Dynamic Logic for Security (DYLOS)

Abstract

The research proposed in this document focuses on the application of dynamic logic in the analysis and design of security protocols. We adopt the intentional approach of multi-agent systems, and aim at protocol specifications in terms of the informational attitudes of the agents involved. To this end, we shall develop and employ dynamic and temporal versions of modal (epistemic, doxastic) logics. We envisage the use of automated verification tools (model checking and automated theorem proving) for analysis and prototyping of protocols.

1 Description of proposed research

The ever increasing role of computers and computer networks in our daily life makes the issue of computer security more and more important. This is reflected in the large amount of security research worldwide. An important part of this research is devoted to the design and analysis of trustworthy security protocols. The complexity of the context in which these protocols are to be used (large and diverse networks with malicious and powerful intruders) requires powerful methods to guarantee their correctness and reliability, indispensable properties for trustworthiness (we give an example below). Only formal methods based on mathematics and logic, supported by mechanized reasoning tools like model checkers and theorem provers, have the potential to become powerful enough for this challenging task. The research proposed here aims to contribute to the realization of this potential.

Our overall reseach goal is

the application of *dynamic logic* to the analysis and design of security protocols.

This contrasts with most other approaches to the analysis and design of security protocols, which either focus on static logical aspects of the intermediate states in the execution of a protocol, or on the dynamic interaction of subprocesses that are part of the protocol. The research proposed here will explore the integration of these two aspects, by exploiting the potential of dynamic logic to provide a unifying framework for describing and analyzing both static and dynamic aspects of systems involving information and information exchange. In doing so, we expect to significantly improve the design of security protocols via logical methods, and to inspire theoretical logic research with research questions emanating from our more applied research.

We consider the security protocols in question from the perspective of *multi-agent systems* with both humans and computer processes in the role of agents [27]. Multi-agent systems are also referred to as *intentional* systems, where agents are characterized by 'mental states' described in terms of beliefs and knowledge (their informational attitudes), as well as goals and intentions (their motivational attitudes). Here, goals represent options available to the agent, i.e. states of affairs that the agent may choose to commit to, and intentions represent the chosen options. The multi-agent perspective is a high-level view of complex systems, abstracting away from implementation details. This makes it a suitable abstraction of security protocols.

We are aware of the pitfalls on the road to logical analysis of security protocols: the problems of semantics, derivability and adequacy. We expect to deal with the first two problems (finding proper semantics and proof systems for the logics at hand) by making use of well-established logics and combining them carefully. The problem of adequacy, i.e.

finding proper abstractions and idealization when translating protocols in logical terms, is of a different nature: solving it requires continuous attention for applications and applicability, and interaction with the security community.

Below, we give a selective overview of the state of the art in logic and security, followed by a description of our approach.

Logic

Modal logic is the logic of modalities, also called intensional attitudes. There is great variety of modalities: 'it is necessary that', 'it will be the case that', 'it has been the case that', 'it should be the case that', 'it is provable that', 'I know that', 'agent a believes that', 'agent b desires it to be the case that', 'when program π is executed, it is the case that', etc. Modalities are denoted by modal operators $\Box$, so if logical formula φ denotes 'it is sunny weather in Groningen' and $\Box$ denotes the epistemic modality 'I know that', then $\Box\varphi$ denotes 'I know that it is sunny weather in Groningen'. Modal logic was first studied as the logic of necessity and possibility. About fifty years ago, Von Wright introduced in [53] epistemic logic (the logic of knowledge) and doxastic logic (the logic of belief) as instances of modal logic. The subject started to flourish a decade later after Kripke's development of *possible-worlds semantics* for modal logic. In the context of epistemic logic, one can view worlds that are possible for a certain agent in a world as *epistemic alternatives* that are compatible with the agent's information in that world. The eighties and early nineties have seen a flurry of activity in the field of epistemic, doxastic and other modal logics, such as temporal logic. To mention a few examples: theoretical computer scientists have applied it to distributed systems, and economists to negotiation (see [15, 30, 49] for overviews of epistemic logic and more applications). More recently, epistemic and doxastic logics were extended with goal operators for application in multi-agent systems [13, 41].

Dynamic logic is an instance of modal logic, first proposed by Pratt in 1976 (see [19, 20, 40]). The modal operator $[\pi]$ is related to action (or program, or protocol) π. The formula $[\pi]\varphi$ expresses: after π has been executed, φ holds. So, in dynamic logic, we can speak and reason about actions and their effects: consider, e.g., the statement $x = 1 \rightarrow [x := x + 1]x = 2$. A related dynamic logic was proposed by Renardel in [42]. Plaza's logic of public communication [39] is the first application of epistemic action (public communication) in a modal context. Independently, Gerbrandy and Groeneveld developed dynamic *epistemic* logic (see [16, 17]), the first modal logic with modalized actions. Baltag presents (mostly with Moss and Solecki) in [2, 4, 5, 3] several related logics for epistemic actions and belief updates; see also [43] by Renardel for related work from a general modal perspective. Applications of dynamic epistemic logic on so-called knowledge actions (such as showing a card in a game like bridge) can be found in [2, 17, 48, 50].

Security

We give an example of a security protocol: the Needham-Schroeder authentication protocol (see [31]), which two agents can execute to mutually verify their identities.

$$A \rightarrow B : \{A, N_1\}_{K_B}$$
$$B \rightarrow A : \{N_1, N_2\}_{K_A}$$
$$A \rightarrow B : \{N_2\}_{K_B}$$

In the first step of the protocol, agent A sends to an agent B his name A and a *nonce* N_1 (i.e. a large pseudo-random number that serves as a challenge) encrypted with B's *public key* K_B. If we assume that cryptography is perfect and that each agent is the sole possessor of his inverse *private* key, then B is the only agent who can decrypt the message $\{A, N_1\}_{K_B}$. In the second step of the protocol, agent B responds to A's challenge by returning the nonce N_1, and posing his own challenge N_2. A now decrypts the message, sees that his challenge has been responded to, and thus authenticates B, i.e. A "believes" that he is indeed communicating with B. In the third message, A responds to B's challenge, and B can thus similarly authenticate A.

The Needham-Schroeder protocol has been analyzed with logical and other methods (see [8, 18, 47]), and it was generally thought to be correct. So it was quite a surprise when, eighteen years after its publication, Lowe [28] discovered in 1996 an error in the protocol and showed that it could be broken (and he proposed an improved version, now known as the Needham-Schroeder-Lowe protocol). He pointed out that the final step of the Needham-Schroeder protocol fails to authenticate the initiator agent to the responder agent: it is possible for a responder to finish a protocol execution with an initiator who is other than he claimed to be in the first step. The next diagram shows a so-called 'man-in-the-middle' attack:

$$
\begin{array}{ll}
A \rightarrow spy : \{A, N_1\}_{K_{spy}} & \\
 & spy \rightarrow B : \{A, N_1\}_{K_B} \\
 & B \rightarrow spy : \{N_1, N_2\}_{K_A} \\
spy \rightarrow A : \{N_1, N_2\}_{K_A} & \\
A \rightarrow spy : \{N_2\}_{K_{spy}} & \\
 & spy \rightarrow B : \{N_2\}_{K_B}
\end{array}
$$

A executes one protocol run with *spy*, who fakes A's identity with respect to B. At the end of the two interleaved protocol executions, A believes, correctly, that he is talking to *spy*, but B has been tricked by *spy* into believing that he is talking to A!

The reason that Lowe found an error while others seemingly proved Needham-Schroeder correct is that he studied the protocol in the context of open networks, which did not exist in 1978 when NSPK was proposed. Lowe also used the (by now standard) Dolev-Yao attacker model [12]. This model postulates that *spy* is a powerful opponent who is in control of the network and can intercept, read and alter messages at will. In this context, a full analysis of the protocol is a quite complex task, and Lowe used a model checker.

Examples of this kind show that protocols may (and too often do) contain subtle errors, which can only be found via formal analysis supported by automated verification. This observation, already made in [31] by Needham and Schroeder in 1978, justifies the kind of research proposed here.

Process formalisms and theories have been developed and used successfully in the analysis and verification of communication protocols in general. To deal with security protocols, existing theories have been applied (e.g. trace theory in [9]) and several new process theories have been proposed, such as the Spi calculus [1] and strand spaces [14].

The application of logic and logical theories in security started with the so-called BAN logic of authentication defined by Burrows, Abadi and Needham in 1989 [8]. It provides a language, axioms and inference rules to express different notions of correctness for security protocols (e.g. secrecy of data and validity of cryptographic keys), and to describe how the beliefs (or knowledge) of agents evolve as messages are exchanged. BAN is not perfect (see e.g.

[32]), and several improvements and extensions have been proposed (see e.g. [18, 47, 51, 54]). For the proper definition of the semantics of BAN, strand spaces are used (see [46], [26]). Recently, Reynolds' separation logic [44] has gained interest as a more solid logical base for reasoning about security issues.

The applicability of the security logics mentioned above in the analysis of security protocols is limited by the fact that the protocols are not part of the language, but part of the structure over which the logic is interpreted. As a result, it is unclear how to express the interplay between protocol steps and knowledge in a non ad-hoc way. In contrast, in dynamic logic programs are an explicit part of the language, admitting the description of the properties of the interaction between programs and propositions that are dependent of the domain of computation. So dynamic logics close the gap between programs and propositions, and this leads to our research goals.

Research goals and objectives

As we stated above, our overall goal is *to apply dynamic logic to the analysis and design of security protocols*. We divide this goal into two subgoals, one for functional aspects of the protocols involved, another for the operational aspects. Functional specifications focus on *what* the protocol guarantees (e.g. that agent b will receive information φ from agent a while no other agent will learn φ), while operational specifications focus on *how* a property is ensured to hold (e.g. by using a public/private key encryption system and a trusted third party). The distinction between functional and operational specification and behaviour, quite common in software engineering, was made by Roscoe in [45] for security protocols, but he used the terms extensional (for functional) and intensional (for operational).

The functional behaviour of security protocols for information exchange can be described in a natural way in terms of modalities of agents with respect to information: what do they know, what do they want to know, what are they allowed to know, what are they allowed to do, and what do they want to do (with certain information). This corresponds to the intentional perspective on multiagent systems. As we indicated above, many instances of modal logic have already been developed to describe and reason about security protocols. However, these logics are static in the sense that they do not have the expressive power to speak about actions and their effects. But dynamic logics have been devised precisely for this purpose. This leads to our **first subgoal**, viz. *to develop and apply dynamic modal logics for functional specification and analysis of security protocols*. This use of dynamic modal logic for security protocols is fairly new: we are only aware of the initial attempts of Bleeker and Van Eijck in this direction [6, 7], and of the application of dynamic epistemic logic on the SRA Three Pass protocol and the Wide-Mouthed Frog protocol in [23]. They all use action structures in the spirit of Baltag, Moss and Solecki. These action structures will be a starting point for our research, although we shall also consider the alternative notion of modal structures as proposed by Renardel in [43].

For the description and analysis of the operational behaviour of security protocols it is common practice to use formalisms based on process theories, where the behaviour of processes composed from subprocesses (also called threads, strands, cords or bundles in the extensive literature about this subject) can be studied. The input-output style of characterizing actions in dynamic logic is too coarse for this purpose, as it does not deal with the intermediate results of processes. Here temporal logic will fill the gap, another instance of modal logic with temporal operators F 'once in the future, it will be the case that...' and G 'from now on, it will always be the case that ...'. Moreover, temporal logics are the linguistic vehicle

for the transfer of process properties to model checkers. So, our **second subgoal** reads: *combine dynamic modal logic with temporal logic, so as to facilitate the specification of both functional and operational behaviour of security protocols in one linguistic framework, and their analysis via model checking.* The combination of dynamic logic with temporal logic has been scarcely explored, but we are aware of the work of Datta and others (see [10, 11]) where a Hoare-style formalism with temporal logic formulae is proposed and used as a derivation system for security protocols. This will be a starting point for our research.

We shall not only design and study new logical formalisms, but we shall also apply them in the specification, analysis and design of security protocols. We shall consider well-known protocols (Needham-Schroeder-Lowe, Diffie-Hellman, Kerberos, Otway-Rees, Yahalom, Wide-mouthed Frog, Neumann-Stubblebine).

Experience learns that the inherent complexity of the analysis of protocols makes the use of verification tools indispensable, and we shall use both model checking and theorem proving tools in the research proposed here. Model checking has become very popular as a verification tool of the behaviour of processes using temporal logic (SPIN [21, 22], FORSPEC [29, 52]). Popular theorem provers are Isabelle [35], well known by Paulson's seminal work on security protocol verification (see [36, 37, 38]), and PVS [33, 34], based on higher-order logic and used e.g. by Jacobs in the context of security protocols (see [24, 25, 26]).

Scientific interest

The scientific mission of the research project is twofold. We shall apply theoretical concepts and results, e.g. dynamic modal logic and temporal logic, to contribute to the further development and improvement of formal methods for design and analysis of security protocols. But we shall also inspire ourselves and others to gain more theoretical insight in the logical concepts and systems that we encounter on our way.

References

[1] Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The Spi calculus. In *Fourth ACM Conference on Computer and Communications Security*, pages 36–47. ACM Press, 1997.

[2] A. Baltag. A logic for supicious players: Epistemic actions and belief updates in games. *Bulletin of Economic Research*, 54:1–45, 2002.

[3] A. Baltag and L.S. Moss. Logics for epistemic programs. *Synthese*, 139(2):165–224, 2004.

[4] A. Baltag, L.S. Moss, and S. Solecki. The Logic of Public Announcements, Common Knowledge, and Private Suspicions (extended abstract). In I. Gilboa, editor, *Proceedings of the 7th Conference on Theoretical Aspects of Reasoning and Knowledge (TARK'98)*, pages 43 – 56. Morgan Kaufmann Publishers, 1998.

[5] A. Baltag, L.S. Moss, and S. Solecki. The Logic of Public Announcements, Common Knowledge, and Private Suspicions. CWI Technical Report SEN-R9922, Centrum voor Wiskunde en Informatica (Centre for Mathematics and Computer Science), Amsterdam, 1999.

[6] A. Bleeker and J. van Eijck. The epistemics of encryption. Technical Report INS-R0019, CWI, September 2000.

[7] Annette Bleeker and Jan van Eijck. Epistemic action and change. In G. Bonanno, E. Colombatto, and W. van der Hoek, editors, *LOFT-4 Proceedings*, Torino, 2000.

[8] Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8:18–36, 1990.

[9] C.J.F. Cremers, S. Mauw, and E.P. de Vink. Defining authentication in a trace model. In Theo Dimitrakos and Fabio Martinelli, editors, *Fast 2003*, Proceedings of the first international Workshop on Formal Aspects in Security and Trust, pages 131–145, Pisa, September. IITT-CNR technical report.

[10] A. Datta, A. Derek, J.C. Mitchell, and D. Pavlovic. A derivation system for security protocols and its logical formalization. In *Proceedings of the 16th IEEE Computer Security Foundations Workshop*, pages 109–125. IEEE, 2003.

[11] A. Datta, A. Derek, J.C. Mitchell, and D. Pavlovic. Secure protocol composition. *Electronic Notes in Theoretical Computer Science*, 83, 2004.

[12] D. Dolev and A.C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 2(29):198–208, 1983.

[13] Barbara Dunin-Keplicz and Rineke Verbrugge. Collective intentions. *Fundamenta Informaticae*, 51:271–295, 2002.

[14] F. Javier Thayer Fábrega, Jonathan C. Herzog, and Joshua D. Guttman. Strand spaces: proving security protocols correct. *Journal of Computer Security*, 7:191–230, 1999.

[15] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge, Mass., 1995.

[16] Jelle Gerbrandy. *Bisimulations on Planet Kripke*. PhD thesis, Institute for Logic, Language and Computation, Amsterdam, 1999.

[17] Jelle Gerbrandy and Willem Groeneveld. Reasoning about information change. *Journal of Logic, Language and Information*, 6:147–169, 1997.

[18] Li Gong, Roger Needham, and Raphael Yahalom. Reasoning about belief in cryptographic protocols. In *Proceedings of the IEEE 1990 Symposium on Security and Privacy (Oakland, Calif.)*, pages 234–248. IEEE Computer Society Press, 1990.

[19] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic Logic*. The MIT Press, Cambridge (Mass.), London, 2000.

[20] David Harel, Dexter Kozen, and Jerzy Tiuryn. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic (second edition)*, volume 4, pages 99–217. Kluwer Academic Publishers, 2002.

[21] G. Holzmann. *Design and Validation of Computer Protocols*. Prentice Hall International: Hemel Hempstead, England, 1991.

[22] G. Holzmann. The Spin model checker. *IEEE Transaction on Software Engineering*, 23(5):279–295, 1997.

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

[24] B. Jacobs. JavaCard Program Verification. In R.J. Boulton and P.B. Jackson, editors, *Theorem Proving in Higher Order Logics*, number 2152 in LNCS, pages 1–3. Springer Verlag, 2001.

[25] B. Jacobs. Java's Integral Types in PVS. In P. Stevens E. Najim, U. Nestmann, editor, *Formal Methods for Open Object-Based Distributed Systems (FMOODS 2003)*, number 2884 in LNCS, pages 1–15. Springer Verlag, 2003.

[26] Bart Jacobs. Semantics and logic for security protocols. `http://www.cs.ru.nl/B.Jacobs/PAPERS/protsemlog.pdf`, 2004.

[27] Nicholas R. Jennings, Katia Sycara, and Michael Wooldridge. A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.

[28] Gavin Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Proceedings of TACAS'96*, LNCS 1055, pages 147–166. Springer-Verlag, 1996.

[29] K. L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers: Boston, MA, 1993.

[30] J.-J.Ch. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science*. Cambridge University Press, Cambridge, 1995.

[31] Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21:993–999, 1978.

[32] Dan M. Nessett. A Critique of the Burrows, Abadi and Needham Logic. *ACM SIGOPS Operating Systems Review*, 24(2):35–38, April 1990.

[33] S. Owre, S. Rajan, J.M. Rushby, N. Shankar, and M.K. Srivas. PVS: Combining specification, proof checking, and model checking. In R. Alur and T.A. Henzinger, editors, *Computer-Aided Verification. Proceedings CAV'96*, pages 411–414. Springer-Verlag (LNCS 1102), 1996.

[34] S. Owre, N. Shankar, J.M. Rushby, and D.W.J. Stringer-Calvert. PVS Version 2.4 (2001). System Guide, Prover Guide, PVS Language Reference. `http://pvs.csl.sri.com`, 2001.

[35] L.C. Paulson. *Isabelle – a generic theorem prover*. Springer-Verlag (LNCS 828), 1994. (with contributions of T. Nipkow).

[36] L.C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6:85–128, 1998.

[37] L.C. Paulson. Inductive analysis of the internet protocol TLS. *ACM Transactions on Computer and System Security*, 2(3):332–351, 1999.

[38] L.C. Paulson. Relations between secrets: the Yahalom protocol. In J.A. Malcolm B. Christianson, B. Crispo and M. Roe, editors, *Proceedings of the 7th Cambridge International Workshop on Security Protocols*, LNCS 1796, pages 73–77. Springer-Verlag, 1999.

[39] 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. North-Holland, 1989.

[40] Vaughan R. Pratt. Semantical considerations on Floyd-Hoare logic. In *17th annual symposium on foundations of computer science*, pages 109 – 121, New York, 1976. IEEE.

[41] Anand S. Rao and Michael P. Georgeff. Modeling rational agents within a BDI-architecture. In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1991.

[42] Gerard R. Renardel de Lavalette. A logic of modification and creation. In Cleo Condoravdi and Gerard R. Renardel de Lavalette, editors, *Logical Perspectives on Language and Information*, pages 197 – 219. CSLI publications, Stanford, USA, 2001.

[43] Gerard R. Renardel de Lavalette. Changing modalities. *Journal of Logic and Computation*, 14(2):253 – 278, 2004.

[44] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *17th IEEE Symposium on Logic in Computer Science (LICS 2002), 22-25 July 2002, Copenhagen, Denmark, Proceedings*, pages 55–74. IEEE Computer Society, 2002.

[45] A. W. Roscoe. Intensional specifications of security protocols. In *Proceedings of the 9th IEEE Computer Security Foundations Workshop: CSFW'96*, pages 28–38. IEEE Computer Society Press, New York, 1996.

[46] Paul Syverson. Towards a strand semantics for authentication logics. In S. Brookes, A. Jung, M. Mislove, and A. Scedrov, editors, *Mathematical Foundations of Programming Semantics*, number 20 in Electronic Notes in Theoretical Computer Science. Elsevier, Amsterdam, 1999.

[47] Paul F. Syverson and Paul C. van Oorschot. A unified cryptographic protocol logic. Technical Report 5540-227, Naval Research Laboratory CHACS, 1996.

[48] J.F.A.K. van Benthem. Games in dynamic-epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2001.

[49] W. van der Hoek and R. Verbrugge. Epistemic logic: a survey. In L.A. Petrosjan and V.V. Mazalov, editors, *Game Theory and Applications, vol. 8*, pages 53–94. Nova Science Publishers, New York, 2002. ISBN: 1-59033-373-X.

[50] H.P. van Ditmarsch. Knowledge games. *Bulletin of Economic Research*, 53(4):249–273, 2001.

[51] Paul C. van Oorschot. Extending cryptographic logics of belief to key agreement protocols (extended abstract). In *Proceedings of the 1st ACM Conference on Communications and Computer Security (Fairfax, Virginia)*, pages 232–243. ACM Press, 1993.

[52] M. Y. Vardi. Branching vs. linear time: Final showdown. In T. Margaria and W. Yi, editors, *Proceedings of the 2001 Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2001 (LNCS Volume 2031)*, pages 1–22. Springer-Verlag, 2001.

[53] G.H. von Wright. *An Essay in Modal Logic*. North-Holland, 1953.

[54] Gabriele Wedel and Volker Kessler. Formal semantics for authentication logics. In *Computer Security - ESORICS 96 (4th European Symposium on Research in Computer Security, Rome)*, Lecture Notes in Computer Science 1146, pages 219–241. Springer-Verlag, 1996.