American National Standard for Financial Services

ANSI X9.69–2006

Framework for Key Management Extensions

Accredited Standards Committee X9, Incorporated
Financial Industry Standards

Date Approved:

American National Standards Institute

Contents

Page

Figures

Forword

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made toward their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretation should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken to reaffirm, revise, or withdraw this standard no later than five years from the date of approval.

Published by

Accredited Standards Committee X9, Incorporated
Financial Industry Standards
P.O. Box 4035
Annapolis, MD 21403 USA
X9 Online http://www.x9.org

Introduction

This Standard is concerned with symmetric key systems in which the encrypting key and decrypting key are identical. The security and reliability of any process based on a symmetric cryptographic algorithm is directly dependent on the protection afforded to the secret quantity, called the key. Thus, no matter how strong the algorithm, the system is only as secure as its key management method.

This Standard defines two specific key management methods for controlling and handling keys, called (1) Constructive Key Management and (2) Key Usage Control. Each method can be used independently; or the methods can be used in combination. However, the combined use of the methods is highly recommended by the ASC X9 Subcommittee responsible for this Standard. Each method is described in a separate section of the Standard.

The section on CONSTRUCTIVE KEY MANAGEMENT, systematizes key creation, implementing “dual control” or “split knowledge” by using key components to construct the final working key. This working key may be used in several ways including as a session key, for a store-and-forward (i.e. e-mail) application, and for file encryption applications, such as archiving, or protecting filed information until needed again by the user. Other applications are also possible. Until now, this practice of split knowledge key creation has been used mainly to transport key parts into systems where “master keys” were used to protect keys in storage, and to recover the working keys for a current application. With the methodology of this Standard, a working key will be created as needed for a specific encryption process, and re-created when needed to decrypt the object. Depending on the application, the key may be saved or destroyed after each use. The working key is never transmitted; the application program only knows it while it is in use.

The section on KEY USAGE CONTROL, allows the creator of a key to specify the allowed uses of the key. For example, key usage control information can be used to distinguish key types (data, PIN, or key-encrypting). The type “data key” can be further sub-divided to distinguish data privacy keys—keys used to encrypt and decrypt data—from Message Authentication Code (MAC) keys---keys used to protect the integrity of data. The method attaches or binds a “key usage vector” to each generated key, for the life of the key, and is used by the system to ensure that keys are used properly. In short, the key usage vector prevents abuses and attacks against the key. The key usage vector can be used to protect keys stored within a single system, or to protect keys transmitted from one system to another.

This Standard is algorithm independent, and as new cryptographic algorithms with perhaps longer key lengths than currently in use are developed and adopted by the Financial Community this Standard will still apply.

NOTE The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights.

By publication of this standard, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. The patent holder has, however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the standards developer.

Suggestions for the improvement or revision of this Standard are welcome. They should be sent to the X9 Committee Secretariat, Accredited Standards Committee X9, Inc., Financial Industry Standards, P.O. Box 4035, Annapolis, MD 21403 USA.

This Standard was processed and approved for submittal to ANSI by the Accredited Standards Committee on Financial Services, X9. Committee approval of the Standard does not necessarily imply that all the committee members voted for its approval.

The X9 committee had the following members:
Jim Schaffer, X9 Chairman
Vincent DeSantis, X9 Vice-Chairman
Cynthia Fuller, Executive Director
Susan Yashinskie, Managing Director

Organization Represented	Representative	
American Bankers Association	C. Diane	Poole
American Express Company	John	Allen
American Financial Services Association	Mark	Zalewski
Bank of America	Daniel	Welch
Certicom Corporation	Daniel	Brown
Citigroup, Inc.	Gary	Word
Clarke American Checks, Inc.	John	McCleary
CUSIP Service Bureau	James	Taylor
Deluxe Corporation	John	FitzPatrick
Diebold, Inc.	Bruce	Chapa
Discover Financial Services	Katie	Howser
Federal Reserve Bank	Dexter	Holt
First Data Corporation	Rick	Van Luvender
Fiserv	Skip	Smith
FSTC, Financial Services Technology Consortium	Daniel	Schutzer
Hewlett Packard	Larry	Hines
Hypercom	Scott	Spiker
IBM Corporation	Todd	Arnold
Ingenico	John	Spence
Intuit, Inc.	Jana	Hocker
J.P. Morgan Chase & Co	Jacqueline	Pagan
KPMG LLP	Mark	Lundin
MagTek, Inc.	Carlos	Morales
MasterCard International	William	Poletti
National Association of Convenience Stores	Michael	Davis
National Security Agency	Sheila	Brand
NCR Corporation	Steve	Stevens
Proofspace	Paul	Doyle
SWIFT/Pan Americas	James	Wills
U.S. Bank	Marc	Morrison
University Bank	Stephen	Ranzini
VECTORsgi	Ron	Schultz
VeriFone, Inc.	Brad	McGuinness
VISA	Richard	Sweeney
Wachovia Bank	Raymond	Gatland
Wells Fargo Bank	Ruven	Schwartz

The X9F subcommittee on Information Security had the following members:

Dick Sweeney, X9F Chairman
Sandra Lambert, X9F Vice Chair

Organization Represented	Representative	
3PEA Technologies, Inc.	Mark	Newcomer
ACI Worldwide	Douglas	Grote
ACI Worldwide	Julie	Samson
ACI Worldwide	Jim	Shaffer
ACI Worldwide	Sid	Sidner
American Bankers Association	Tom	Judd
American Express Company	John	Allen
American Express Company	Richard	Rodriguez
American Express Company	Vicky	Sammons
American Financial Services Association	Mark	Zalewski
Bank of America	Daniel	Welch
Certicom Corporation	Daniel	Brown
Citigroup, Inc.	Paul	Gubiotti
Citigroup, Inc.	Susan	Rhodes
Citigroup, Inc.	Gary	Word
Clarke American Checks, Inc.	John	McCleary
Clarke American Checks, Inc.	John	Petrie
ClearWave Electronics	Mark	Ross
CUSIP Service Bureau	Scott	Preiss
CUSIP Service Bureau	James	Taylor
DeLap, White, Caldwell and Croy, LLP	Darlene	Kargel
Deluxe Corporation	John	FitzPatrick
Deluxe Corporation	Mike	Valiquet
Depository Trust and Clearing Corporation	Robert	Palatnick
Diebold, Inc.	Bruce	Chapa
Diebold, Inc.	Laura	Drozda
Discover Financial Services	Julie	Shaw
ECCHO	Phyllis	Meyerson
Federal Reserve Bank	Jeannine M.	DeLano
Federal Reserve Bank	Neil	Hersch
Federal Reserve Bank	Dexter	Holt
First Data Corporation	Tina	McGowan
First Data Corporation	Rick	Van Luvender
Fiserv	Bud	Beattie
Fiserv	Mary	Bland
Fiserv	Kevin	Finn
Fiserv	Dennis	Freiburg
FSTC, Financial Services Technology Consortium	Frank	Jaffe
FSTC, Financial Services Technology Consortium	Daniel	Schutzer
Futurex	Jason	Anderson
Futurex	Greg	Schmid
Hewlett Packard	Larry	Hines
Hewlett Packard	Susan	Langford
Hypercom	Scott	Spiker

IBM Corporation	Todd	Arnold
InfoGard Laboratories	Tom	Caddy
Ingenico	John	Spence
Innove	Steven	Teppler
J.P. Morgan Chase & Co	Edward	Koslow
John H. Harland Company	Curt	Siroky
MagTek, Inc.	Terry	Benson
MagTek, Inc.	Jeff	Duncan
MagTek, Inc.	Carlos	Morales
MasterCard International	Jeanne	Moore
MasterCard International	Michael	Ward
National Institute of Standards and Technology	Elaine	Barker
National Institute of Standards and Technology	William	Burr
National Institute of Standards and Technology	David	Cooper
National Institute of Standards and Technology	Randall	Easter
National Institute of Standards and Technology	Sharon	Keller
National Institute of Standards and Technology	John	Kelsey
National Institute of Standards and Technology	Fernando	Podio
National Security Agency	Mike	Boyle
National Security Agency	Sheila	Brand
National Security Agency	Greg	Gilbert
National Security Agency	Tim	Havighurst
National Security Agency	Debby	Wallner
NCR Corporation	Ali	Lowden
NCR Corporation	David	Norris
NCR Corporation	Ron	Rogers
NCR Corporation	Ally	Whytock
NCR Corporation	Hui	Wu
NTRU Cryptosystems, Inc.	Nick	Howgrave-Graham
NTRU Cryptosystems, Inc.	William	Whyte
Pitney Bowes, Inc.	Leon	Pintsov
Proofspace	Paul	Doyle
Rosetta Technologies	Jim	Maher
Rosetta Technologies	Paul	Malinowski
RSA Security, Inc.	James	Randall
RSA Security, Inc.	Steve	Schmalz
Surety, Inc.	Dimitrios	Andivahis
TECSEC Incorporated	Ed	Scheidt
Thales e-Security, Inc.	Tim	Fox
Thales e-Security, Inc.	James	Torjussen
The Clearing House	Vincent	DeSantis
The Clearing House	Henry	Farrar
The Clearing House	Susan	Long
Triton Systems of Delaware, Inc.	Daryll	Cordeiro
U.S. Bank	Marc	Morrison
Unisys Corporation	David J.	Concannon
Unisys Corporation	Navnit	Shah
University Bank	Stephen	Ranzini
VECTORsgi	Ron	Schultz

VeriFone, Inc.	John	Barrowman
VeriFone, Inc.	David	Ezell
VeriFone, Inc.	Dave	Faoro
VeriFone, Inc.	Brenda	Watlington
VISA	Chackan	Lai
VISA	Stoddard	Lambertson
Voltage Security, Inc.	Luther	Martin
Wachovia Bank	Raymond	Gatland
Wachovia Bank	David	Naelon
Wachovia Bank	Keith	Ross
Wells Fargo Bank	Mick	Bauer
Wells Fargo Bank	Jeff	Jacoby
Wells Fargo Bank	Eric	Lengvenis
Wells Fargo Bank	Farah	Moaven
Wells Fargo Bank	Chuck	Perry
Wells Fargo Bank	Ruven	Schwartz
Wells Fargo Bank	Craig	Shorter
Wells Fargo Bank	Tony	Stieber

Under ASC X9, Inc. procedures, a working group may be established to address specific segments of work under the ASC X9 Committee or one of its subcommittees. A working group exists only to develop standard(s) or guideline(s) in a specific area and is then disbanded. The individual experts are listed with their affiliated organizations. However, this does not imply that the organization has approved the content of the standard or guideline. (Note: Per X9 policy, company names of non-member participants are listed only if, at the time of publication, the X9 Secretariat received an original signed release permitting such company names to appear in print.)

The X9F4 Cryptographic Protocol and Application Security working group which developed this standard had the following members:

Jeff Stapleton, X9F4 Chairman and Project Editor
Sandra Lambert, X9F4 Vice Chair

Organization Represented	**Representative**	
Bank of America	Andi	Coleman
Certicom Corporation	Scott	Vanstone
Clarke American Checks, Inc.	John	McCleary
Clarke American Checks, Inc.	John	Petrie
Clarke American Checks, Inc.	Steve	Smith
Comet Capital, LLC	Lawrence T.	Levine
Comet Capital, LLC	Miranda	Watson
DeLap, White, Caldwell and Croy, LLP	Steve	Case
DeLap, White, Caldwell and Croy, LLP	Darlene	Kargel
Diebold, Inc.	Bruce	Chapa
Diebold, Inc.	Anne	Doland
Diebold, Inc.	Scott	Harroff
Discover Financial Services	Julie	Shaw
Entrust, Inc.	Miles	Smid
Ernst and Young	Keith	Sollers
Federal Reserve Bank	Neil	Hersch

Federal Reserve Bank	Deb	Hjortland
Federal Reserve Bank	Dexter	Holt
First Data Corporation	Lilik	Kazaryan
First Data Corporation	Todd	Nuzum
Fiserv	Dennis	Freiburg
Fiserv	Dan	Otten
FSTC, Financial Services Technology Consortium	Frank	Jaffe
FSTC, Financial Services Technology Consortium	Christine	Nautiyal
Futurex	Jason	Anderson
Futurex	Greg	Schmid
Griffin Consulting	Harriette	Griffin
Griffin Consulting	Phil	Griffin
Hewlett Packard	Larry	Hines
Hypercom	Scott	Spiker
Hypercom	Gary	Zempich
IBM Corporation	Todd	Arnold
IBM Corporation	Phil	Griffin
IBM Corporation	Michael	Kelly
InfoGard Laboratories	John	Attala
InfoGard Laboratories	Tom	Caddy
InfoGard Laboratories	Ken	Kolstad
Ingenico	Alexandre	Hellequin
Ingenico	John	Spence
Innove	Jarid	Cottrel
Innove	Brad	Morrison
Innove	Cindy	Morrison
Innove	Ralph	Poore
Innove	Steven	Teppler
KPMG LLP	Steven	Berhorst
KPMG LLP	Mark	Lundin
MagTek, Inc.	Terry	Benson
MasterCard International	William	Poletti
National Institute of Standards and Technology	Elaine	Barker
National Institute of Standards and Technology	Lily	Chen
National Security Agency	Sheila	Brand
National Security Agency	Greg	Gilbert
National Security Agency	Tim	Havighurst
National Security Agency	Paul	Timmel
nCipher Corporation Ltd.	Ron	Carter
NCR Corporation	Wayne	Doran
NCR Corporation	Charlie	Harrow
NCR Corporation	Steve	Stevens
NTRU Cryptosystems, Inc.	Ari	Singer
NTRU Cryptosystems, Inc.	William	Whyte
Proofspace	Paul	Doyle
Proofspace	Yuxin	Ruan
Proofspace	Bob	West
RSA Security, Inc.	Burt	Kaliski
RSA Security, Inc.	James	Randall
Sun Microsystems PS	Joel	Weise

Surety, Inc.	Dimitrios	Andivahis
TECSEC Incorporated	Ed	Scheidt
TECSEC Incorporated	Dr. Wai	Tsang
Thales e-Security, Inc.	Tim	Fox
Thales e-Security, Inc.	James	Torjussen
Triton Systems of Delaware, Inc.	Daryll	Cordeiro
Triton Systems of Delaware, Inc.	Bob	Douglas
U.S. Bank	Peter	Skirvin
U.S. Bank	Rush	Wilson
Unisys Corporation	David J.	Concannon
University Bank	Stephen	Ranzini
University Bank	Michael	Talley
VECTORsgi	Jerry	Bowman
VECTORsgi	Ron	Schultz
VeriFone, Inc.	Dave	Faoro
VeriFone, Inc.	Doug	Manchester
VISA	Chackan	Lai
VISA	Richard	Sweeney
Voltage Security, Inc.	Luther	Martin
Wachovia Bank	David	Naelon
Wells Fargo Bank	Mick	Bauer
Wells Fargo Bank	Jeff	Jacoby
Wells Fargo Bank	Eric	Lengvenis
Wells Fargo Bank	Mike	McCormick
Wells Fargo Bank	Farah	Moaven
Wells Fargo Bank	Doug	Pelton
Wells Fargo Bank	Mike	Rudolph
Wells Fargo Bank	Ruven	Schwartz

This document cancels and replaces X9.69-1999 Framework for Key Management Extensions in whole. X9.69-2006 was revised to address the industry transition from single DES to Triple DES, with the following changes:

A. Reference to the following documents were added:

- X9.52 Triple Data Encryption Algorithms (3DEA) Modes of Operation
- FIPS 197 Advanced Encryption Standard (AES)
- IEEE Cryptography Transitions

B. References to the following withdrawn X9 standards were deleted:

- X9.9 Financial Institution Message Authentication Codes (MAC) Wholesale; refer to Technical Guideline: Technical Guideline: Managing Risk and Migration Planning: Withdrawal of ANSI X9.9 (X9 TG-24-1999)
- X9.17 Financial Institution Key Management (Wholesale); refer to Technical Guideline: Managing Risk and Migration Planning: Withdrawal of ANSI X9.17 (X9 TG-26 – 1999)

C. The follow terms were changed:

- Policy Manager was changed to CKM Administration
- Labels was changed to Credentials

- Credential Manager was changed to Token Distribution

D. The document was converted to the current X9/ISO standards template.

At the time the original X9.69-1999 was approved, the X9 committee had the following members:

Harold Deal, X9 Chairman
William E. Lyons, X9 Vice Chairman
Cynthia Fuller, Managing Director

Organization Represented	**Representative**
American Bankers Association	Anne Livingston
	Kawika Daguio
American Express Company	Bonnie Howard
Applied Communications	Douglas Grote
	Cindy Rink
Automated Financial Services	Tom Clute
Banc One Services Corporation	William Lyons
Bank of America	Gretchen Breiling
Bankers Roundtable	Kit Needlam
	Keviar Warner
Canadian Bankers Association	Christine ArjoonLaL
	Mark Bakic
Chase Manhattan Bank	Christopher Dowdell
	Francis Keenan
Citibank	Seymour Rosen
Cybersafe Corp	Glenda Barnes
Deluxe Corporation	Maury Jansen
Ernst & Young, LLP	Geoffery Turner
	Richard Kastner
	Ralph Poore
Federal Reserve Bank	Dexter Holt
	Susan Belisle
Ferris & Associates, Inc.	Martin Ferris
First Data Corporation	Gene Kathol
Greenlee & Associates	Blake Greenlee
IBM Corporation	Harry Hankla
	Donald Harman
Intel Corporation	Pamela Warren
	Steve Ellis
KPMG Peat Marwick LLP	Jon Graff
	Jeff Stapleton
MARS Electronic International	E. E. Barnes
	Ron Bernardini
MasterCard International	Melinda Yee
Mellon Bank, N.A.	David Taddeo
	Genien Carlson
Merrill Lynch	John Dolan
Moore Business Forms Inc.	Thomas Oswald
National Association of Convenience Stores	Robert Swanson
National Security Agency	Gerard Rainville
NationsBanc	Harold Deal
NCR	Suzette Albert
New York Clearing House	Vincent DeSantis
NOVUS Services, Inc.	Thomas Kossler

	Peggy Douds
	David Pratscher
Pitney Bowes, Inc.	Leon Pintsov
Price Waterhouse Coopers	Jeff Zimmerman
Russell Technology Associates	James Russell
SPYRUS	Peter Yee
	Karen Randall
Unisys Corporation	Thomas Hayosh
	James Graziano
VeriFone, Inc.	John Sheets
	Glenn Kramer
	Stuart Taylor
Visa International	Bill Chen
Wells Fargo Bank	Tim Silva
Xcert International	Marc Branchaud
	Sandra Lambert

At the time the original X9.69-1999 was approved, the X9F subcommittee on Information Security had the following members:

Glenda Barnes, X9F Chairman

Organization Represented	**Representative**
American Bankers Association	Kawika Daguio
American Express Company	Bonnie Howard
	Glenn Weiner
Applied Communications Inc.	Cindy Rink
	Douglas Grote
	Dennis Abraham
Bank of America	Kathleen Gibbons
	Mack Hicks
	Richard Phillips
	Martin Johnson
Bank One Corp	Duane Baldwin
Bankers Roundtable	Keviar Warner
	Frederick Honold
CertCo LLC	Richard Ankney
	Daniel Geer
Certicom Corporation	Don B. Johnson
Chase Manhattan Bank	Gene Rao
	Richard Yen
Communications Security Establishment	Michael Chawrun
	Alan Poplove
Cybersafe Corp.	Glenda Barnes
	David O'Brien
Cylink Corporation	Kamy Kavianian
	Lily Lidong Chen
Deluxe Corporation	Cory Surges
	Maury Jansen
	Chuck Bram
Digital Equipment corporation	Donald Holden
Entrust Technologies	Robert Zuccherato
	Tim Moses
Ernst & Young, LLP	Richard Kastner
	Ralph Spencer Poore

Federal Reserve Bank	Richard Sweeney
	Michael Versace
	Gary Chaulklin
First Data Corporation	Gene Kathol
First Union Corporation	James Ramsay
	Sandra Lambert
Food Marketing Institute	Ted Mason
	Joy Nicholas
Fortress Technologies	Eva Bozoki
Gilbarco Inc.	Rena Smith
Griffin Consulting	Phillip Griffin
GTE Internetworking	Patrick Cain
Harmonic Systems Incorporated	Daniel Hunt
IBM Corporation	Mohammad Peyravian
	Harry Hankla
	Stephen Mike Matyas
Intel Corporation	Pamela Warren
	Steve Ellis
IIT Research Institute	Roger Westman
KPMG Peat Marwick LLP	Jeffrey Stapleton
M. Blake Greenlee Associates, Ltd.	Blake Greenlee
MasterCard International	R.O. Karlin
	William Poletti
Mellon Bank, N.A.	David Taddeo
Merrill Lynch	Lawrence LaBella
	John Dolan
	Ted Gerbracht
National Association of Convenience Stores	Robert Swanson
National Security Agency	Gerard Rainville
NCR	Mark Liddle
NIST	Donna Dodson
	Miles Smid
Northstar Technology Group, Inc.	John Bowman
Pitney bowes, Inc.	Andrei Obrea
Price Waterhouse Coopers	John Hunt
	David Oshman
	Jeffrey Zimmerman
Pulse EFT Association	Karen Gardstein
	Leslie Hendrix
Racal Guardata, Inc.	Scott Petersen
	Emile Soueid
	Samuel Epstein
SAIC	Wanda Gamble-Braggs
Security Dynamics	Burt Kaliski
SPYRUS	Karen Randall
	Peter Yee
Technical Communications Corporation	John Gill
TECSEC Incorporated	Edward Scheidt
	Pud Reaver
	Jay Wack
VeriFone, Inc.	John Sheets
	Stuart Taylor
	Trong Nguyen
VISA International	Willaim Chen
Wells Fargo Bank	Azita Amini
	Terry Leahy

Xcert International, Inc.	Marcus Branchaud
	Sandra Lambert

At the time the original X9.69-1999 was approved, the X9F3 working group that developed this Standard had the following members:

Gary Chaulklin, Chair X9F3

Organization Represented	Representative
Abraham & Associates	Dennis G. Abraham
AT&T	Bill Oeschger
Certco LLC	Richard Ankney
Certicom	Don B. Johnson
Chase Manhattan Bank	Richard Yen
Citigroup	Perry Gleason
Communications Devices Inc	Tadgh Kelly
Compaq	Don Holden
Coopers & Lybrand	Victor Blanchard
CyberSafe	Glenda Barnes
Cylink Corporation	Kamy Kavianian
Delap, White, Caldwell & Croy, LLP	Darlene Kargel
Diebold	Sandra Morgan
Dresser Industries	Mike Biskobing
Ernst & Young	Ralph Poore
	Rick Kastner
EXXON Company	John Pratt
Federal Reserve Bank	Richard Sweeney
	Gary Chaulklin
First Union Bank	Jim Ramsay
Gilbarco Inc	Rena Smith
	Tim Dickson
GTE Internetworking	Pat Cain
Hitachi Data Systems	Bill Cox
IBM	Stephen M. Matyas
	Mohammad Peyravian
InfoGard Labs	Les Biggs
IRE	Doug Kozlay
IVI Checkmate	John Spence
JL Information Solutions	Jan Lovorn
Jones Futurex	Gerry Scott
KPMG Peat Marwick	Eric Ashdown
MasterCard	Carl Campbell
NIST	Elaine Barker
	Jim Foti
NSA	Gerard Rainville
PNC Bank	Tim Garland
PULSE	Vivian M. Banki
Schlumberger Ind	Richard Carpenter
Spyrus	Karen Randall
	Peter Yee
TECSEC	Ed Scheidt
	Jay Wack
	Clarence Reaver
U. S. Bancorp	Jeanne Fagan
Verifone	John Sheets

	Ken Gillman
VISA International	Rick Hite
Wells Fargo Bank	Azita Amini

Framework for Key Management Extensions

1 Scope

This Standard defines methods for the generation and control of keys used in symmetric cryptographic algorithms. The Standard defines a *constructive method* for the creation of symmetric keys, by combining two or more secret key components. The Standard also defines a method for attaching a *key usage vector* to each generated key that prevents abuses and attacks against the key. The two defined methods can be used separately or in combination.

The Standard does not cover aspects of key management, such as:

- Key establishment mechanisms;

  See for example ANSI X9.24 Financial Institution Key Management (Retail), or ISO/IEC 11770-2, Key Management, Part 2: Mechanisms using symmetric techniques.

- Mechanisms to store, archive, delete, destroy, etc. keys;

- Mechanisms for key recovery in the event of the failure or loss of keys.

The Standard also does not define the implementation of key management mechanisms; there may be different products that comply with this Standard and yet are not interoperable.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ANS X3.92-1981 Data Encryption Algorithm

ANS X3.106-1983 Data Encryption Algorithm - Modes of Operation

ANS X9.19 Financial Institution Retail Message Authentication

ANS X9.52 Triple Data Encryption Algorithms (3DEA) Modes of Operation

FIPS 197 Advanced Encryption Standard (AES)

3 Terms, symbols and abbreviated terms

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

3.1 3DEA
Triple Data Encryption Algorithm

3.2 AES
Advanced Encryption Standard

3.3 C
Key Usage Control Vector

3.4 CBC
Cipher Block Chaining - one of the four modes of 3DEA

3.5 CKM
Constructive Key Management

3.6 ECB
Electronic Codebook - one of the four modes of 3DEA

3.7 K
Key

3.8 MAC
Message Authentication Code

3.9 PIN
Personal Identification Number

3.10 PR
Private (secret) key of a public key encryption algorithm

3.11 PU
Public (non-secret) key of a public key encryption algorithm

3.12 S
Cryptographic Services and Modes provided in the Key Management System

3.13 SMIB
Security Management Information Base

3.14 U
Usage Field; a binary vector where the bit field specifies the use(s) for each key

4 Application

4.1 General

In a cryptographic system it may be desirable to generate keys using a constructive process, where keys are derived from system-specified control information, as well as secret random data. It may also be desirable to attach a "key usage vector" to each key, defining how the key shall be used.

4.2 The Use of Constructive Key Management

With Constructive Key Management (CKM), key components, called splits, shall be generated with a random or pseudorandom number generator. Each of these splits shall be given a name, called a Credential that provides some meaningful information to the sender, and allows the sender to direct the encrypted object to a selected set of end-users. The working key shall be constructed by combining the addressee splits with the system generated and controlled splits. Thus, with CKM it is possible to create a group key for a particular set of end-users. Other recipients, who are not members of the group, will be unable to re-construct that particular group key.

4.3 The Use of Key Usage Control Vector

With Key Usage Control Vectors, keys shall be generated using any acceptable method of key generation. Then a key usage vector shall be attached to the key. This vector specifies cryptographic services, modes and key parameters, in which the associated key shall be used. This usage vector shall be securely bound to the key to prevent misuse of the key or misinterpretation of its use.

4.4 System Algorithm and System Key

The CKM operations system shall use a common system-wide encryption algorithm and key to wrap the header of encrypted objects as they transit communications networks. The purpose is privacy, not security. For example, when multiple objects are sent in a batch mode, recipients need to be able to unwrap the bundle and determine from the encrypted header information which object(s) is addressed to them. Security is not compromised because the objects themselves are encrypted using secret splits that only the addressees of each object possess.

5 Constructive Key Management

5.1 Overview

Constructive Key Management is exactly what the name implies: key is constructed as needed by the originator of the message, and can only be re-constructed by intended recipients. In the interim, Credentials of the key components are associated with the encrypted object. For example, in an e-mail message, they might be passed, encrypted under a system key in the message header (depending on the protocol). In a session-oriented protocol, they might be exchanged as part of the key management protocol, and stored locally in the security management information base (SMIB). This means the encrypting key is always fully recoverable and the message is always decryptable by the appropriate audience.

There are two major administrative functions required to manage the CKM system: the CKM Administration (see §5.2 CKM Administration) and the Token Distribution (see §5.3 Token Distribution). In large organizations, these could be independent of each other. The CKM Administration function shall design the overall interconnectivity and read-write privileges in the system, and create the Credentials and splits. The Token Distribution function shall include the day-to-day management of the system, the creation, distribution and update of Credentials, and maintenance of a current users list (see Figure 1 - Token Distribution). The Token Distribution function shall be accomplished through a secure channel (see §5.3 Token Distribution).

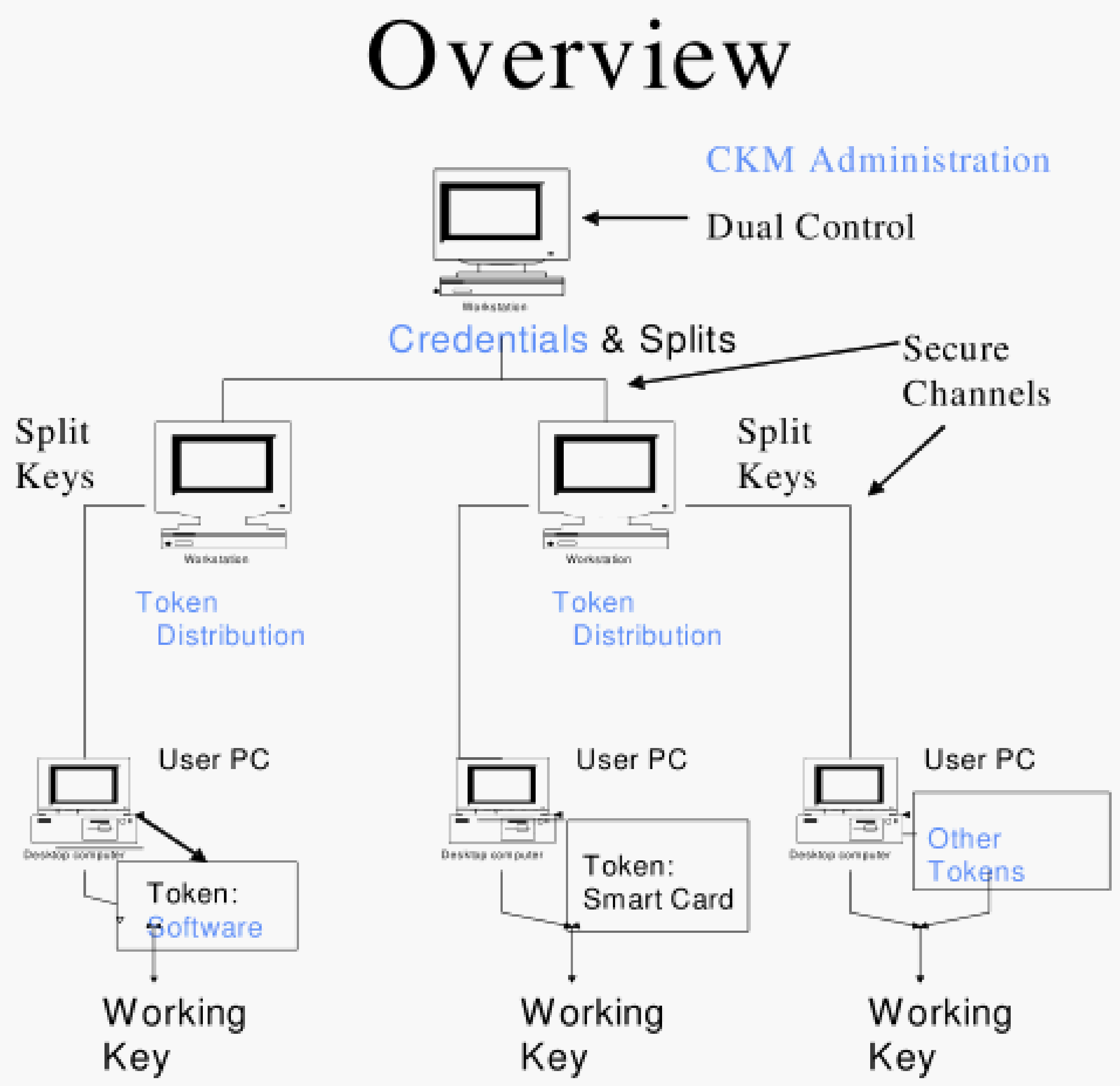

Figure 1 - Token Distribution

After the CKM Administration has defined the system and system parameters, except for those instances where major changes must be made (for instance, interaction with another organization that requires each Token Distribution to set up a common communication path), it basically lies dormant. It is at the Token Distribution level where Dual Control of key material shall be accomplished, if required in the system. On the other hand, the CKM Administration function is a daily management process.

Credentials consist of Splits, Algorithm-Access, and Administrative Information.

Credentials are the categories and sub-categories of addressees. In many applications they are humanly readable, so they are comprehensible to the sender who is determining the recipients of the encrypted object. There are also one or more system Credentials that are used in every encryption and decryption, and are transparent to the user.

Every Credential points to a unique Split, a secret, random number which is a component in the working key. After the object is properly addressed, (i.e. credentialed) the appropriate splits are combined to produce the working key for the encryption/decryption process.

Depending on the application, Algorithm-Access may be used to accomplish data separation and access control. For example, the CBC mode of 3DEA/AES is used in MACing, and ECB mode is called for in key management applications.

Administrative Information should control such things as user read-write privileges, and what devices in the system can be used by individuals and applications.

In addition to management functions, there are two essential cryptographic-related functions used by the system. These are: a) an encryption algorithm used to protect CKM system information as it transits a communication network; and b) a source for random numbers used to create the splits associated with the Credentials. This random source should be used as the "object unique random split" invoked with every encryption to ensure the working key is unique for each encrypted object (see §5.2.2.3 Random Split).

5.2 CKM Administration

The CKM Administration shall design the communications and cryptographic connections by creating the major addressable categories, and the sub-categories called Credentials. Examples of these "major categories" could be message_classification, addressee, department, external, etc. Under each major category, Credentials, or sub-categories, of addressees are defined. These Credentials are used to further define, or narrow, the audience for any particular message or object. For example, message_classification could have confidential, company_proprietary, internal_use_only, and general Credentials. Addressee could include vice_president, manager, supervisor, and staff Credentials. Department could include administration, EEO, R&D, Security, and Training Credentials. External could include the outside agencies or companies dealt with in a secure environment, such as Armored_Car_Inc, Bail_Bondsman, Stock_Broker Credentials, etc.

Most users will have subsets of the Credentials from each category, but there are instances where some users may not be given Credentials within all categories. For example, a newly hired employee may not have a need for the external category, and a security guard probably will not need access to an *R&D* Credential. Assigning the appropriate Credentials to each user is the responsibility of the CKM Administration.

5.2.1 Credentials

There are two kinds of Credentials used in the CKM system: fixed Credentials and user-selectable Credentials.

5.2.1.1 6.2.1.1 Fixed Credentials

The fixed Credentials point to the Shared Secret Key of the system. One of these fixed Credentials is the Organizational component. The organizational component is private to the organization, and ensures that outsiders cannot enter the system. This key component is used in every encryption. Other fixed components may be used: to update the system key when system administration deems it necessary; for archival_purposes, which could insure current users do not have access to older company information; or for other reasons deemed appropriate by the CKM Administration. All fixed Credentials are used in every encryption and decryption.

5.2.1.2 6.2.1.2 User Selectable Credentials

These selectable Credentials are names or addresses the encrypting entity uses to designate the recipients of the encrypted object. These would include cross-organizational Credentials when appropriate. Each selectable Credential shall have an associated split, i.e. a random, secret number that is combined with other splits to form the final working key for the encryption/decryption process. The Credentials, but not the splits, shall be associated with the encrypted object. Only recipients with all the Credentials have access to all the needed splits; only they can successfully reconstruct the key and perform the decryption.

The selectable Credentials are personal to each user or user group. They are maintained on a (physical or logical) token that is distributed by the CKM Administration. The token shall define the user's privileges, and is protected by a user changeable password. It may also be constrained so that it can only be used on specific sets of devices. The physical token is a portable device such as a floppy disk or a smart card, or a similar device that can be introduced into the workstation. The logical token is a file, resident on the system, and protected by a user-managed password.

5.2.2 Splits

The CKM Administration shall also create the *Splits*, the random numbers that become components of the working key for the encryption/decryption process. Each split is assigned to one of the fixed or selectable Credential described above. Splits are never seen nor transferred as part of a transaction; they are referenced by name Credential only.

5.2.2.1 Fixed Splits

Fixed splits are constant components in the encryption process and are the Shared Secret Key for the organization. They are used in each encryption/decryption, but still are named credentialed for those cases where cross-organization communication is required. For example the Organizational Key shall be a fixed split that defines the overall membership of the system. Everyone with access to the system shall have this key component; and conversely, without this component, access is denied to all encrypted objects.

5.2.2.2 Selectable Splits

Selectable splits are optional components that are distributed among the user community. These are the components in the address of the encrypted object that ensure the secrecy of the object from unauthorized readers.

5.2.2.3 Random Split

The random split is the only unnamed split. It shall be created for every generated key. Each generated key has a different random split which could be generated by a) an ANSI key establishment mechanism, or b) an ANSI approved random or pseudorandom number generator.

The Random Number component of the key shall be generated on a per object basis; that is with every new encryption. This ensures that every working key is unique. Even if periodic messages are sent to the same set of Credentials, this component will guarantee the uniqueness of the working key. The CKM Administration determines how this random split is generated.

5.3 Token Distribution

CKM Administration shall distribute tokens to appropriate users. The distribution of these tokens along with an initial password may be accomplished via personal visit or by an automated distribution process.

5.3.1 Workstation

The CKM Administration shall install, or cause to be installed, the software necessary to configure and enable a workstation. The software package includes the Random Number Generator and the Organizational Key.

5.3.2 Token

The Token shall be a user-controlled component of the system. It shall contain the Credentials and splits assigned to that user, and other information the particular user needs for the various applications. The token may be a floppy disk, a smart card, or some other device than can be manually introduced to the workstation for authentication and keying usage, and then removed and securely stored by the user. All the information on the token is securely wrapped by a fixed system key, and is accessed by a user-invoked (and user-changeable) password. The token need not be transportable as it may be a password-protected file on the user's workstations(s).

5.4 Key Creation

The user creates a key at the time of encryption and decryption by combining the appropriate splits invoked by the Credentials associated with each transaction. In most applications, after the key is used, it is destroyed, and except for the random key split, only pointers, i.e. the Credentials, of the key components are retained. Knowledge of the Credentials employed in a particular cryptologic application gives no information toward deducing the final key or key components.

There may be times when it is more efficient for an application to retain the key, instead of having to recreate it on demand. A key with an extensive use or a predefined life cycle, such as a key used to verify MACs, may fall into this category. At the appropriate time a message can be sent to all authorized users, which will issue a new MAC_verification key.

5.4.1 Key Component Selection

At a minimum, there shall always be two components in the working key; these are the fixed split(s) of the system and a random split that will change with every encryption. If these are the only splits, then all valid system users on the system can decrypt the object. (This may be an announcement of a general nature that is to be kept private within the organization.) As user-selectable splits are added from the token, the audience of the message is reduced; that is, the originator can be more restrictive as to the readership. These personalized splits are used to accomplish data separation across the network. The random split is required so that even if messages are routinely addressed to the same readership, the random component guarantees the working key will be unique to that message.

5.4.2 Key Combiner

The Combining Function in the software takes the Fixed Split(s), the Random Split, and the User-selected Splits from the Token, and combines them to produce the Working Key for the current object, be it for session, e-mail (store and forward), or file encryption applications. Typically, this combining function will be a non-linear function of all the components (see Figure 2 - Combiner Function).

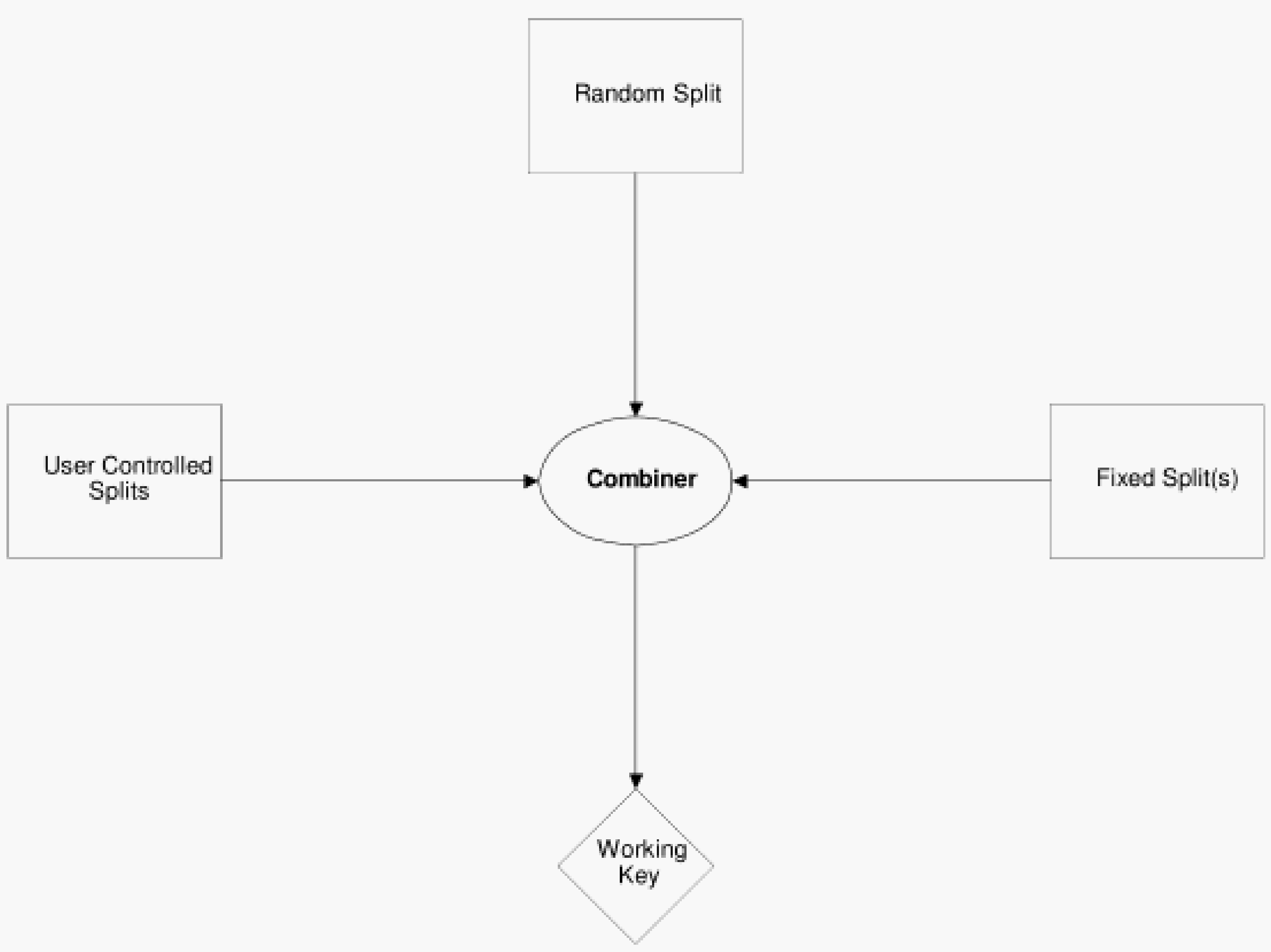

Figure 2 - Combiner Function

5.4.3 Key Reconstruction

The encrypted object (message or file) should contain a header that is also encrypted using the system algorithm with the constant system key which all users have. This header shall contain administrative information (e.g. the identity of the token used in the encryption, file length, date_time encryption information, etc.), the random number component, and all the Credentials used to pick and form the key. Since the Credentials are not the splits, but only pointers to the splits, only those users with access to all the proper Credentials can reconstruct the working key and perform the decryption.

For session-oriented applications, during session start-up, or when a message header is not used, all key reconstruction information shall be made available to the recipients by other means, such as SMIB or a secondary secure channel.

6 Key Usage Control

6.1 Overview

The key usage control procedure allows the creator of a key to specify the allowed use of the key. For example, key usage control information is used to distinguish key types (e.g., data key, key-encrypting key, PIN encrypting key). The usage control information is bound to the key and is used by the system as a means to ensure that the key is used properly. [1], [2], [3]

In a key management system with key usage control, each key K shall have an associated key usage vector C. This key usage vector shall specify the cryptographic services, including individual modes and roles, in which the key is permitted to be used. With key usage control, keys can be generated using any ANSI approved key generation or key establishment Standards.

The cryptographic services within the cryptographic system are invoked via an Application Programming Interface (API). These services, defined as $S_1, S_2, \ldots, S_n$, include, but are not limited to, data-operation services, key management services, and PIN management services. Each cryptographic service can have multiple modes and key parameters. For example, C might specify that K can be used as the second key parameter in cryptographic service S when mode M is specified.

To prevent abuses and attacks on the key K, the key usage vector C shall be bound to the key K using one of the six alternative binding methods provided by this Standard. In each case, the binding method shall satisfy the conditions below:

- If a cryptographic service is invoked, and all key usage vectors required by cryptographic service are provided and allow the requested service to be performed, then the requested service can be executed.

- If a cryptographic service is invoked with an incorrect value of C, then either;

    a. The incorrect value of C is detected and the requested service is aborted, or

    b. The requested service is executed and the incorrect value of C causes an incorrect and spurious result to be produced.

NOTE: In practice, the latter condition is easily satisfied by storing keys in encrypted form and making the decryption process dependent on C. In that case, if an incorrect value of C is specified to the decryption process, the decryption process will in turn cause a spurious key value K' to be recovered instead of the correct key value K.

A key usage vector shall contain a TYPE field and a USAGE field (see Figure 3 - Key Usage Vector Fields), where the TYPE field specifies the type of key and the USAGE field specifies the use of key. The USAGE field shall consist of a set of one or more 1-bit fields ($U_1, U_2, \ldots, U_k$), where each bit relates to;

a. a key parameter in a cryptographic service, or

b. a key parameter in a particular mode of a cryptographic service.

Type	Usage $U_1, U_2, \ldots U_K$

U_i: refers to a key parameter of a cryptographic service or particular mode of a cryptographic service, and hence indirectly to the defined usage of the key parameter in the cryptographic service or particular mode of the cryptographic service.
$U_i = 1$: yes, the key can be used
$U_i = 0$: no, the key cannot be used

Figure 3 - Key Usage Vector Fields

6.2 Key Binding Methods

In the following sections the six alternative key binding methods are described. Binding method 1 is not cryptographic and relies on the protection provided by the cryptographic system, whereas binding methods 2-6 are cryptographic.

6.2.1 Binding Method 1

With binding method 1, a copy of the key (K) and a copy of the key usage vector (C) are stored together, as a single key record (K,C), inside the secure boundary of the cryptographic system. Binding method 1 is effective as long as the cryptographic system is able to protect the integrity of the key usage vector (C) and the secrecy and integrity of the key (K).

NOTE: Binding method 1 is non-cryptographic.

NOTE: This Standard does not specify how the key record (K,C) is addressed or specified to the cryptographic service. This could be done using any convenient means of addressing, e.g., an index value, a pointer, or Credential.

6.2.2 Binding Method 2

With binding method 2, the key K is stored outside the secure boundary of the cryptographic system under the encryption of a variant key value (KK'). The variant key value (KK') is derived from a key value (KK). KK is a key-encrypting key, e.g., a system master key belonging exclusively to the cryptographic system or a key shared by the cryptographic system with another cryptographic system.

The variant key (KK') is calculated as a function (F) of the key usage vector (C) and a given key-encrypting key (KK). That is,

KK' = F(KK,C)

The binding is then accomplished by encrypting K with the variant key KK' to produce the encrypted key value eKK'(K). The encrypted key value eKK'(K) together with C is stored as a single key record.

With this binding method, the key record (eKK'(K),C) need not be stored inside the secure boundary of the cryptographic system. If a cryptographic service is invoked with an incorrect value of C, for some key K, then the output of the service is incorrect and spurious. This is because K is stored in encrypted form and the decryption process is dependent on C. Therefore, if an incorrect value of C is specified to the decryption process, the decryption process will in turn calculate an incorrect value of KK' and decrypt eKK'(K) with the incorrect value of KK' thus causing a spurious key value K' to be recovered instead of the correct key value K.

NOTE: This Standard does not specify function F. F could be a simple operation, such as the Exclusive-OR operation (KK' = KK xor C), or it could be a complex cryptographic function that involves encryption or hashing operations.

NOTE: This Standard does not specify the cryptographic algorithm to be used in the calculation of the encrypted key value eKK'(K). But it does require the cryptographic algorithm to be a symmetric cryptographic algorithm since for an asymmetric cryptographic algorithm, if PU and PR are a public and private key pair, and C is a key usage vector, then the variant values (PU xor C) and (PR xor C) will, in general, not be a valid public and private key pair.

6.2.3 Binding Method 3

With binding method 3, a non-secret authentication code (AC) is calculated on K and C using an authentication code algorithm (AuthAlg). That is,

AC = AuthAlg(K,C)

The authentication code (AC) is stored together with K and C in a single key record (AC,K,C). The authentication code binds C to K. AuthAlg is a function such that AuthAlg(K,C) does not reveal information about K. [4]

With this binding method the key usage vector (C) and the authentication code (AC) need not be stored inside the secure boundary of the cryptographic system. If a cryptographic service is invoked with an incorrect value of C, for some key K, the incorrect value of C is detected. This is accomplished by computing AuthAlg(K,C) using the received C and comparing the result against the received AC. NOTE: if (AC,K,C) is stored outside the secure boundary of the cryptographic system, then K must be protected; e.g., by storing it under the encryption of a key known to the cryptographic system.

NOTE: This Standard does not specify the authentication code algorithm AuthAlg.

6.2.4 Binding Method 4

With binding method 4, a non-secret digital signature (DS) is calculated on f(K) and C using a digital signature algorithm (SigAlg) and the private key (PR) of a public key algorithm. That is,

DS = SigAlg(PR,f(K),C)

The digital signature (DS) is stored together with K and C in a single key record (DS,K,C). The digital signature binds C to K. f is a public function such that f(K) does not reveal information about K, e.g., f(K) could be computed by encrypting K with a public key of a public key algorithm.

With this binding method, the key usage vector (C) and the digital signature (DS) need not be stored inside the secure boundary of the cryptographic system. If a cryptographic service is invoked with an incorrect value of C, for some key K, the incorrect value of C is detected. This is accomplished using the public key of the public key algorithm as a verification key to verify the signature. Signature verification also indirectly verifies the values of K and C. NOTE: if (DS,K,C) is stored outside the secure boundary of the cryptographic system, then K must be protected; e.g., by storing it under the encryption of a key known to the cryptographic system.

6.2.5 Binding Method 5

With binding method 5, a non-secret message authentication code (MAC) is calculated on f (K) and the key usage vector (C) using a MAC-generation mode of an encryption algorithm (MacAlg) and a secret MAC key (K-MAC). That is,

MAC = MacAlg(K-MAC,f(K),C)

The message authentication code (MAC) is stored together with K and C in a single key record (MAC,K,C). The generated MAC binds C to K. f is a public function such that f(K) does not reveal information about K, e.g., f(K) could be computed by encrypting K with a public key of a public key algorithm and then hashing the encrypted key valued using a hash function.

With this binding method the key usage vector (C) and the message authentication code (MAC) need not be stored inside the secure boundary of the cryptographic system. If a cryptographic service is invoked with an incorrect value of C, for some key K, the incorrect value of C is detected. This is accomplished by calculating a MAC using the received C and comparing the result against the received MAC. NOTE: if (MAC,K,C) is stored outside the secure boundary of the cryptographic system, then K must be protected; e.g., by storing it under the encryption of a key known to the cryptographic system.

NOTE: This Standard does not specify the MAC-generation algorithm (MacAlg) to be used in calculating the MAC. One possibility is to use the 3DEA/AES and the MAC generation procedure defined in ANS X9.19.

6.2.6 Binding Method 6

With binding method 6, the key (K) and the key usage vector (C) are encrypted with the public key of a public key algorithm. The encrypted key value ePU(K,C) is stored as a single key record.

NOTE: For the binding method to be effective, it must be infeasible to exhaustively determine K using forward encryption of trial values of K under the public key.

With this binding method the key record ePU(K,C) is non-secret and need not be stored inside the secure boundary of the cryptographic system. If a cryptographic service is invoked with an incorrect value of C, for some key K, the incorrect value of C is detected. This is accomplished using the private key (PR) corresponding to PU, belonging to the receiving or verifying entity. That is, ePR(ePU(K,C)) is calculated to obtain K and C from the key record. The received key usage vector is valid if the recovered C is equal to the received C.

Annex A
(informative)

Example Key Usage Vector Formats

A.1 General

The appendix provides several examples of key usage vectors. Each key usage vector has a length of 64 bits. The bits in the key usage vector are numbered 0 to 63, from most significant to least significant bit position.

A.2 Examples

Each key usage vector contains a 7-bit TYPE field (bit positions 8-14). The TYPE field contains a 4-bit MAIN-TYPE field (bit positions 8-11) and a 3-bit SUB-TYPE field (bit positions 12-14).

Each key usage vector contains a 7-bit USAGE field (bit positions 16-22).

The example depicts key usage vectors that support three main types--based on the kind of cryptographic service to be performed--as follows:

MAIN TYPE	Description
B'0000'	Data Key (used in data-operation services)
B'0010'	PIN Key (used in PIN-management services)
B'0100'	Key-Encrypting Key (used in key-management services)

Data keys are further divided into two subtypes:

SUB TYPE	Description
B'001'	Privacy Key (used to encrypt and decrypt data)
B'010'	MAC Key (used to generate and verify Message Authentication Codes)

Key-encrypting keys are also further divided into two subtypes:

SUB TYPE	Description
B'000'	Exporter Key (used to encrypt and transmit keys to another cryptographic system)
B'001'	Importer Key (used to decrypt and receive keys from another cryptographic system)

Dividing key-encrypting keys into exporter and importer key types has the advantage that the keys become uni-directional. That is, the key can be used to establish a key distribution channel in one direction only. Thus, keys encrypted under an exporter key cannot be re-imported into the sending cryptographic device. This feature permits a device to generate and export keys without necessarily having a capability to use the keys.

In the key usage specification that follows we assume the existence of a cryptographic system that provides the following cryptographic services: Encipher, Decipher, MAC Generate, MAC Verify, Key Export, Key Import, and Translate Key. These services are explained below.

A possible key usage vector specification for a Privacy key is the following:

Bits	Description	Specification
08-11	Main Type = 'data key'	B'0000'
12-14	Sub-type = 'privacy'	B'001'
18	Encipher	B'1' : this key can be used in an Encipher Service to encipher data. B'0' : this key cannot be used in an Encipher Service to encipher data.
19	Decipher	B'1' : this key can be used in a Decipher Service to decipher data. B'0' : this key cannot be used in a Decipher Service to decipher data.

A possible key usage vector specification for a MAC key is the following:

Bits	Description	Specification
08-11	Main Type = 'data key'	B'0000'
12-14	Sub-type = 'MAC'	B'010'
20	MAC Generate	B'1' : this key can be used in a MAC Generate Service to generate MACs. B'0' : this key cannot be used in a MAC Generate Service to generate MACs.
21	MAC Verify	B'1' : this key can be used in a MAC Verify Service to verify MACs. B'0' : this key cannot be used in a MAC Verify Service to verify MACs.

A possible key usage vector specification for an Exporter key is the following:

Bits	Description	Specification
08-11	Main Type = 'key- encrypting key'	B'0100'
12-14	Sub-type = 'exporter'	B'000'
21	Key Export (reenciphers a key from encryption under the Master key to encryption under an Exporter key)	B'1' : this key can be used in a Key Export Service. B'0' : this key cannot be used in a Key Export Service.
22	Translate Key (reenciphers a key from encryption under an Importer key to encryption under an Exporter key)	B'1 : this key can be used as an Exporter key in a Translate Key Service. B'0' : this key cannot be used as an Exporter key in a Translate Key Service.

NOTE: In the example, we assume that keys stored locally in a cryptographic device are encrypted under a Master key stored within the protected boundary of the cryptographic device.

A possible key usage vector specification for an Importer key is the following:

Bits	Description	Specification
08-11	Main Type = 'key- encrypting key'	B'0100'
12-14	Sub-type = 'importer'	B'001'
21	Key Import (reenciphers a key from encryption under an Exporter key to encryption under the Master key)	B'1' : this key can be used in a Key Import Service.

		B'0' : this key cannot be used in a Key Import Service.
22	Translate Key (reenciphers a key from encryption under an Importer key to encryption under an Exporter key)	B'1' : this key can be used as an Importer key in a Translate Key Service. B'0' : this key cannot be used as an Importer key in a Translate Key Service.

Bibliography

[1] Matyas, S.M.; Key Processing with Control Vectors; Journal of Cryptology-3, 113-136 (1991)

[2] Matyas, S.M.; Key Handling with Control Vectors; IBM Systems Journal 30, No2, 151-174 (1991)

[3] Matyas, S.M., Le, A.V., Abraham, D.G.; A Key-management Scheme Based on Control Vectors;

[4] IBM Systems Journal 30, No. 2, 175-191 (1991)

[5] Canetti, R: Toward Realizing Random Oracles: Hash Functions that Hide All Partial Information; Advances in Cryptology - Crypto '97, LNCS 1294, Springer-Verlag, 1997, pp 455-469

[6] Stapleton, J; Poore, R; Cryptography Transitions; IEEE Region 5 Conference, 1-4244-0359-6/06 ©2006 IEEE