What is QCM?

Carl A. Gunter and Trevor Jim

August 26, 1999

QCM stands for 'Query Certificate Manager'; it is a software system that has been developed at the University of Pennsylvania as part of the SwitchWare project on active networks. QCM is a *Public Key Infrastructure (PKI)* intended to support secure maintenance of distributed data sets like *Access Control Lists (ACL's)* or *public key certificate* repositories. An ACL is a list of 'principals', identified by public keys; such lists can be used to describe who is permitted to access resources such as the ability to read and modify a file, or run a program. A public key certificate is an association between a public key and an individual or entity. QCM allows *policies*, such as the ACL of principals allowed to access a resource, to be described in a special-purpose langauge (the language is also called QCM). The system provides two services. First, it verifies whether a policy is satisfied by a request, and, second, it uses the policy verification to assist in retrieving the certificates (digitally signed documents) that are relevant to the verification. This integrated verification and retrieval mechanism is known as *policy directed certificate retrieval* and is the primary novel contribution of the QCM system.

Example. An example drawn from [11] will help to illustrate the idea of QCM. Suppose a research group L is collaborating with a group R at a remote site and wishes to maintain an ACL for the resources at L to be used in the project. L can define a set, named ACL, of the permitted users with a QCM definition:

$$\mathrm{ACL} = \mathrm{LocalUsers} \cup K_R\$\mathrm{ACL}. \qquad (\dagger)$$

Here K_R is the public key for the group R. The notation $K_R\$\mathrm{ACL}$ is pronounced, "K_R's ACL," and it is the name of a set ACL defined by R consisting of the participants from R that should be allowed access to the resources at L. K_R's ACL is distinct from the ACL defined by (†), which is known globally as K_L's ACL. By qualifying names with keys, QCM ensures that the sets defined by different principals will not be confused and establishes a means of *delegating* maintenance of part of the ACL to a remote site. The definition (†) says that K_L's ACL is the union of a set, LocalUsers, and the set $K_R\$\mathrm{ACL}$. LocalUsers is defined separately by L: for example, its members can be listed explicitly as a QCM expression such as $\mathrm{LocalUsers} = \{K_{\mathrm{Alice}},\ K_{\mathrm{Bob}}\}$. Alternatively this set may be linked to an external data source that interoperates with QCM; the

QCM implementation currently supports definitions in an LDAP database [23] or a flat file in a specific format.

After L has made these definitions, a QCM evaluator on the local machine can be queried with set expressions involving ACL and LocalUsers. For example, if QCM were asked whether K is in ACL, it would check whether it is in LocalUsers or K_R's ACL. Exactly how it does this is largely hidden from the user. LocalUsers is easy to obtain, of course, but to obtain K_R\$ACL QCM might use a variety of different strategies. The most straightforward would be to send a message to a QCM evaluator at R's site. An optimization would be to hold the response at L in case the question is asked again in another query. An extension of this optimization is to mirror (that is, make a copy of) the set K_R\$ACL locally at L. This option breaks into two possibilities: one in which the QCM process at R 'pushes' the ACL (or just the part of it that has changed) whenever it changes, the second in which the QCM process at L 'pulls' the ACL (if it has changed) whenever it needs to use it. In each case, it is essential to secure the integrity of the communications, so QCM signs messages and verifies signatures when appropriate. Moreover, if a principal K from R seeks to access a resource at L, then it is convenient to supply a certificate, signed by K_R, asserting that K is an element of R's ACL; this will allow the QCM verifier at L to prove that K is in L's ACL without consulting remote data at R. QCM automatically and seamlessly supports all of these mechanisms, as well as other commonly-used mechanisms. The key objective is to support local autonomy, that is, the ability of each participant to determine its own access control policies *and* certificate retrieval strategies, while providing significant automation and acceptable global behavior.

Applications of QCM. The use of a PKI system requires the existence of *PKI-aware* applications. These are applications that know about QCM, or some other PKI, and can use it to find keys or determine whether a request satisfies a policy. Two QCM-aware applications have been built so far.

The first application [12] provides policies for the evaluation of PLAN programs. PLAN [19] is a programming language for active networks, that is, it is a language for programming the network routing elements which forward packets in an internet. Internets that support such programmability are called *active networks*. PLAN allows QCM to provide access control policies for functions on active routers invoked by PLAN packets. This capability has been applied [13] to the development of an active network *firewall*, that is, an active network router that examines packets to provide security for a portion of a network.

The second application [17] provides ACL maintenance for a test bed of computers for active networks known as the *ABONE* [1]. The ABONE uses a program called *ANET* to allow users to set up active network experiments on a collection of machines located around the world. The testbed allows active networks to be tested in the context of actual Internet traffic, but significant security concerns are raised by availability on the public network, which is notorious for mischief makers. QCM provides support for ANET's ACL's, which determines who is allowed to use the ABONE. This support is scheduled for full deployment in September of 1999.

Related Work. The best-known PKI architecture is derived from the ISO Directory series of standards [15] and is usually referenced by its certificate format standard, X.509 [16]. This approach to PKI has been widely developed and there are a number of vendors of PKI products or PKI-aware applications that rely on X.509-based solutions. In particular, the Internet Engineering Task Force (IETF), the organization that develops standards for the Internet, has a working group called *PKIX* that is developing X.509-based PKI standards [22, 14] for the Internet as well as PKI-aware applications like secure electronic mail [18]. To find out more about X.509, [9] and [8] are good sources.

Although X.509 has been widely developed and implemented, its architecture is not necessarily appropriate for all applications, and, in particular, there is a desire to explore more light-weight, non-hierarchical approaches. An example of an alternative model is the popular PGP system for secure electronic mail [24], which bases its PKI on a simple system of key servers on the World Wide Web and *key rings* maintained by users. Key rings support policy definitions and provide a local database of certificates to be used in verifications. QCM was inspired by a seminal work on an alternative to X.509 called PolicyMaker [4], and has drawn a number of ideas from similar systems such as SDSI [21], SPKI [7], and KeyNote [3]. These systems provide approaches to defining security policies such as access control lists and aim to provide good support with simplicity. QCM is similar to these systems, but bases its policy definition language on a database language called *comprehensions* [5, 6]. While these other systems base certificate retrieval on a separate as-yet-undefined mechanism, QCM uses query decomposition and optimization techniques to provide automated certificate retrieval as part of policy verification.

Finding Out More. The QCM home page [20] provides pointers to papers on QCM, and related web sites. The principle papers on QCM are [10], which introduces policy directed certificate retrieval, and [11], which describes the mathematical semantics of QCM. A discussion of the SwitchWare implementation generally, including QCM, can be found in [2] and software for QCM has been released as part of the distribution of PLAN 3.2 [19].

References

[1] Active network backbone (ABONE). `http://www.csl.sri.com/ancors/abone`.

[2] D. Scott Alexander, Michael W. Hicks, Pankaj Kakkar, Angelos D. Keromytis, Marianne Shaw, Jonathan T. Moore, Carl A. Gunter, Trevor Jim, Scott M. Nettles, and Jonathan M. Smith. The switchware active network implementation. In Greg Morrisett, editor, *ML Workshop*, Baltimore, Maryland, September 1998. `http://www.cis.upenn.edu/~switchware/papers/ml.ps`.

[3] Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos D. Keromytis. The keynote trust-management system version 2. `http://www.cis.upenn.edu/~angelos/Papers/rfcnnnn.txt.gz`, March 1999.

[4] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of the 17th Symposium on Security and Privacy*, pages 164–173. IEEE Computer Society Press, 1996.

[5] Peter Buneman, Leonid Libkin, Dan Suciu, Val Tannen, and Limsoon Wong. Comprehension syntax. *SIGMOD Record*, 23(1):87–96, March 1994.

[6] Peter Buneman, Shamim Naqvi, Val Tannen, and Limsoon Wong. Principles of programming with complex objects and collection types. *Theoretical Computer Science*, 149(1):3–48, September 1995.

[7] Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian M. Thomas, and Tatu Ylonen. SPKI certificate theory. Internet Draft, March 1998.

[8] Entrust white papers. `http://www.entrust.com/downloads/whitepapers.htm`.

[9] Warwick Ford and Michael S. Baum. *Secure Electronic Commerce*. Prentice Hall, 1999.

[10] Carl A. Gunter and Trevor Jim. Policy directed certificate retrieval. `http://www.cis.upenn.edu/~qcm/papers/qcm-abstract.html`, July 1998.

[11] Carl A. Gunter and Trevor Jim. Guneralized certificate revocation. `http://www.cis.upenn.edu/~qcm/papers/crl.ps`, July 1999.

[12] Michael Hicks. PLAN system security. Technical Report MS-CIS-98-25, Department of Computer and Information Science, University of Pennsylvania, April 1998.

[13] Michael Hicks and Angelos D. Keromytis. A secure PLAN. In Stefan Covaci, editor, *Proceedings of the First International Workshop on Active Networks*, volume 1653 of *Lecture Notes in Computer Science*, pages 307–314. Springer-Verlag, June 1999. Extended version at `http://www.cis.upenn.edu/~switchware/papers/secureplan.ps`.

[14] R. Housley, W. Ford, W. Polk, and D. Solo. *Internet X.509 Public Key Infrastructure: Certificate and CRL Profile*. IETF RFC 2459, January 1999.

[15] ISO/IEC 9594–1. *Information technology—Open Systems Interconnection—The Directory: Overview of concepts, models and services*, 1997. Equivalent to ITU-T Rec. X.500, 1997.

[16] ISO/IEC 9794–8. *Information technology—Open Systems Interconnection—The Directory: Authentication framework*, 1997. Equivalent to ITU-T Rec. X.509, 1997.

[17] Pankaj Kakkar, Michael McDougall, Carl A. Gunter, and Trevor Jim. Credential distribution with local autonomy. `http://www.cis.upenn.edu/~qcm/papers/autonomy-abstract.html`, March 1999.

[18] S. Kent. *Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management*. IETF RFC 1422, February 1993.

[19] Plan: A packet language for active networks. `http://www.cis.upenn.edu/~switchware/PLAN`.

[20] Query certificate managers. `http://www.cis.upenn.edu/~qcm`.

[21] Cryptography and information secrurity group research project: A Simple Distributed Security Infrastructure (SDSI).

[22] Sean Turner and Alfred Arsenault. *Internet X.509 Public Key Infrastructure: PKIX Roadmap.* IETF, 1999. `www.ietf.org/internet-drafts/draft-ietf-pkix-roadmap-01.txt`.

[23] W. Yeong, T. Howes, and S. Kille. *Lightweight Directory Access Protocol.* IETF RFC 1777, 1995.

[24] P. R. Zimmerman. *The Official PGP User's Guide.* MIT Press, 1995.