INTERNATIONAL STANDARD

ISO/IEC 17960

First edition
2015-09-01

Information technology — Programming languages, their environments and system software interfaces — Code signing for source code

Technologies de l'information — Langages de programmation, leur environnement et interfaces des logiciels de systèmes — Signature numérique pour le code source

Reference number
ISO/IEC 17960:2015(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

ISO/IEC 17960, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

Introduction

Source code is written and is used in many critical applications. Knowing that the source code being relied upon is the same as that which was used in testing is vital to ensuring the safety and security of a particular application. Given the ease with which source code can be modified, some method of protecting the integrity and authenticity of the source code is necessary. Sequestration of the source code throughout the supply chain is one possible method, but ensuring protection in that way is impractical and unreliable. Virtual protection through the use of a digital signature offers a practical solution and provides integrity and authentication even though the source code may traverse an insecure supply chain.

Source code may be modified for legitimate reasons as it moves through the supply chain or over time. Modifications to source code may be made to correct the software or to adapt it for other purposes. Modifications may only involve changes to a few lines of code and in most cases is not made by the original author or team of authors. Revision control software facilitates tracking of the software changes, but such tracking can easily be spoofed. The use of a digital signature provides a means to restrict the ability to spoof. Digital code signing assigns a responsible party to each revision of the source code and thus can demonstrate the authenticity of the responsible party, the source code and the software changes that have been made between revisions. By doing this, an electronic pedigree for the source code can be established.

This standard specifies the process for signing source code in order to ensure the integrity and authenticity of the source code and a means for rolling back the source code to signed previous versions. Clause 5 provides an overview of the concepts of code signing. Conformance requirements for this standard are specified in Clause 6. Annex A is informative and provides a step by step description of a typical application for the standard specified in Clause 6 to assist in understanding code signing. The bibliography lists documents that were referenced during preparation of this standard.

Information technology — Programming languages, their environments and system software interfaces — Code signing for source code

1 Scope

This International Standard specifies a language-neutral and environment-neutral description to define the methodology needed to support the signing of software source code, to enable it to be uniquely identified, and to enable roll-back to signed previous versions. It is intended to be used by originators of software source code and the recipients of their signed source code. This International Standard is designed for transfers of source code among disparate entities.

The following areas are outside the scope of this International Standard:

- Determination of the trust level of a certification authority;
- Format used to track revisions of source code files;
- Digital signing of object or binary code;
- System configuration and resource availability;
- Metadata
  - This is partially addressed by ISO/IEC 19770-2;
- Transmission and representation issues
  - Though this could be an issue in implementation, there are techniques such as Portable Document Format (PDF)[1] that can be used to mitigate these issues. This applies in particular to the transmission of digital signatures.

2 Conformance

An implementation of code signing conforms to this International Standard if it meets the requirements specified in Clause 6.

3 Normative references

The following documents, in whole or in part, are normatively referenced in this standard and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 9594-8:2014, *Information technology — Open Systems Interconnection — The Directory — Part 8: Public-key and attribute certificate frameworks*[2]

ISO/IEC 10118-3:2004, *Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash functions*

[1] ISO 32000-1:2008 Document management — Portable document format — Part 1: PDF 1 specifies a digital form for representing electronic documents to enable users to exchange and view electronic documents independent of the environment in which they were created or the environment in which they are viewed or printed.

[2] This is equivalent to ITU-T Recommendation X.509: 2005, "*Information Technology —Open Systems Interconnection — The Directory: Public-Key and attribute certificate frameworks*"

ISO/IEC 13888-1:2009, *Information technology — Security techniques — Non-repudiation — Part 1: General*

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1
certificate
entity's data rendered unforgeable with the private or secret key of a certification authority

[SOURCE: ISO/IEC 13888-1:2009]

4.2
certification authority
authority trusted by one or more users to create and assign certificates

[SOURCE: ISO/IEC 13888-1:2009]

4.3
changeset
set of all changes that are applied to a configuration to derive a new configuration

4.4
digital signature
data appended to, or a cryptographic transformation of, a data unit that allows the recipient of the data unit to prove the source and integrity of the data unit and protect against forgery, e.g. by the recipient

[SOURCE: ISO/IEC 13888-1:2009]

4.5
hash code
string of bits that is the output of a hash-function

[SOURCE: ISO/IEC 13888-1:2009]

4.6
hash-function
function which maps strings of bits to fixed-length strings of bits, satisfying the following two properties: 1) it is computationally infeasible to find for a given output an input which maps to this output; 2) it is computationally infeasible to find for a given input a second input which maps to the same output

[SOURCE: ISO/IEC 13888-1:2009]

4.7
originator
entity that sends a message to the recipient or makes available a message for which non-repudiation services are to be provided

[SOURCE: ISO/IEC 13888-1:2009]

4.8
private key
key of an entity's asymmetric key pair which should only be used by that entity

[SOURCE: ISO/IEC 13888-1:2009]

4.9
public key
key of an entity’s asymmetric key pair which can be made public

[SOURCE: ISO/IEC 13888-1:2009]

4.10
public key certificate
public key information of an entity signed by the certification authority and thereby rendered unforgeable

[SOURCE: ISO/IEC 13888-1:2009]

4.11
recipient
entity that gets (receives or fetches) a message for which non-repudiation services are to be provided

[SOURCE: ISO/IEC 13888-1:2009]

4.12
snapshot
complete copy of a configuration

5 Concepts

This clause provides an overview of the concepts of code signing.

Code signing is a technique for providing a digital signature for source code to support a verification of the originator and a verification that the code has not been altered since it was signed.

Code signing can provide several valuable functions such as:

- knowledge of the history of the source code
- confidence that the source code has not been accidentally or maliciously altered
- verification of the identity of the responsible party for the source code
- accountability for the source code
- non-repudiation of the originator of the source code

Code signing identifies to customers the responsible party for the source code and confirms that it has not been modified since the signature was applied. Verification of the originator of the source code of the software is extremely important since the security and integrity of the receiving systems can be compromised by faulty or malicious code. In addition to protecting the security and integrity of the software, code signing provides authentication of the author, originator or distributor of the source code, and protects the brand and the intellectual property of the developer of the software by making applications uniquely identifiable and more difficult to falsify or alter maliciously.

When source code is associated with an originator’s unique signature, distributing source code on the Internet is no longer an anonymous activity. Digital signatures ensure accountability, just as a manufacturer’s brand name ensures accountability with packaged software. Distributions on the Internet lack this accountability and code signing provides a means to offer the needed accountability. Accountability can be a strong deterrent to the distribution of harmful code. Even though software may be acquired or distributed from an untrusted site or a site that is unfamiliar, the fact that it is signed by a known and trusted entity allows the software to be used with confidence that it has not been changed as compared to the most recently signed version.

In addition to the valuable functions that code signing offers, this International Standard will specifically facilitate the following capabilities:

- a mechanism to show what has been altered in the source code and the responsible party for such changes;
- multiple signatures to allow for an audit trail of the signed source code;
- versioning information;
- storage of other metadata about the source code.

The capability for a tracking mechanism and multiple signatures for one piece of source code is needed in some cases in order to create a digital trail through the history of the source code. Consider a signed piece of source code. Someone should be able to modify a portion of the source code, even if just one line or even one character, without assuming responsibility for the remainder of the source code. A recipient of the source code should be able to identify the responsible party for each portion of the source code. For instance, a very trustworthy company A produces source code for a driver. Company B modifies company A's source code for a particular use. Company B is not as trusted or has an unknown reputation. The recipient should be able to determine exactly what part of the source code originated with company A and what was added or altered by company B so as to be able to concentrate their evaluation on the sections of source code that company B either added or altered. This necessitates a means to keep track of the modifications made from one signed version to the next.

An alternative scenario is source code offered by company B that contains source code from company A. Company B does not alter company A's source code, but incorporates it into a package or suite of software. It would be useful to a customer to be able to identify the originator of each portion of Company B's software package.

6 Requirements

6.1 General

The code signing standard described below is intended to be language and platform independent.

This International Standard is not prescriptive as to the precise syntax of the APIs supporting the code signing activities. However, any API conforming to this International Standard shall provide interfaces that:

- create a hash code for the source code as specified in 6.2;
- generate a signature as specified in 6.2;
- perform initial signing of snapshots as specified in 6.3;
- perform signing of changesets as specified in 6.3;
- provide for the recording of sufficient information in signed versions to allow the recreation of ancestor versions, at a minimum of the immediate signed ancestor version, as specified in 6.4;
- provide for the retrieval of signed ancestor versions, at a minimum of the immediate signed ancestor version, as specified in 6.4.

6.2 Certificates

The originator shall obtain an X.509 (ISO/IEC 9594-8)-compliant certificate. The level of trust in the Certification Authority (CA) that issues the X.509 (ISO/IEC 9594-8)-compliant certificate is an important factor in the amount of trust associated with the signed code. The CA should be a trusted party to both the originator and potential recipients. Though very important to the execution of a trusted transfer of software from an originator to a recipient, the establishment or determination of the trust level associated with a CA is beyond the scope of this International Standard.

Protection of the originator’s private key shall be ensured to prevent impersonation by others. The private key part of the originator’s certificate shall not be compromised from the control of whoever is authorized to sign the code.

6.3 Hash code

A digital signature shall be generated on the source code using the private key of the originator. The signature technique to be used shall be one of those specified in ISO/IEC 9796-3 or ISO/IEC 14888. Generation of a signature using one of the techniques specified involves the use of a hash-function to compute a hash code of the source code. The preferred hash-function to be used shall be the Secure Hash Algorithm-256 (SHA-256), as specified in ISO/IEC 10118-3; alternatively another hash-function specified in ISO/IEC 10118-3 could be used.

The recipient shall then use the originator’s public key to verify that the source code file has not been altered since it was digitally signed so that the history of the source can be traced back to the first signed version.

6.4 Initial code signing

The initial signing of a source code file may be in any format such as a snapshot or a changeset. If a changeset is used, it should be based on an empty file.

6.5 Modifying signed previous versions

Sufficient information shall be recorded in a signed version to allow the source code file and digital signature of the signed previous version to be recovered. This allows the series of modifications from one version to the next, which can be thought of as encapsulations, to be reversed one at a time.

The information contained in each encapsulation shall contain sufficient revision control information in order to recreate the previous version. Once an encapsulation is reversed, the recipient shall be able to use the digital signature of the encapsulated version to verify its integrity.

A mechanism shall allow for the recreation of the most recently signed version of the source code. It is implementation-defined whether intermediate unsigned versions can also be recreated by this mechanism.

6.6 Revision format

This International Standard is not prescriptive as to which format shall be used to create or track revisions. A conforming implementation of this International Standard shall provide specifications so that recipients can reconstitute the signed previous version.