Architecture, Usage and Extension of the .NET-Krypto-Framework

L. Lo Iacono [1], M. Schneider [2]

1 C&C Research Laboratories, NEC Europe Ltd.

2 Department of Information Technology, Districtal Environmental Authority Siegen (Germany)

Abstract: The .NET framework includes classes for many different cryptographic algorithms. Further algorithms can be implemented and integrated into the existing framework. At first, this article describes the structure of the cryptographic framework. After that, it shows the different possibilities of creating and using cryptographic objects.

Due to the architecture of the crypto framework, .NET is flexibly adjustable. The second part of this article focuses on the possibilities of adjusting the crypto framework. Finally, the integration of a new cryptographic hash algorithm into the .NET framework is shown by using the Whirlpool hash function.

1. Introduction

For realizing security mechanisms in .NET software systems, the .NET framework provides cryptographic procedures in a very flexible way.

So the framework enables selecting between several implementations of the same algorithm.

Furthermore the architecture of the crypto framework supports the possibility of extending the regular algorithm standard scope.

Especially with regard to the cryptographic algorithms, these flexibilities are very important, because the discovery of new weak points poses a serious threat and so the algorithms usually have an expiration date.

2. Three Layer Architecture

In .NET all cryptographic classes are realized in a three-level model.

On the first level, there are abstract classes. Each one of them represents a category of cryptographic procedures and/or techniques, for example the *HashAlgorithm* class.

On the second level, there are also abstract classes, which

extend the fist level classes by adding some algorithm - specific methods, fields, properties, etc.

Classes of the third level implement these abstract interface definitions.

Both the version 1.1 and the version 2.0 of .NET framework include base classes for the following crypto systems:

- symmetric crypto systems
- hash functions
- message authentication codes
- random number generators
- X.509 certificates
- XML security

3. Using Crypto-Objects

Independent of the underlying algorithm, the .NET framework offers four different ways to create cryptographic objects:

1. with the *new-operator* of an implementation class, located on the 3rd level
2. with the *Create()-method* of an abstract base class, located on the 1st level
3. with the *Create()-method* of an abstract algorithm class, located on the 2nd level
4. or by calling the static method *CryptoConfig. CreateFromName()*

Based on some examples the article demonstrates the four ways of creating cryptographic objects and their dependencies on the crypto configuration of .NET.

4. Crypto-Configuration

Finally, the extension of .NET crypto framework by porting the whirlpool hash function is subject for discussion.

So all essential steps of the implementation and integration into the three-level model are shown.

Additionally both name-to-class-mapping and class-to-oid-mapping are described.