CryptoManager++

An object oriented software library for cryptographic mechanisms

Thilo Baldin,
CNI - Communications Network International GmbH
D-65760 Eschborn, Germany, Thilo.Baldin@cni.net

Gerrit Bleumer
Institut für Informatik, Universität Hildesheim
D-31141 Hildesheim, Germany, bleumer@acm.org

Abstract

An object oriented approach to implementing non-interactive cryptographic mechanisms is presented. The primary design goals are object reuse, minimal code redundancy, easy update and extension by new algorithms and an intuitive application programming interface. The object orientation proved to cause a run time overhead of no more than 4%.

Keywords

Cryptographic library, application programming interface, object oriented programming

1 INTRODUCTION

Several cryptographic application programming interfaces are in use and so are different standards for exchange formats of the involved data, e.g., public cryptographic keys and digital signatures. Moreover, there are hundreds of customized versions of C-code for the underlying algorithms, e.g., Schneier (1996). So what is new with CryptoManager++?

Firstly, it is an object oriented concept by which the vast majority of current crypto graphic mechanisms can be implemented and that achieves code reusability, minimal code redundancy, and makes it easy to integrate new algorithms.

Secondly, the resulting C++ software library can be customized to the security needs of most applications: in particular to those using more than two cryptographic algorithms or those needing quick or periodic update by improved cryptographic algorithms. It is also well suited for training and education in practical cryptography.

Thirdly, an easy to use application programming interface (API) is provided. Intuitive programming is supported by exploiting polymorphism, and distractions by unnecessary details or parameters are avoided. However, any other cryptographic API like those of FIPS and X/Open (1994) can also be supported by our concept.

2 OBJECT ORIENTED DESIGN

At the first stage we characterize cryptographic mechanisms by two orthogonal properties (see Figure 1): *key structure* (symmetric, asymmetric, keyless) and *cryptographic purpose* (encipherment, authentication, hashing, etc.). There are practical examples of almost every

combination of a key structure and a cryptographic purpose. Thus, we capture each property by an abstract class and use multiple inheritance in order to combine properties at the next stage. An example at this stage is the class “Sign”, which provides asymmetric authentication. The corresponding mechanisms, i.e., digital signature mechanisms, are found at the next stage, where we have specific classes for RSA, DSA, ElGamal signatures, etc. All of them are descendents of the class “Sign”. This gives a rather unified API, which provides, e.g., only one method “sign” hiding all the different algorithms of the various signature mechanisms.

In addition to the *elementary mechanisms* considered so far, there are *hybrid mechanisms* that use the services of other (elementary or hybrid) mechanisms. Hybrid mechanisms are of practical relevance because they combine the advantages of asymmetric key management with the performance of symmetric or keyless mechanisms. A concrete example is to sign a message by first applying a fast hash function and then signing only the resulting hash value. Since we have not found hybrid mechanisms with cyclic dependencies, we regard each as a tree of elementary mechanisms. The root mechanism, digital signature in our example, is suggested to be inherited from the corresponding class “asymmetric authentication”. Within the tree, mechanisms can be parametrized by other mechanisms by means of templates. The application programmer neither has to care about managing different keys nor about coordinating several elementary mechanisms. He need not even be aware of how many different keys he works with.

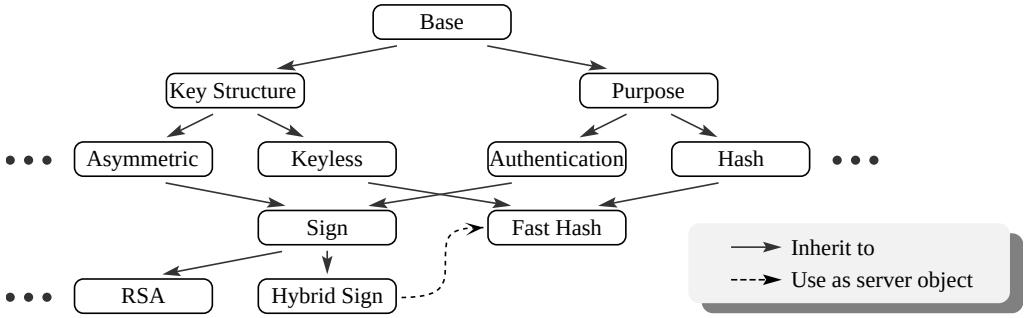

Figure 1. Excerpt from the hierarchy of inheritance of CryptoManager++

3 SECURITY, PORTABILITY, EFFICIENCY

The security parameters, including key length, of each single instance of a cryptographic mechanism can be adapted to the risks faced by an application. The first implementation of the software library is in C++. For faster performance, we use a small (about 700 loc) engine for multiple precision integer calculations written in SPARC assembler.

A Sun SPARC Station 5 achieves a hash performance of about 28 MBit/s for MD5, 25 MBit/s for RIPE-MD and 11 MBit/s for SHS. Encryption performance is about 1,6 MBit/s for IDEA, 1 MBit/s for DES and 25 KBit/s (decryption: 1 KBit/s) for RSA with a key length of 1024 bit (all in CBC mode). The run-time overhead caused by (multiple) inheritance and overloading is less than 4%.

Acknowledgment: We would like to thank Ralf Kanne and Frank Sudholt for their practical support as well as for many stimulating discussions on object oriented design.

4 REFERENCES

Schneier, B. (1996) Applied Cryptography (2nd ed.). John Wiley & Sons, New York.

X/Open Company Ltd. (1994) Generic Security Service API (GSS-API) Base, Preliminary Specifications. X/Open Document Number: P308, ISBN: 1-85912-025-3.

5 BIOGRAPHIES

Thilo Baldin successfully finished his studies in computing science in 1995 at the University of Hildesheim, where he worked as a researcher at the time this paper was written. His main interests are security in open communication networks (especially telecommunications networks) and mathematical cryptography. He currently works at the IT-security department of CNI - Communications Network International GmbH, a commercial German telecommunications network provider.

Gerrit Bleumer received his diploma in computing science from the University of Karlsruhe in 1991 and is now a researcher at the University of Hildesheim. His main interests are security and privacy in open communication networks and distributed systems, mathematical cryptography, software implementation of cryptosystems and security critical applications such as in health care. He has worked for an EU research project and contributed several publications to international conferences on the above topics.