Presented at the National Security Agency's third High Confidence Software and Systems Conference, Baltimore MD, April 2003

Introducing Cyberlogic*

Harald Rueß and Natarajan Shankar

Computer Science Laboratory
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025, USA
{ruess, shankar}@csl.sri.com

Abstract. Cyberlogic is an enabling foundation for building and analyzing protocols that involve the exchange of electronic forms of evidence. The key ideas underlying Cyberlogic are extremely simple. First, evidence is encoded by means of numbers using digital certificates and nonces. Second, predicates are signed by private keys so that a decryption of such a certificate with the corresponding public key is a proof or evidence for the assertion contained in the certificate. Third, protocols are distributed logic programs that gather evidence by using both ordinary predicates and digital certificates. These simple building blocks can be used to construct a rich variety of services in a variety of domains ranging from digital government to access control in computer systems.

1 Introduction

With the advent of widespread network connectivity, many transactions that previously required the exchange of physical evidence can now take place electronically. While a great deal of attention has been focused on electronic commerce, the larger class of transactions involving evidence in the form of licenses, registration documents, visas, and certificates have been mostly ignored. The modern public-key infrastructure can be exploited to construct electronic versions of these transactions so that they can employ digital certificates instead of physical evidence. To digitize such transactions, we need an agreed upon framework that can be used to exchange this kind of electronic evidence. This framework should be up to the task of describing evidential protocols so that they can be shown to achieve their intended purpose. Cyberlogic, as described below, is meant to be just such a framework.

Cyberlogic is a semantic foundation for implementing evidential transactions using the public key infrastructure. Evidential transactions form the basis of frameworks for authorization and authentication, electronic commerce, business workflow, and digital government. Such transactions involve the exchange of physical evidence in the form of identity cards, driver's licenses, money, checks, visas, airline tickets, traffic tickets, birth certificates, and stock certificates, as well as electronic evidence including PIN numbers, passwords, keys, certificates, and nonces. The key ideas underlying Cyberlogic are:

* This research was supported by SRI internal funding and NSF under contract CCR-0208779.

1. Evidence is encoded by means of numbers using digital certificates and nonces.
2. Usually, keys are associated with agents, but in Cyberlogic, we use them to identify specific *authorities*.
3. Statements such as $P(s)$ are signed by private keys K to obtain a certificate c, so that c is evidence for the claim that K *attests* that s has property P. This can be verified by decrypting c with the corresponding public key $\bar{K}$ to see if it yields $P(s)$.
4. Protocols are distributed logic programs that gather evidence by using both ordinary predicates and digital certificates.

Cyberlogic builds on the existing public key infrastructure while noting that the utility of such an infrastructure depends on a coherent and rigorous semantic foundation. Without such a foundation, terms like *authentication*, *anonymity*, *trust*, and *certification* are not meaningful, and the resulting protocols cannot by themselves provide the necessary guarantees. The key observation is that transactions involve the exchange of various forms of evidence. Encryption provides the mechanism for transferring evidence. A foundation must build on a logic of evidence that can be implemented by cryptographic protocols and digital certificates. Such a logic provides the semantics relative to which we can judge the protocols for their correctness and security. It also allows these protocols to be executed as a logic programming language [Kow79]. In this way, the simple foundation of Cyberlogic serves as the basis for explanation as well as implementation.

The rigmarole of acquiring a visa to travel to a foreign country serves as an illustrative example. Such visas require the possession of a valid passport of the country of citizenship, airline tickets for the intended travel dates, bank account information, hotel reservations in the destination country, and valid documents for countries visited in the onward part of the journey. In the Cyberlogic framework, the requester R asks the consulate of the country C for a visa meeting certain requirements. The consulate responds with an outline of the required evidence. The requester then gathers this evidence from the relevant bank, airline, or other consulates, and forwards it to the consulate for country C. The consulate upon verifying the evidence by, if necessary, consulting various certification authorities (CAs), delivers a digital certificate v authorizing requester R's travel. The evidence v might itself be needed in other transactions such as setting up appointments or exchanging currency. The requester can also use the visa certificate to electronically check that the certificate has in fact been issued by a valid authority and does admit the traveler. In electronic form, a visa could be a more interesting artifact. For example, one could make the visa a certificate that can be used only a bounded number of times or over a bounded period. The certificate could be renewable with only incremental evidence when this bound has run out.

It is easy to extrapolate from the above scenario to other uses of digital evidence in electronic commerce, business and administrative processes, and digital government. In particular, it is possible to envision uses within an agent-based framework where agents execute Cyberlogic protocols to carry out specific tasks that require the exchange of authorization and authentication information.

While it is easy to imagine specific scenarios where electronic protocols can be used to provide governmental and commercial services, it is less clear how one designs these protocols systematically and correctly. Cyberlogic aims to provide the foundation for

the design and analysis of electronic service protocols so that they can be shown to meet the stated service requirements.

The building blocks of Cyberlogic services are various kinds of encryption, digital certificates, and trusted certification authorities. The logic itself is based on a first-order logic with variables, function symbols, predicate symbols, propositional connectives, and quantifiers. In addition to the usual connectives it also includes a specialized form implication for expressing delegation and a weaker form of negation capturing the idea that the evidence for a proposition is absent up to some given point in time.

The unifying conception in Cyberlogic is that of a logic of evidence where some of the evidence can be in the form of digital certificates. The assertion that c is evidence that K attests that s has property P is equivalent to checking that c when decrypted with the corresponding public key yields $P(s)$. This is similar to the intuitionistic logic and in its realizability interpretation [Dum77]. In particular, it includes a declarative reading as a logic for reasoning about evidence as well as a procedural reading as a logic programming language.

Altogether, Cyberlogic is an attempt to formulate a uniform semantics and logic of electronic evidential transactions including those of electronic commerce. It serves as the foundation for a rigorous understanding of evidential protocols that are employed in electronic services and commerce. Cyberlogic is also used to formalize the building blocks and protocols involved in such transactions in such a way that their security and reliability can be rigorously verified and safely used in conjunction with other protocols. It also leads to techniques for automating the analysis of these protocols [BAN90, Low96, Pau98, Coh00, MR00, CMR01]. The work that is closest in spirit are logic-based frameworks for authorization and trust management such as PolicyMaker [BFL96], the KeyNote framework [BFK98], of Appel and Felten's logic for authorization and delegation [AF99], and Li, Grosof, and Feigenbaum's delegation logic [LGF00].

One point to emphasize with Cyberlogic is that since it relies on a medium-free formulation of protocols based on evidence, the medium for presenting the evidence is not important. In the case of a prepayment protocol, for example, the payment could be made electronically or physically, and the result would still be the same. Also, there might be multiple ways of obtaining the same evidence, and it is possible to design protocols that are flexible with respect to the means by which the evidence is presented.

References

[AF99] Andrew W. Appel and Edward W. Felten. Proof-carrying authentication. In *ACM Conference on Computer and Communications Security*, pages 52–62, 1999.

[BAN90] Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, February 1990.

[BFK98] Matt Blaze, Joan Feigenbaum, and Angelos D. Keromytis. Keynote: Trust management for public-key infrastructures (position paper). In *Security Protocols Workshop*, pages 59–63, 1998.

[BFL96] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *IEEE Conference on Security and Privacy*. IEEE Press, 1996.

[CMR01] V. Cortier, J. Millen, and H. Rueß. Proving secrecy is easy enough. In *14th IEEE Computer Security Foundations Workshop*. IEEE Computer Society, 2001.

[Coh00] Ernie Cohen. TAPS: A first-order verifier for cryptographic protocols. In E. A. Emerson and A. P. Sistla, editors, *Computer-Aided Verification (CAV 2000)*, volume 1855 of *Lecture Notes in Computer Science*, pages 568–571, Chicago, IL, July 2000. Springer-Verlag.

[Dum77] Michael Dummett. *Elements of Intuitionism*. Oxford University Press, 1977.

[Kow79] R. Kowalski. *Logic for Problem Solving*. Elsevier North Holland, Inc., New York, NY, 1979.

[LGF00] Ninghui Li, Benjamin Grosof, and Joan Feigenbaum. A practically implementable and tractable delegation logic. In *IEEE Conference on Security and Privacy*. IEEE Press, 2000.

[Low96] Gavin Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Tools and Algorithms for the Construction and Analysis of Systems TACAS '96*, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166, Passau, Germany, March 1996. Springer-Verlag.

[MR00] J. Millen and H. Rueß. Protocol-independent secrecy. In *2000 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2000.

[Pau98] L. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6(1):85–128, 1998.