

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

IEEE Computer Society

Sponsored by the
Design Automation Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 1735™-2014
(Incorporates
IEEE Std 1735-2014/Cor 1-2015)

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

Sponsor

**Design Automation Standards Committee
of the
IEEE Computer Society**

Approved 10 December 2014

IEEE SA-Standards Board

Acknowledgments

Grateful acknowledgment is made to the following for the permission to use the following source material:

Accellera/VSIA—*IP_Encrypt_VSIAtoIEEE.doc*: VSIA contribution from the IP-Encrypt Working Group.

Cadence—Material in clause titled “Rights Management” is derived from the document titled “Rights Management Specification for Verilog® Protected Envelopes” © 2007, 2011, Cadence Design Systems Inc. Used, modified, and reprinted by permission of Cadence Design Systems Inc.

Synopsys—*IP_Licensing_Recommendations_for_P1735.pptx*: Synopsys IP licensing overview.

Abstract: Guidance on technical protection measures to those who produce, use, process, or standardize the specifications of electronic design intellectual property (IP) are 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 © 2015 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 23 September 2015. Printed in the United States of America.

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

POSIS is a trademark of The Institute of Electrical and Electronics Engineers, Incorporated.

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

Print: ISBN 978-0-7381-9492-9 STD20084
PDF: ISBN 978-0-7381-9493-6 STDPD20084

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/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 documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading “Important Notice” or “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

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

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (“IEEE-SA”) Standards Board. IEEE (“the Institute”) develops its standards through a consensus development process, approved by the American National Standards Institute (“ANSI”), which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute 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 does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. 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: PROCUREMENT OF 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 should be considered 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, or 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 his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide 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 Standards Coordinating Committees 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 revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

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 imply 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.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. 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, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, 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. 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 ten years. When a document is more than ten 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 the IEEE-SA Website at <http://ieeexplore.ieee.org/expel/standards.jsp> or contact IEEE at the address listed previously. For more information about the IEEE-SA or IOWA's standards development process, visit the IEEE-SA Website at <http://standards.ieee.org>.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: <http://standards.ieee.org/findstds/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Patents

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 <http://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 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.

Participants

IEEE Std 1735-2014

The Electronic Design Intellectual Property (IP) Working Group is entity based. At the time this standard was completed, the Electronic Design Intellectual Property (IP) Working Group had the following membership:

Dave Graubart, *Chair*
Joe Daniels, *Technical Editor*

Luis Humberto
Rezende Barbosa
Dave Clemans
Steven Dovich
Jeff Fox
Parminder Gill

Satyam Jani
Jarek Kaczynski
Ray Martin
Gael Paul

Rod Price
Adam Sherer
John Shields
Michael Smith
Sourabh Tandon
Ruchi Tyagi

The following members of the entity balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Accellera Organization, Inc.
ALDEC, Inc.
Altera Corporation

Atrenta Inc.
Cadence Design Systems, Inc.

Mentor Graphics
Synopsys, Inc.
Xilinx Inc.

When the IEEE-SA Standards Board approved this standard on 10 December 2014, it had the following membership:

John Kulick, *Chair*
Jon Walter Rosdahl, *Vice Chair*
Richard H. Hulett, *Past Chair*
Konstantinos Karachalios, *Secretary*

Peter Balma
Farooq Bari
Ted Burse
Clint Chaplain
Stephen Dukes
Jean-Phillippe Faure
Gary Hoffman

Michael Janezic
Jeffrey Katz
Joseph L. Koepfinger*
David J. Law
Hung Ling
Oleg Logvinov
T. W. Olsen
Glenn Parsons

Ron Peterson
Adrian Stephens
Peter Sutherland
Yatin Trivedi
Phil Winston
Don Wright
Yu Yuan

*Member Emeritus

IEEE Std 1735-2014/Cor 1-2015

At the time this standard was completed, the Electronic Design Intellectual Property (IP) Working Group had the following membership:

Dave Graubart, *Chair*
Joe Daniels, *Technical Editor*

Luis Humberto
Rezende Barbosa
Dave Clemans
Steven Dovich
Jeff Fox
Parminder Gill

Satyam Jani
Jarek Kaczynski
Ray Martin
Gael Paul

Rod Price
Adam Sherer
John Shields
Michael Smith
Sourabh Tandon
Ruchi Tyagi

The following members of the entity balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Accellera Organization, Inc.
ALDEC, Inc.
Altera Corporation

Atrenta Inc.
Cadence Design Systems, Inc.
Marvell Semiconductor, Inc.
Mentor Graphics

Micron Technology, Inc.
Synopsys, Inc.
Xilinx Inc.

When the IEEE-SA Standards Board approved this standard on 3 September 2015, it had the following membership:

John D. Kulick, *Chair*
Jon Walter Rosdahl, *Vice Chair*
Richard H. Hulett, *Past Chair*
Konstantinos Karachalios, *Secretary*

Masayuki Ariyoshi
Ted Burse
Stephen Dukes
Jean-Philippe Faure
J. Travis Griffith
Gary Hoffman
Michael Janezic

Joseph L. Koepfinger*
David J. Law
Hung Ling
Andrew Myles
T. W. Olsen
Glenn Parsons
Ronald C. Petersen
Annette D. Reilly

Stephen J. Shellhammer
Adrian P. Stephens
Yatin Trivedi
Phillip Winston
Don Wright
Yu Yuan
Daidi Zhong

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 1735™-2014, 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). The audience for this standard includes IP producers, IP consumers, vendors of tools that process protected IP, and standards development groups for IP specification formats.

When the electronic design automation (EDA) industry began creating standards for use in specifying, simulating, and implementing electronic circuits, there was no active market for the exchange of electronic designs. As interest in software reuse developed, the EDA industry began to share design collateral as a means of controlling the cost of development and managing the timeline for completing their design projects. Because that shared IP had a measurable development cost, business leaders began to insist on recovering those costs through licensing to other potential uses. Technical measures for IP protection were developed to augment the legal contracts that governed such shared use.

As the means for technical protection of IP proliferated, there was increased pressure to incorporate that technology in the existing standards for representation of IP. Expertise in protection technologies (such as encryption) was scarce in those standards development organizations (SDOs), which resulted in a slow pace of development within a given standard and technical divergence as independent organizations attempted to solve similar problems. As the marketplace for IP exchange continued to mature, concerns about technical protection of IP extended to include an interest in defining and managing the scope of use for protected IP.

This standard has been created to consolidate EDA industry efforts to unify the technical protections for IP and to guide SDOs in aligning their work for interoperability and compatibility. It is also intended to share best practices in IP protection for those who use standards that incorporate such technology, such as VHDL (IEC 61691-1-1, IEEE Std 1076™) and SystemVerilog (IEEE Std 1800™).^a

Corrections were made to 7.4.3 as required by IEEE Std 1735-2014/Cor 1-2015.

^aInformation on references can be found in Clause 2.

Contents

1.	Overview	1
1.1	Scope	1
1.2	Purpose, value, and approach	2
1.3	Key characteristics of this standard	2
1.4	Conventions used in this standard	3
1.5	Use of color in this standard	3
1.6	Contents of this standard	4
2.	Normative references	5
3.	Definitions, acronyms, and abbreviations	5
3.1	Definitions	5
3.2	Acronyms and abbreviations	7
4.	Trust model	8
4.1	Stakeholders	8
4.2	Role of IP protection	8
4.3	Protection via encryption	9
4.4	Components of trust	10
5.	Interoperability	12
5.1	Background	12
5.2	version pragma	12
5.3	Basic interoperability	13
5.4	Use cases	16
5.5	Secure keyring	19
6.	Key management	21
6.1	Overview	21
6.2	Key considerations	21
6.3	Basic key exchange	22
6.4	Standard key exchange	23
6.5	Key scope recommendations	25
7.	Rights management [V2]	27
7.1	Introduction	27
7.2	Rights scope	27
7.3	Tool types and rights	27
7.4	Syntax and markup	28
7.5	Tamper-proof requirements	34
7.6	Complete tool block and rights example	34

8.	License management [V2]	36
8.1	Introduction	36
8.2	License system	36
8.3	License proxy	36
8.4	License specification	36
8.5	License proxy parameters.....	37
8.6	License use	38
8.7	Multiple envelopes	39
8.8	Proxy communication	40
8.9	License proxy transactions	40
8.10	License proxy commands.....	44
8.11	Deprecated licensing pragmas.....	45
9.	Visibility management	46
9.1	Introduction	46
9.2	Background [V1].....	46
9.3	Visibility in tool phases [V1]	47
9.4	Visibility and encryption envelopes [V1]	50
9.5	viewport pragmas	52
9.6	Programming language interfaces.....	58
9.7	Controlling visibility with rights [V2]	60
9.8	Visibility of dynamic objects [V2].....	61
9.9	Unresolved visibility issues.....	61
10.	Common rights [V2]	62
10.1	Defining common rights.....	62
10.2	Overriding common rights	62
10.3	Defaults and the delegated value.....	62
10.4	Common conditions for rights.....	63
10.5	Common right for error handling	64
10.6	Common right for visibility.....	65
10.7	Common right for child visibility.....	66
10.8	Common right for decryption.....	67
	Annex A (informative) Bibliography	68
	Annex B (informative) Other known issues with IP protection	69
	Annex C (normative) Protection pragmas	75

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

1.1 Scope

This standard specifies embeddable and encapsulating markup syntaxes for design intellectual property encryption and rights management, together with recommendations for integration with design specification formats described in other standards. It also recommends use models for interoperable tool and hardware flows, which will include selecting encryption and encoding algorithms and encryption key management. The recommendation includes a description of the trust model assumed in the recommended use models. This 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 indicating elements and 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 is partially derived from the common practice for the collection to be considered the intellectual property of one party. Hardware and software descriptions are encompassed by this term.

1.2 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 compile-time licenses. An integrated rights management scheme is also an element of the flow, which helps IP authors to control tool behavior including, but not limited to, IP visibility, 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 vendor arranged in a series of 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 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 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 practice for use of IP protection markup syntax and key management to enable interoperable tool flows with IP and tools from a wide array of suppliers. It includes 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.

1.3 Key characteristics of this standard

The marketplace for electronic design data depends on trust between the parties to the transactions. Exchanges require a confidence 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.

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 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 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 need to be accompanied by technical security to help enforce the terms of the transaction. The recommended practices described by this standard assist in the construction and maintenance of technical security consistent with the intent of the contracted terms of the transaction.

1.4 Conventions used in this standard

Each command description (syntax) consists of a command keyword followed by one or more parameters. The meta-syntax for the description of the syntax rules uses the conventions shown in [Table 1](#).

Table 1—Document conventions

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>
::=	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>

1.5 Use of color in this standard

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

- 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**.

1.6 Contents of this standard

The organization of the remainder of this standard is as follows:

- [Clause 2](#) provides references to other applicable standards that are presumed or required for this standard.
- [Clause 3](#) defines terms, acronyms, and abbreviations used throughout the different specifications contained in this standard.
- [Clause 4](#) describes a trust model for protecting and sharing IP.
- The next several clauses provide the technical background and make general recommendations for encrypting (and sharing) IP in specific areas: **interoperability**, **key management**, **rights management** (and rights control structures), **license management**, **visibility management**, and **common rights**.
- Annexes. Following [Clause 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

¹Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

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.

IEC 61691-1-1, IEEE Std 1076™, Behavioural languages—Part 1-1: VHDL language reference manual (Adoption of IEEE Std 1076-2002).^{2, 3}

IEEE Std 802®, IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture.⁴

IEEE Std 1003.1™, IEEE Standard for Information Technology—Portable Operating System Interface (POSIX™).

IEEE Std 1800™, IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language.

IETF RFC 3447, Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1.⁵

IETF RFC 5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.

ISO/IEC 9594-8, Information technology—Open Systems Interconnection—The Directory—Part 8: Public-key and attribute certificate frameworks.⁶

NIST Special Publication 800-131A, Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths.⁷

3. Definitions, acronyms, and abbreviations

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.⁸

3.1 Definitions

asymmetric encryption: An encryption/decryption scheme where the key to decrypt the data (private key) is different from the key to encrypt the data (public key).

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

²IEC publications are available from the International Electrotechnical Commission (<http://www.iec.ch/>). IEC publications are also available in the United States from the American National Standards Institute (<http://www.ansi.org/>).

³IEEE publications are available from The Institute of Electrical and Electronics Engineers (<http://standards.ieee.org/>).

⁴The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

⁵IETF documents (i.e., RFCs) are available for download at <http://www.rfc-archive.org/>.

⁶ISO/IEC publications are available from the ISO Central Secretariat (<http://www.iso.org/>). ISO publications are also available in the United States from the American National Standards Institute (<http://www.ansi.org/>).

⁷NIST publications are available from the National Institute of Standards and Technology (<http://www.nist.gov/>)

⁸*IEEE Standards Dictionary Online* subscription is available at:

http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html.

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

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

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

DER format: Distinguished Encoding Rules (X.690, ASN.1). A method of canonically representing certain data structures, here used to encode RSA Public Keys.

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—to ensure the message came from the claimed source.

digital envelope: A bounded data collection that uses encryption technology for protection and/or identification. *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](#).

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

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 RFC 2104)

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.

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 a interface to a third-party license server.

plaintext: Unencrypted data.

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.

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).

3.2 Acronyms and abbreviations

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
HDL	hardware description language
HMAC	Hash-based Message Authentication Code
IP	electronic design intellectual property
IV	initialization vector
LRM	language reference manual
NIST	National Institute of Standards and Technology
PLI	programming language interface
SDO	standards development organization
VHDL	VHSIC hardware description language
VHPI	VHDL programming interface
VHSIC	Very High Speed Integrated Circuit
VPI	SystemVerilog programming interface

4. Trust model

It is necessary for the stakeholders in IP protection to trust each other in order to do business. If one is to understand how encryption plays a role and therefore to trust it sufficiently, one needs to have a concept of the trust model it supports. In general, understanding the trust model helps 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, and why it is valuable. Finally, the components of the trust model are examined to understand the foundation they provide for trusting this form of IP protection.

4.1 Stakeholders

In any transaction there are parties of interest, those who hold a stake in the outcome of the transaction. 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 author. Their interest includes providing a valuable product and preventing loss through disclosure of their IP to either competition or IP users. Their ultimate interest is maximizing their return on 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. After providing satisfactory compensation to the IP author, the IP user receives the transacted rights to use the IP. Their interest includes choosing the right IP of the appropriate usefulness and quality and minimizing the risk and cost of mistakes in doing so. Their ultimate interest is in minimizing the total cost to use the right IP in their designs.

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

4.2 Role of IP protection

The pricing of an IP is based on a complex set of factors. 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 of an IP author. It is referred to here as the *risk premium*.

The goal of an IP protection mechanism is to mitigate the risk of loss due to disclosure. 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 does not happen. An IP protection mechanism itself may 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 should have a positive effect, of course. While loss due to disclosure is an IP author interest, it is not 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, 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](#) gives an overview of IP protection via encryption, described at a level to support further discussion of its trust model.

4.3 Protection via encryption

Encryption is a lossless transformation, not obfuscation or abstraction. The protected IP has 100% fidelity to its original representation. This IP can also be delivered to the IP user premises. In its encrypted form, and 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 allow IP use only through vendor tools 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 standard builds on previous work to define a common scheme of markup for encryption and decryption of IP. Following are the high level ideas for how it works.

- a) The IP author marks up the IP lightly to describe the encryption scheme, what aspects are to be protected, what remains visible, what rights the IP user has, and so on.
- b) In terms of that encryption scheme, the primary use case supported by this standard is the *Digital Envelope* model.
 - 1) In this model, the IP author chooses a set of vendor tools to enable access to their IP.
 - 2) Each tool has its own public/private key pair that is used to encrypt the common session key used to protect the IP.
- c) In the resulting protected IP, there is a key block for each vendor tool that contains the encrypted, common session key.
 - 1) That key block can only be decrypted by the specific tool vendor whose key was used to protect it.
 - 2) With that session key, the tool can decrypt the protected IP.

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

4.4 Components of trust

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](#) for details on the required algorithms.

4.4.2 Digital identity verification

In the primary digital envelope use 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 tokens, that are offered correspond to the implied stakeholder. There are many strategies for introducing fraudulent identities, and there are corresponding defenses against this risk. This standard accounts for this trust model component in its key management recommendations (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., to trust that the tool vendor will not intentionally disclose the 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. There is also an opportunity for the IP author to consider: An agreement with the tool vendor for restricted access to decrypted source code can aid both parties with IP user support issues.

4.4.4 Security of design flow

The weakest link in a chain is where it is most vulnerable. It is important to consider all the tools in the design flow that the IP author 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 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, but also depends on other data flow, no protected information should be exposed in that other data flow either.

Other well-known data formats that are unprotected, for example, Standard Delay Format (SDF) and Unified Power Format (UPF), present a challenge in this respect. It does not matter whether it is a standard or proprietary data representation, allowing interoperability with exposed data is a gap that risks disclosure. 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 choices of what tools to use in a design flow 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 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 use here.

Tool vendors instill confidence in their tools through quality of 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 (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 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 give stakeholders an ability to consider the risks and rewards of this IP protection mechanism. All the stakeholders, acting in their own self-interests, should find considerable value in it and recognize its strengths in safeguarding IP and 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

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.

5.2 version pragma

A new pragma is introduced here to provide versioning for the encryption pragma definitions and use models. It allows the flexibility to make changes and preserve backward compatibility for IP encrypted with a previous version. The syntax [as expected to be expressed in the respective language reference manuals (LRMs)]:

For SystemVerilog:

```
`pragma protect version = <number>
```

For VHDL:

```
`protect_version_directive :: = `protect version = integer
```

The legal values are 0, 1, 2 and defined as follows:

0—conforms to the SystemVerilog and VHDL LRMs with tool-specific interpretation of the use model and pragma syntax. There is no expectation of interoperability. This version only exists to support backward compatibility to older implementations.

⁹Information on references can be found in [Clause 2](#)

1—conforms to IEEE 1735 basic interoperability specifications, which are important both because they allow basic interoperability of protected IP across a set of tools and because there is already defacto conformance to them at this time. Clauses and subclauses marked [V1] in this standard contain level 1 conformance requirements.

2—conforms to all IEEE 1735 recommendations for use model and pragmas, including all *Version 1* requirements. Clauses and subclauses marked [V2] in this standard contain level 2 conformance requirements.

Clauses or subclauses not marked [V1] or [V2] are applicable to all versions.

If the **version** pragma is not present in an encryption or decryption envelope, the version is presumed to be level 0 (no expectation of interoperability).

5.3 Basic interoperability

5.3.1 Recommendations required for encryption interoperability [V1]

The required interoperable use case is the digital envelope. The digital envelope uses both symmetric and asymmetric encryption algorithms as well as encoding algorithms. As a consequence, the recommendations on these algorithms are also required and are considered fundamental to *Version 1* interoperability.

In general, algorithms should be freely available with open reference implementations and be complete enough to ensure interoperability. However, in recommending algorithms, there are some concerns. First, other algorithms are already specified in the existing LRMs. Second, these algorithms need to be considered, along with newer algorithms, in light of their adequacy to protect against the latest security threats. To do so, IEEE Std 1735 uses the National Institute of Standards and Technology (NIST) Special Publication 800-131A as the relevant source and standard.

The following terms are used in the remainder of this subclause.

Definitions of NIST recommendations

- a) *Approved* means the algorithm is specified in a Federal Information Processing Standards (FIPS) or NIST Recommendation (published as a NIST Special Publication).
- b) *Acceptable* means the algorithm and key length is safe to use; no security risk is currently known.
- c) *Disallowed* means the specific algorithm and key length have been withdrawn from FIPS or NIST Recommendations.
- d) *Legacy use* means the algorithm or key length may be used to process already protected information (e.g., to decrypt ciphertext data), but there may be risk in doing so. Methods for mitigating this risk should be considered.

Definitions of IEEE 1735 recommendations

- e) *Recommended* means these algorithms should be used to achieve security and interoperability. Tools are expected to support recommended algorithms in creating or reading *Version 1* or *Version 2* digital envelopes.
- f) *Deprecated* means the algorithms should not be used. Tools are only expected to support these algorithms for non-interoperable *Version 0* digital envelopes. Tools are not expected to support deprecated algorithms in creating or reading *Version 1* or *Version 2* digital envelopes.
- g) *Legacy use* means the algorithm is no longer recommended, but is still in frequent practical use by either tools or IP authors. Tools are expected to support Legacy use algorithms.

5.3.1.1 Recommendation for session key encryption

For symmetric session key encryption, the IEEE 1735 recommendations are given in [Table 2](#).

Table 2—Symmetric session key encryption

Algorithm (key length)	Security	NIST recommendation	IEEE 1735 recommendation
DES	Compromised	Disallowed	<i>Deprecated</i>
3DES(TDEA) 3 keys version	Secure	Acceptable	<i>Deprecated</i>
AES (128 and 256)	Secure	Acceptable	<i>Recommended</i>

Although the existing LRMs require the `DES` algorithm, it is no longer allowed by the NIST as its security has been compromised. Therefore, IEEE Std 1735 recommends that the `des-cbc` algorithm be considered *Deprecated* for use in symmetric session key encryption. Although secure at the moment, `3des_cbc` is not generally believed to be a long-term option, so it too carries the recommendation of *Deprecated*. AES encryption, with key lengths of either 128 or 256, is required (`aes128-cbc` and `aes256-cbc`, respectively).

5.3.1.2 Recommendation for asymmetric encryption

For asymmetric encryption, the IEEE 1735 recommendations are given in [Table 3](#).

Table 3—Asymmetric encryption

Algorithm (key length)	Operation	Security	NIST recommendation	IEEE 1735 recommendation
Diffie-Hellman	All operations	Varies	Most disallowed	<i>Deprecated</i>
RSA (<2048)	Encryption/signing	Questionable	Disallowed	<i>Legacy use</i>
RSA (<2048)	Decryption/ verification	Questionable	Legacy use	<i>Legacy use</i>
RSA (>=2048)	All operations	Secure	Acceptable	<i>Recommended</i>

While mentioned in the existing LRMs, most versions of `Diffie-Hellman` are disallowed by NIST now. Some tools are using `RSA` with key lengths less than 2048. IEEE Std 1735 recommends that tool vendors stop using these key lengths; but for now, they should be supported by tools consistent with the *Legacy use* model.

Instead, `RSA` encryption with a key length greater than or equal to 2048 is the recommended asymmetric encryption methodology. For interoperability, it is also necessary to specify encoding and padding. Therefore, `RSAAES-PKCS1-V1_5` should be used (see IETF RFC 3447).

5.3.1.3 Recommendation for encoding

While raw encoding may be useful for tool vendors during internal debugging, `base64` is required for interoperable encoding. Any other methodology may be considered *Deprecated*.

5.3.1.4 Recommendation for digest algorithms

For computation of digests (see [7.4.3](#)), the IEEE 1735 recommendations are given in [Table 4](#).

Table 4—Calculating digests

Algorithm (key length)	Security	NIST recommendation	IEEE 1735 recommendation
MD4	Compromised	Disallowed	<i>Deprecated</i>
MD5	Compromised	Disallowed	<i>Deprecated</i>
SHA-1	Questionable	Legacy use	<i>Deprecated</i>
SHA-2 (SHA-256,512)	Secure	Acceptable	<i>Recommended</i>

The recommended algorithm for calculating digests is SHA-2, either 256 or 512.

5.3.2 Salting the session key for version tampering [V2]

To support tamper-proofing of the **version** pragma, the session key needs to be salted for `version` pragma values of 2 or greater, with a formula based on the version value at encryption time. This ensures an IP author's rights cannot be ignored simply by changing the plain text value of the **version** pragma.

During *Version 2* IP creation, the IP can be encrypted with a **session** key, then salt it by adding (*version pragma value*−1), then encrypt the result with public keys to create the key blocks. Since *Version 1* does not allow for salting of the **session** key, the preceding formula effectively adds 0 to a *Version 1* pragma, thus allowing encryption tools to use a single algorithm for both *Version 1* and *Version 2* session keys. When a tool decrypts a *Version 2* IP, decrypt the key block, subtract (*version*−1) to recreate the original session key and then proceed to decrypt the IP data. Addition and subtraction is done modulo $2^{<\text{key length}>}$ to avoid overflow.

For example, for a version pragma value of 2, to encrypt an IP with a 128-bit AES key of 0123456789ABCDEF, the IP author adds 1 (*Version 2* minus 1) for a salted key of 0123456789ABCDF0. This is the value encrypted and encoded into the key blocks. When processing the decryption envelope, the tool retrieves 0123456789ABCDF0 from its key block, subtracts 1 (*Version 2* minus 1) to get the unsalted original key 0123456789ABCDEF that is used to decrypt the data block.

5.3.3 Recommended changes to VHDL and SystemVerilog for basic IP exchange [V1]

There are four major issues to resolve and achieve interoperability here.

The first issue clarifies the RSA algorithm conventions. Although both the SystemVerilog and VHDL LRMs just specify “rsa,” the conventions that shall be used are RSAES-PKCS1-V1_5 (see IETF RFC 3447). This essentially specifies the DER encoding/padding as the representation and standardizing this makes interoperable solutions possible. This is listed as a requirement in [5.3.1.2](#).