

Action de Recherche Amont

SECURITE, SYSTEMES EMBARQUES ET INTELLIGENCE AMBIANTE

Scientific Description of the project COPS

(Composition Of Policies and Services)

1 Goal and context

1.1 What are Web services ?

A recent trend in the design of distributed applications is the paradigm of Service Oriented Computing (SOC). Developed on the top of various communication networks, this paradigm is based on the definition and composition of independent computational units called *services*. For example a company may enable its workforce, business partners and customers the access to some selected corporate resources through a collection of such services.

The most challenging aspect of services is that they should be build with little knowledge on their execution environment in order to meet the independence criteria. This implies that each service should rely on the existence and availability of some other dynamically retrieved service to perform its computation. For example a service granting the access to a corporate resource should rely on an authentication service to assess the identity of a client and on an authorization service to decide whether the requested resource may be granted or not.

Thus communications happen between human agents and services as well as between services. Since we consider services in an open environment, the medium on which these communications take place must be considered unsafe in order to reflect the potential vulnerabilities of communications on *e.g.* Internet. Since a malicious user could try to mis-use a service to obtain *e.g.* confidential data, the access to services has to be regulated by a *security policy*. The most usual way to securise services on the Internet is to define a security policy based on the source network of a request and to enforce this policy by a firewall. The drawback of this method is its inadequation the to the desired behavior of a security policy: One usually wants to define access to a resource with respect to the requester instead of with respect to her current access point on the Internet.

A way of defining a more discriminative security policy is to impose all communications to pass through an open port and to grant or deny access with respect to credentials accompanying a request. This is the solution adopted for *Web Services* (WS): They are services that communicate over the port used for Web browsing. This embedding of communications into HTTP requests also has the advantage of providing a simple and infrastructure-independent communication scheme *via* the SOAP protocol. This simplicity has lead to the evolution of WS from a promising technology to a fast-adopted technology for the development of Web-based applications.

The possible corporate applications are numerous and range from the access and usage control of a GRID to the booking of plane tickets on the Web. But there are also numerous possible applications in the domain of simplified access to administrations, such as on-line declaration and payment of taxes or the on-the-web accessibility of medical records. While lots of these applications are desirable from a user's point of view, their eventual acceptance will depend on the level of security provided by these services. For example while an immediate on-line access to the medical records of an unconscious person in the emergency unit of a hospital may permit to save this person's life, the same access by a potential employer would lead to unacceptable discrimination.

The aim of our proposal is to build technologies enabling the security analysis of web services that take into account the potential flaws at communication level, at the access policy level or at the interface between

communications and access policy, *e.g.* when protocols are executed in order to update the security policy of a site.

1.2 Objectives

Integration of applications and resources through WS technology is expected to grow dramatically though this implies an increased degree of exposure to illegal accesses. Protecting critical resources requires a thorough analysis of new security issues and in particular better understanding and modeling of interactions of WS underlying protocols and access control policies. In this project we will focus only on 3 critical sub-problems that we have identified.

Modeling security policies for services The modeling of access control for services encompasses three questions:

- Definition of an access control model that fits the needs of SOC;
- Decidability of the consistence of a security policy in this model;
- Decidability of the *security* of a given policy;

Codification of access control models in standards defined mainly two important access control modes: discretionary access control (DAC) and mandatory access control (MAC). The problem is that neither DAC nor MAC are sufficient for implementing the needs and the specificities of SOC. Hence, our first aim will be to elaborate an access control mode that fits well the needs of SOC.

Within this access control mode, we will precisely define what the usual concepts of “protection state” and “protection system” mean for services. From then, seeing that security policies are defined in terms of permissions, obligations, and prohibitions, our second aim will be to elaborate methods for automatically proving that the security policy of a given WS is and will remain consistent.

Finally, we will define what “security” means for a WS. In this context, we will consider the decidability/complexity of the security problem for services, *i.e.* the problem of determining whether or not the security policy of a given WS satisfies such-and-such security property. We will also study the restrictions that are needed to make the security problem decidable. More precisely, we want to characterize the services that we can prove to be safe.

Interaction between services and the security policy. As a first step toward analysis of WS we will consider the case of cryptographic protocols written in XML. In contrast with existing approaches we will aim at analyzing the XML implementation instead of a higher level term abstraction of messages. Our goal is to capture possible flaws that can be introduced by an ambiguous reading of a message. The consequence will be the ability to conduct the analysis of a protocol directly on its SOAP implementation. The main difficulty here is that messages contain open-ended data structures such as list and a careful analysis and model will be needed to preserve known decidability results on the analysis of protocols.

Once the problem of analysis of simple XML-based protocols will be solved we will consider the more interesting combination of protocols and security policy which is at the core of Web services. While it is clear that the execution of a service must depend on the security policy of this service, a service may also change the security policy of a site as in the case *e.g.* of administrator tools for distant configuration of a system. The main difficulty of this goal is to combine the two up-to-now independent formalisms of security policy and protocols.

In this context we will first focus on the expression of security policies into the message-passing formalism used to describe protocols. Then we will consider the problem of compatibility of security policies when services are executed without an attacker. Finally we will address the most difficult case of an attacker that actually tries to break the security policy, first by a *passive* intruder able to divert and delay messages, then by an *active* intruder that is also able to change messages and build new ones from its knowledge.

A further, more prospective goal will be to consider the problem of modular certification of a service. While aforementioned work is concerned with the certification of a given protocol we will consider the certification of a service within a set of currently available services. This will permit to monitor on-the-fly dynamic service compositions with respect to security policies of component services.

Synthesis of WS. Some works on the synthesis (or composition) of a complex services from elementary WS have already been published but none of them consider the security policy during the synthesis. We plan to start our work on synthesis of security-related services such as those provided by authentication and authorization authorities. An extension will be to consider the synthesis of services in general and to devise a generic procedure that takes into account the constraints introduced by the security policies.

1.3 Existing standards

The purpose of our project is to study security issues related to web services. As mentioned in the preamble, web service is a generic term for a collection of open protocols and standards used to exchange messages between applications running on disparate platforms. Hence web services have a role similar to other technologies ensuring interoperability between distributed applications, like the “remote procedure call” paradigm used by CORBA or Microsoft DCOM. Nonetheless web services display a distinctive set of features: (1) messages are exchanged in *eXtensible Markup Language* (XML) format and transmitted using the *HyperText Transfert Protocol* (HTTP) ; (2) the interface of web services, that is the set of methods available to clients and their types, is defined in a common XML grammar and can be made public; (3) there is a mechanism by which interested parties can look up web services information in order to determine whether to use them, locate these services and inspect their public interface. In the general case, the XML standard used for messages is the SOAP protocol [62], interface definitions are expressed with the *Web Service Description Language* (WSDL) [63] and web services directories are available via the *Universal Description, Discovery, and Integration* language (UDDI).

SOAP messages and WSDL description are particular examples of XML documents, hence they can be manipulated algorithmically, for example using XPath queries. In contrast, UDDI is designed to be interrogated by SOAP messages and to provide access to WSDL documents describing the protocol bindings and message formats required to interact with the web services listed in its directory.

SOAP and XML Cryptographic Primitives. SOAP is an XML vocabulary that provides the basic messaging framework of web services. Basically, a SOAP message is contained in an envelope and made of two main sections: a body containing the payload of the message, and an optional header containing a sequence of relevant information about the message, such as routing or authentication information’s.

Confidentiality and integrity of SOAP messages can be obtained by means of XML-based security standards. Most notably *XML-Encryption*, which defines the syntax and processing rules for encrypting XML content, and *XML-Signature* its counterpart for creating digital signatures on XML content. These standards are both W3C recommendations [58, 59]. XML Encryption addresses the issue of data confidentiality using encryption techniques. The specification is quite flexible: encrypted data are wrapped inside XML tags and it is possible to encrypt a complete XML file, to encrypt any single element, to encrypt only the contents of an element or to encrypt an already encrypted element. Similarly, the goal of XML Signature is to provide support for data integrity and authentication (both message and signer authentication) wrapped inside XML format.

Web Service Security, Authentication and Access Control. As it is the case with classical distributed protocols, cryptography is required, but not sufficient, to ensure the security of an application. For instance, even in the presence of signed messages, a malicious application could use faked identities to access sensitive information shielded by a web service. A solution to this problem is to build the equivalent of a firewall at the SOAP level that is able to inspect messages, trying to match user roles with access lists, policy

levels, ... In practice, such filtering application should skim the headers of a SOAP message and inspect only the security-related parts of the message body.

The *Web Services Security* (WSS) specification from OASIS defines the details of how to apply XML Signature and XML Encryption concepts in SOAP messaging. It provides a standardized approach to ensure message integrity, user and single message authentication, and confidentiality. WSS describes mechanisms to include digital signatures, message digests and encrypted binary streams inside a SOAP body.

At a more abstract level, the *Security Assertion Markup Language* (SAML) is an XML-based framework for sharing user authentication and authorization information between on-line partners. Actually, SAML is a prominent standard underlying many of the single sign-on services found in commercial applications. SAML assumes the principal has enrolled with at least one identity provider that is expected to provide local authentication services to the principal. As its name suggests, SAML can be used to make assertions regarding the identity, authorizations, and attributes of a subject to other entities. These assertions are passed as separate XML messages, either pushed from the Asserting Party to the Relying Party, or pulled from the Asserting Party by the Relying Party.

The last standard considered in this section deals with another important problem not solved by simple cryptographic primitives, namely access-control. This is the scope of the *eXtensible Access Control Markup Language* (XACML), an XML vocabulary used to define authorization policies. An XACML document provides a fine-grained description of the authorized operations (e.g. read, write, copy, ...) based on, among other criteria, access requester characteristics, the protocol over which the request is made (e.g. HTTPS), or the authentication mechanism in use.

In the course of our research project, we expect to deal with (and partially model) each one of the standards cited in this section. Actually, these standards may be viewed as the components of a *web services (security) protocol stack*, that is the ordered sequence of operations encountered by a message in the implementation of a service: (1) messages between services are written in the SOAP vocabulary and transported over HTTP (a message embeds an action, that is the name of a service and a sequence of parameters); (2) the interface of the action embedded in the SOAP envelope is matched against the service description, defined in WSDL; (3) the authenticity and integrity of headers and action parameters are tested using the WSS standard (itself possibly parametrized by a SAML description); (4) finally the action is performed if it is allowed by the access-control policy, e.g. expressed using XACML.

1.4 Related projects

We now briefly describe the state of the art in the design of validated secure service. Let us first note that many works have been dedicated to automated analysis of security protocols [12, 47, 13, 22, 26, 29, 43, 45, 48, 50, 52] that can be considered as basic components of security services. Some recent works have applied standard verification techniques such as model-checking to the verification of web services properties that are not security-related [54, 57]. Typically [32] shows how to verify whether composed web services specified in the *Web Services Business Processes Execution Language* (WSBPEL) [49] can be synchronized using SPIN model-checker. However very few works address the challenge of validating security properties of web services.

We shall discuss now these works. The most related project is Samoa [7] from Microsoft Research, Cambridge. The goal of Samoa is to exploit recent advances in the analysis of security protocols in the practical setting of XML web services and in particular a logic-based approach to checking SOAP-based protocols. Samoa proposes a specification language TulaFale extending the π -calculus by XML syntax for handling SOAP envelopes. TulaFale [8] is compiled into the applied π -calculus, and then the Proverif resolution-based protocol verifier of Blanchet [11] can be applied to the result in order to verify automatically authentication and secrecy properties of SOAP protocols. To summarize this work rather focus on XML encoding of security tokens and messages. It has been applied to relatively simple protocols with few roles and small series of messages [44]. It is unclear whether the approach will scale up when considering complex/composed services with many roles. Also Proverif is a tool oriented towards verification; it is not designed for efficient attack detection. For instance due to some nonces abstraction it may report false

attacks.

The validation tools we will base the project on have been developed by partners and dispose of both the capabilities of attack detection and positive verification. In particular in the category of attack detection tools they are the most efficient ones and have been tested on protocols considerably more complex than those considered in Samoa. We also believe that we can handle in a more complete way algebraic properties of cryptographic primitives due to recent theoretical results we obtained [14, 15, 17].

The network security and cryptography research group at IBM ZURICH has been actively investigating web services security especially in the context of Federated Identity Management protocols. They have given special attention to the incorporation of cryptography in the service development process with the main goal of achieving specifications that are amenable to formal verification, and allow for a provably secure cryptographic implementation [4]. Since they have not developed automatic support for designing validated business process, we think our work complements nicely theirs.

Jan Jürjens and his group from TU Munich have developed UMLsec an extension of Unified Modeling Language to include security requirements into system models. Theorem proving or model-checking tools can then be employed to establish practical security guarantees. Jan Jürjens stresses the challenge of ensuring security properties for combined layered protocols. He tackled this problem on a banking application [33]. Since the cryptographic protocols are analyzed with SMV model checking tools they had to simplify the adversary to make it a finite state machine, with the risk of missing some attacks. This would not be required with our approach. They have recently applied their development technique to a secure-service based system, the authentication process of a driver against an Automotive System [28]. They have detected and fixed a flaw in the system but cannot ensure due to the incompleteness of this finite-state model-checking approach that the resulting system is safe.

From this survey we draw the conclusion that existing solutions to secure service design do not scale up for complex validated security services; in particular they do not; 1) provide sufficient automated support; 2) take into account XML specificities; and 3) consider the impact of security policies.

1.5 Competences of partners

1.5.1 Laboratoire d'Informatique Fondamentale de Marseille (LIF, UMR 6166)

Two researchers from the MoVe group, Silvano Dal Zilio and Denis Lugiez, take part in this project proposal. Their main contributions in the domain deal with the verification of cryptographic protocols and with formal methods for XML processing. They will also provide their expertise in research problems related to the project, namely language-based security for mobile code and static analysis methods for concurrent systems.

In the area of verification of cryptographic protocols, Denis Lugiez has participated in the design of a process calculus focused on the analysis of bounded number of protocol sessions. A prototype analyzer that relies on this calculus, called TRUST [3], has been developed. More recent works treat the case of passive intruders in a model enriched with algebraic properties of the cryptographic operations such as the combination of exclusive-or and homomorphism [40]).

In the area of XML processing, Silvano Dal Zilio and Denis Lugiez have defined an automata theoretic approach for managing semi-structured data. A new class of tree automata has been defined, based on previous works of this group on ambient logic. Decidability and undecidability results have been proved that indicate which kind of restrictions are useful for typing semi-structured data [25]. In particular, this approach has been applied to the study of the XML-Schema standard [24].

1.5.2 Laboratoire Lorrain de Recherche en Informatique et Automatique (LORIA, UMR 7503)

Three members of the LORIA will be involved in this project, Michaël Rusinowitch and Laurent Vigneron from the CASSIS team and Olivier Perrin from the ECOO team.

CASSIS team has designed a fully automatic tool CASRUL for detecting flaws in security protocols for Dolev-Yao model. Protocols are formulated in a high-level specification language. A translator compiles the

protocol and intruder activities into first-order clauses further processed by an automatic theorem prover [38]. The system has been applied to many protocols: most of the known flaws listed in the Clark and Jacob authentication protocols library have been detected using an efficient original search strategy: the lazy intruder strategy [18]. New flaws have also been revealed [20]. CASSIS has published the first decision procedures for protocols with pairing, unbounded message size, and composed keys [19]. We have also shown that this problem is NP-complete [53]. We have recently extended this result to handle the algebraic properties of the XOR operators occurring in the protocols [14] and a theory of exponentiation [15].

ECOO team is concerned with the cooperation of distributed applications. Olivier Perrin has focused on WS composition, oriented towards reliable computation (in terms of termination of the WS as expected by the designers) and WS coordination/monitoring.

The coordination (also called choreography) of a set of WS is a protocol that defines the sequences, the conditions of the messages exchanges, and the data exchanges. A conversation is an instance of a protocol. Given a protocol, our interest is to check the reliability of the coordination protocol, on a transactional point of view. In order to get a reliable execution of a composite WS, the transactional behavior of the composite service is inferred. This behavior describes what happens in case of failure or what kind of compensation policies can be applied in this case. Conversely, given a transactional behavior for a composite WS, one wants to check whether the set of chosen WS is compatible with this transactional behavior [9].

The second aspect is WS coordination and monitoring. These are important in order to get the awaited result, despite the distribution in space, time and organizations of services. A contract can be considered as a facility to define and check the WS compositions, and we try to detect violations of policies and clauses in the protocol. For that, we have proposed a event calculus based approach, coupled with non-classical logics devoted to the formalisation of permissions, obligations and prohibitions.

1.5.3 Institut de Recherche en Informatique de Toulouse (IRIT, UMR 5505)

The research activity on security is a rather new research theme at IRIT. Philippe Balbiani, a full-time researcher, is working on modal logics and their applications to computer science. His current research is oriented toward the formalization of obligations and permissions and the translation of these concepts into more expressive security policy languages. The preliminary work on obligations [5] has already been partially applied in [23] to model execution of non-atomic actions. Within the framework of the RNRT project MP6, “Modèles et politiques de sécurité pour les systèmes d’informations et de communications en santé et social”, he has participated to the elaboration of the access control mode ORBAC [1], “Organisation-Based Access Control”, which overcomes the limitations of more classical modes like DAC, MAC, or RBAC.

Fahima Cheikh has obtained her master diploma this year at Université Paul Sabatier. She has explored various extensions of the matrix model which is the primary abstraction model in access control. In particular, she has studied matrix models where the availability of resources can be expressed, an issue where a lot of questions remain unsettled.

Yannick Chevalier is a lecturer at Université Paul Sabatier and has worked during the preparation of his thesis in the CASSIS research group. His main contributions are in the area of symbolic analysis of cryptographic protocols both in term of flaw detections and of validation as well as in the incorporation of the properties of low-level primitives into the analysis of these protocols.

1.5.4 Microsoft Research, Cambridge (UK)

Two researchers from the “Programming Principles and Tools” group, Cédric Fournet and Andrew Gordon, are involved in this project proposal. Their general research interests are in the area of computer programming languages, with a focus on applying type theory and other formal techniques to problems of computer security.

Cédric Fournet works include the study of secure implementations of communication abstractions, access control for mobile code, authorization policies and private authentication. Andrew Gordon is a co-developer (with Martin Abadi) of the spi calculus, an extension of the π -calculus with cryptographic primitives. This work is at the basis of a type-checker for cryptographic protocols, named Cryptic. They are both leading

the *Samoa* project [46] on the verification of cryptographic protocols for Internet and Web services security. Some early outcomes of this project include an implementation of declarative security attributes for web services and the design of a logic-based approach devoted to the verification of SOAP-based protocols. The TulaFale tool [8], for instance, is a new specification language for writing machine-checkable descriptions of SOAP-based security protocols and their properties. This tool has been applied to check the safety of “industrial strength” toolkits for web services, such as Microsoft’s WSE.

2 Project description

Before proceeding to the objectives of our project introduced in Section 1.2 we review in the first subsection the security issues related to WS.

2.1 Security issues

2.1.1 Centralized vs distributed systems

Models for centralized access control. Access control is the most fundamental security mechanism in use. It has the potential to preserve the confidentiality and integrity of information: only authorized users can read information and only authorized users can alter information. It has also the potential to preserve the availability of information, seeing that an attacker who gains unauthorized access to a system is likely to prevent its smooth running. A security policy is a statement of what is permitted and what is forbidden. In its simplest form, it is represented mathematically as an authorization table which describes the rights of users over files [41]. Codification of access control models in standards defines two important access control modes: discretionary access control (DAC) and mandatory access control (MAC) [10, 27]. DAC is a mode in which the creators or owners of files assign access rights: a subject with discretionary access to information can pass that information on to another subject. By itself, DAC is insufficient for implementing the protection of confidentiality. To provide a truly secure scheme in which a system is guaranteed to remain secure, MAC is required. The key feature of MAC is that users are limited in the actions they can take according to their security clearances. These clearances represent sensitivity levels: the higher the clearance of a user is, the more sensitive the information this user can read is. In the late 1980s and early 1990s, researchers began recognizing the virtues of roles as an abstraction for managing authorizations. Role-based access control (RBAC) is a mode in which access to computer system objects is based on a user’s role in an organization. The formal description of the model is given in [30, 55]. It basically says that a role is essentially a collection of permissions and all users receive permissions only through the roles to which they are assigned. None of these models is fully satisfactory to model security policies that are not restricted to static permissions but also include contextual rules related to obligations and prohibitions. To overcome the limitations of DAC, MAC, and RBAC, several authors [1] have recently considered organization-based access control (ORBAC). ORBAC is a mode in which the concept of organization and the concept of context are essential, seeing that each organization has to define its own internal security policy as a set of contextual rules. In ORBAC, using these concepts, a security policy that applies to a given organization is defined as a collection of permissions, obligations, and prohibitions.

Access control in a decentralized system. Access control necessarily depends on proper authentication: if the system cannot be certain of a user’s identity then there is no valid way of determining if the user should be granted access. Since authorization refers to a “yes” or “no” decision as to whether a user is granted access to a system resource, an information system must maintain some relationship between users and resources, possibly by attaching a list of authorized users to each resource, also known as an access control list, or by storing a list of accessible resources with each user, also known as a capability list. Within the context of centralized systems, there exists exactly one authority controlling accesses of users to resources and maintaining access control lists or capability lists. Authentication mechanisms persuade this authority that users are who they claim to be whereas users know in advance who mediate their attempts to access a

resource in the system. Within the context of decentralized systems like web services, authorities controlling accesses of users to resources and users wishing to access resources have only a partial view of the situation. Seeing that authorities are unaware of who will attempt to access the resources they are responsible for and that users do not know in advance if the resources they are looking for exist, the problem becomes trickier. Therefore, before SOC becomes reality, there is a number of challenging issues that need to be addressed including among other things service security and service composition. What we are mainly concerned with here is the mode in which access to web services by potential users should be elaborated.

Service security. Before granting access to the resources they are responsible for, services set their security policies and describe the conditions under which such-and-such resource can be legally used. Seeing that services do not know in advance the users who will attempt to use them, they interact with these potential users and with other services via cryptographic protocols in order to obtain their credentials and to characterize their rights. Hence, cryptographic protocols restrict the availability and use of a service and affect its access control mechanism. Any service, then, must take into account in its security policy considerations about the cryptographic protocols it uses and service security considers all relevant aspects of confidentiality, integrity, and availability relating to the issue of combining protocols and policies.

Security in composition. SOC allows for designing a pool of organizations that are able to export services to users and to cooperate by composing services over networks. Hence, services are independent computational elements that can be composed in order to build collaborating applications distributed within and across organizational boundaries. Any service, then, must take into account in its security policy considerations about the ways in which it accepts to be composed with other services and service composition addresses the situation when a user request can only be satisfied by suitably combining available services.

2.1.2 Cryptographic protocols and Web services

Within the context of decentralized access control in a distributed environment, several authorities control accesses of subjects to objects. In order to ensure that all accesses are authorized, messages specifying the access rights of subjects to objects are exchanged between the different authorities. Given that these messages are conveyed over insecure networks, it is necessary to use cryptographic protocols capable of providing both data confidentiality and data integrity.

Since messages exchanged between Web services are written in XML, our first goal will be to extend the existing work on cryptographic protocols to protocols written in XML. This is necessary to take into account flaws stemming from an improper parsing of a XML message.

However the work on Web services only starts when considering the impact of its security policy on the execution of a service and, conversely, the changes in the policy that result from the execution of a service. There has been considerable research on the use of access control mechanisms and cryptographic protocols. The innovative challenge of our research work is to develop general techniques and methods for the combination and integration of access control mechanisms and cryptographic protocols and to implement automatic methods of validating Web services within the context of decentralized access control in a distributed environment.

A related issue is to analyze security properties of services involving several different cryptographic protocols. Very few works have investigated the problem [34, 2]. More generally it remains a challenge to modularly verify services composed from several security sub-services, protocols, access control managers...

Even more challenging is the goal of automated composition (orchestration) of security services in order to obtain Web services that satisfy some specified security properties.

2.1.3 Composition of Web services

The notion underlying the service oriented computing architecture is one of a distributed library of applications. As it is the case for libraries of functions in programming languages the goal is to provide a potential

user with elements that will be combined in the same way that functions in a library are combined to build more complex functions.

The similarity with a library stops here since, in contrast with a library that should provide a stable interface, some services may appear or disappear at any moment. To cope with this dynamic environment one is interested in building tools that automatically *compose* a new service from services that are currently available. This is done using *e.g.* the WSBPEL language to express the functionalities a service provide and by relying on a broker to semantically compose services in order to obtain one new service offering the functionalities desired by a customer.

Current approaches to composition of services do not take into account the security policy of services. This may lead to the construction of a non-executable new service. Our first goal toward the incorporation of security policies into composition will be to consider the composition of security services provided by authentication and authorization authorities. Then we will consider the problem of integration of security policies into a composition algorithm.

2.2 Modeling of security policy for Web services

The ability of services to interact with their clients and other services is one of the main purpose of SOC. This interoperability necessitates to describe what services and clients are allowed to do, that is, their security policies. Usually, security policies define the constraints that define what “secure” means for computer systems. Regarding services, the specificities of their functioning entail that standard modes of access control cannot cope with the requirements for ensuring confidentiality and integrity of data and communications. Hence, a new mode of access control has to be elaborated: service-based access control. This new mode must be able to deal with specific aspects of SOC like the notions of obligation, prohibition, and delegation. It must also be able to treat the problems brought up by the combination and integration of cryptographic protocols with security policies. What access control mode fits the needs of SOC?

DAC? We believe that discretionary access control (DAC) does not enable the expression of security policies that fit the basic needs for enforcing data confidentiality and data integrity in web services, for the simple reason that, within the context of decentralized access control in a distributed environment, services do not know in advance who will try to interact with them whereas clients do not know in advance if the service they want to use already exists. Authentication of clients and services is organized via the notion of credentials establishing the identities principals claim to have and via the notion of assertions conveying information about principals. An access control mode that fits the need of SOC must take into account the fact that the interaction between services and clients is preceded by an authentication step where credentials and assertions are exchanged through the use of cryptographic protocols in order to provide trust.

MAC? Mandatory access control (MAC) is based on the possibility to limit the actions users can take according to their security clearances and to the security levels of the objects they try to access to. This model corresponds to military-style classifications where clearances of users are rarely updated and levels of objects do not changed once they have been created. It has influenced the development of many other models. Nevertheless, within the framework of SOC, as new services and clients are constantly added over the networks, there can be no consensus about the nature of clearances and levels that services and clients must use to classify the objects they are responsible for. Hence, the use of MAC in web services seems problematic.

RBAC? ORBAC? The case of RBAC and the case of ORBAC are not better than the case of MAC. In RBAC, access is granted to a user on the basis of its role in an organization. Seeing that services are autonomous and independent software artifacts, there is no hope to base their security policies on a fixed and well-defined set of roles. Moreover, RBAC does not tackle the notions of obligation and delegation that appears to be essential with respect to the situation when a client request can only be satisfied by suitably combining available services. At first sight, ORBAC seems really suited to fit the needs of SOC, seeing that,

in ORBAC, each organization defines its own security policy whereas, in SOC, each service elaborates its own rules for granting access to the resources it is responsible for. Moreover, the notion of obligation is already built in ORBAC whereas it does not seem to be difficult to integrate in it the notion of delegation. Nevertheless, seeing that the manner in which the cryptographic protocols are executed between a composite service and its clients acts upon the accesses that will be granted by the service to its clients and, vice versa, the rights of clients over resources in a composite service influence the way the service interacts with its clients, ORBAC is unsuited to constitute the access control mode of SOC.

Service-based access control. In this project, we intend to overcome the limitations of the above-mentioned modes by considering the concept of service as central. In service-based access control (SBAC), the specification of a security policy will be completely parametrized by the service so that it will be possible to handle simultaneously and independently several security policies associated with different services. Roughly speaking, a service can be seen as an organized group of resources. To control access to its resources, a service defines its security policy, i.e. a set of rules which determine the credentials and the assertions that potential clients must produce in order to be granted access. Credentials may specify, for example, the name of a potential client and one of its public keys. Assertions are, for example, pieces of data produced by recognized authorities that concern attribute informations about a client or authorization data applying to a client with respect to such-and-such resources. There are significant application requirements that SBAC will address. Most of these requirements are related to the notions of obligation, prohibition, delegation, session, and hierarchy.

Obligation and prohibition. The principal motivation behind SBAC will be the ability to specify, in security policies, the ways in which a service accepts to be used by clients or accepts to be composed with other services that collaborate together to fulfil their clients. Access permissions that are assigned to clients or other services reflect policy decision. It appears that services must also take decisions regarding the obligations and the prohibitions they would like to assign to their clients and to the other services. Let us consider two examples. When a service is used by a client, it may put, as a prerequisite, a preliminary condition demanding, for reasons that concern its confidentiality policy, that the client be strictly obliged to do such-and-such action. When a service is requested to form a coalition of services, it may put, as a prerequisite, a preliminary condition demanding, for reasons that concern its integrity policy, that the component services of the coalition be strictly prohibited to do such-and-such action.

Delegation. When a service enables a client to access the desired resources, it may need, for reasons that concern its availability policy, to delegate one part of the task it has undertaken to perform. Hence, security policies must be able to treat this notion of delegation. The specificities of web services for what concerns confidentiality and integrity implies that their security policies must also address the intricacies raised by the combination and integration of protocols and policies for determining the access rights of clients and for building up composite services. We may address in this respect the situation when a service delegates one part of its task to another service who requires more credentials and assertions than the service contacted in the first place.

Session and hierarchy. In SBAC, security policies will not directly grant access to such-and-such client. They will rather define, for each of their resources, the credentials and the assertions that clients must produce in order to be granted access. A client will obtain access by producing these credentials and these assertions, in which case it will inherit all the accesses associated with these credentials and these assertions. Of course, it will be possible to refine this model by including the concepts of session and hierarchy. Within a session, a client will be obliged to produce its credentials and its assertions only once. The hierarchy of credentials and the hierarchy of assertions will be useful because accesses will be inherited through this hierarchy. This will simplify the security policy specification.

Safety. A security policy specifies the conditions under which a WS accepts to grant accesses to the resources it is responsible for. These conditions are defined by the protection state of the WS, i.e. its access matrix which characterizes the rights of its clients with respect to its resources, and the protection system of the WS, i.e. its set of rules which characterizes the ways the protection state may change. As a WS execute the operations allowed by the rules of its protection system in connection with the credentials and assertions it has received from potential clients or from other services, its protection state changes. Given a security policy, how can we determine if it is secure? This question embodies in fact two issues. The first issue is the definition of “secure”. The second issue is the elaboration of an algorithm deciding whether a security policy is secure. Within the context of the primary abstraction mechanism in computer security studied by Harrison, Ruzzo, and Ullman [37], a security policy is defined to be secure when its protection system can never add a specific right to its protection state. Thus, safety refers here to the leaking of rights in the protection state by the protection system. It has been proved that, within the framework studied by [37], the safety issue is undecidable. Restricting the number of conditions or the number of operations in the rules defining a protection system makes the safety issue decidable [37, 36, 35]. Within the context of services, the safety problem can be defined in a similar way: can the protection system of a WS adds a specific right to its protection state after executing the operations allowed by its rules? Nevertheless, the use of permissions, obligations, and prohibitions in policies as well as the combination and integration of policies and protocols introduce subtleties that complicate the solution of the safety issue. First, it is of the utmost importance to be able to decide whether the permissions, the obligations, and the prohibitions that a given security policy entails are together consistent. A similar question has been addressed in [21]. Its solution within the context of services is still unfound. Second, we should also be able to decide whether the security policies of a given WS will remain consistent whatever operations allowed by the rules of its protection system are executed. This problem is surely undecidable, all the more so since a WS interacts with its potential clients and with other services via cryptographic protocols and since the execution of these protocols may affect its protection state or its protection system. Hence, we should find restrictions on security policies that make this problem decidable. Third, we should find algorithms able to decide whether a WS can step-by-step modify its protection state in such a way that it will contain a given specific right. This problem is surely undecidable as well and we should find restrictions on security policies that make it decidable.

2.3 Protocols parameterized by security policies

The core of our work will be the combination of policies and protocols. This goal is a long-term effort and several difficulties need to be solved before its resolution. We first examine the key issues, and we then detail our research program, which is to consider first protocols (Web services without a security policy), then static Web services and finally to consider a dynamic system where available services may change over the time.

2.3.1 Information protection: from access control to cryptographic protocols

A common example. Let us consider the simple example of paper reviewing for a conference. The PC Chair assigns papers to PC members. They can report on their papers or nominate sub-reviewers, and so on. Hence a delegation chain is generated, and when a report is sent back to PC Chair, the fact that the review can be entered in the reports database should be deduced from the policy. Confidential information should be communicated encrypted with suitable cryptographic primitives. This shows a typical situation where authorization decisions are intermingled with other operations such as protocol message exchanges. Moreover in general we have to consider composition of several interacting services. Due to the fact that composed web services cross organizational boundaries, access control management in SOC is particularly challenging.

Case of Web services. The WSBPEL language has become a de facto standard for Web service composition. However it does not yet address the security aspects. The WSBPEL specification recommends to implement WS-Security to ensure reliable message exchanges. This is obtained by systematic use of digital

signatures, security tokens and various kind of encryption mechanisms. Therefore the secure communication of different services in a business transaction can be achieved by cryptographic protocols. On the other hand the security of Web service compositions should also be expressed in terms of clear and coherent access control policies. This is a difficult issue since a permission given to a process or subject may depend on specific context information.

Goals. Our objectives are to express policies and protocols in a homogeneous setting so that we will be able to enforce authorization when executing the protocols. This should allow us to check statically whether the protocols are executable in conformance with the specified policy. Moreover this analysis should be extended to detect and prevent the potential attacks of a Dolev Yao intruder, both in passive and active case. This logical setting will also permit to use policy expressions to be checked dynamically at service invocation, in order to enforce conformance within a service orchestration. Our aimed contribution here will be the formal and algorithmic treatment of policy composition in order to identify decidable and tractable situations.

2.3.2 Verification of XML cryptographic protocols

We first consider simple Web service, that is applications exchanging request/response messages under a specified XML format e.g. SOAP envelope. To allow the verification of properties like authentication and security, one must provide a formal framework and a translation of Web service into this model. Furthermore this model must encompass the behavior of a malicious attacker. In the flavor of [7], we shall describe web services as a process in a process calculus similar to the calculus of [3], which derives from the spi-calculus a variant of π -calculus designed for cryptographic protocol verification. The formalism will be enriched to embed the XML format required by SOAP envelope and we shall use the classical Dolev-Yao model for an attacker who can intercept all messages, decrypt messages with a corrupted key, replay messages and forge new messages from previous messages. Due to the lack of type verification by web services, there exists type attacks directly related to the XML format. Contrary to [8] where the verification relies on a translation to Horn clauses, we aim, assuming that the number of participants and sessions is bounded, at efficient decision procedures able to verify simple web services. We shall also emphasize trace analysis which is mandatory to get a comprehensive analysis of the logical failures that will be exhibited by the symbolic analysis. Since this model works for a finite number of participants and sessions, we shall study its extension to the general case of an unbounded number of sessions and participants, for which we shall study sensible restrictions that allow decidability results. We shall study also how classical attacks like dictionary attacks can be defined and verified in this model. This framework is intended to be a first proposal on which further work will be done such as investigating composition of web services.

2.3.3 Combining policies and protocols

Expression. Our first goal will be to investigate how one can *express* access-control policies in the context of message-passing formalisms, thus extending cryptographic protocols. Some recent works by Microsoft Research partner [31] have shown how to incorporate policies in the spi calculus by representing them as logic programs. This will be a starting point for us. Basically we will define too a Datalog-like language to express authorization and we will embed this logic in the process-oriented formalisms that the other partners [3, 38] have been developing for several years to specify and verify cryptographic protocols. This Datalog-like language will be based on the access-control mode that will result from the work on security policies. Classically we will deduce that a request is valid if it is entailed by the combination of the policy and the set of available credentials.

Execution without attacker. Then we will investigate decision methods to check whether an execution of a service combining message exchanges with policies, assuming all honest participants respect the policy. We should be able to propose automatic check that a service has the credentials to perform an operation at each step, that each step satisfy the policy that has been specified declaratively in the Datalog framework.

We aim at using the type information to study orchestration of services. Although the work [31] was also, focused on typing, no decidability results were given. Type constraints on the type of messages can express that the composition of the relevant security policies is feasible and type inclusion can express that a type policy, here the behavior we should expect for the composition, is entailed by another one, the composition of the security policies. This can also be applied to more general policies (also called commitments), *i.e.* those defining the business behavior that has to be ensured by all parties. When two or more parties interact, they first agree on subsequent interactions. An objective will be to use the typing to perform both static and dynamic checking in order to prevent undesired behavior and to ensure the consistency of the execution.

Finally let us mention that this logical approach might be also interesting to compute by logical abduction what are the credentials required or policies required by a calling or called service for invoking a component service, or what are the policies requirements that a service should satisfy in order to be invoked [39].

Execution in presence of an attacker. The final step of analysis of a combination of protocols and security policies will be to extend the results that will be obtained on the analysis of XML protocols to consider the case of an intruder attacking a Web service. We aim to consider a standard Dolev Yao model of intruder but we will start with the simpler case where the intruder can only eavesdrop communications. If some decidability result can be derived then we will consider the general case of an active intruder. This should require a significant extension of the recent decidability results for cryptographic protocols. We have already identified several possibilities to solve this problem.

The most mature one is to use a language for security policies relying on a type system for XML schemas in the flavor of [24]. This language will permit us to enrich the type-system of our previous process-calculus for verifying protocols in XML format. This will combine a type checking algorithm relying on tree-automata for unranked trees and the symbolic computation that allows to state the reachability of some undesired state (e.g. disclosing a secret or stating that some message is not authenticated).

Another possibility will be to express security policies in a formalism similar to the *oracle rules* technique we have introduced in [16]. These special protocol rules have been employed to approximate the effect of protocol steps that can be fired an unbounded number of times. We believe that similar rules can be applied to specify and compute access rights dynamically. For instance, in the former example of a conference reviewing process, the validity of an authorization is a consequence of an unbounded number of delegations. We also believe that the associated decidability result for cryptographic protocols will also hold for analysis of Web services.

2.3.4 On-the-fly policy checking

According to [42] the largest challenge of implementing security for service-oriented computing is the fact that services being invoked dynamically it is impossible to predict who the service providers will be and especially which authorization they should have. Hence the security model and the executability and security analysis above have to be refined and extended in order to adjust to a rapidly changing environment allowing for on-the-fly evaluation of authorizations according to the policy of the discovered service. For instance, if privacy is the property to be ensured, then the new service invocation should not lead to generate message exchanges containing security tokens or information whose combination with other services may lead to identity disclosure. Whether the called service is fully or only partially authenticated will lead either to further trust negotiation between the services or to restrictions on calling service's accesses. These mechanisms are still subject for investigation.

However the efficiency of the protocol analysis techniques we have developed in previous project [51], discovering potential flaws for large protocols in fractions of seconds, show that it should be possible to design procedure to monitor dynamic service compositions with respect to policies of component services. These monitors can be automatically defined as additional services and linked to the service composition as proposed in [6] for QoS parameters.

2.4 Synthesis of Web services

2.4.1 Semantic composition of services

The Web is moving from having just human-readable information to being a world-wide network of cooperating services. The underlying assumption being that while a single Web service may be useful on its own, only the automated discovery, composition and usage of these services will permit to provide the end user with a more readily access to highly sophisticated added-value services. For example, one may consider a service that automatically and dynamically retrieves hotel bookings services, and is able to interact with each of these services in order to present the user a small set of hotels meeting its criteria on the localization, the price and the check-in and check-out dates.

This evolution lead to the adoption of standard languages of description of resources available on the Web. The *Web Services Description Language* (WSDL) [63] is concerned with the description of the implementation of a web service in terms of exchanged messages. At a higher level, the *Resource Description Framework* (RDF) [61] and the *Web Ontology Language* (OWL) [60] aim at describing the service offered by an application in terms of classes of resources and relations between these classes.

While these description languages may in principe cover any kind of data, it has been useful to rely on a more focused formalism for some specific domains. The most advanced domain concerns mathematics for which the OpenMath standard [56] already permits to express theorems, proofs, calculus, etc. A signature that defines a set of operators is defined (*e.g.* the application of a function). This signature can then be extended with the addition of new symbols together with their properties. These properties are expressed by logical formulas and equalities.

2.4.2 Our goal

While the automatic synthesis of complex Web services is out of the scope of this proposal, we plan to work on the composition of security-related Web services. Authentication and authorization authorities may indeed be seen as applications delivering well defined services in the form of assertions. We plan to give a formal semantics of these services and to apply a recent combination result obtained in the context of the analysis of cryptographic protocols [17] to compose security-related Web services.

We plan to work along the lines of OpenMath: First we will define a set of basic operators describing the security properties relevant to web services. Second we will express the security policy of web services as functions on this signature. Finally and given a client and the credentials that this client has to obtain in order to access a service, we will compose these functions in order to automatically design a sequence of transactions between this client and several authorities.

This composition will ideally be computed during the resolution of reachability problems similar to those handled in [17]. In the case these reachability problems are not decidable we will rely on generic description logic based procedures to try to compose this sequence of deductions.

2.4.3 Impact of this work

The immediate consequence of this work will be the possibility of implementing a user-transparent security infrastructure in which on the basis of the credentials needed to grant the access of a resource to a client, assertions will be exchanged between authorities and this client to gather these credentials. The order of transactions to perform between the client and asserting authorities will be derived automatically from the needed credentials.

A more far-sighted consequence may be the possibility, at least with respect to some well-identified domains for which synthesis procedures may be given, to answer complex requests from an end-user such as the ordering of a book with minimal price or the access to a computation center that provides calculus primitives adapted to a calculus requested by the user.

3 Intended results

3.1 Expression of security policy

First Year. We will explore the mode of access control that fits the needs of SOC best, service-based access control. Then, we will define what it means for a WS to be safe.

Second year. Seeing that security policies will be defined in terms of permissions, obligations, and prohibitions, our aim will be to elaborate methods for automatically proving that the security policy of a given WS is and will remain consistent.

Third year. Seeing that the interactions of services, via cryptographic protocols, with their clients and with other services may change their security policies, we will consider the decidability/complexity of the safety problem, i.e. the problem of determining whether or not the security policy of a given WS satisfies such-and-such safety property.

3.2 Combination of security policy and Web services

First Year. We shall provide an updated calculus for protocol verification that embeds XML format and the relevant operations on XML messages. Our previous work on symbolic analysis of cryptographic protocols should extend smoothly to this case. It will provide a more precise idea of the interaction between the real format use in services and the security issues related to cryptographic protocol.

Second year. We shall provide a more general calculus with a type calculus for expressing security policies that relies for XML schemas or a variant of XML schemas. This point is more challenging since the language and semantics that we shall use should be amenable to automated verification. This raises issues concerning theoretical results (decidability of our process) and practical ones (efficiency).

Third year. We aim at providing results for the composition of Web services in this framework, including composition and verification of web services and their security policies. Composition and modularity of verification is a major issue since little is known in this area except that new problem happens when we combine services that works well when they are isolated.

3.3 Synthesis of Web services

First year. We shall provide a fonctionnal description of services provided by authentication and authorisation authorities. This implies the definition of operators and properties for assertions defined in the SAML language.

Second year. We shall provide an automatic composition algorithm for security based services. In conjunction with the expression of security policies into Web services, this will provide an algorithm for automatic construction of a sequence of interaction with services to provide an agent with the credentials needed to use an arbitrary service.

Third year. We shall extend the preceding algorithm to other specific areas. Our main focus will be the expression of computational services with an emphasis on the library provided by GRID systems for matrix computation.

4 References

- [1] A. Abou El Kalam, R. El Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miège, C. Saurel, and G. Trouessin. Organization based access control. In *Proceedings of the 4th International Workshop on Policies for Distributed Systems and Networks*. IEEE Computer Society Press, 2003.
- [2] J. Alves-Foss. Multi-protocol attacks and the public-key infrastructure. In *Proc. 21st National Information Systems Security Conference, Arlington*, pages 566–576, 1998.
- [3] R. Amadio, D. Lugiez, and V. Vanackère. On the symbolic reduction of processes with cryptographic functions. *Theoretical Computer Science*, 1(290), 2002.
- [4] M. Backes, B. Pfitzmann, and M. Waidner. Security in business process engineering. In Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, and Mathias Weske, editors, *Business Process Management, International Conference, BPM 2003, Eindhoven, The Netherlands, June 26-27, 2003, Proceedings*, volume 2678 of *Lecture Notes in Computer Science*, pages 168–183. Springer, 2003.
- [5] P. Balbiani. Constitution et développement d’une logique de modalités aléthiques, déontiques, dynamiques et temporelles en vue de la formalisation du raisonnement sur les actions et sur les normes. In *Actes des Troisièmes Journées Francophones – Modèles Formels de l’Interaction (MFI’05)*, pages 23–34, May 2005.
- [6] L. Baresi, C. Ghezzi, and S. Guinea. Smart monitors for composed services. In *Proceedings of the 2nd international conference on Service oriented computing (ICSOC’04)*, pages 193–202. ACM Press, 2004.
- [7] K. Bhargavan, C. Fournet, and A. D. Gordon. A semantics for web services authentication. In Neil D. Jones and Xavier Leroy, editors, *Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2004, Venice, Italy, January 14-16, 2004*, pages 198–209. ACM, 2004.
- [8] K. Bhargavan, C. Fournet, A. D. Gordon, and R. Pucella. Tulafale: A security tool for web services. In F. S. de Boer, M. M. Bonsangue, S. Graf, and W. P. de Roever, editors, *Formal Methods for Components and Objects, Second International Symposium, FMCO 2003, Leiden, The Netherlands, November 4-7, 2003, Revised Lectures*, volume 3188 of *Lecture Notes in Computer Science*, pages 197–222. Springer, 2003.
- [9] S. Bhiri, O. Perrin, and C. Godart. Ensuring required failure atomicity of composite web services. In *in Proceedings of the 14th international conference on World Wide Web, WWW 2005*, pages 138–147. ACM, 2005.
- [10] M. Bishop. *Computer Security: Art and Science*. Addison-Wesley, 2003.
- [11] B. Blanchet. Automatic verification of cryptographic protocols: a logic programming approach. In *Proceedings of the 5th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming*, pages 1–3, 2003.
- [12] D. Bolignano. Towards the formal verification of electronic commerce protocols. In *Proceedings of the IEEE Computer Security Foundations Workshop*, pages 133–146. IEEE Computer Society, 1997.
- [13] M. Boreale. Symbolic trace analysis of cryptographic protocols. In *Proceedings of the 28th ICALP’01*, LNCS 2076, pages 667–681, Berlin, 2001. Springer-Verlag.
- [14] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP Decision Procedure for Protocol Insecurity with XOR. In *Proceedings of the Logic In Computer Science Conference LICS’03*, pages 261–270, June 2003. Long version available as Technical Report RR-4697, INRIA, France.

- [15] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the Security of Protocols with Diffie-Hellman Exponentiation and Products in Exponents. In *Proceedings of the Foundations of Software Technology and Theoretical Computer Science, FSTTCS'03*, Lecture Notes in Computer Science. Springer, December 2003. Long version available as Christian-Albrecht Universität IFI-Report 0305, Kiel (Germany).
- [16] Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani, and L. Vigneron. Extending the Dolev-Yao Intruder for Analyzing an Unbounded Number of Sessions. In M. Baaz, editor, *Computer Science Logic (CSL 03) and 8th Kurt Gödel Colloquium (8th KCG)*, volume 2803 of *Lecture Notes in Computer Science*, Vienna, Austria, August 2003. Springer.
- [17] Y. Chevalier and M. Rusinowitch. Combining intruder theories. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, 2005.
- [18] Y. Chevalier and L. Vigneron. A Tool for Lazy Verification of Security Protocols (short paper). In *Proceedings of ASE-2001: The 16th IEEE Conference on Automated Software Engineering*, pages 373–376, San Diego (CA), November 2001. IEEE CS Press.
- [19] Y. Chevalier and L. Vigneron. Towards Efficient Automated Verification of Security Protocols. In *Proceedings of the Verification Workshop (VERIFY'01) (in connection with IJCAR'01)*, Università degli studi di Siena, TR DII 08/01, pages 19–33, Siena (Italy), June 2001.
- [20] Y. Chevalier and L. Vigneron. Automated Unbounded Verification of Security Protocols. In E. Brinksma and K. Guldstrand Larsen, editors, *14th International Conference on Computer Aided Verification, CAV'2002*, volume 2404 of *Lecture Notes in Computer Science*, pages 324–337, Copenhagen (Denmark), July 2002. Springer.
- [21] L. Cholvy and F. Cuppens. Analyzing consistency of security policies. In *IEEE Symposium on Research in Security and Privacy*. IEEE, 1997.
- [22] E. Cohen. TAPS: A first-order verifier for cryptographic protocols. In *PCSFV: Proceedings of The 13th Computer Security Foundations Workshop*. IEEE Computer Society Press, 2000.
- [23] F. Cuppens, N. Cuppens-Boulahia, and T. Sans. Nomad: A security model with non atomic actions and deadlines. In *Proceedings of the Computer Security Foundations Workshop (CSWF'05)*, pages 186–196. IEEE, June 2005.
- [24] S. Dal Zilio and D. Lugiez. XML Schema, tree logic and sheaves automata. In *Proceedings of the Conference on Rewriting Techniques and Applications (RTA)*, 2003.
- [25] S. Dal Zilio, D. Lugiez, and C. Meyssonier. A logic you can count on. In *31st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, 2004.
- [26] G. Denker, J. Millen, and H. Rueß. The CAPSL integrated protocol environment. Technical report, SRI International, October 2000.
- [27] D. Denning. *Cryptography and Data Security*. Addison-Wesley, 1982.
- [28] M. Deubler, J. Grünbauer, J. Jürjens, and G. Wimmel. Sound development of secure service-based systems. In M. Aiello, M. Aoyama, F. Curbera, and M. P. Papazoglou, editors, *Service-Oriented Computing - ICSOC 2004, Second International Conference, New York, NY, USA, November 15-19, 2004, Proceedings*, pages 115–124. ACM, 2004.
- [29] B. Donovan, P. Norris, and G. Lowe. Analyzing a library of security protocols using casper and fdr. In *Proceedings of the Workshop on Formal Methods and Security Protocols*, 1999.
- [30] D. Ferraiolo, D. Kuhn, and R. Chandramouli. *Role-Based Access Control*. Artech House, 2003.

- [31] C. Fournet, A. G. Gordon, and S. Maffei. A type discipline for authorization policies. In *14th European Symposium on Programming (ESOP)*, volume 3444 of *Lecture Notes in Computer Science*, pages 141–156. Springer, 2005.
- [32] X. Fu, T. Bultan, and J. Su. Analysis of interacting bpel web services. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 621–630. ACM Press, 2004.
- [33] J. Grünbauer, H. Hollmann, J. Jürjens, and G. Wimmel. Modelling and verification of layered security protocols: A bank application. In S. Anderson, M. Felici, and B. Littlewood, editors, *Proceedings of the 22nd International Conference on Computer Safety, Reliability, and Security, (SAFECOMP'03)*, volume 2788 of *Lecture Notes in Computer Science*, pages 116–129. Springer, 2003.
- [34] J. D. Guttman and F. J. Thayer. Protocol independence through disjoint encryption. In *Proceedings of the Computer Security Foundations Workshop (CSFW'00)*, pages 24–34, 2000.
- [35] M. Harrison. Theoretical issues concerning protection in operating systems. *Advances in Computers*, 24:61–100, 1985.
- [36] M. Harrison and W. Ruzzo. Monotonic protection systems. In *Foundations of Secure Computation*, pages 337–365. Academic Press, 1978.
- [37] M. Harrison, W. Ruzzo, and J. Ullman. Protection in operating systems. *Communications of the ACM*, 19:461–471, 1976.
- [38] F. Jacquemard, M. Rusinowitch, and L. Vigneron. Compiling and Verifying Security Protocols. In M. Parigot and A. Voronkov, editors, *Logic for Programming and Automated Reasoning*, volume 1955 of *Lecture Notes in Computer Science*, pages 131–160, St Gilles (Réunion, France), November 2000. Springer-Verlag.
- [39] H. Koshutanski and F. Massacci. Interactive credential negotiation for stateful business processes. In P. Herrmann, V. Issarny, and Simon Shiu, editors, *Third International Conference on Trust Management (iTrust'05)*, volume 3477 of *Lecture Notes in Computer Science*, pages 256–272. Springer, May 2005.
- [40] P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for AC-like equational theories with homomorphisms. In Jürgen Giesl, editor, *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA)*, volume 3467 of *Lecture Notes in Computer Science*, pages 308–322, Nara, Japan, April 2005. Springer.
- [41] B. Lampson. Protection. *Operating Systems Review*, 8(1), 1974.
- [42] K. Leune, M. P. Papazoglou, and W.-J. van den Heuvel. Specification and querying of security constraints in the efsoc framework. In *Proceedings of the 2nd international conference on Service oriented computing (ICSOC'04)*, pages 125–133. ACM Press, 2004.
- [43] G. Lowe. Casper: a compiler for the analysis of security protocols. *Journal of Computer Security*, 6(1):53–84, 1998. See also <http://www.mcs.le.ac.uk/~gl7/Security/Casper/>.
- [44] K. D. Lux, N. L. Bhattad, and C. A. Gunter. Wsemail: Secure internet messaging based on web services. In *Proceedings of the 3rd International Conference on Web Services (ICWS'05a)*. IEEE, July 2005. to appear.
- [45] C. Meadows. Open issues in formal methods for cryptographic protocol analysis. In *Proceedings of the DARPA Information and Survivability Conference and Exposition: DISCEX 2000*, pages 237–250. IEEE Computer Society Press, January 2000.
- [46] Microsoft Research. Samoa project web page. <http://research.microsoft.com/projects/samoa/>, May 2005.

- [47] J. Millen and V. Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *Proceedings of the 8th ACM Conference on Computer and Communication Security*, pages 166–175, November 2001.
- [48] J.C. Mitchell, M. Mitchell, and U. Stern. Automated analysis of cryptographic protocols using murphi. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 141–153, 1997.
- [49] OASIS. Oasis web services business process execution language. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel, june 2004.
- [50] L. C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6(1):85–128, 1998.
- [51] Cassis Project. The casrul software suite. <http://www.loria.fr/equipes/cassis/software/casrul/>.
- [52] A. W. Roscoe and M. Goldsmith. The perfect “spy” for model-checking cryptoprotocols. In *Proceeding of DIMACS Workshop on Design and Formal Verification of Crypto Protocols*, 1997.
- [53] M. Rusinowitch and M. Turuani. Protocol Insecurity with Finite Number of Sessions and Composed Keys is NP-complete. *Theoretical Computer Science*, 299:451–475, April 2003.
- [54] G. Salaün, L. Bordeaux, and M. Schaerf. Describing and reasoning on web services using process algebra. In *Proceedings of the IEEE International Conference on Web Services (ICWS’04), June 6-9, 2004, San Diego, California, USA*, pages 43–. IEEE Computer Society, 2004.
- [55] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. Role-based access control models. *IEEE Computer*, 29(2), 1996.
- [56] The OpenMath Society. The openmath standard 2.0. <http://www.openmath.org/cocoon/openmath/standard/om20-2004-06-30/index.%html>, 2004.
- [57] M. Solanki, A. Cau, and H. Zedan. Augmenting semantic web service descriptions with compositional specification. In *WWW ’04: Proceedings of the 13th international conference on World Wide Web*, pages 544–552. ACM Press, 2004.
- [58] W3 Consortium. Xml encryption syntax and processing. <http://www.w3.org/TR/xmlenc-core/>, December 2002.
- [59] W3 Consortium. Xml signature syntax and processing. <http://www.w3.org/TR/xmlsig-core/>, February 2002.
- [60] W3 Consortium. OWL web ontology language overview. <http://www.w3.org/TR/2004/REC--owl--features--20040210/>, February 2004.
- [61] W3 Consortium. Resource description framework (RDF): Concepts and abstract syntax. <http://www.w3.org/TR/rdf-concepts/>, February 2004.
- [62] W3 Consortium. SOAP version 1.2. <http://www.w3.org/TR/soap/>, june 2004.
- [63] W3 Consortium. Web services description language (wsdl) version 2.0. <http://www.w3.org/TR/wsd120/>, May 2005.

5 Bibliographical references of the researchers involved in the project

5.1 Laboratoire d'Informatique Fondamentale (LIF, UMR 6166)

Silvano Dal Zilio

- [1] S. Dal Zilio and A. D. Gordon. Region analysis and a pi-calculus with groups. *Journal of Functional Programming*, 12(3):229–292, May 2002.
- [2] S. Dal Zilio and D. Lugiez. XML Schema, tree logic and sheaves automata. In *Proceedings of the Conference on Rewriting Techniques and Applications (RTA)*, 2003.
- [3] S. Dal Zilio, D. Lugiez, and C. Meyssonnier. A logic you can count on. In *Proceedings of the 31st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, 2004.

Denis Lugiez

- [1] R. Amadio, D. Lugiez, and V. Vanackère. On the symbolic reduction of processes with cryptographic functions. *Theoretical Computer Science*, 1(290), 2002.
- [2] S. Dal Zilio and D. Lugiez. XML Schema, tree logic and sheaves automata. In *Proceedings of the Conference on Rewriting Techniques and Applications (RTA)*, 2003.
- [3] S. Dal Zilio, D. Lugiez, and C. Meyssonnier. A logic you can count on. In *Proceedings of the 31st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, 2004.
- [4] P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for AC-like equational theories with homomorphisms. In J. Giesl, editor, *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA)*, volume 3467 of *Lecture Notes in Computer Science*, pages 308–322, Nara, Japan, April 2005. Springer.

5.2 Laboratoire Lorrain de Recherche en Informatique et Automatique (LORIA, UMR 7503)

Michael Rusinowitch

- [1] M. Rusinowitch and M. Turuani. Protocol Insecurity with Finite Number of Sessions and Composed Keys is NP-complete. *Theoretical Computer Science A*, 2003, vol 299/1-3, pp 451 - 475.
- [2] Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani. Deciding the Security of Protocols with Diffie-Hellman Exponentiation and Products in Exponents. In *Conference on the Foundations of Software Technology and Theoretical Computer Science (FSTTCS 03)*. December 15–17, 2003. Indian Institute of Technology, Bombay. Mumbai, INDIA
- [3] Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani, L. Vigneron. Extending the Dolev-Yao Intruder for Analyzing an Unbounded Number of Sessions. In *Proceedings of the Conference on Computer Science Logic CSL'03*, Vienna, Austria, 25th - 30th August 2003. Series: Lecture Notes in Computer Science. Vol. 2803.
- [4] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP Decision Procedure for Protocol Insecurity with XOR. *Theoretical Computer Science*, 338(1-3):247-274, June 2005 (long version of a LICS 2003 paper)
- [5] Y. Chevalier and M. Rusinowitch. Combining Intruder Theories. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, to appear, July 2005

Laurent Vigneron

- [1] Y. Chevalier and L. Vigneron. Strategy for Verifying Security Protocols with Unbounded Message Size. *Journal of Automated Software Engineering*, 2004, April, volume 11, number 2, pp141-166, Kluwer Academic Publishers.
- [2] Y. Chevalier and L. Vigneron. Automated Unbounded Verification of Security Protocols. In *Proceedings of the 14th International Conference on Computer Aided Verification (CAV'02), Lecture Notes in Computer Science*. Springer-Verlag, 2404:324–337, 2002. Also available as Technical Report 4369, LORIA, Nancy (France).
- [3] F. Jacquemard, M. Rusinowitch, and L. Vigneron. Compiling and Verifying Security Protocols. In *Proceedings of 7th Conference on Logic for Programming and Automated Reasoning*, volume 1955 of *Lecture Notes in Artificial Intelligence*, pp. 131–160. Springer-Verlag, 2000.
- [4] L. Bachmair, A. Tiwari and L. Vigneron., Abstract Congruence Closure, *Journal of Automated Reasoning*, 2003, 31(2):129–168, Kluwer Academic Publishers.

Olivier Perrin

- [1] S. Bhiri, O. Perrin, C. Godart. Ensuring Required Failure Atomicity of Composite Web services. In *Proceedings of the World-Wide Web Conference WWW 2005*, Japan, May 2005.
- [2] S. Bhiri, C. Godart, O. Perrin. Reliable Web services composition using a transactional approach In *Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service*. Hong Kong, March 2005.
- [3] S. Bhiri, C. Godart, O. Perrin. A Transaction-oriented Framework for Composing Transactional Web Services. In *Proceedings of the IEEE International Conference on Services Computing ICSC 2004*, Shanghai, China, IEEE digital library, Sep 2004, p. 654-663.
- [4] M. Rouached, O. Perrin, C. Godart. A Contract-based Approach for Monitoring Collaborative Web Services using Commitments in the Event Calculus. *submitted to WISE 05*.

5.3 Institut de Recherche en Informatique de Toulouse (IRIT, UMR 5505)

Philippe Balbiani

- [1] P. Balbiani. Eliminating unorthodox derivation rules in an axiom system for iteration-free *PDL* with intersection. *Fundamenta Informaticæ*, 56:211–242, 2003.
- [2] P. Balbiani. Constitution et développement d’une logique des modalités aléthiques, déontiques, dynamiques et temporelles en vue de la formalisation du raisonnement sur les actions et sur les normes. In *Actes des Troisièmes Journées Francophones – Modèles Formels de l’Interaction (MFI’05)*, pages 23–34, May 2005.
- [3] P. Balbiani. Logical approaches to deontic reasoning: from basic questions to dynamic solutions. *International Journal of Intelligent Systems*, to appear. 2005.
- [4] P. Balbiani and F. Cheikh. Safety problems in access control with temporal constraints. In *Mathematical Methods, Models and Architectures for Computer Networks Security Workshop (MMM-ACNS’05)*, to appear. September 2005.
- [5] P. Balbiani and D. Vakarelov. Dynamic extensions of arrow logic. *Annals of Pure and Applied Logic*, 127:1–15, 2004.

Fahima Cheikh

- [1] P. Balbiani and F. Cheikh. Safety problems in access control with temporal constraints. In *Mathematical Methods, Models and Architectures for Computer Networks Security Workshop (MMM-ACNS'05)*, to appear. September 2005.
- [2] F. Cheikh. Le problème de la disponibilité dans les systèmes de protection temporisée. Graduation thesis, Université Paul Sabatier Toulouse 3, June 2005.

Yannick Chevalier

- [1] Y. Chevalier and M. Rusinowitch. Combining Intruder Theories. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, to appear, July 2005
- [2] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP Decision Procedure for Protocol Insecurity with XOR. *Theoretical Computer Science*, 338(1-3):247-274, June 2005 (long version of a LICS 2003 paper)
- [3] Y. Chevalier and L. Vigneron. Strategy for Verifying Security Protocols with Unbounded Message Size. *Journal of Automated Software Engineering*, 11(2):141-166, April 2004. (long version of a CAV 2002 paper)
- [4] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the Security of Protocols with Diffie-Hellman Exponentiation and Products in Exponents. In *Proceedings of the Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS'03*, December 2003.
- [5] A. Armando et al. The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. In *Proceedings of the 17th International Conference on Computer-Aided Verification (CAV'05)*. Tool presentation, to appear.