



IEEE Recommended Practice for Encryption and Management of Electronic Design Intellectual Property (IP)

STANDARDS

IEEE Computer Society

Developed by the
Design Automation Standards Committee

IEEE Std 1735™-2023
(Revision of IEEE Std 1735-2014)



IEEE Recommended Practice for Encryption and Management of Electronic Design Intellectual Property (IP)

Developed by the

Design Automation Standards Committee
of the
IEEE Computer Society

Approved 29 June 2023
IEEE SA Standards Board

THIS IS A REDLINE DOCUMENT PRODUCED BY TECHSTREET, AN ASME COMPANY.

This document is intended to provide users with an indication of changes from one edition to the next. It includes a full-text version of the new document, plus an indication of changes from the previous version.

Redlines are designed to save time and improve efficiencies by using the latest software technology to find and highlight document changes. More professionals are using valuable new technologies like redlines, to help improve outcomes in a fast paced global business world.

Because it may not be technically possible to capture all changes accurately, it is recommended that users consult previous editions as appropriate. In all cases, only the current base version of this publication is to be considered the official document.

This Redline is produced solely by Techstreet. IEEE has not prepared this Redline and is not responsible for its accuracy.

Redline Processing Notes:

1. ~~Red Text~~ - Red strikethrough text denotes deletions.
2. Blue Text - Blue underlined text denotes modifications and additions.

Abstract: Guidance on technical protection measures to those who produce, use, process, or standardize the specifications of electronic design intellectual property (IP) is provided in this recommended practice. Distribution of IP creates a risk of unsanctioned use and dilution of the investment in its creation. The measures presented here include protection through encryption, specification, and management of use rights that have been granted by the producers of electronic designs, and methods for integrating license verification for granted rights.

Keywords: digital envelope, encrypted IP, IEEE 1735™, keys, rights management, trust model

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2023 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 6 November 2023. Printed in the United States of America.

IEEE, POSIX, and 802 are registered trademarks in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

Microsoft and Windows are trademarks of the Microsoft group of companies.

UNIX is a registered trademark of The Open Group.

Verilog is a registered trademark of Cadence Design Systems, Inc.

Print: ISBN 978-1-5044-9993-4 STDGT26363
PDF: ISBN 978-1-5044-9994-1 STDPD26363

IEEE Prohibits discrimination, harassment, and bullying.

For more information, visit <https://www.ieee.org/about/corporate/governance/p9-26.html>

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s).

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of Governors are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the [IEEE SA myProject system](#).¹

Comments on standards should be submitted using the [Contact Us](#) form.²

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, neither IEEE nor its licensors waive any rights in copyright to the documents.

¹ Available at: <https://development.standards.ieee.org/myproject-web/public/view.html#landing>.

² Available at: <https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html>.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; <https://www.copyright.com/>. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit [IEEE Xplore](#) or [contact IEEE](#).³ For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#).⁴ Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the [IEEE SA Patent Policy](#).⁵

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at <https://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that

³ Available at: <https://ieeexplore.ieee.org/browse/standards/collection/ieee>.

⁴ Available at: <https://standards.ieee.org/standard/index.html>.

⁵ Available at: <https://standards.ieee.org/about/sasb/patcom/materials.html>.

determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

Participants

At the time this standard was submitted to the IEEE SA Standards Board for approval, the Electronic Design Intellectual Property (IP) Working Group had the following membership:

Abigail Moorhouse, *Chair*
David Rickel, *Secretary*
Rita Tharakan, *Technical Editor*

Luis Humberto Rezende Barbosa
Riju Biswas
Brian Durwood
Domenic Forte
Dave Graubart*

Roger B. Hughes
Nathan P. Jachimiec
Satyam Jani
Tayung Liu

G. Richard Newell
Bin Ochotta
Gael Paul
Vivek Popli
Ruchi Tyagi

*Chair Emeritus

The following members of the individual Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Luis Humberto Rezende Barbosa
Ted Bardusch
Riju Biswas
Dennis Brophy
Peter Capelli
Aditya Chaudhuri
Diego Chiozzi
Michael Dood
Donald Dunn
Brian Durwood
Dave Graubart
Marco Hernandez
Jonathan Herrera

Werner Hoelzl
Piotr Karocki
Quist-Aphetsi Kester
Rameshchandra Ketharaju
Alexis Kokkonos
Krzysztof Kozminski
Wang Lei
Jim Lewis
Daozhuang Lin
Greg Luri
Johnny Marques
Arjuna Marzuki

Abigail Moorhouse
Rajesh Murthy
Bin Ochotta
Dev Paul
David Rickel
Cole Scott
Gerald Stueve
Yatin Trivedi
John Vergis
Lisa Ward
Janusz Zalewski
Bingsheng Zhang
Daidi Zhong

When the IEEE SA Standards Board approved this standard on 29 June 2023, it had the following membership:

David J. Law, *Chair*
Ted Burse, *Vice Chair*
Gary Hoffman, *Past Chair*
Konstantinos Karachalios, *Secretary*

Sara R. Biyabani
Doug Edwards
Ramy Ahmed Fathy
Guido R. Hiertz
Yousef Kimiagar
Joseph L. Koepfinger*
Thomas Koshy
John D. Kulick

Joseph S. Levy
Howard Li
Johnny Daozhuang Lin
Gui Lin
Xiaohui Liu
Kevin W. Lu
Daleep C. Mohla
Andrew Myles

Paul Nikolich
Annette D. Reilly
Robby Robson
Lei Wang
F. Keith Waters
Karl Weber
Philip B. Winston
Don Wright

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 1735-2023, IEEE Recommended Practice for Encryption and Management of Electronic Design Intellectual Property (IP).

The purpose of this recommended practice is to provide guidance on protection of electronic design intellectual property (IP) written in the SystemVerilog and VHDL hardware design and verification languages. The audience for this recommended practice includes IP producers, IP consumers, vendors of tools that process protected IP, and standards development groups for IP specification formats.

This recommended practice provides a way to transform IP written in SystemVerilog or VHDL for the purpose of IP exchange, such that the exchange does not reveal all the original source. The transformation is performed using encryption, and the transform operation limits usage of transformed files to specific tools selected by the IP owner. The transformed IP is as usable by the selected tools as it was prior to the transformation; however the original SystemVerilog or VHDL text is no longer visible to the IP consumer in the transformed regions of the file.

This recommended practice explicitly does not authenticate or provide any traceability related to the authorship and therefore the content of encrypted code. The IP author and IP user stakeholders should assure content authentication and trust in the pre-encryption text separately from the HDL decryption and tool usage described in this recommended practice.

Acknowledgments

Grateful acknowledgment is made to Cadence Design Systems Inc. for permission to use the following source material:

[Clause 7](#) is derived from the document titled “Rights Management Specification in Verilog® Protected Envelopes” © 2007, 2011, Cadence Design Systems Inc. Used, modified, and reprinted by permission of Cadence Design Systems Inc.⁶

⁶ Verilog is a registered trademark of Cadence Design Systems, Inc.

Contents

1.	Overview.....	11
1.1	Scope.....	11
1.2	Purpose.....	11
1.3	Word usage	12
1.4	Version history.....	12
1.5	Value and approach	13
1.6	Key characteristics of the IEEE 1735 recommended practice for encryption.....	13
1.7	Conventions	14
1.8	Use of color.....	14
1.9	Security considerations	15
1.10	Overview of contents	15
2.	Normative references	16
3.	Definitions, acronyms, and abbreviations.....	17
3.1	Definitions	17
3.2	Acronyms and abbreviations	22
4.	Trust model	24
4.1	IP stakeholders	24
4.2	Role of IP protection.....	24
4.3	Protection through encryption	25
4.4	Components of trust.....	26
5.	Interoperability.....	28
5.1	Background.....	28
5.2	Character set, case-sensitivity, and related issues.....	28
5.3	Basic interoperability overview	30
5.4	Elements of interoperability.....	31
5.5	Anatomy of a decryption envelope	36
5.6	Input to encryption tools	39
5.7	Transform tools.....	41
5.8	Signature data marshalling.....	41
6.	Asymmetric key management	43
6.1	Overview.....	43
6.2	Key considerations.....	43
6.3	Standard key exchange	44
6.4	Key scope recommendations	44
7.	Rights Management	46
7.1	Introduction.....	46
7.2	Scope of rights	46

7.3	Types of hTools and associated rights.....	46
7.4	Syntax and markup for rights.....	47
7.5	Complete tool block and rights example	48
8.	Proxy version and license management.....	49
8.1	License system.....	49
8.2	License proxy.....	49
8.3	License specification.....	49
8.4	Licensing use model	50
8.5	License types.....	51
8.6	Return value reuse	51
8.7	License server connections	51
8.8	Message formats	52
8.9	License irregularities.....	54
9.	Visibility management.....	56
9.1	Introduction to visibility management.....	56
9.2	Background.....	56
9.3	Visibility in tool phases	58
9.4	Visibility and protected text.....	61
9.5	viewport pragmas.....	63
9.6	General rules and definitions for viewport pragmas.....	67
9.7	Programming language interfaces	70
9.8	Controlling visibility with rights.....	72
9.9	Visibility of dynamic objects.....	72
9.10	Unresolved visibility issues	73
	Annex A (normative) Syntax specification for IEEE 1735	74
	Annex B (normative) Test Vectors	78
	Annex C (informative) Best practices and outstanding issues	81
	Annex D (informative) Protection pragmas.....	85
	Annex E (informative) Bibliography	88

IEEE Recommended Practice for Encryption and Management of Electronic Design Intellectual Property (IP)

~~IMPORTANT NOTICE: IEEE Standards documents are not intended to ensure safety, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.~~

~~This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.~~

1. Overview

This document describes the IEEE 1735 recommended practice for encryption of intellectual property. This is the second publication of this recommended practice. This revision resolves ambiguities, improves the security of encryption and licensing protocols, and provides more detailed descriptions of various concepts.

1.1 Scope

This [recommended practice standard](#) specifies embeddable and encapsulating markup syntaxes [to manage rights for the encryption and decryption of](#) ~~for~~ design intellectual property ([IP](#)) ~~encryption and rights management~~, together with recommendations for integration with design specification formats described in [IEEE Std 1800™ \(SystemVerilog\) and IEEE Std 1076™ \(VHDL\)](#).⁷ ~~other standards~~. It also recommends use models for interoperable tool and hardware flows, which will include selecting encryption and encoding algorithms, [as well as](#) ~~and~~ encryption key management. The recommendation includes a description of the [assumed](#) trust model [on which](#) ~~assumed in~~ the recommended use models [are based](#). This [recommended practice standard](#) does not specifically include any consideration of digitally encoded entertainment media.

In the context of this document, the term *IP* will be used to mean electronic design intellectual property. Electronic design intellectual property is a term used in the electronic design community. It refers to a reusable collection of design specifications that represent the behavior, properties, and/or representation of the design in various media. Examples of these collections include, but are not limited to, the following: a unit of electronic system design; a design verification and analysis scheme (e.g., test bench); a netlist [that indicates](#) ~~indicating~~ elements and [their interconnections](#) ~~the interconnection thereof~~ to implement a function; a set of fabrication instructions; a physical layout design or chip layout; a design intent specification. The term [IP](#) is partially derived from the common practice for the collection to be considered the intellectual property of one party. [The term encompasses both](#) hardware and software descriptions ~~are encompassed by this term~~.

^{7,9}Information on references can be found in Clause 2.

1.2 Purpose

The purpose of this document is to provide interoperability between IP authors, tool providers, integrators and users of encrypted SystemVerilog and VHDL IP. The interoperability supports licensing and management schemes to help IP authors control tool behavior. This document also makes the overall interoperability flow transparent. It addresses security concerns and enhancement requests to IEEE Std 1735-2014. This document provides guidelines and recommended practices for the use of IP protection markup syntax and key management to enable interoperable tool flows with IP and tools from a wide variety of suppliers.

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall equals is required to*)^{8,9}

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should equals is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may equals is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can equals is able to*).

⁸The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

⁹The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

1.4 Version history

Hardware description language (HDL) encryption markup has been described in multiple IEEE publications, specifically versions of IEEE Std 1800 (SystemVerilog), IEEE Std 1076 (VHDL), and this document, IEEE Std 1735-2023. The first published version, IEEE Std 1735-2014, assigned standardized HDL encryption with specific envelope versions for pragma markup. Note that envelope version is distinct from the IEEE 1735 publication version. Legacy markup (retroactively assigned envelope Version 0) did not require an explicit version pragma. See Table 1 for an explanation of envelope versions.

Table 1—Envelope versions

Envelope version	Description	IEEE 1735 publication year
Version 0	The earliest HDL encryption implementations, defined to be Version 0. The implementations were not interoperable. This version did not use explicit version pragmas.	
Version 1	An aggregate of the first interoperable encryption methodologies broadly shared by the SystemVerilog and VHDL standards was defined as Version 1.	2014

(New)

Table 1—Envelope versions (continued)

Envelope version	Description	IEEE 1735 publication year
Version 2	IEEE Std 1735-2014 advanced interoperability with a new definition, Version 2. Version 2 was a single standard, designed to apply to both SystemVerilog and VHDL. Version pragmas are required in Version 2.	2014
Version 3	The current recommended practice. Continues the development of the single encryption recommended practice and defines Version 3.	2023

(New)

1.5 ~~Purpose, Value, and approach~~

The intent of this document is to enable design flows that provide interoperability among IP authors, tool providers, integrators, and users of the IP. The resulting flow ~~identified~~ aids IP authors in providing IP that can be processed by tools without sharing protected information with IP users. Furthermore, this flow can support an integrated licensing scheme, enabling the IP authors to specify runtime ~~compile-time~~ licenses to further restrict access or allow behavior to be customized. An integrated rights management scheme is also an element of this recommended practice; it ~~the flow, which~~ helps IP authors to control tool behavior including, but not limited to, IP visibility, permitted ~~allowed~~ tool versions, and output file encryption.

~~There is currently no defined, independent standard for describing IP encryption markup for design information formats. Each design format that incorporates IP encryption describes their markup differently, leading to confusing interpretation. Users of those standards also lack a recommended practice for interoperable use of IP encryption.~~

For the IP author, the digital envelope containing the rights, key, and data blocks is the main focus. This envelope contains the source, encrypted with a symmetric cipher using a one-time session key. The session key is encrypted with a tool vendor public key for each supported tool or vendor, arranged in a series of tool ~~key blocks. The license constraints and rights constraints are similarly encrypted with a tool vendor public key for each supported vendor arranged in a series of rights~~ blocks.

The tool vendor supporting the recommended practice ~~standard~~ can then parse and process the envelope and its contents. When the tool vendor finds the blocks with their public keys, each block is authenticated and decrypted with their private key, thus extracting the data key, licensing requirements, and rights constraints. If required, the tool then checks for ~~the presence of~~ a valid license and; if successful, decrypts the source using the data key and obeying the rights constraints. Output files from the tool may be encrypted for use in downstream tools, provided such action is granted in the IP rights.

~~A standard defined with all these IP author and tool vendor features would make the overall flow transparent for the IP user. Therefore,~~ This document provides guidelines and recommended ~~practices~~ practice for ~~use of IP protection~~ markup syntax and key management to enable interoperable tool flows with protected IP and tools from a wide array of suppliers. It includes information about selecting algorithms ~~algorithm selection~~ for encryption and encoding.

This document also specifies a subset of markup syntax for ~~hardware description language (HDL)~~ formats that could be adapted to other file formats. These files represent potential inputs and outputs of tools that would otherwise expose IP. The generic syntax of these directives may be suitably modified for a particular file format if there are syntactic conflicts ~~and variations that may be described in recommended practices.~~

1.6 Key characteristics of the IEEE 1735 recommended practice for encryption ~~this standard~~

The marketplace for electronic design data is built ~~depends~~ on trust between the parties to the transactions. Exchanges require mutual ~~a~~ confidence that ~~between~~ the parties ~~that they~~ represent legitimate interests, that the product of exchange is accurately described, and that the terms of the transaction will be fairly observed.

The recommended practice utilizes the concept of trusted partners, who are required to maintain the security of confidential information provided to them and who are expected to comply with end-user license agreements (EULAs) as well as all applicable laws. Implementers of this recommended practice shall take into consideration required mitigation strategies against activities that would conflict with the expectations of trusted partners. In addition, implementers should advise their customers of the need to consider similar precautions, such as not sharing confidential information with untrusted partners.

As a form of *intellectual property*, electronic design data is a virtual object and is susceptible to the evolving social conventions of respect for ownership rights. Despite uneven acceptance of their rights to control the use of IP, those who produce such property ~~frequently~~ want to monetize their investment and receive compensation for their investment in its creation. Risk premiums to compensate for loss are not commonly itemized, but can be assumed to be as a factor in compensation demands. The inverse of a risk premium is the trust discount, which is most commonly reflected in a greater willingness to market the IP as the strength of the trust model improves. This concept of the trust model is explained in more detail in Clause 4.

Trade secret protection is usually impractical as the security and reliability of the protection decreases with each new transaction. It also requires a degree of confidence in the integrity of the receiving party, which could be ~~that is an~~ unwise ~~assumption~~ if the objective is to preserve the financial value of the IP.

Since party integrity is a concern, ~~the other forms of~~ legal protection may need to accompany the be ~~accompanied by~~ technical security described in this document to help enforce the terms of the transaction. The recommended practices described by this document ~~standard~~ assist in the construction and maintenance of technical security that is consistent with the intent of the contracted terms of the transaction.

1.7 Conventions ~~used in this standard~~

Note that this document might show examples on multiple lines because of page width restrictions, but real-world syntax would be on a single line. Each command description (syntax) consists of a command keyword, followed by one or more parameters. The meta-syntax that describes ~~for the description of~~ the syntax rules uses the conventions shown in Table 2. For a detailed discussion of the syntax, see Annex A 1.

Table 2—Document conventions [used in syntax examples](#)

Text	Represents
<code>courier</code>	The Courier font indicates source code from one of the relevant HDL languages, e.g., <code>module test(a,b,c,d);</code>
bold	The bold font indicates key terms, or text that shall be typed exactly as it appears. For example, in the following markup directive, the keyword pragma shall be typed as shown: <code>pragma pragma_keyword</code>
<i>italic</i>	The italic font represents variables. For example, an encryption algorithm identifier needs to be specified in the following line (after the <code>key_method</code> term): <code>key_method algorithm_identifier</code>

(New)

Visual cue	Represents
<code>courier</code>	The courier font indicates source code from one of the relevant HDL languages, e.g., <code>module test(a,b,c,d);</code>
bold	The bold font is used to indicate key terms, text that shall be typed exactly as it appears. For example, in the following markup directive, the keywords “pragma” shall be typed as it appears: <code>pragma pragma_keyword</code>
<i>italic</i>	The italic font represents variables. For example, an encryption algorithm identifier needs to be specified in the following line (after the “key_method” key term): <code>key_method algorithm_identifier</code>
<code>::=</code>	A production consists of a left-hand side, the symbol “::=” (which is read as “can be replaced by”), and a right-hand side. The left-hand side is always a syntactic category; the right-hand side is a replacement rule. The meaning of a production is a textual-replacement rule: any occurrence of the left-hand side may be replaced by an instance of the right-hand side.
[] square brackets	Square brackets indicate optional items on the right-hand side of a production. For example, the following productions are equivalent: <code>author_spec ::= author_directive [author_info_directive]</code> <code>author_spec ::= author_directive author_directive author_info_directive</code>
separator bar	The separator bar () character separates alternative choices unless it occurs immediately after an opening brace, in which case it stands for itself. For example, the following line shows the “delegated” or “true” or “false” key terms are possible values for the “Boolean_rights_value” production: <code>boolean_rights_value ::= delegated true false</code>

(Deleted)

1.8 Use of color ~~in this standard~~

This [recommended practice](#) ~~standard~~ uses a minimal amount of color to enhance readability. The coloring is not essential and does not affect the accuracy of this [recommended practice](#) ~~standard~~ when viewed in pure black and white. The places where color is used are the following:

- [Links are shown in blue text. The links work when this recommended practice is viewed interactively as a PDF file.](#)
- [The use of color and descriptions of syntactic keywords and tokens in the formal language definitions are described in Annex A.](#)
- ~~— Cross references that are hyperlinked to other portions of this standard are shown in underlined blue text (hyperlinking works when this standard is viewed interactively as a PDF file).~~
- ~~— Syntactic keywords and tokens in the formal language definitions are shown in **boldface**.~~

[1.9 Security considerations](#)

[Security considerations are described in context throughout this recommended practice. Special formatting \(an enclosing box\) indicates a security consideration:](#)

[Throughout this recommended practice, text within a box is used to highlight implementation concerns that significantly impact overall security without necessarily having functional visibility. Please review all these notes when creating any IEEE-173 5-compliant tool.](#)

1.10 Overview of contents

~~1.6 Contents of this standard~~

The organization of the remainder of this document for the recommended practice is shown
~~standard is~~ as follows. The later clauses define the technical subparts needed to help ensure the trust
model and share IP. ÷

- Clause 2 provides references to other applicable standards that are presumed or required for this recommended practice ~~standard~~.
- Clause 3 defines terms, acronyms, and abbreviations used throughout the different specifications contained in this recommended practice ~~standard~~.
- Clause 4 describes a trust model for protecting and sharing IP.
- Clause 5 through Clause 9 describe ~~The next several clauses provide~~ the technical background, and make general recommendationss for encrypting (and sharing) IP, covering
topics like ~~in specific areas~~: interoperability, key management, rights management (and rights control structures), license management, visibility management, and common rights.
- Annexes ~~—~~ Following Clause 9 ~~10~~ are a series of informative and normative annexes.

~~NOTE—Clause 4 through Clause 10 define the technical subparts needed to ensure the trust model and share IP.¹~~

2. Normative references

The following referenced documents are indispensable for the application of this document (~~i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained~~). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

[FIPS 186-4, Digital Signature Standard \(DSS\).](#)¹⁰

IEC 61691-1-1, ~~IEEE Std 1076TM~~, Behavioural languages—Part 1-1: VHDL Language Reference Manual (Adoption of IEEE Std 1076TM-2002).^{11 2,3}

[IEEE 1735 Test Vectors \[Source code\], Rickel, David, 2022.](#)¹²

IEEE Std 1800TM, IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language.^{13,14}

IETF RFC 3447, Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1.^{15, 16 5}

[IETF RFC 4648, The Base6, Base32, and Base64 Data Encodings, Proposed Standard, October 2006.](#)

[ISO/IEC 8825-1:2021, Information technology—ASN. 1 encoding rules—Part 1: Specification of Basic Encoding Rules \(BER\), Canonical Encoding Rules \(CER\) and Distinguished Encoding Rules \(DER\) \(Recommendation ITU-T X.690\).](#)¹⁷

¹⁰FIPS publications are available from the NIST Computer Security Resource Center (<https://csrc.nist.gov/>)

¹¹IEC 61691-1 is the published adoption of IEEE Std 1076.² IEC publications are available from the International Electrotechnical Commission (<http://www.iec.ch/>) ~~and - IEC publications are also available in the United States from~~ the American National Standards Institute (<http://www.ansi.org/>).

¹²Available at <https://opensource.ieee.org/1735/test-vectors/-/archive/main/test-vectors-main.zip> (supplement to <https://opensource.ieee.org/1735/test-vectors>).

¹³The IEEE standards or products referred to in Clause 2 are trademarks owned by The Institute of Electrical and Electronics Engineers, Incorporated.

¹⁴IEEE publications are available from The Institute of Electrical and Electronics Engineers (<http://standards.ieee.org/>).

¹⁵IETF documents (i.e., RFCs) are available for download at <http://www.rfc-archive.org/>.

¹⁶Note that RFC 8017, PKS #1: RSA Cryptography Specifications, Version 2.2, November 2016, obsoletes RFC 3447.

¹⁷ISO/IEC publications are available from the International Organization for Standardization (<http://www.iso.org/>) and the American National Standards Institute (<http://www.ansi.org/>).

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.^{18 8}

AD, AAD: Associated data, authenticated associated data. *See:* **associated data**.

AEAD (authenticated encryption with associated data): Encryption with authentication of the data to be encrypted and associated data to be retained *in plaintext*. Authentication verifies that neither the encryption nor the *associated data* have been altered.

AES (Advanced Encryption Standard): A specification for a number of symmetric encryption ciphers established by the U.S. National Institute of Science and Technology.

AES256-CBC-HMAC-SHA512: An authenticated encryption with associated data (AEAD) scheme consisting of a 256-bit key Advanced Encryption Standard (AES) cipher operating in cipher block chaining (CBC) mode using an //M4C-based SHA512 for authentication.

AES key: Key for Advanced Encryption Standard (AES) encryption. *See:* **AES**.

associated data (AD): Unencrypted data associated with protected data, which should be authenticated along with the protected data.

NOTE—For this recommended practice, this is the *decryption header*.¹⁹

^{18 8}IEEE Standards Dictionary Online ~~subscription~~ is available at: <http://dictionary.ieee.org>. An IEEE Account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page. <http://www.ieee.org/portal/innovate/products/standard/standards-dictionary.html>.

^{19 4}Notes in text, tables, and figures [of a standard](#) are given for information only and do not contain requirements needed to implement [this the](#) standard.

asymmetric encryption: Also known as *public key encryption*. An encryption/decryption scheme where the key to ~~encrypt~~ ~~decrypt~~ the data (e.g., private key) is different from the key to ~~decrypt~~ ~~encrypt~~ the data (e.g., public key) and where the encryption key is not readily derived from the decryption key. See also: **RSA**; **symmetric encryption**.

authentication: Verification that the message (both *ciphertext* and *associated data*) has not been tampered with since it was generated. Note that authentication does not verify that the IP originated with the claimed source.

Base64 encoding: A method of representing a string of 8-bit text as a sequence of simple printable characters. Valid base64 data shall contain only characters from the valid base64 alphabet (65 characters including the padding '=' sign). (Adapted from RFC 4648, October 2006, Section 4)

canonicalization: The decryption envelope text is canonicalized prior to cryptographic signature calculation in encryption and decryption. The canonicalization algorithm adjusts for file format differences that do not carry relevant semantic meaning; for example line-end characters on different platforms. See: **semantically equivalent**.

certificate: A standardized wrapper for the public key portion of an *asymmetric encryption* cipher. The wrapper includes the public key and the issuer, and may contain additional data such as the country of origin or the expiration date.

certificate authority (CA): A trusted entity that manages and issues security *certificates* used for secure communications.

cipher block chaining (CBC): A technique for block cipher chaining where each block of *plaintext* is XORed with the previous *ciphertext* block before being encrypted.

ciphertext: Encrypted data. This may refer to The data ~~may be~~ encrypted by either a symmetric cipher or an asymmetric cipher.

common block: The portion of the *decryption header* or *encryption recipe* that describes rights common to all *hTools* allowed to process the *decryption envelope*.

NOTE—For this recommended practice, the only *pragmas* allowed in the common block are *license pragmas*.

cryptographic hash function: A cryptographically strong function that maps a string of arbitrary size to a string of fixed size (the hash value). One property of a cryptographic hash function is that it be computationally infeasible to find two strings that hash to the same value.

~~**common rights:** An optional section describing rights common to all tools.~~

data block: The encrypted and encoded electronic design intellectual property (*IP*).

NOTE—For this recommended practice, this consists of the *IV* used to generate the *ciphertext* of the *IP*, concatenated with the *ciphertext*.

decryption envelope: A region of electronic design intellectual property (*IP*) with encrypted text and markup instructions to direct the process of decryption.

NOTE—For this recommended practice, this consists of the *decryption header* and the *data block*.

decryption header: The portion of the *decryption envelope* that contains the data needed to decrypt and annotate the *ciphertext*.

DER format: Distinguished Encoding Rules (~~X.690, ASN.1~~). A method of canonically representing certain data structures, ~~here~~ used in this recommended practice to encode *RSA* public keys and certificates. It is also used to encode information transmitted between the tool and the *license proxy*. (Adapted from X.690, ASN.1) :

digest: A scheme where a validation string for a message is generated. This is used to check the integrity of the message and can be used to authenticate it and —to verify that ensure the message has not been tampered with since it was generated. See: cryptographic hash function ~~came from the claimed source.~~

digital envelope: A bounded data collection that uses encryption technology for protection and/or identification. See: [decryption](#) ~~Digital envelope is the name for a primary IEEE 1735 use case, in which there is just such a digital envelope constructed for each tool. That envelope is the key block, and it "contains" a session key encrypted with a public key.~~

~~NOTE—See 5.4.1.~~

disclosed name: A protected name which has been made accessible to the end user through some mechanism.

encryption markup: Pragmas describing the text to encrypt and the method for encryption. Encryption pragmas may be grouped together, apart from hardware description language (HDL) statements, or may be intermixed with the HDL.

encryption recipe: A collection of encryption markup pragmas that together describe how to encrypt and annotate the IP. The encryption recipe only consists of the encryption markup that directs the encryption of the IP, not the encryption markup delimiting the IP to be encrypted.

envelope version: The version of IEEE 1735 being used, which is indicated by the version pragma.

NOTE—The current document describes Version 3. See Table 1.

eTool (encryption tool): A tool used to transform unprotected hardware description language (HDL) into HDL containing decryption envelopes.

FQDN (fully qualified domain name): A dot-separated (.) list of domain fields specifying an exact path in the DNS (domain naming system hierarchy), e.g., *userName.companyName.com*

~~**encryption envelope:** A region of electronic design intellectual property (IP) with markup instructions to direct the process of encryption.~~

HMAC (hash-based message authentication code (~~HMAC~~)): A scheme that uses a message *digest* algorithm and a secret key to generate a string for a message that can be used to authenticate the message. (Adapted from [IETF](#) RFC 2104)

hTool [hardware description language (HDL) tool]: A tool whose input is HDL, possibly containing *decryption envelopes*. The hTool is capable of decrypting and using the contents of the *decryption envelopes*.

initialization vector (IV): An initial value provided to the symmetric encryption algorithm. Typically, this is generated by the encryption engine and must be made available to the decryption engine.

NOTE—For this recommended practice, the first bytes of the data block are the IV.

IP (intellectual property): A work or invention that is the result of creativity, which may be covered by legal property rights such as patents, trademarks, copyrights, etc.

IP author: The provider of an *IP*.

IP user: The user of an *IP*; the company or the person running the *hTool* that utilizes the *IP*.

license pragmas: *Pragmas* within a common block, used to describe the interface to the license proxy.

~~**key block**: The *session key* encrypted with a particular tool vendor's public key and encoded with Base64 Encoding.~~

~~**key owner/key name**: This pair uniquely describes the public/private key pair used to encrypt the *session key*.~~

license proxy: A program provided by the ~~electronic design intellectual property (IP)~~ *author* to control access by the tools to the *IP*. A license proxy may be an actual license server or an ~~a~~ interface to a third-party license server.

licensing call: An invocation from within the *tool block* that interfaces with the *license proxy* to gather information about what actions the *hTool* is allowed to perform on the *IP* contained in the *decryption envelope*.

line-end sequence: Text files from different operating systems follow different line-end conventions.

NOTE—The Unix® convention uses a single line feed character, a byte of hex value 0A, to indicate the end of the line. The Microsoft® Windows® convention uses two characters, the carriage return byte of hex value 0D, followed by the line feed character, hex value 0A. ²⁰ ²¹

MAC (message authentication code): A cryptographic algorithm for verifying that a message has not been tampered with. *See also: HMAC; MAC key.*

MAC key: The secret key used to initialize the *HMAC* function.

model fidelity: The concept that, with the exception of certain operations which may grant insight into the design of the IP, the encrypted IP shall behave to the *hTool* identically to how it would if the IP were not encrypted.

NOTE—For details on *model fidelity*, refer to the discussion in 9.2.2.

NIST: National Institute of Standards and Technology, a government standards organization of the United States of America.

octet: 8 bits of data (one byte).

oracle: A tool that holds cryptographic secrets, such as keys, that it must not directly reveal. An oracle may be queried, and its responses might indirectly reveal information about these secrets that would be otherwise unavailable. Specifically, an *hTool* is an oracle because it can decrypt the hardware description language (HDL) and present the results of applying the tool to the HDL. The *hTool* should present the results without actually revealing the keys it uses.

padding: Additional data that is inserted into text to be encrypted to make the message a multiple of the encryption block size.

plaintext: Unencrypted data, as opposed to *ciphertext*.

²⁰UNIX is a registered trademark of The Open Group.

²¹ Microsoft and Windows are trademarks of the Microsoft group of companies.

pragma: Directives in the hardware description language (HDL) that affect the processing of the HDL.

printable characters: The set of ASCII characters from space (hex 20) to ~ or tilde (hex 7e), inclusive.

protected name: A name defined within a protected (encrypted) region. This name by default is not visible to or accessible by the end user.

protected text: Protected IP. This is either *plaintext* surrounded by text selection *pragmas* in hardware description language (HDL) that is to be encrypted, or the *ciphertext* portion of the *data block*.

protection envelope: Another name for *decryption envelope*. See: **decryption envelope**.

NOTE—For this recommended practice, the preferred term of *decryption envelope* will be used.

protect pragma: A *pragma* that is part of the *encryption markup* or *decryption envelope*, which affects the processing of *protected text*.

proxy version: The license proxy protocol being used.

NOTE—This recommended practice describes Version 3.

public key cryptography: See: **asymmetric encryption**.

random number: A "high quality" (high entropy, cryptographically secure) random number. Most computers do not have access to truly random sources, so these random numbers are generated by deterministic algorithms from presumably unpredictable sources, such as hardware driver timing. Each bit of the random number should be unpredictable and not correlated to any other bit.

re-encryption: Encrypting a (possibly) modified version of input intellectual property (IP) using the same decryption header as the original. By necessity, re-encryption is done using the same *root key* as the original IP; the initialization vector (IV) is required to differ. See: **tTool**.

recipe pragma: An *encryption markup* pragma which is part of an *encryption recipe*.

rights: Directives to the *hTool* describing the operations that may be performed on the *protected text*.

rights identifiers: Rights identifier names are specific to a tool.

NOTE—The rules governing the allowable character set and recognition rules are defined in 5.2.

root key: A base key used to derive the Advanced Encryption Standard (AES) *key* and the message authentication code (MAC) *key*.

RSA: Rivest-Shamir-Adleman Public Key cryptosystem, named after its authors; one of the earliest and most widely used public key systems. The security of the system is based on the difficulty of factoring large numbers. Therefore the keys are large.

NOTE—This recommended practice advocates a minimum of 3072 bits (384 bytes).

rsa no padding: A variant of RSA encryption where the data to be encrypted is padded with random data. The length of the original text cannot be derived from the unencrypted text; the decrypting tool must determine the expected length through another channel.

semantically equivalent: The pragmas that are described in IEEE Std 1735-2023 carry semantic information in a lexical (text-based) form. It is possible for the text representations to differ but the semantic meaning of the pragmas to be identical; if so, they are considered semantically equivalent. Encryption tools may introduce lexical differences in their output pragmas that are semantically equivalent to the input pragmas. *See also:* **canonical**.

SHA512: One of the SHA-2 family of cryptographic hash functions. It is designed to operate with 64-bit words and produces a hash of 512 bits (64 bytes).

signed text block: A consecutive sequence of bytes for encryption and decryption, beginning with the first character of the line with the **begin protected** pragma, and ending with the final character of the text word **endprotected**.

NOTE—See 5.8.1 and 5.8.2.

string: IEEE 1735 compliant strings are a sequence of characters from the allowed alphabet, enclosed within double-quote characters. An example of a standardized string is "rsa no padding". The recommended practice also uses non-standardized strings such as the key keyname value, which is provided by the tool vendor.

NOTE—String character sets and recognition rules are defined in 5.2.1.

~~**proxy session key:** A random key used to encrypt data for communication between the license proxy and the tool.~~

~~**pseudo-random number:** A number whose bits may appear random but that may be correlated with other bits or past values.~~

~~**random number:** A completely random number, meaning it is unpredictable with no bit sharing any relationship with any other bit, each bit equally likely to be a 1 or a 0, and the values of any bit is completely unrelated to any previous random numbers.~~

~~**rights digest:** A Hash based Message Authentication Code (HMAC) generated from the session key, key owner, key name, (optional) common rights, and (optional) tool-specific rights.~~

~~**session key:** A random key used to encrypt data in an encryption block.~~

symmetric encryption: An encryption/decryption scheme where the key to encrypt the data is the same as the key to decrypt the data. See also: [asymmetric encryption](#).

text selection pragma: Encryption markup pragma that delimits the text to be encrypted according to the existing recipe pragmas.

tool block: A section within the decryption envelope that contains information to identify vendor tools that are allowed to access the block, information the vendor tool needs to decrypt the data block, and information about the operations the vendor tool is allowed to perform on the IP in the data block.

transport layer security (TLS): A set of protocols for securely transmitting data over a computer network.

~~**tool block:** The section containing the **key block** and optional **tool-specific rights**.~~

~~**tool-specific rights:** An optional section describing rights specific to one tool.~~

trust model: The defined trust relationships and orientation of the communities participating with the organization in the continued operation of the business. This term also describes the common elements that need to be in place for a clear understanding of trust in systems related to information sharing, authentication, reliability, and many other endeavors. *Trust models* are often considered in information security contexts when there is a need to decide whether a specific security methodology (e.g., digital *certificates*) is appropriate for achieving a specific security objective (e.g., user authentication).

tTool (transform tool): An *hTool* that is capable of outputting re-encrypted *decryption envelopes*. Note that tTools should preserve the *decryption header* of transformed *decryption envelopes*, and also generate a new *IV* for the transformed *data block*.

visible name: A name that has complete visibility, both within and outside the protected region.

VPI: Verification Procedural Interface, as defined in IEEE Std 1800.

3.2 Acronyms and abbreviations

AAD authenticated associated data

AD associated data

AEAD authenticated encryption with associated data

API Application Programming Interface

CLI Command Line Interface

CBC cipher block chaining

~~**API** application programmers interface~~

CA certificate authority

~~CLI command line interface~~

~~DER Distinguished Encoding Rules~~

DPI Direct Programming Interface

EDA electronic design automation

FIPS Federal Information Processing Standards

FQDN fully qualified domain name

HDL hardware description language

HMAC hash-based message authentication code

hTool hardware description language (HDL) tool

IP ~~electronic design~~ intellectual property

IV initialization vector

LRM language reference manual

MAC message authentication code

~~NIST National Institute of Standards and Technology~~

PLI programming language interface

SDO standards development organization

SDF Standard Delay Format

TLS transport layer security

UPF Unified Power Format

VHDL VHSIC Hardware Description Language

VHPI VHDL Programing Interface

VHSIC very high speed integrated circuit

VPI [Verification Procedural](#) ~~System~~Verilog programming Interface

4. Trust model

[While encryption provides a level of IP protection](#), it is [essential](#) ~~necessary~~ for the [IP](#) stakeholders ~~in IP protection~~ to trust each other in order to do business. ~~If one is~~ To understand [the](#) ~~how encryption plays a~~ role [of encryption](#) and ~~therefore to~~ trust it, [it is necessary to first understand](#) ~~sufficiently, one needs to have a concept of~~ the trust model it supports. ~~In general,~~ Understanding the trust model helps [in deciding the appropriateness of](#) ~~one decide that~~ a given security mechanism ~~is appropriate~~ to achieve the security objective.

This clause identifies the stakeholders and their interests. It explains the role of IP protection, how ~~it can be achieved via~~ encryption [can help achieve protection](#), and why [encryption](#) ~~it~~ is valuable. Finally, [this clause examines](#) the components of the trust model ~~are examined~~ to understand the foundation they provide for trusting this form of IP protection.

4.1 [IP](#) stakeholders

In any transaction there are [interested](#) parties ~~of interest, those~~ who hold a stake in the outcome of the transaction. [In](#) ~~Within~~ this trust model, there are three primary stakeholders who represent the spectrum of concerns: the *IP author*, the *IP user*, and the *tool vendor*.

- The *IP author* stakeholder ~~is~~ The legal owner of the IP and ~~is~~ the grantor of any rights that are conveyed in the course of transactions involving that IP. The nature of the rights offered and the compensation demanded for those rights are specified by the IP [authors](#) ~~author~~. Their [interests include](#) ~~interest includes~~ providing a valuable product and [reducing](#) ~~preventing~~ loss through [unauthorized](#) disclosure of their IP to either [the](#) competition or IP users. Their ultimate interest is maximizing their return on [the](#) investment in the IP.

- The *IP user* stakeholder ~~is~~

The receiving party who seeks possession and rights to the IP to further their own interests. ~~In exchange for~~ After providing satisfactory compensation to the IP author, ~~the~~ IP ~~users~~ ~~receive~~ ~~user receives~~ the transacted rights to use the IP. Their ~~interests include~~ ~~interest~~ ~~includes~~ choosing the right IP ~~that is~~ of the appropriate usefulness and quality, and ~~to~~ ~~reduce~~ ~~minimizing~~ the risk and cost of ~~mistaken choices~~ ~~mistakes in doing so~~. Their ultimate interest is ~~to reduce~~ ~~in minimizing~~ the total cost ~~of using~~ ~~to use~~ the right IP ~~for~~ ~~in~~ their designs.

- The *tool vendor* stakeholder

~~The party that~~ provides a tool to the IP user ~~or the IP author~~. ~~The tool~~ ~~that~~ is enabled to work with the IP, ~~and~~. ~~Indeed, they provide a tool to the IP author as well~~. Here, ~~the tool~~ provides ~~a~~ ~~an IP~~ protection mechanism based on encryption that can be used by the IP author to deliver protected IP to the IP user. ~~The same~~ ~~This~~ mechanism allows the IP user to ~~use the~~ ~~work with the tool for its intended purpose with that~~ protected IP ~~within the tool~~, while providing a high barrier to ~~unauthorized~~ disclosure of the IP. The tool vendor ~~has~~ interests ~~that~~ are similar to those ~~of~~ ~~described for~~ the IP author. ; They include providing a valuable product, their tool, and ~~reducing~~ ~~preventing~~ loss through disclosure of tool features and internal behavior to their competition. Their ultimate interest is ~~to improve~~ ~~maximizing~~ their return on investment in the tool, which includes the investment in the IP protection mechanism.

4.2 Role of IP protection

The ~~price~~ ~~pricing~~ of an IP is based on a complex set of factors, ~~including~~ : accounting for the risk of loss due to disclosure ~~is one of them~~. Though not commonly itemized, it is fair to assume this adds a premium to the compensation demands ~~by~~ ~~of~~ an IP author. ~~This document refers~~ ~~It is referred~~ to ~~it~~ ~~here~~ as the *risk premium*.

The goal of an IP protection mechanism is to mitigate the risk of loss due to disclosure. One such mechanism is IP protection through encryption ~~is one mechanism~~, but there are many others, including altering the representation of the IP through obfuscation or abstraction; ~~;~~ limiting access to secure, fixed, ~~"brick-and-mortar"~~ locations; ~~;~~ and a variety of legal mechanisms. None of these are perfect. An IP protection mechanism provides a degree of trust that a loss will ~~does~~ not happen. The ~~An~~ IP protection mechanism itself may require ~~have~~ considerable cost to employ, which is a part of the risk premium.

When IP authors are uncertain about how to price the risk premium, they are usually more comfortable discounting ~~"the loss aspect"~~ when they trust that the IP protection mechanisms bring the risk down to an acceptable level. This may be reflected in lower prices or a greater willingness to engage in commerce with a wider audience of potential IP users. It is referred to here as the *trust discount*.

There is usually nothing specific an IP user can do to substantially improve the trust discount for a right to access the IP. Establishing a long-term trusted relationship with an IP author will ~~should~~ have a positive effect, ~~of course~~. ~~While~~ Loss due to disclosure is of prime ~~an IP author~~ interest to the IP author, but this ~~;~~ ~~it~~ is not as important to ~~a shared interest of~~ the IP user. It is fair to say this is instead a conflict of interest.

The tool vendor can provide an IP protection mechanism that, if done well, provides ~~should provide~~ a trust discount for its use. IP protection via encryption is just such a mechanism, which is valuable to IP authors and IP users, both of whom seek a trust discount for their own self-interest. The tool vendor is an objective party that does not have an interest in the IP itself. Acting as a proxy to protect the IP and make it useful to the IP user presents no conflict of interest for the tool vendor.

Subclause 4.3 provides ~~4.3 gives~~ an overview of IP protection via encryption, described at a level that supports ~~to support~~ further discussion of the encryption ~~its~~ trust model.

4.3 Protection ~~through~~ ~~via~~ encryption

Encryption is a lossless transformation; ~~it is~~ ⁷ not obfuscation or abstraction. The protected IP ~~is identical~~ ~~has 100% fidelity~~ to ~~the~~ ~~its~~ original representation. This IP can ~~also~~ be delivered to the IP user premises. In its encrypted form⁷ and ~~without in the absence of~~ the decryption key, the data is secure both in transmission and at rest in a file. This encrypted data is based on cryptographically strong encryption algorithms. There are no independent means to decrypt and access it at the IP user premises.

The IP protection mechanism is designed to ~~only~~ allow ~~the IP to be used~~ ~~use only~~ through vendor tools ~~that are~~ designated by the IP author. A tool conforms to the expectations of the applicable standards with respect to rights granted by the IP author and disclosure of information during its execution. There is no intentional disclosure of protected information in an IP, except as permitted. The original description of the IP is processed to produce its encrypted form. This single encrypted form can be delivered and used by all authorized tools, i.e., the IP protection mechanism is interoperable.

This ~~recommended practice standard~~ builds on previous work to define a common ~~markup~~ scheme ~~of markup~~ for encryption and decryption of IP. Following ~~is a~~ ~~are the~~ high-level ~~explanation of the process: ideas for how it works.~~

- The IP author marks up the IP lightly to describe the encryption scheme, ~~define the~~ ~~what~~ aspects ~~are~~ to be protected ~~or remain~~ ~~what remains~~ visible, ~~set out the~~ ~~what~~ rights ~~of~~ the IP user ~~has~~, and so on.
- ~~For the chosen~~ ~~In terms of that~~ encryption scheme, the primary use case supported by this ~~recommended practice standard~~ is the *digital envelope* model.
 - In this model, the IP author chooses a set of vendor tools ~~that can~~ ~~to enable~~ access ~~the~~ ~~to their~~ IP.
 - Each tool has its own public-/private key pair ~~where the public key~~ ~~that~~ is used to encrypt the common session key, ~~which is~~ used to protect the IP.

- In the resulting protected IP, there is a [tool](#) ~~key~~ block for each vendor tool ~~and it that~~ contains the encrypted⁷ common session key [for that vendor tool](#).
- [Within the tool](#) ~~That key~~ block, [the key](#) can only be decrypted by the specific tool vendor whose key was used to protect it.
- With that session key, the tool can decrypt the protected IP.

See Clause 5 for a more [detailed description](#) ~~complete understanding~~ of interoperability and use cases supported by this IP protection mechanism.

4.4 Components of trust

[The following subclauses provide a high-level view of the essential components of the trust model for IP encryption: strong encryption, identity verification, vendor integrity, security of the design flow, and tool quality.](#)

~~It is good to know what needs to be trusted. For that, it is important to take a systematic look at the components of the trust model for this IP protection mechanism. This subclause provides a high level identification of those components.~~

4.4.1 Encryption strength

Guarding the IP against unauthorized disclosure requires ~~use of~~ a high-quality encryption algorithm that uses a sufficiently long key. Secure encryption relies on the computational difficulty of reconstructing the key, even when the details of how the algorithm applies the key are known. Longer keys make that reconstruction difficult and eventually impractical. There comes a point where the weakness in the overall cryptosystem moves away from the choice of encryption algorithm and key, towards other aspects.

This applies to the protected IP when it is "at rest" in a file and when it is "in transit" over a network as part of delivery to IP users. See [5.3](#) ~~53~~ for details on the required algorithms.

4.4.2 Digital ~~identity~~ verification of identity

In the primary digital envelope use model ~~case~~, the IP author designates a particular tool as authorized to decrypt their IP by using the public key for that tool. In general, trust relies on the ability to establish that such keys, and any other digital identity tokens⁷ that are offered₄, correspond to the implied stakeholder. There are many strategies for introducing fraudulent identities, with ~~and~~ ~~there are~~ corresponding defenses against these risks ~~this risk~~. This recommended practice ~~standard~~ accounts for the digital verification component of the ~~this~~ trust model component in its key management recommendations in ~~(see~~ Clause 6).

4.4.3 Tool vendor integrity

The structure of this IP protection mechanism requires the IP author to delegate access to their IP to the IP user by proxy. The tool is enabled to decrypt the IP and entrusted with enforcing the rights to use it during execution. The IP author is presumed to trust the integrity of the tool vendor, ~~i.e.~~⁷ to trust that the tool vendor will not intentionally disclose the protected ~~author's~~ IP.

Unauthorized disclosure by the tool vendor is more difficult to guard against with technical measures. ~~The principal tool for managing this risk is the integrity and reputation of the tool vendor.~~ Functional restrictions on disclosure are expected before IP authors entrust their IP to a particular tool, but ~~There is also an opportunity for~~ the primary risk management mechanism continues to be the integrity and reputation of the tool vendor. The IP author might also want to consider⁸ an agreement to share ~~with the tool vendor for~~ restricted access to the decrypted source code with the tool vendor, if it is mutually beneficial to ~~can aid~~ both parties for ~~with~~ IP user support issues.

4.4.4 Security of the design flow

~~The weakest link in a chain is where it is most vulnerable.~~ It is important ~~that to consider~~ all ~~the~~ tools in the design flow ~~enabled by that~~ the IP author ~~be examined for vulnerabilities, because the weak link could be anywhere in the chain enables to use their IP.~~ It is expected that each tool, ~~taken~~ individually, implements the IP protection mechanism correctly with best practices for security and that tool vendor integrity is not in question. Care ~~should shall~~ also be taken with how a particular tool fits together with others in the flow. If the interface between tools ~~not only~~ uses the protected IP representation ~~along with another ,but also depends on other~~ data flow, no protected information should be exposed in ~~the that~~ other data flow ~~either~~.

~~There are Other~~ well-known data formats that are unprotected, ~~such as for example,~~ Standard Delay Format (SDF) and Unified Power Format (UPF), ~~and they~~ present a challenge in this respect. It does not matter whether it is a ~~recommended practice standard~~ or proprietary data representation, allowing interoperability with exposed data is a gap that risks disclosure ~~of proprietary IP.~~ Tools that are designed to have a common architecture for data protection are stronger ~~in this respect~~ than those that have no framework for collaborative protection over the whole flow.

~~The selection The choices of what~~ tools to use in a design flow ~~is are~~ commonly based on factors like functionality, performance, cost, and quality. IP users want the best tools they can afford for their purposes. The IP author has additional considerations to apply ~~when they decide which to the choices of what~~ tools to enable to work with their protected IP.

4.4.5 Tool quality

~~In this context,~~ The quality of the IP protection mechanism is both with respect to correctness and secure coding practices. If a tool discloses protected information during its execution because of a tool defect or incorrect interpretation of standards, that is a correctness issue. If a disclosure happens because a tool is hacked during execution by an untrusted IP user, using any combination of binary tampering and debugging techniques, that is an implementation issue. Each tool vendor needs to decide what, if any, anti-debug, anti-tamper, and anti-key-discovery technologies they will ~~incorporate use here.~~

Tool vendors instill confidence in their tools through [the](#) quality of [their](#) implementation. The effort invested in robust and secure programming practices results in fewer IP user disclosure events. There is no absolute protection, and constant reinvestment is required to stay ahead of evolving public expertise in hacking methodologies.

Understanding that there is no absolute protection, the following are some relevant considerations for agreements between stakeholders.

- Between tool vendors and their users ([who](#) ~~which~~ are IP users and IP authors), An agreement for use of the tool should forbid tampering and reverse engineering without being granted explicit permission. With such permission, the results of any reverse engineering should also be disclosed to the tool vendor.
- Between an IP author and IP user,

An agreement for use of protected IP should forbid [attempts](#) ~~an attempt~~ to tamper or reverse engineer any protection mechanisms in any tool in order to obtain access to IP source code or encryption keys.

4.4.6 Summary

The components of the trust model [outline for](#) ~~give~~ stakeholders [the considerations](#) ~~an ability~~ to [take into account when implementing](#) ~~consider the risks and rewards of~~ this IP protection mechanism. All ~~the~~ stakeholders, acting in their own self-[interest](#) ~~interests~~, should find considerable value in it and recognize [the strength of](#) ~~its strengths in~~ safeguarding IP and [the mutually beneficial](#) ~~its~~ potential for reducing the costs of doing business.

5. Interoperability

~~This recommended practice uses the terms *required* and *requirement* to indicate content that is necessary to be compliant with IEEE Std 1735, including material that is an extension of current usage in IEEE Std 1800 (SystemVerilog) or IEC 61691-1-1, IEEE Std 1076 (VHDL).⁹ This standard also uses the terms *recommended* and *recommendation* for best practices and use cases.~~

~~The recommendations in this clause are intended to be incorporated by reference in other standards, e.g., VHDL, where they shall be read as mandatory requirements on tools conforming to those standards. The intended audience for these recommendations are any working groups considering incorporating this protection framework into other standards.~~

5.1 Background

[IEEE Std 1735-2014 describes envelope Versions 1 and 2, and this IEEE 1735 recommended practice only describes envelope version 3. See 1.4 for details about the history.](#)

~~The historical promise of IEEE IP protection for HDL representations is the ability to deliver a faithful representation of the IP and to be able to protect it economically for delivery to a wide audience of IP users. Support of a markup language, a set of pragmas that could be used to augment the original source code, should be all the IP author needs. A model of using a markup that is minimally intrusive to the code is the expectation. The ideal model, in fact, would require no edits to the source code and might be best described as encapsulation of HDL code by these pragmas. If this was truly standardized, it would be possible for an IP author to protect their IP in an interoperable way that could be delivered in a single form that is usable across a set of tools from different suppliers.~~

~~However, this did not occur. The state of SystemVerilog, VHDL, and IP protection prior to IEEE Std 1735 was IP protection pragmas could have some ambiguities in definition in the first two standards. It was also not clear how to use these pragmas effectively to achieve the basic goal. Furthermore, some of the pragmas had to change and new ones were needed. Plus, the definition ambiguities needed to be resolved. Therefore, a key first step is to define a mechanism for versioning the set of pragmas to manage change and backward compatibility. IEEE Std 1735 defines a **version** pragma and three distinct versions for the set of protection pragmas covered by this standard.~~

5.1.1 Basic terminology

This recommended practice uses the term *decryption envelope*, which is inherited from the VHDL standard. The term is equivalent to the SystemVerilog term *protection envelope*. The recommended practice also uses the term *decryption header* to refer to the text between the `begin protected` pragma and the `data block` pragma.

These and other terms used in this clause, like *disclosed name*, *recipe pragma*, and *text selection pragma*, are also defined in Clause 3.

5.1.2 Changes in Version 3 syntax for encryption/decryption envelopes

This subclause lists the main syntax differences between Version 2 and 3 for encryption and decryption envelopes. For the full syntax for Version 3, see Annex A.

- The end toolblock SHA-HMAC component has been removed, together with its '=' character.
- The encoding and rights digest method pragmas are no longer valid.
- The end protected pragma has been extended with the AEAD signature. For example:
protect
end protected=^MpA8Y9p918cDCS5adyLP15eN+qcZVAdtI3HY//zFg0CVALqXh0NAR-
mucH07srvO6pODKDrmvBwjeEa927CsXmPw==^M

- RSA encryption now requires no padding mode. The recommended key size is 3072 bits.
- The change to an AEAD scheme is detailed as follows.
- Version 3 explicitly disallows nested decryption envelopes (i.e., a re-encryption of a decryption envelope).

5.2 Character set, case-sensitivity, and related issues

This recommended practice defines usage for interoperability across languages (VHDL and System Verilog) and OS platforms and file systems (Unix and Microsoft Windows). In order to support unambiguous interoperability without significantly sacrificing expressibility, there are some restrictions to the character set and case-sensitivity rules, which are described in this subclause. Tools cannot always detect violations of these rules directly; tool developers are responsible for compliance.

The base64 encoding character set is defined independently from the rules in this subclause.

5.2.1 Basic character set

Four classes of usage are identified for the IEEE 1735 recommended practice: standardized keywords, rights identifiers, standardized string values, and non-standardized string values.

String values shall always be enclosed in double-quote characters.

The IEEE 1735 recommended practice does not support any escape mechanism to add otherwise-excluded characters into strings.

For all string values, the recommended practice allows the use of the printable character set of ASCII characters from the inclusive range of byte values 0x20 (the space character) through 0x7e (the tilde character) with the following exceptions:

- 0x22—the double quote character. This is excluded from string content as no escape mechanism is defined and strings are delimited by double quotes.

- 0x5c—the backslash. This is excluded from string content as it is often treated as an escape character in string processing and this usage may be confusing.
- Note that the tab (0x09) and delete (0x7F) characters are already both excluded from the original specified range.

5.2.2 Case sensitivity

Case sensitivity is a concern for the IEEE 1735 recommended practice, because VHDL is a case-insensitive language while SystemVerilog is case-sensitive. This means that VHDL treats objects named "ABC" and "abc" as the same object, but System Verilog distinguishes between them and treats them as different objects.

The IEEE 1735 principle is that the pragmas that introduce IEEE 1735 behavior belong to the language itself and follow language rules. Syntax *after* the introductory pragma is governed by the IEEE 1735 rules. Hence, the System Verilog pragma (case-sensitive) shall always be in lower case:

'pragma protect

As the VHDL pragma is case insensitive, it may be written in any case form. For example:

'PROTECT

'protect

The user should follow the IEEE-1735-specific case rules for the remainder of the syntactical protection pragma text. A tool should make no differentiation of behavior based solely on alphabetic case. This prohibition is most relevant within user-defined rights.

- **Standardized keywords** shall be lower-case in all languages, even if the introductory pragma is in upper or mixed case. The content of strings is limited to the basic character set described in 5.2.1.
- **Standardized strings** shall be in lower-case only and matched in a case-sensitive way. For example, "rsa no padding" is correct, and "RSA NO PADDING" is incorrect.

- Non-standardized strings shall be matched in a **case-sensitive** way and can be defined in any mix of case characters by the string owner. The tool is strictly required to avoid case-only differentiation and may, but is not required to, restrict everything to lower-case to make this easier. For example, two keys owned by the same key owner may be called "key 1" and "KEY 2". Adding a third key to this set called "KEY 1" violates these principles as it is impossible to know if "key 1" and "KEY 1" are the same or different keys.
- Identifiers for user rights shall also be matched in a **case-sensitive** way. Identifiers can have any mix of case characters, subject to the rule to avoid case-only differentiation. The format for identifiers is described in Annex A. To make it easier, the recommendation is to use lower case only.

5.2.3 File splitting and end-of-file characters in decryption envelopes

The System Verilog standard allows for multi-file compilation input. The purpose of this is to allow very large files to be split into multiple smaller files that can be compiled equivalently to the original large file. The VHDL language does not have this feature.

The IEEE 1735 recommended practice shall restrict the System Verilog file splitting feature, and disallow a file from being split anywhere within a IEEE 1735 decryption envelope. If the end-of-file (EOF) character (0x05) is encountered in a decryption envelope, the hTool or tTool shall produce an error.

The first decryption operation is AEAD signature verification. A post-encryption file split shall cause this validation check to fail.

5.3 Basic interoperability overview

5.3.1 Reference interfaces

This recommended practice defines two interoperable interfaces as follows:

- The *plaintext input* files to encryption tools, containing encryption pragma markup for SystemVerilog and VHDL HDL. Encryption tools are referred to in this recommended practice as *eTools*.
- The *encrypted input* to tools that consume SystemVerilog and VHDL HDL. HDL tools capable of IEEE 1735 decryption are referred to in this recommended practice as *hTools*.

The hTool input files contain SystemVerilog or VHDL HDL, with zero or more embedded decryption envelopes. If there are no decryption envelopes, the hTool should process it as regular HDL. The content of the hTool input files is the primary interoperability definition. These files are considered genesis-neutral because the exact mechanism by which the decryption envelope was constructed cannot be known from the decryption envelope itself. A IEEE-1735-compliant tool that can process these unencrypted HDL files and that has the appropriate key, shall also be able to decrypt and process compliant decryption envelopes inserted into the HDL files.

Compliant encryption may also be performed by a transform tool, a *tTool*. In this case all the encryption choices, with the exception of the initialization vector (IV), are inherited from the original encryption.

A tTool can be classified as both an eTool and an hTool, because it decrypts original tTool envelopes just as an hTool does, and then re-encrypts them to output envelopes that are semantically equivalent to eTool output. That is, an hTool cannot distinguish whether its decryption envelopes are from an eTool or a tTool, and this is the principle of genesis neutrality.

A tTool shall retain the original decryption header with the following optional exceptions:

- Changes to language-specific pragma prefixes.

— Platform-specific newline changes.

Specifically, the tTool shall reuse the root key for encryption and signature.

Historically, the eTools have accepted partial or variant encryption control that is understood only by specific tools. The embedded encryption pragmas defined in this recommended practice are only one possible way to provide this guidance. This recommended practice should not restrict tool development for the purpose of partial or variant encryption; however the following are minimum requirements for IEEE-1735-compliant encryption tools:

— eTools must be able to parse fully specified embedded protection pragmas in pre-encryption plaintext.

— eTools must generate compliant output decryption envelopes as input for hTools. The eTools may use other methods besides the encryption pragmas defined in this recommended practice to generate the output.

Further, all encryption tools should respect cryptographic best practices in their encryption implementation, especially with the provision of random data that is required for key and IV selection. It is critical to encryption security that these elements be unguessable.

The requirements at the hTool input interface are stricter. This interface defines the expectations of the output productions of the eTools and tTools, and the input acceptance of the hTools. hTool inputs are in the form of SystemVerilog or VHDL text files with embedded pragma directives to mark up the encrypted portions known as decryption envelopes.

Version 3 compliant decryption envelopes shall explicitly set the version pragma. The following syntax is for SystemVerilog and VHDL decryption envelopes. The whitespace around the '=' sign is optional.

— SystemVerilog: "pragma protect version = 3

— VHDL: " protect version = 3