CASENET: ONE YEAR LATER

Computer Aided solutions to SEcure electroNic commercE Transactions

Isaac Agudo
University of Malaga (Spain)
isaac@lcc.uma.es

Sigrid Gürgens
Fraunhofer SIT (Germany)
guergens@sit.fraunhofer.de

Javier Lopez
University of Malaga (Spain)
jlm@lcc.uma.es

Abstract This paper presents CASENET, a Fifth European Framework research project whose objectives are to develop and implement a tool-supported framework for the systematic specification, design and analysis of e-commerce and e-government transactions to produce protocols with proven security properties, and to assist in code generation for these protocols. The methodologies and tools developed by the project will enable the designer of an e-commerce or e-government application to generate a formal protocol specification with the desired security properties and be usable for the security analysis of protocols. After successful analysis, the methodologies and tools will assist in transforming the formal protocol specification into final code, provide test cases for testing the code with respect to the initial requirements and services for real-time auditing in order to check that the participants of a protocol act according to the description.

1. Introduction

The use of Internet, in particular in the form of world-wide-web and its wireless counterpart, has opened a whole new arena for electronic

commerce and electronic business. New services and the existing ones moved on to Internet with an improved quality have created large revenues.

The potential growth in e-business can, however, be held back by concerns about the security of the software systems involved. Internet is notorious for lack of security. Online e-commerce applications are susceptible to failures and exposed to active attacks when not properly designed and tested.

A common solution for securing electronic business is to employ cryptographic protocols (e.g., encryption, digital signature, authentication, identification, key management, etc.) at application levels. However, many cryptographic protocols are being and will be designed and/or implemented by "hackers", engineers oriented to application problems without appropriate methodologies at hand.

Systems complexity, in particular due to concurrency among systems, has been the main cause of design and/or implementation failures that are introduced into hardware/software systems. Cryptographic protocols are open to a further cause of failures: unlike normal hardware/software systems which are likely to interact with a friendly environment (for instance, a user will try carefully to only input valid data to a program in order to avoid a run time crash) cryptographic primitives are assumed to interact with a hostile environment. They must be resilient to all imaginable misuses, not only by an attacker who may interact with the system without being invited, but also by a legitimate user (attacker from inside). Attackers make deliberate abnormal uses of the system, and if necessary, they may collude. That is why often cryptographic protocols, even designed and implemented by security experts, are vulnerable to failures. The long hidden failures in Needham-Schroeder protocols are a well-known lesson on unreliability of security experts [8].

In the area of design of cryptographic protocols there exists well-thought engineering guidelines for the design of cryptographic protocols (e.g., prudent engineering practice of Abadi-Needham [1]and robustness principles of Anderson-Needham [2]). However, these guidelines do not form a computational theory and therefore cannot lead to an automatic or even a computer-aided design method.

Formal analysis, on the other hand, has been shown to be effective in identifying security flaws in many key distribution and authentication protocols. Several different approaches have been developed. Meadows developed a PROLOG based model checker (NRL Protocol Analyzer) [6–7]. The user supplies a description of an insecure state and the PROLOG searches backwards in an attempt to find an initial state. One

serious problem is that the back-tracing algorithm is not guaranteed to terminate. Lowe [5]used the FDR model checker for CSP [4]to find successfully a previously unknown error in the Needham-Schroeder public-key authentication protocol. Schneider [12]used CSP to model protocols in a hostile environment and to express security properties. Verification proceeds by the discovery of a rank function. A prominent approach in the category theorem proving is [10, 3]. Here protocols are specified as traces of events (an event being the sending or storing of a message) and an adversary is included by using operators *parts*, *analz* and *synth* on the messages. By induction over the possible traces it is then proven (under certain assumptions) that a protocol provides certain security properties.

To date, there exists no integrated design and analysis method for the development of secure cryptographic protocols for e-commerce/e-government applications. CASENET aims at closing this gap.

1.1 CASENET objectives

The objective of the CASENET project is to develop and implement a software-based tools framework for the formal and systematic specification, modelling, analysis and implementation of cryptographic protocols for securing electronic commerce and business transactions. This subsumes the following goals:

- Develop specification and modelling methodologies that enable formal specification and modelling of cryptographic protocols:

  A tool under such methodologies can assist a designer of a secure transaction to specify security requirements of the transaction and to model the behaviour of a protocol which is designed to fulfil the secure transaction. The formality of the methodologies allows these to be performed in a routine manner and hence can ease the specification and increase the precision of the modelling.

- Develop analysis methodologies that enable formal analysis of the cryptographic protocols:

  With formal semantics defined under these methodologies, a tool can understand the specification (of the security requirements), the modelled protocol behaviour and specific attack scenarios, and perform qualitative and/or quantitative comparisons between them. In the case of presence of security flaws (i.e., the behaviour does not meet the requirements), the protocol design can be modified with the behaviour re-modelled, and the analysis repeated, until the analysis finds no flaw.

- Develop a set of software-based tools that implements the above methodologies:

  The software-based tools will include specification, modelling and analysis packages. They will also include a package for protocol design, and a package for testing a protocol which has gone through the formal analysis process.

2. CASENET components

Figure 1 depicts the various components of the CASENET approach.

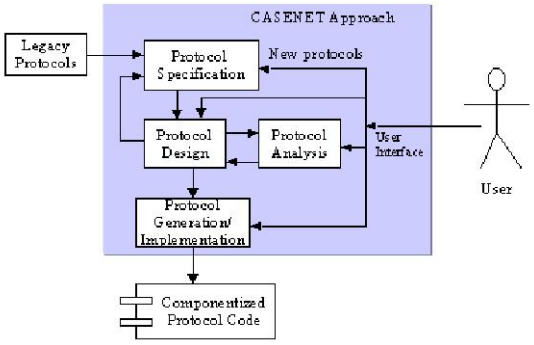

Figure 1. CASENET approach

CASENET aims to achieve these objectives in four main work components. The development of methodologies and tools for the specification, design (WP2) and analysis (WP3) of security protocols constitutes the scientific component of the project.

These methodologies and tools will be integrated into software packages to provide a computer-aided and systemic approach for developing secure e-commerce protocols. Results of this work will be a software application providing specification, design and analysis capabilities (WP4).

Three different user trials will provide real world applications. The trials will cover three different activity domains. Business to Government: *City of Cologne*'s application for tax collection and other financial services via Internet. Business to Business: Continuing work by *NetUnion* on integrating Online Contracting and E-notary applications in various deployment scenarios. Finally, *Sadiel* adds the area Citizen Access to Administrative Services (see below for a description of the Consortium).

The Consortium. The CASENET consortium includes two industrial and one governmental trial partner: *City of Cologne* (governmen-

tal, Germany), *NetUnion* (Switzerland), and *Sadiel S.A.* (Spain). The roles of these partners are to provide scenarios for e-commerce and e-government applications for the validation and evaluation of methodologies and tools. *SOLINET* (Germany) together with *teletel* (Greece, associated to SOLINET), also industrial partners, aim at commercializing CASENET's results. *Hewlett-Packard Laboratories* (HP, Great Britain) is industrial, but is considered a research partner. Further research partners are *Fraunhofer Institute for Secure Telecooperation* (FhG-SIT, Germany), *Norwegian Computing Centre* (NR, Norway), and the *University of Malaga* (UMA, Spain).

3. The CASENET approach

We now give a brief overview of the CASENET approach. For a detailed description we refer the reader to the CASENET web site (http://www.casenet-eu.org).

The protocol generation process of CASENET will be embedded into an integrated development environment including the business process and the system model. UML *Message Sequence Charts* (MSC) will be used to specify the security relevant actions of the application in question. User friendly macros (that can be viewed as a shorthand for logical formulas of the *Security Requirements Language* SRL) will be used to specify the security requirements (the macros will be added to the MSC by use of XMI annotations). This is translated to a system specification which is based on the theory of formal languages (a system and its security properties are given by all possible sequences of actions). The resulting specification consitutes the most abstract level of the protocol specification.

In the following step, the security properties are converted to appropriate protocol modules, mathematical proofs based on language homomorphisms assure that these modules indeed provide the respective properties. Then the modules are combined, and again proofs assure certain security properties. On this abstraction level, cryptographic primitives are modelled by abstract communication channels. In the last step of the design process, these channels are converted to symbolic functions. Again mathematical proofs have to assure security properties.

The resulting specification serves as input for both the analysis tool and SAFIRE, a tool for testing communication processes. Protocol analysis can be performed by using the *Simple Homomorphism Verification Tool* (SHVT). Security properties are translated into properties of global system states, and the tool computes all states that can be reached ac-

cording to the protocol specification and attack scenario used, to search for a state in which a security property is violated.

With SAFIRE, on the other hand, the symbolic functions are implemented by function calls of a cryptographic library, and the protocol is tested, and test cases are generated.

3.1 UML: The user level

A UML-based framework will be developed for representing security semantics in an integrated development environment including the business process and the system model, which is supported by the use of a repository of patterns at several levels of abstraction represented in an XML notation. The focus is on the entire system including the environment or context in which the software system will eventually been inserted.

The contribution of this approach to the whole CASENET project consists mainly in giving a framework for the elicitation, specification, and analysis of the requirements of the user partners whose applications consist largely of replicating electronically current manual systems (e.g. Sadiel's PASEN and Cologne's Stadtkasse Online). Moreover, a unified framework is established for the analysis, design and implementation of similar applications. This part of the work will be jointly performed by HP and UMA.

3.2 SRL and UML macros

In CASENET a language based on first order logic extended with a small set of relations and modal operators is defined for the specification of security requirements.

In order to be user-friendly, the language supports an abstraction mechanism, i.e., the *macro* mechanism, that allows to hide complex formulæ behind terminologies familiar to the application owners. The macros will be used as pre and post conditions to specify which security properties have to hold throughout and after a certain action.

The CASENET design methodology uses a well-defined formalism based on sequences of actions. Hence, a security specification language with well-defined semantics is needed to bridge the gap between UML-like notations and the design formalism. SRL is designed to that end. The SRL language and macros are being developed by NR.

3.3 The design formalism

The specification of the application resulting from MSC and macros/SRL provides specified behaviour of the application in terms of sequences of

actions, and security requirements in terms of macros which are translated into security properties in terms of properties of formal languages (represented by the box "Specification of application and security requirements" in figure 2 below).

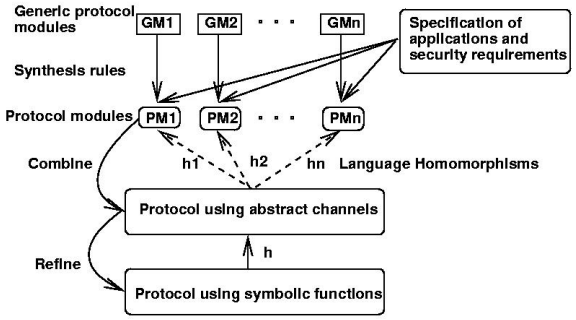

Figure 2. Main parts of design process

As part of the design methodology, so-called generic protocol modules are being developed (boxes "GM1" to "GMn" in figure 2) for providing "atomic" security properties. These modules use the formalism of Asynchronous Product Automata (APA, see [9]for a formal definition) to model the communication of participants in the transactions. In order to provide certain security properties, the use of cryptographic primitives is modelled by abstract communication channels providing the respective properties. Language homomorphisms are being used to mathematically prove that each of the modules provides the respective security property.

These modules can be used in the next step of protocol design: The system specification can now be translated to a less abstract system based on APA. In order to implement the desired security properties, the appropriate generic protocol modules have to be chosen. For each security property that shall hold for a certain action or actions, one or more generic modules will be available to provide this property. This step of protocol generation results in the modules "PM1" to "PMn" of figure 2. These modules will then be combined to form one protocol that implements the transaction of the applications. Again, appropriate language homomorphisms can be used to prove that for the combined protocol still certain security properties hold. This part of protocol design formalism is being developed by FhG-SIT.

The last step of the protocol design will be to translate the abstract channels into appropriate symbolic functions with the respective symbolic cryptographic keys (work to be done jointly by FhG-SIT and HP).

Again, homomorphisms will be used to transfer the security proofs to this lower level of abstraction.

On the high level of abstraction, the respective security properties are the requirements on the system. These requirements correspond to security mechanisms on the lower abstraction level that are realized by using abstract communication channels that provide certain security properties and other security mechanisms. Finally, the abstract secure channels are implemented using a refined model of cryptography, namely symbolic functions.

3.4 Protocol analysis

The protocol specification resulting from the design phase will be transformed into input for the Simple Homomorphism Verification Tool (SHVT) in order to perform protocol analysis. The SHVT supports a verification method for cooperating systems based on formal languages, thus is suitable for the analysis of protocols specified by APA. For CASENET, it has been adopted to provide the functionality specifically necessary for the purpose of security analysis of protocols. Both the tool and the analysis methodology are being developed by FhG-SIT. For further details, we refer the reader to the CASENET web page.

3.5 SAFIRE

The SAFIRE System owned by SOLINET is an advanced environment for the implementation and validation of signalling systems, with special features for executing SDL based systems. The advanced features that SAFIRE provides allow the execution of SDL based systems directly on the target without performing any code generation and modifications to the final code.

The work on SAFIRE, being performed by SOLINET and Teletel, includes the developement of an interface that translates protocol specification in APA into SDL.

Furthermore, the design and implementation of security protocols using SAFIRE requires external interfaces with cryptographic libraries in order to accelerate their development.

In more precise technical terms, the interfacing of crypto libraries is required, which will allow special function calls to external libraries during the execution of an SDL system. Using this interface, special SDL functions can be called directly from an implementation of a security protocol in SDL in order to perform complex tasks and actions including, encryption, decryption, hashing, key generation etc.

References

[1] Martin Abadi and Roger Needham. Prudent Engineering Practice for Cryptographic Protocols. In *Proceedings of the 1994 IEEE Computer Society Symposium on Security and Privacy*, pages 122–136, Los Alamitos, California, 1994. IEEE Computer Society Press.

[2] Ross Anderson and Roger Needham. Robustness principles for public key protocols. In D. Coppersmith, editor, *Advances in Cryptology – CRYPTO '95*, volume 963 of *Lecture Notes in Computer Science*, Berlin, 1995. Springer Verlag.

[3] G. Bella and L.C. Paulson. Kerberos version iv: Inductive analysis of the secrecy goals. In *5th European Symposium on Research in Computer Security*, Lecture Notes in Computer Science, pages 361–375. Springer-Verlag, 1998.

[4] C. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.

[5] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using CSP and FDR. In *Second International Workshop, TACAS '96*, volume 1055 of *LNCS*, pages 147–166. SV, 1996.

[6] C. Meadows. Applying formal methods to the analysis of a key management protocol. *Journal of Computer Security*, 1992.

[7] C. Meadows. The NRL protocol analyzer: An overview. In *Proceedings of the Second International Conference on the practical Applications of PROLOG*, LNCS. SV, 1995.

[8] R. Needham and M. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, pages 993–999, 1978.

[9] P. Ochsenschläger, J. Repp, R. Rieke, and U. Nitsche. The SH-Verification Tool – Abstraction-Based Verification of Co-operating Systems. *Formal Aspects of Computing, The Int. Journal of Formal Methods*, 11:1–24, 1999.

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

[11] L. C. Paulson. Inductive Analysis of the Internet Protocol TLS. *ACM Trans. on Information and System Security*, 2(3):332–351, 1999.

[12] S. Schneider. Verifying authentication protocols with CSP. In *IEEE Computer Security Foundations Workshop*. IEEE, 1997.