. However, we instead define the relation with respect to the realtions $\sim_i$ and $\sim_j$ that are already in the logic. We informally say that i trusts j to be able to distinguish the

states w adn v just in case i *believes* that j can distinguish them. This assertion can be made using the $\sim$ operators For each pair of agents i, j, define T_i^j as follows:

$$T_i^j wv \iff \text{there exist } x \text{ such that } w \sim_i x \text{ and } x \sim_j v. \quad (1)$$

We remark that this is clearly an equivalence relation. As such, we can use T_i^j to define a notion of trust-sensitive revision by following the approach in (Booth and Hunter, 2018).

Let $\Uparrow$ be the order transformation that defines lexicographic revision with respect to a particular plausibility ordering. As stated previously, $\Uparrow$ defines a transformation on plausibility models in which the plausibility orderings are re-arranged according to the lexicographic revision policy. We need to modify this order transformation to produce a new order transformation $\Uparrow_j^T$ for each agent j. The agent j here is understood to represent the *reporting agent*; this is the agent that has provided some new piece of information. In the parametrized transformation, each agent will revise their beliefs so that they only believe the formulas over which the reporting agent is trusted.

Definition 2. *Let $M = \langle W, \{\leq_i\}_{i\in I}, V\rangle$ be a plausibility model, let φ be a formula, and let $j \in I$. Define $M \Uparrow_j^T \varphi$ to be the model with the same set of worlds and valuations, but where the orderings $\leq_i^\varphi$ are defined as follows:*

$$\leq_i^\varphi = \leq_i \Uparrow \{v \mid M, w \models \varphi \text{ and } T_i^j wv\}$$

where $\Uparrow$ is the lexicographic order transformation.

We now have a new mapping $M \Uparrow_j^T \varphi$ parametrized by an agent j, in which each ordering is revised differently. The transformation associated with $\Uparrow_j^T \varphi$ captures the way that each agent i revises their plausibility ordering when the formula φ is reported by the agent j. As such, rather than simply revising by φ, each agent now revises by the set of states that j can not distinguish from models of φ according to the trust relation T_i^j.

3.3 A Logic for Trust-Sensitive Revision

In the previous section, we introduced a special order transformation on plausibility models that takes trust into consideration.

A complete axiomatization of the logic $\mathcal{L}(\Uparrow)$ is given in (van Bentham, 2014). The axiomatization includes axioms for modelling static conditional beliefs, along with the following five axioms for $[\Uparrow \varphi]$:

$[\Uparrow \varphi]q \leftrightarrow q$, for all atomic propositional q
$[\Uparrow \varphi]\neg\psi \leftrightarrow \neg[\Uparrow \varphi]\psi$
$[\Uparrow \varphi]\psi \wedge \alpha \leftrightarrow [\Uparrow \varphi]\psi \wedge [\Uparrow \varphi]\alpha$
$[\Uparrow \varphi]K_i\varphi \leftrightarrow K_i[\Uparrow \varphi]\varphi$
$[\Uparrow \varphi]B_i^\alpha\psi \leftrightarrow (\tilde{K}_i(\varphi \wedge [\Uparrow \varphi]\alpha) \wedge B_i^{\alpha\wedge[\Uparrow\varphi]\alpha}[\Uparrow \varphi]\psi) \vee$
$(\neg\tilde{K}_i(\varphi \wedge [\Uparrow \varphi]\alpha) \wedge B_i^{[\Uparrow\varphi]\alpha}[\Uparrow \varphi]\psi)$

We can define an extension $\mathcal{L}(\Uparrow)^T$ for trust-sensitive revision, by introducing the following modalities:

- $K_i\varphi$: Agent i *knows* φ.
- $B_i^\varphi\psi$: Agent i would believe ψ if they were given φ.
- $TR_i^j\psi$: Agent i trusts agent j when reporting ψ.
- $[\Uparrow_j \varphi]$: The dynamic modality parametrized by j .

The axioms of $\mathcal{L}(\Uparrow)^T$ include:

1. The standard $S5$ axiomatization for static knowledge for each modality K_i.
2. The standard $S5$ axiomatization for static knowledge for each TR_i^j.
3. An axiomatization of static conditional belief for each modality B_i^φ, as given in (van Bentham, 2014).

As described in (Booth and Hunter, 2018), we can modify the five axioms for $[\Uparrow \varphi]$ to define each $[\Uparrow_j \varphi]$. Moreover, the following condition is guaranteed:

$$M, s \models [\Uparrow_j \varphi]\psi \iff M \Uparrow_j^T \varphi, s \models \psi.$$

Hence $[\Uparrow_j \varphi]\psi$ is true at a state s, just in case ψ would be true if each agent's plausibility ordering is re-ordered by lexicographic update, with respect to the set of states that j can not distinguish from models of φ. It is straightfoward to specify the axiomatization.

3.4 Lying

As an illustration, we describe how the logic in question deals with agents who are lying. The notion of lying is discussed in detail in the context of DEL in (van Ditmarsch, 2014). However, our approach to lying here is simpler, as we do not introduce any new formal machinery to describe lying announcements.

We acknowledge first that an agent in our framework will not be able to determine when a trusted agent is being deceptive. If an agent i trusts j on the formula φ, then they will certainly believe φ when it is reported by j. So if j falsely reports φ when they are trusted as an authority on φ, we have a problem. In order to address this, we would need to add some mechanism for modelling *trust change*. Such a mechanism is introduced in the contxnt of belief revision in (Hunter, 2024). However, trust change is beyond the scope of our current discussion.

There is, however, a case of lying that our framework can address. Consider an agent i that receives a report φ from another agent when i actually *knows* that φ is false. Clearly, in this case, we would like to be certain that i will not believe the new information. We state a relevant result with respect to knowledge.

Proposition 1. *Let M be a plausibility model. For any agents i, j, if $M^*, s \models K_j \neg\varphi$ and $M^*, s \models K_i K_j \neg\varphi$ then*

$$M^*, s \not\models [\Uparrow_j \varphi] \neg K_i \varphi.$$

Hence, a deceptive report will not be incorporated when we define trust with respect to perceived knowledge. In fact, the plausibility ordering for each agent that knows j is being deceptive will be unchanged. The important point here is that the framework automatically ensure that deceptive reports will not be believed.

4 USING TRUSTED THIRD PARTIES

4.1 Establishing Trust

For the moment, we put aside the manner in which a TTP might be established. In practice, this might be done through extra-logical means. In other words, it might be the case that a certain agent is created under the joint supervision of all participating parites; that agent is then trusted, but the justification can not be established in the logic.

Another possibility would be to establish the TTP within the logic. This would require a logical methodology that not only models trust, but also models *trust change*. As noted previously, trust change is not something we consider in the present paper; we leave this extension for future work.

4.2 Definition

Although we are not concerned with showing how an agent becomes trusted, we can still define what it means for an agent to be a TTP.

Definition 3. *Let P be a propositional signature, and let $L \subseteq P$. We say that an agent W is a trusted third party for A, C over L for the set of models $\mathcal{M}$ if the following conditions hold for all $M \in \mathcal{M}$ and all $p \in L$:*

1. $M \models TR^A_W p \wedge B_A T^C_W p$.
2. $M \models TR^C_W p \wedge B_C T^A_W p$.

We let TTP denote the conjunction of these two formulas. Hence a trusted third party for L is someone that is trusted when they assert W is true, and both parties believe the other feels the same. We remark that this definition is quite simple to state, but it would be quite difficult to guarantee that these conditions are true. But combining a modal logic with trust allows this kind of condition to be stated very compactly.

4.3 Protocol Verification

To prove that a TTP protocol is correct, we simply need to encode the protocol as a set of logical formulas. Consider the simple identity exchange protocool from the introduction. In order to show that this protocol is correct, one would need to perform the following steps.

- Formalize the protocol as a sequence of announcements $P_1, \ldots, P_n$, made respectively by agents $a_1, \ldots, a_n$.
- Formalize the goal of the protocol as another formula G.
- Prove that $TTP \models [\Uparrow_{a_1}] \cdots [\Uparrow_{a_n} P_n] G$.

This is an established method for verifying simple authentication protocols, which was pioneered in (Burrows et al., 1990). For a more recent discussion of symbolic approaches to proving protocol correctness, we refer to reader to (Delaune and Hirschi, 2017). The novel aspect of our work here is that the protocol steps are encoded as the application of dynamic modalities. The logic is a more sophisticated modal logic that those typically used for protocol verification, and it permits the representation of nested beliefs, explicit trust, and belief revision.

We demonstrate by revisiting our example protocol.

Example 2. *In order to formalize the protocol at a high level, we assume the propositional vocabulary includes atomic formulas of hte form $init(x)$ for $x \in \{A, B\}$. These are formulas that are true when A (resp. B) want to initialize a communication session. We then assume that we have a finite set K of keys. For each key $k \in K$ and each pair of agents x, y we have an atomic formula of the form $\{safe(k, x, y)$. Such a formula is true when k is a safe key for communication between x and y.*

We will use lower case a, b, t for the agents below, in order to avoid ambiguity with the belief modality. The Simple Key Agreement protocol can be represented as the following announcements:

1. $M_1 = B_b(init(a))$
2. $M_2 = B_t(init(b))$
3. $M_3 = B_a(safe(k_0, a, b))$
4. $M_4 = B_b(safe(k_0, a, b))$

Note that our approach is to encode message as belief changing announcements. The goal is the following:

$$G = B_a(safe(k_0,a,b)) \wedge B_a(safe(k_0,a,b)).$$

In this case, the agent T is a trusted third party over $safe(k_0,a,b)$). With TTP defined over this single formula, in order to prove the protocol is correct, we would need to prove the following:

$$TTP \models [\Uparrow_t M_4][\Uparrow_t M_3][\Uparrow_b M_2][\Uparrow_a M_1]G.$$

It turns out that this can not actually be proved. The problem is that there is no connection between the first two messages sent and the second two messages sent; so there is no guarantee that agent a will believe the key they are given is sent from a current run of the protocol.

It is worth noting that the actual protocol proposed in (Perrig et al., 2001) actually includes message authentication codes *in the last three messages of the protocol in order to avoid this problem. For our simplified version of the protocol, however, our logic can not provide a guarantee of security.*

5 DISCUSSION

5.1 Establishing Trusted Third Parties

Note that our approach to protocol analysis does not address the problem of establishing a trusted third party. Instead, we simply start by defining what it means to be a trusted third party. Specifically, for an agent A, we basically say T is a trusted third party over some formula if they are trusted on that formula and if A believes that B also trusts them on that formula.

For simple domains where the formulas of interest make statements about keys or signatures, it is actually possible to set up a real-world scenarios where this is true. For example, in most practical situations, a certificate authority would satisfy this condition for the certificates that they control.

However, if we consider cases where the potential trusted third party may have affiliations or independent goals, then this condition is significantly harder to satisfy. Of course, this is the case in real scenarios as well; it is essentially impossible to guarantee that a trusted third party protocol works in the general case. However, in our setting, the limitation is clear. By restricting trusted third parties to a specific set of formulas, we may be able to actually prove that an agent can be a trusted third party by exchanging a suitable sequence of messages.

5.2 Future Work

This has been a largely speculative paper. The aim has simply been to show how a standard dynamic epistemic logic could define a notion of trust that was suitable formal reasoning about trusted third parties. However, there are several directions for future work.

First of all, in order for this logic to be useful, it must be applied to real protocols. In this preliminary work, we have focused on defining the formalism and applying it to a toy protocol for illustrative purposes. However, in future work, we will apply the same model to more complex protocols. At present, the best candidate protocols are those used by certificate authorities.

The second issue that must be addressed is the fact that the current methodology is not straightforward to automate. In the past, a variety of authentication logics have been defined. However, proving correctness by hand is simply not feasible for real protocols; formal verification of protocols requires power solvers like (Cremers, 2008) as well as precise methodologies for computer-aided design and analysis of protocols (Barbosa et al., 2021). Hence, we need to address how our new logic can be implemented efficiently to quickly analyze protocols. This is not a trivial task, because there really are not many exisiting implementations of modal logic. When we move to dynamic epistemic logic, the problem is worse. At a theoretical level, even basic reasoning tasks in DEL can be intractable (Charrier et al., 2019) or even undecidable (French and van Ditmarsch, 2008).

Fortunately, we do not need to implement a fully general system for reasoning in DEL. Instead, we need only worry about an efficient solver that works with the trace-based exchanges message sequences that are familiar to protocol analysis. We believe for this restricted class of problems, a suitable implementation will indeed be feasible.

6 CONCLUSION

In this paper, we have introduced a logic for reasoning about trusted third parties. The logic is a variation of dynamic epsitemic logic, where agents can reason about knowledge, belief, and announcements. We have extended the basic framework by showing how the existing plausibility orderings in DEL can be used to model knowledge-based trust. This simple extension allows us to then reason not only about trust, but also about our beliefs about trust.

The logic proposed here is flexible enough to model and perform ”by hand” analysis and verifica-

tion of simple, toy TTP protocols. However, in future work, we will look to implement the system to develop a tool that can automatically find holes in real protocols.

REFERENCES

Alchourrón, C. E., Gärdenfors, P., and Makinson, D. (1985). On the logic of theory change: Partial meet functions for contraction and revision. *Journal of Symbolic Logic*, 50(2):510–530.

Barbosa, M., Barthe, G., Bhargavan, K., Blanchet, B., Cremers, C., Liao, K., and Parno, B. (2021). SoK: Computer-aided cryptography,. In *IEEE Symposium on Security and Privacy*, pages 777–795.

Booth, R. and Hunter, A. (2018). Trust as a precursor to belief revision. *Journal of Artificial Intelligence Research*, 61:699–722.

Burrows, M., Abadi, M., and Needham, R. (1990). A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36.

Charrier, T., Pinchinat, S., and Schwarzentruber, F. (2019). Symbolic model checking of public announcement protocols. *Journal of Logic and Computation*, 29(8):1211–1249.

Chellas, B. (1980). *Modal Logic: An Introduction*. Cambridge University Press.

Cremers, C. (2008). The scyther tool: Verification, falsification, and analysis of security protocols. In *Proceedings of the 20th International Conference on Computer Aided Verification*.

Darwiche, A. and Pearl, J. (1997). On the logic of iterated belief revision. *Artificial Intelligence*, 89(1-2):1–29.

Delaune, S. and Hirschi, L. (2017). A survey of symbolic methods for establishing equivalence-based properties in cryptographic protocols. *Journal of Logical and Algebraic Methods in Programming*, 87:127–144.

Erata, F., Deng, S., Zaghloul, F., Xiong, W., Demir, O., and Szefer, J. (2023). Survey of approaches and techniques for security verification of computer systems. *Journal on Emerging Technologies in Computing Systems*, 19(1).

Fakroon, M., Alshahrani, M., Gebali, F., and Traore, I. (2020). Secure remote anonymous user authentication scheme for smart home environment. *Internet of Things*, 9:100158.

French, T. and van Ditmarsch, H. (2008). Undecidability for arbitrary public announcement logic. In *Advances in Modal Logic*, pages 23–42.

Hofer-Schmitz, K. and Stojanović, B. (2020). Towards formal verification of iot protocols: A review. *Computer Networks*, 174:107233.

Hunter, A. (2024). Combined change operators for trust and belief. In *Australasian Joint Conference on Artificial Intelligence*.

Hunter, A. and Booth, R. (2019). Implicit and explicit trust in dynamic epistemic logic. In *21st International Workshop on Trust in Agent Societies*.

Kim, M., Yu, S., Lee, J., Park, Y., and Park, Y. (2020). Design of secure protocol for cloud-assisted electronic health record system using blockchain. *Sensors*, 20(10).

Nayak, A., Pagnucco, M., and Peppas, P. (2003). Dynamic belief change operators. *Artificial Intelligence*, 146:193–228.

Perrig, A., Szewczyk, R., Wen, V., Culler, D., and Tygar, J. (2001). Spins: security protocols for sensor networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pages 189–199.

Tolmach, P., Li, Y., Lin, S.-W., Liu, Y., and Li, Z. (2021). A survey of smart contract formal specification and verification. *ACM Comput. Surv.*, 54(7).

Ulrich, A., Holz, R., Hauck, P., and Carle, G. (2011). Investigating the openpgp web of trust'. In *Computer Security - ESORICS.*, pages 489–507. Lecture Notes in Computer Science.

van Bentham, J. (2007). Dynamic logic for belief revision. *Journal of Applied Non-Classical Logics*, 17(2):129–155.

van Bentham, J. (2014). *Logical Dynamics of Information and Interaction*. Cambridge University Press.

van Ditmarsch, H. (2014). Dynamics of lying. *Synthese*, 191(5):745–777.

Zissis, D., Lekkas, D., and Koutsabasis, P. (2011). Cryptographic dysfunctionality - a survey on user perceptions of digital certificates. In C.K., G., H., J., E., P., R., B., and A., A.-N., editors, *Global Security, Safety and Sustainability and E-Democracy*.