Document Number: XXXX-01-04-October 2004

Draft Proposal for Key Backup Format for Wide-block Encryption
Draft 1.04:00

September 24, 2004

Sponsor
IEEE P1619

Abstract: This document describes a format for specifying key material and related parameters which are necessary for backup and recovery of encrypted data on storage media that has been encrypted with wide-block encryption transform from the IEEE P1619 family. This key backup format allows interoperability among different vendors of encrypted storage.

Keywords: Encryption, Storage, Key-Backup.

Contacts

This proposal editor:
Dalit Naor
IBM Haifa Research Center.
Haifa University Campus
Mount Carmel, Haifa 31905 ISRAEL
Tel: +972.3.640.1625
Email: dalit@il.ibm.com

James Hughes 10/19/04 10:58 AM
Deleted: 7

1. Revision History

Revision 0.01 – Feb 16 2004
Revision 0.02 – Feb 17 2004
Revision 1.01 – March 30, 2004
- ✓ Incorporating changes discussed in IEEE P1619 meeting of Feb 26, 04:
- ✓ Use "key backup"
- ✓ Update fields

Revision 1.02 – April 13, 2004
- ✓ Remove Key Origin
- ✓ Vague reference to key wrapping in XML

Revision 1.03 – April 14, 2004
- ✓ Minor editorial changes

Revision 1.04 – October 03, 2004
- ✓ Format of Standard Number
- ✓ Scope of key in units of bytes
- ✓ Removed definition of Tweak (to reside in transform document)
- ✓ Removed EME-1-AES
- ✓ Allow for vendor specific parameters
- ✓ Add a section on XML Format and a DTD
- ✓ Add a section on Encrypting the key backup structure and reference KEK standards

2. Preface

The IEEE P1619 family of standards specify the architecture for protection of user data in sector-level storage devices. The purpose of P1619 related documents is to describe cryptographic algorithms and methods for encrypting data before it is sent to storage (disk or tape). It also includes modes and formats to be used in data protection to create interoperable solutions.

IBM_USER 9/26/04 1:34 AM
Comment: Page: 2
A first attempt to define a standard preface for all P1619 documents.

3. Motivation

Encryption of data-at-rest on the storage media (whether it is a hard drive or tape) poses a serious problem of recoverability and interoperability. Once the data has been stored encrypted, there must be a way to recover (i.e. decrypt) the data in the future, possibly by a vendor/application that is different from the one that had originally encrypted it. Moreover, the recovery may happen far later in time than the encryption. This calls for a standard key backup mechanism.

The IEEE P1619 family of standards for data-at-rest encryption defines an encryption mechanism, the Tweakable Block Encryption [EME, TWEAK], but does not specify the meta-data format of the stored data. This document proposes a standard Key Backup format for the key material that was used to encrypt stored data. This unified

format may allow vendors to exchange key material if desired, to facilitate future recoverability and interoperability.

The Key Backup format will be defined in XML, to facilitate a unified format as well as the automatic generation and parsing of Key Recovery documents.

4. Glossary and Conventions

AES	Advanced Encryption Standard
base64	The Base64 Content-Transfer-Encoding
Bit	A binary digit
Byte	A group of eight bits
FIPS	Federal Information Processing Standards
LBA	Logical Block Address, specified in bytes
Sector	512 bytes of data
Wide-Block Encryption	An IEEE Encryption mode specific for data-at-rest encryption
Wide-block	Unit of the storage encryption (aka Blob)

5. The Key Backup Information

The Key Backup structure must provide all the information that is needed in order to decrypt an integral number of wide blocks (blobs) that were encrypted with a tweakable wide-block encryption scheme. These must be an ordered sequence of wide blocks, numbered consecutively on a storage media. The sequence of wide blocks that are associated with a specific Key Backup structure (in particular, a specific key) is called the SCOPE of the Key Backup structure.

The key backup structure consists of the following elements:

Field	Format	Length[1]	Description
ID	base64	128 bits	General identifier for the key recovery structure.
COMMENT	Text		A free text description provided by the vendor.

[1] Length indicates an upper limit on the length of the Value, and not necessarily the actual representation length. For example, if Length= 96 bits then this field may take any value between 0 and 2^{96}-1.

STANDARD_NUMBER	Text		String of the form X.Y
STANDARD_VERSION	base64	8 bits	
STANDARD_COMMENT	Text		Any free text relevant to the encryption standard.
KEY_SCOPE_START	base64	96 bits	The LBA (in bytes) of the first wide-block in the SCOPE of the key backup structure.
KEY_SCOPE_LENGTH	base64	96 bits	The number of bytes in the SCOPE of the key backup structure.
TRANSFORM_NAME	Unique name		A reserved string identifier for a wide-block encryption scheme.
KEY_PRIMARY_LENGTH	base64	8 bits	Length (in bits) of the symmetric block cipher key.
KEY_SECONDARY_LENGTH	base64	8 bits	Length (in bits) of the second key - when applicable.
KEY_ PRIMARY	base64		The actual symmetric block cipher key.
KEY_SECONDARY	base64		The actual second key, when applicable
OPTIONAL_PARAMETER	Text		In (Attribute, Value) format

Clement Kent 9/30/04 6:10 PM
Comment: I've forgotten why we chose 96 here. For LRW, I think it can go up to 120. Ask Shai if there is some reason why EME needs 96. Same comment applies to SCOPE length

5.1 Document Information

The ID and COMMENT fields may be used to identify a particular Key Backup structure.

5.2 Standard-Related Information

The STANDARD_NUMBER field specifies the IEEE standard defining this key backup format. The STANDARD_NUMBER is specified as a string of the form 'string1.string2'.

The STANDARD_VERSION field specifies the version of the STANDARD_NUMBER.

Any additional standard-related information may be specified in the STANDARD_COMMENT. This includes the type of storage for which this key backup information applies.

Dalit Naor
Comment: Page: 4
Do we want a special field for that named 'STORAGE_TYPE', or is it enough to put it in the COMMENT?

5.3 Scope of the Key Backup Structure

KEY_SCOPE_START and KEY_SCOPE_LENGTH determine the scope of the key that is identified in the key backup structure. The scope is an ordered sequence of KEY_SCOPE_LENGTH bytes, numbered consecutively starting at byte KEY_SCOPE_START. Both are 96-bits integers. Different transforms may have different interpretations for these integers.

Note that a given key-backup structure can not be used to decrypt wide blocks that are not in its scope, namely either smaller than KEY_SCOPE_START or greater or equal to KEY_SCOPE__START + KEY_SCOPE__LENGTH.

5.4 Transform Parameters

The TRANSFORM_NAME string identifies the block-level encryption and must be one of the following values:

String	Description
EME-32-AES	IEEE P1619 EME transform that is based on an AES cipher, with R=32 [EME] (wide block of 512 bytes)
LRW	The IEEE P1619 tweak algorithm that is based on an AES cipher [TWEAK]

The KEY_PRIMARY_LENGTH field defines the length (in bits) of the symmetric block cipher key. For example, for AES the key may be of length 128,192 or 256 bits.

The KEY_SECONDARY_LENGTH field specifies the length (in bits) of the tweak key when such a key is applicable (for example, for an LRW transform).

The ADDITIONAL_PARAMETERS field provides a way to extend the structure parameters for other transforms beyond what is currently needed for EME and LRW.

5.5 Keys

KEY_PRIMARY is the symmetric block cipher key and is KEY_PRIMARY_LENGTH bits long.

KEY_SECONDARY is the second (tweak) key and is KEY_SECONDARY_LENGTH bits long. This field is relevant for some transforms only, such as LRW. For all other transforms, its value should be 0x00. For EME-32-AES or EME-1-AES transforms, the L_VALUE will be fixed and set to $L=2 \cdot AES_k(0)$, and hence is not specified in the format. If needed in the future, it may be specified in the KEY_ SECONDARY field.

5.6 Optional Parameters

The OPTIONAL_PARAMETER field allows the specification of additional parameters to the key backup structure. These parameters may either be defined by the standard or user-defined attributes. Parameters that are defined by the P1619 standard (and described in other P1619 documents, e.g. a particular transform document) must be assigned an attribute name beginning with the string 'P1619'. Other, vendor specific attributes, should start with the vendor name and MUST NOT start with the string 'P1619'.

6. XML Format

The Key Backup structure is defined in XML, to facilitate a unified format and allow an application independent way of sharing key material. This also provides an automatic generation and parsing of Key backup structures. A DTD (Document Type Definition) for the P1619 Key Backup format will be published along with the other P1619 standard documents.

Below is an example of a DTD:

```
<?xml
version="1
.0"
encoding=
"ISO-
8859-1"
?>

<!DOCTYP
E
```

7. Encryption of Key Backup material

A Key backup structure is typically a high value structure. It may therefore be protected as follows.

1. The actual keys (KEY PRIMARY and KEY SECONDARY) will be wrapped encrypted with xmlenc [XML-ENC] and embedded within the XML key backup structure. [XML-ENC] does not mandate any single key wrapping algorithm, but to be compliant with this standard (P1619) vendors shall support NIST AES 256 Key Wrap (see http://www.w3.org/2001/04/xmlenc#kw-aes256). Other key wrap algorithms allowed by [XML-ENC] may be used.
2. The keys that are used to wrap the key elements (kek) may be referenced with xkms [XML-KMS]. The location of wrapping keys is not specified by the P1619 standard. The cryptographic strength of wrapping keys should be equivalent to the strength of the storage encryption keys wrapped (see [KEY-MGMT]).
3. Vendor should provide integrity to the key file, using standard methods.

Dalit Naor 9/26/04 12:38 AM
Comment: Page: 1
Needs to be discussed.

These protections (encryption and integrity) must be implemented by the vendor. However, the use of these mechanisms on a particular instance is optional.

An example of the XML encoding of the KeyPrimary or KeySecondary fields is:

```
<KeyPrimary>
  <EncryptedKey>
    Type='http://www.w3.org/2001/04/xmlenc#Content'
    xmlns="http://www.w3.org/2001/04/xmlenc#"
    <EncryptionMethod
      Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes256"/>
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:KeyName>WrapKey</ds:KeyName>
    </ds:KeyInfo>
    <CipherData>  <CipherValue>B457V645B45.....</CipherValue> </CipherData>
  </EncryptedKey>
</KeyPrimary>
```

In thie above example, the KeyPrimary value is encrypted using AES-256 Key Wrap, whose wrapping key has identifier “WrapKey”.

An alternative format allowed by [XML-ENC] would use a wrapping key reference, rather than a wrapping key name:

```
<KeyPrimary>
  <EncryptedKey>
    <EncryptionMethod
      Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes256"/>
    <ds:KeyInfo > <ds:RetrievalMethod URI='#WK'> </ds:KeyInfo>
    <CipherData>  <CipherValue>B457V645B45.....</CipherValue> </CipherData>
  </EncryptedKey>
</KeyPrimary>
```

If a wrapping key reference is used as above the wrapping key would be stored as an EncryptedKey element elsewhere in the document with ID='WK'.

8. Further Extensions

Transform names will be given global string identifiers.

9. Bibliography and References

[HR03] S. Halevi and P. Rogaway. “A tweakable enciphering mode.” In *Advances in Cryptology – CRYPTO '03*, volume 2729 of Lecture Notes in Computer Science, pages 482-499. Springer-Verlag, 2003.

[HR04] S. Halevi and P. Rogaway. “A parallelizable enciphering mode.” The RSA conference - Cryptographer's track, RSA-CT '04. LNCS vol. 2964, pages 292-304. Springer-Verlag, 2004.

[LRW02] M. Liskov, R. Rivest and D. Wagner. Tweakable Block Ciphers. In *Advances in Cryptology – CRYPTO '02*. Lecture Notes in Computer Science. Springer-Verlag. 2002. www.cs.berkeley.edu/~daw/.

[EME] Shai Halevi, Draft Proposal for Tweakable Wide-Block Encryption, March 2003.

[**KEY-MGMT]** NIST Key Management Guideline. http://csrc.nist.gov/CryptoToolkit/kms/key-management-guideline-(workshop).pdf

[TWEAK] Cyril Guyot, IEEE Tweak, Dec 2003

[**XML-ENC]** XML Encryption Working Group. XML Encryption Syntax and Processing. Dec 2002. http://www.w3.org/TR/xmlenc-core/

[**XML-KMS]** XML Key Management Working Group. XML Key Management Specification. http://www.w3.org/2001/XKMS/Drafts/XKMS-PR-DRAFT/PR-DRAFT-xkms-part-1.html